which remote file transfer 1958 (e.g. downloading and uploading) and remote communications such as Internet browsing can be made through access points to the fixed network.local wireless audio streaming.

It should be noted, however, that certain communications systems, such as many modes of telephony, do not distinguish between mobile communications and communications through fixed access points, and that both file transfer 1958 and audio streaming 1954 can be available through the same mode. Even in those cases, however, it can be convenient to have two modes in order to make optimal use of the advantages of the different modes. In such cases, however, the two modes can alternatively be supported by multiple hardware and software systems within the same device – for example, for remote communications to be made through a telephony system (e.g. GSM or CDMA), while the local audio streaming 1954 can be made through a parallel communications system (e.g. Bluetooth or 802.11) – indeed, the two systems can operate simultaneously with one another.

Inter-unit Transmission Segmentation

Preferably, the broadcast unit 710 and the receive units 730 exchange information in addition to the audio signal. For example, each user preferably has indications as to the number of total units (broadcast units 710 and receive units 730) within a cluster, since the knowledge of cluster 700 sizes is an important aspect of the social bond between the users. This also will help search units 750 that are not part of the cluster determine which of the clusters 700 that might be within their range are the most popular.

The additional information shared between members of a cluster 700 would include personal characteristics that a person might allow to be shared (images, names, addresses, other contact information, or nicknames). For example, the broadcast unit 710 will preferably, along with the music, transmit their nickname, so that other users will be able to identify the broadcast unit 710 for subsequent interactions, and a nickname is significantly easier to remember than a numerical identifier (however, such numerical identifier can be stored in the unit 100 for subsequent searching).

Such additional information can be multiplexed along with the audio signal. For example, if the audio signal is transferred as an MP3 file, assuming that there is additional bandwidth beyond that of the MP3 file itself, the file can be broken into pieces, and can be interspersed with other information. Figure 7 is a schematic diagram of a broadcast unit 710 transmission 820. The transmission is comprised of separate blocks of information, each represented in the figure as a separate line. In the first line, a block code 800 is transmitted,

which is a distinctive digital code indicating the beginning of a block, so that a search unit 750 receiving from the broadcast unit 710 for the first time can effectively synchronize itself to the beginning of a digital block. Following the block code 800 is a MP3 block header 802, which indicates that the next signal to be sent will be from a music file (in this case an MP3 file). The MP3 block header 802 includes such information as is needed to interpret the following block of MP3 file block 804, including such information as the length of the MP3 block 804, and characteristics of the music (e.g. compression, song ID, song length, etc.) that are normally located at the beginning of a MP3 file. By interspersing this file header information at regular intervals, a user can properly handle music files that are first received in the middle of the transmission of an MP3 file. Next, the MP3 block 804 containing a segment of a compressed music file is received.

Dependent on the amount of music compression and the bandwidth of the inter-unit communications, other information can be sent, such as user contact information, images (e.g. of the user), and personal information that can be used to determine the "social compatibility" of the user with the broadcast unit 710 and the receive unit 730. This information can be sent between segments of MP3 files or during "idle" time, and is generally preceded by a block code 800, that is used to synchronize transmission and reception. Next, a header file is transmitted, which indicates the type of information to follow, as well as characteristics that will aid in its interpretation. Such characteristics could include the length of information, descriptions of the data, parsing information, etc. In Figure 7, an ID header 806 is followed by an ID block 808, which includes nicknames, contact information, favorite recording artists, etc. Later, an image header 810 can be followed by an image block with an image of the user. The image header 810 includes the number of rows and columns for the image, as well as the form of image compression.

It should be understood that the communications format described in Figure 7 is only illustrative of a single format, and that a large number of different formats are possible within the present invention. Also, the use of MP3 encoding is just an example, and other forms of digital music encoding are within the spirit of the present invention, and can alternatively comprise streaming audio formats such as Real Audio, Windows Media Audio, Shockwave streaming audio, QuickTime audio or even streaming MP3 and others. Furthermore, these streaming audio formats can be modified so as to incorporate means for transmitting DJ 200 control signals and other information.

LINKPLAY EXHIBIT 1004
Page 2906 of 4822

Transmitting Dynamic Data and Control Information

As described above, there are benefits to two-way communications between the broadcast unit 710 and the receive unit 730. There are many methods of carrying out this communication, even if the inter-unit transmitter/receiver 110 does not permit simultaneous transmission and reception. For example, additional transmission and reception hardware could be included in each unit 100. Alternatively, in the transmission 820 above, specific synchronization signals such as the block code 800 can be followed by specific intervals during which the inter-unit transmitter/receiver 110 that is transmitting switches into receive mode, while the inter-unit transmitter/receiver 110 that was receiving switches to transmit mode. This switch in communications direction can be for a specific interval, or can be mediated through conventional handshake methods of prior art communications protocols.

It should be noted that in addition to transfer of static information (e.g. identifiers, contact information, or images), dynamic information and control information can also be transferred. For example, the user at the receive unit 730 can be presented with a set of positive and negative comments (e.g. "Cool!" "This is awful!") that can be passed back to the broadcast unit 710 with the press of a button. Such information can be presented to the user of the broadcast unit 710 either by visual icon on, for example, an LCD screen, by a text message on this screen, or by artificial voice synthesis generated by the broadcast unit 710 and presented to the user in conjunction with the music.

Alternatively, the user of the receive unit 730 can speak into a microphone that is integrated into the receive unit 730, and the user voice can be sent back to the broadcast unit 710. Indeed, the inter-unit communications can serve as a two-way or multi-way communications method between all units 100 within range of one another. This two-way or multi-way voice communication can be coincident with that of the playing of the audio entertainment, and as such, it is convenient for there to be separate amplitude control over the audio entertainment and the voice communication. This can be implemented either as two separate amplitude controls, or alternatively as an overall amplitude control, with a second control that sets the voice communications amplitude as a ratio to that of the audio entertainment. In this latter mode, the overall level of audio output by the unit is relatively constant, and the user then selects only the ability to hear the voice communication over the audio entertainment.

In order to express their feelings and appreciation about the music they are hearing, users within a cluster 700 can also press buttons on their units 100 that will interrupt or

LINKPLAY EXHIBIT 1004
Page 2907 of 4822

supplement the control signals being sent to their respective DJs 200, providing light shows that can be made to reflect their feelings. For example, it can be that all lights flashing together (and not in synchrony with the music) can express dislike for music, whereas intricate light displays could indicate pleasure.

It is also possible to send control requests between units 710. For example, a receive unit 730 can make song requests (e.g. "play again", "another by this artist") that can show on the broadcast unit 710 user interface. Alternatively, the user of a receive unit 730 can request that control be switched, so that the receive unit 730 becomes the broadcast unit 710, and the broadcast unit 710 becomes a receive unit 730. Such requests, if accepted by the initial broadcast unit 710 user, will result in the memory storage of the identifier of the broadcast unit 710 being set in all units in the cluster 700 to that of the new broadcast unit 730. Descriptions of the communications resulting in such a transfer of control will be provided below.

Additionally, it is also possible for users of units 100 to privately "chat" with other users while they are concurrently receiving their audio broadcasts. Such chat can be comprised of input methods including keyboard typing, stylus free-form writing/sketching, and quickly selectable icons.

It should be understood that within the spirit of the present invention that the functional configuration can be supported by the extension of certain existing devices. For example, the addition of certain wireless transmitter and receiver, as well as various control and possibly display functionality to a portable audio player would satisfy some embodiments of the present invention. Alternatively, by the addition of music storage and some wireless transmitter and receiver functionality, a mobile telephone would also allow certain embodiments of the present invention. In such case, the normal telephony communications, perhaps supported by expanded 3G telephony capabilities, could serve to replace aspects of the IP communications described elsewhere in this specification.

IP Socket Communication Embodiments

A standard set of protocols for inter-unit communications is provided through IP socket communications, which is widely supported by available wireless communications hardware, including 820.11a, b and g (Wi-Fi). An embodiment of inter-unit communications is provided in Figures 14A-B. Figure 14A is a schematic block diagram of the socket configurations on the broadcast unit 710 and the receive unit 730.

LINKPLAY EXHIBIT 1004
Page 2908 of 4822

In the discussion below, transfer of the different messages and audio information are provided, generally but not always, through an Internet protocol. At the transport layer of such protocols, there will generally be used either a connectionless protocol or a connection-oriented protocol. Among the most common of these protocols are respectively the User Datagram Protocol (UDP) and the Transmission Control Protocol (TCP), and wherever these protocols are used below, it should be noted that any like protocol (connectionless or connection-oriented), or the entire class of protocol can generally be substituted in the discussion.

The broadcast unit 710, prior to the membership of the receive unit 730, broadcasts the availability of the broadcast on a broadcast 1050, which is generally a TCP socket. The annunciator 1050 broadcasts on a broadcast address with a predetermined IP address and port. The receive unit 730 has a client message handler 1060 that is also a TCP socket that is looking for broadcasts on the predetermined IP address and port. When it receives the broadcast, a handshake creates a private server message handler 1070 on a socket with a new address and port on the broadcast unit 710. The broadcast unit 710 and the receive unit 730 can now exchange a variety of different messages using the TCP protocol between the server message handler 1070 and the client message handler 1060. This information can comprise personal information about the users of the broadcaster unit 710 and the receive unit 730. Alternatively or additionally, the broadcast unit 710 can transfer a section of the audio signal that is currently being played, so that the user of the receive unit 730 can "sample" the music that is being played on the broadcast unit 710. It should be noted that, in general, the broadcast unit 710 continues its broadcast on the broadcast annunciator 1050 for other new members.

Once it is established that the broadcast unit 710 and the receiver unit 730 are mutually desirous of providing and receiving an audio broadcast, respectively, sockets optimized for broadcast audio are created both on the broadcast unit 710 and the receiver unit 730. These sockets will often be UDP sockets – on the broadcast unit 710, a multicast out socket 1080 and on the receiver unit 730, a multicast in socket 1090.

Figure 14B is a schematic block flow diagram of using IP sockets for establishing and maintaining communications between a broadcast unit 710 and the receive unit 730, according to the socket diagram of Figure 14A. In a step 1100, the broadcast annunciator 1050 broadcasts the availability of audio signals. In a step 1102, the receiver unit 730 searches for a broadcast annunciator 1050 on the client message handler 1060 socket. Once a connection is initiated in a step 1104, the broadcast unit 710 creates the message handler

LINKPLAY EXHIBIT 1004
Page 2909 of 4822

socket 1070 in a step 1106, and the receiver unit 730 retasks the message handler socket 1060 for messaging with the broadcast unit 730. The broadcast annunciator 1050 continues to broadcast availability through the step 1100.

In a step 1110, the broadcast unit 710 and the receiver unit 730 exchange TCP messages in order to establish the mutual interest in audio broadcasting and reception. Should there not be mutual acceptance, then the system returns to the original state in which the broadcast unit 710 is transmitting the broadcast annunciation in the step 1100, and the receive unit 730 searches for broadcasts in the step 1102. Given that the receive unit 730 and the broadcast unit 710 will be within communications distance, and that the broadcast unit 710 is transmitting an annunciation for which the receive unit 730 is receptive, the broadcast unit 710 will be set into a state where it will not establish communications with the receive unit 730 in the step 1106. This can occur either by not creating the message socket in the step 1106 when connection is made with the receiver unit 730, or that the annunciator 1050 remains silent for a predetermined period, perhaps for a period of seconds.

If the broadcast unit 710 and the receiver unit 730 do mutually accept a multicasting relationship, the broadcast unit 710 creates the multicast out UDP socket 1080 in a step 1112 and the receiver unit 730 creates the multicast in UDP socket 1090 in the step 1114, and multicast audio transmission and reception is initiated in a step 1116. It should be noted that should the broadcast unit 710 already be multicasting audio to a receiver unit 730 prior to the step 1112, the multicast out socket 1080 is not created, but that the address of this existing socket 1080 is communicated to the new cluster member.

Given that a cluster can comprise many members, the system of Figures 14A-B must be able to expand to include multiple members. Figure 15 is a schematic block diagram of the IP socket organization used with clusters comprising multiple members. The broadcast unit 710 includes a broadcast annunciator 1050 indicating broadcast availability for new members. For each member in the cluster, the broadcast unit further comprises a message handler 1070 dedicated to the specific member, whose receive unit 730 in turn comprises a message handler 1060, generally in a one-to-one relationship. The broadcast unit comprises N messaging sockets 1070 for the N receive units of the cluster, while each member has only a single socket 1060 connected to the broadcast unit. Thus, when a member wishes to send a message to the other members of the cluster, the message is sent via the receive unit message handler 1060 to the broadcast unit message handler 1070, and which is then multiply sent to the other receive unit message handlers 1060. It is also within the teachings of the present invention for each member of the cluster to have direct messaging capabilities with each

LINKPLAY EXHIBIT 1004
Page 2910 of 4822

other member, assisted in the creation of the communications by the broadcast unit 710, which can share the socket addresses of each member of the cluster, such that each member can assure that it is making connections with other members of the cluster rather than units of non-members. The broadcast unit 710 also comprises a multicast out socket 1080 which transfers audio to individual receiver sockets 1090 on each of the members of the cluster.

Members of the cluster may come and go, especially since members will frequently move physically outside of the transmission range of the broadcast unit 710. In order for the broadcast unit 710 to determine the current number of members of its cluster, it is within the teachings of the present invention for the broadcast unit 710 to use the messaging sockets 1060 and 1070 to "ping" the receive units 730 from time to time, or otherwise attempt to establish contact with each member of the cluster 700. Such communications attempts will generally be done at a predetermined rate, which will generally be more frequent than once every ten seconds. Information about the number of members of a cluster can be sent by the broadcast unit 710 to the other members of the cluster, so that the users can know how many members there are. Such information is conveniently placed on a display on the unit (see, for example, Figures 18A-B).

Music Synchronization

It will be generally desirable that the synchronicity of the audio playback on the broadcast unit 710 and the receive units 730 be highly synchronized, preferably within 1 second (i.e. this provides a low level functionality of listening to music together), more preferably within 100 milliseconds (i.e. near-simultaneous sharing of music, but an observer would be able to hear – or see through DJ 200 visible cues – the non-synchronicity), and most preferably within 20 milliseconds of one another. In a simple embodiment of the present invention, all members of a cluster 700 must communicate directly with the broadcast unit 710, without any rebroadcast. In such cases, making playback on the two units 710 and 730 as similar as possible will tend to synchronize their audio production.

Figure 8A is a schematic block diagram of audio units 100 with self-broadcast so that audio output is highly synchronized. Two audio units 100 are depicted, including a broadcast unit 710 and a receive unit 730. The organization of audio unit 100 elements is chosen to highlight the self-broadcast architecture. The audio media 1500, which can be compressed audio storage 310, stores the audio signals for broadcast. The output port 1502, which can comprise the inter-unit transmitter/receiver 110, transmits a broadcast audio signal, provided by the audio media 1500. The audio media comprise a variety of different storage protocols

LINKPLAY EXHIBIT 1004
Page 2911 of 4822

and media, including mp3 files, .wav files, or .au files which are either compressed or uncompressed, monoaural or stereo, 8-bit, 16-bit or 24-bit, and stored on tapes, magnetic disks, or flash media. It should be understood that the spirit of the present invention is applicable to a wide variety of different audio formats, characteristics, and media, of which the ones listed above are given only by way of example. This broadcast audio signal transmitted from the output port 1502 is received at the input port 1504, which can also comprise aspects of the inter-unit transmitter/receiver 110. The signal so received is then played to the associated user via the audio output 1508.

It should be noted that the audio output is normally connected to the audio media 1500 for audio playing when the unit 710 is not broadcasting to a receive unit 730. In such case, there is no need for the audio signals to go to the output port 1502 and thence to the input port 1504. Indeed, even when broadcasting, the audio signal within the broadcast unit 710 can go both directly to the audio output 1508 as well as to be broadcast from the output port 1502.

However, in order to assure the synchronicity of the audio output on the broadcast unit 710 and the receive unit 730, the broadcast unit 710 can present all audio signal from the audio media 1500 for output on the output port 1502. The signal will be received not only on the receiver 730 input port 1504, but also on the input port 1504 of the broadcast unit 710. This can take place either through the physical reception of the broadcast audio signal on a radio frequency receiver, or through local feedback loops within the audio unit 100 (e.g. through employment of IP loopback addresses).

In the receive unit 730, the audio signal received at the input port 1504 goes directly to the audio output 1508, and the other elements of the unit 100 depicted are not active. In the broadcast unit 710, however, if means are used to transfer audio signal between the output port 1502 and the input port 1504 are utilized, and if such transfer means requires less time than that taken for transmitting signal from the output port 1502 of the broadcast unit 710 to the input port 1504 of the receive unit 730, then a delay means 1506 is introduced to provide a constant delay between the input port 1504 and the audio output 1508. This delay can comprise a digital buffer if the signal is digitally encoded, or an analog delay circuit if the signal is analog. Generally, the delay introduced into the audio playback will be a predetermined amount based on the characteristics of the unit hardware and software.

Alternatively, in the case of a digital signal, the delay can be variably set according to the characteristics of the communications system. For example, if there are IP-based communications between the units, the units can "ping" one another in order to establish the

time needed for a "round-trip" communications between the systems. Alternatively, each receive unit 730 of a cluster 700 can transmit to the broadcast unit 710 a known latency of the unit based on its hardware and transmission characteristics. It should be noted that in order to handle different delays between multiple members of a cluster, a delay can be introduced into both the broadcast unit 710 and the receive unit 730, should a new member to the cluster have a very long latency in communications.

Note that the delay 1506 can serve a second purpose, which is to buffer the music should there be natural interruptions in the connections between the members of the cluster 700 (for example, should the receive units 730 move temporarily outside of the range of the broadcaster unit 710). In such case, should enough audio signal be buffered in the delay 1506, there would not be interruption of audio signal in the receive unit 730. Even in such cases, however, in order to accommodate the differences in time to play audio between units and within a unit, the delays in the broadcast unit 710 can be larger than those in the receive unit 730.

If the music compression and the bandwidth of the inter-unit communications are large enough, it can be that the broadcast unit 710 will broadcast less than half of the time. This will generally allow the receive unit 730 to rebroadcast the information from an internal memory store, allowing the effective range of the broadcast signal to potentially double. This can allow, through multiple rebroadcasts, for a very large range even if each individual unit 100 has a small range, and therefore for a potentially large number of users to listen to the same music.

In order to synchronize those that listen to the music through first, second and Nth rebroadcast, a scheme for multi-broadcast synchronization is presented in Figure 8B, a schematic flow diagram for synchronous audio playing with multiple rebroadcast. In such a case, the cluster 700 is considered to be all units 100 that synchronize their music, whether from an original broadcast or through multiple rebroadcasts. In a first step 780, a unit 100 receives a music broadcast along with two additional data. The first data is the current "N", or "hop" of the broadcast it receives, where "N" represents the number of rebroadcasts from the original broadcast unit 710. Thus, a unit 100 receiving music from the original broadcast unit 710 would have an "N" of "1" (i.e. 1 hop), while a unit 100 that received from that receiving unit 100 would have an "N" of "2" (2 hops), and so on. A second piece of information would be the "largest N" that was known to a unit 100. That is, a unit 100 is in contact generally with all units 100 with which it either receives or transmits music, and each send the "largest N" with which it has been in contact.

In a second step 782, the unit 100 determines the duration between signals in the broadcasts it is receiving. Then, two actions are taken. In a step 786, the unit 100 rebroadcasts the music it has received, marking the music with both its "N" and the largest "N" it knows of (either from the unit from which it received its broadcast or from a unit to which it has broadcast).

Also, in a step 784, the music that has been received is played after a time equal to the duration between signals and the "largest N" minus the unit's "N". This will allow for all units 100 to play the music simultaneously. Consider, for example the original broadcast unit 710. It's "N" is "0", and its "largest N" is the maximum number of rebroadcasts in the network. It will store music for a period of "largest N" (equals "largest N" minus "0") times the duration of a rebroadcast cycle, and then play it. For a unit 100 at the furthest rebroadcast, it's "N" and "largest N" will be equal to one another, so that it will store music for no time (i.e. "largest N" minus "N" = 0), but will play it immediately. This will allow all units 100 in the cluster to play music simultaneously. The limitation, however, is that there is memory in each unit 100 to store the music for a sufficient period of time. The units 100 on the system, however, can transfer the amount of storage that is available with the other information, and the number of rebroadcasts can be limited to the amount of memory available within the units 100 that comprise the cluster 700.

As the size of this multi-broadcast cluster 700 changes, the "largest N" can vary, and it will take generally on the order of "largest N" steps for the system to register "largest N". In such cases, there can be temporary gaps in the music on the order of the duration between signals, which will generally be on the order of tens of milliseconds, but which can be longer.

It should be noted that the synchronization of music does not need to accompany the transfer of an actual music signal. Figure 34A is a schematic block flow diagram of the synchronization of music playing from music files present on the units 100. In this embodiment, in a step 1900, the broadcast unit establishes the presence or absence of the music file comprising the music signals to be played on the receive unit. The music file can be referenced either with respect to the name of the file (e.g. "Ooops.mp3"), or a digital identifier that is associated with the music file.

If the music file is not present, then transfer of the music file from the broadcast unit to the receive units can automatically proceed through a file transfer mechanism such as peer-to-peer transfer in a step 1904. If the file was already present, or if the file has been transferred, or alternatively, if the file transfer has begun and enough of the file is present to

LINKPLAY EXHIBIT 1004
Page 2914 of 4822

allow the simultaneous playing of music between the two units 100, transmission of synchronization signals between the two units 100 can commence in a step 1902.

These synchronization signals can comprise many different forms. For example, the synchronization signal can be the time stamp from the beginning of the music file to the current position of the music file being played on the broadcast unit. Alternatively, the broadcast unit can send the sample number that is currently being played on the broadcast unit 100. In order to allow receiving units to begin synchronous playing in the middle of a transmission from a broadcast unit, the synchronization signals will preferably include information about the song being played, such as the name of the file or the digital identifier associated with the file.

Transmission of this synchronization signal continues until the termination of the song, or until a manual termination (e.g. by actuating a Pause or Stop key) is caused (the frequency of transmission of the synchronization signal will be discussed below). At this point, the broadcast unit can send a termination, pause or other signal in a step 1906. Note that this method of synchronization can operate when the receiving unit establishes connection with the broadcast unit even in the middle of a song.

Figure 34B is a schematic layout of a synchronization signal record 1910 according to Figure 34A. The order and composition of the fields can vary according to the types of music files used, the means of establishing position, the use of digital jewelry, the desire for privacy, and more.

The position field 1912 (SAMPLE#) which contains an indicator of position in a music file – in this case the sample number within the file. The music file identifier field 1914 (SONGID) comprises a textual or numerical identifier of the song being played. The third field is the sample rate field 1916 (SAMPLERATE), and is primarily relevant if the position field 1912 is given in samples, which allows a conversion into time. Given that the same audio entertainment can be recorded or saved at different sample rates, this allows the conversion from a potentially relative position key (samples) to one independent of sample rate (time). The jewelry signal field 1918 (JEWELSIGNAL) is used to encode a digital jewelry 200 control signal for controlling the output of the digital jewelry 200, should the receiver unit be associated with jewelry 200. The order and composition of the fields can vary according to the types of music files used, the means of establishing position, the use of digital jewelry, the desire for privacy, and more.

The frequency with which the record 1910 is broadcast can vary. The time of reception of the record 1910 sets a current time within the song that can adjust the position of

LINKPLAY EXHIBIT 1004
Page 2915 of 4822

the music playing on the receiver unit. It is possible for the record to be broadcast only once, at the beginning of the song, to establish synchronization. This, however, will not allow others to join in the middle of the music file. Furthermore, if the record 1910 is received or processed at different times for the single record, the music can be poorly synchronized. With multiple synchronization signals, the timing can be adjusted to account for the most advanced reception of the signal – that is, the music playing will be adjusted forward for the most advanced signal, but not be adjusted back for a more laggard signal.

If the record further contains a jewelry signal field 1918, the frequency with which the record 1910 should be sent should be comparable or faster than the rate with which these signals change, and should be preferably at least 6 times a second, and even more preferably at least 12 times a second. If less frequent record 1910 transmission is desired, then multiple jewel signal fields 1918 can be included in a single record 1910.

It should be noted that given units 100 of different design or manufacture, there can be different intrinsic delays between reception of music and/or synchronization signals and the playing of the music. Such delays can result from different speeds of MP3 decompression, different sizes of delay buffers (such as delay 1506), different speeds of handling wireless transmission, differing modes of handling music (e.g. directly from audio media 1500 to audio output 1508 on the broadcast unit, but requiring transmission through an output port 1502 and input port 1504 for the receiver unit), and more. In such cases, it is preferable for receiver units to further comprise a manual delay switch that can adjust the amount of delay on the receiver unit. This switch will generally have two settings: to increase the delay and to decrease the delay, and can conveniently be structured as two independent switches, a rocker switch, a dial switch or equivalent. It is useful for the increments of delay determined by the switch be adjustable so as to allow users to sense the music from the broadcast unit and the receiver unit as being synchronous, and it is preferable for the increments of delay to be less than 50 milliseconds, and even more preferable for the increments of delay to be less than 20 milliseconds, and most preferable for the units of delay to be less than 5 milliseconds.

Creation and Maintenance of Clusters

Search units 750 can be playing music themselves, or can be scanning for broadcast units 710. Indeed, search units 750 can be members of another cluster 700, either as broadcast unit 710 or receive unit 730. To detect a different cluster 700 in which it might desire membership, the search unit 750 can either play the music of the broadcast unit 710 to

the search unit 750 user, or it can scan for personal characteristics of the broadcast unit 710 user that are transmitted in the ID block 808. For example, a user can establish personal characteristic search criteria, comprising such criteria as age, favorite recording artists, and interest in skateboarding, and respond when someone who satisfies these criteria approaches.

Alternatively, the search unit 750 user can also identify a person whose cluster he wishes to join through visual contact (e.g. through perceiving the output of the person's light transducer 240).

Before a search unit 750 user can establish contact, it is preferable for a broadcast unit 710 user, or a receive unit 730 user, to provide permissions for others to join the cluster. For example, each unit 100 will generally be able to changeably set whether no one can join with their unit 100, whether anyone can join with their unit 100, or whether permission is manually granted for each user who wishes to join with their unit into a cluster. For a cluster 700, membership in the cluster can be provided either if any one member of the cluster 700 permits a search unit 750 user to join, or it can be set that all members of a cluster 700 need to permit other users to join, or through a variety of voting schemes. The permissions desired by each member will generally be sent between units 100 in a cluster as part of the ID block 808 or other inter-unit communications. Furthermore, these permissions can be used to establish the degree to which others can eavesdrop on a unit 100 transmission. This can be enforced either through the use of cryptography, which can only provide decryption keys as part of becoming a cluster 700 member, through provision of a private IP socket address or password, through standards agreed by manufacturers of unit 100 hardware and software, or by unit 100 users limiting the information that is sent through the ID block 808 through software control.

The search unit 750 user can then establish membership in the group in a variety of ways. For example, if the search unit 750 is scanning music or personal characteristics of the unit 100 user, it can alert the search unit 750 user about the presence of the unit 100. The search unit 750 user can then interact with the search unit 750 interface to send the unit 100 user a message requesting membership in the cluster 700, which can be granted or not. This type of request to join a cluster 700 does not require visual contact, and can be done even if the search unit 750 and cluster are separated by walls, floors, or ceilings.

Another method of establishing contact between a search unit 750 user and a cluster 700 member is for the search unit 750 user to make visual contact with the cluster 700 member. In such case that physical contact or physical proximity is easily made between the unit 100 of the cluster member and the search unit 750, digital exchange can be easily made

LINKPLAY EXHIBIT 1004
Page 2917 of 4822

either through direct unit 100 contact through electrical conductors, or through directional signals through infra-red LEDs, for example. For example, the search unit 750 user can point his unit 100 at the cluster 700 member unit, and then if the cluster member wishes the search unit 750 user to join the cluster, could point his unit 100 at the search unit 100, and with both pressing buttons, effect the transfer of IDs, cryptography keys, IP socket addresses or other information that allows the search unit 750 user to join the cluster 700.

Alternatively, the broadcast DJ 720 (or the receive DJ 740) can present digital signals through the light transducer. For example, most DJ 720 light transduction will be modulated at frequencies of 1-10 Hz, with human vision not being able to distinguish modulation at 50 Hz or faster. This means that digital signals can be displayed through the light transducer 240 at much higher frequencies (kHz) that will not perceived by the human eye, even while lower frequency signals are being displayed for human appreciation. Thus, the broadcast DJ 720 can receive a signal from the broadcast unit 710 DJ transmitter 120 containing information needed for a search unit 750 to connect to the broadcast unit's cluster 700. This information will be expressed by the light transducer 240 of the broadcast DJ 720 in digital format. The search unit 750 can have an optical sensor, preferably with significant directionality, that will detect the signal from the light transducer 240, so that the search unit 750 is pointed in the direction of the broadcast DJ 720, and the identifier information required for search unit 750 to become a member of cluster 700. This optical sensor serves as the DJ directional identifier 122 of Figure 1. At this point, if desired, the broadcast unit 710 user can determine if they want the search unit 750 user to join the cluster 700.

A summary of means to effect joining of a cluster is provided in Figures 13A through E, which display means for a search unit 750 to exchange information prior to joining a cluster 700 via a broadcast unit 710. It is also within the teachings of the present invention for the search unit 750 to institute communications with a receive unit 730 for the purposes of joining a cluster in a similar fashion, particularly since it may be difficult for a person outside of the cluster 700 to determine which of the cluster 700 members is the broadcast unit 710, and which is a receiver unit 730.

If should be noted in the Figures 13A-G that limited range and directionality are preferred. That is, there can be a number of broadcast units 710 within an area, and being able to select that one broadcast unit 710 whose cluster one wishes to join requires some means to allow the search unit 750 user to select a single broadcast unit 710 among many. This functionality is generally provided either by making a very directional communication between the two devices, or by depending on the physical proximity of the search unit 750

LINKPLAY EXHIBIT 1004
Page 2918 of 4822

and the desired broadcast unit 710 (i.e. in a greatly restricted range, there will be fewer competing broadcast units 710). In the following description, the "broadcaster" denotes the user using the broadcast unit 710, and the "searcher" denotes the user using the search unit 750.

In the Figures 13A-G, the selection of the cluster by the searcher occurs in three ways, that will referred to as "search transmission mode", "broadcast transmission mode", and "mutual transmission mode", according to the entity that is conveying information. In search transmission mode, the searcher sends an ID via the search unit 750 to the broadcast unit 710. This ID can comprise a unique identifier, or specific means of communication (e.g. an IP address and port for IP-based communication). With this ID, the broadcast unit can either request the searcher to join, or can be receptive to the searcher when the searcher makes an undifferentiated request to join local units within its wireless range. In broadcast transmission mode, the broadcaster sends an ID via the broadcast unit 710 to the search unit 750. With this ID, the searcher unit can then make an attempt to connect with the broadcast unit 710 (e.g. if the ID is an IP address and port), or the search unit can respond positively to a broadcast from the broadcast unit 710 (e.g. from a broadcast annunciator 1050), wherein the ID is passed and checked between the units early in the communications process. Mutual transmission mode comprises a combination of broadcast transmission mode and search transmission mode, in that information and communication is two way between the broadcaster and the searcher.

Figure 13A is a schematic cross-section through a search unit 750 and a broadcast unit 710 in which communications are provided via visible or infrared LED emission in search transmission mode. On the right of the figure, a LED 1044 with an associated lens 1046 (the two of which can be integrated) transmits a directional signal from the unit case 1000. This light can optionally pass thorough a window 1048 that is transparent to the light. On the left of the figure, a lens element 1040 collects light through a broad solid angle and directs it onto a light sensing element 1042, which is conveniently a light-sensing diode or resistor. The directionality of the communication is conferred by the transmitting lens 1046 and the collecting lens 1040.

Alternatively, the LED 1044 can be replaced by a visible laser. Figure 13B is a schematic cross-section through a search unit 750 and a broadcast unit 710 in which communications are provided via a visible or infrared laser in search transmission mode. The search unit 750 comprises a diode laser 1041 that is conditioned by a lens 1043 to form a beam that is sensed by the light sensing element 1042 on the broadcast unit 710. Because a

LINKPLAY EXHIBIT 1004
Page 2919 of 4822

collimated laser beam can be difficult to aim with precision at a photosensing element carried by a person, the optics can comprise a two focus lens 1043 that has a portion that produces a collimated beam 1045, and a second portion that produces a diverging beam 1047. The collimated beam is used by the user of the search unit 750 as a guide beam to direct the pointing of the unit 750, while the divergent beam provides a spread of beam so that the human pointing accuracy can be relatively low. The means for creating the two focus lens 1043 can include the use of a lens with two different patterns of curvature across its surface, or the use of an initial diverging lens whose output intersects a converging lens across only a part of its diameter, where the light that encounters the second lens is collimated, and the light that does not encounter the second lens remains diverging. It is also within the teachings of the present invention for the lens to be slowly diverging without a collimating portion, such that the user does not get visible feedback of their pointing accuracy. In such case, the laser can emit infrared rather than visible wavelengths.

Figure 13C is a schematic cross-section through a search unit 750 and a broadcast unit 710 in which communications are provided via visible or infrared emission from a digital jewelry element 200 in broadcast transmission mode. The digital jewelry 200 is carried by the broadcaster on a chain 1033, with the digital jewelry 200 visible. The digital jewelry is emitting through a light transducer 1031 a high frequency signal multiplexed within the visible low frequency signal. The search unit 750 is pointed in the direction of the digital jewelry 200, and receives a signal through the light-sensing element 1042. This manner of communication is convenient because the searcher knows, via the presence of the visible signal on the digital jewelry 200, that the broadcaster is receptive to cluster formation.

Figure 13D is a schematic cross-section through a search unit 750 and a broadcast unit 710 in which communications are provided via contact in mutual transmission mode. In this case, the broadcast unit 710 and the search unit 750 both comprise a contact transmission terminus 1030, and electronic means by which contact transmission is performed. This means can operate either inductively (via an alternative current circuit), through direct electrical contact with alternating or direct current means, or other such means that involves a direct physical contact (indicated by the movement of the search unit 750 to the position of the unit depicted in dotted lines). The search unit 750 or the broadcast unit 710 can, via automatic sensing of the contact or manual control, initiate communications transfer. Given the mutuality of contact as well as the physical equivalence of the two units 710 and 750, information transfer is possible in both directions. It should be noted that in the case of direct

LINKPLAY EXHIBIT 1004
Page 2920 of 4822

current connection, the termini 1030 will comprise two contact points, both of which must make electrical contact in order for communications to occur.

Figure 13E is a schematic cross-section through a search unit 750 and a broadcast unit 710 in which communications are provided via sonic transmissions in broadcast transmission mode. The broadcaster (or receivers) will be listening to the audio information generally through headphones 1020 or earphones, all of which comprise speakers 1022 that, to one extent or another, leak sonic energy. The use of audio output devices as depicted in Figure 10 and Figures 11A and 11B that admit external sound, will also increase the amount of sound energy lost. This sound energy can be detected by the searcher via a directional speaker comprising a sound collector 1024 and a microphone 1026. This system requires that the sound output of the broadcast unit 710 and the receiver unit 750 also output an ID encoded in the sound. Such sound can be conveniently output at inaudible frequencies, such as 3000-5000 Hz, which carry sufficient bandwidth to encode short messages or identifiers (e.g. an IP address and port number can be carried in 5 bytes). Sound energy, especially at higher frequencies, can be quite directional, depending on the shape of the collector 1024 and the structure of the microphone 1024, allowing good directional selection by the searcher.

Figure 13F is a schematic cross-section through a search unit 750 and a broadcast unit 710 in which communications are provided via radio frequency transmissions in broadcast transmission mode. The radio frequency transmissions are not strongly directional (and for the purposes of the broadcast of audio information, are designed to be as directionless as possible). In order to distinguish a desired cluster 700 to join and an undesired cluster 700, a number of strategies can be employed. For example, the strengths of the various signals can be measured and the strongest chosen for connection. Alternatively, if there are multiple broadcast connections available, the search unit 750 can sequentially attempt a connection with each broadcast unit 710. When the attempt is made, the broadcast unit 710 can, prior to alerting the broadcaster of the attempted joining by a new member, cause the digital jewelry 200 associated with the broadcast unit 710 to visibly flash a characteristic signal. The searcher can then verify by pressing the appropriate button on the search unit 750 his desire to join the cluster 700 of the broadcast digital jewelry 200 that had just flashed. If the searcher decided not to join that cluster 700, the search unit 750 could search for yet another unit broadcast unit 750 within range, and attempt to join.

At any time, the members of a cluster 700 can share personal characteristics (nickname, real name, address, contact information, face or tattoo images, favorite recording artists, etc.) through selection of choices of the unit 100 interface, with all such characteristics

LINKPLAY EXHIBIT 1004
Page 2921 of 4822

or a subset thereof to be stored on the units 100. In order to assist cluster 700 members in determining whether or not to accept a person into their cluster 700, a search unit 750 member can display either the total number of people with whom he has shared personal characteristics, or he can alternatively allow the cluster members to probe his store of persons with whom personal characteristics have been stored to see whether a particular trusted person or group of common acquaintances are present therein. It is also within the spirit of the present invention for individuals to rate other individual members of their cluster, and such ratings can be collated and passed from person to person or cluster to cluster, and can be used for a cluster 700 to determine whether a search unit 750 person should be added to the cluster 700.

Figure 17 is a matrix of broadcaster and searcher preferences and characteristics, illustrating the matching of broadcaster and searcher in admitting a searcher to a cluster. A broadcaster preference table 1160 includes those characteristics that the broadcaster wishes to see in a new member of a cluster. These characteristics can include gender, age, musical "likes" and "dislikes", the school attended, and more. The searcher similarly has a preference table 1166. The searcher preference table 1166 and broadcaster preference table 1160 are not different in form, as the searcher will at another time function as the broadcaster for another group, and his preference table 1166 will then serve as the broadcaster preference table.

The broadcaster preference table 1160 can be automatically matched with a searcher characteristics table 1162. This table 1162 comprises characteristics of the searcher, wherein there will be characteristics that overlap in type (e.g. age, gender, etc.) which can then be compared with the parameters in the broadcaster preference table. This matching occurs during the period when the searcher is interrogating the cluster with interest in joining. Similarly, there is a broadcaster characteristics table 1164 indicating the characteristics of the broadcaster, which can be matched against the searcher preferences tale 1166.

The algorithm used in approving or disapproving of an accord between a preference table and a characteristics table can be varied and set by the user – whether by the broadcaster to accept new members into a cluster, or by a searcher to join a new cluster. For example, the user could require that the gender be an exact match, the age within a year, and the musical preferences might not matter. The user can additionally specify that an accord is acceptable if any one parameter matches, specify that an accord be unacceptable if any one parameter does not match, specify an accord be acceptable based on the overlap of a majority of the individual matches, or other such specification.

LINKPLAY EXHIBIT 1004
Page 2922 of 4822

It should be noted that the broadcaster preferences table 1160 and the broadcaster characteristics table 1164 (and likewise with the searcher tables 1162 and 1166) can be a single table, according to the notion that a person will prefer people who are like themselves. Each user could then express the acceptable range of characteristics of people with which to join as a difference from their own values. For example, the parameter "same" could mean that the person needs to match closely, whereas "similar" could indicate a range (e.g. within a year) and "different" could mean anyone. In this way, there would not be the burden on the user to define the preference table 1160 or 1166 in a very detailed manner.

In the case of a cluster, the transfer of information between the searcher and the cluster can, as mentioned above, involve not only the broadcaster, but also other members of the cluster (especially since the searcher may not know the identity of a cluster's broadcaster from external observation). The cluster can also make communal decisions about accepting a new member. That is, if there are 4 members of a cluster, and a searcher indicates an interest in joining the cluster, there can be voting among the members of a cluster regarding the acceptance of the new member. The procedure of voting will normally be done by messaging among the members, which can be assisted by structured information transfer as will be described below.

A number of such voting schemes are described in Figure 19, a table of voting schemes for the acceptance of new members into a cluster. The first column is the name of the rule, and the second column describes the algorithm for evaluation according to the rule. In the "BROADCASTER" rule, the broadcaster decides whether or not the new member will be accepted. The new member is accepted when the broadcaster indicates "yes" and is otherwise rejected.

In the "Majority" rule, the members are polled, and whenever a majority of the members vote either acceptance or rejection, the new member is accordingly accepted or rejected. It should be noted that this rule (as well as the rules to follow) depends on the broadcaster or other member of the cluster having knowledge of the number of members in the cluster, which will generally be the case (e.g. in an IP socket based system, the broadcaster can simply count the number of socket connections). Thus, if the number of members in a cluster is given as N_{mem}, as soon as $(N_{mem}/2) + 1$ members have indicated the same result, that result is then communicated to the broadcaster, the members and the prospective new member. If the number of members is even, and there is a split vote, the result goes according to the broadcaster's vote.

LINKPLAY EXHIBIT 1004
Page 2923 of 4822

According to the "Unanimous" rule, a new member is accepted only on unanimous decision of the members. Thus, the prospective new member is rejected as soon as the first "no" vote is received, and is accepted only when the votes of all members of the cluster are received, and all of the votes are positive.

The "Timed Majority" rule is similar to that of the "Majority" rule, except that a timer is started when the vote is announced, the timer being of a predetermined duration, and in a preferred embodiment, is indicated as a count down timer on the unit 100 of each member of the cluster 700. The vote is completed when $(N_{mem}/2)+1$ members vote with the same indication ("yes" or "no") if the timer has not completed its predetermined duration. If all of the members have voted, and the vote is a tie, the result goes in accordance with that of the broadcaster. If the timer has expired, and the vote has not been decided, the number of members that have voted is considered a quorum of number Q. If $(Q/2)+1$ members have voted in some fashion, that is the result of the vote. Otherwise, in the case of a tie, the result goes according to the vote of the broadcaster. If the broadcaster has not voted, the vote goes according to the first vote received.

The "Synchronized Majority" rule is similar to the Timed Majority rule, but instead of initiating the vote, and then waiting a predetermined period for members to vote, the vote is announced, and then there is a predetermined countdown period to the beginning of voting. The voting itself is very limited in time, generally for less than 10 seconds, and preferably for less than 3 seconds. Counting votes is performed only for the quorum of members that vote, and is performed according to the rules for the Timed Majority.

There are many different voting schemes consistent with creating, growing and maintaining clusters within the spirit of the present invention. For instance, in cases where there are close votes, the voting can be reopened for individuals to change their vote. In cases, members can request a new round of voting. Furthermore, the voting can be closed ballot, in which the votes of individuals are not known to the other members, or open voting, in which the identity of each member's vote is publicly displayed on each unit 100.

In addition, the voting can be supported and enhanced by information made available to each member through displays on the units 100. Figure 18A is a screenshot of an LCD display 1170 of a unit 100, taken during normal operation. The display 1170 is comprised of two different areas, an audio area 1172 and a broadcaster area 1174. The audio area 1172 includes information about the status of the audio output and the unit 100 operation, which can include battery status, the name of the performer, the title of the piece of music, the time the audio has been playing, the track number and more. The broadcaster area 1174 comprises

LINKPLAY EXHIBIT 1004
Page 2924 of 4822

information about the status of the cluster 700. In the example given, the broadcaster area includes the number "5", which represents the number of people current in the cluster, the text "DJ", which indicates that the unit 100 on which the display 1170 is shown is currently the broadcaster of the cluster 700, and the text "OPEN", which indicates that the cluster is open for new members to join (the text "CLOSED" would indicate that no new members are being solicited or allowed).

Figure 18B is a screenshot of an LCD display 1170 of a unit 100, taken during voting for a new member. The audio area 1172 is replaced by a new member characteristics area 1176, in which characteristics of the prospective new member are displayed. Such characteristics can include the name (or nickname) of the prospective new member, their age, and their likes (hearts) and dislikes (bolts). In the broadcaster area 1174, the digit "3" indicates that there are three current members of the cluster 700, and an ear icon indicates that the current unit 100 is being used to receive from the broadcaster rather than being a broadcaster, and the name [ALI] indicates the name of the current broadcaster. The text "VOTE-MAJ" indicates that the current vote is being done according to the Majority rule. The broadcaster area 1174 and the new member characteristics areas 1176 provide the information needed by the existing member to make a decision about whether to allow the prospective new member to join.

The displays 1170 of Figures 18A-B are indicative only of the types of information that can be placed on a display 1170, but it should be appreciated that there are many pieces of information that can be placed onto the displays 1170 and that the format of the display can be very widely varied. Furthermore, there need not be distinct audio areas 1172 and broadcaster areas 1174, but the information can be mixed together. Alternatively, especially with very small displays 1170, the display 1170 can be made to cycle between different types of information.

It is also within the spirit of the present invention for individuals to rate other individual members of their cluster, and such ratings can be collated and passed from person to person or cluster to cluster, and can be used for a cluster 700 to determine whether a search unit 750 person should be added to the cluster 700. Figure 27 is a schematic block flow diagram of using a prospective new member's previous associations to determine whether the person should be added to an existing cluster.

In a step 1400, from a search unit 750, the prospective new member places an external communication request with an operational broadcast annunciator 1050 by a broadcast unit 710. In a step 1402, a temporary message connection is established through which

LINKPLAY EXHIBIT 1004
Page 2925 of 4822

information can be passed mutually between the search unit 750 and the broadcast unit 710. The broadcast unit 710 requests personal and cluster ID's from the search unit 750. The personal ID is a unique identifier that can be optionally provided to every audio unit 100, and which can further be optionally hard-encoded into the hardware of the unit 100. The cluster IDs represent the personal ID's of other units 100 with which the search unit 750 has been previously associated in a cluster. In a step 1406, the broadcast unit 710 matches the incoming personal IDs and cluster IDs with personal ID's and cluster IDs that are stored in the memory of the broadcast unit 710. If there exist a sufficient number of matches, which can be computed as a minimum number or as a minimum fraction of the IDs stored in the broadcast unit 710, the new member of the search unit 750 can be accepted into the cluster. In a step 1412, the search unit 750 can then store the ID of the broadcast unit 710 and the other members of the existing cluster 700 into his cluster IDs, and the broadcast unit 750 and the other receive units 730 of the cluster can then store the personal ID of the search unit 750 into their cluster IDs. If there does not exist a sufficient number or quality of matches, the broadcast unit 710 will reject the prospective new member, optionally send a message of rejection, and then close the socket connection (or other connection that had been created) between the broadcast unit 710 and the search unit 750. No new IDs are stored on either unit 710 or 750.

It is also within the spirit of the present invention for other information associated with the personal and cluster IDs to be shared and used in the algorithm for determining whether to accept or reject a prospective new member into a cluster 700. This information can include rating information, the duration of association with another cluster 700 (i.e. the longer the association, the more suitable the social connection of that person with the cluster 700 would have been), the size of the cluster 700 when the searcher was a member of a particular cluster 700, the popularity of a cluster 700 (measured by the number of cluster IDs carried by the broadcast unit 710), and more. The matching program, likewise, would weight the existence of a match by some of these quality factors in order to determine the suitability of the searcher to join the cluster.

While the comparisons can be made between a search unit 750 personal and cluster IDs and those from the broadcast unit 710, representing the personal experience of the owners of the respective units, it is also possible that the reputation or desirability of individuals with a given personal ID can be posted to or retrieved from trusted people. For example, two friends can swap the information of which IDs are to be trusted or not between two units 100, or alternatively, can be posted onto or retrieved from the Internet. For

LINKPLAY EXHIBIT 1004
Page 2926 of 4822

example, after a bad personal experience with a unit 100 with a personal ID of 524329102, a person could post that ID on the Internet to share with friends, so that the friends could avoid allowing that person to join, or avoid joining a cluster with that person.

It should be noted that publishing a list of personal IDs allows people to establish the breadth of their contacts. By posting their contacts on web sites, people can demonstrate their activity and popularity. This also encourages people to join clusters, in order to expand the number of people with whom they have been associated. Furthermore, the personal ID serves as a "handle" by which people can further communicate with one another. For example, on the Internet, a person can divulge a limited amount of information (e.g. an email address) that would allow other people with whom they have been in a cluster together to contact them.

It should be noted that the formation and maintenance of a cluster 700 requires the initial and continued physical proximity of the broadcast unit 710 and the receive unit 730. In order to help maintain such physical proximity conducive to cluster maintenance, feedback mechanisms can be used to alert the users to help them maintain the required physical proximity, as will be discussed below.

Figure 28 is a block flow diagram indicating the steps used to maintain physical proximity between the broadcast unit 710 and the receive unit 730 via feedback to the receive unit user. In a step 1530, the wireless connection between the broadcast unit 710 and the receive unit 730 is established. In a step 1532, the connection between the two units 710 and 730 is tested. There are a number of different means by which this testing can take place. For example, in IP-based communications, the receive unit 730 can from time to time – though generally less than every 10 seconds, and even more preferably less than every 1 second – use the "ping" function to test the presence and speed of connection with the broadcast unit 710. Alternatively, the receive unit 730 will be receiving audio signals wirelessly almost continuously from the broadcast unit 710, and a callback alert function can be instituted such that loss of this signal determined at a predetermined repeat time – which is conveniently less than 5 seconds, and even more preferably less than every 1 second – and which is then reported to the system.

While the methods above determine the absolute loss of a signal, they do not anticipate loss of signal. A method that does anticipate signal issues prior to loss is the measurement of signal strength. This can be done directly in the signal reception hardware by measuring the wireless signal induced current or voltage.

LINKPLAY EXHIBIT 1004
Page 2927 of 4822

In a step 1534, the results of the connection testing performed in the step 1532 is analyzed in order to determine whether the signal is adequate. It should be noted that a temporary loss of signal, lasting even seconds, may or may not be of importance. For example, the broadcast unit 710 user and receive unit 730 users could walk on opposite sides of a metallic structure, enter a building at different times, change their body posture such that the antennae are not optimally situated with respect to one another, etc. Thus, an algorithm is generally used to time average the results of the step 1532, with the results conveniently time averaged over a matter of seconds.

Whatever the results of the signal test of the step 1534, the step 1532 is continuously repeated as long as the connection between the broadcast unit 710 and the receive unit 730 is present. If the signal is deemed inadequate, however, feedback to that effect is provided to the receive unit 730 user in a step 1536. The user feedback can occur through a variety of mechanisms, including visual (flashing lights) and tactile (vibration) transducers, emanating either from the audio unit 100 or the digital jewelry 200. For example, the receiver unit 730 can send a signal to the associated digital jewelry 200 to effect a special sequence of light transducer output.

It is most convenient, however, for the audio output of the receiver unit 730 as heard by the user to be interrupted or overlain with an audio signal to alert the user to the imminent or possible loss of audio signal. This audio signal can include clicks, beeps, animal sounds, closed doors, or other predetermined or user selected signals heard over silence or the pre-existing signal, with the signal possibly being somewhat reduced in volume such that the combination of the pre-existing signal and the feedback signal is not unpleasantly loud.

It should be noted that the flow diagram of Figure 28 refers specifically to alerting the receive unit 730 user of potential communications issues. Such alerting can also be usefully transferred to or used by the broadcast unit 710. For example, with knowledge of the communications issues, the broadcast unit 710 user can move more slowly, make sure that the unit is not heavily shielding, that any changes in posture that could relate to the problems are reversed, etc. The broadcast unit 710 can perform communications tests (as in the step 1532) or analyze the tests to determine if the communications are adequate (as in the step 1534) – particularly through use of the messaging TCP channels. Given that there can be multiple receive units 730 connected to s single broadcast unit 710, it is generally preferable for the tests to be performed on the receive units 730, and problems to be communicated to the broadcast unit 710 – provided, however, that communications still exist for such communication.

LINKPLAY EXHIBIT 1004
Page 2928 of 4822

In order to overcome this deficiency, it is possible for the receiver unit 730 to communicate potential problems in communications to the broadcast unit 710 at an early indication. The broadcast unit 710 then starts a timer of predetermined length. If the broadcast unit 710 does not receive a "release" from the receive unit 730 before the timer has completed its countdown, it can then assume that communications with the receive unit 730 have been terminated, and it can then send feedback to the broadcast unit 710 user.

It is also within the teachings of the present invention for both the broadcast unit 710 and the receive unit 730 to independently monitor the connections with each other, and alert their respective users of communications problems.

It should be noted that the use of audio alerts can be used more generally within the user interface of the audio units 100. Thus, audio alerts can be conveniently used to inform the user of the joining of new members to the cluster 700, the initiation of communications with search units 750 outside of the group, the leaving of the group by existing cluster 700 members, the request by a receive unit 730 to become the broadcast unit 710, the transfer of cluster control from a broadcast unit 710 to a receive unit 730, and more. These alerts can be either predetermined by the hardware (e.g. stored on ROM), or can be specified by the user. Furthermore, it can be convenient for the broadcast unit 710 to temporarily transfer to new members of the cluster custom alerts, so that the alerts are part of the experience that the broadcast unit 710 user shares with the other members of the cluster. Such alerts would be active only as long as the receive units were members of the cluster 700, and then would revert back to the alerts present before becoming cluster members.

Cluster Hierarchy

A receive unit 730 can also be the broadcast unit 710 of a separate cluster 700 from the cluster 700 of which it is a member. This receive unit is called a broadcasting receiver 770. In such case, it is convenient for the receive units 730 that are associated with the broadcasting receiver 770 to become associated with the cluster 700 of which the broadcasting receiver 770 is a member. This can conveniently be accomplished in two different ways. In a first manner, the receive units 730 that are associated with the broadcasting receiver 770 can become directly associated with the broadcast unit 710, so that they are members only of the cluster 700, and are no longer associated with the broadcasting receiver 770. In a second manner, the receive units 730 associated with the broadcasting receiver 770 can remain primarily associated with the broadcasting receiver 770, as shown in Figures 9A and 9B, which are schematic block diagrams of hierarchically-related clusters. In

LINKPLAY EXHIBIT 1004
Page 2929 of 4822

Figure 9A, the receive units 730 that are members of a sub-cluster 701 of which the broadcast unit is a broadcast receive unit 770, can receive music directly from the broadcast receive unit 710, while retaining their identification with the broadcasting receiver 770, such that if the broadcasting receiver 770 removes itself or is removed from the cluster 700, these receive units 730 similarly are removed from the cluster 700. In order to provide this form of hierarchical control, the sub-cluster 701 receive units 730 can obtain an identifier, which can be an IP socket address, from the broadcast receive unit 770, indicating the desired link to the broadcast unit 710. The sub-cluster receive units 730, however, maintain direct communications with the broadcast receive unit 770, such that on directive from the unit 770, they break their communications with the unit 710, and reestablish normal inter-unit audio signal communications with the broadcast receive unit 770. In an embodiment using IP addressing and communications, this can involve the maintenance of TCP messaging communications between the sub-cluster 701 receive units 730 with the broadcast receive unit 770, during the time that the sub-cluster 701 is associated with the cluster 700.

In Figure 9B, the receive units 730 of the sub-cluster 701 receive music directly from the broadcasting receiver 770, which itself receives the music from the broadcast unit 710. In such case, as the broadcasting receiver 770 is removed from the cluster 700, the receive units 730 of the sub-cluster 701 would also not be able to hear music from the cluster 700.

It would be apparent that such an arrangement can be hierarchically arranged, such that the receive unit 730 of the sub-cluster 701 can itself be the broadcast receiver 770 of another sub-cluster 701, and so forth. The advantage of this arrangement is that people that are associated with one another, forming a cluster 700, can move as a group from cluster to cluster, maintaining a separate identity.

It should be also noted that the configuration of communications between members of a hierarchical cluster can be variously arranged, not only as shown in Figures 9A and 9B. For example, every member of the cluster 700 can have a direct link between every other member of the cluster 700, such that no re-broadcast of messages needs to take place. Furthermore, given that there are different inter-unit communications (for example, messaging versus audio broadcast), it is within the teachings of the present invention that the configuration for the different modes of communication can be different – for example direct communications between the broadcast unit 710 for audio broadcast, but peer-to-peer communications between individual units for messaging.

LINKPLAY EXHIBIT 1004
Page 2930 of 4822

Maintaining Private Communications

In order to restrict membership in a cluster 700, either the information transfer must be restricted, such as by keeping private the socket IP addresses or passwords or other information that is required for a member to receive the signal, or the signal can be transmitted openly in encrypted form, such that only those members having been provided with the encryption key can properly decode the signal so sent. Both of these mechanisms are taught within the present invention, and are described at various points within this specification.

Figure 32A is a schematic block diagram of maintaining privacy in open transmission communications. In this case, the transmission is freely available to search units 750 in a step 1830, such as would occur with a digital RF broadcast, or through a multicast with open a fixed, public socket IP address available in certain transmission protocols. In this case, the broadcast audio signal or information signal is made in encrypted form, and membership in the cluster is granted through transfer of a decoding key in a step 1832.

Figure 32B is a schematic block diagram of maintaining privacy in closed transmission communication. In a step 1834, the broadcast unit 710 makes a closed transmission broadcast, such as through a socket IP address, that is not publicly available. In a step 1836, the broadcast unit 710 provides the private address to the search unit 750, which can now hear the closed transmission from the step 1834, which is not encrypted. Alternatively, or in addition to the provision of the private address in the step 1836, the establishment of the connection through the private, closed transmission is effected via a password provided in a step 1838. This password can, for example, be used in the step 1110 (e.g. see Figure 14B) to determine whether the broadcast unit 710 accepts the search unit 750 for audio multicasting.

In this section, the encryption of the musical signal and/or associated information about personal characteristics of members of the cluster 700 is described. The custom compressor 330 of the unit 100 can perform the encryption. In such a case, before joining a cluster, the search unit 750 can only receive some limited information, such as characteristics of the music being heard or some limited characteristics of the users in the cluster 700. If the search unit 750 user requests permission to join the cluster 700 and it is granted, the broadcast unit 710 can then provide a decryption key to the search unit 750 that can be used to decrypt the music or provide a private IP address for multicasting, as well as supply additional information about the current members of cluster 700.

LINKPLAY EXHIBIT 1004
Page 2931 of 4822

It should be noted that in certain cases, it can be useful to have multiple forms of privacy protection. For example, a broadcast unit 710 can provide a search unit 750 access to audio signals and information for the cluster 700, but can reserve certain information based on encryption to only some members of the cluster 700. For example, if a group of friends comprise a cluster 700, and accept some new members into the cluster 700, access to more private information about the friends, or communications between friends, can be restricted on the basis of shared decryption keys.

Figure 33 is a schematic block diagram of a hierarchical cluster, as in Figure 9A, in which communications between different units is cryptographically or otherwise restricted to a subset of the cluster members. Thus, there are three types of communication that are used in the communication: channel A, which takes place between the members of the original cluster; channel B, which takes place between the members of the original cluster (mediated through the broadcast unit 710) and members of the sub-cluster 701; and channel C, which takes place between the members of the sub-cluster 701. Thus, a communications originating from the broadcast unit 710 can be directed either through channel A or channel B, and likewise, a communications originating from the broadcast receive unit 770 can be directed either at members only of the sub-cluster 701 through channel C, or to all members of the cluster 700 through both channels C and B, which is then communicated trough channel A.

A number of means can be used to maintain such independent channels. For example, separate socket communications can be established, and the originators of the communications can determine that information which is carried on each separate channel. For example, given an open transmission scheme such as digital RF signal, the information can be encoded with separate keys for the different channels of communication – thus, the cryptographic encoding determines each channel. A given unit 100 can respond to more than one encoding. Indeed, a channel identifier can be sent with each piece of information indicating the ID of the decoding key. If a unit 100 does not have the appropriate decoding key, then it is not privy to that channel communications.

Alternatively, if the communications is IP socket based, then each channel is determined by IP socket addresses. Furthermore, access to those addresses can be, for example, password controlled. Also, the socket communications can be broadcast so that any unit 100 can receive such broadcast, but that decoding of the broadcast can be mediated through cryptographic decoding keys.

It should be noted that there can be multiple forms of communication, which can comprise messaging communications using the TCP/IP protocols, versus multicasting using

LINKPLAY EXHIBIT 1004
Page 2932 of 4822

UDP protocols, and also DJ 200 control signals using yet another protocol. The access to each of these communications can be controlled via different privacy hierarchies and techniques. For example, the audio multicasting will be available to all members within a cluster, while the messaging may retain different groupings of privacy (e.g. hierarchical), while the DJ control signals will generally be limited to communications between a given unit 100 and its corresponding DJs 200.

Broadcast Control Transfer

The dynamics of cluster 700 can be such that it will be desirable for a receive unit 730 to become the broadcast unit for the cluster. Such a transfer of broadcast control will generally require the acquiescence of the broadcast unit 710 user. To effect such a transfer, the user of the receiver unit 730 desiring such control will send a signal to the broadcast unit 710 expressing such intention. If the user of the broadcast unit 710 agrees, a signal is sent to all of the members of the cluster indicating the transfer of broadcast control, and providing the identifier associated with the receive unit 730 that is to become the broadcast unit 710. The broadcast unit 710 that is relinquishing broadcast control now becomes a receive unit 730 of the cluster 700.

It should be noted that the transfer of control as described above requires the manual transfer of control, such as actuation of a DJ switch. This switch can be limited to this function, or can be part of a menu system, in which the switch is shared between different functions. It is also within the spirit of the present invention that there be voice-activated control of the unit 100, in which the unit 100 further comprises a microphone for input of voice signals to a suitable controller within the unit 100, wherein the controller has voice-recognition capabilities.

In the case of a cluster 700 whose broadcast unit 710 is no longer broadcasting (e.g. it is out of range of the receive units 730, or it is turned off), the cluster can maintain its remaining membership by selecting one of the receive units 730 to become the new broadcast unit 710. Such a choice can happen automatically, for example by random choice, by a voting scheme, or by choosing the first receive unit 730 to have become associated with the broadcast unit 710. If the users of the cluster-associated units deem this choice to be wrong, then they can change the broadcast unit 710 manually as described above.

The receive unit 730 that is chosen to become the broadcast unit 710 of the cluster 700 will generally prompt its user of the new status, so that the newly designated broadcast unit 710 can make certain that it is playing music to the rest of the cluster 700. It can be

LINKPLAY EXHIBIT 1004
Page 2933 of 4822

further arranged so that a newly-designated broadcast unit 710 will play music at random, from the beginning, or a designated musical piece in such case.

An embodiment of a transfer of broadcast control using IP socket communications protocols is described here. Figure 16 is a schematic block flow diagram of transfer of control between the broadcast unit 710 and the first receive unit 730. In a step 1130, the receive unit 730 requests broadcast control (designated here as "DJ" control). In a step 1132, the user of the broadcast unit 710 decides whether control will be transferred. The decision is then transferred back to the first receive unit 730 via the TCP messaging socket. If the decision is affirmative, the first receive unit 730 severs its UDP connection to the broadcast unit 710 multicast. The reason for this is to allow the receive unit 730 opportunity to prepare the beginning of its broadcast, if such time is required, and the user cannot both listen to the multicast as well as prepare its own audio selections, which occurs in a step 1136. In a step 1138, the receive unit 730 creates a multicast UDP socket with which it will later broadcast audio to other members of the cluster, while in a step 1140, the receive unit 730 creates a broadcast annunciator TCP socket with which to announce availability of the cluster, as well as to accept transfers of members from the broadcast unit 710 to itself as the new broadcast unit.

When the two new sockets (multicast and annunciator) are created, the receive unit 730 transmits the new socket addresses to the broadcast unit 710 in a step 1142. Since the other members of the cluster are guaranteed to be in contact with the broadcast unit, they can get addresses of the new, soon-to-be broadcast unit from the existing broadcast unit. In a step 1144, the original broadcast unit 710 transmits to the other cluster members (receive units 730 numbers 2-N) the addresses of the sockets on the receive 1 unit 730 that is now the new broadcast unit 710, and terminates its own multicast. The termination is performed here because the other receive units will be transferring to the new multicast, and because the original broadcast unit 710 is now becoming a receive unit 730 in the reconstituted cluster. In the step 1148, multicast of audio is now provided by the receive 1 unit 730 that has now become the new broadcast unit 710), and the original broadcast unit is listening to audio provided not by itself, but rather by the new broadcast unit.

In a step 1146, performed roughly synchronously with the step 1144, the original broadcast unit 710 transmits the socket addresses of the message handler TCP sockets of the other members of the cluster 700 (i.e. the receive units 730 numbers 2-N). In the subsequent step 1150, the original broadcast unit 710 and the receive units 730 numbers 2-N establish new messaging connections with the receive 1 unit 730 that is now the new broadcast unit

LINKPLAY EXHIBIT 1004
Page 2934 of 4822

710. While there can be a set of criteria for the acceptance of a new member to a cluster, because the receive 1 unit 730 has received the message socket addresses of the other members of the cluster in the step 1144, the receive 1 unit 730 accepts new members with the socket addresses received. It should be noted that instead of socket addresses being the identifiers passed, the identifiers can also be unique machine IDs, random numbers, cryptograpically encoded numbers, or other such identifiers that can be transmitted from one member of the cluster to another.

It should be noted in certain embodiments, that there can be insufficient time for the new broadcast unit 710 to determine a set of music to broadcast to the members of its cluster. It is within the spirit of the present invention for a user to set a default collection of music that is broadcast when no other music has been chosen. This set of music can comprise one or more discrete audio files.

Audio and DJ Choreography

One of the attractions of the present invention is that it allows users to express themselves and share their expressions with others in public or semi-public fashion. Thus, it is highly desirable for users to be able to personalize aspects of both the audio programming as well as the displays of their DJs 200.

Audio

Audio personalization comprises the creation of temporally linked collections of separate musical elements in "sets." These sets can be called up by name or other identifier, and can comprise overlapping selections of music, and can be created either on the unit 100 through a visual or audio interface, or can be created on a computer or other music-enabled device for downloading to the unit 100.

In addition, the unit 100 or other device from which sets are downloaded can comprise a microphone and audio recording software whereby commentary, personal music, accompaniment, or other audio recordings can be recorded, stored, and interspersed between commercial or pre-recorded audio signals, much in the manner that a radio show host or "disc jockey" might alter or supplement music. Such downloads can be accessible from a variety of sources including Internet web sites and private personal computers.

Automatic Generation of DJ 200 Control Signals

In this section, we will describe the automatic and manual generation of control signals for the DJ 200 transducers. The control signals are generally made to correspond to

audio signals played on the units 100, although it is within the spirit of the present invention for such control signals to be made separate from audio signals, and to be displayed on the digital jewelry independently of audio signals played on the unit 100. Figure 20 is a time-amplitude trace of an audio signal automatically separated into beats. Beats 1180, 1182 and 1183 are denoted by vertical dashed and dotted lines and, as described below, are placed at locations on the basis of their rapid rise in low-frequency amplitude relative to the rest of the trace. As can be seen, the beats 1180 are generally of higher amplitude than the other beats 1182 and 1183, and represent the primary beats of a 4/4 time signature. The beat 1183 is of intermediate nature between the characteristics of the beats 1180 and 1182. It represents the third beat of the second measure. Overall, the audio signal thus displayed can be orally represented as ONE-two-Three-four-ONE-two-Three-four ("one" is heavily accented, and the "three" is more lightly accented), which is common in the 4/4 time signature.

Processing of this data can proceed via a number of different methods. Figure 21A is a block flow diagram of a neural network method of creating DJ 200 transducer control signals from an audio signal as shown in Figure 20. In a step 1200, audio data is received either at the unit 100 or the DJ 200. It should be noted that the creation of control signals from audio signals can, within the present invention, take place at either the unit 100 or the DJ 200, or even at a device or system not part of or connected to the unit 100 or DJ 200 (as will be described in more detail below). In an optional step 1202, the data is low pass filtered and/or decimated so that the amount of data is reduced for computational purposes. Furthermore, the data can be processed for automatic gain to normalize the data for recording volume differences. Furthermore, the automatic gain filtering can provide control signals of significant or comparable magnitude throughout the audio data.

In general, the creation of the audio signal depends on audio representing a period of time, which can be tens of milliseconds to tens of seconds, depending on the method. Thus, the audio data from the step 1202 is stored in a prior data array 1204 for use in subsequent processing and analysis. At the same time, the current average amplitude, computed over an interval of preferably less than 50 milliseconds, is computed in a step 1208. In broad outline, the analysis of the signal compares the current average amplitude against the amplitude history stored in the prior data analysis. In the embodiment of Figure 21A, the comparison takes places through neural network processing in a step 1206, preferably with a cascading time back propagation network which takes into account a slowly varying time signal (that is, the data in the prior data array changes only fractionally at each computation, with most of the data remaining the same). The use of prior steps of neural network processing in the

LINKPLAY EXHIBIT 1004
Page 2936 of 4822

current step of neural network processing is indicated by the looped arrow in the step 1206. The output of the neural network is a determination whether the current time sample is a primary or a secondary beat. The neural network is trained on a large number of different music samples, wherein the training output is identified manually as to the presence of a beat.

The output of the neural network is then converted into a digital jewelry signal in a step 1210, in which the presence of a primary or secondary beat determines whether a particular light color, tactile response, etc., is activated. This conversion can be according to either fixed, predetermined rules, or can be determined by rules and algorithms that are externally specified. Such rules can be according to the aesthetics of the user, or can alternatively be determined by the specific characteristics of the transducer. For example, some transducers can have only a single channel or two or three channels. While light transducers will generally work well with high frequency signals, other transducers, such as tactile transducers, will want signals that are much more slowly varying. Thus, there can be algorithm parameters, specified for instance in configuration files that accompany DJ 200 transducers, that assist in the conversion of beats to transducer control signals that are appropriate for the specific transducer.

Figure 21B is a block flow diagram of a deterministic signal analysis method of creating DJ 200 transducer control signals from an audio signal as shown in Figure 20. The data is received in the step 1200. In this case, a running average over a time sufficient to remove high frequencies, and preferably less than 50 milliseconds, is performed in a step 1212. Alternatively, a low pass filter and/or data decimation as in the step 1202 can be performed.

In a step 1214, the system determines whether there has been a rise of X-fold in average amplitude over the last Y milliseconds, where X and Y are predetermined values. The value of X is preferably greater than two-fold and is even more preferably three-fold, while the value of Y is preferably less than 100 milliseconds and is even more preferably less than 50 milliseconds. This rise relates to the sharp rises in amplitude found in the signal at the onset of a beat, as shown in Figure 20 by the beat demarcations 1180, 1182, and 1183. If there has not been a rise meeting the criteria, the system returns to the step 1200 for more audio input.

If the signal does meet the criteria, it is checked to ensure that the rise in amplitude is not the "tail end" of a previously identified beat. For this, in a step 1216, the system determines whether there has been a previous beat in the past Z milliseconds, where Z is a predetermined value preferably less than 100 milliseconds, and even more preferably less

LINKPLAY EXHIBIT 1004
Page 2937 of 4822

than 50 milliseconds. If there has been a recent beat, the system returns to the step 1200 for more audio input. If there has not been a recent beat, then a digital jewelry signal is used to activate a transducer. The level of transduction can be modified according to the current average amplitude which is determined in a step 1208 from, in this case, the running average computed in the step 1212.

The embodiment of Figure 21B provides transducer activation signals at each rapid rise in amplitude, with the activation signal modulated according to the strength of the amplitude. This will capture much of the superficial musical quality of the audio signal, but will not capture or express more fundamental patterns within the audio signal.

Figure 21C is a schematic flow diagram of a method to extract fundamental musical patterns from an audio signal to create DJ 200 control signals. In the step 1200, the audio data is received into a buffer for calculations. In a step 1220, a low pass filter is applied to remove high frequency signal. Such high frequency signals can alternatively be removed via decimation, running averages, and other means as set forth in the embodiments of Figures 21A and B. As in the embodiment of Figure 21B, beat onsets are extracted from the audio signal in the steps 1214 and 1216, and a current average amplitude is computed in a step 1208.

The amplitudes and times of the onsets of beats are placed into an array in a step 1222. From this array, a musical model is created in a step 1224. This model is based on the regularity of beats and beat emphasis – as seen in the amplitudes – that is independent of the beats and amplitudes in any one short section of music (corresponding, for instance, to a measure of music).

In general, music is organized into repeating patterns, as represented in a time signature such as 3/4, 4/4, 6/8 and the like. Within each time signature, there are primary and secondary beats. In general, the downbeat to a measure is the first beat, representing the beginning of the measure. The downbeat is generally the strongest beat within a measure, but in any given measure, another beat may be given more emphasis. Indeed, there will be high amplitude beats that may not be within the time signature whatsoever (such as an eighth note in ¾ time that is not on one of the beats). Thus, by correlating the beats to standard amplitude patterns, the output to the music model identifies the primary (down) beats, secondary beats (e.g. the third beat in 4/4 time) and the tertiary beats (e.g. the second and fourth beats in 4/4/ time).

Figure 21D is a schematic flow diagram of an algorithm to identify a music model, resulting in a time signature. In a step 1600, the minimum repeated time interval is

LINKPLAY EXHIBIT 1004
Page 2938 of 4822

determined, using the array of beat amplitude and onset 1222. This is, over a period of time, the shortest interval for a quarter note equivalent is determined, wherein the time signature beat frequency (i.e. the note value of the denominator of the time signature, such as 8 in 6/8) is preferably limited to between 4 per second and one every two seconds, and even more preferably limited to between 3 per second and 1.25 per second. This is considered the beat time.

From the array of beat amplitudes and onsets 1222, the average and maximum amplitudes over a time period of preferably 3-10 seconds is computed in a step 1604. For the beginning of the audio signal, shorter periods of time can be used, though they will tend to give less reliable DJ 200 control signals. Indeed, in this embodiment, the initial times of an audio signal will tend to follow audio signal amplitude and changes in amplitude more than fundamental musical patterns until the patterns are elicited.

In a step 1606, the amplitude of a beat is compared with the maximum amplitude determined in the step 1604. If the beat is within a percentage threshold of the maximum amplitude, wherein the threshold is preferably 50% and more preferably 30% of the maximum amplitude, the beat is designated a primary beat in a step 1612. In a step 1608, the amplitude of non-primary beats is compared with the maximum amplitude determined in the step 1604. If the beat is within a percentage threshold of the maximum amplitude, wherein the threshold is preferably 75% and more preferably 50% of the maximum amplitude, and the beat is greater than a predetermined fraction of the average amplitude, wherein the fraction is preferably greater than 40% and even more preferably greater than 70% of the average beat amplitude, the beat is designated a secondary beat in a step 1614. The remaining beats are denoted tertiary beats in the step 1610.

In a step 1616, the sequence of the three types of beats is compared with that of established time signatures, such as 4/4, 3/4, 6/8, 2/4 and others, each with their own preferred sequence of primary, secondary and tertiary beats, in order to determine the best fit. This best fit is identified as the time signature in a step 1618.

Returning to Figure 21C, the channels of the DJ are pre-assigned to four different beats in a step 1225. Thus, if there are four channels, each channel is given a separate assignment. With a smaller number of channels, a single channel is assigned multiple beats. Some beats can also be unassigned, thus not being represented in a DJ 200 transducer output. Thus, a high jewelry signal, medium jewelry signal, low jewelry signal and an amplitude dependent signal are each assigned to a channel for DJ 200 transduction.

LINKPLAY EXHIBIT 1004
Page 2939 of 4822

In a step 1226, a beat determined to be a primary/down beat is assigned to a high jewelry signal 1228. In a step 1230, a beat determined to be a secondary beat is assigned to a medium jewelry signal 1232. In a step 1234, a beat determined to be a tertiary beat is assigned to a low jewelry signal 1236. Beats which are then unassigned, and which will generally be beats that occur not within the music model of the step 1224 (e.g. rapid beats not falling on beats of the time signature) are then assigned in a step 1238 to an amplitude dependent (and not music model dependent) signal 1240.

It should be noted that the computations performed in the flow methods of Figures 21A-C may take time on the order of milliseconds, such that if the computations are made in real time during the playing of music, the activation of the transducers in the DJ 200 are "behind" in time relative to the audio playing of the corresponding music in the audio unit 100. This can be compensated for by carrying out the computations while the audio signal is still in buffers prior to being played in the unit 100, as is described above for numerous embodiments of the present invention. Thus, signals to the DJ 200 can then be made simultaneously with respect to the audio signal to which it corresponds.

It should be noted that many of the parameters described above can conveniently be affected by manual controls either on the DJ 200 or the unit 100 that transmits signals to the DJ 200. For example, if can be convenient for the user to be able to set, for a given DJ 200 response amplitude, the threshold audio amplitude level at which the output transducer (e.g. light transducer 240) responds, or to set the output transducer amplitude corresponding to a maximum audio amplitude, or to set the frequency bands for which different DJ 200 channels respond, or to set other similar parameters. The manual controls for such parameters can comprise dials, rocker switches, up/down button, voice or display menu choices, or other such controls as are convenient for users. Alternatively, these choices can be set on a computer or other user input device, for download onto the unit 100 or DJ 200.

A preferable means of setting the parameters is for the parameters to be stored in a configuration file that can be altered either on the unit 100, the DJ 200 or a computer, so that the same DJ 200 can take on different characteristics dependent on the configuration settings within the file. The configuration settings can then be optimized for a particular situation, or set to individual preference, and be traded or sold between friends or as commercial transactions, for instance over the Internet. For a most preferable use of these configuration files, each file with its set of configurations can be considered to represent a "mode" of operation, and multiple configuration files can be set on the DJ 200 or the unit 100, depending on where the automatic generation of control signals is performed. The user can

LINKPLAY EXHIBIT 1004
Page 2940 of 4822

then select from the resident configuration files, appearing to the user as different modes, for use of his system, and can change the mode at will. This can be arranged as a series of choices on a voice or display menuing system, as a list toggled through by pressing a single button, or through other convenient user interfaces.

Manual Generation of DJ 200 Control Signals

In the description above, the use of filtering and digital modification of audio signals can be used to create control signals for DJ 200 transducers 240, 250, and 260. In addition, manual choreography of DJ 200 signals can be accomplished. For example, buttons or other interface features (e.g. areas on a touch-screen) on the unit 100 can correspond to different arrays of transducers, such as the LED arrays 290 and 292 of Figure 2A. While playing the audio signal, the user can press the buttons, where pressing of the buttons can correspond to a control signal for a transducer being ON, and otherwise the signal can be off. To aid in choreography where rapid changes in transducers are desired, the audio can be played at less than normal speed.

Figure 22A is a top-view diagram of an audio unit 100 user interface 1250, demonstrating the use of buttons to create DJ 200 control signals. The interface 1250 comprises a display screen (e.g. LCD or oLED), which can display information to the user, such as shown in Figures 18 A-B. Standard music control buttons 1254 for playing, stopping, pausing, and rewinding allow the user to control the audio signal musical output. Buttons 1252 further control aspects of the music output, such as volume control, musical tracks, and downloading and uploading of music. The number of buttons 1252 is conveniently three as shown, but can be more or less than three.

In addition, buttons are provided to allow the user to input DJ 200 control signals, comprising a record button 1256, a first channel button 1258, a second channel button 1260 and a third channel button 1262. The channel buttons 1258, 1260 and 1262 are prominent and accessible, since the user will want to easily depress the buttons. A record button 1256 allows the user to activate the channel buttons 1258, 1260 and 1262, and has a low profile (even below the nominal surface of the interface 1250) so that it is not accidentally activated. The record button can serve various purposes, including recording into a permanent storage file the sequence of DJ control signals relative to music being played, or controlling the DJ transducers in realtime, synchronously with music being played on the audio unit 100.

Pressing the buttons 1258, 1260 and 1262 create DJ control signals for the corresponding channels. The number of buttons is conveniently three as shown, but can also

LINKPLAY EXHIBIT 1004
Page 2941 of 4822

be two or four or more buttons. If a telephone is being used as the unit 100, keys on the telephone keypad can alternatively be used. The channel buttons will generally be used with thumbs, and the buttons are spaced so that two of the buttons can be depressed with a single thumb, so that all three buttons can be activated with only two fingers. In is also convenient for the two secondary buttons 1260 and 1262 to be spaced more closely together, as it will be a preferred mode of operation that the secondary buttons be operated together from time to time.

To further aid in the choreography of the DJs 200, a separate "keyboard" with the number of keys related to the number of possible arrays can be used. The amplitude of the corresponding transducer signal can be modified either according to the pressure on the keys, according to the length of time that a key is depressed, or according to a foot pedal. Figure 22B is a top-view diagram of a hand-pad 1270 for creating DJ control signals. The hand-pad 1270 comprises a platform 1271, a primary transducer 1272, a secondary transducer 1274 and a tertiary transducer 1276. The platform 1271 has a generally flat top and bottom, and can conveniently be placed on a table, or held in the user's lap. The size of the platform is such that two hands are conveniently placed across it, being preferably more than 6 inches across, and even more preferably more than 9 inches across. The pressure transducers 1272, 1274 and 1276 respond to pressure by creating a control signal, with said control signal preferably capturing both the time and amplitude of the pressure applied to the corresponding transducer. The primary transducer 1272 creates a primary control signal, the secondary transducer 1274 creates a secondary control signal and the tertiary transducer 1276 creates a tertiary control signal. The sizes and placements of the transducers can be varied within the spirit of the present invention, but it is convenient for the primary transducer 1272 to be larger and somewhat separate from that of the other transducers 1274 and 1276. In one more method of user interaction, both hands can be rapidly and alternately used to make closely spaced control signals on the primary transducer 1272. In addition, it can be convenient on occasion for the user to activate both the secondary transducer 1274 and the tertiary transducer 1276 with different fingers on one hand, and thus these can be conveniently placed relatively near to one another. In general, while a single transducer will provide minimal function, it is preferable for there to be at least two transducers, and even more preferable that there be three transducers.

The control signals can be transferred to the audio unit 100 for playing and/or storage, or to the DJ 200 unit directly for playing, either wirelessly, or through wired communication. In addition, the hand-pad can also be configured to create percussive or other sounds, either

LINKPLAY EXHIBIT 1004
Page 2942 of 4822

directly through the incorporation of hollow chambers in the manner of a drum, or preferably by the synthesis of audio waveform signals that can be played through the audio unit 100 (and other audio units 100 participating in a cluster 700), or directly through speakers within the hand-pad 1270 or attached to the hand-pad 1270 through wired or wireless communications. Such audible, percussive feedback can aid the user in the aesthetic creation of control signals.

It is within the spirit of the present invention for the hand-pad to take on various sizes and configurations. For instance, it is also convenient for the hand-pad 1270 to be configured for the use of index and middle fingers, being of dimensions as small as two by four inches or less. Such a hand-pad is highly portable, and can be battery powered.

Additionally, DJ 200 control signals can also be manually generated live, during broadcast at a party, for example, by a percussionist playing a set of digital drums. Figure 22C is a schematic block diagram of a set of drums used for creating DJ control signals. The set of drums comprises four percussive instruments 1280, 1282, 1284 and 1286, which can include snare drums, foot drums, cymbals, foot cymbals and other percussive musical instruments, such as might be found with a contemporary musical "band". Microphones 1290 are positioned so as to receive audio input primarily from instruments to which they are associated. One microphone can furthermore be associated with multiple instruments, as with the drums 1282 and 1284. The microphones 1290 are connected with a controller 1292 that takes the input and creates DJ control signals therefrom. For example, the drums 1282 and 1284 can be associated with the primary channel, the drum 1280 can be associated with the secondary channel, and the drum 1286 can be associated with the tertiary channel. The association of the microphone input with the channel can be determined in many ways. For example, the jack in the controller 1292 to which each microphone 1290 attaches can correspond to a given channel. Alternatively, the user can associate the jacks in the controller to different channels, with such control being manual through a control panel with buttons or touch control displays, or even through prearranged "sets". That is, a set is a pre-arranged configuration of associations of microphones to channels, and thus a set can be chosen with a single choice that instantiates a group of microphone-channel associations.

In general, the inputs from the microphones 1290 will be filtered in frequency and also to enhance audio contrast. For instance, control signals can be arranged to be the highest when the low-frequency envelope is rising the quickest (i.e. the beat or sound onset). The algorithms for conversion of audio signal to DJ control signal can be pre-configured in the controller 1292, or can be user selectable.

LINKPLAY EXHIBIT 1004
Page 2943 of 4822

It should be noted that the methods and systems of Figures 22 A-C need to synchronize the control signals so generated with the audio files to which they correspond. This can be accomplished in many ways. For example, the first control signal can be understood to correspond to the first beat within the audio file. Alternatively, the audio unit 100 or other device that is playing the audio signal to which the control signal is to correspond can send a signal to the device that is creating the control signals indicating the onset of playing of the audio file. The control signal can then be related to the time from the onset of the audio file. In addition, with regards to this synchronization, the user manually inputting the control signals will always be listening to the music during the control signal input. If the device on which control signals are being input is the same as the device that is playing the music, a control signal input cam be easily related to the sound that is currently being played by the audio output – many such devices allow information to within less than a millisecond of what sample or time within the audio files is currently being output by the audio device. With the arrangement of the control signal input device being also an audio player, close calibration of the control signals and the audio output is easily accomplished.

DJ 200 Control Signal Files

The control signals can be in a variety of formats within the spirit of the present invention. Such formats include pairs of locations within the associated music file and the corresponding amplitudes of the various DJ channels, and pairs of locations and the amplitudes of those DJ channels which are different from before. The locations can be either time from the start of the song (e.g. in milliseconds) or in terms of sample number. If the location is given in terms of sample number, the sample rate of the music will generally also be provided, since the same song can be recorded at different sample rates, and the invariant in terms of location will generally be time from onset of the music.

Other formats include an amplitude stream, corresponding to each DJ channel, provided in a constant stream with a fixed sample rate, which may be equal to or different from that of the corresponding music file. This format can be stored, for example, as additional channels into the music file, such that one channel corresponds to monoaural sound, two channels correspond to stereo sound, three channels correspond to stereo sound and one channel of control signals, and additional channels correspond to stereo sound plus additional channels of DJ control signals. Another arrangement is to allow for only a small number of states of the transduction in the control signal, so that multiple channels of control signal can be multiplexed into a single transmitted channel for storage and transmission with

LINKPLAY EXHIBIT 1004
Page 2944 of 4822

the audio signal. For example, if the audio is stored as a 16-bit signal, 3 channels of 5 bit DJ 200 control signal could be stored in a single channel along side the one or two audio channels normally used.

It should be appreciated that these different control signal storage formats are largely interchangeable. For instance, as described above, control signals can be stored as if they are additional audio channels within a music file, but then be extracted from the file for separate transfer (e.g. over the Internet), and then be reintegrated into an audio file at the destination location.

It should be appreciated that there are a number of means by which DJ 200 control signals can be generated, either automatically or manually, and can include the use of devices other than the unit 100 that can have sophisticated digital or analog filtering and modification hardware and software. The control signals so created can be stored in files that are associated with the music files (e.g. MP3) that the control signals are meant to accompany. To aid in their distribution, particularly in reference to limitations on the commercial and private distribution of the corresponding music files, the signal files will generally be separate from the music files, and transferable between units 100 either through inter-unit communication mediated by the inter-unit transmitter/receiver 110, or alternatively through computers or computer networks to which the unit 100 can be connected.

The audio signals and the DJ control signals should also be well synchronized during playback. Figure 23 is a schematic block flow diagram of the synchronized playback of an audio signal file with a DJ control signal file, using transmission of both audio and control signal information. For purposes of convenience in discussion, the audio signal file will be called a "song file" and the "control signal file" will be called a "dance file." In a step 1300, the user is provided a list of song files for display, preferably on the display 1170. In a step 1302, the user then selects a song from the display to play. In a step 1304, the dance files that are associated with the selected song file from the step 1302 are displayed for the user. These song files can be either locally resident on the unit 100, or can alternatively be present on other audio units 100 to which the audio unit 100 is connected, as in a cluster, or can alternatively be on the Internet, if the audio unit 100 is connected to the Internet. If there is a dance file that has been previously preferred in association with the song file, this file can be more prominently displayed than other associated dance files.

In a step 1306, the user selects the dance file to play along with the song file. This association is stored in a local database of song file/dance file associations in a step 1307, to be later used in a subsequent step 1304, should such an association not have been previously

LINKPLAY EXHIBIT 1004
Page 2945 of 4822

made, or if the preferred association is different from the previously preferred association. If the dance file is not locally resident, it can be copied to the audio unit 100 to ensure that the dance file is available throughout the duration of the song file playback.

In a step 1308, a timer is initialized at the beginning of the song file playback. In the step 1310, the song file is played on the local unit 100, and is also streamed to the other units 100 within the cluster 700. The corresponding DJ control signal accompanies the streaming song, either multiplexed within the song file audio signal, on another streaming socket, or through other communications (e.g. a TCP socket) channels between the two units. In a step 1312, the time advances along with the playback of the music. In a step 1314, this timer information is used to obtain current control signals from the dance file – that is, the dance file is arranged so that at each moment, the status of the different transducer channels can be determined. The control signals to be streamed along with the song file information can be either the current status of each transducer, or alternatively, can only send changes from the current transducer state.

The matching of the files in the database of song file and dance file associations of the step 1307 can be performed both within a machine, but also over a local or wide area network. In such cases, the association can either be external to the file – that is, using the name of the file, that is available the normal system file routines – or can use information internal to one or both files. For example, the dance file can have stored within it a reference to the song to which it is associated, either as the name of the song file, the name and/or other characteristics of the song (such as the recording artist, year of publication, music publisher) or alternatively as a numerical or alphanumerical identifier associated with the song. Then, given a song file, the relationship of the dance file with the song file can be easily determined.

For ease in creating an association, it is convenient for the names of the song files and the associated dance files to have a relationship with one another that is easily understood by casual users. For example, given a song file with the name "oops.mp3", it is convenient for an associated dance file to share the same root (in this case "oops") with a different extension, creating for example the dance file name "oops.dnc". Because of the multiplicity of dance files that will often be associated with a particular song file, the root itself can be extended to allow for either a numerical or descriptive filename, which can be preferably done in conjunction with a known punctuation mark to separate the song file root from the dance file description, such as the file names "oops.david2.dnc" or "oops$wild.dnc". It is

LINKPLAY EXHIBIT 1004
Page 2946 of 4822

preferable to use a punctuation mark that is allowed within a range of different operating systems.

Dance files can be stored on the Internet or other wide area network in a store for access by users who want dance files associated with a particular song file. In such case, if the storage is through the root of the filename, the user, requesting dance files corresponding to "oops.mps" would then be returned the names of related files such as "oops$wild.dnc". If the dance file internally carries the relationship with "oops.mps" as described above, either through the name or other characteristics, or alternatively, through a numerical or alphanumerical identifier, it is preferable to store the information in a database on the storage computer or unit 100, so that it is not necessary to open the file each time for perusal of the dance file information. Thus, if the music file has a substantially unique identifier associated with it internally, it is also useful for the dance file to also have the same identifier associated internally as well. In such case, the identifier is conveniently used to reference both files within a database.

In operation, a remote user would request a dance file for a particular song file by providing the name of the song file, along possibly with other information about the song file, which could include the name of the choreographer, the number of channels of DJ 200 transduction, the specific brand or type of DJ 200, or other information. The database would then return a listing of the various dance file that met the criteria requested. The remote user would then choose one or more of the files to download to the remote computer, and then the database would retrieve the dance files from storage and then transmits the dance file over the wide area network. On the remote computer or unit 100, the dance file would become associated with the corresponding song file through means such as naming the dance file appropriately or making an association between the song file and the dance file in a database or indexing file. Alternatively, the dance file can be integrated into the song file as mentioned elsewhere within this specification.

It can be useful to preview a dance file for its desirability or suitability. Since the dance files can be retrieved from a wide area network such as the Internet, it is convenient for such an emulator to operate on a computer that may not be portable or have the proper transmitter that allows communications with a DJ 200. In such case, it is preferable to have an emulator which places an image or drawing of a DJ 200 on the screen, and which is provided the name of a song file and a dance file, and which then plays the song file through the audio of the computer and displays appropriate images or drawings of transducers being activated within the emulator image or drawing. The characteristics of the DJ 200 being

LINKPLAY EXHIBIT 1004
Page 2947 of 4822

emulated (e.g. colors of lights, frequency responses, levels of illumination, arrangement of lights, response to amplitude, etc.) can be simulated by a number of means. For example, the user can move slider controls, set checkboxes and radio boxes, enter numerical values, click-and-drag icons and use other standard user interface controls to make the DJ 200 operate as desired. Alternatively, manufacturers of DJ 200s can create configuration files (including, for example, bitmaps of photos of the actual DJ 200) that can be downloaded for this purpose (and which can also be used by prospective purchasers to view the "virtual" operation of the DJ 200 prior to purchase, for example, through an Internet merchant). The configuration files would contain the information necessary for the emulator to properly display the operation of the specific DJ.

Alternatively, as described above, the dance file information can be stored within the song file as, for example, another channel in place of an audio channel, or alternatively within MP3 header or other file information. In such case, the step 1307 would have the alternative function of looking through song files to find the song file with the particular desired embedded dance file within.

In addition to sending dance files from computers to units 100 or between units 100, the dance files can be streamed from unit 100 to unit 100 through the normal unit-to-unit communications, in the manners described above for audio communications. This is particularly convenient given that DJ 200 displays can be used to show group identification, and such displays can be more effective if the DJs for each user are nearly identical (which might not be the case if the users were using, for example, different dance files). The dance file control signal information can be transmitted in a variety of ways, including multiplexing the control signals into the same packets as the audio information as if it were a different audio channel, alternating packets of control signals with packets of the audio information, or broadcasting control signals on a different UDP socket as the audio. Alternatively, if the receiving unit has a copy of the dance file corresponding to the song file being transferred by unit-to-unit communication, the receiving unit can determine the current time being played, and to extract from the local dance file the control signals for the receiving unit DJ 200.

It should be known that most streaming protocols have relatively small data packets that are communicated, due to the fact that reception at the source is not guaranteed and it is not desirable to lose a large amount of information in any one stream. Thus, it is possible with smaller transmission buffers and higher data rates to send a single DJ control signal in each transmission. For example, with a buffer size of 600 bytes, and an audio rate of 22,050 Hz with two single byte channels, each transmission covers only about 12 milliseconds, and

any signal would therefore be at most 13 milliseconds from its correct time. Alternatively, each control signal can be accompanied by an offset in time from the beginning of the transmitted audio signal. Also, the time or packet number of each transmission buffer can be sent, as well as the time or packet number of the DJ audio signals, so that the audio unit 100 can compute the proper offset.

Stationary Transducers

DJs 200 that have been previously described are portable devices, usually associated with a particular user and unit 100. Figures 5A and 5B indicate the ways in which DJs 200 associated with multiple users can be controlled by a single unit 100.

It is also convenient for transducers to be non-portable and stationary. Consider, for example, a user who is at home listening to music. Instead of a DJ 200 worn by the user, the user can alternatively have a bank of lights or other transducers in fixed locations through the room that operate under the same or similar control signals as to which DJs respond. Such fixed transducers can operate at far higher power than portable DJs 200, and can each incorporate a large number of separate transducers.

Furthermore, in a party, concert or other large social gathering, the effects of portable DJs worn by guests can be supplemented by large transducers that are generally perceptible by most guests. For example, such transducers can include spark or smoke generators, strobe lights, laser painters, arrays of lights similar to Christmas light strings, or mechanical devices with visible (e.g. a flag waving device) or tactile effects (e.g. a machine that pounds the floor). In general, transducers for large gatherings will not communicate with a unit 100, but will be directed by a wide-area broadcast unit 360, as in Figure 5B.

Because of the large area over which such stationary transducers can operate, the communications between the unit 100 and the stationary transducers can be through wired rather than wireless transmission. Furthermore, there can be mixed communication, such as wireless transmission of control signals from a portable unit 100 to a stationary receiver, and thence wired transmission to one or multiple transducers.

Modular Configurations

In the embodiments above, the audio player 130 is directly integrated with the inter-unit and unit-to-DJ communications. This requires both a re-engineering of existing audio players (e.g. CD, MP3, MO and cassette players), and furthermore does not allow the communications functionality to be reused between players.

LINKPLAY EXHIBIT 1004
Page 2949 of 4822

An alternative embodiment of the present invention is to place the communications functions external to the audio playing functions, and to adjustably connect the two via the audio output port of the audio player. Figure 12A is a schematic diagram of a modular audio unit 132. Audio player 131 is a conventional audio player (e.g. CD or MP3 player) without the functionality of the present invention. Analog audio output is sent via audio output port 136 through the cable 134 to the audio input port 138 of the modular audio unit 132. The modular audio unit 132 comprises the inter-unit transmitter/receiver 110 and the DJ transmitter 120, which can send and receive inter-unit and unit-to-DJ communications in a manner similar to an audio unit 100. A switch 144 chooses between audio signals from the audio player 131 and from the inter-unit transmitter/receiver 110 for output to the output audio port 142 to the earphone 901 via cable 146 (the earphone 901 can also be a wireless earphone, wherein the output port 142 can be a wireless transmitter, which can also be a DJ transmitter 120). A convenient configuration for the switch 144 is a three way switch. In an intermediate position, the unit 132 acts simply as a pass-through, in which output from the audio player 131 is conveyed directly to the earphone 901, and the transmitter/receiver functions of the unit 132 do not operate. In another position, the unit 132 operates as a receiver, and audio from the inter-unit transmitter/receiver 110 is conveyed to the earphone 901.

When the combined system operates as a broadcast unit 710, audio input from the audio unit 131 is directed to the inter-unit transmitter/receiver 110 for transmission to receive units 730, as well as for output to the earphone 901 (which can be direct to the earphone 901 through the switch, or indirectly through the inter-unit transmitter/receiver 110).

When the combined system operates as a conventional audio player, the switch directs audio signals from the input port 138 directly through to the output port 142. In this mode of operation, it can be arranged for the audio output to traverse the modular audio unit 132 without the unit being powered up. In case there is a transmission delay to the receive unit 730 such that audio played locally through the earphone 901 and audio played remotely on the receive unit 730 are not in synchrony, the system can incorporate a time delay in the output port 142 such that the local and remote audio output play with a common time delay, and are thus in synchrony.

When the combined system operates as a receiver unit 730, audio input from the input port 138 is ignored, and signals to the audio output port are delivered solely through the inter-unit transmitter/receiver 110.

LINKPLAY EXHIBIT 1004
Page 2950 of 4822

It is convenient for the modular audio unit 132 to be able to operate independently of the associated audio player 131. In such a case, the unit 132 must have an independent energy store, such as one or more batteries, which can be rechargeable. In that case, the unit 132 has no audio signals locally to listen to through the earphone 901 or to transmit over the transmitter/receiver 110. However, the unit 132 can in that case receive external audio signals sent by other units 132 or units 100 for listening.

The audio player 131 can be placed in a backpack, purse, or other relatively inaccessible storage location, while the modular audio unit is, like a "remote control", accessible for interaction with other users.

Video

While the units 100 described above have comprised audio players 130, within the spirit of the present invention, such units can also comprise video or audio/visual players (both of which are referred to below as video players). Such video players would be used generally for different entertainment and educational purposes, not limited to films, television, industrial training and music videos. Such video enabled units can operate similarly to audio units, including the capability of sharing video signals, synchronously played, with nearby units through inter-unit communication, as well as the use of DJ's that can produce human-perceptible signals (such as light transduction for accompaniment of audio signals in music videos). It should be noted, however, that there is a larger bandwidth requirement for the inter-unit transmitter/receiver 110 for the communication of video signals as compared with audio signals. In the case of shared video, wire connections (e.g. FireWire) between two units can allow simultaneous viewing of a single video signal.

In addition, text, including language-selectable closed caption and video subtitling, can accompany such video, as well as chat or dubbing to allow the superposition of audio over the audio normally accompanying such video.

Music Distribution Using Audio Units

The music industry is suffering from reduced sales due to the advent of Internet-based music file sharing; in addition, the manufacturers of personal audio devices are bringing to market audio devices that can wirelessly transfer music files between the devices. Such sharing-enabled devices could significantly reduce the sales of music. Audio units of the present invention, however, can be used to provide new means of music distribution and thereby increase the sales of music.

LINKPLAY EXHIBIT 1004
Page 2951 of 4822

Figure 25 is a schematic flow diagram indicating music sharing using audio devices, providing new means of distributing music to customers. Three entities are involved in the transactions – the DJ (operating a broadcast unit 710), the cluster member (operating a receive unit 730), and the music distributor, and their actions are tracked in separate columns. In this case, the term DJ is used to indicate the person operating a broadcast unit 710, and has no meaning with respect to a DJ unit 200. Indeed, the DJ unit 200 is a part of the system only inasmuch as it provides for heightened pleasure of the DJ and the member in enhancing their experience of the music. For the rest of this section, DJ will refer specifically to the person operating the broadcast unit 710.

In a first step 1340, the DJ registers with the distributor, who places information about the DJ into a database in a step 1342. Part of this information is a DJ identifier (the DJ ID), which is unique to the DJ, and which DJ ID is provided to the DJ as part of the registration process. This ID is stored in the unit 100 for later retrieval. The DJ at some later time broadcasts music of the type distributed by the distributor, in a step 1344. The broadcast of the music by the DJ can be adventitious (that is, without respect to the prior registration of the DJ with the distributor), or the distributor can provide the music to the DJ either free of charge, at a reduced charge, or free of charge for a limited period of time.

In a step 1346, the member becomes a part of the cluster 700 of which the DJ is the broadcaster broadcasting the distributor's music, and has thereby an opportunity to listen to the music. Along with the transfer of the audio signal of the music, in a step 1348, the DJ can send information about the song, which can include a numerical identifier of the music or album from which the music is derived. Furthermore, the DJ ID is provided to the member, and is associated with the music ID and stored in a database on the member unit 100 in a step 1350. In order to prevent this database from becoming too sizable, music IDs and DJ IDs can be purged from it on a regular basis (for example, IDs which are older than 60 or 120 days can be removed).

If the member requests purchase of the music from the distributor in a step 1352, in a step 1354, the distributor stores the member information, the music ID, and the DJ ID associated with the music (i.e. the person who introduced the member to the music). The distributor then completes the transaction with the member, providing a copy of the music in exchange for money, in a step 1356. As the member receives the music copy, he also becomes registered as a DJ as well in a step 1358. Thus, if the member now becomes the DJ of his own cluster, and introduces people to this music, he will also be known to the distributor as an introducer of the music.

LINKPLAY EXHIBIT 1004
Page 2952 of 4822

In a step 1360, the distributor provides points to the DJ who introduced the member to the music and facilitated the sale of the music. In a step 1362, the DJ accumulates points related to the sale of the music to the member, as well as points related to the sale of other music to other members. These points can at that point or later be redeemed for money, discounted music, free music, gifts, access to restricted activities (e.g. seats at a concert) or other such real or virtual objects of value to the DJ.

In a step 1364, the DJ is optionally further linked to the music and member for whom he has received points. If this member introduces the music to yet other members, who are induced to buy the music from the distributor, the DJ is further awarded points in a step 1366, given that the "chain" of members introduced directly or indirectly to the music includes the original DJ.

This set of interactions does not decrease music sales as does file sharing, but rather increases sales of music, as the DJ has incentives to encourage others to buy the music, and the offering of the music by the DJ through his broadcasts introduces music to people who may not have already had the opportunity to hear the music.

Figure 31 contains tables of DJ, song and transaction information according to the methods of Figure 25. A USER table 1810 comprises information about the USER, which can include the name of the person (Alfred Newman), their nickname/handle ("WhatMeWorry"), their email address (AEN@mad.com), and the machine ID of their unit 100 (B1B25C0). This information is permanently stored in the audio unit 100. A second set of information relates to music that the USER has heard while in other clusters 700 that the USER liked, and which is indicated as the USER's "wish list". This set of information includes a unique ID associated with the song (or other music or audio signal), which is transmitted by the broadcast unit 710 of the cluster 700. This information can alternatively or additionally include other information about the music, such as an album name, an artist name, a track number, or other such information that can uniquely identify the music of interest.

Along with each song ID is a DJ identifier, indicating the unique ID associated with the DJ who introduced the desired music to the USER. Additionally or alternatively, the information can comprise the DJ's email address, personal nickname/handle, name, or other uniquely identifying information.

The Wish List can either be permanent, or it can be that each song entry is dated, and that after a predetermined amount of time, which can be set by the user, the songs that are still on the Wish List are removed. It is also convenient that songs that are purchased

according to the methods of the present invention, such as Figure 25, are also removed from the list automatically.

A DISTRIBUTOR table 1812 comprises information about purchases made by USERS with the DISTRIBUTOR. The table 1812 has numerous records keyed according to unique USER identifiers, which in this case is the MAC ID of the unit 100. A single record from the table is provided, of which there can be hundreds of thousands or millions of such records stored.

The record can include contact information about the USER, including name, email address, or other business related information such as credit card number. In addition, each record comprises a list of all of the songs known to have been purchased through the DISTRIBUTOR, as identified by a unique song ID. In addition, the DJ associated with the purchase of the given song by the USER is also noted. This information was previously transmitted from the USER table 1810, which includes the associated DJ identifier along with the song identifier, at the time of purchase of the song. This association allows the DISTRIBUTOR to compensate the DJ for his part in introducing the USER to the song.

It should also be noted that such an arrangement of information allows the compensation, if desired, of the individual who introduced the DJ to the song, prior to the DJ introducing the USER to the song. For example, when the user purchased the song with song ID 230871C40, points were credited with the DJ whose ID is 42897DD. Looking in the record for the DJ 42897DD, one can determine whether there is another individual (DJ) associated with the purchase of the song 230871C40 by the DJ. If so, that individual can also receive compensation for the purchase of the song by the USER.

Use of Internet Connections

It is within the teachings of the present invention to allow normal Internet connections of the audio unit 100 with non-mobile devices connected with the Internet. Figure 29A is a schematic block diagram of the connection of an Internet-enabled audio unit 100 with an Internet device through the Internet cloud 1708, using an Internet access point 1704. An Internet-enabled audio unit 1700, unit A, is wirelessly connected to an audio unit 100, denoted unit B, as members in a cluster 700. The dashed line connecting the two units A and B indicates that the connection is wireless, whereas the solid connecting lines indicate wired connections. The unit A is connected to a wireless access point 1704, such as an 802.11 access point, which is connected to an Internet device 1706 via wired connections through the Internet cloud 1708.

Figure 29B is a schematic block diagram of the connection of an Internet-enabled audio unit 1702 with an Internet device through the Internet cloud, with an audio unit 1702 directly connected to the Internet cloud 1708. In this case the audio unit 1702 is capable of directly connecting to the Internet cloud 1708, and thence to the Internet device 1706, through a wired connection. This could be through a high speed connection (such as a twisted wire Ethernet connection) or through a lower speed connection (e.g. a serial port connection, or a dial-up modem).

The connection of the unit 1700 or unit 1702 is illustrated in Figure 30, tables of ratings of audio unit 100 users. As described above, members of a cluster can decide whether or not to admit a new member to the cluster using a variety of automatic or manual methods. One method of determining the suitability of a user to become a member of the cluster 700 is to determine the user's ratings by members of other clusters to which the user has previously been a member. In this case, the Internet device 1706 is a computer hosting a database, which can be queried and to which information can be supplied by the unit A (either 1700 or 1702). On the Internet device 1706 are stored ratings of units 100, as indicated by the table 1802. The left hand column is the primary key of the database, and is a unique identifier associated with each unit 100. This ID can be a numerical MAC ID, associated with the hardware and software of each unit 100, a unique nickname or word handle (e.g. "Jen412smash") associated with each audio unit user, or other such unique identifier.

The second and third columns, indicated as numbers with dollar signs, are the total summed positive ratings (column two) and the negative ratings (column three) registered with each user by another member of a cluster 700 with which the user has been associated, and in which the user was operating the broadcast unit 710. This rating can, for example, reflect the perceived quality of music provided by the user. The fourth and fifth columns are the total, summed ratings of the user by other members of clusters 700 with which the user has been associated, in which the user was the operator of a receive unit 730. This rating can, for example, indicate the good spirits, friendliness, dress or other characteristics of the user as perceived by other members of the cluster. The sixth column indicates the largest cluster 700 for which the user has been the broadcaster. This is a good indicator of a broadcaster's popularity, since a poor or unpopular broadcaster would not be able to attract a large group of members for a cluster.

There are many other characteristics that can be stored in such a database, and can also include IDs of other members of groups with which the user has been associated (so that members can accept new members who have been associated with friends of those in the

LINKPLAY EXHIBIT 1004
Page 2955 of 4822

cluster), specific music that the user has played (in order to determine musical compatibility), information on the individuals making each rating (in order to determine rating reliability), and gradations of ratings (rather than simply a positive or negative response).

The cluster members can access the ratings of the user requesting membership in the cluster 700 in order to determine their desirability and suitability. This would require a connection with the Internet device 1706 at the time that the user was requesting to join, and would preferably involve a wireless connection through an access point, as in Figure 29A. The information from the database on the device 1706 can either be displayed to the members of the cluster 700, or can be used by an automatic algorithm to determine whether the person can join.

The table 1800 represents the ratings of a cluster 700 of 5 total members (comprising a broadcaster with ID 12089AD, and four additional members with IDs E1239AC, F105AA3, B1B25C0, and ED5491B). The ratings are supplied by ED5491B (whose ID is preceded by a zero), and then specific ratings of each member are made. The DJ is indicated by a dollar sign preceding his ID. These ratings can be made by putting the nicknames/handles of the cluster members on a screen, and allowing the member to indicate positive or negative ratings by pressing one of two buttons. A plus in the first column indicates a positive response, and a minus sign indicates a negative response. These ratings can then be sent during either wired communications directly to the Internet device 1706 or via the access point 1704. It should be noted that the ratings, once made, can be stored on the unit 1700 or 1702 indefinitely, until connection with the Internet cloud 1708 can be made. As indicated by the arrow, the information for B1B25C0 can be added to the table 1802 -- in this case, by incrementing the value in the fourth column (a positive rating for a user who is not the broadcaster).

Other applications of connections to Internet devices 1706 include exchanging (via uploading and downloading) dance files with distant individuals, and obtaining music via downloading , which can include transactions with distributors similar to that seen in Figure 25. Such connections also allow the integration of other connectivity, such as telephone and messaging capabilities, expanding the usefulness and attractiveness of audio units 100.

Many Embodiments Within the Spirit of the Present Invention

It should be apparent to one skilled in the art that the above-mentioned embodiments are merely illustrations of a few of the many possible specific embodiments of the present invention. For example, the elements of a unit 100, including the inter-unit transmitter/receiver 110 protocol and hardware, the DJ transmitter 120 and the audio player

LINKPLAY EXHIBIT 1004
Page 2956 of 4822

130 can be chosen from a range of available technologies, and can be combined with user interface elements (keyboards, keypads, touch screens, and cursor buttons, without significantly affecting the operation of the unit 100. Furthermore, many different transducers can be combined into DJs 200, which can further comprise many decorative and functional pieces (e.g. belt clasps, functional watches, microphones, or wedding rings) within the spirit of the present invention. Indeed, the unit 100, itself, can comprise transducers 240, 250 or 260.

It should also be appreciated that communications protocols provide a nearly uncountable number of arrangements of communications links between units in a cluster, that the links can be of mixed software protocols (e.g. comprising both TCP and UDP protocols, and even non-IP protocols) over a variety of hardware formats, including DECT, Bluetooth, 802.11 a, b, and g, Ultra-Wideband, 3G/GPRS, and i-Beans, and that communications can include not only digital but also analog communications modes. Furthermore, communications between audio units and digital jewelry can further comprise analog and digital communications, and a variety of protocols (both customized as well as well-established IP protocols).

It is important, as well, to note that the inter-unit communication and the unit-to-DJ communication can operate and provide significant benefits independently of one another. For example, members listening to music together gain the benefits of music sharing, even without the use of DJs 200. Alternatively, an individual's appreciation of music and personal expression can be augmented through use of a DJ 200, even in the absence of music sharing. However, the combination of music sharing along with enhanced personal expression through a DJ 200 provides a synergistic benefit to all members sharing the music.

Numerous and varied other arrangements can be readily devised by those skilled in the art without departing from the spirit and scope of the invention. Moreover, all statements herein reciting principles, aspects and embodiments of the present invention, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof. Additionally, it is intended that such equivalents include both currently known equivalents as well as equivalents developed in the future, i.e. any elements developed that perform the same function, regardless of structure.

In the specification hereof any element expressed as a means for performing a specified function is intended to encompass any way of performing that function. The invention as defined by such specification resides in the fact that the functionalities provided by the various recited means are combined and brought together in the manner which the

LINKPLAY EXHIBIT 1004
Page 2957 of 4822

specification calls for. Applicant thus regards any means which can provide those functionalities as equivalent as those shown herein.

LINKPLAY EXHIBIT 1004
Page 2958 of 4822

Claims

What is claimed is:

1. A method for sharing music from stored musical signals between a first user with a first music player device and at least one second user with at least one second music player device, comprising:

playing the musical signals for the first user on the first music player device while essentially simultaneously wirelessly transmitting the musical signals from the first music player device to the at least one second music player device; and

receiving the musical signals by the at least one second player device, wherein the musical signals can be played on the at least one second player device essentially simultaneously with the playing of the musical signals on the first music player device;

wherein the first and the at least one second users are mobile and maintain less than a predetermined distance.

2. The method of claim 1 wherein the predetermined distance is 40 feet.

3. The method of claim 1 wherein the predetermined distance is 100 feet.

4. The method of claim 1 wherein the predetermined distance is 500 feet.

5. The method of claim 1 wherein the transmitting and receiving involve the use of an Internet protocol.

6. The method of claim 5 wherein the Internet protocol uses a connection-less transport layer.

7. The method of claim 5 wherein the Internet protocol uses a link layer involving a protocol chosen from the group consisting of 802.11, Bluetooth, ultrawideband, HiperLAN and WDCT.

8. The method of claim 1 wherein the transmitting and receiving use analog radio frequency transmission.

9. The method of claim 1 wherein the transmitted music signals are digital.

10. The method of claim 9 wherein the transmitted digital music signals are compressed.

11. The method of claim 1 wherein the first music player device is carried by the first user.

12. The method of claim 1, additionally comprising displaying by the first user a humanly-perceptible light display that is roughly synchronized with the beat of the musical signals.

LINKPLAY EXHIBIT 1004
Page 2959 of 4822

13. The method of claim 1, additionally comprising communicating an identifier of the at least one second user to the first music player device, and indicating the identifier to the first user on a display located on the first music player device.

14. The method of claim 1, additionally comprising transferring a substantially unique music identifier of the musical signal wherein the personal identifier is stored on the at least one second music player device in association with the musical identifier.

15. The method of claim 14 wherein the user of the at least one second music player device retrieves the musical signal from a remote system by using the music identifier.

16. The method of claim 15 wherein the remote system is located on the Internet.

17. The method of claim 15, additionally comprising a substantially unique personal identifier of the first user wherein the personal identifier is transferred to the at least one second music player device and stored in association with the music identifier of the music transmitted by the first music player device to the at least one second music player device.

18. The method of claim 1, additionally comprising monitoring the strength of the musical signal received by the at least one second music player device and communicating with the at least one second user the strength of the musical signal.

19. A system of music sharing for a plurality of users, comprising:

a first sharing device and at least one second sharing device, each comprising a musical signal store, a musical signal transmitter, a musical signal receiver, and a musical signal player;

a broadcast user operating the first sharing device; and

at least one member user operating the at least one second sharing device, wherein the broadcast user plays the musical signal for his own enjoyment on the first sharing device and simultaneously transmits the musical signal to the receiver of the at least one second sharing device of the at least one member user, on which the musical signal is played for the at least one member user, wherein the broadcast user and the at least one member user hear the musical signal substantially simultaneously.

20. The system of claim 19, additionally comprising a manual switch on at least one second sharing device, wherein actuation of the switch causes the first sharing device to cease transmitting the musical signal and for the second sharing device to begin transmitting musical signal from the musical signal store of the second sharing device, wherein the broadcast user and the at least one member user hear the musical signal from the second sharing device substantially simultaneously.

LINKPLAY EXHIBIT 1004
Page 2960 of 4822

21. The system of claim 19 wherein the first sharing device and the at least one second sharing device additionally comprise a substantially unique identifier, wherein the first sharing device communicates its substantially unique identifier to the at least one second sharing device and the at least one second sharing device communicates its substantially unique identifier to the first sharing device.

22. The system of claim 21, wherein the identifier includes a least one piece of information about the user of the sharing device taken from the group consisting of the name of the user, the nickname of the user, a numerical identifier, an alphanumerical identifier, the musical preferences of the user, the age of the user, and the educational institution that the user attends.

23. The system of claim 22 wherein the first sharing device and the at least one second sharing device additionally comprise a visual display on which the substantially unique identifiers which have been communicated to the respective device are displayed.

24. The system of claim 19 additionally comprising at least one requesting member user operating an at least one requesting sharing device, wherein the at least one requesting member user requests to receive transmission of the musical signal from the first sharing device.

25. The system of claim 24 wherein the at least one requesting sharing device is associated with a substantially unique identifier which is communicated by the requesting sharing device to the first sharing device and wherein the first sharing device additionally comprises a display on which the identifier is displayed.

26. The system of claim 25 wherein the broadcast user elects whether to transmit the musical signal from the first sharing device to the requesting sharing device.

27. The system of claim 25 wherein the broadcast user and the at least one member user elect through voting whether to transmit the musical signal from the first sharing device, wherein a majority vote is determinative.

28. The system of claim 25 wherein the broadcast user and the at least one member user elect through voting whether to transmit the musical signal from the first sharing device, wherein a unanimous vote is required for transmitting.

29. The system of claim 25 wherein the identifier of the requesting sharing device is compared with identifiers stored in the first device, such that the match of the identifier of the requesting sharing device with a stored identifier allows the first sharing device to transmit the musical signal to the requesting sharing device.

LINKPLAY EXHIBIT 1004
Page 2961 of 4822

30. The system of claim 29 wherein the stored identifiers are retrieved from the Internet.

31. The system of claim 19 wherein the first sharing device additionally comprises a visual display.

32. The system of claim 31 wherein the visual display indicates the number of member users receiving the musical signal.

33. The system of claim 31 wherein the visual display can display whether the musical being played by the sharing device comes from the musical signal store of the same device, or from the musical signal store another sharing device.

34. The system of claim 19 wherein the first sharing device is mobile.

35. The system of claim 19 wherein the first sharing device is carried by the user.

36. The system of claim 19 wherein the first sharing device and at least one sharing device maintain less than a predetermined distance.

37. The system of claim 36 wherein the predetermined distance is 40 feet.

38. The system of claim 36 wherein the predetermined distance is 100 feet.

39. The system of claim 36 wherein the predetermined distance is 500 feet.

40. The system of claim 19, additionally comprising a substantially unique user identifier associated with the broadcast user and a substantially unique musical signal identifier associated with each musical signal, wherein the user identifier and the musical signal identifier are transmitted from the first sharing device to the at least one second sharing device and stored in association with one another on the at least one second sharing device.

41. A wireless communications system for sharing audio entertainment between a first mobile device and a second mobile device in the presence of a non-participating third mobile device, comprising:

an announcement signal transmitted by the first mobile device for which the second mobile device and the third mobile device are receptive;

a response signal transmitted by the second mobile device in response to the announcement signal for which the first mobile device is receptive and for which the third mobile device is not receptive;

an identifier signal transmitted by the first mobile device to the second mobile device in response to the response signal, and which is not receptive to the third mobile device; and

a broadcast signal comprising audio entertainment that is transmitted by the first mobile device, and which is receptive by the second mobile device on the basis of the reception of the identifier signal.

LINKPLAY EXHIBIT 1004
Page 2962 of 4822

42. The system of claim 41 wherein the announcement signal is transmitted via an Internet protocol.

43. The system of claim 42 wherein the Internet protocol is a connection-oriented transport protocol.

44. The system of claim 43 wherein the connection-oriented protocol is TCP.

45. The system of claim 42 wherein the Internet protocol further comprises a link layer protocol selected from the group consisting of 802.11, Bluetooth, ultrawideband, HiperLAN and WDCT.

46. The system of claim 41 wherein the broadcast signal is transmitted via an Internet protocol.

47. The system of claim 46 wherein the Internet protocol is a connectionless transport protocol.

48. The system of claim 47 wherein the connectionless transport protocol is UDP.

49. The system of claim 46 wherein the Internet protocol further comprises a link layer protocol selected from the group consisting of 802.11, Bluetooth, ultrawideband, HiperLAN and WDCT.

50. The system of claim 41 wherein the broadcast signal is an analog radio frequency signal.

51. The system of claim 41 wherein the broadcast signal is transmitted via telephony.

52. The system of claim 41 wherein the broadcast signal is compressed.

53. The system of claim 52 wherein the compression is MP3.

54. The system of claim 41 wherein the broadcast signal is encrypted.

55. The system of claim 41 wherein the identifier signal is an IP socket address.

56. The system of claim 41 wherein the identifier signal is a decryption code.

57. The system of claim 41 wherein the identifier signal is a digital password.

58. The system of claim 41 wherein the announcement signal and the broadcast signal use different IP transport protocols.

59. The system of claim 41 additionally comprising a transfer signal that is transmitted by the first mobile device upon halting the transmission of its broadcast signal, and which transfer signal is receptive by the second mobile device, such that reception by the second mobile device causes the device to transmit a second broadcast signal comprising audio entertainment for which the first unit is receptive.

60 The system of claim 41 wherein the broadcast signal is transmitted via peer-to-peer communications.

61 The system of claim 60, additionally comprising a switch, wherein actuation of the switch causes the device to toggle between peer-to-peer communications and infrastructure-based communications.

62 The system of claim 41 wherein the broadcast signal is transmitted via infrastructure-based communications.

63. An audio entertainment device, comprising:

a signal store that stores an audio entertainment signal;

a transmitter that can transmit the stored audio entertainment signal;

a receiver that can receive the transmitted audio entertainment signal from a transmitter of another such device; and

a player that can play audio entertainment from a member selected from the group of stored audio entertainment signals or audio entertainment signals transmitted from the transmitter of another such device.

64. The device of claim 63 wherein the transmitter transmits to the receiver of the same device in order to play through the player of that device.

65. The device of claim 63 wherein the player can play audio entertainment from signals obtained from either the receiver or from the audio store.

66. The device of claim 63 additionally comprising a visual display that is perceptive to the user of the device, indicating information about the audio entertainment signal.

67. The device of claim 66 wherein a third device can receive the transmitted audio signal, wherein all such devices transmitting and receiving the same audio entertainment signal constitute a sharing group.

68. The device of claim 67 additionally comprising a substantially unique identifier that identifies the user of the device, wherein at least one device of the sharing group displays the identifier of the user of another device within the sharing group.

69. The device of claim 68 wherein the identifier includes a least one piece of information about the user taken from the group consisting of the name of the user, the nickname of the user, the musical preferences of the user, a numerical identifier, an alphanumerical identifier, the age of the user, or the educational institution that the user attends.

70. The device of claim 68 wherein the identifier is interspersed within the transmitted audio entertainment signal.

71. The device of claim 66 wherein the visual display can display the number of devices in the sharing group.

72. The device of claim 66 wherein the visual display can display whether the audio entertainment being played by a device comes from the audio store of the device of which the display is a part, or from another device.

73. The device of claim 63 wherein the second device provides information to the second user about the strength of the audio signal received by the receiver.

74. The device of claim 73 additionally comprising a visual display, wherein the signal strength information is provided in the visual display

75. The device of claim 73 wherein the signal strength information is provided secondary audio signals played substantially simultaneously with the audio entertainment signal through the player of the device.

76. The device of claim 63 wherein the signal store stores a compressed audio signal.

77. The device of claim 63 wherein the compressed audio signal is transmitted through the transmitter.

78. The device of claim 63 wherein the compressed audio signal is decompressed prior to being transmitted through the transmitter.

79. The device of claim 63 wherein the identifier is interspersed within the transmitted audio signal.

80. The device of claim 63 additionally comprising a delay circuit.

81. The device of claim 80 wherein the playing of the audio entertainment signal is delayed in each device so that the audio entertainment signal in all devices playing from the same stored audio entertainment signal is played substantially simultaneously.

82. The device of claim 63 additionally comprising a manual control with at least three states, wherein in the first state the player plays a stored audio entertainment signal and does not transmit the stored entertainment signal, wherein in the second state the player plays a stored audio entertainment signal and substantially simultaneously transmits the same audio entertainment signal, and wherein in the third state the player plays an audio entertainment signal received by the receiver.

83. The device of claim 63 wherein the ambient sound admitted by the earphone can be manually controlled.

84. The device of claim 63 additionally comprising a microphone wherein the device can transmit vocal sounds from the user of the device.

85. The device of claim 63 additionally comprising a manual control such that actuation transmits a message.

86. The device of claim 63 wherein the device can connect with the telephone network.

87. The device of claim 63 wherein the device is mobile and can be carried by a user.

88. The device of claim 63 additionally comprising a control signal transmitter that transmits control signals that controls a humanly-perceptible display in a wearable accessory.

89. The device of claim 88 wherein the control signal transmitter is the same as the transmitter that transmits stored audio entertainment signals.

90. The device of claim 88, additionally comprising a store of control signals.

91. The device of claim 88, additionally comprising a generator wherein the generator coverts stored audio entertainment signals into control signals.

92. An audio entertainment device comprising:

a wireless transmitter for the transmission of audio entertainment signals;

a wireless receiver for the reception of the transmitted audio entertainment signals from a transmitter of audio entertainment signals;

a first manually-separable connector for electrically connecting with an audio player, wherein the connector allows transfer of audio entertainment signals from the player to the device;

a second connector for connecting with a speaker; and

a control to manually switch between at least three states, wherein in the first state the speaker plays audio entertainment signals from the audio player and the transmitter does not transmit the audio entertainment signals, wherein in the second state the speaker plays audio entertainment signals from the audio player and the transmitter essentially simultaneously transmits the audio entertainment signals, and wherein in the third state the speaker plays audio entertainment signals received by the receiver.

93. The device of claim 92 wherein the first connector connects to the audio player through the audio output jack.

94. The device of claim 92 wherein the speaker is chosen from the group consisting of earphone and headphone.

95. The device of claim 92 wherein the device additionally comprises a control signal generator that converts audio entertainment signals into control signals that control a humanly-perceptible display in a wearable accessory.

96. The device of claim 92 additionally comprising a second transmitter that transmits control signals that control a humanly-perceptible display in a wearable accessory.

97. The device of claim 92 additionally comprising a display that indicates the state of the device.

98. The device of claim 92 wherein the receiver is receptive of the audio entertainment signals transmitted by the transmitter.

99. The device of claim 92 wherein the receiver is operational when the first connector is manually separated from the audio player.

100. A system for the sharing of stored music between a first user and a second user, comprising:

a first device for playing music to the first user, comprising a store of musical signals, a first controller that prepares musical signals from the first store for transmission and playing, a first player that takes musical signals from the first controller and plays the signals for the first user, and a transmitter, that is capable of taking the musical signals from the controller and transmitting the musical signals via wireless broadcast; and

a second device for playing music to the second user, comprising a receiver receptive of the transmissions from the transmitter of the first device, a second controller that prepares musical signals from the receiver for playing, and a second player that takes musical signal from the second controller and plays the signals for the second user;

wherein the first user and the second user hear the musical signals at substantially the same time.

101. The system of claim 100 wherein the store of musical signals comprises compressed digital audio files.

102. The system of claim 100 wherein the compressed digital audio files are in formats taken from the group consisting of mp3, wav, cda, wma, and au.

103. The system of claim 100 wherein the preparation of the musical signals by the first controller comprises decompression of compressed binary files.

104. The system of claim 100 wherein the transmitter transmits via an Internet protocol.

105. The system of claim 104 wherein the Internet protocol has a link level selected from the group consisting of 802.11, Bluetooth, ultrawideband, HiperLAN and WDCT.

LINKPLAY EXHIBIT 1004
Page 2967 of 4822

106. The system of claim 100 wherein the first device is a telephone.

107. The system of claim 100 wherein the first device transmits compressed musical signals and the second controller decompresses the compressed musical signals received by the receiver.

108. The system of claim 100 wherein the transmitter additionally transmits an annunciator signal indicating the presence of a transmitter of musical signals.

109. The system of claim 100 wherein the second device receiver becomes receptive of transmissions from the first device by obtaining an identifier from the first device.

110. The system of claim 109 wherein the first device additionally comprises an associated visual transducer and the second device additionally comprises a visual receiver, wherein the visual transducer outputs a visual identifier signal that is received by the visual receiver.

111. The system of claim 109 wherein the first device additionally comprises a first electrical connector and the second device additionally comprises a second electrical connector, wherein the identifier is communicated via an electrical signal passed from the first electrical connector to the second electrical connector.

112. The system of claim 111 wherein the communication of the identifier signal occurs via an inductive coupling between the first connector and the second connector.

113. The system of claim 109 wherein the second device additionally comprises a microphone, wherein the identifier is communicated in an audio signal that is produced by the first player and which is received by the microphone.

114. The system of claim 109 wherein the identifier is transmitted by the transmitter and received by the receiver.

115. The system of claim 100 wherein the second device receiver becomes receptive of transmissions from the first device by transmitting an identifier to the first device.

116. The system of claim 115 wherein the second device additionally comprises a visual transducer and the first device additionally comprises a visual receiver, wherein the visual transducer outputs a visual identifier signal that is received by the visual receiver.

117. The system of claim 115 wherein the first device additionally comprises a first electrical connector and the second device additionally comprises a second electrical connector, wherein the identifier is communicated via an electrical signal passed from the second electrical connector to the first electrical connector.

118. The system of claim 117 wherein the communication of the identifier signal occurs via an inductive coupling between the first connector and the second connector.

119. The system of claim 115 wherein the first device additionally comprises a microphone, wherein the identifier is communicated in an audio signal that is produced by the second player and which is received by the microphone.

120. The system of claim 115 wherein the identifier is transmitted by the transmitter and received by the receiver.

121. The system of claim 100 wherein the first controller additionally generates control signals related to the musical signals.

122. The system of claim 121 wherein the store additionally comprises stored control signals that are used by the first controller.

123. The system of claim 121 wherein the controller generates control signals from the musical signals.

124. The system of claim 123 wherein the controller performs frequency analysis of the musical signals in order to generate the control signals.

125. The system of claim 121 wherein the first device additionally transmits the control signals to the second device using the first transmitter.

126. The system of claim 125 wherein the control signals and the musical signals transmitted by the first device are interspersed within a single digital stream.

127. The system of claim 125 wherein the control signals and the musical signals are transmitted in two digital streams.

128. The system of claim 121 wherein the first device additionally comprises a second transmitter that transmits control signals.

129. The system of claim 128 wherein the second transmitter uses an IP protocol.

130. The system of claim 129 wherein the Internet protocol uses a link layer selected from the group consisting of 802.11, Bluetooth, and ultrawideband.

131. The system of claim 121 wherein the second transmitter uses analog radio frequency transmissions.

132. The system of claim 100, additionally comprising a wearable accessory, comprising a control signal receiver that is receptive of control signals transmitted by the first device, an accessory controller that generates transducer signals from the received control signals, and a transducer that generates humanly-perceptive music accompaniment signals from the transducer signals.

133. The system of claim 132 wherein the music accompaniment signals are visual signals.

134. The system of claim 133 wherein the visual signals are generated by a light emitting diode

135. The system of claim 132 wherein the music accompaniment signals are tactile signals.

136. The system of claim 132 wherein the accessory controller performs frequency analysis on the received control signals.

137. The system of claim 132 wherein the wearable accessory additionally comprises a switch, wherein actuation of the switch causes the accessory to be receptive to an entraining signal from the first device, such that the accessory will accept signals from the first device.

138. An earphone for listening to audio entertainment allowing for the controlled reception of ambient sound by a user, comprising

a speaker that is oriented towards the user's ear;

an enclosure that reduces the amount of ambient noise perceptive to the user; and

a manually-adjustable characteristic of the enclosure that adjusts the amount of ambient sound perceptive to the user.

139. The earphone of claim 138 wherein the manually-adjustable characteristic is the size of apertures in the enclosure that admit ambient sound.

140. The earphone of claim 138 wherein the manually-adjustable characteristic is the number of apertures in the enclosure that admit ambient sound.

141. The earphone of claim 138 wherein the manually-adjustable characteristic is the surface area of the enclosure.

142. The earphone of claim 138 wherein the manually-adjustable characteristic is the thickness of the enclosure.

143. The earphone of claim 138, additionally comprising a spacer that attaches to the earphone and which increases the distance between the speaker and the user's ear.

144. The earphone of claim 138, additionally comprising a secondary enclosure that encloses the enclosure and the speaker, and which is detachable from the earphone.

145. A mobile device for the transmission of audio entertainment signals comprising

an audio signal store for the storage of the audio entertainment signals;

an audio signal player for the playing of the audio entertainment signals;

LINKPLAY EXHIBIT 1004
Page 2970 of 4822

an wireless transmitter for the transmission of the audio entertainment signals; and

a transmitter control to manually switch between two states consisting of the operation and the non-operation of the audio transmitter.

146. The device of claim 145 wherein the transmitter is connected to the audio jack of the audio signal player.

147. The device of claim 146 wherein the audio signal that is transmitted via the audio jack is subsequently divided between the transmitter and the headphone.

148. The device of claim 146 wherein the transmitter additionally comprises a second jack to which the earphone plug can connect, such that the insertion of the earphone plug into the second jack allows the audio signal to be heard in the earphone

149. A mobile device for the reception of digital audio entertainment signals comprising:

an audio signal store for the storage of the digital audio entertainment signals;

an audio receiver for the reception of external digital audio entertainment signals from a mobile audio signal transmitter located within a predetermined distance of the audio receiver;

a receiver control with at least a first state and a second state; and

an audio signal player that plays digital audio entertainment signals from the audio signal store when the receiver control is in the first state, and that plays digital audio entertainment signals from the audio receiver when the receiver control is in the second state.

150. The device of claim 149, additionally comprising a jack to which a listening device chosen from the group consisting of earphone and headphone can be connected.

151. A method for the shared enjoyment of music from stored musical signals between a first user with a first music player device and at least one second user with at least one second music player device, comprising:

playing the musical signals for the first user on the first music player device while essentially simultaneously wirelessly transmitting synchronization signals from the first music player device to the at least one second music player device; and

receiving the synchronization signals by the at least one second player device, wherein the synchronization signals allow the musical signals on the at least one second player device to be played essentially simultaneously with the playing of the musical signals on the first music player device;

wherein the first and the at least one second users are mobile.

LINKPLAY EXHIBIT 1004
Page 2971 of 4822

152. The method of claim 151 wherein the same musical signals are stored on the first music player device and the at least one second music player device.

153. The method of claim 152 wherein the musical signals are stored as compressed audio files.

154. The method of claim 151 wherein the synchronization signals comprise the current sample number being played on the first music player device.

155. The method of claim 154 wherein the synchronization signals further comprise the sample rate of the musical signal being played on the first music player device.

156. The method of claim 151 wherein the synchronization signals comprise the current time relative to the beginning of the musical signal being played on the first music player device.

157. The method of claim 151, additionally comprising transferring the musical signals from the first music player device to the at least one second music player device.

158. A wireless communications system for sharing audio entertainment between a first mobile device and a second mobile device, comprising:

a broadcast identifier transmitted by the first mobile device to the second mobile device;

a personal identifier transmitted by the second mobile device to the first mobile device; and

a broadcast signal comprising audio entertainment that is transmitted by the first mobile device of which the second device is receptive;

wherein the first mobile device and the second mobile device have displays which can display the identifier signal that they receive and the second mobile device can play the audio entertainment from the broadcast signal that it receives.

159. The system of claim 158 wherein the identifier signals are transmitted via an Internet protocol.

160. The system of claim 159 wherein the Internet protocol is a connection-oriented transport protocol.

161. The system of claim 159 wherein the Internet protocol further comprises a link layer protocol selected from the group consisting of 802.11, Bluetooth, ultrawideband, HiperLAN and WDCT.

162. The system of claim 158 wherein the broadcast signal is transmitted via an Internet protocol.

LINKPLAY EXHIBIT 1004
Page 2972 of 4822

163. The system of claim 162 wherein the Internet protocol is a connectionless transport protocol.

164. The system of claim 162 wherein the Internet protocol further comprises a link layer protocol selected from the group consisting of 802.11, Bluetooth, ultrawideband, HiperLAN and WDCT.

165. The system of claim 158 wherein the broadcast signal is an analog radio frequency signal.

166. The system of claim 158 wherein the broadcast signal is transmitted via telephony.

167. The system of claim 158 wherein the broadcast signal is compressed.

168. The system of claim 158 wherein the broadcast signal is encrypted.

169. The system of claim 158, additionally comprising a transfer signal that is transmitted by the first mobile device upon halting the transmission of its broadcast signal, and which transfer signal is receptive by the second mobile device, such that reception by the second mobile device causes the device to transmit a second broadcast signal comprising audio entertainment for which the first unit is receptive.

170. The system of claim 158 wherein the broadcast signal is transmitted via peer-to-peer communications.

171. The system of claim 170 additionally comprising a switch, wherein actuation of the switch causes the device to toggle between peer-to-peer communications and infrastructure-based communications.

172. The system of claim 158 wherein the broadcast signal is transmitted via infrastructure-based communications.

173. The system of claim 158 wherein the personal identifier includes a least one piece of information about the user of the second mobile device taken from the group consisting of the name of the user, the nickname of the user, a numerical identifier, an alphanumerical identifier, the musical preferences of the user, the age of the user, and the educational institution that the user attends.

174. The system of claim 158 additionally comprising a third mobile device whose user requests to receive the broadcast signal and additionally comprising a requester identifier signal transmitted by the third mobile device to the first mobile device.

175. The system of claim 174 wherein the requester identifier signal is additionally transmitted by the third mobile device to the second mobile device.

176. The system of claim 174 wherein the requester identifier signal is additionally transmitted by the first mobile device to the second mobile device.

177. The system of claim 174 wherein the requester identifier is displayed on the display of the first mobile device.

178. The system of claim 174 wherein the third mobile device receives the broadcast signal without the affirmative action of the first mobile device.

179. The system of claim 174 wherein the third mobile device must receive the affirmative action of the first mobile device in order to receive the broadcast signal.

180. The system of claim 174 wherein the third mobile device must receive the affirmative action of the first mobile device and all other mobile devices receptive of the broadcast signal.

181. The system of claim 180 wherein the affirmative action is a unanimous vote.

182. The system of claim 180 wherein the affirmative action is a majority vote.

183. The system of claim 158, additionally comprising a substantially unique musical identifier related to the audio entertainment, wherein the musical identifier is transmitted by the first mobile device to the second mobile device in association with the broadcast signal, and the musical identifier is stored on the second mobile device in association with the broadcast identifier.

184. A system for identifying a first device that introduces a music selection to a second device, comprising:

a mobile music transmitter operated by the first device;

a mobile music receiver operated by the second device;

a music signal comprising the music selection transmitted by the transmitter and received by the receiver;

an individual musical identifier that is associated with the music selection; and

an individual transmitter identifier that identifies the transmitter, wherein the transmitter identifier and the individual music identifier are stored in association with each other in the receiver.

185. A method for enhancing enjoyment of a musical selection, comprising:

obtaining control signals related to the musical selection;

transmitting the control signals wirelessly;

receiving the control signals; and

converting the control signals to a humanly-perceptible form.

LINKPLAY EXHIBIT 1004
Page 2974 of 4822

186. The method of claim 185 wherein the distance over which the control signals are transmitted has a minimum range of at least 1 foot.

187. The method of claim 185 wherein the distance over which the control signals are transmitted has a maximum range of less than 500 feet.

188. The method of claim 185 wherein the humanly-perceptible form is a visually-perceptible form.

189. The method of claim 185 wherein the humanly-perceptible form is a tactilely-perceptible form.

190. The method of claim 185 wherein the transmitting and receiving use radio frequency communications.

191. The method of claim 185 wherein the radio frequency communications use an Internet protocol.

192. The method of claim 185 wherein the Internet protocol involves a link layer chosen from the group consisting of 802.11, Bluetooth, ultrawideband, HiperLAN and WDCT.

193. The method of claim 185 wherein obtaining additionally comprises extracting the musical beat from the musical selection.

194. The method of claim 185 wherein obtaining additionally comprises retrieving the control signals from a store of control signals.

195. The method of claim 185 wherein the control signals are divided into multiple channels.

196. The method of claim 195 wherein the step of converting comprises a step of demultiplexing the control signals.

197. The method of claim 185 wherein the steps of transmitting and receiving are performed using means conveyed by a single person.

198. The method of claim 197 wherein the user is mobile.

199. The method of claim 185 wherein the step of transmitting is performed by a stationary user and the step of receiving is performed by a mobile user.

200. The method of claim 185 wherein the musical selection is provided by musicians in concert playing essentially simultaneously with obtaining control signals.

201. The method of claim 185 wherein the control signals are transmitted with a substantially unique identifier.

202. The method of claim 185 wherein the control signals control the appearance of a wearable device.

LINKPLAY EXHIBIT 1004
Page 2975 of 4822

203. A method for generating and storing control signals corresponding to musical signals, comprising:

playing musical signals for a user;

receiving manual input signals from the user that are produced substantially in synchrony with the music;

generating control signals from the input signals; and

storing the control signals so that they can be retrieved with the musical signals.

204. The method of claim 203 wherein receiving comprises transducing tactile input from the user.

205. The method of claim 204 wherein the tactile input is produced using a user appendage selected from the group consisting of hand, finger and foot.

206. The method of claim 203 wherein receiving comprises transducing sound output produced by the user.

207. The method of claim 203 wherein storing comprises interleaving control signals within the musical signals.

208. The method of claim 203 wherein storing additionally comprises creating a digital file.

209. The method of claim 203 wherein storing additionally comprises associating the digital file with the musical signals.

210. A wearable personal accessory, comprising:

an input transducer taken from the group consisting of a microphone and an accelerometer, said transducer generating a time-varying input transduction signal;

a controller which accepts the input transduction signal, and which generates an output transducer signal whose signal varies in amplitude with time;

an output transducer receptive of the output transducer signal, and which provides a humanly-perceptible signal; and

an energy source for powering the input transducer, controller and output transducer.

211. The accessory of claim 210 wherein the accelerometer responds to the movements of the user in one axis of movement.

212. The accessory of claim 210 wherein the accelerometer responds to the movements of the user in more than one axis of movement.

213. The accessory of claim 210 wherein the controller performs automatic gain control on the signal from the input transducer.

LINKPLAY EXHIBIT 1004
Page 2976 of 4822

214. The accessory of claim 210 wherein the output transducer is a light transducer.

215. The accessory of claim 214 wherein the output transducer comprises a light emitting diode.

216. The accessory of claim 214 wherein the output transducer comprises a liquid crystal display.

217. The accessory of claim 214 wherein the output transducer comprises an organic light emitting diode.

218. The accessory of claim 210 wherein the output transducer is a tactile transducer.

219. The accessory of claim 210 wherein the output transducer comprises a first and a second transduction element.

220. The accessory of claim 219 wherein the first transduction element responds preferentially to a first band of frequencies of the input transduction signal and the second transduction element responds preferentially to a second band of frequencies of the input transduction signal.

221. The accessory of claim 219 wherein the first transduction element responds preferentially to a first band of amplitudes of the input transduction signal and the second transduction element responds preferentially to a second band of amplitudes of the input transduction signal.

222. A wearable personal accessory controlled via wireless communications, comprising:

a wireless communications receiver, receptive of an external control signal;

a controller that accepts the external control signal and that generates a time-varying visual output transducer signal;

a visual output transducer that is receptive of the output transducer signal, and which provides a humanly-perceptible visual signal; and

an energy store for powering the receiver, controller and output transducer;

wherein the visual output transducer generates visually-perceptive output.

223. The accessory of claim 222 wherein the external control signal is a time-varying amplitude signal.

224. The accessory of claim 223 wherein the amplitude signal is analog.

225. The accessory of claim 223 wherein the amplitude signal is digital.

226. The accessory of claim 223 wherein the amplitude signal is substantially a musical signal.

LINKPLAY EXHIBIT 1004
Page 2977 of 4822

227. The accessory of claim 226 wherein the controller performs frequency filtering of the external control signal.

228. The accessory of claim 226 wherein the controller determines the primary beat of the music.

229. The accessory of claim 228 wherein the primary beat of the music is determined according to the positions in the music where the amplitude rises at least a predetermined factor over less than a predetermined time.

230. The accessory of claim 222 wherein the external control signal is a digital signal of pairs comprising positions within the music and the amplitudes to which the output transducers are set at such times.

231. The accessory of claim 230 wherein the positions are chosen from the group of characteristics consisting of samples and times.

232. The accessory of claim 222 wherein the receiver is an analog radio frequency receiver.

233. The accessory of claim 232 wherein the analog radio frequency communications are at a frequency set by a crystal oscillator, wherein the crystal oscillator can be manually exchanged in order to change the frequency of operation.

234. The accessory of claim 222 wherein the receiver receives Internet protocol signals.

235. The accessory of claim 234 wherein the Internet protocol further comprises a link layer selected from the group consisting of Bluetooth, 802.11 and ultrawideband.

236. The accessory of claim 222 wherein the visual output transducer comprises a first light transducer and a second light transducer.

237. The accessory of claim 236 wherein the external control signal comprises a multiplexed first signal and second signal, and wherein the first signal relates to the first light transducer and the second signal relates to a second light transducer.

238. The accessory of claim 237 wherein the controller comprises at least two ports, wherein the first port relates to the first light generating element and the second port related to the second light generating element.

239. The accessory of claim 236 wherein the first light transducer responds preferentially to a first range of frequency in the musical signal, and the second light transducer responds preferentially to a second range of frequency in the musical signal.

LINKPLAY EXHIBIT 1004
Page 2978 of 4822

240. The accessory of claim 236 wherein the first light transducer responds preferentially to a first range of amplitude in the musical signal, and the second light transducer responds preferentially to a second range of amplitude in the musical signal.

241. The accessory of claim 222 wherein the visual output transducer comprises a light emitting diode.

242. The accessory of claim 222 wherein the visual output transducer comprises an organic light emitting diode.

243. The accessory of claim 222 wherein the visual output transducer comprises a liquid crystal display.

244. The accessory of claim 222 additionally comprising a tactile transducer responsive to the external control signal.

245. The accessory of claim 222 additionally comprising a sound transducer responsive to the external control signal.

246. The accessory of claim 222 wherein the external control signal additional comprises a digital identifier.

247. The accessory of claim 246 wherein the visual output transducer additionally provides an identifier signal that corresponds to the digital identifier, wherein such identifier signal is higher frequency than is humanly perceptible, and wherein the identifier signal is output substantially simultaneously with the humanly-perceptible visual signal.

248. The accessory of claim 222 additionally comprising a manually-separable electrical connection, wherein the connection connects the receiver to the controller.

249. The accessory of claim 222 additionally comprising a manually-separable electrical connection, wherein the connection connects the controller to the output transducer.

250. The accessory of claim 222 wherein the accessory functions as an accoutrement selected from the group consisting of a ring, a pendant, a hair clip, an earring, a necklace, a brooch, a belt buckle, watches, and a bracelet.

251. The accessory of claim 222 wherein the accessory is incorporated into an article of clothing selected from the group consisting of a backpack, a wallet, a purse, a hat and a shoe.

252. The accessory of claim 222 additionally comprising a manual control whose operation sets either a first state and a second state and wherein in the first state the accessory responds exclusively to external control signals comprising a substantially unique identifier.

LINKPLAY EXHIBIT 1004
Page 2979 of 4822

253. The accessory of claim 252 wherein the identifier to which the accessory responds in the first state is received by the receiver when the accessory is set to the second state.

254. The accessory of claim 253 additionally comprising a third state which is manually set, wherein in the third state, the accessory responds to external control signals in the absence of a substantially unique identifier.

255. A device for converting user tactile responses to stored music into a stored control signal, comprising:

a player that plays stored music audible to the user;

a manually-operated transducer that outputs an electrical signal, wherein the transducer is actuated by the user in response to the music;

a controller that receives the electrical signal and outputs a control signal; and

a store that receives the control signal and stores it.

256. The device of claim 255 wherein the control signal comprises a list of positions, wherein the positions are chosen from the set of characteristics consisting of samples and time relative to the beginning of the stored music.

257. The device of claim 255 wherein the control signal further comprises a list of amplitudes corresponding to the positions.

258. The device of claim 257 additionally comprising a second manually-operated transducer that outputs a second electrical signal, wherein the controller additionally receives the second electrical signal.

259. The device of claim 255 wherein the transducer comprises a musical drum operated by the user whose audible output is received by a microphone that generates the electrical signal received by the controller.

260. The device of claim 255 wherein the controller performs frequency filtering on the electrical signal.

261. The device of claim 255 wherein the transducer is a tactile transducer that is operated by an appendage of the user selected from the group consisting of finger, hand, and foot.

262. The device of claim 255 wherein the store is a file system and the control signal is stored as control file and the stored music is stored as a music file, wherein the control file and the music file are stored in association with each other.

263. The device of claim 255 wherein the control signals are stored in an interleaved fashion with the stored music.

LINKPLAY EXHIBIT 1004
Page 2980 of 4822

264. The device of claim 71 additionally comprising a visual display on which a visual representation of the control signals is displayed.

265. A music player that wirelessly transmits control signals related to the music, wherein the control signals control a wearable electronic accessory, comprising:

a store of music signal files;

a controller that reads a musical signal file from the store and generates audio signals and that further generates the control signals;

a transducer that converts the audio signals into sound audible to the user; and

a wireless transmitter that transmits the control signal to the wearable electronic accessory.

266. The player of claim 265 wherein the music signal files comprise compressed audio files.

267. The player of claim 265 wherein the controller performs frequency filtering of the external control signal.

268. The player of claim 265 wherein the controller determines the primary beat of the music.

269. The player of claim 268 wherein the primary beat of the music is determined according to the positions in the music where the amplitude rises at least a predetermined factor over less than a predetermined time.

270. The player of claim 265 wherein the control signals are pairs comprising positions within the music and the amplitudes to which the output transducers are set at such times.

271. The player of claim 265 wherein the positions are chosen from the set of characteristics consisting of samples and times.

272. The player of claim 265 wherein the transmitter is an analog radio frequency transmitter.

273. The player of claim 265 wherein the transmitter transmits according to an Internet protocol.

274. The player of claim 273 wherein the Internet protocol has a link layer selected from the group consisting of 802.11, Bluetooth, ultrawideband, HiperLAN and WDCT.

275. The player of claim 265 additionally comprising a visual transducer that generates visually-preceptive output in response to the control signal, wherein the visual transducer receives the control signal from the controller.

276. The player of claim 275 wherein the visual transducer is a light emitting diode.

LINKPLAY EXHIBIT 1004
Page 2981 of 4822

277. The player of claim 275 wherein the visual transducer is a liquid crystal display.

278. The player of claim 265 additionally comprising a store of configuration files wherein the configuration files include parameters read by the controller that affect the generation of control signals from the musical signals.

279. A music player that wirelessly transmits control signals related to the music, wherein the control signals control a wearable electronic accessory, comprising:

a store of music signal files;

a second store of control signal files associated with the music signal files;

a controller that reads a musical signal file from the store and generates audio signals and that further reads an associated control signal file;

a transducer that converts the audio signals into sound audible to the user, and

a wireless transmitter that transmits the control signals from the associated control signal file to the wearable electronic accessory.

280. The player of claim 279 wherein the music signal files comprise compressed audio files.

281. The player of claim 279 wherein the transmitter is an analog radio frequency transmitter.

282. The player of claim 279 wherein the transmitter transmits according to an Internet protocol.

283. The player of claim 282 wherein the Internet protocol has a link layer selected from the group consisting of 802.11, Bluetooth, ultrawideband, HiperLAN and WDCT.

284. The player of claim 279 wherein the store of control information comprises at least one binary data file that relates to a single music signal file.

285. The player of claim 279 wherein the store of control information is interleaved with a related music signal file in binary data format.

286. The player of claim 279 additionally comprising a visual transducer that generates visually-preceptive output in response to the electrical control signal, wherein the visual transducer receives the electrical control signal from the controller.

287. A system for exhibition of music enjoyment, comprising:

a source of music signals;

a controller that generates control signals from the music signals;

a transmitter of the control signals, wherein the transmission of the control signals is synchronized with the playing of the music signals;

LINKPLAY EXHIBIT 1004
Page 2982 of 4822

a receiver of the control signals; and

a transducer that responds to the control signals.

288. The system of claim 287 wherein the source of music signals are stored binary music files.

289. The system of claim 288 wherein the stored binary data files are compressed audio files.

290. The system of claim 288 wherein the music signals are stored on a medium selected from the group consisting of compact disk, digital video disk, hard disk drive, removable disks, flash memory, magneto-optical disk, and audio tape.

291. The system of claim 287 wherein the source of music signals are musicians playing live music.

292. The system of claim 287 wherein the receiver and transducer are worn by a user.

293. The system of claim 287 wherein the receiver and the transducer are physically separated from the transmitter by more than 5 feet.

294. The system of claim 287 wherein the transducer is a visual transducer that generates humanly-perceptible visual signals.

295. The system of claim 294 wherein the visual transducer is a light emitting diode

296. The system of claim 294 wherein the visual transducer is a liquid crystal display

297. The system of claim 287 wherein the transducer is a tactile transducer.

298. The system of claim 287 wherein the minimum range of the transmitter for which the receiver is receptive is at least 1 foot.

299. The system of claim 287 wherein the maximum range of the transmitter for which the receiver is receptive is less than 500 feet.

300. The system of claim 287 wherein the controller determines the primary beat of the music.

301. The system of claim 287 wherein the controller performs frequency filtering.

302. The system of claim 287, additionally comprising a second controller that is electrically connected to the receiver, and which generates transducer control signals from the control signals received by the receiver, wherein the transducer control signals directly control the transducer.

303. The system of claim 302 wherein the control signals are time-varying amplitude signals and wherein the second controller performs frequency filtering.

304. The system of claim 302 additionally comprising a switch that is electrically connected to the second controller such that actuation of the switch places the second controller into entraining mode, wherein the receiver becomes receptive of entraining signals from the transmitter.

305. The system of claim 304 wherein a receiver that is entrained will receive control signals other than from the transmitter in the absence of signals from the transmitter.

306. The system of claim 304 wherein additional actuation of the switch will allow reception of control signals by the receiver from a transmitter to which it is not entrained.

307. The system of claim 287 wherein the transmitter and receiver communicate via an Internet protocol.

308. The system of claim 287 wherein the transmitter and receiver communicate via digital radio frequency communications.

309. The system of claim 287 wherein the transmitter and receiver communicate via analog radio frequency communications.

310. The system of claim 309 wherein the analog radio frequency communications are at a frequency set by a crystal oscillator and wherein the crystal oscillator can be manually replaced.

311. The system of claim 287 additionally comprising a transferror for the transmission of music signals.

312. The system of claim 311 wherein the transmitter and the transferror are the same.

313. The system of claim 311 wherein the transmitter and the transferror are different, and communicate via different communications protocols.

314. The system of claim 287 additionally comprising a second receiver for the reception of music signals, wherein the music signals that are received are played by the music player.

315. A method for transferring a wearable-accessory control file stored on a first device to a second device in which an associated music file is stored, comprising:

storing on the first device the name of the music file in conjunction with the control file with which it is associated;

requesting by the second device of the first device for a control file stored in conjunction with the name of the music file; and

transferring the control file from the first device to the second device, wherein the control file is stored on the second device in conjunction with the name of the associated music file.

316. The method of claim 315 additionally comprising the step of extracting, wherein the control file is stored on the first device interspersed with the music file as a combined file, and the control file is extracted from the music file prior to being transferred to the second device.

317. The method of claim 315 wherein the requesting is performed over a wide area network.

318. The method of claim 315 wherein the name of the control file comprises the root of the name of the associated music file appended with a predetermined extension that is different from that of the music file.

319. The method of claim 315 wherein the control file and the associated music file both comprise the same substantially unique identifier.

320. A device for transmitting control signals to a wearable accessory receptive of such control signals, comprising:

a manually-separable input connector for connecting to an output port of an audio player, wherein audio signals are conveyed from the audio player to the device across the connector;

a controller for generating control signals from the audio signals; and

a transmitter for transmitting the control signals.

321. The device of claim 320 additionally comprising a second connector, wherein a portion of the audio signal conveyed to the device is conveyed to the second connector, and wherein an earphone can connect to the second connector.

322. The player of claim 320 wherein the controller performs frequency filtering of the audio signal.

323. The player of claim 320 wherein the controller determines the primary beat of the audio signal.

324. The player of claim 320 additionally comprising a visual transducer that generates visually-preceptive output in response to the control signals, wherein the visual transducer receives the control signals from the controller.

325. The system of claim 320 wherein the transmitter transmits via an Internet protocol.

LINKPLAY EXHIBIT 1004
Page 2985 of 4822

326. The system of claim 320 wherein the transmitter transmits via digital radio frequency communications.

327. The system of claim 320 wherein the transmitter transmits via analog radio frequency communications.

328. The system of claim 327 wherein the analog radio frequency communications are at a frequency set by a crystal oscillator and wherein the crystal oscillator can be manually replaced.

329. The system of claim 320 wherein the transmitter additionally transmits the audio signal to wireless earphones.

Fig. 1

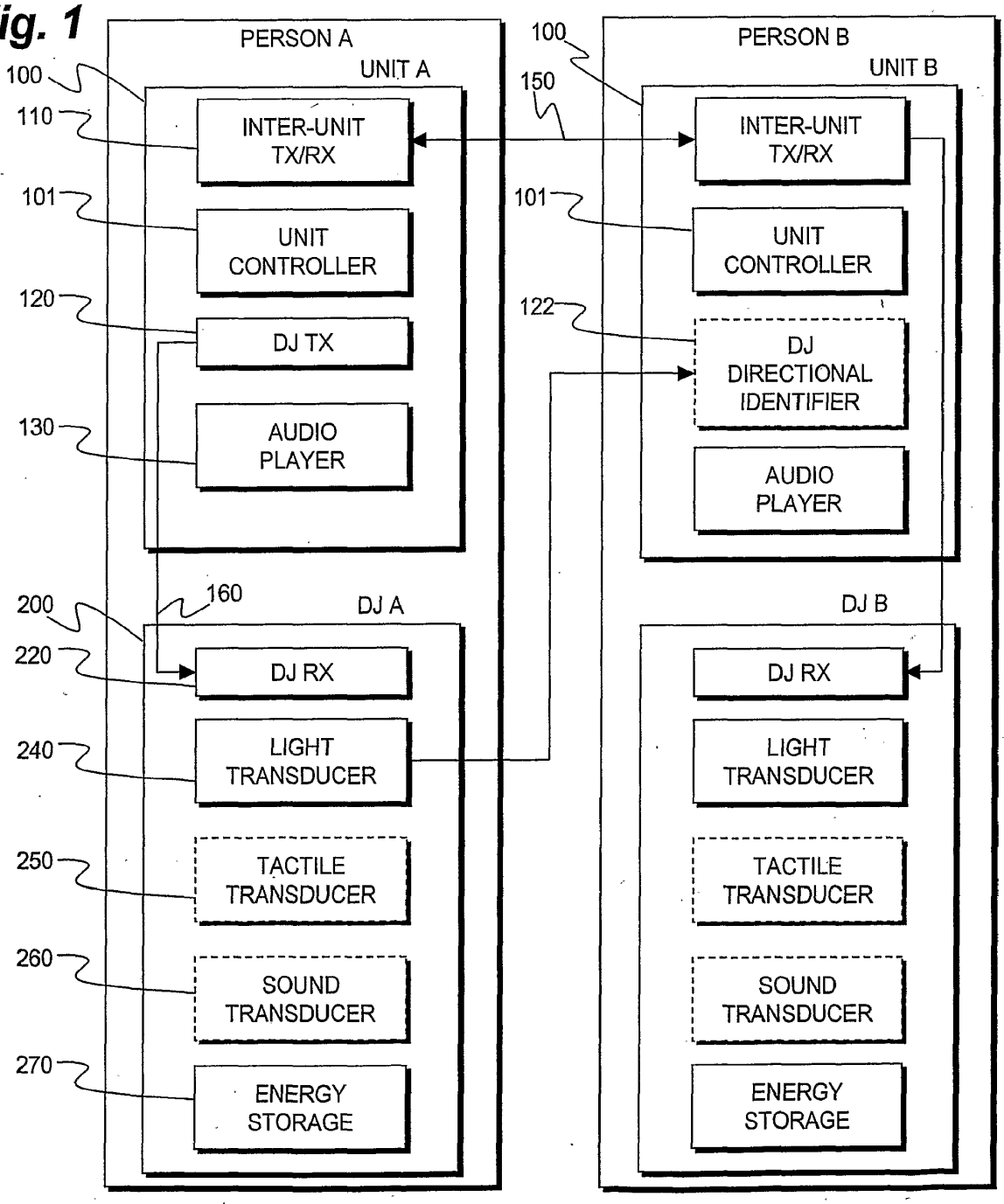

PERSON A
UNIT A
100
110
INTER-UNIT TX/RX
101
UNIT CONTROLLER
120
DJ TX
130
AUDIO PLAYER
160
200
DJ A
220
DJ RX
240
LIGHT TRANSDUCER
250
TACTILE TRANSDUCER
260
SOUND TRANSDUCER
270
ENERGY STORAGE
150
PERSON B
UNIT B
100
INTER-UNIT TX/RX
101
UNIT CONTROLLER
122
DJ DIRECTIONAL IDENTIFIER
AUDIO PLAYER
DJ B
DJ RX
LIGHT TRANSDUCER
TACTILE TRANSDUCER
SOUND TRANSDUCER
ENERGY STORAGE

Fig. 2A

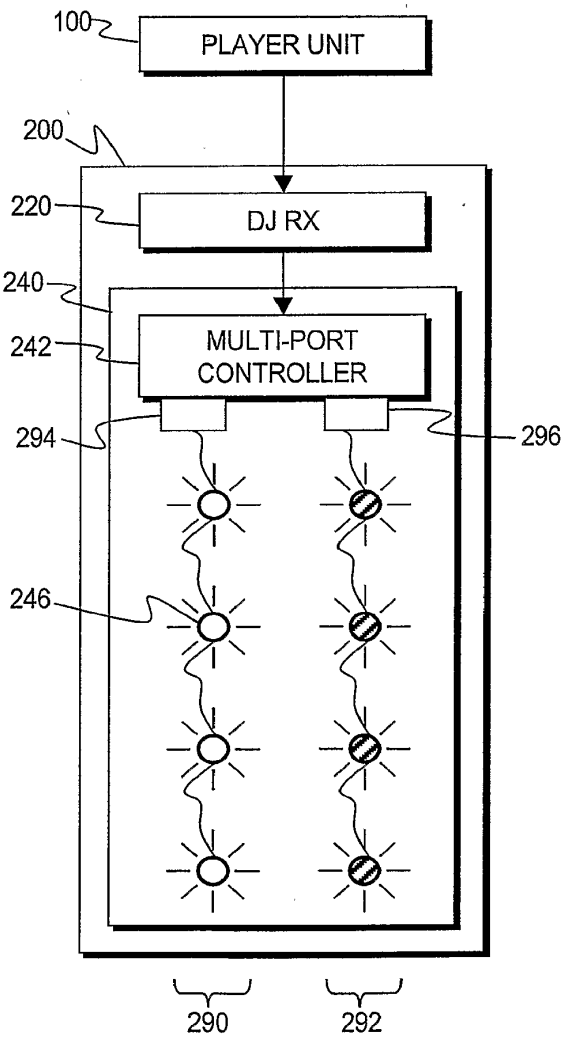

100
PLAYER UNIT
200
220
DJ RX
240
242
MULTI-PORT CONTROLLER
294
296
246
290
292

Fig. 2B

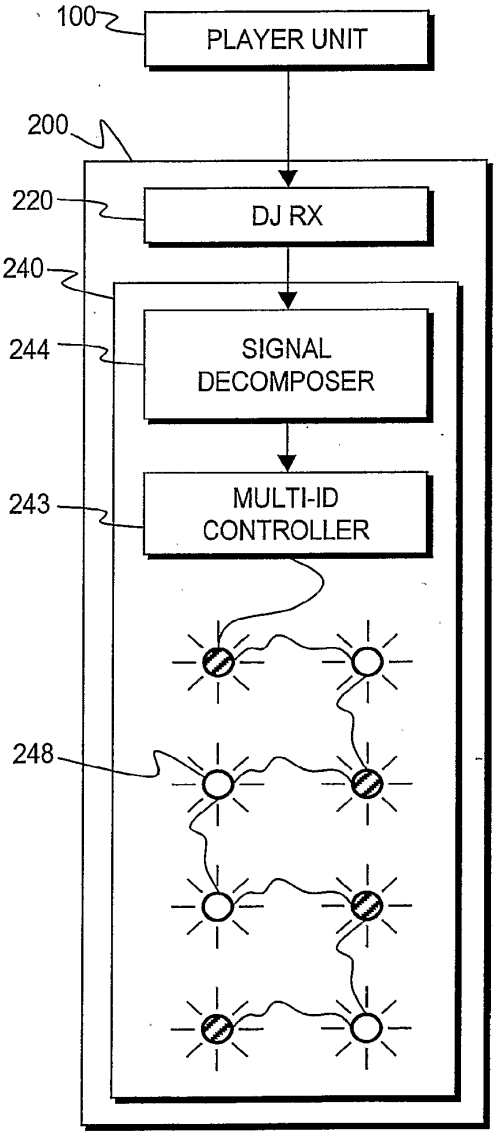

100
PLAYER UNIT
200
220
DJ RX
240
244
SIGNAL DECOMPOSER
243
MULTI-ID CONTROLLER
248

Fig. 3A

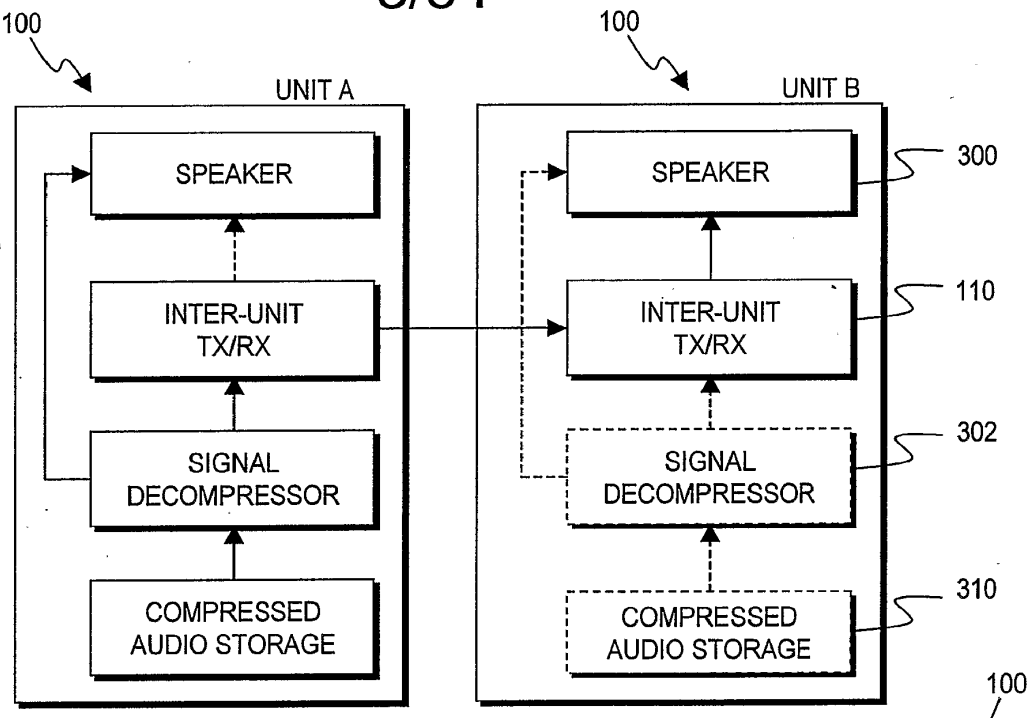

100
UNIT A
SPEAKER
INTER-UNIT TX/RX
SIGNAL DECOMPRESSOR
COMPRESSED AUDIO STORAGE
100
UNIT B
SPEAKER
INTER-UNIT TX/RX
SIGNAL DECOMPRESSOR
COMPRESSED AUDIO STORAGE
300
110
302
310

Fig. 3B

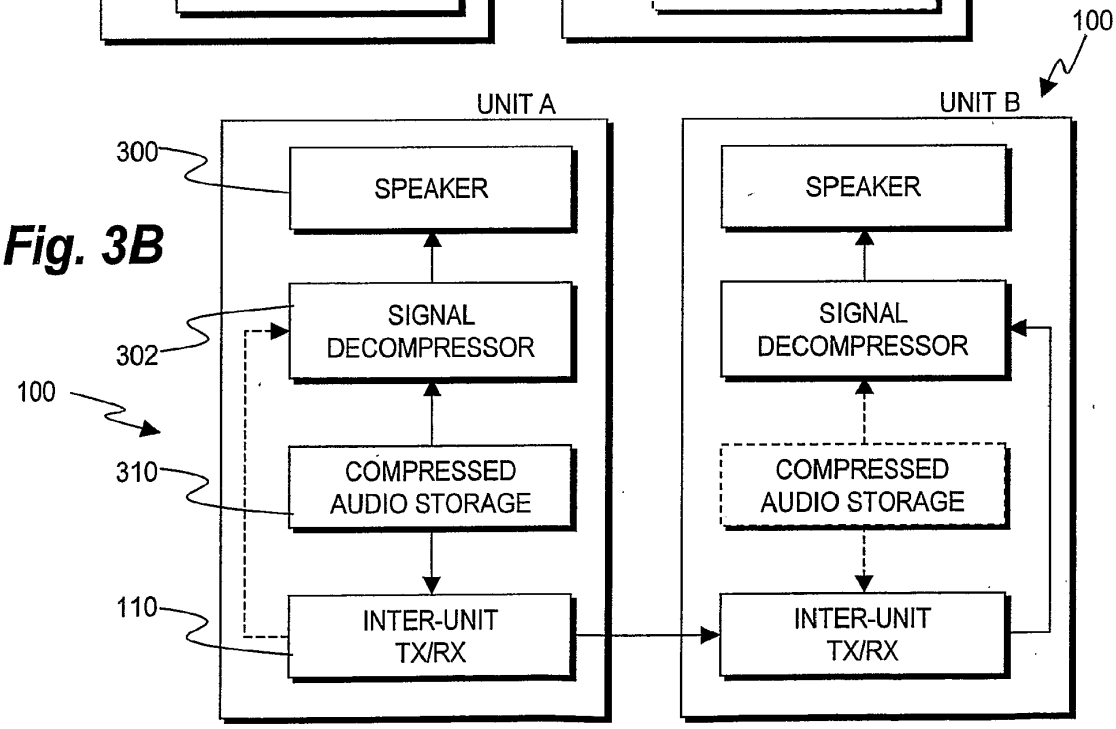

100
UNIT A
UNIT B
300
SPEAKER
SIGNAL DECOMPRESSOR
302
COMPRESSED AUDIO STORAGE
310
INTER-UNIT TX/RX
110

Fig. 3C

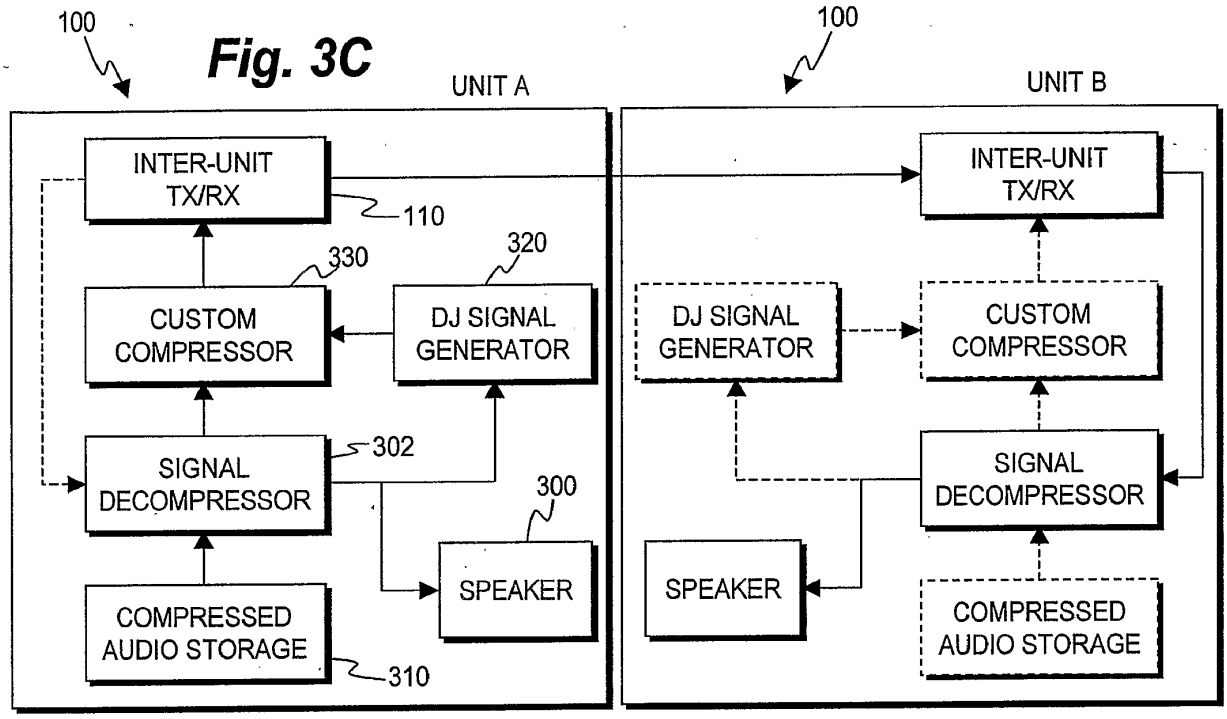

100
UNIT A
INTER-UNIT TX/RX
110
330
320
CUSTOM COMPRESSOR
DJ SIGNAL GENERATOR
SIGNAL DECOMPRESSOR
302
300
SPEAKER
COMPRESSED AUDIO STORAGE
310
100
UNIT B
INTER-UNIT TX/RX
DJ SIGNAL GENERATOR
CUSTOM COMPRESSOR
SIGNAL DECOMPRESSOR
SPEAKER
COMPRESSED AUDIO STORAGE

Fig. 4

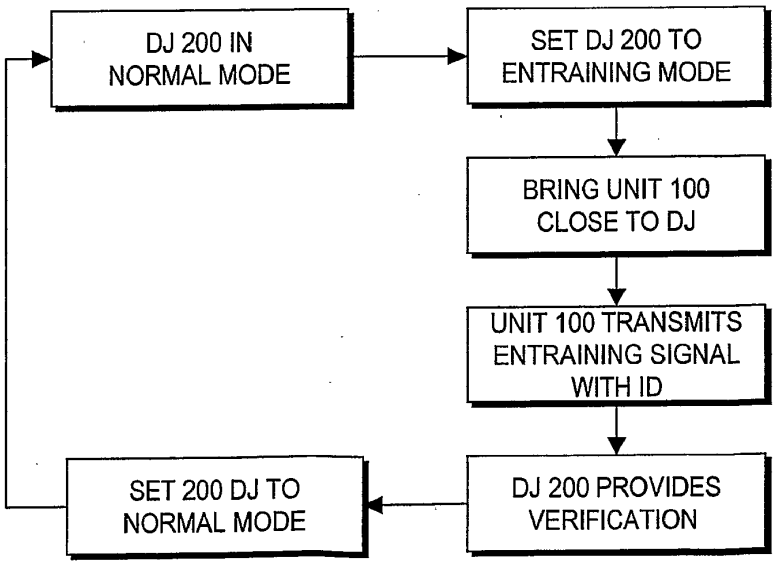

DJ 200 IN NORMAL MODE
SET DJ 200 TO ENTRAINING MODE
BRING UNIT 100 CLOSE TO DJ
UNIT 100 TRANSMITS ENTRAINING SIGNAL WITH ID
DJ 200 PROVIDES VERIFICATION
SET 200 DJ TO NORMAL MODE

Fig. 5A

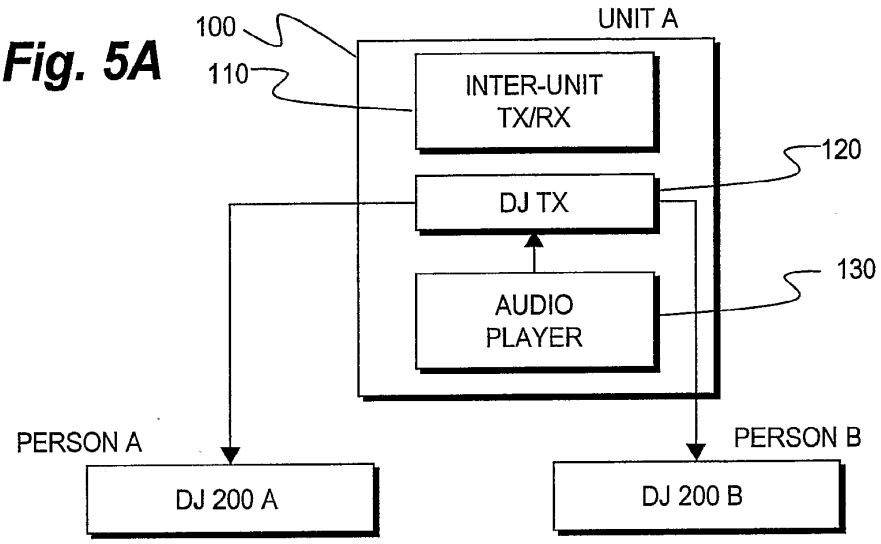

100
UNIT A
110
INTER-UNIT TX/RX
120
DJ TX
130
AUDIO PLAYER
PERSON A
PERSON B
DJ 200 A
DJ 200 B

Fig. 5B

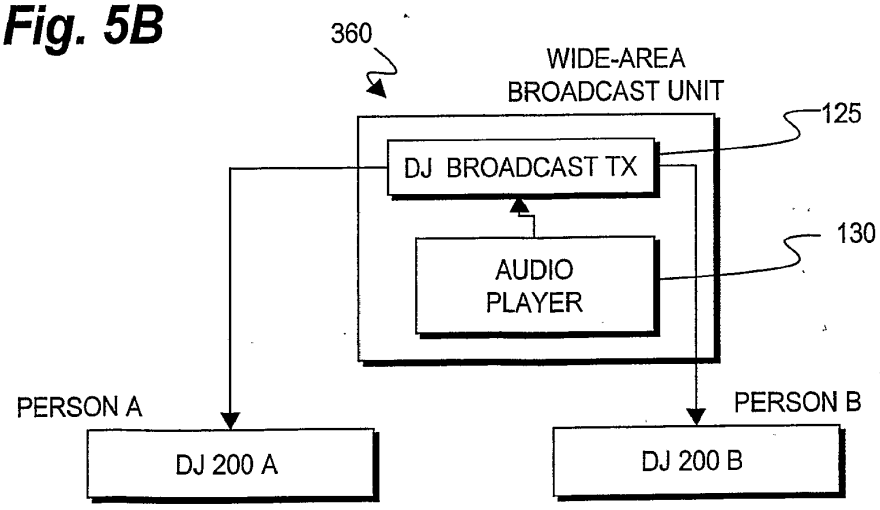

360
WIDE-AREA BROADCAST UNIT
125
DJ BROADCAST TX
130
AUDIO PLAYER
PERSON A
PERSON B
DJ 200 A
DJ 200 B

Fig. 6

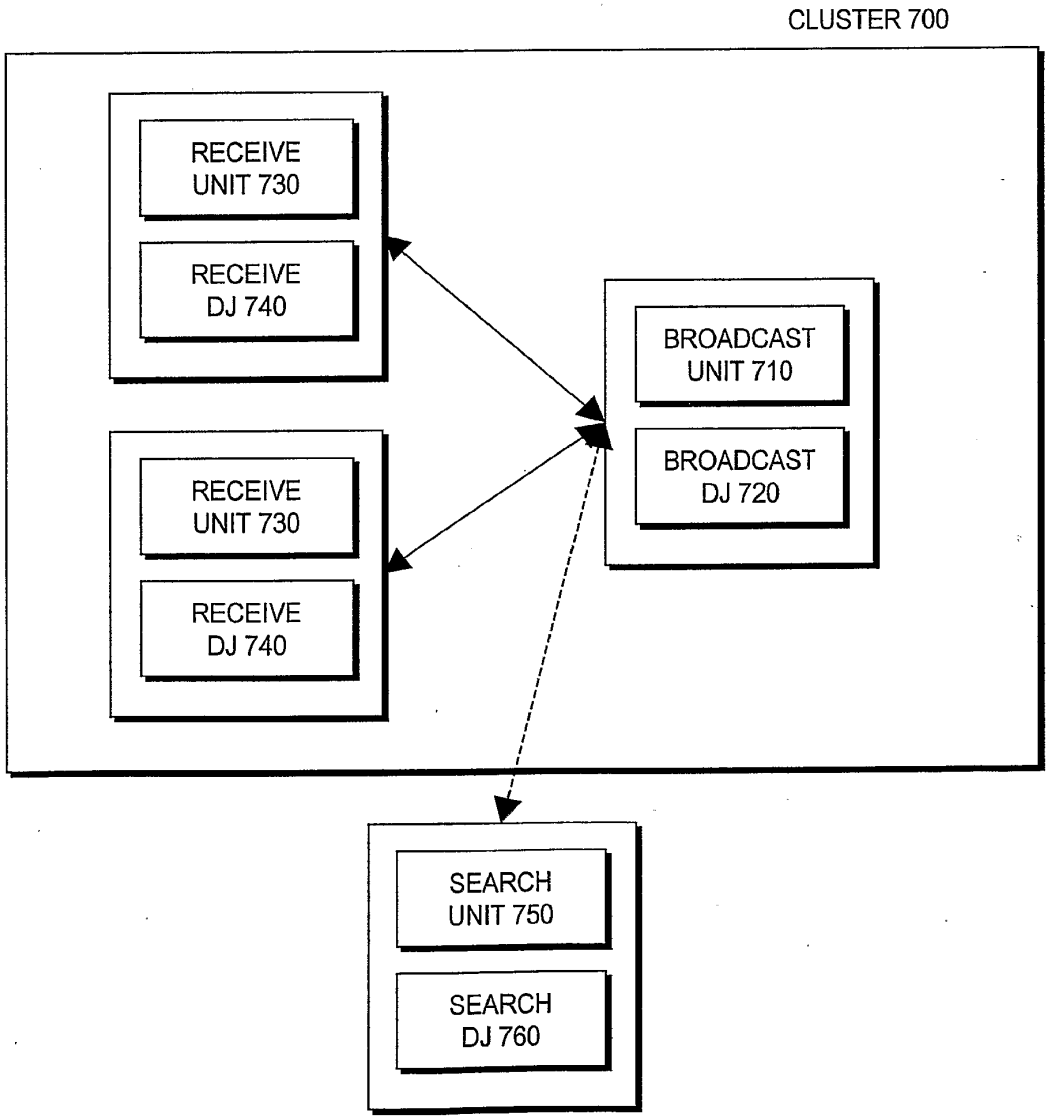

CLUSTER 700
RECEIVE UNIT 730
RECEIVE DJ 740
RECEIVE UNIT 730
RECEIVE DJ 740
BROADCAST UNIT 710
BROADCAST DJ 720
SEARCH UNIT 750
SEARCH DJ 760

Fig. 7

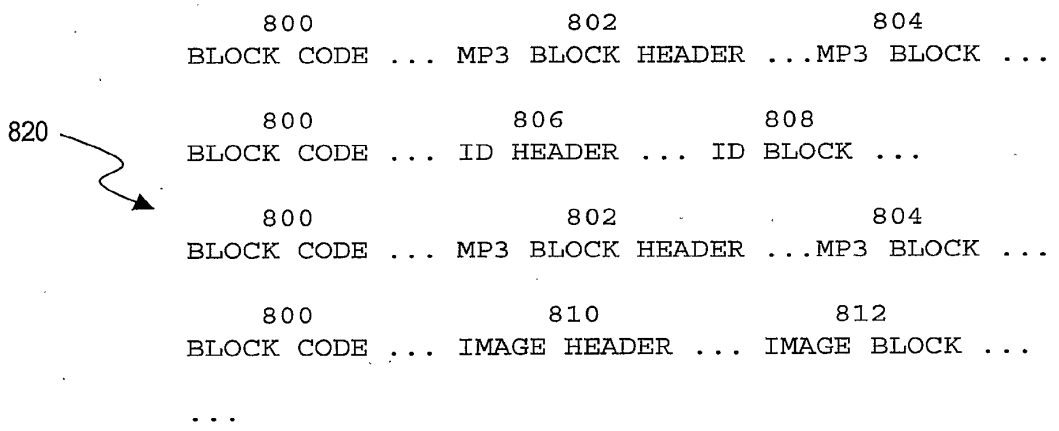

800
802
804
BLOCK CODE ... MP3 BLOCK HEADER ...MP3 BLOCK ...
820
800
806
808
BLOCK CODE ... ID HEADER ... ID BLOCK ...
800
802
804
BLOCK CODE ... MP3 BLOCK HEADER ...MP3 BLOCK ...
800
810
812
BLOCK CODE ... IMAGE HEADER ... IMAGE BLOCK ...
...

Fig. 36

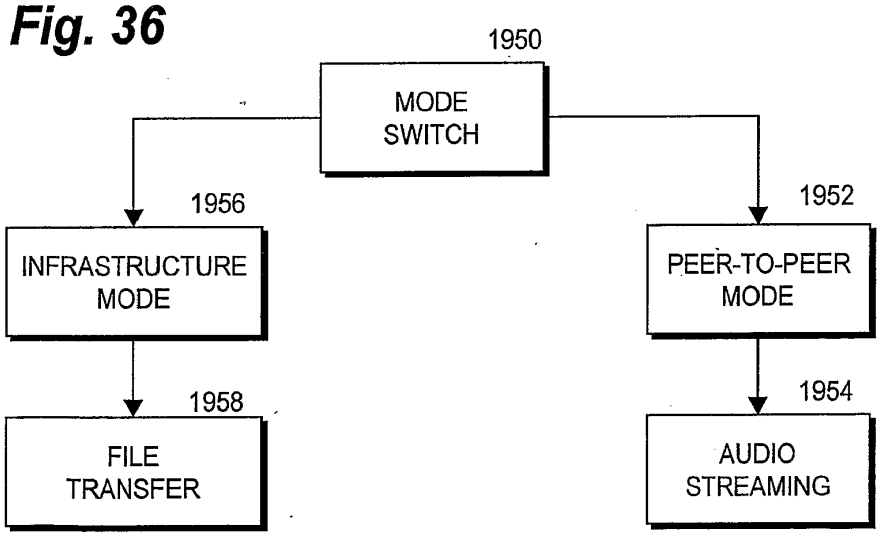

1950
MODE SWITCH
1956
INFRASTRUCTURE MODE
1952
PEER-TO-PEER MODE
1958
FILE TRANSFER
1954
AUDIO STREAMING

Fig. 8A

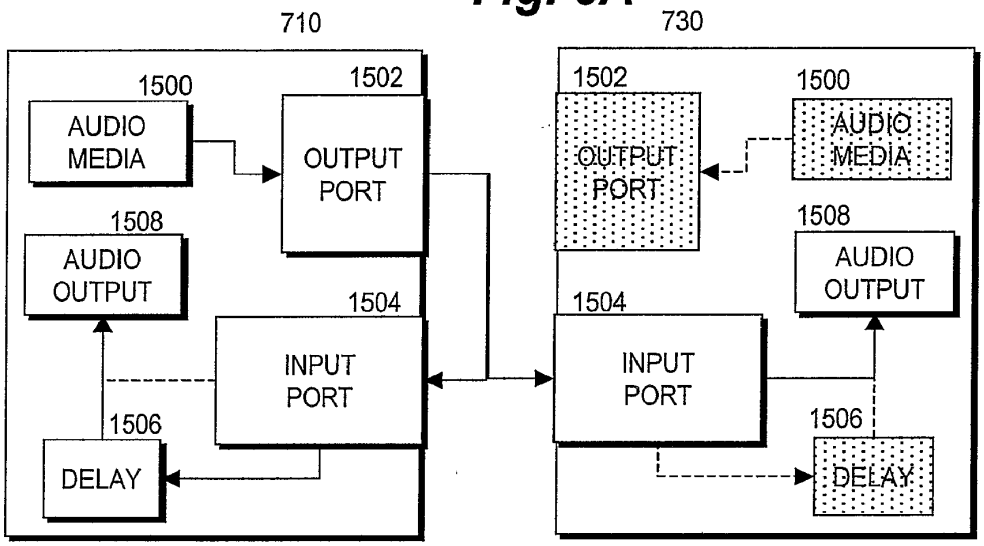

710
730
1500
AUDIO MEDIA
1502
OUTPUT PORT
1508
AUDIO OUTPUT
1504
INPUT PORT
1506
DELAY
1502
OUTPUT PORT
1500
AUDIO MEDIA
1508
AUDIO OUTPUT
1504
INPUT PORT
1506
DELAY

Fig. 8B

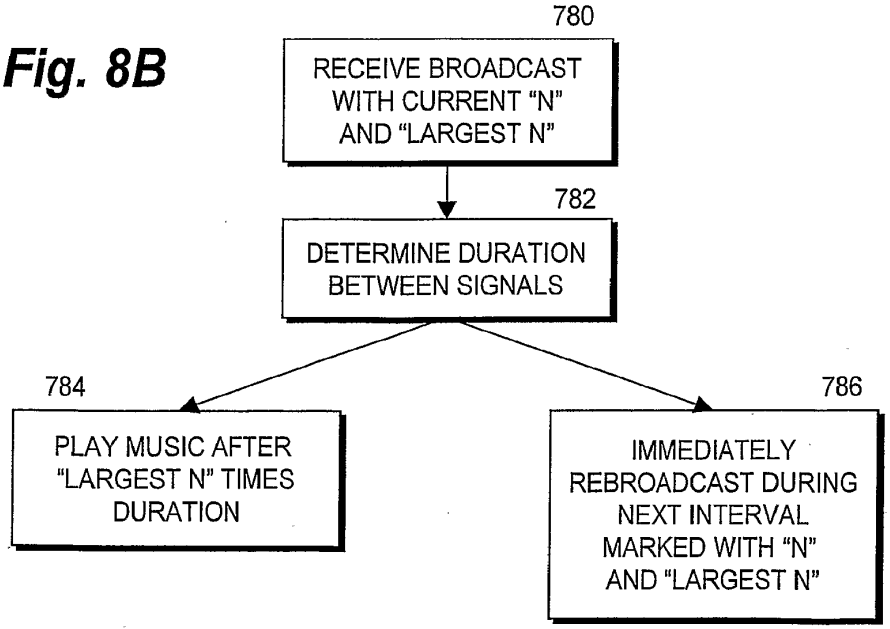

780
RECEIVE BROADCAST WITH CURRENT "N" AND "LARGEST N"
782
DETERMINE DURATION BETWEEN SIGNALS
784
PLAY MUSIC AFTER "LARGEST N" TIMES DURATION
786
IMMEDIATELY REBROADCAST DURING NEXT INTERVAL MARKED WITH "N" AND "LARGEST N"

Fig. 9A

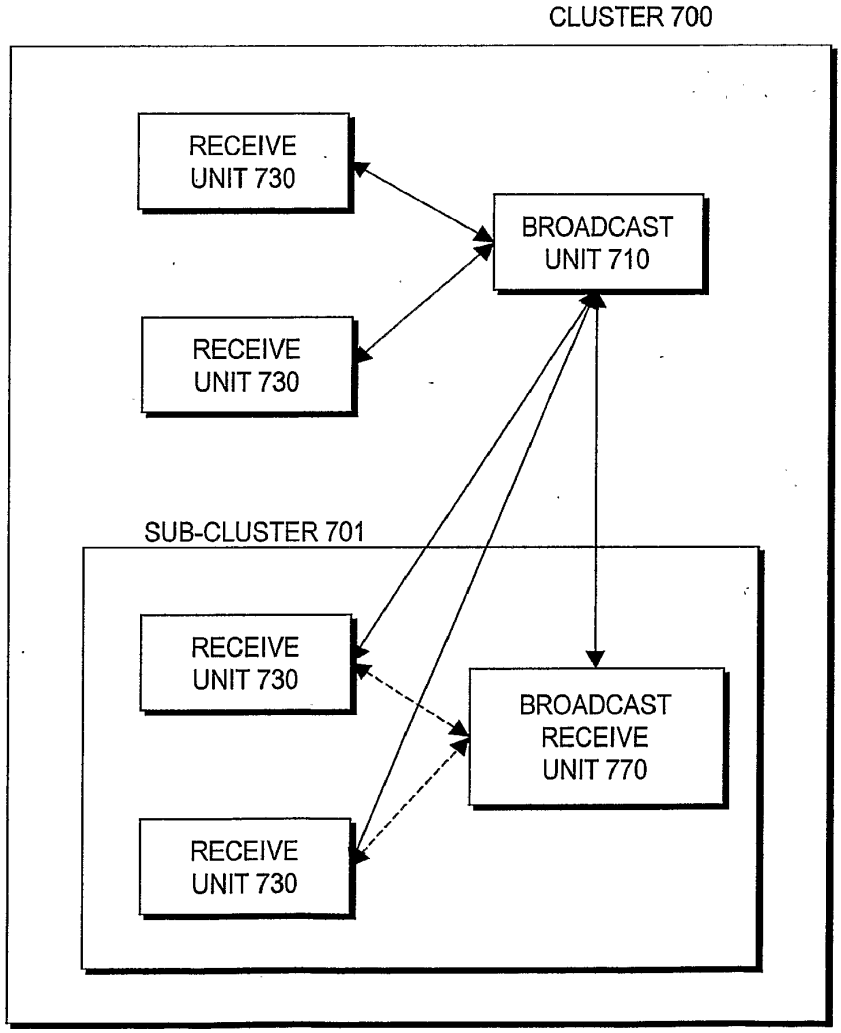

CLUSTER 700
RECEIVE UNIT 730
BROADCAST UNIT 710
RECEIVE UNIT 730
SUB-CLUSTER 701
RECEIVE UNIT 730
BROADCAST RECEIVE UNIT 770
RECEIVE UNIT 730

Fig. 9B

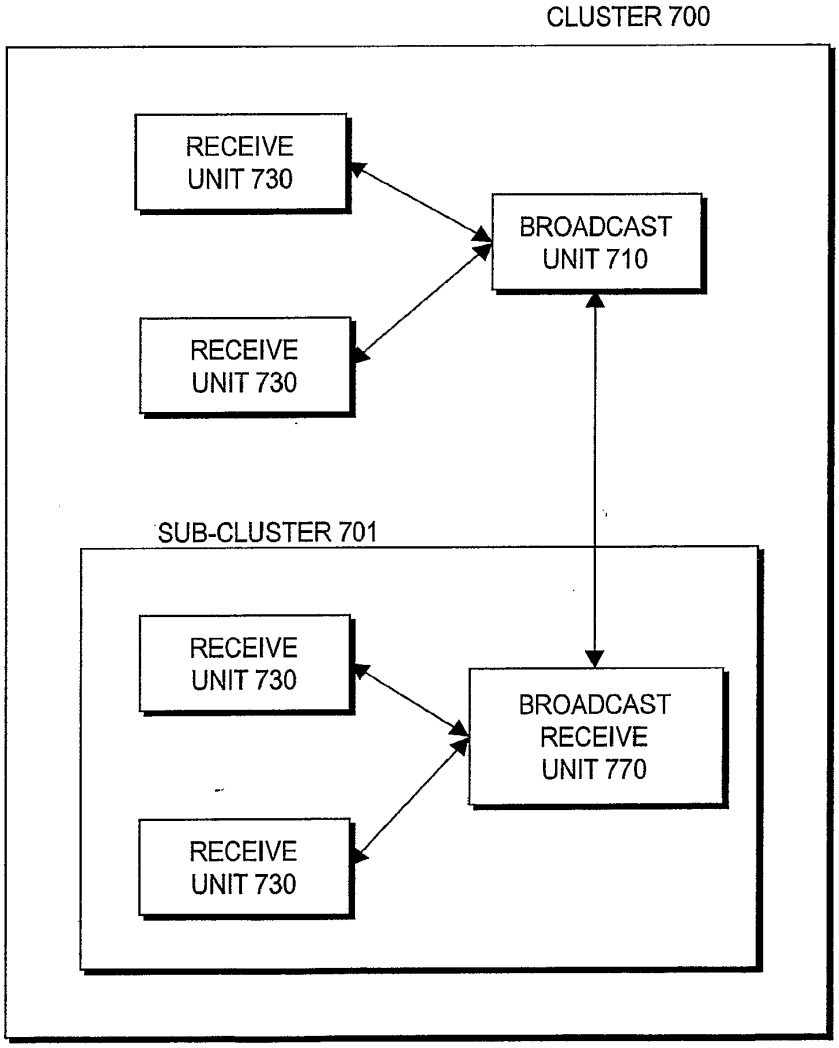

CLUSTER 700
RECEIVE UNIT 730
BROADCAST UNIT 710
RECEIVE UNIT 730
SUB-CLUSTER 701
RECEIVE UNIT 730
BROADCAST RECEIVE UNIT 770
RECEIVE UNIT 730

Fig. 10

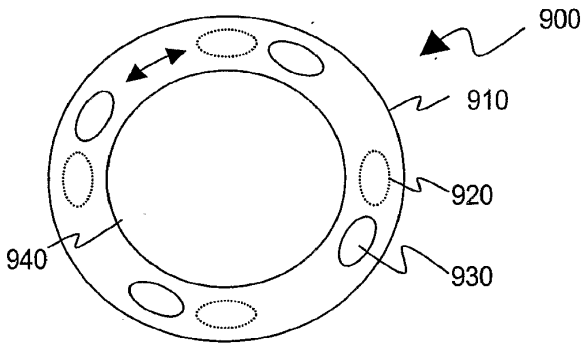


Fig. 11A

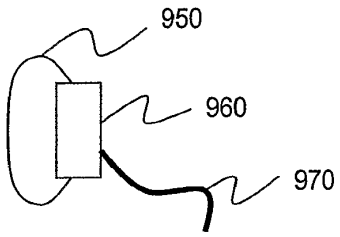


Fig. 11B

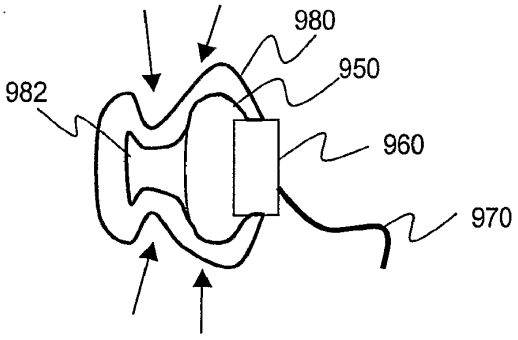


Fig. 12A

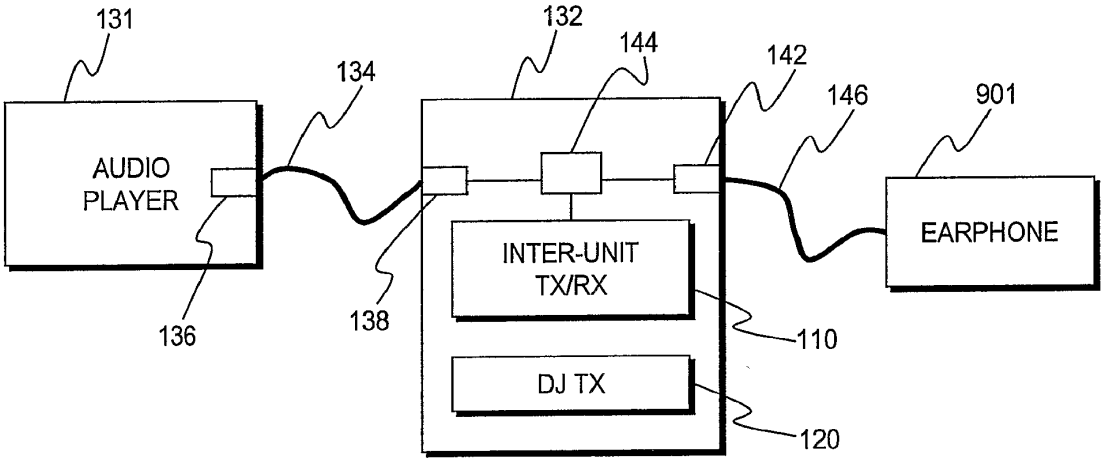


Fig. 12B

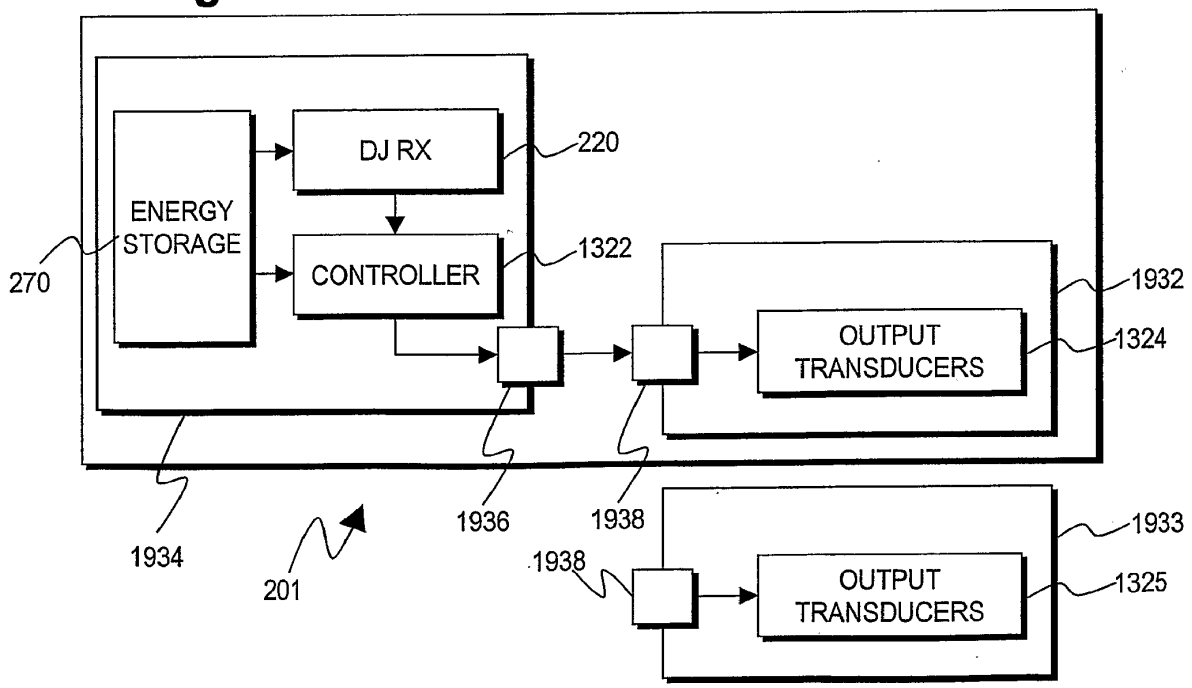


Fig. 12C

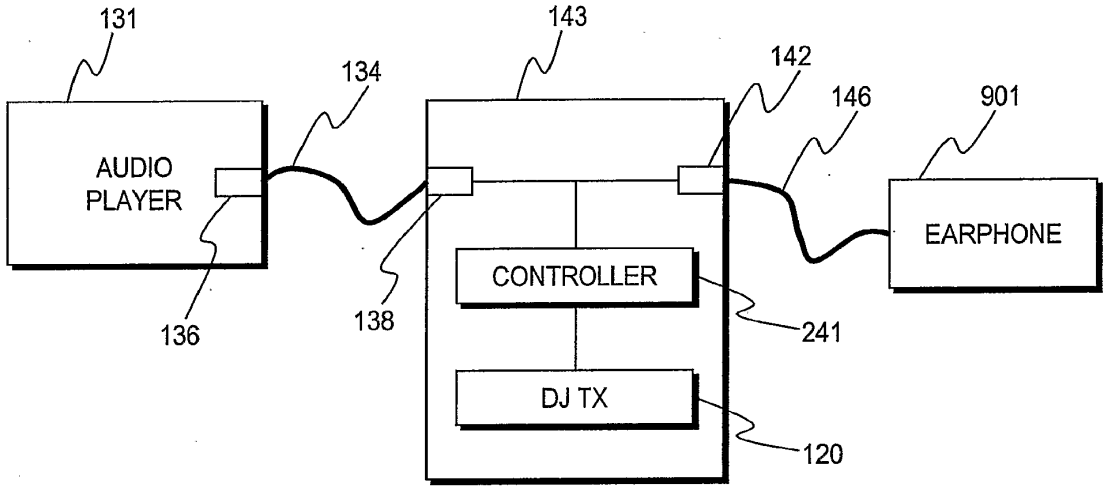

131
134
143
142
146
901
AUDIO PLAYER
EARPHONE
CONTROLLER
DJ TX
136
138
241
120

Fig. 13A

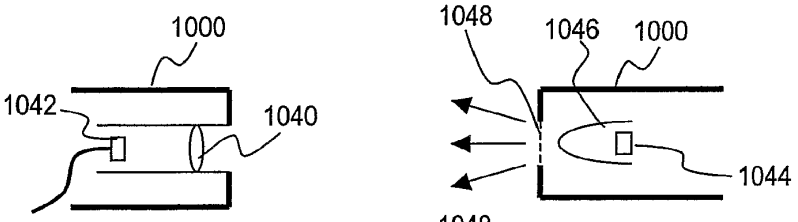


Fig. 13B

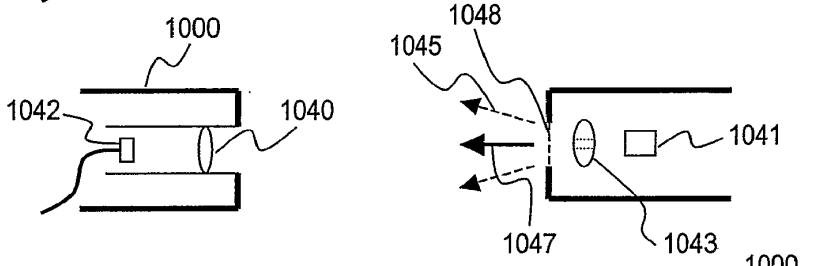


Fig. 13C

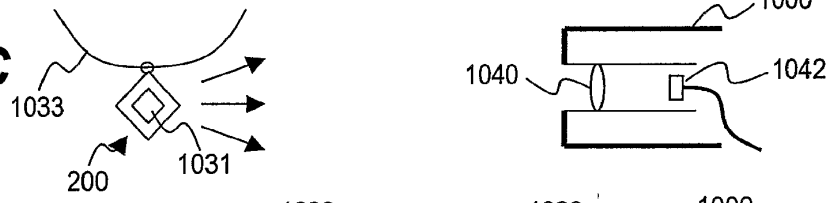


Fig. 13D

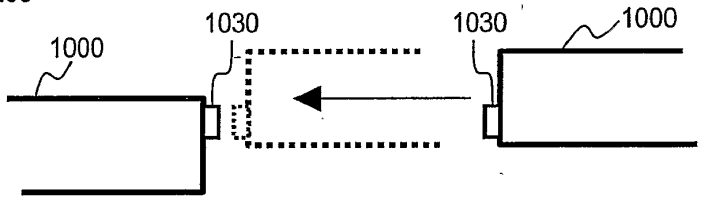


Fig. 13E

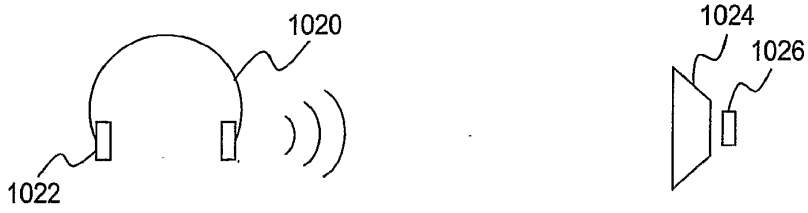


Fig. 13F

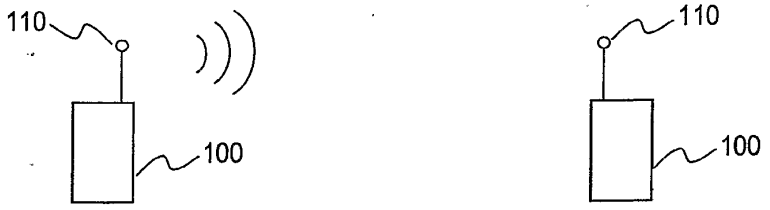


Fig. 14A

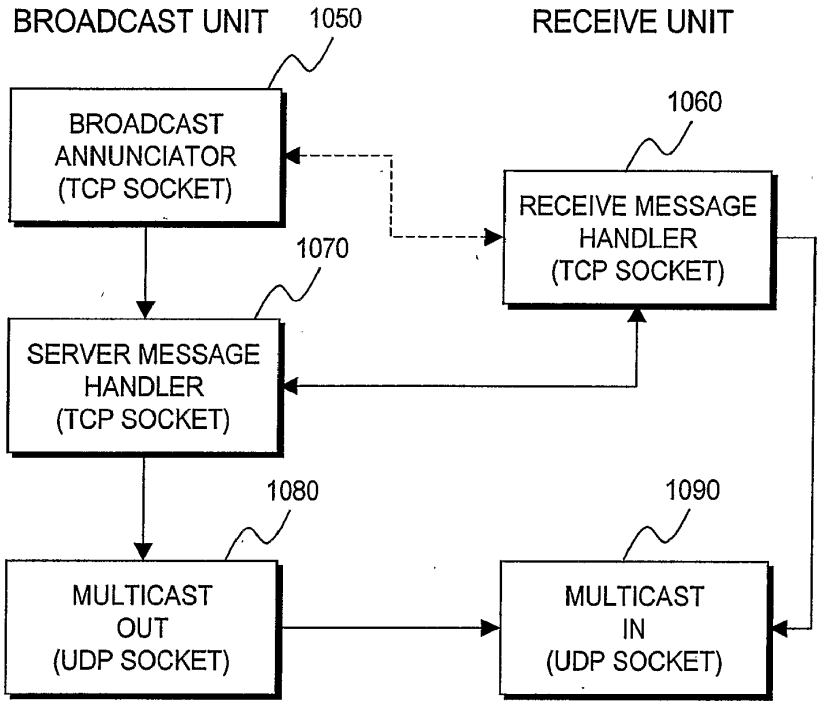

BROADCAST UNIT
1050
RECEIVE UNIT
BROADCAST ANNUNCIATOR (TCP SOCKET)
1060
RECEIVE MESSAGE HANDLER (TCP SOCKET)
1070
SERVER MESSAGE HANDLER (TCP SOCKET)
1080
1090
MULTICAST OUT (UDP SOCKET)
MULTICAST IN (UDP SOCKET)

Fig. 14B

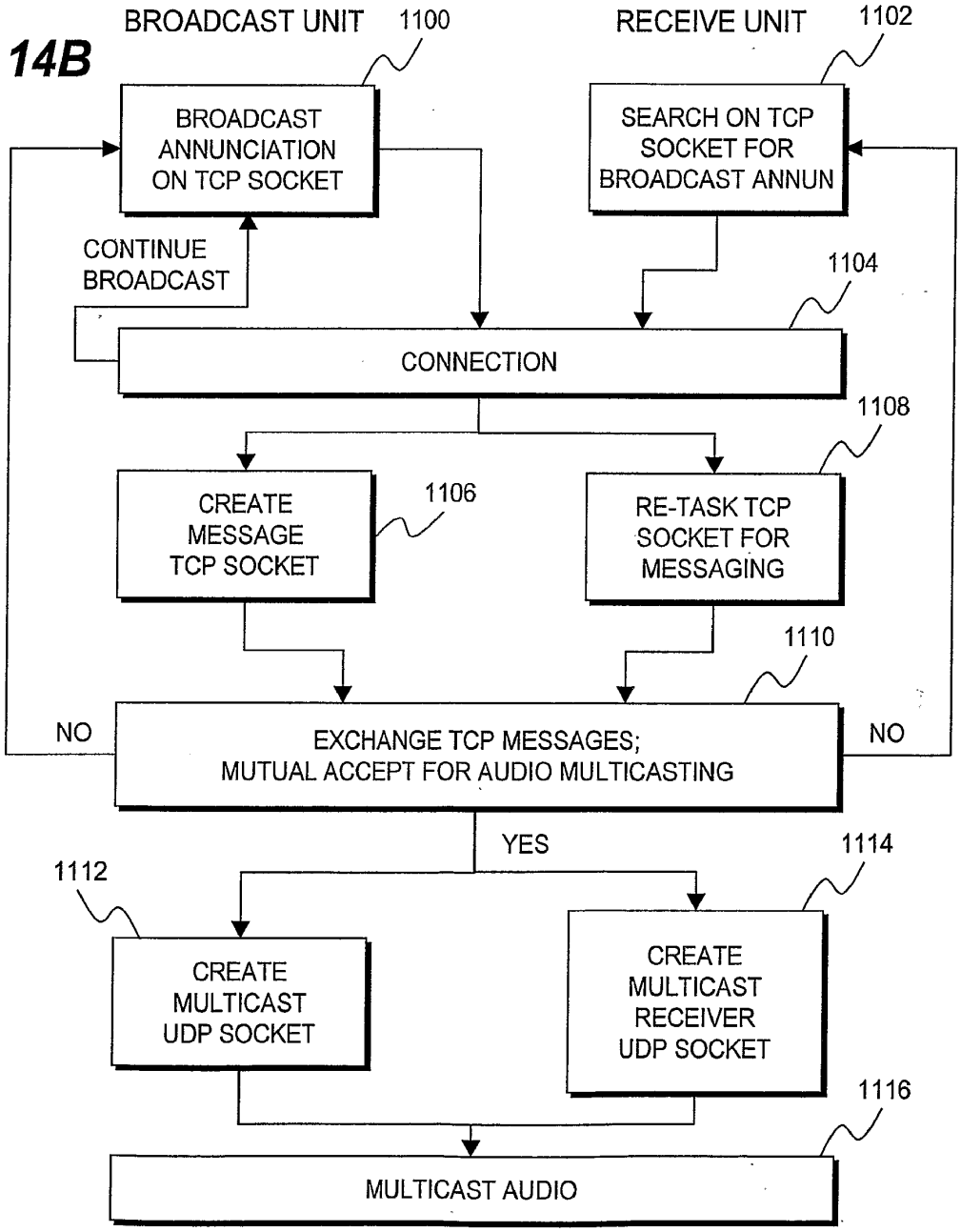

BROADCAST UNIT
1100
RECEIVE UNIT
1102
BROADCAST ANNUNCIATION ON TCP SOCKET
SEARCH ON TCP SOCKET FOR BROADCAST ANNUN
CONTINUE BROADCAST
1104
CONNECTION
1108
CREATE MESSAGE TCP SOCKET
1106
RE-TASK TCP SOCKET FOR MESSAGING
1110
NO
EXCHANGE TCP MESSAGES; MUTUAL ACCEPT FOR AUDIO MULTICASTING
NO
YES
1112
1114
CREATE MULTICAST UDP SOCKET
CREATE MULTICAST RECEIVER UDP SOCKET
1116
MULTICAST AUDIO

Fig. 15

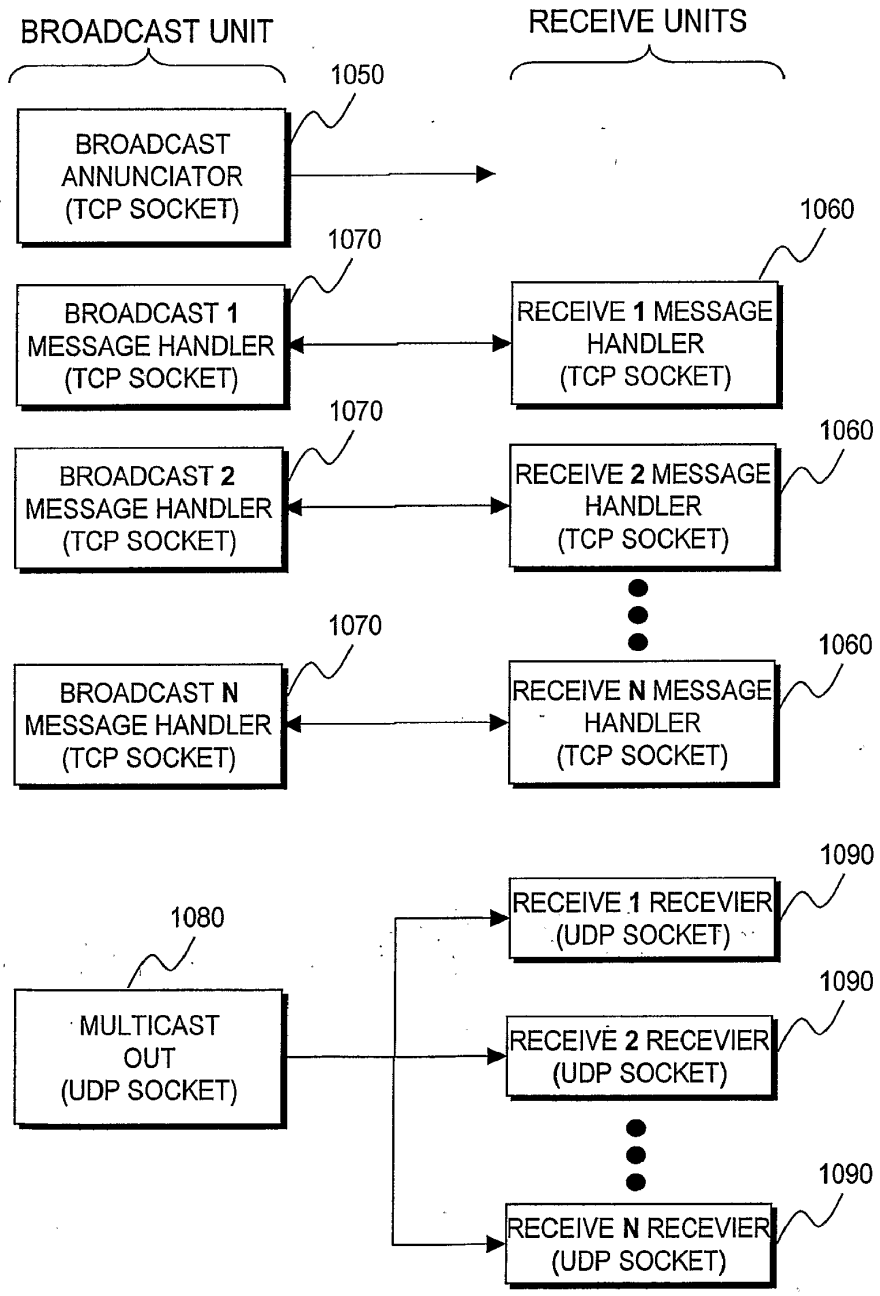

BROADCAST UNIT
RECEIVE UNITS
1050
BROADCAST ANNUNCIATOR (TCP SOCKET)
1060
1070
BROADCAST 1 MESSAGE HANDLER (TCP SOCKET)
RECEIVE 1 MESSAGE HANDLER (TCP SOCKET)
1070
1060
BROADCAST 2 MESSAGE HANDLER (TCP SOCKET)
RECEIVE 2 MESSAGE HANDLER (TCP SOCKET)
1070
1060
BROADCAST N MESSAGE HANDLER (TCP SOCKET)
RECEIVE N MESSAGE HANDLER (TCP SOCKET)
1090
1080
RECEIVE 1 RECEVIER (UDP SOCKET)
MULTICAST OUT (UDP SOCKET)
1090
RECEIVE 2 RECEVIER (UDP SOCKET)
1090
RECEIVE N RECEVIER (UDP SOCKET)

Fig. 16

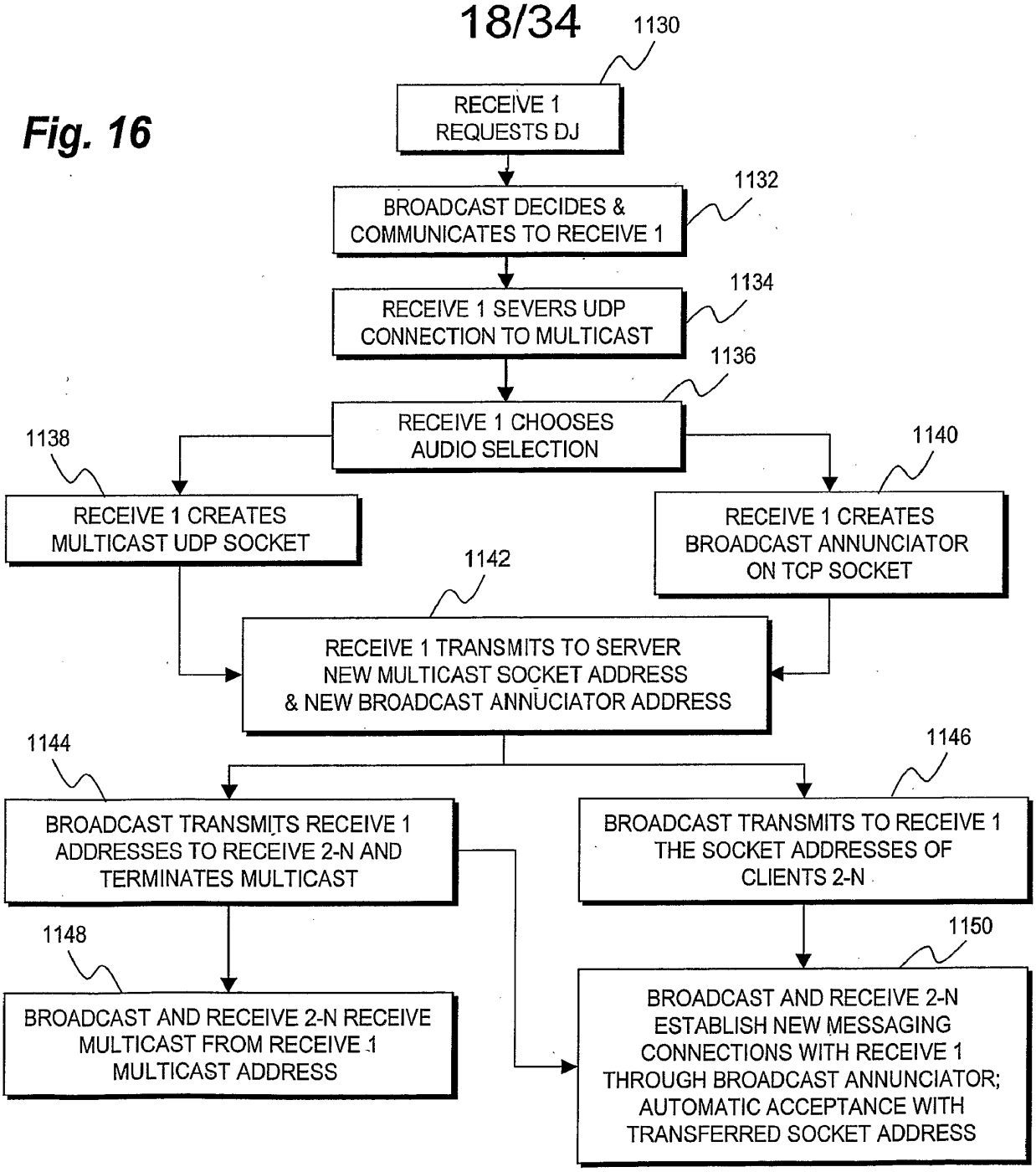

1130
RECEIVE 1 REQUESTS DJ
1132
BROADCAST DECIDES & COMMUNICATES TO RECEIVE 1
1134
RECEIVE 1 SEVERS UDP CONNECTION TO MULTICAST
1136
RECEIVE 1 CHOOSES AUDIO SELECTION
1138
RECEIVE 1 CREATES MULTICAST UDP SOCKET
1140
RECEIVE 1 CREATES BROADCAST ANNUNCIATOR ON TCP SOCKET
1142
RECEIVE 1 TRANSMITS TO SERVER NEW MULTICAST SOCKET ADDRESS & NEW BROADCAST ANNUCIATOR ADDRESS
1144
BROADCAST TRANSMITS RECEIVE 1 ADDRESSES TO RECEIVE 2-N AND TERMINATES MULTICAST
1146
BROADCAST TRANSMITS TO RECEIVE 1 THE SOCKET ADDRESSES OF CLIENTS 2-N
1148
BROADCAST AND RECEIVE 2-N RECEIVE MULTICAST FROM RECEIVE 1 MULTICAST ADDRESS
1150
BROADCAST AND RECEIVE 2-N ESTABLISH NEW MESSAGING CONNECTIONS WITH RECEIVE 1 THROUGH BROADCAST ANNUNCIATOR; AUTOMATIC ACCEPTANCE WITH TRANSFERRED SOCKET ADDRESS

Fig. 17

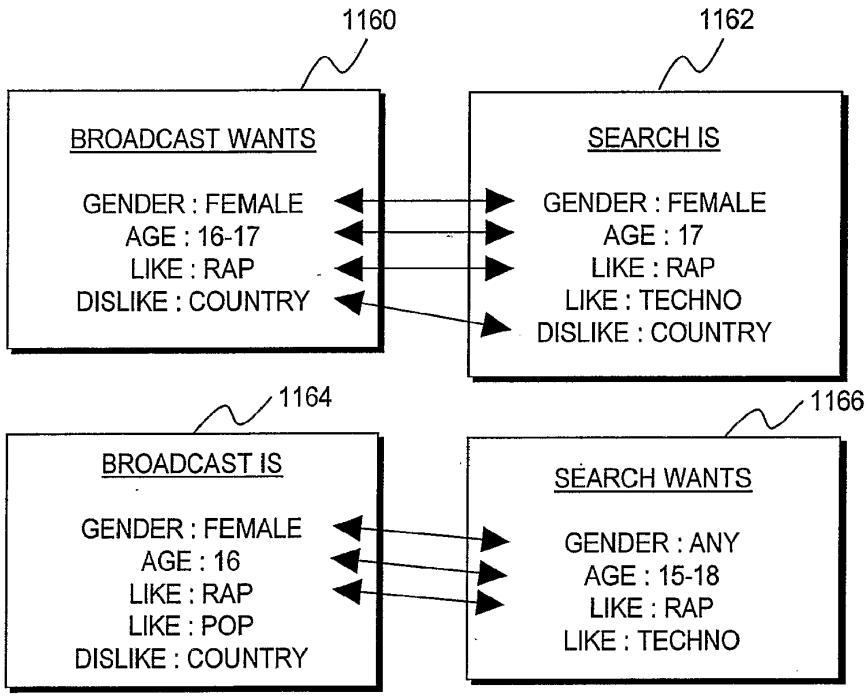


Fig. 18A

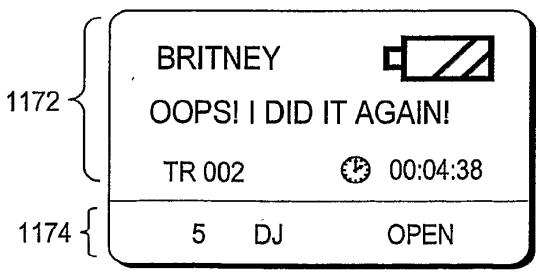


Fig. 18B

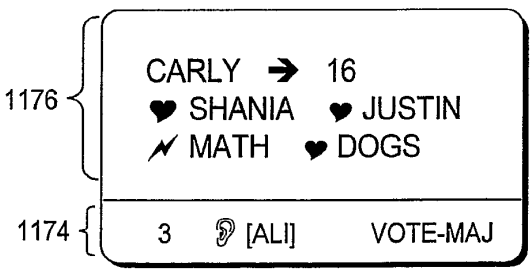


Fig. 19

RULE	METHOD
BROADCASTER	ACCEPTED WHEN BROADCASTER INDICATES "YES". REJECTED WHEN BROADCASTER INDICATES "NO"
MAJORITY	ACCEPTED WHEN (N/2)+1 INDICATE "YES". REJECTED WHEN (N/2) INDICATE "NO".
UNANIMOUS	ACCEPTED WHEN N MEMBERS INDICATE "YES". REJECTED WHEN 1 MEMBER INDICATES "NO".
TIMED MAJORITY	MACHINE INDICATES ONSET OF VOTING, AND COUNTS DOWN TO VOTING TERMINATION. ACCEPTED IF (Q/2)+1 INDICATE "YES" BEFORE TERMINATION; OTHERWISE "NO".
SYNCHRONIZED MAJORITY	MACHINE COUNTS DOWN TO VOTE. ACCEPTED IF (Q/2)+1 INDICATE "YES" WITHIN PREDETERMINED TIME OF COUNTDOWN. OTHERWISE "NO".

Fig. 20

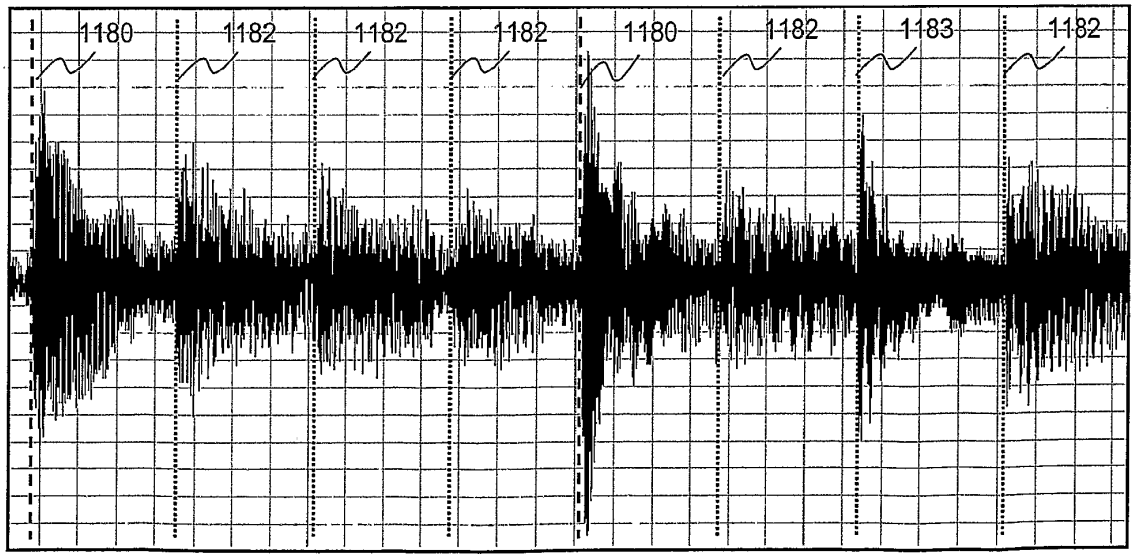


Fig. 21A

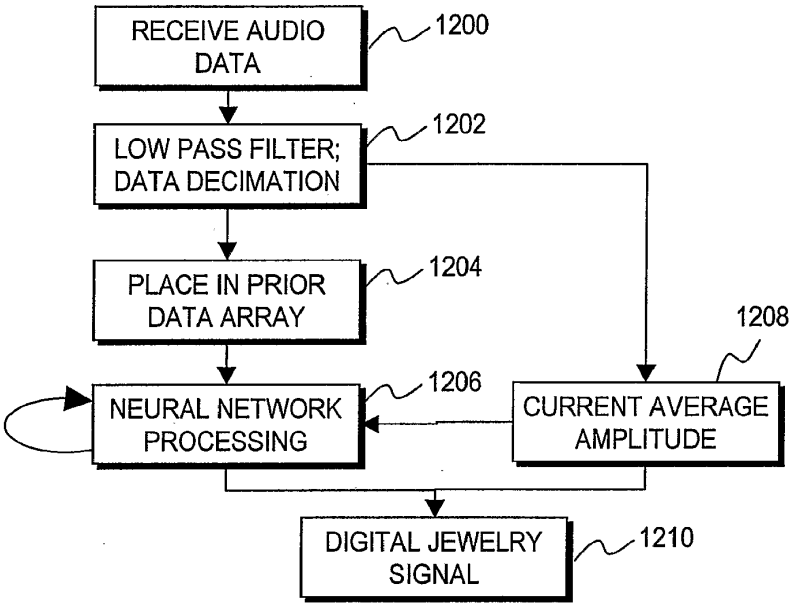


Fig. 21B

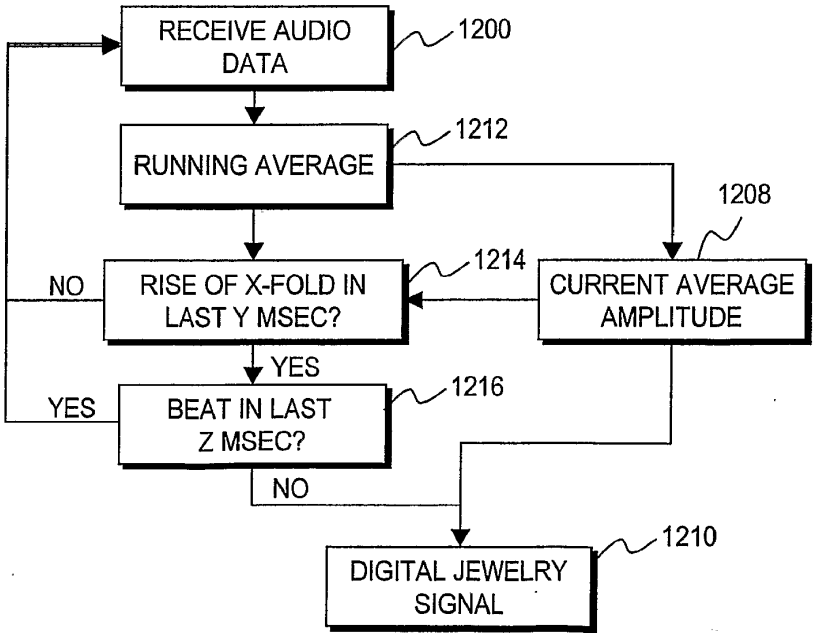


Fig. 21C

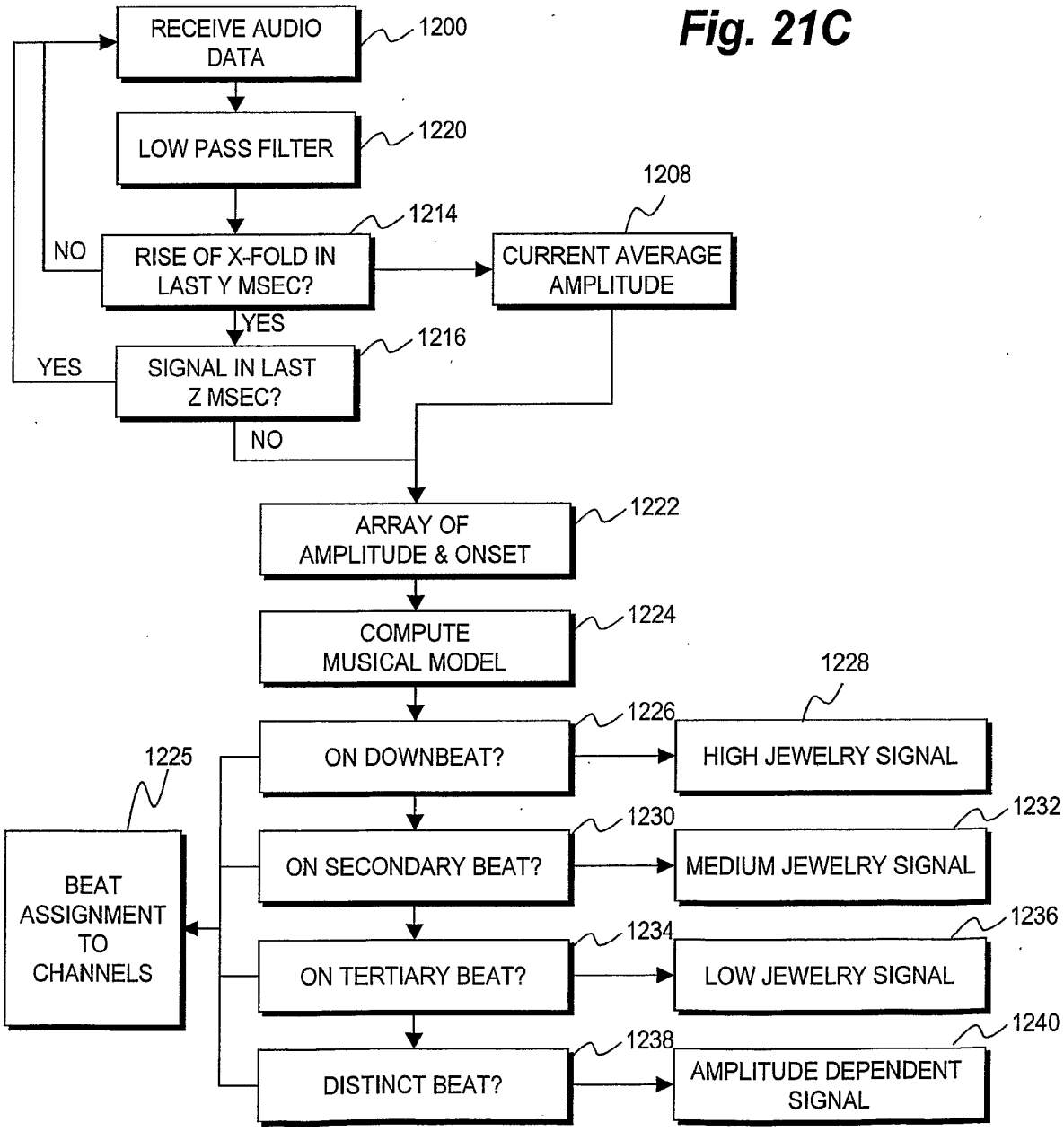
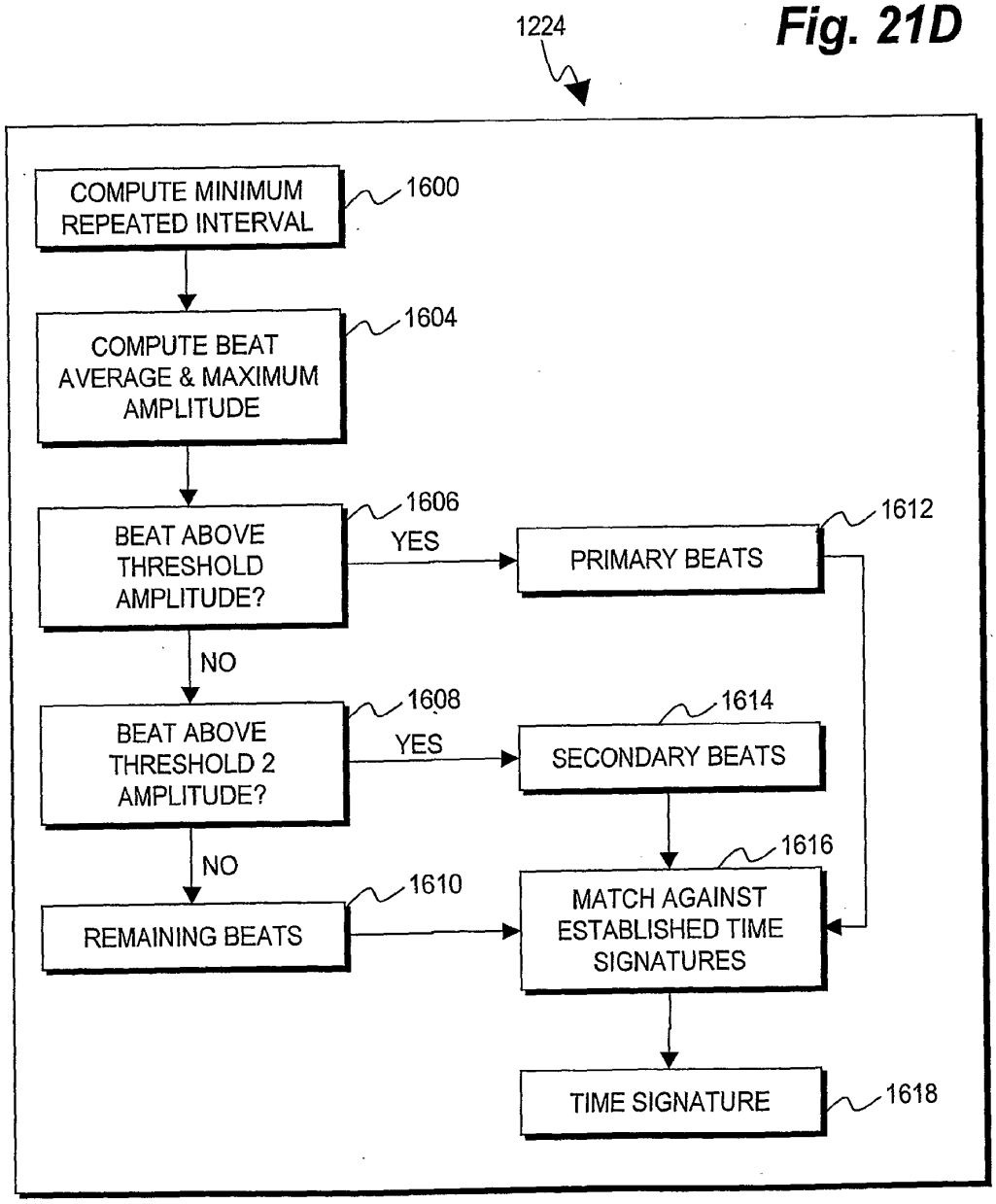

RECEIVE AUDIO DATA
1200
LOW PASS FILTER
1220
1208
1214
NO
RISE OF X-FOLD IN LAST Y MSEC?
CURRENT AVERAGE AMPLITUDE
YES
1216
YES
SIGNAL IN LAST Z MSEC?
NO
ARRAY OF AMPLITUDE & ONSET
1222
COMPUTE MUSICAL MODEL
1224
1228
1226
ON DOWNBEAT?
HIGH JEWELRY SIGNAL
1225
1230
1232
ON SECONDARY BEAT?
MEDIUM JEWELRY SIGNAL
BEAT ASSIGNMENT TO CHANNELS
1234
1236
ON TERTIARY BEAT?
LOW JEWELRY SIGNAL
1238
1240
DISTINCT BEAT?
AMPLITUDE DEPENDENT SIGNAL

Fig. 21D


1224
COMPUTE MINIMUM REPEATED INTERVAL
1600
COMPUTE BEAT AVERAGE & MAXIMUM AMPLITUDE
1604
BEAT ABOVE THRESHOLD AMPLITUDE?
1606
YES
PRIMARY BEATS
1612
NO
BEAT ABOVE THRESHOLD 2 AMPLITUDE?
1608
YES
SECONDARY BEATS
1614
NO
REMAINING BEATS
1610
MATCH AGAINST ESTABLISHED TIME SIGNATURES
1616
TIME SIGNATURE
1618

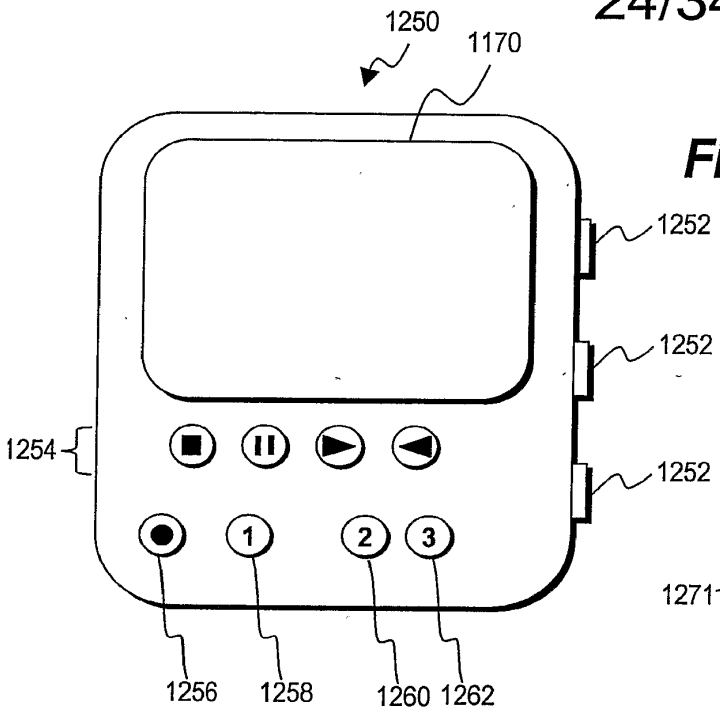


Fig. 22A

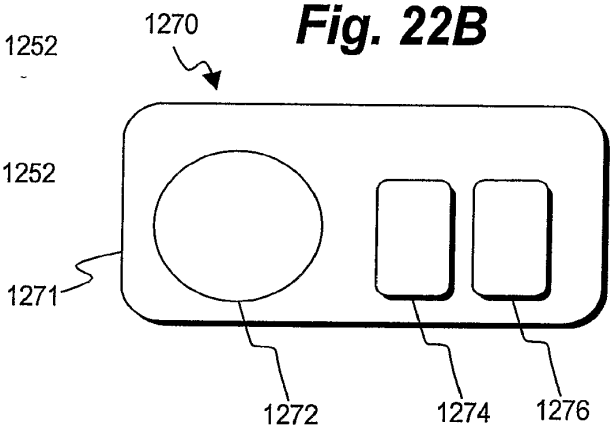


Fig. 22B

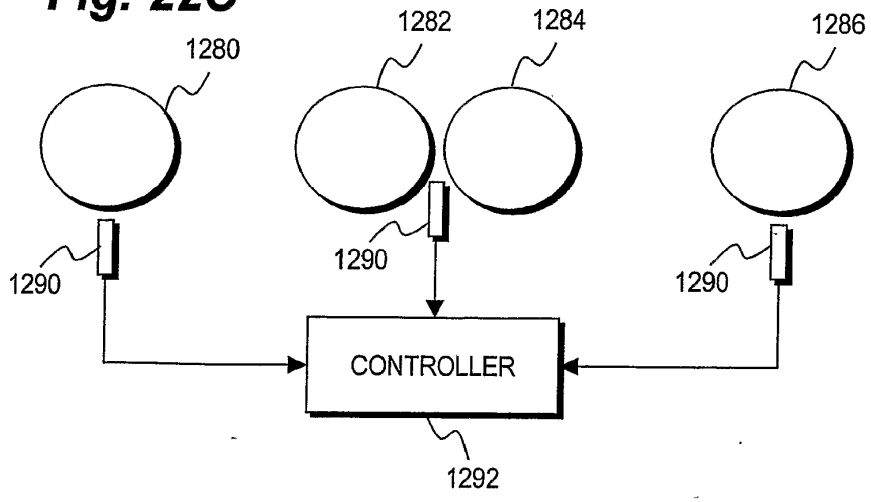


Fig. 22C

Fig. 23

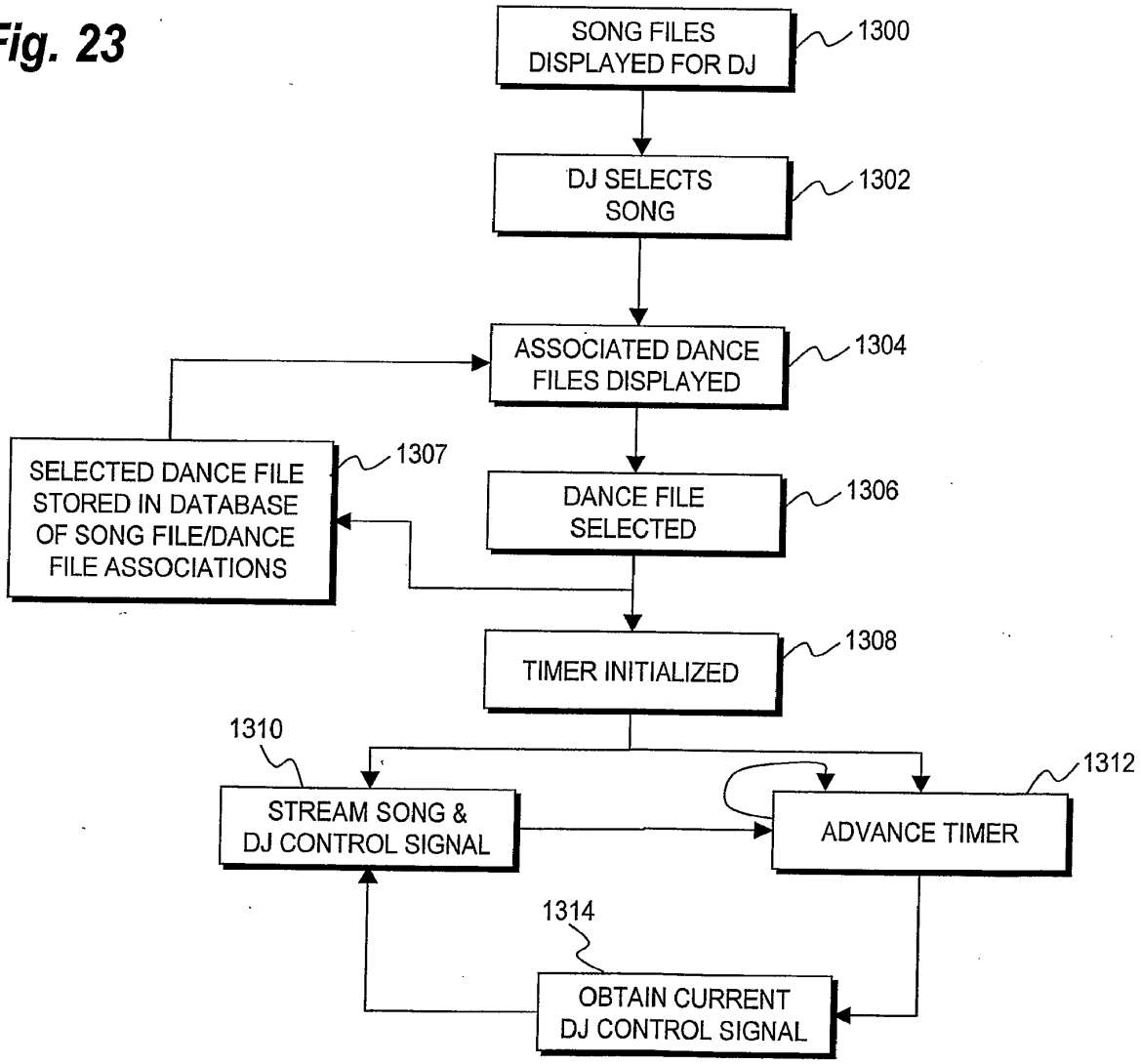

SONG FILES DISPLAYED FOR DJ
1300
DJ SELECTS SONG
1302
ASSOCIATED DANCE FILES DISPLAYED
1304
SELECTED DANCE FILE STORED IN DATABASE OF SONG FILE/DANCE FILE ASSOCIATIONS
1307
DANCE FILE SELECTED
1306
TIMER INITIALIZED
1308
1310
STREAM SONG & DJ CONTROL SIGNAL
ADVANCE TIMER
1312
1314
OBTAIN CURRENT DJ CONTROL SIGNAL

Fig. 24

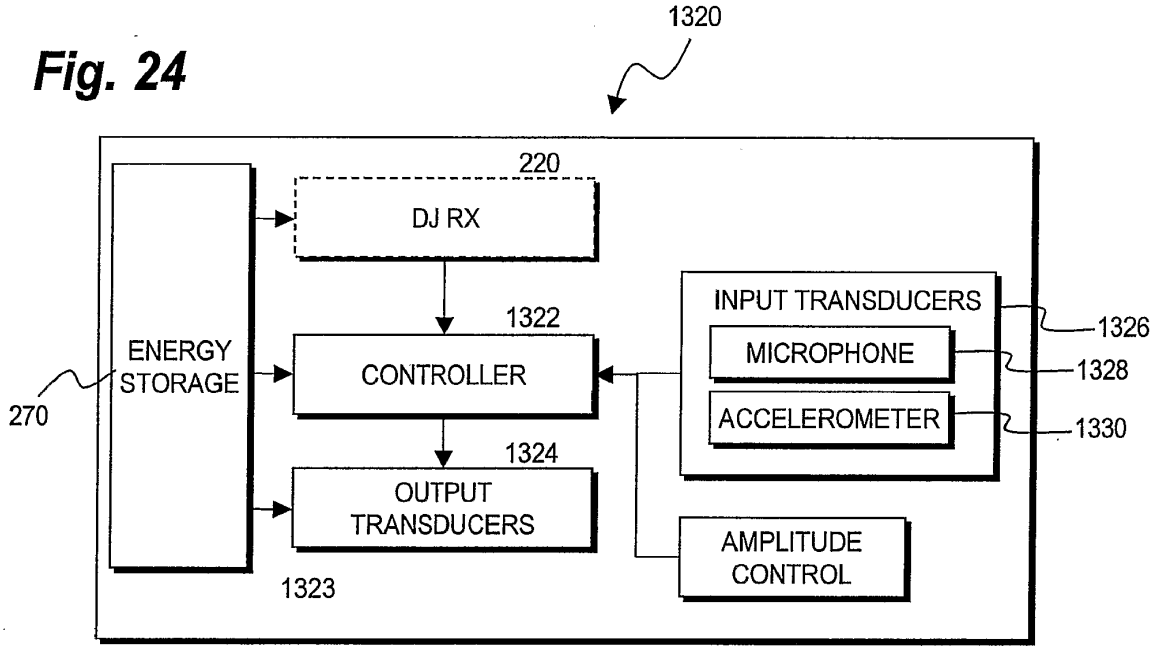

1320
220
DJ RX
1322
ENERGY STORAGE
CONTROLLER
270
1324
OUTPUT TRANSDUCERS
1323
INPUT TRANSDUCERS
MICROPHONE
ACCELEROMETER
1326
1328
1330
AMPLITUDE CONTROL

Fig. 25

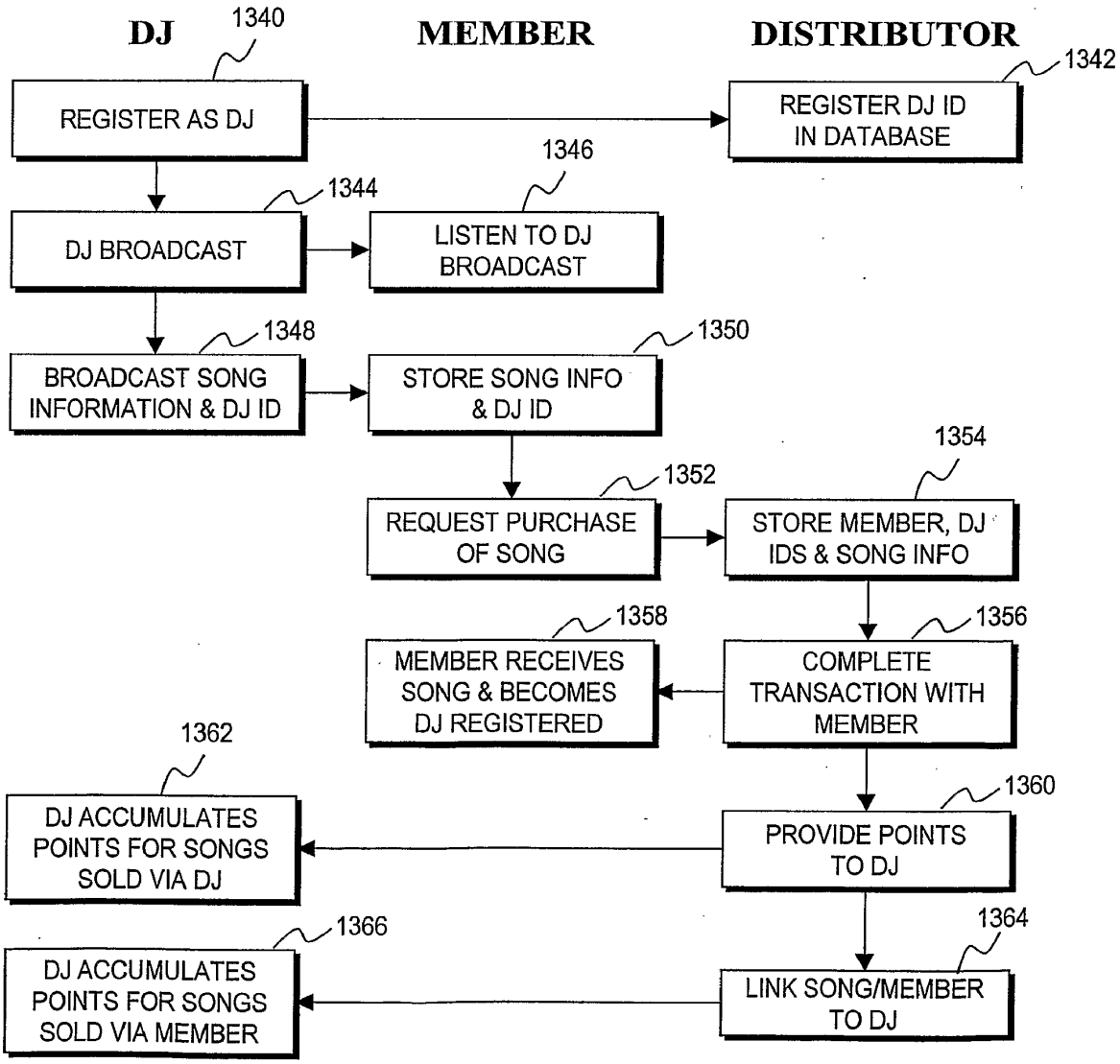

DJ
MEMBER
DISTRIBUTOR
1340
1342
REGISTER AS DJ
REGISTER DJ ID IN DATABASE
1346
1344
DJ BROADCAST
LISTEN TO DJ BROADCAST
1348
1350
BROADCAST SONG INFORMATION & DJ ID
STORE SONG INFO & DJ ID
1354
1352
REQUEST PURCHASE OF SONG
STORE MEMBER, DJ IDS & SONG INFO
1358
1356
MEMBER RECEIVES SONG & BECOMES DJ REGISTERED
COMPLETE TRANSACTION WITH MEMBER
1362
1360
DJ ACCUMULATES POINTS FOR SONGS SOLD VIA DJ
PROVIDE POINTS TO DJ
1366
1364
DJ ACCUMULATES POINTS FOR SONGS SOLD VIA MEMBER
LINK SONG/MEMBER TO DJ

Fig. 26

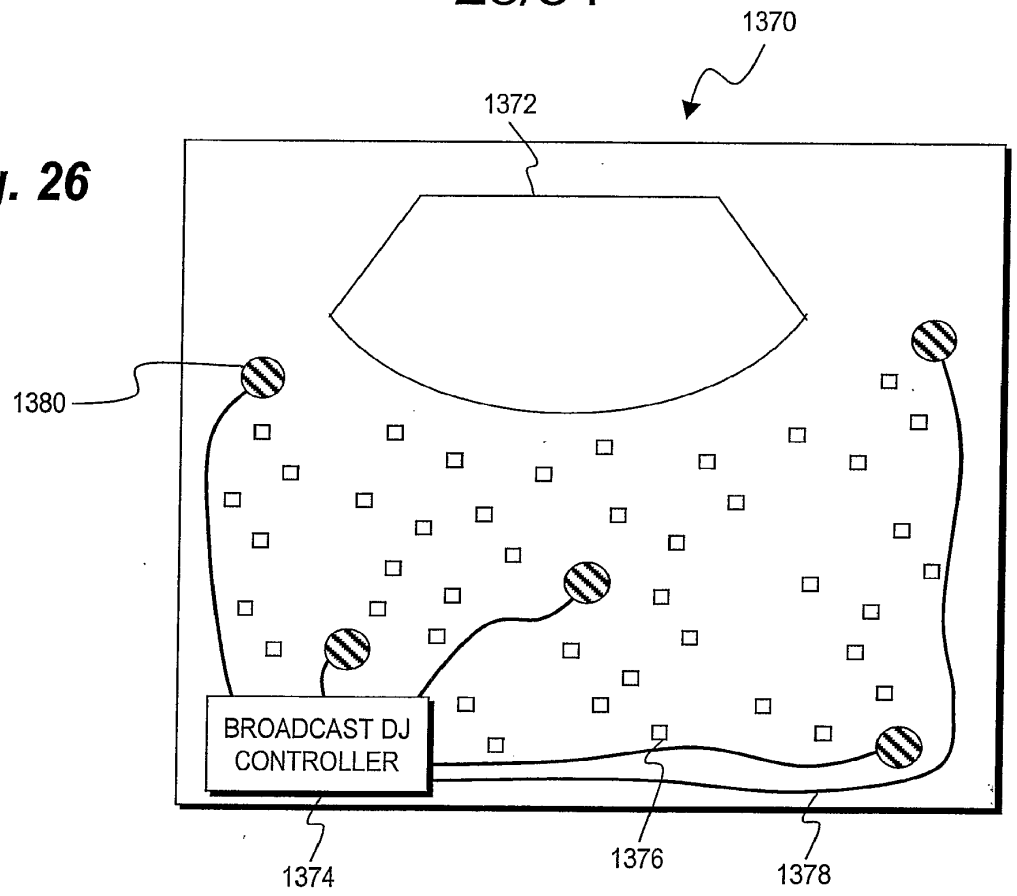

1370
1372
1380
BROADCAST DJ CONTROLLER
1374
1376
1378

Fig. 35

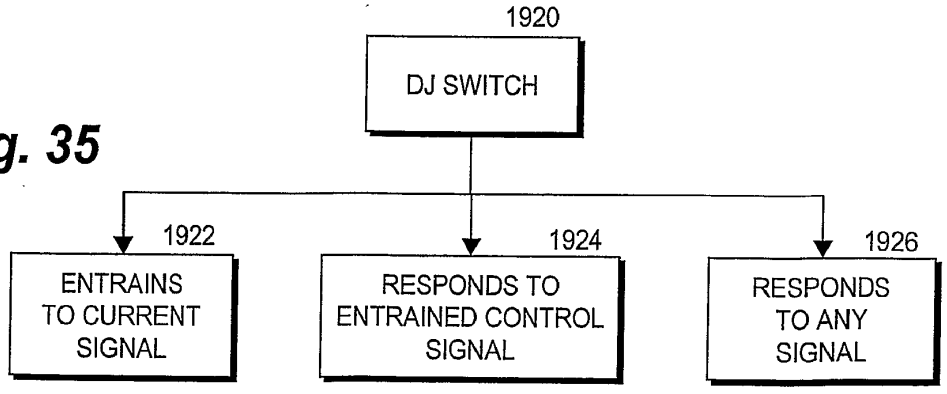

1920
DJ SWITCH
1922
ENTRAINS TO CURRENT SIGNAL
1924
RESPONDS TO ENTRAINED CONTROL SIGNAL
1926
RESPONDS TO ANY SIGNAL

Fig. 27

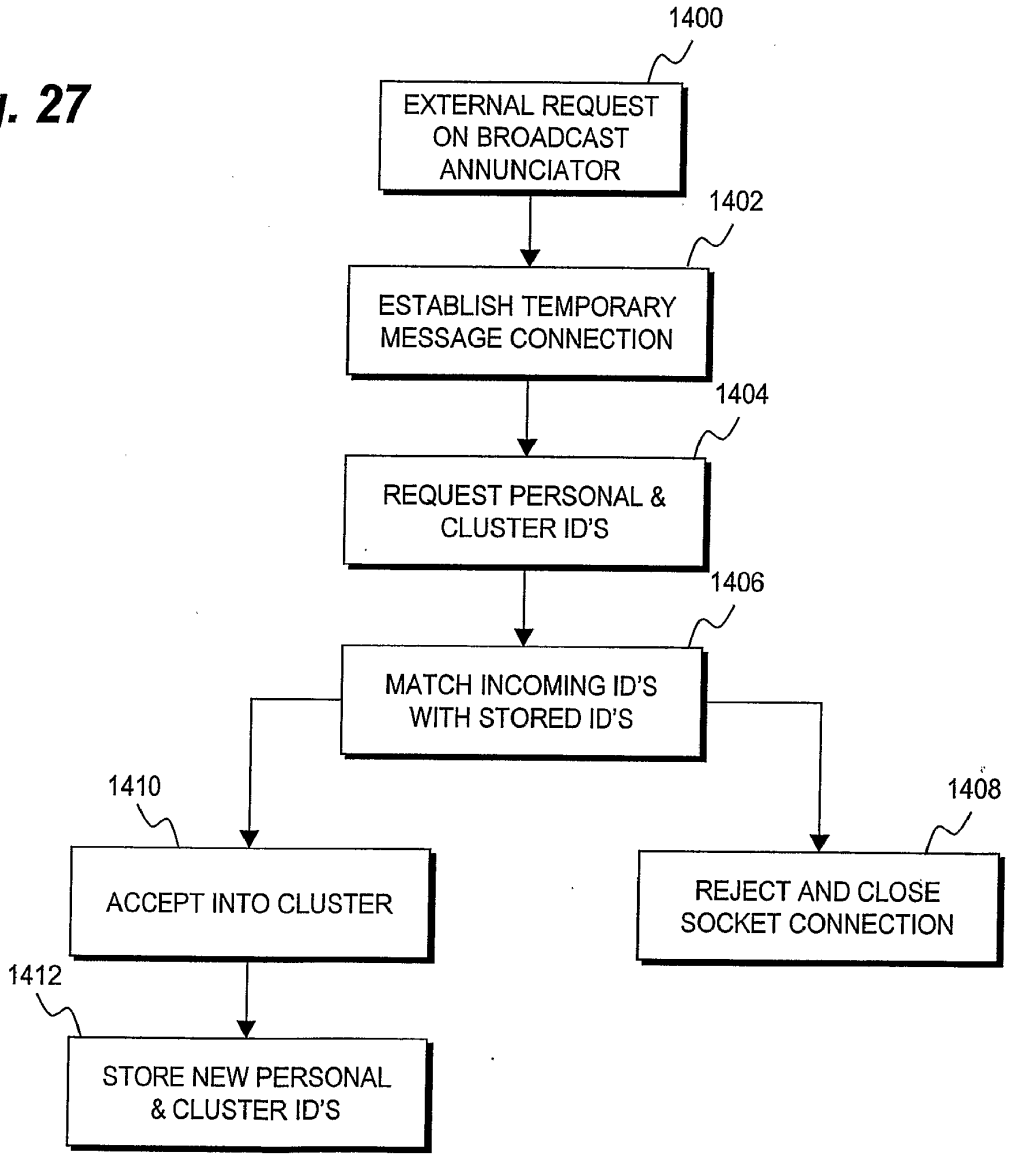

1400
EXTERNAL REQUEST ON BROADCAST ANNUNCIATOR
1402
ESTABLISH TEMPORARY MESSAGE CONNECTION
1404
REQUEST PERSONAL & CLUSTER ID'S
1406
MATCH INCOMING ID'S WITH STORED ID'S
1410
ACCEPT INTO CLUSTER
1408
REJECT AND CLOSE SOCKET CONNECTION
1412
STORE NEW PERSONAL & CLUSTER ID'S

Fig. 28

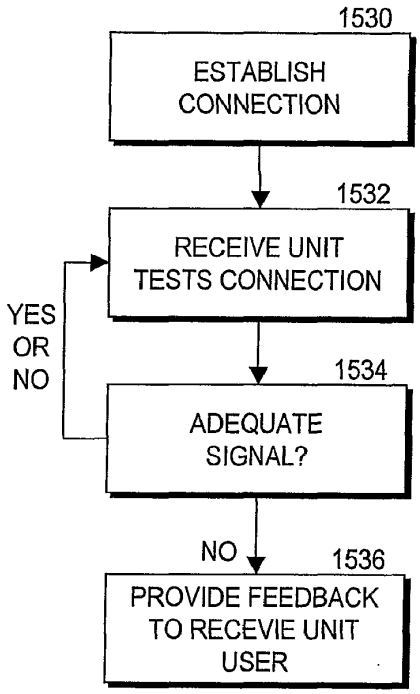

1530
ESTABLISH CONNECTION
1532
RECEIVE UNIT TESTS CONNECTION
YES OR NO
1534
ADEQUATE SIGNAL?
NO
1536
PROVIDE FEEDBACK TO RECEVIE UNIT USER

Fig. 29A

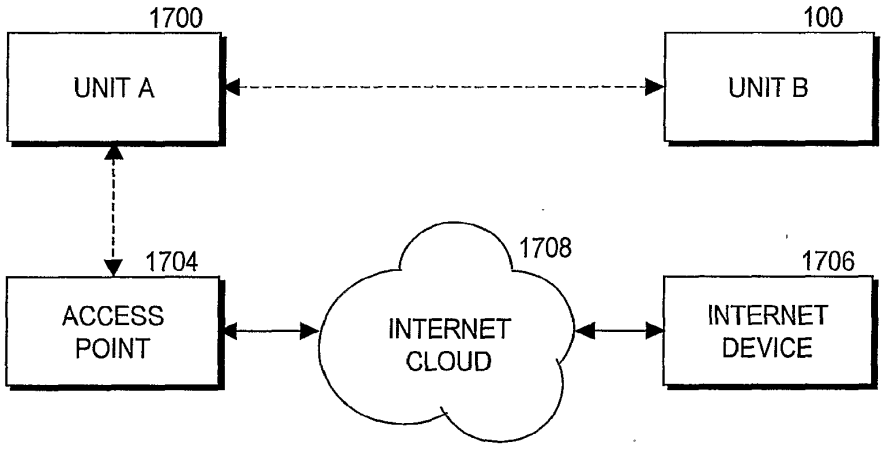


Fig. 29B

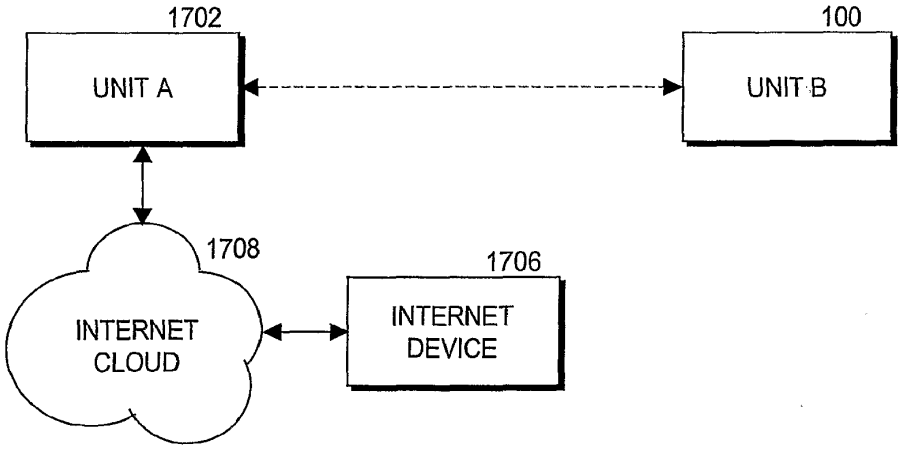


Fig. 30

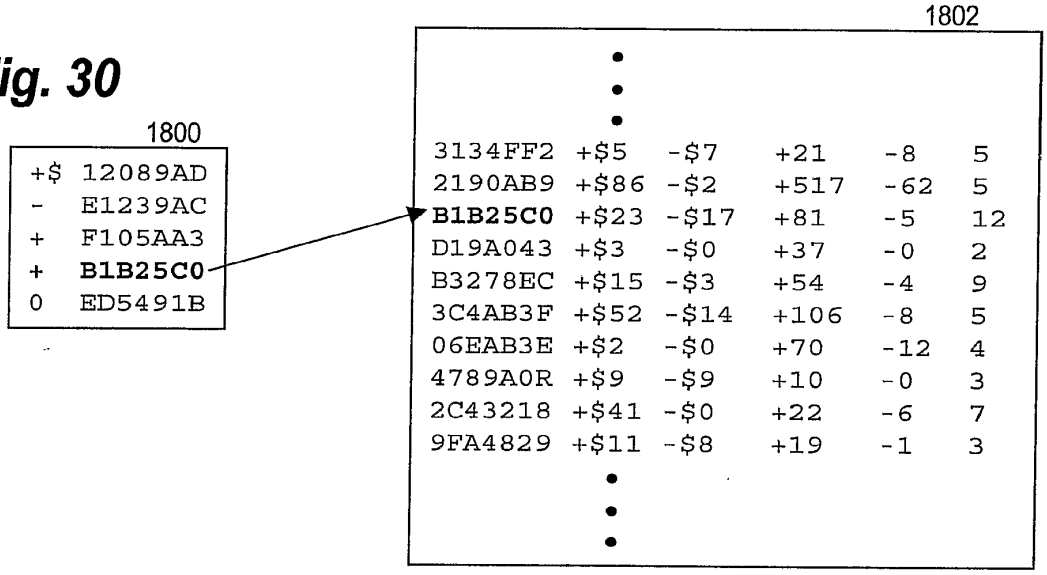

1800
+$ 12089AD
- E1239AC
+ F105AA3
+ B1B25C0
0 ED5491B
1802
3134FF2 +$5 -$7 +21 -8 5
2190AB9 +$86 -$2 +517 -62 5
B1B25C0 +$23 -$17 +81 -5 12
D19A043 +$3 -$0 +37 -0 2
B3278EC +$15 -$3 +54 -4 9
3C4AB3F +$52 -$14 +106 -8 5
06EAB3E +$2 -$0 +70 -12 4
4789A0R +$9 -$9 +10 -0 3
2C43218 +$41 -$0 +22 -6 7
9FA4829 +$11 -$8 +19 -1 3

Fig. 31

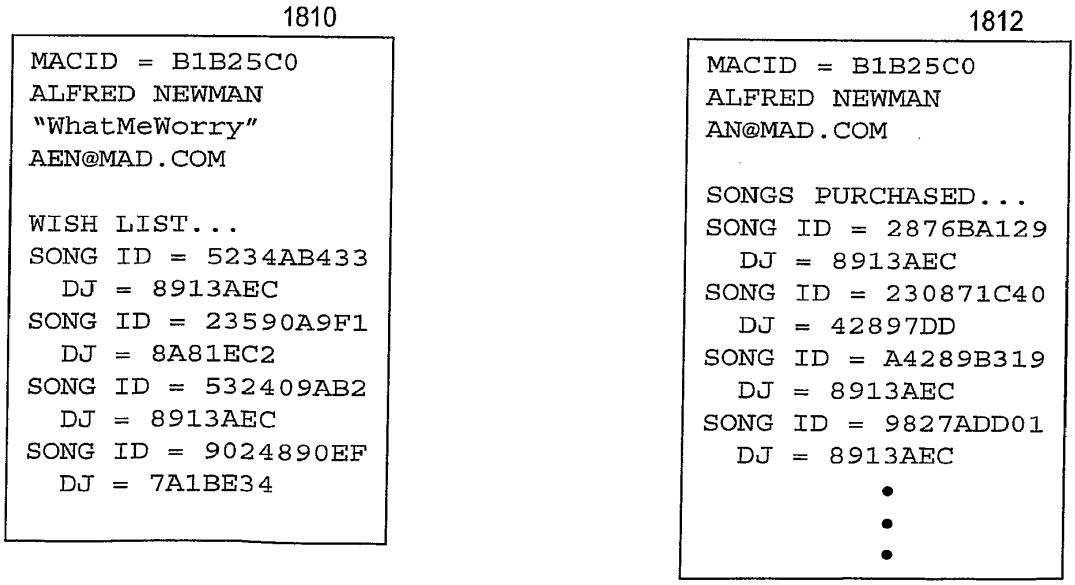

USER
DISTRIBUTOR
1810
MACID = B1B25C0
ALFRED NEWMAN
"WhatMeWorry"
AEN@MAD.COM
WISH LIST...
SONG ID = 5234AB433
DJ = 8913AEC
SONG ID = 23590A9F1
DJ = 8A81EC2
SONG ID = 532409AB2
DJ = 8913AEC
SONG ID = 9024890EF
DJ = 7A1BE34
1812
MACID = B1B25C0
ALFRED NEWMAN
AN@MAD.COM
SONGS PURCHASED...
SONG ID = 2876BA129
DJ = 8913AEC
SONG ID = 230871C40
DJ = 42897DD
SONG ID = A4289B319
DJ = 8913AEC
SONG ID = 9827ADD01
DJ = 8913AEC

Fig. 32A

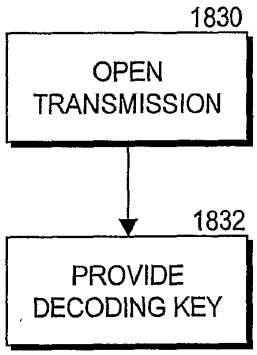


Fig. 32B

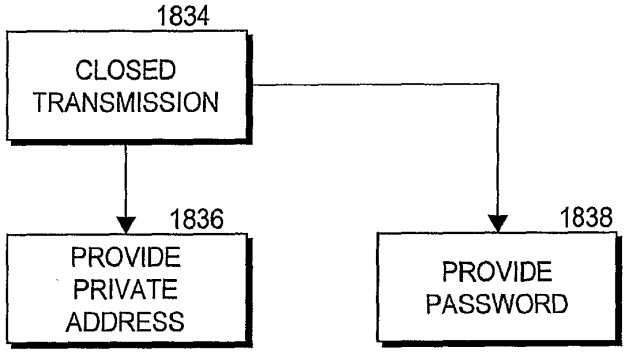


Fig. 33

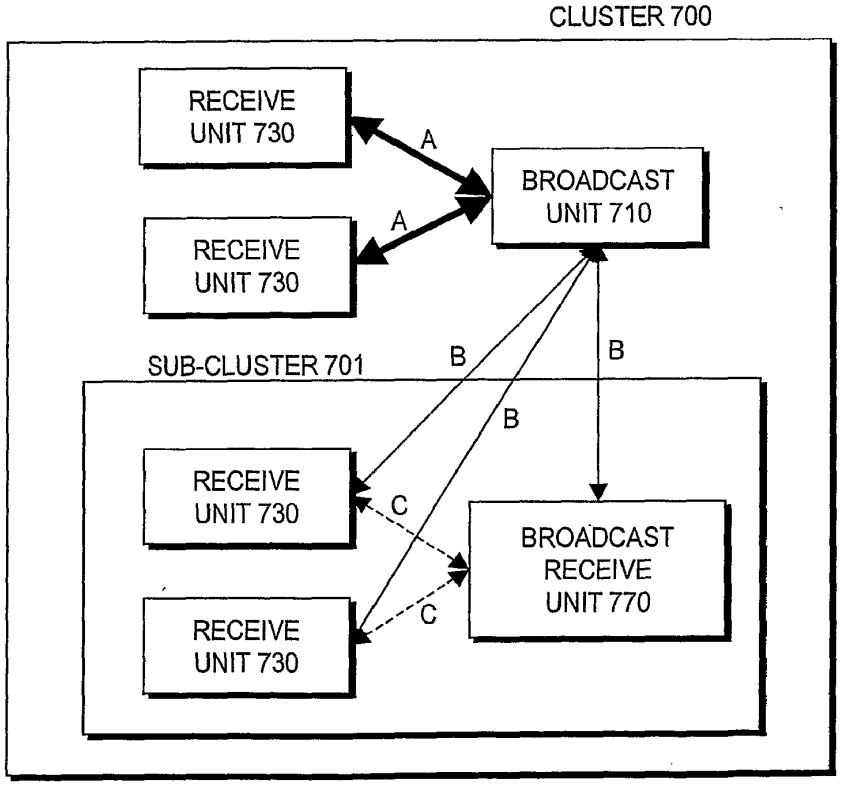


Fig. 34A

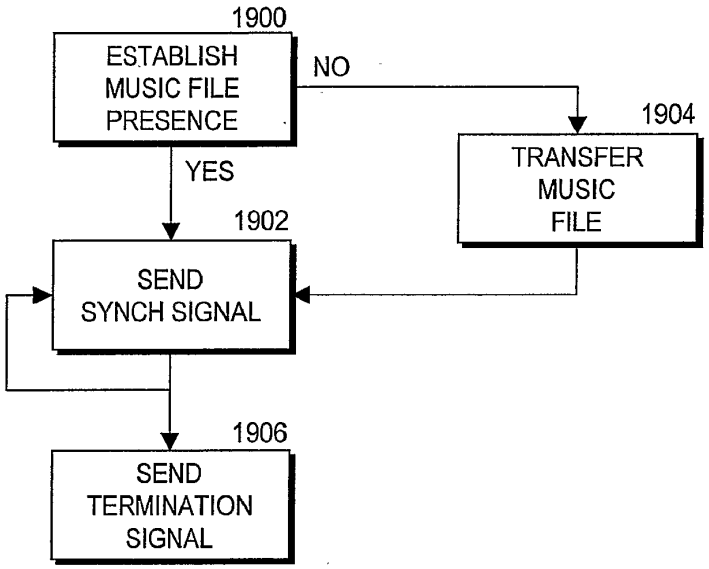
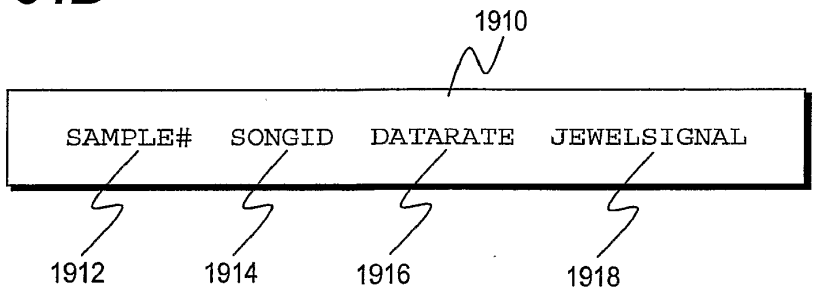


Fig. 34B



PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/30 | A1

(11) International Publication Number: **WO 95/25313**

(43) International Publication Date: 21 September 1995 (21.09.95)

(21) International Application Number: PCT/EP94/00833

(22) International Filing Date: 16 March 1994 (16.03.94)

(71) Applicant *(for all designated States except US)*: INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; Old Orchard Road, Armonk, NY 10504 (US).

(72) Inventors; and

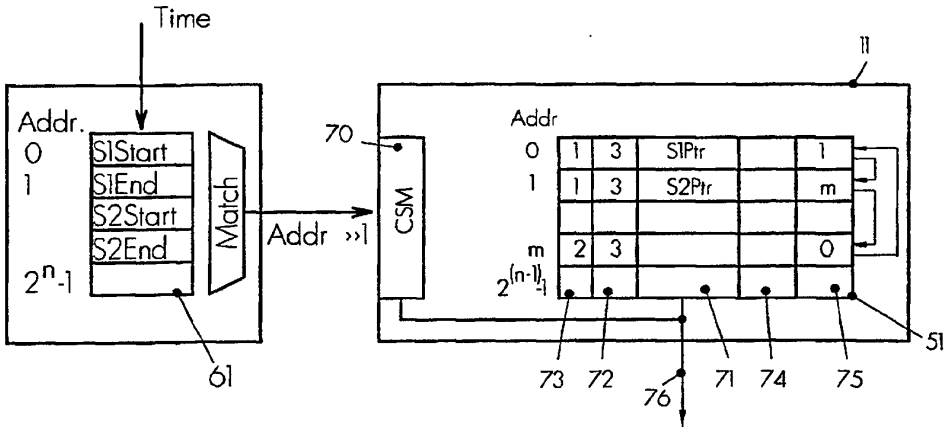
(75) Inventors/Applicants *(for US only)*: KAISERSWERTH, Matthias [DE/CH]; Leigrubenstrasse 2, CH-8805 Richterswil (CH). RÜTSCHE, Erich [CH/FR]; 1500, chemin des Adrets, F-06370 Mouans-Sartoux (FR).

(74) Agent: BARTH, Carl, O.; IBM Research Laboratory, Intellectual Property Dept., Säumerstrasse 4, CH-8803 Rüschlikon (CH).

(81) Designated States: JP, US, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: METHOD AND APPARATUS FOR SYNCHRONIZATION AND SCHEDULING OF MULTIPLE DATA STREAMS AND REAL TIME TASKS



(57) Abstract

The present application concerns a method and an apparatus for the synchronization and the scheduling of multiple data streams and real time tasks. In accordance with the present invention, pointers pointing to the information streams to be synchronized and processed are kept in the table (51) of a selector (11). The time marks (S1Start etc.) of said information streams are stored in a table (61) of a timer. The system time is checked against said time marks and if a time mark has been reached, the table (51) of said selector (11) is updated. If the status of an information stream is such that its trigger condition is fulfilled, the pointer to said information stream is made available at the output of the selector (11). This pointer is then used to retrieve the respective information stream and to forward it to a playout device.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	**GB**	United Kingdom	**MR**	Mauritania
AU	Australia	**GE**	Georgia	**MW**	Malawi
BB	Barbados	**GN**	Guinea	**NE**	Niger
BE	Belgium	**GR**	Greece	**NL**	Netherlands
BF	Burkina Faso	**HU**	Hungary	**NO**	Norway
BG	Bulgaria	**IE**	Ireland	**NZ**	New Zealand
BJ	Benin	**IT**	Italy	**PL**	Poland
BR	Brazil	**JP**	Japan	**PT**	Portugal
BY	Belarus	**KE**	Kenya	**RO**	Romania
CA	Canada	**KG**	Kyrgystan	**RU**	Russian Federation
CF	Central African Republic	**KP**	Democratic People's Republic of Korea	**SD**	Sudan
CG	Congo			**SE**	Sweden
CH	Switzerland	**KR**	Republic of Korea	**SI**	Slovenia
CI	Côte d'Ivoire	**KZ**	Kazakhstan	**SK**	Slovakia
CM	Cameroon	**LI**	Liechtenstein	**SN**	Senegal
CN	China	**LK**	Sri Lanka	**TD**	Chad
CS	Czechoslovakia	**LU**	Luxembourg	**TG**	Togo
CZ	Czech Republic	**LV**	Latvia	**TJ**	Tajikistan
DE	Germany	**MC**	Monaco	**TT**	Trinidad and Tobago
DK	Denmark	**MD**	Republic of Moldova	**UA**	Ukraine
ES	Spain	**MG**	Madagascar	**US**	United States of America
FI	Finland	**ML**	Mali	**UZ**	Uzbekistan
FR	France	**MN**	Mongolia	**VN**	Viet Nam
GA	Gabon				

DESCRIPTION

Method and Apparatus for Synchronization and Scheduling of Multiple Data Streams and Real Time Tasks

TECHNICAL FIELD

The present invention concerns a method and apparatus for the synchronization and the scheduling of multiple data streams as well as for the scheduling of tasks in operating systems with hard real-time requirements.

BACKGROUND OF THE INVENTION

Multimedia information is composed of several objects that constitute a document. These objects could be of different data types, such as text, still images, video, and sound. In order for these objects to be meaningful to a user, there must be a logical and temporal structure of the objects constituting the documents' contents. To ensure this logical and temporal structure, e.g. when displaying a video sequence on a display device and providing a soundtrack for this video sequence via an adapter card to a speaker, the respective data streams have to be synchronized and scheduled before being sent to the output devices. The same principle applies to other related multimedia information which has to be synchronized in communicating with other devices or users (e.g. teleconferencing), before processing and presentation.

Synchronization and scheduling is a critical task for a wide spectrum of applications ranging from computer-integrated telephony, text with voice annotations, any kind of multi media systems, and interactive services (e.g. interactive television) to cooperative teleworking on documents including

foils and video, and distributed games. A further application, where information streams have to be synchronized, is known as medical imaging, for which high-resolution images must be distributed reliably and fast throughout a hospital.

To date, the synchronization is typically performed in software. However, if the scheduling intervals are too short, or if too many information streams must be synchronized, a software solution, which is very CPU intensive, is often the limiting factor as far as processing speed is concerned.

In real-time systems time-critical tasks have to be scheduled reliably and fast. In the present context, any system in which the time at which an output is produced is significant, is referred to as real-time system. The lag from the input time of a real-time system to its output time, i.e. the delay, must be sufficiently small for acceptable timeliness. With other words, a real-time system has to respond to an externally-generated input stimuli within a finite and specified period. In hard real-time systems, i.e. in those systems where it is absolutely imperative that responses occur within a specified deadline, scheduling of time-critical tasks is of particular interest. However, the present invention is not limited to hard real-time systems. It might be used in soft real-time systems, too. Such a soft real-time system would still function correctly if deadlines are occasionally missed, and a hardware solution in accordance with the present invention might be to costly under certain circumstances.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a method and apparatus for off-loading the synchronization of multiple information streams from the processor of the system handling these information streams.

It is an object of the present invention to provide a method and apparatus for fast and reliable synchronization and scheduling of multiple information streams.

It is another object of the present invention to provide a method and apparatus for scheduling tasks or processes in real-time systems.

The above objects have been accomplished by handling pointers pointing to the information streams to be synchronized and by maintaining a table with status information of each information stream. The time marks of said information streams are stored in a table and the system time is checked against said time marks. If a time mark has been reached the status information of the respective information stream is updated. If the status of an information stream is such that its trigger condition is fulfilled, the pointer to said information stream is made available for playout of this information stream.

DESCRIPTION OF THE DRAWINGS AND NOTATIONS USED

The invention is described in detail below with reference to the following drawings:

FIG. 1 shows a system, in accordance with the present invention, for playout of two information streams.

FIG. 2 illustrates the synchronization of multiple data streams.

FIG. 3 illustrates the synchronization and play out of two data streams when the respective trigger condition is fulfilled.

FIG. 4 illustrates how a priority value might be increased with advancing time.

FIG. 5 shows the two core components of an apparatus for synchronization and scheduling of data streams and real time tasks, in accordance with the present invention.

FIG. 6A shows a first exemplary embodiment of the timer.

FIG. 6B shows a second exemplary embodiment of the timer.

FIG. 7 shows an embodiment of the timer and selector, in accordance with the present invention.

FIG. 8 shows an example of an 8-bit timer where every second bit is used.

GENERAL DESCRIPTION

In multimedia systems multiple data streams must be synchronized and scheduled for playout to, for example, a speaker 15 and a video display 14, as illustrated in Figure 1. The synchronization of multiple data streams is illustrated in Figure 2. The three streams are synchronized at so-called synchronization points (SyncPoint) 22. A stream may be synchronized to one or multiple other streams or to time stamps 23 defined by an external clock. This synchronization is typically performed in software, as mentioned in the introductory portion of the present description. In the following, a mechanism is presented which performs synchronization and output scheduling of multiple data streams in hardware without central processing unit (CPU) interaction. The same mechanism can also be used for scheduling tasks in real-time systems.

A data stream is a sequence of data units. In Figure 3, the synchronization of two data units is illustrated. The SyncPoints are translated to time marks relative to the system which processes the received data streams. The SyncPoints might be translated to time marks by means of a clock or counter of said system. In the example given in Figure 3, a data unit from stream 1 cannot be played out before a time mark 30 (S1Start) and it must be played out before a time mark 31 (S1End). The same applies to the second data stream (stream 2). As illustrated in Figure 3, at least two different states (WAIT, READY) can be associated with a data stream. Before SxStart of a data stream x has been reached, the respective data unit is in the WAIT state. After SxStart and before SxEnd it is in the READY state. It is conceivable to assign further states, such as a DELAYED state, to these data streams. The DELAYED state might be associated to a data unit after SxEnd.

In order to be able to synchronize two data units for playout, trigger conditions are defined in accordance with the present invention. A typical trigger condition might be:

IF both data units are in the READY state
THEN they can be played out.

The trigger conditions are checked and as soon as the condition is fulfilled, the data units are played out. This concept is schematically illustrated in Figure 3, where means for checking the trigger condition 32 are shown.

In general, two or more date units can be related in time by trigger conditions, which can be an arbitrary logical combination of the different states of the respective data units. Fulfilling a given trigger condition then initiates the appropriate operation on the units.

The synchronization can be further enhanced with a priority value P to distinguish between data units (or tasks) which fulfill a trigger condition to select an older or more important stream (or task). This, for example, allows to prioritize streams that should not be delayed, whereas other data streams might be delayed. In Figure 4 an example is shown how the priority value P of a data unit (or task) can be increased in time. In this example, three different states (WAIT, READY, DELAYED) are associated with the data stream. The priority value P might also be increased or decreased by interaction of a processor or upon receipt of a feedback signal of a playout device.

A first embodiment of the present invention is schematically illustrated in Figure 1. According to this example, two data streams are received by means 18 for extraction of time marks. The first data stream originates from a storage disk 16, whereas the second data stream is sent via a network 17 to said means 18 for extraction of time marks. After extraction of the time marks, the data units of the two data streams are stored in buffers 19 and 20, and the respective time marks are provided to a timer 10. The buffer addresses are forwarded to a selector 11. In addition, a system clock or counter 13 is connected to said timer 10, and a processor, e.g. a CPU, might

be linked to the timer 10 and selector 11. Two playout devices, a display unit 14 and a loudspeaker 15, are connected via an interfacial playout device to the buffers 19 and 20.

The two core components of the present invention are the timer 10 and the selector 11. Their operation is described in connection with Figure 5.

The timer 10 holds a table 50 with entries concerning the time marks of each data unit received and stored in a buffer. These time marks are relative to a time signal either being generated by a system clock 13 or a counter. Each time mark reached is then output to the selector 11. The table 50 is written by an external processor. The selector comprises a table 51 where for each data unit in a buffer a pointer to said data unit, i.e. the respective buffer address, the state of said data unit, and the trigger condition are held. Data units to be synchronized are linked into a circular list via their table entries. For each time mark received from the timer 10, the sector 11 updates the status field of the concerned data unit and then checks the trigger condition of all data units to be synchronized with said data unit. Once the trigger condition is fulfilled, the pointer to the data unit (or units) is made available at the output of selector 11. The output of the pointer then triggers retrieval and playout of the data unit by means of the playout device 21. Table 51 of the selector 11 might be maintained and updated by an external processor, e.g. CPU 12.

The timer 10 reads the current system time and returns the address of a unit pointer in the selector 11. This function can be provided in several ways. First, a solution based on a content addressable memory (CAM) is presented which is suitable for a high resolution timer but relatively expensive. The second solution is less expensive and is suitable for lower resolution timers and for implementing timer priority as described below. The two solutions are shown in Figures 6A and 6B, respectively.

The CAM timer 60 is illustrated in Figure 6A. It comprises a CAM 61. The start and end time mark for each data unit to be scheduled are written into consecutive addresses of the CAM 61. The addresses then differ only in the least significant bit. Thus, for a CAM of 2^n addresses, n-1 bits determine the address of the scheduling unit pointer in the selector table 51. The width of a CAM entry is determined by the resolution of the time signal received at input 62. For a 32-bit time signal also a 32-bit wide CAM 61 is required.

For lower time resolution a timer 63 with counter 64, as shown in Figure 6B, might be employed. In this case the difference between the current time and the time mark is calculated and its two's complement is written to the timer table 65. The counter 64 increments each entry for each time tick at input 66. If the carry bit of an entry becomes set, its address is shifted right by one bit and output to the selector 11. As the counter 64 continues to increment the entries, the content of the entry (with its carry bit = 1) can then be used to reflect the time passed since reaching the start and/or end time. The counter value thus describes a time based priority. With this counter value a priority value P might be generated. The width of a timer entry can range from 8 to 32 bits depending on the timer resolution and the required timer differences. A free flag 67 might be provided in order to mark free table entries in table 65. This flag 67 is used by the counter to determine which entries it must increment.

For both schemes, a FIFO can hold free timer table entries to reduce the management load of the external processor. The processor gets a free entry from the FIFO and writes delayed entries, or entries which are freed, to the FIFO. The selector might directly write delayed entries to this FIFO.

It is obvious that the present invention is not limited to the above two implementations of timers since other timers are conceivable which are well suited, too.

The selector 11 comprises a table 51 and a comparator state machine (CSM) 70. The table 51 has $2^{(n-1)}$ entries (rows or columns). The exemplary implementation of the selector 11, given in Figure 7, shows a table 51 with $2^{(n-1)}$ rows. Each row has a pointer field 71 holding the pointer, i.e. the buffer address S1Ptr, to the data unit to be scheduled, a p-bit wide priority field 72, a k-bit wide status field 73, a 2-bit wide trigger field 74, and an (n-1)-bit wide synchronization field 75. The width of said status field 73 depends on the number of different states. In case of three different state (WAIT, READY, DELAYED) the status field 73 is 2-bits wide. The synchronization field 75 holds a pointer linking all those entries with which a data unit must be synchronized into a circular list. This circular list is schematically illustrated by means of arrows on the right hand side of table 51. The CSM 70 receives the address of a unit pointer from the timer 10 and updates the state of the status field 73 in this row. The CSM 70 then follows the linked list of synchronizations entries and computes whether the trigger condition is fulfilled by comparing the status fields 73 with the trigger fields 74. The trigger condition might also be checked by means of simple logical elements such as AND-gates, OR-gates etc.

In the following, it is assumed, without loss of generality, that the trigger condition is fulfilled when all entries in the respective trigger fields 74 are in the READY state. In this case, the trigger state fields are redundant and might be omitted.

Once all the entries linked by the synchronization field 75 are READY, the CSM 70 outputs the entries according to their priorities, if any. With $p=4$, 16 priority classes can be distinguished. The width w of the scheduling unit pointer is determined by the memory (buffer) architecture of the system. The width s of the synchronization field 75 is determined by the size of the selector table 51, with $s=n-1$.

In Figure 7 the problem shown in 3 is elaborated. The time marks 30 (S1Start) and 31 (S1End) of data unit S1 are written to the addresses 0 and 1

of the CAM 61. The address of the pointer to the data unit S1 is the CAM address shifted right by one bit. An external processor 12 writes the time marks to the timer table, i.e. the CAM 61 and the entries to the selector table 51. For a data unit to be scheduled which requires priority, the priority field 72 is set. In this example a 4-bit wide priority field 72 is used. For each time mark reached, the state, reflected by the status field 73, is updated. If the state is READY, the CSM 70 reads the synchronization field 75 and follows the linked list to determine whether the state of all other entries fulfills the trigger condition. If the trigger condition is reached, the unit pointer with the highest priority is made available at output 76 first. If multiple units have the same priority, or in case that no priority fields are provided, the unit pointers might either be output in parallel or in a consecutive manner, e.g. by order of their position in the selector table 51. It is obvious that the priority fields 72 can be omitted, too.

In many systems, multimedia as well as real-time, the scheduling of actions, data streams, tasks, processes, or the playout of data units is determined by a priority which increases with time. If a counter timer, as illustrated in Figure 6B, is used as the timer component 10, the entry in the timer table 65 also describes its priority. This priority can be used in the priority field 72 of the selector 11 to achieve time dependent scheduling. The four most significant bits of the timer entry are used to distinguish again 16 priorities. A different set of bits might be used if the priority must increase non-linearly. Figure 8 shows an example of an 8-bit timer where every second bit is used to get a priority mechanism that maps the 256 timer values to 16 non-linear priority values P.

According to the present invention, a synchronization point (SyncPoint) is set up by writing the time marks to the free entries in the timer table and setting up the associated entries in the selector table. For each stream this takes one write operation to the timer and to the selector table, plus the operations to set up the selector table entry. The actual scheduling takes place in hardware without any further intervention of an external processor.

For a solution in software a similar number of setup operations would be necessary but the search in a list of timers would take much longer. Typically timers involve interrupt processing which is very time consuming in particular in modern RISC processors with a large register file and instruction pipeline. In the present invention these operations are avoided and implemented in hardware. Additionally, the present invention can handle timers with much finer granularity than any software solution.

The present invention, which has been described by means of a multimedia related example, can be used in various multimedia communication subsystems, e.g. for multimedia multiparty teleconferencing, or multimedia protocol adapters. In ST-II and RTP protocol systems, for example, a playout time stamp can be extracted from the protocol header and used in the present scheduler (ST-II is the second version of an experimental Internet stream protocol, and RTP is a transport protocol for real-time applications). The present invention might also be used in multimedia bus systems which provide a time signal. Data units to be triggered at a specified time are written to the scheduler and the multimedia bus provides the timer input to the timer unit. The invention can also be used in a multimedia adapter such as the IBM ActionMedia II adapter (see IBM ActionMedia II Technical Reference, IBM 04G5144, 2nd Edition, June 1992, Mechanicsburg), or the MWave System (see "MWave Multimedia System", Technical Brief, Texas Instruments, 1992) to synchronize multiple data streams. The term ActionMedia is a trademark of the Intel Corporation and MWave is a trademark of the IBM Corporation.

As already mentioned in the introduction, the present invention might also be used in real-time systems for the scheduling of time-critical tasks or processes. For these kind of application there is no synchronization field in the table of the selector required. The CSM selects from the selector table the entry with the highest priority. The scheduling unit pointer, i.e. a pointer pointing to a task control block or to executable code, is then output. The timer priority mechanism described above can be used to implement

scheduling policies with an explicit deadline, e.g. the earliest-deadline-first or least-slack time scheduling algorithm as disclosed in the book "Real-Time Systems and their Programming Languages", by A. Burns and A. Wellings, Addison-Wesley Publishing Company, International Computer Science Series, 1990. An external processor calculates the difference to the time marks and writes it to the timer table. The entries are then used to derive the priority of a job.

CLAIMS

1. Apparatus for synchronization of at least two information streams which comprises:
   a) a timer (10) for holding a first table (50) with time marks for each one of said information streams, and for making each time mark reached available at the timer output,
   b) a selector (11) for holding a second table (51) with
      - pointer fields (71) carrying pointers to each of said information streams,
      - status fields (73) carrying information concerning the status of each of said information streams, the status field of an information stream being updated each time a time mark of said information stream is reached and made available at said timer output,
   c) means for checking whether a trigger condition of an information stream is fulfilled by watching the corresponding status field, and
   d) means for making the pointer to an information stream whose trigger condition is fulfilled available for playout of said information stream.

2. The apparatus of claim 1, wherein said second table (50) further comprises:
   - trigger fields (74) carrying information concerning the trigger conditions of each of said information streams.

3. The apparatus of claims 1 or 2, wherein said second table (51) further comprises:
   - synchronization fields (74) carrying, by means of a linked list, entries concerning information streams to be played out in a certain relationship to each other.

4. The apparatus of claims 1, 2, or 3, wherein said second table (51) further comprises:
   - priority fields (72) carrying priority values (P) such that the information stream with highest priority is played out first if the respective trigger condition has been fulfilled.

5. The apparatus of claim 4, further comprising means for increasing said priority values (P) with advancing time.

6. The apparatus of any of the preceding claims, further comprising:
   - means (18) for extracting said time marks from incoming information streams,
   - buffer means (19, 20) for storing said information streams,
   - playout means (21) for retrieval of said information streams stored in said buffer means (19, 20) and for playout to the respective playout devices (14, 15).

7. A multimedia system comprising an apparatus in accordance with any of the preceding claims.

8. Apparatus for scheduling at least two real time tasks or two processes which comprises:
   a) a timer (10) for holding a first table (50) with time marks for each one of said tasks or processes, and for making each time mark reached available at the timer output,
   b) a selector (11) for holding a second table (51) with
      - pointer fields (71) carrying pointers to task control blocks or executable code,
      - means for making a pointer to a task control block or an executable code available at an output of said selector each time a time mark of said task or process is reached in said timer.

9. The apparatus of claim 8, wherein the pointer to a task or process with highest priority is made available first.

10. Method for synchronization of at least two information streams, characterized by the following steps:
    a) checking whether a time mark of one of said information streams has been reached,
    b) updating the status information of said information stream whose time mark has been reached,
    c) determining whether the trigger condition of said information stream is fulfilled by watching said status information, and
    d) providing a pointer to said information stream for further processing if the trigger condition of said information stream is fulfilled.

11. The method of claim 10, wherein the playout of information streams, which have to be played out in a predefined relationship to each other, is scheduled by means of a linked list being organized such that each time when a status information of one of said information streams is updated, the status information of the other information streams are checked, and said information streams played out when all trigger conditions are fulfilled.

12. The method of claim 10, wherein a priority value (P) is assigned to each of said information streams, the information streams with highest priority being played out first.

13. The method of claim 12, wherein the priority value (P) of an information stream is increased with advancing time.

14. The method of claim 10, further comprising the following steps:
    - extracting and/or synchronizing the time marks of said information streams,
    - storing said information streams in buffers,

- making the addresses of said buffers available as pointers for later retrieval of said information streams,

15. The method of claim 14, wherein said information stream is retrieved from said buffer by means of said pointer, prior to being send to a playout device.

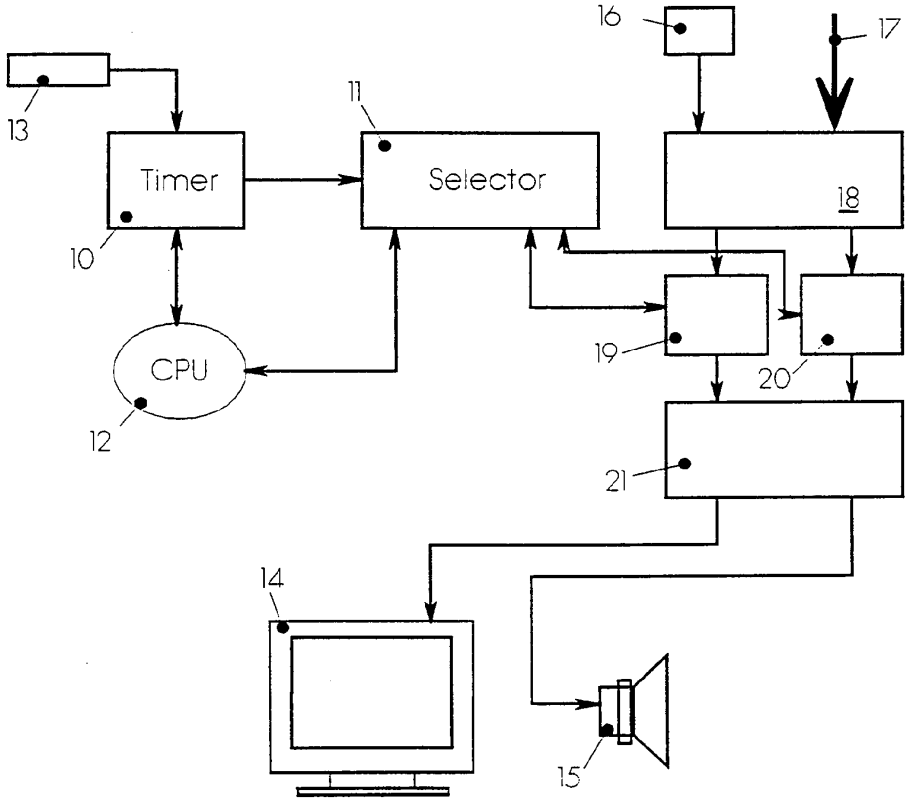

16
17
13
11
Timer
Selector
18
10
CPU
19
20
12
21
14
15

FIG. 1

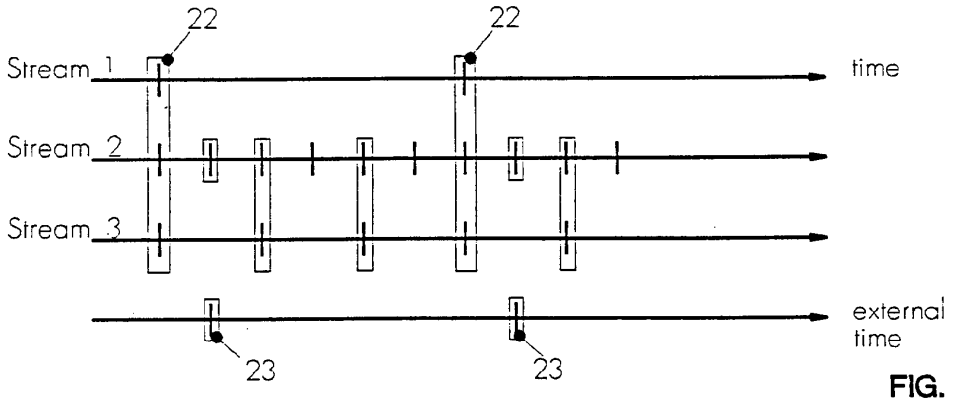

22
22
Stream 1
time
Stream 2
Stream 3
external time
23
23

FIG. 2

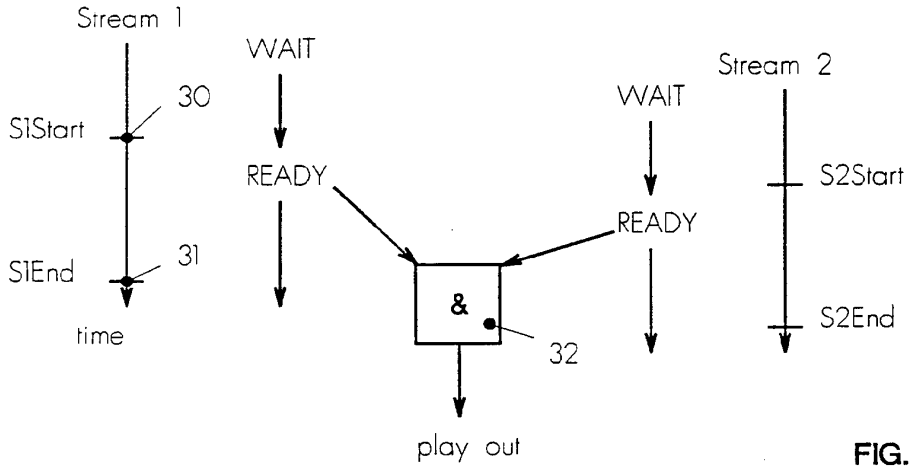


FIG. 3

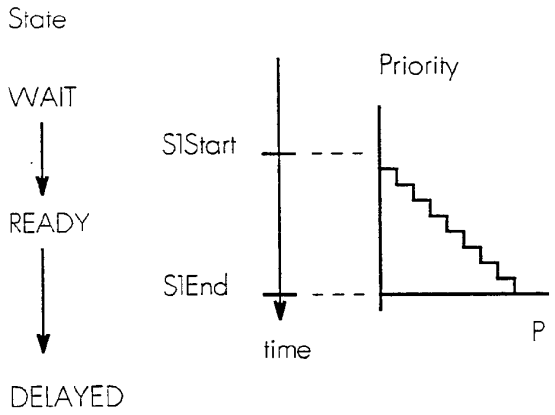


FIG. 4

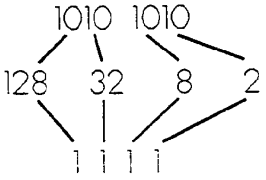


FIG. 8

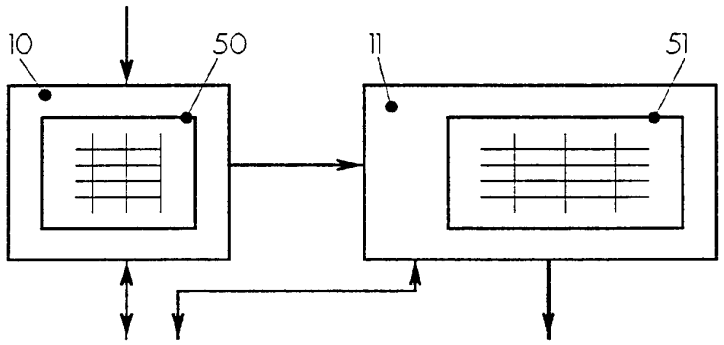


FIG. 5

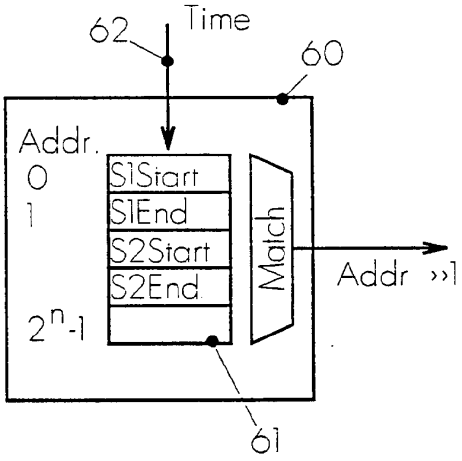


FIG. 6A

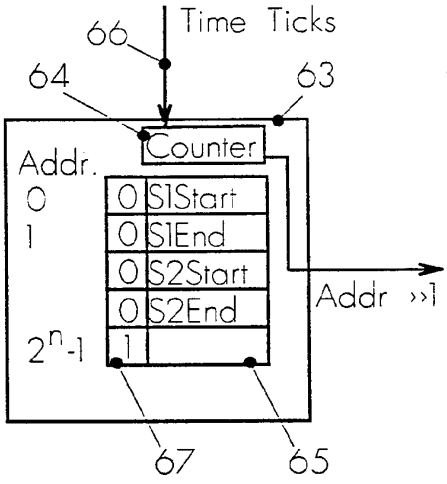


FIG. 6B

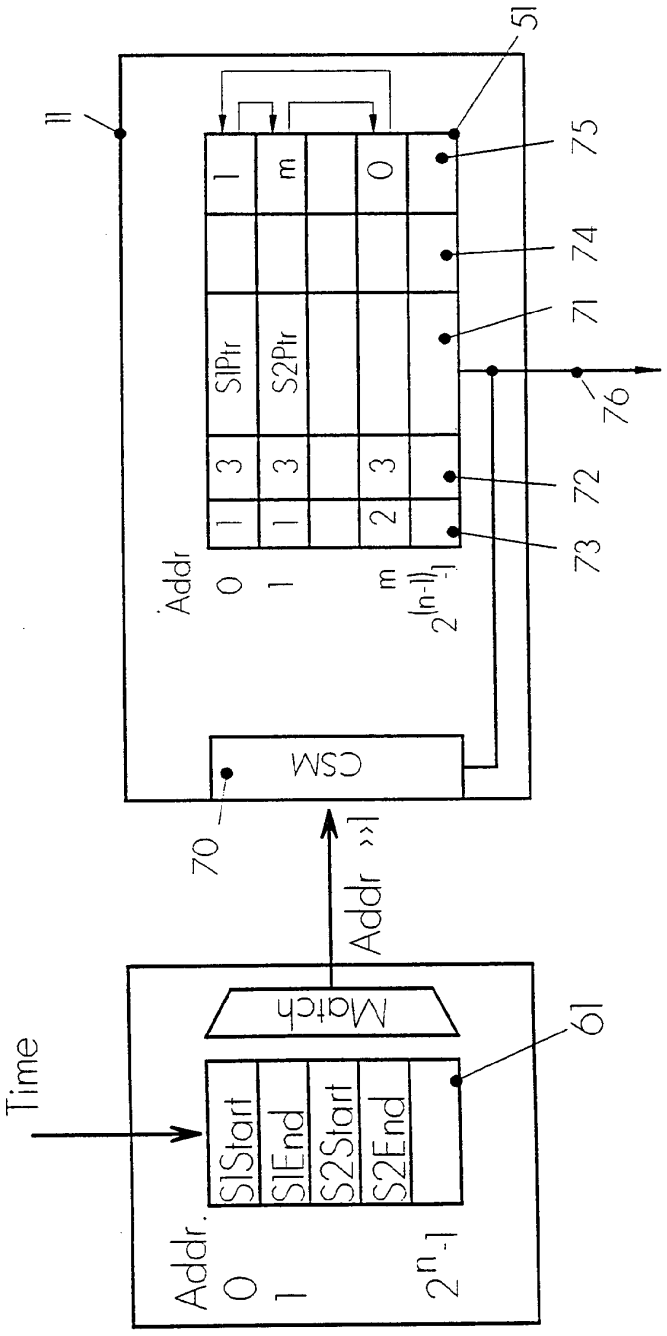

11
51
75
74
71
76
72
73
S1Ptr
S2Ptr
1
m
0
3
2
Addr
0
1
m
$2^{(n-l)}-1$
CSM
70
Addr >>l
Match
Time
61
S1Start
S1End
S2Start
S2End
Addr.
0
1
$2^{n}-1$

FIG. 7

INTERNATIONAL SEARCH REPORT

Intern ..al Application No
PCT/EP 94/00833

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP,A,0 239 884 (I.B.M. CORPORATION) 7 October 1987 see the whole document ---	1,8,10
A	EP,A,0 550 197 (I.B.M. CORPORATION) 7 July 1993 see the whole document ---	1,8,10
A	COMPUTER COMMUNICATIONS, vol.16, no.3, March 1993, GUILDFORD GB pages 168 - 176 P.V. RANGAN ET AL. 'Techniques for multimedia synchronization in network file systems' see the whole document --- -/--	1,8,10

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 November 1994	01.12.94

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+ 31-70) 340-3016	Katerbau, R

Form PCT/ISA/210 (second sheet) (July 1992)

LINKPLAY EXHIBIT 1004
Page 3043 of 4822

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	COMPUTER, vol.24, no.10, October 1991, LONG BEACH US pages 51 - 57 D.P. ANDERSON ET AL. 'A continuous media I/O server and its synchronization mechanism' see the whole document ---	1,8,10
A	BYTE, vol.18, no.2, February 1993, LONDON, GB pages 225 - 230 A.W. DAVIS ET AL. 'The Mac goes to the movies' see the whole document -----	1,8,10

1

LINKPLAY EXHIBIT 1004
Page 3044 of 4822

INTERNATIONAL SEARCH REPORT

Information on patent family members

Intern. ıal Application No
PCT/EP 94/00833

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
EP-A-0239884	07-10-87	US-A-	4893256	09-01-90
		DE-D-	3787553	04-11-93
		DE-T-	3787553	11-05-94
		JP-B-	6054469	20-07-94
		JP-A-	62239233	20-10-87
EP-A-0550197	07-07-93	US-A-	5333299	26-07-94
		CA-A-	2084574	01-07-93
		JP-A-	5265778	15-10-93

Form PCT/ISA/210 (patent family annex) (July 1992)

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) **International Patent Classification** 6 :
G06F 9/46, H04L 12/28, 12/56, 12/40

A1

(11) **International Publication Number:** **WO 99/23560**

(43) **International Publication Date:** 14 May 1999 (14.05.99)

(21) **International Application Number:** PCT/US98/23596

(22) **International Filing Date:** 4 November 1998 (04.11.98)

(30) **Priority Data:**
60/064,266 4 November 1997 (04.11.97) US

(71) **Applicant** *(for all designated States except US)*: COLLABORATION PROPERTIES, INC. [US/US]; Suite 7, 913 Tahoe Boulevard, Incline Village, NV 89451 (US).

(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only)*: LUDWIG, Lester [US/US]; 1230 Southdown Road, Hillsborough, CA 94010 (US). BROWN, William, Blake [US/US]; 218 Jo Drive, Los Gatos, CA 95032 (US). YUL, Inn, J. [US/US]; 22281 Bellevue, Cupertino, CA 95014 (US). VUONG, Anh, T. [AU/US]; 1180 Reed Avenue #47, Sunnyvale, CA 94086 (US). VANDERLIPPE, Richard, W. [US/US]; Apartment 7, 1247 Fulton Street, San Francisco, CA 94117 (US). BURNETT, Gerald [US/US]; 207 Atherton Avenue, Atherton, CA 94027 (US). LAUWERS, Chris [BE/US]; 350 Sharon Park Drive #H210, Menlo Park, CA 94025 (US). LUI, Richard [CN/US]; 4624 Norwalk Street, Union City, CA 94587 (US). APPLEBAUM, Daniel [US/US]; 307 Winding Way, Incline Village, NV 89450 (US).

(74) **Agents:** OPPERMAN, Craig, P.; Cooley Godward LLP, 3000 El Camino Real, Five Palo Alto Square, Palo Alto, CA 94306–2155 (US) et al.

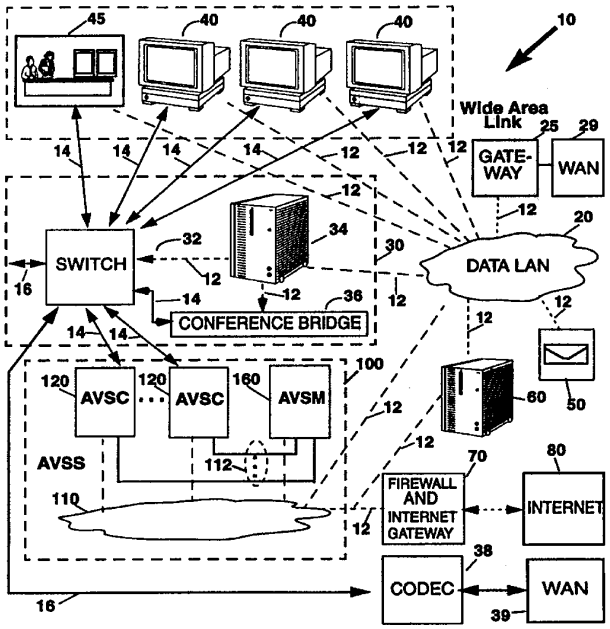
(81) **Designated States:** AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) **Title:** SCALABLE NETWORKED MULTIMEDIA SYSTEM AND APPLICATIONS

(57) **Abstract**

A networked multimedia system (10) comprises a plurality of networks (40) and at least one storage server (100). A signal path interconnects the workstations (12) and the storage server (100). Each workstation (40) includes video and audio reproduction capabilities, as well as video and audio capture capabilities. Any given storage server (100) comprises a set of storage cells (120) that operate under the direction of a storage cell manager (160). A storage cell (120) may include one or more encoding (132) and transcoding converters configured to convert or transform audio and video signals originating at a workstation into a form suitable for storage. A storage cell (120) may further include one or more decoding converters (134) configured to convert stored signals into a form suitable for audio and video signal reproduction at a workstation. Each storage cell (120) additionally includes at least one storage device (150) and storage device controller (152) capable of storing, for later retrieval, signals generated by one or more converters. The storage cell controller responds to signals received from the workstations (40), and oversees the operation of the storage cells to facilitate the storage of converted audio and video signals in at least one file that can be simultaneously accessed by one or more application programs executing on one or more workstions



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

SCALABLE NETWORKED MULTIMEDIA SYSTEM AND APPLICATIONS

1. BACKGROUND OF THE INVENTION

1.1 Field of the Invention

The present invention relates generally to a scalable networked multimedia system, and more particularly to a scalable audio-video server system and Application Program Interface (API) together with a range of associated software applications that together provide high-quality audio-video and multimedia processing capabilities.

1.2 Background

In recent years, considerable effort has been directed toward the development of hardware and software for network-based audio-video (A/V) and, more generally, networked multimedia systems. Such development has been driven by technology push from equipment manufacturers, as well as the commercial potential of entertainment applications such as video-on-demand and multi-player gaming; and business applications such as video messaging and multimedia conference collaboration.

A critical factor impacting the usefulness and value of a network-based multimedia system is the manner in which computational resources are organized to define a video file storage/encoding/decoding/distributionsystem, hereafter referred to an Audio-Video Storage System (AVSS). Several key requirements exist with regard to defining a Audio Video Storage System applicable to business applications. With particular emphasis on deployment in a business environment supporting multimedia conference collaboration and characterized by on-premises (i.e., local) and wide-area networks, these requirements include the following:

1) ubiquitous premises scalability;
2) ubiquitous wide-area scalability;
3) limited impact upon network loading;
4) low implementation and operational costs;
5) accommodation of multiple desktop platforms (modern, existing, and outdated);
6) accommodation of multiple compression standards;

LINKPLAY EXHIBIT 1004
Page 3048 of 4822

7) support for a wide range of high-performance, high-quality video-enabled applications;

8) cost-effective ability to upgrade across successive technology and standards generations; and

9) API extensibility.

A collaborative multimedia environment employing a video storage system necessarily includes a plurality of desktop workstations, encoding and decoding resources, video file storage resources, and a premises video distribution network. Two architectural design factors greatly influence the extent to which the aforementioned key requirements can be simultaneously met, namely, 1) the organization of the encoding/decoding resources and video file storage resources relative to the desktop workstations and video distribution network; and 2) the nature of the video distribution network itself.

Encoding and decoding resources, as well as video file storage resources, can be allocated on a desktop-by-desktop basis, or a network (i.e., shared) basis. ***Figure*** 1 illustrates an exemplary Erlang resource sharing utilization relationship 2 showing computational resource utilization efficiency 1 relative to the number of users sharing the resource 6 under fixed blocking conditions. As indicated in ***Figure*** 1, network-based resource allocation 3, 4 results in much higher resource utilization efficiency than desktop-dedicated resources, which are fully allocated to one user each (5). This is turn implies that video storage systems characterized by desktop-based resource allocation leverage technology investments for less effectively than systems that employ resource sharing, particularly in situations involving a significant number of workstation users 3 and/or relatively low usage rates ($<20\%$ of workday) by users. Additionally, desktop-based resource allocation undesirably results in greater system upgrade costs.

A video distribution network can be based upon analog technology, digital technology, or their combination. Figure 2 is a graph showing the relative cost 2.1 of local-area analog and digital video signal distribution technology as a function of video quality 2.2, and hence includes an analog cost-versus-performance/quality curve 2.3 and a digital cost-versus-performance/quality curve 2.4. Across a first, lower-performance and lower-quality region 2.5 in ***Figure*** 2 that could provide performance and quality suitable,

LINKPLAY EXHIBIT 1004
Page 3049 of 4822

for example, for technology experiments, digital signal distribution technology is significantly more expensive than its analog counterpart. The slope of the digital cost-versus-performance/quality curve 2.4 throughout this first region 2.5 is nearly constant, while that for analog cost-versus-performance/quality 2.3 gradually rises with increasing performance and image quality 2.2. A second region 2.6 shown in ***Figure*** 2 spans a practical region of operation 2.7 relative to business-performance and business-quality levels, which herein correspond to video delivered at 30 frames per second (fps) at a resolution ranging from approximately 320 x 240 to 640 x 480 pixels or other standard resolution. Across the second region 2.6, the analog cost-versus-performance/quality curve 2.3 begins to increase rapidly as performance and quality 2.2 improves, eventually meeting and exceeding the digital cost-versus-performance/quality curve 2.4. However, throughout most of the aforementioned practical region of operation 2.7, analog signal distribution technology remains significantly less expensive than its digital counterpart. Thus, for most business environment performance and quality requirements, analog signal distribution technology is more cost-effective than digital signal distribution technology. Readily-available digital network technology lacks sufficient bandwidth for delivering business-quality, real-time or near-real-time video to a large number of users. Finally, a third region 2.8 in ***Figure*** 2 spans high-end or special-situation performance and quality levels 2.2. Within the third region 2.8, the cost of digital signal distribution technology begins to rapidly escalate.

Figure 2 additionally indicates the manner in which the analog and digital cost-versus-performance/quality curves 2.3, 2.4 can be expected to evolve over time. For each technology type, overall cost 2.1 will decrease relative to a given performance and quality level 2.2 as the technology evolves. The general shape of the curves 2.3, 2.4 shown in ***Figure*** 2, however, can be expected to remain essentially the same. Moreover, digital signal distribution technology is likely to evolve at a much more rapid pace than analog distribution technology in the near term, which implies higher costs 2.1 over a system's lifetime due to system upgrade frequency. Thus, video storage systems that rely upon all-digital video distribution technology are significantly less cost-effective than those employing analog distribution technology.

LINKPLAY EXHIBIT 1004
Page 3050 of 4822

Known premises-based networked video storage systems fail to come anywhere close to meeting the aforementioned key requirements. Much of the reason for this results from the design approaches taken relative to the aforementioned architectural considerations, particularly when the architectural cornerstones are driven by established technology marketing trends rather than designing to meet true business requirements. What is needed is a video storage system that utilizes resource sharing and the full evolvable range of networked signal distribution technology to meet key cost and application quality requirements described above.

2. SUMMARY OF THE INVENTION

The present invention is a networked multimedia system comprising a plurality of workstations and at least one storage server. At least one signal path interconnects the workstations and the storage server. Each workstation includes video and audio reproduction capabilities, as well as video and audio capture capabilities. Any given storage server comprises a set of storage cells that operate under the direction of a storage cell manager. A storage cell may include one or more encoding and/or transcoding converters configured to convert or transform audio and video signals originating at a workstation into a form suitable for digital storage. A storage cell may further include one or more decoding converters configured to convert digitally-stored signals into a form suitable for audio and video signal reproduction at a workstation. Each storage cell additionally includes at least one storage device and storage device controller capable of storing, for later retrieval, signals generated by one or more converters.

The storage cell manager is responsive to signals received from the workstations, and oversees the operation of the storage cells to facilitate the storage of converted audio and video signals in at least one file that can be simultaneously accessed by one or more application programs executing on one or more workstations. In one embodiment, the storage cell manager directs the storage of converted signals by selecting a storage cell, and determining whether a converter contained therein has available bandwidth or capacity. If so, the storage cell manager selects a storage device controller within the selected storage cell, and determines whether the controller has available bandwidth or capacity. If so, the storage cell manager determines whether a storage device associated

LINKPLAY EXHIBIT 1004
Page 3051 of 4822

with the controller has available bandwidth or capacity, in which case the storage cell manager directs the converter and the selected storage device controller to convert and store the converted signals upon the selected storage device. In the event that bandwidth or capacity is unavailable for the converter, the storage device controller, or the storage device, the storage cell manager selects other such devices within the selected storage cell for consideration relative to bandwidth or capacity availability. If the selected storage cell has reached a bandwidth or capacity limit, the storage cell manager selects another storage cell for consideration. In the event that the storage cell manager or the storage server itself has reached a bandwidth or capacity limit, the storage cell manager requests another storage cell manager to direct the storage of the converted signals.

The storage cell manager further selectively directs the copying or transfer of stored converted signals between 1) any plurality of storage devices within a given storage cell; and/or 2) any set of storage cells, where one or more such storage cells may be under the control of another storage cell manager. Such copying or transfer maximizes the likelihood that any given file containing converted signals will be simultaneously accessible to one or more application programs executing upon multiple workstations. The storage cell manager may also direct the transfer of stored converted signals to workstations or other types of servers coupled to the networked multimedia system.

The storage cell manager also oversees the retrieval and decoding conversion of stored converted signals under the direction of one or more application programs executing on one or more workstations. Such retrieval and decoding conversion facilitates real-time and/or near real-time audio and video signal reproduction at such workstations.

In common usage, a multimedia file comprises one or more types of files and/or references to files, where such files may include textual, graphical, image, audio, and/or video information and/or commands or event sequences for generating or rendering such information. The multimedia file also includes temporal correlation data specifying one or more manners in which its constituent files and/or file references are associated in time. In addition to the above operations, the storage cell manager in one embodiment oversees the storage of these multimedia files or portions of them. The storage cell manager further oversees the retrieval of multimedia file contents and the distribution of such

LINKPLAY EXHIBIT 1004
Page 3052 of 4822

contents to one or more workstations in conjunction with audio and video signal reproduction at such workstations, in a manner specified by the temporal correlation data.

The system is configured to create a pointer referencing a predetermined file or group of files on one or more of the storage cells; to transmit at least the pointer to a receiving application program, and to render the contents of the file or group of files at a workstation by retrieving data from the file or group of files identified by the pointer.

The system also provides a viewer application for setting up connections and sessions; preparing files for viewing, and providing the necessary viewer interface to enable a user to access and manipulate files on the workstation, as permitted.

Further, the system has at least one stored-video application program on at least one of the storage sub-system and a workstation; and share at least one process primitive across at least two of the stored-video application programs. The stored-video application programs are one or more from the group consisting of: videoconference recording; video mail; video answering system; video documents; and video publishing. The system is further configured to share at least one data primitive across at least two of the plurality of stored-video application programs.

The system can also invoke a process primitive and a data primitive as standardized attachments to a file. Such attachment is in a format that is accepted by third parties in accordance with a standard data interchange protocol.

The system can also be configured to establish real time data sharing sessions between at least two workstations using a synchronized data sharing process as a shared conference window. It can use a video conference application to establish a video conference session among the interconnected workstations, and to use a video conference recording application to record at least a portion of the established video conference session on at least one storage sub-system.

The system is further configured to provide at least one video mail application, which can do at least one of authoring a video mail message for transmission to storage subsystem and reading a video mail message previously stored on the storage subsystem. The system typically utilizes a storage cell to record at least one of an audio, a video message, and a multimedia message from an incoming caller whose call attempt is one of

LINKPLAY EXHIBIT 1004
Page 3053 of 4822

not answered by the recipient, refused by the recipient, and made solely to leave a message without connecting to the recipient. Video files stored on a storage cell can be included in an on-line electronic document.

The system further can incorporate a shareboard or other window-sharing session, either static or synchronized with video or audio information to enable a multimedia implementation of at least one of video conference recording, video mail, video answering system, video documents, and video publishing.

Moreover the system can incorporate a video editing program. The video editing program may be implemented either as an integral part of at least one of the applications or by incorporating a third-party video editor.

The system can support video mail applications either using conventional third-party e-mail systems or e-mail systems enhanced to offer the ability to obtain and or file copy events from the e-mail system.

The system can support video document applications either using conventional third-party document systems or document systems enhanced to offer the ability to obtain and or file copy events from the document system.

In addition, the system can function as an internet proxy server for reducing the total number of copies of video files in the enterprise by file sharing, or reducing enterprise network load by transporting video files over a separate audio-video network, or transcoding among multiple audio-video formats, and or allowing any audio-video workstation in the enterprise to serve as a video publisher for the enterprise. Finally, the system, functioning as an internet proxy server, enables the implementation of at least one animated annotation on the internet.

3. BRIEF DESCRIPTION OF THE DRAWINGS

For fuller understanding of the principles of the present invention, reference is made to the several figures of the drawing. In the drawing:

Figure 1 is an exemplary Erlang diagram showing computational resource utilization efficiency relative to the number of users sharing the resource under fixed blocking conditions.

LINKPLAY EXHIBIT 1004
Page 3054 of 4822

Figure 2 is a graph showing the relative cost of analog and digital distribution technology as a function of performance and image quality.

Figure 3 is a block diagram of a Collaborative Multimedia Computing Environment (CMCE) employing an Audio/Video Server System (AVSS) constructed in accordance with the present invention.

Figure 4 is a block diagram of an alternate embodiment of a CMCE constructed in accordance with the present invention.

Figure 5 is a block diagram of an Audio/Video Storage Cell (AVSC) constructed in accordance with the present invention is shown.

Figure 6 is a block diagram of a first embodiment of a shared coding unit of the present invention.

Figure 7 is a block diagram of a second embodiment of the shared coding unit.

Figure 8 is a block diagram of a third embodiment of the shared coding unit.

Figure 9 is a block diagram of an Audio/Video Server Manager (AVSM) constructed in accordance with the present invention.

Figure 10 is a block diagram illustrating client-server session communication in one embodiment of the present invention.

Figure 11 is a block diagram showing an exemplary recording control Graphical User Interface (GUI).

Figure 12 is a block diagram showing an exemplary playback control GUI.

Figure 13 is a block diagram showing an exemplary browsing control GUI.

Figure 14 is a block diagram of an exemplary networked AVSS organization.

Figure 15 is a representation of a communication class hierarchy in the present invention.

Figure 16 is a representation of an AVSM class relationship in the present invention.

Figure 17 is a representation of an AVSC class relationship in the present invention.

Figure 18 is an exemplary request sequence diagram corresponding to AVSC channel acquisition.

LINKPLAY EXHIBIT 1004
Page 3055 of 4822

Figure 19 is an exemplary request sequence diagram for an "Open" request.

Figure 20 is an exemplary request sequence diagram for a "Close" request.

Figure 21 is an exemplary request sequence diagram for a "Delete" request.

Figure 22 is an exemplary request sequence diagram for a "Record" request.

Figure 23 is a exemplary request sequence diagram corresponding to a request for a file transfer from a non-premises AVSS.

Figure 24 is an exemplary request sequence diagram corresponding to file replication operations.

Figure 25 is an exemplary request sequence diagram corresponding to particular AVSS administrative operations.

Figure 26 is a block diagram showing client application programs communicating with the AVSM and A/V network manager.

Figure 27 is an overview of the file hierarchy.

Figure 28 is an exemplar data primitive.

Figure 29 is an exemplar data primitive implemented as an attachment.

Figure 30 is an exemplar system for implementing the software applications of the present invention.

Figure 31 is a transaction flow diagram among the application elements which facilitate local-area video mail.

Figure 32 is a transaction flow diagram among the application elements which facilitate wide-area video mail.

Figure 33 is a representation of the life cycle of an exemplar video message file.

Figure 34 is a comparative diagram of the ownership and readability of video files associated with a video mail attachment, over the message lifetime.

Figure 35 is an overview of one embodiment of the video answering system of the present invention.

Figure 36 is a data flow diagram detailing the operation of one embodiment of the present invention.

Figure 37 an overview of the video conference recording application of the present invention.

LINKPLAY EXHIBIT 1004
Page 3056 of 4822

Figure 38 an overview of several of the components of the present invention required to implement video conference recording.

Figure 39 is a signal state diagram of a two-party video conference.

Figure 40 is a signal state diagram of a video conference, including a third party or implementing conference recording.

Figure 41 is a signal state diagram at the initiation of conference recording.

Figure 42 is a signal state diagram of a first video conference representation.

Figure 43 is a signal state diagram of a second video conference representation.

Figure 44 is a signal state diagram of a third video conference representation.

Figure 45 is a data flow diagram of one software implementation of the video conference application of the present invention.

Figure 46 is a signal state diagram at the initiation of video conference playback.

4. DETAILED DESCRIPTION

4.1 Architectural Framework

The present invention comprises at least one shared centralized audio-video (A/V) file storage and processing system within a collaborative or networked multimedia computing environment. Herein, this collaborative or networked multimedia computing environment comprises a plurality of user workstations plus multimedia-enabled servers that are linked together via one or more networks. The present invention further comprises workstation-based application programs plus control software executing on the servers that facilitate the exchange of A/V and/or multimedia information among workstations and servers in real-time, near-real-time, and/or non-real time. The A/V file storage and processing system provides a wide range of video storage and playback services to workstation application programs and/or other servers, including several of analog to digital encoding, digital to analog decoding, transcoding digital file storage, multimedia file recording, multimedia file playback analog streaming digital format transcoding, digital file streaming, digital file transfer, and file administration operations that facilitate the recording/ encoding, storage, distribution, decoding/playback, copying, archival, and deletion of A/V and/or multimedia files.

LINKPLAY EXHIBIT 1004
Page 3057 of 4822

Figure 3 is a block diagram of a Collaborative Multimedia Computing Environment (CMCE) 10 employing an Audio/Video Server System (AVSS) 100 constructed in accordance with the present invention. The CMCE 10 comprises a data network 20; an A/V network 30; a plurality of user workstations 40 and/or a set of A/V conference rooms 45; an Audio/Video Server System (AVSS) 100; and a set of supporting server systems that include an e-mail system 50, an intranet server system 60, and a firewall/internet gateway system 70. The data network 20 couples the A/V network 30, the workstations 40, the A/V conference room(s) 45, each supporting server system 50, 60, 70, and the AVSS 100. The data network 20 also maintains a wide area link that is coupled to a Wide Area Network (WAN) gateway 25, which is coupled to a first WAN 29. The A/V network 30 couples the workstations 40, the A/V conference rooms 45, and the AVSS 100. The A/V network 30 maintains at least one trunk line coupling 16 to a remote and/or another local A/V network 30. The A/V network 30 additionally maintains a trunk line coupling 16 to a coder/decoder (codec) gateway 38, which is coupled to a second WAN 39.

4.2 Signal Distribution Networks

The present invention supports both real-time (or near real-time) and delayed A/V signal exchange or distribution on a local and/or remote basis. In the context of real-time A/V signal distribution via currently-available networking technologies, analog premises networking provides, for the foreseeable future, higher video quality and real-time performance at a lower cost than digital premises networking. In the embodiment shown in *Figure* 3, the CMCE 10 relies upon the data network 20 to facilitate the exchange of digital information between CMCE elements, and the A/V network 30 to facilitate analog signal exchange. Thus, the analog signal premises distribution network shown in *Figure* 3 provides a low-cost CMCE implementation capable of delivering high-quality A/V signals (i.e., NTSC television-quality video at 640 x 480 pixels or similar standard television resolution, plus 7-15 kHz high-fidelity audio) in real-time. This embodiment additionally ensures that real-time A/V signal distribution has essentially no impact upon local data network loading. As described in detail below, alternate CMCE embodiments could rely upon a single physical network that utilizes any of a variety of suitable analog

and/or digital multiplexing scheme; these will become increasingly important over time as suggested by the time evolution of the curves in ***Figure*** 2.

In ***Figure*** 3, the data network 20 provides for both local- and wide-area digital networking, while the A/V network 30 provides for both local- and wide-area analog networking. In terms of local analog networking, a single instance of the A/V network 30 typically directly serves one AVSS 100 plus an associated group of workstations 40 and/or A/V conference rooms 45 located within a single premises. Hence, an A/V network 30, its corresponding AVSS 100 and workstations 40 and A/V conference rooms 45 are referred to herein as a premises group. Multiple proximate premises groups may be locally linked via trunk lines among corresponding data and A/V networks 20, 30 to form a common campus. One example of this is a large office building where different areas or building floors are served by different premises groups. Another example involves situations where a corporate headquarters facility includes multiple generally-proximate buildings, one or more premises groups could be deployed within each building. Each of the premises groups within these buildings could then further be locally coupled to form a common campus, in a manner consistent with generally-accepted use of the term "campus" in corporate environments. The structure and functionality of the data and A/V networks 20, 30, and the manners in which they support premises, campus, and wide-area or remote networking, are described in detail hereafter.

4.2.1 Data Network

The data network 20 comprises a conventional network hub, plus data links 12 that implement an Ethernet, Asynchronous Transfer Mode (ATM), or other type of network, and which facilitates the exchange of data and control signals between the elements to which it is coupled either locally or via the wide area link. In the context of the present invention, the data network 20 is a Local Area Network (LAN) that spans an extent ranging from approximately a few tens of meters to perhaps one kilometer. The data links 12 can comprise unshielded twisted pair (UTP) wiring, compatible to standard telephone system wiring, or essentially any other type of network communication coupling (coax, optical, wireless radio, etc.).

LINKPLAY EXHIBIT 1004
Page 3059 of 4822

The data network 20 facilitates digital communication within at least one premises group, where such communication may include control signal transfer, digital file transfer, and/or digital streaming. Via a set of data links 12 locally coupled to conventional routers, data switching hubs, or a network backbone, one or more data networks 20 can readily serve multiple premises groups (i.e., a common campus) subject to geographic, network-loading, and signal quality constraints, in a manner readily understood by those skilled in the art. The data network's wide area link facilitates digital communication between premises and non-campus or remote CMCE elements. The wide area link is coupled to the gateway and WAN 25, 29, which are implemented in a conventional manner such as that described in U.S. Patent No. 5,617,539.

4.2.2 A/V Network

The A/V network 30 distributes A/V signals under the direction of a server that is responsive to requests and messages received from other CMCE elements. The A/V network 30 comprises an A/V switch 32, an A/V network manager 34, and a conference bridge 36. The A/V switch 32 is coupled to the conference bridge 36, the workstations 40, and the AVSS 100 via analog links 14. The A/V switch 32 maintains trunk line couplings 16 to one or more campus A/V networks, and also to the codec gateway 38, which couples the A/V switch 32 to the second WAN 39. The A/V network manager 34 is coupled to the data network 20, the A/V switch 32, and the conference bridge 36 via digital links 12.

The A/V switch 32 comprises standard commercially-available circuitry for selectively establishing analog signal couplings between a source port and one or more output ports. The A/V switch 32 may be implemented using CMOS analog switching elements that are linked by buses to form a cross-bar switch. Such CMOS analog switches are controlled by one or more microprocessors that receive commands through serial or data network ports.

A trunk line 16 couples the A/V switch 32 to other campus-based A/V switches 32. In addition, a trunk line 16 couples the A/V switch 32 to the codec gateway 38. In one embodiment, the codec gateway 38 is conventional. Other embodiments may involve physically separate switching multiplexers and/or access multiplexers. The codec gateway 38 could be implemented, for example, using a Zydacron codec (Zydacron, Inc.,

Manchester, NH) or a Tandberg codec (Tandberg, Lysaker, Norway and Herndon, VA). The codec gateway 38 is coupled to the second WAN 39, thereby facilitating wide-area A/V networking. In general, the second WAN 39 comprises a conventional network capable of providing a guaranteed quality-of-service level and low latency data transfer, such as a T1, DS3, ISDN, public-switched, or other network. Those skilled in the art will understand that the WAN 39 utilized for wide-area A/V networking may be the same as that used for wide-area data networking (i.e., a single conventional ISDN, T-carrier, ATM, or frame relay telecommunications WAN), where the A/V and data signals are multiplexed in accordance with priority and quality-of-service considerations in a standard manner. That is, the present invention could be served by a single WAN rather than separate WANs 29, 39.

The conference bridge 36 comprises conventional audio mixing and video mosaicing circuitry. The conference bridge 36 selectively provides A/V and/or multimedia conference participants with one or more conference video images, as well as conference audio streams. In one embodiment, the conference video images comprise mosaiced subsets of video images generated by conference participants, as well as a video mosaic of each video image associated with all conference participants. Similarly, the conference audio streams comprise subsets of audio signals generated by conference participants, plus an audio stream corresponding to all conference participants.

The A/V network manager 34 comprises a server plus attendant software that coordinates or manages the operation of the A/V switch 32 and conference bridge 36 in response to requests received over the data network 20. The A/V network manager 34 provides an Application Program Interface (API) through which client application programs may request A/V and/or multimedia switching and/or conferencing services. Thus, the A/V network manager 34 directs the A/V switch 32 in establishing A/V and/or multimedia sessions and/or conferencing sessions between workstations 40, A/V conference rooms 45, the AVSS 100, and/or the trunk line 16.

In one embodiment, the A/V network manager comprises a collaborative A/V and/or multimedia conferencing system that is implemented in the manner described in U.S. Patent No. 5,617,539, entitled "Multimedia Collaboration System with Separate Data

Network and A/V Network Controlled by Information Transmitting on the Data Network."

4.2.3 Multimedia Network

Taken together, the data network 20 and the A/V network 30 comprise a Multimedia Local Area Network (MLAN). Wide-area networking is facilitated by the data network's wide-area link gateway and WAN, 25, 29, plus the A/V network's trunk line coupling 16 to the codec gateway and WAN, 38, 39. The MLAN WANs 29, 39 of the present invention may be implemented in the manner described in U.S. Patent No. 5,617,539. A collection of MLANs coupled by WAN, 29, 39 facilitate the exchange of A/V and/or multimedia information between premises, campus, and/or remote CMCE elements.

4.2.4 Alternate Signal Distribution Architecture

As previously indicated, the present invention could rely upon a single network for distributing both data and A/V signals. ***Figure*** 4 is a block diagram of an alternate embodiment of a CMCE 11 constructed in accordance with the present invention. With regard to ***Figure*** 3, like reference numbers have been used to denote like elements in ***Figure*** 4. In the alternate embodiment, the CMCE 11 comprises a data network 20; an A/V conference manager 34 coupled to a conference bridge 36; a set of workstations 40 and possibly one or more AV conference rooms 45; a set of supporting server systems 50, 60, 70; and an AVSS 100. The data network 20 couples the A/V conference manager 34, the conference bridge 36, the workstations 40 and A/V conference room(s) 45, each supporting server system 50, 60, 70, and the AVSS 100.

In the single-network CMCE 11, the data network 20 is typically implemented as an IP or ATM network. A/V file exchange between CMCE elements such as the AVSS 100 and workstations 40 or A/V conference rooms 45 would occur via streaming or file transfer, and each workstation 40 may include compression/decompression resources.

4.3 Workstations And A/V Conference Rooms

Users interact with CMCE elements, including the AVSS 100, via application programs executing on workstations 40. Each workstation 40 comprises a conventional desktop-based computer system having a processing unit, memory, disk drive, display

device, keyboard, mouse, and speaker(s). Any particular workstation 40 can be implemented in accordance with essentially any hardware/software platform, such as a Windows-based (Microsoft Corporation, Redmond, WA) personal computer, an Apple-based (Apple Computer Corporation, Cupertino, CA) computer, a Unix-based computer, or other type of system. Each workstation 40 is also equipped with a video camera and a microphone, such as in the manner described in U.S. Patent No. 5,617,539. The video camera and microphone generate a high-quality A/V signal that is directed to the A/V network 30 via an analog line 14. In like manner, the display device receives a high-quality A/V signal from the A/V network 30 via the analog line 14.

In embodiments in which each workstation 40 is coupled to both the data network 20 and the A/V network 30, a variety of possibilities exist for the transfer of A/V and/or multimedia files to any given workstation 40. In particular, the present invention supports "real-time" analog A/V file transfer (i.e., analog streaming) to any workstation via the A/V network 30, as well as both digital streaming and the transfer of entire digital files to any workstation 40 via the data network 20. The present invention additionally supports real-time analog A/V file transfer simultaneously with either digital streaming or digital file transfer to any combination of workstations 40. The particular types of file transfer employed at any given moment are determined by application programs executing on the workstations 40 as well as premises and/or non-premises application servers coupled to the CMCE 10.

In addition to exchanging A/V signals with the workstations 40, the A/V network 30 can exchange A/V signals with presentation, conferencing, and/or computing resources located in one or more A/V conference rooms 45, where such resources may include cameras, monitors, televisions, microphones, and speakers. This in turn means that a group meeting held in an A/V conference room 45 can exchange A/V and/or multimedia information with individual workstations 40 or other A/V conference rooms 45, on a premises, campus, or remote basis.

4.4 Supporting Servers

The AVSS 100 provides novel A/V and/or multimedia functionality to various application programs, as described in detail below. In so doing, the AVSS 100 selectively

LINKPLAY EXHIBIT 1004
Page 3063 of 4822

leverages the capabilities of the supporting server systems 50, 60, 70. The e-mail system 50 preferably comprises conventional e-mail server hardware and software capable of creating, storing, and distributing e-mail messages with attached files between users or target destinations. The manners in which the AVSS 100 interacts with the e-mail system 50 to provide video attachments and other multimedia e-mail capabilities are described in detail below.

Many corporations or enterprises employ an intranet system, which comprises a private network upon which enterprise-related information is distributed and/or exchanged in accordance with conventional internet protocols. An intranet system leverages readily-available, low cost internet software tools to efficiently provide employees or enterprise members with access to enterprise-wide information. Such enterprise-wide information may include corporate communications documents or bulletins; training materials; product and/or project information; personnel directories; and member- or employee-specific data maintained within an enterprise database, where access to such data is granted only for authorized members. The intranet server system 60 comprises conventional hardware and software that provide information sharing or distribution services to employees or authorized members within a corporation or organization, where such information sharing occurs in accordance with a conventional internet protocol suite (i.e., TCP/IP). Details concerning the manners in which the AVSS 100 interacts with the intranet server system 60 to facilitate A/V and/or multimedia information sharing are provided below.

The internet gateway/firewall system 70 comprises hardware and software that implement a conventional internet firewall security and File Transfer Protocol (FTP) gateway system, for exchanging messages and A/V or multimedia files between the AVSS 100 and the public internet 80. Specific AVSS functionality in this regard is described in detail below.

4.5 AVSS Architecture

The AVSS 100 comprises a repository for A/V file storage and processing resources, to which application programs executing on premises, campus, and/or remote CMCE elements have shared access. In the present invention, application programs initiate or generate service requests directed to the AVSS 100 in response to user actions.

LINKPLAY EXHIBIT 1004
Page 3064 of 4822

Such application programs may be executing on premises, campus, or remote workstations 40, as well as computers coupled to an enterprise intranet or the public internet. Service requests comprise an appeal for either a) A/V resource or service allocation; or b) A/V resource or service state information. The AVSS 100 receives the service requests via the data network 20, and establishes message-based service sessions with the A/V network 30, workstations 40, supporting servers 50, 60, 70, and/or one or more campus-based or remote CMCE elements to provide A/V and/or multimedia services in accordance with such requests. Messages generated during a service session may comprise state information, control commands, and confirmations. The structure and functionality of the AVSS 100, and the manners in which the AVSS 100 processes service requests and generates messages, are described in detail hereafter.

As shown in ***Figure*** 3, the AVSS 100 comprises an internal network 110, at least one Audio/Video Storage Cell (AVSC) 120, and an Audio/Video Server Manager (AVSM) 160. The internal network 110 couples each AVSC 120, the AVSM 160, the data network 20, the intranet server 60, and the internet gateway/firewall system 70. In addition, the AVSM 160 supports coupling to each AVSC 110 via a dedicated subnet 112, as further described below. Moreover, an analog line 14 couples each AVSC 110 to the A/V network 30.

Each AVSC 120 serves as an A/V file repository, as well as a repository for shared A/V processing resources, including encoders, decoders, and possibly transcoders. The AVSM 160 coordinates the activities of the AVSCs 120, and manages processes both internal and external to the AVSS 100 to carry out requests generated by either premises, campus, or remote CMCE elements. The structure and functionality of each AVSC 120 and the AVSM 160 are described in detail below.

The internal network 110 comprises a conventional high-bandwidth network that facilitates high-speed transfer of A/V files between a) individual AVSCs 120; b) AVSCs 120 and the data network 20; and c) AVSCs 120 and the intranet server system 60 or the internet gateway/firewall system 70. The internal network 110 additionally serves as the medium by which service requests and messages are exchanged between the AVSM 160 and the data network 20. The bandwidth of the internal network 110 is sufficient to carry

LINKPLAY EXHIBIT 1004
Page 3065 of 4822

service requests, control messages, file transfers, and the file streaming capacity of the composite AVSS 100. In one embodiment, the internal network 110 may be implemented as a portion of the data network 20.

The dedicated subnet 112 provides for the exchange of messages and control signals between the AVSM 160 and individual AVSCs 120. In the event that the AVSS 100 utilizes few AVSCs 120, such messages and control signals can be carried by the internal network 110, eliminating the need for the subnet 112. As the AVSS 100 is scaled, however, use of the subnet 112 is desirable to minimize internal network loading.

4.5.1 AVSC Architecture

As previously indicated, each AVSC 120 provides A/V file storage and A/V processing resources that are shared by other CMCE elements. ***Figure*** 5 is a block diagram of an AVSC 120 constructed in accordance with the present invention. The AVSC 120 comprises a processing unit 122, a shared data storage unit 124, at least one network interface 126, an AVSM interface 128, at least one shared coding unit 130, at least one A/V interface unit 140, and a memory 150 that includes an AVSC object memory 152, an AVSC state memory 154, and an operating system 156. With the exception of the A/V interface unit 140, each AVSC element is coupled to a common AVSC bus 159. The A/V interface unit 140 serves as an interface between each shared coding unit 130 and the A/V switch 32. Finally, the network interface 126 and the AVSM interface 128 couple the AVSC 120 to the AVSS internal network 110 and subnet 112, respectively.

The processing unit 122 comprises a conventional high-performance processor for executing program instructions stored within the memory 150. The network interface 126 comprises conventional network interface circuitry for managing data exchanges between the AVSC 120 and the internal network 110. In like manner, the AVSM interface 128 comprises conventional network interface circuitry for managing the exchange of messages and control signals between the AVSC 120 and the AVSM 160. The operating system 156 preferably comprises conventional, real-time multitasking operating system software such as Windows NT (Microsoft Corporation, Redmond, WA) or real-time Unix.

LINKPLAY EXHIBIT 1004
Page 3066 of 4822

The AVSC object memory 152 stores a plurality of AVSC software objects that direct AVSC hardware allocation and resource locking; A/V file encoding, decoding, and transcoding operations; and file management operations such as file replication, transfer, and deletion, as described in detail below. The AVSC objects also maintain the contents of the AVSC state memory 154, which includes the following information:

1) encoder/decoder/transcoder resource capability and current status/utilization;
2) current storage device capacity and utilization;
3) a time-stamped and indexed request queue, for both incoming and outgoing requests;
4) a time-stamped and indexed message queue, for both incoming and outgoing messages;
5) a time-stamped and indexed file transfer event queue; and
6) an AVSC event log, specifying standard time-stamped events, as well as occurrences of encoder, decoder, transcoder, storage device, and/or network faults.

To minimize the amount of hardware required to meet any particular A/V file storage and/or multi-user A/V file access performance requirements, premises A/V file storage hardware should be shared or centralized rather than localized at each workstation 40. Centralization of data storage hardware also maximizes system-wide utilization and administration efficiency due to statistical averaging, and can also improve fault-tolerance. The shared data storage unit 124 comprises at least one high-capacity disk drive for storing A/V files and data, as well as a corresponding disk drive controller. Within a CMCE 10 having a premises analog signal distribution network, A/V file storage resources are fully AVSC-based rather than workstation-based. In AVSS implementations having more than one AVSC 120, multiple copies of any given A/V file may be stored across different AVSCs 120, as described in detail below, to improve the system's response in accessing commonly accessed files. Further, when an AVSS 100 employs a single AVSC 120, multiple copies of any given A/V file could be stored across multiple 1) shared data storage units 124 within the AVSC 120; and/or 2) storage devices coupled

LINKPLAY EXHIBIT 1004
Page 3067 of 4822

to the AVSC 120. From an external perspective, the AVSS 100 operating in this manner provides centralized A/V file storage, while A/V files are distributed internally amongst the shared data storage units 124 across the set of AVSCs 120.

Each shared coding unit 130 comprises at least one instance of encoding, decoding, and/or transcoding resources. For shared coding units with more than one encoder and/or decoder, a multichannel version of the A/V interface 140 will be required (or, alternatively multiple A/V interfaces 140 can be provided for a single coding unit 130. In some embodiments, transcoding may be done in software by the Processing unit. In other embodiments, a dedicated hardware transcoder may be used - such an instance of 130 would not require an AV interface 140. The present invention can support essentially any encoding format, including MPEG-based, RealMedia, and NetShow formats. In one embodiment, the AVSC 120 uses a given encoding format as a default format, for example, MPEG1 encoding that is tunable within a compression performance range of approximately 1.1 to 3 Mbps. The particular encoding format employed at any given time is an application-dependent and/or user-dependent parameter, as further described below.

The creation and recording of an A/V file typically requires a single encoding session, which may be stopped or paused and restarted as necessary. Since an A/V file is likely to be played back multiple times, either in repeated or concurrent sessions, decoding is likely to be a much more frequently requested operation than encoding. Across the set of AVSCs 120 within the AVSS 100, one or more shared coding units 130 may support different encoding, decoding, and/or transcoding session options. For example, the hardware within the shared coding unit 130 in any particular AVSC 120 could support, for example, any one of the following:

1) one encoding session or one decoding session, mutually exclusive;
2) one encoding session concurrent with one decoding session;
3) multiple concurrent encoding and decoding sessions;
4) multiple concurrent decoding sessions; or
5) one or multiple transcoding sessions, in conjunction with any of the above.

LINKPLAY EXHIBIT 1004
Page 3068 of 4822

In addition, any AVSC 120 could support one of the above either mutually exclusive of or concurrent with file transfer to or from the shared data storage unit 124.

In general, at least one AVSC 120 is capable of concurrently supporting both A/V file encoding and decoding sessions, as well as multiple sessions simultaneously. Additionally, each AVSC 120 is highly scalable relative to the number of encoding, decoding, and transcoding resources required to meet evolving AVSS implementation needs. Furthermore, each AVSC 120 can support essentially any type of encoding, decoding, and/or transcoding resource without architectural modification, thereby readily accommodating A/V processing resource evolution over time. Thus, a wide variety of shared coding unit embodiments exist, examples of which are described hereafter.

Referring now to ***Figure*** 6, a block diagram of a first embodiment of the shared coding unit 130 is shown. In the first embodiment, the shared coding unit 130 comprises at least one encoder 132, plus a set of decoders 134 capable of supporting multiple simultaneous decoding sessions. This requirement can be met through the use of a plurality of mutually-exclusive decoders 134, or one or more concurrent-capable decoders 134.

Referring also now to ***Figure*** 7, a block diagram of a second embodiment of the shared coding unit 130 is shown. In the second embodiment, the shared coding unit 130 comprises a set of decoders 134 capable of supporting multiple concurrent decoding sessions. As in the example embodiment, the set of decoders 134 can be implemented via multiple mutually-exclusive decoders 134, or at least one concurrent-capable decoder 134. In the second embodiment, the AVSC 120 in which the set of decoders 134 reside(s) is dedicated exclusively to fulfilling decoding requests, which in turn implies that at least one encoder 132 would be present within another AVSC 120.

Those skilled in the art will understand that A/V files may be encoded in accordance with a variety of formats. Thus, at times, transcoding may be required to convert an A/V file into a different format than that which was utilized to record the A/V file. Referring now to ***Figure*** 8, a block diagram of a third embodiment of the shared coding unit 130 is shown, which comprises at least one encoder 132, at least one decoder 134, and a transcoder 136. When an AVSC 120 includes one or more transcoders 136,

LINKPLAY EXHIBIT 1004
Page 3069 of 4822

the present invention preferably provides for real-time transcoding, such that playback of A/V files requiring format conversion can occur without sequential full-file conversion delays.

In a manner analogous to that for the shared data storage unit 124, encoding, decoding, and transcoding resources are AVSC-based rather than workstation-based in the present invention. From an external perspective, the AVSS 100 provides centralized encoding, decoding, and transcoding resources, while internally the aforementioned resources are distributed amongst one or more AVSCs 120. This pooled resource organization maximizes encoding, decoding, and transcoding resource utilization efficiency, while minimizing the number of such resources required to meet any given performance requirements.

In an exemplary embodiment, an AVSC 120 is implemented as a personal computer server having a Pentium II (Intel Corporation, Santa Clara, CA) or generally-equivalent microprocessor, 128 Megabytes of Random Access Memory (RAM), an Optibase MovieMaker encoder and an Optibase VideoPlex decoder (Optibase Inc., San Jose, CA), a network interface card, and an UltraWide and Fast SCSI 18 Gigabyte or larger disk drive for storing A/V and related files.

4.5.2 AVSM Architecture

The AVSM 160 comprises hardware and software that manages or coordinates the processing of requests received from premises-based, campus-based, or remote CMCE elements. The processing of such requests may result in the creation, recording/encoding, storage, distribution, decoding/playback, archival, or deletion of A/V or multimedia files in the context of conference recording, telephone answering, e-mail, document generation, document publishing, or other applications as described in detail below. Those skilled in the art will understand that many architecturally varying embodiments are possible. The embodiment described below provides an enabling architectural example.

The AVSM 160 comprises a network-based multitasking computer. ***Figure*** 9 is a block diagram of an AVSM 160 constructed in accordance with the present invention. The AVSM 160 comprises a processing unit 162, a network interface 164, an AVSC interface unit 166, a data storage unit 168, and a memory 170 wherein an AVSM object

LINKPLAY EXHIBIT 1004
Page 3070 of 4822

memory 172, an AVSS state memory 174, an AVSS database memory 176, and an operating system 178 reside. Each element within the AVSM 160 is coupled to a common AVSM bus 199.

The processing unit 162 comprises a conventional high-performance processor for executing stored program instructions, and the data storage unit 168 comprises at least one disk drive. The network interface 164 and the AVSC interface unit 166 comprise conventional network interface circuitry for managing communication with the internal network 110 and AVSCs 120, respectively. The operating system 178 comprises a conventional multitasking operating system such as Windows NT.

The AVSM object memory 172 stores a plurality of AVSM software objects that perform or manage the AVSS services and resource management operations described in detail below, which include a) the establishment of user request processing sessions; b) allocation of AVSC resources and associated resource locking; c) establishment of A/V network communication couplings; d) user application interfacing and message routing; e) inter- and intra-AVSS file transfer initiation; and f) administrative operations as described in detail below. AVSM objects also maintain the contents of the AVSS database, which is stored in the AVSS database memory 176 and/or upon the data storage unit 168.

The AVSS database stores a variety of information that defines a) the AVSS communications environment; b) the nature and capabilities of the shared AVSC resources; c) administrative parameters and usage data; and d) characteristics of the A/V files stored on the AVSCs 120. Relative to the AVSS communications environment, the information the AVSS database stores includes the following:

1) the number and types of file transfer channels, including digital file transfer channels arising from AVSC couplings to the internal network 110 and the data network 20, plus analog file transfer channels corresponding to AVSC couplings to the A/V network 30;
2) a premises AVSS host name and port identifier;
3) one or more non-premises AVSS host names and port identifiers, plus connection setup information associated with each non-premises AVSS 100;

LINKPLAY EXHIBIT 1004
Page 3071 of 4822

4) A/V network configuration parameters;
5) supporting server system configuration parameters;
6) a user identification (ID) or name for each user within the premises group to which the AVSS 100 belongs, plus password information corresponding to each user ID; and
7) communication preferences corresponding to each user ID, such as preferred encoding format and/or A/V file delivery format.

Relative to shared AVSC resources, the AVSS database includes the following information:

1) the number, types, and capacities of the storage resources within each AVSC 120; and
2) the number, types, and capabilities of the encoding, decoding, and transcoding resources within each AVSC 120.

The administrative parameters that reside within the AVSS database include a maximum allowable A/V file size, and a maximum allowable A/V file age. The maximum allowable A/V file age may be defined in relation to an A/V file's creation time and date, or relative to a most-recent access. The usage data includes an event statistics file that details event occurrence frequencies, such as a number of users that have accessed the AVSS 100 in a given time interval; a number of times an A/V file has been accessed; a most-recent access date and time for an A/V file; and an amount of time spent performing A/V file playback operations during a particular time interval.

For each unique A/V file stored on the AVSCs 120, the AVSS database stores a file parameter table that includes the following information:

1) a filename;
2) a file password;
3) a user ID indicating file authorship;
4) a file ownership or access privilege list, by user ID, which includes an access expiration date and time as well as recent access history (date and time) associated with the user ID;
5) A/V encoding format;

LINKPLAY EXHIBIT 1004
Page 3072 of 4822

6) time and date of most-recent modification;
7) file size;
8) file playback duration;
9) file age;
10) location of each copy of the file across the AVSCs 120;
11) a list identifying and locating any AVSS-resident multimedia synchronization files associated with the A/V file; and
12) a list specifying any non-AVSS target servers to which the file has been published, as described in detail below.

For each premises group user ID, the AVSM objects could also maintain a user-specific file list in the AVSS database. In such an embodiment, the user-specific file list includes the name of each A/V file for which the user ID is specified in the file's access privilege list, and indicates whether the user ID is specified in the file's authorship data.

AVSM objects additionally maintain the contents of the AVSS state memory 174 to reflect current resource, request, and message status across all currently-active request processing sessions. The contents of the AVSS state memory 174 include the following:

1) a current user list that specifies user IDs corresponding to currently logged-in users;
2) a list of currently-active request processing sessions;
3) a time-stamped and indexed request queue identifying currently-pending requests, for both incoming and outgoing requests;
4) currently-available capacity and utilization for each AVSS storage resource;
5) current encoding/decoding/transcoding resource utilization; and
6) an AVSM event log specifying session time-out occurrences; application errors; AVSC faults; network faults; changes in the current user list and currently-active session list; changes in storage resource utilization and available storage space; and changes in encoding, decoding, and/or transcoding resource utilization. The contents of the AVSM event log are used to update the event statistics file.

LINKPLAY EXHIBIT 1004
Page 3073 of 4822

In an exemplary embodiment, the AVSM 160 is implemented as a personal computer server having a Pentium II or generally-equivalent processor, a network interface card, 128 Megabytes of RAM, and a 10 Gigabyte or larger hard disk drive. Those skilled in the art will understand that the AVSM 160 could be implemented using a hardware/software platform that is essentially identical to that used for implementing an AVSC 120.

4.5.3 Scaling Hierarchy

The present invention can be scaled to provide enhanced performance and/or additional capabilities. Two types of scaling are possible, namely, numerical and evolutionary scaling. Numerical scaling implies the incorporation of additional hardware elements, while evolutionary scaling implies replacement of existing hardware elements with higher-performance or more "technologically evolved" hardware. The architectures described herein flexibly accommodate numerical and/or evolutionary scaling across several interrelated hierarchical performance boundaries or levels, which include the following:

4.5.3.1 Bus and processor throughput - the capabilities of the bus and processor within an AVSC 120 or AVSM 160 respectively define one type of AVSC or AVSM performance boundary. For example, bus throughput considerations limit the amount of bandwidth available for transferring information between elements within any given AVSC 120 or AVSM 160. Through a hardware platform upgrade, which is an evolutionary scaling, this performance boundary can be surpassed or extended.

4.5.3.2 Shared data storage unit throughput and capacity - the data transfer rate and storage capacity of the shared data storage unit 124 within any particular AVSC 120 define another performance boundary. This boundary can be surpassed by the inclusion of additional data storage devices within the shared data storage unit 124, and/or the use of higher-performance data storage devices. The performance gain achieved through numerical or evolutionary scaling in this case is dependent upon the bus throughput boundary described above.

LINKPLAY EXHIBIT 1004
Page 3074 of 4822

4.5.3.3 Shared coding unit throughput and capabilities - the A/V file processing capabilities of the shared coding unit 130 within any given AVSC 120 also define a performance boundary. The incorporation of additional encoding, decoding, and/or transcoding resources within a shared coding unit, and/or the use of higher-performance A/V processing resources, results in the extension of this boundary. Those skilled in the art will understand that higher-performance resources in this case would likely perform essentially-simultaneous encoding, decoding, and/or transcoding operations on multiple execution threads. Those skilled in the art will additionally recognize that the performance gain achieved through numerical or evolutionary scaling here is dependent upon the aforementioned bus throughput boundary.

4.5.3.4 Collective AVSC capabilities and throughput - the overall A/V file processing performance and capabilities of the set of AVSCs 120 within the AVSS introduce another performance boundary, which can be surpassed through the incorporation of additional AVSCs 120 into the AVSS, and/or the use of higher-performance AVSCs 120. Those skilled in the art will understand that increasing the performance of any given AVSC 120 involves surpassing one or more of the previously-described performance boundaries.

4.5.3.5 Internal network bandwidth - the data transfer capabilities or loading limitations of the AVSS internal network 110 introduce another performance boundary, which can be surpassed through evolutionary scaling.

4.5.3.6 Collective networked AVSS performance and capabilities - the performance and capabilities of an entire AVSS group coupled via a campus and/or wide-area network also introduces a performance boundary that can be readily surpassed through numerical and/or evolutional scaling in accordance with performance requirements relative to cost constraints.

Those skilled in the art will recognize that just as a system can be scaled upward relative to performance, it can also be scaled downward, which might occur, for example, when cost constraints are of paramount importance. The architecture of the present

LINKPLAY EXHIBIT 1004
Page 3075 of 4822

invention, however, can provide high A/V and/or multimedia processing performance while leveraging the capabilities of readily-available, low-cost technology.

4.6 Functional Partitioning Between AVSM And AVSCs

The types of functions performed by the AVSM 160 are partitioned relative to those performed by the AVSCs 120 to ensure consistent operability across essentially any internal AVSS implementation, particularly in view of numerical and evolutionary scalability. As a result, the AVSM 160 can support a continually-evolving range of AVSC implementations without modification.

The AVSM 160 provides centralized management of AVSS functionality. The AVSM 160 receives AVSS service requests and control messages from user applications. In response to service requests, the AVSM 160 establishes request processing sessions, as described in detail below. The AVSM generates a session handle to uniquely identify each such session. Exemplary AVSS service requests include the following:

1) login {user ID, password};
2) create and encode A/V file {session handle file name, file password, plus other parameters described below};
3) fetch and decode A/V file {session handle, file name, file password};
4) delete A/V file {session handle file name, file password};
5) copy A/V file {source address, source session handle, target address, target session handle plus other parameters described below}; and
6) move A/V file {source address, source session handle, target address, target session handle plus other parameters described below}.

Other types of AVSS service requests include requests for sending or receiving streams; retrieving or distributing files; and performing administrative and diagnostic operations. Control messages received from user applications include acknowledgments; error codes; and interactive control requests such as start, stop, pause, reverse, and/or fast forward commands issued during the recording, playback, and/or editing of A/V files.

In response to AVSS service requests, the AVSM 160 may directly perform one or more functions, and/or issue a set of high-level requests to its associated AVSCs 120 in accordance with AVSC capabilities and availability. In terms of directly-performed

LINKPLAY EXHIBIT 1004
Page 3076 of 4822

functions, the AVSM 160 establishes and manages request processing sessions with user applications; allocates A/V file names and tracks A/V file location and usage information; allocates AVSC resources to request processing sessions based upon AVSC capabilities and availability; allocates AVSC resources to file copy and move transactions; forwards file copy and move requests to campus or remote systems in the event that an A/V file does not reside upon one of the premises AVSCs 120; and issues requests to the A/V network 30 to establish A/V communication and perform A/V network services (such as conferencing) for user applications.

Any given AVSC 120 that receives a high-level request from the AVSM 160 carries out the request by performing a set of operations in accordance with its own methods and constraints. In this manner, the AVSM 160 may be designed to avoid "micro-managing" the details of AVSC 120 operation. Depending upon the nature of a high-level request, an AVSC 120 may allocate and lock an encoding, decoding, or transcoding resource; allocate file storage space; execute an encoding, decoding, or transcoding procedure; perform a file transfer or copy operation to or from a local or remote target destination or source; delete a file; or report request processing, message, storage device, and/or encoding, decoding, or transcoding resource status to the AVSM 160.

4.7 AVSS Services

The AVSM 160 manages or provides a variety of AVSS services, including session management, file management, device management, and administrative services. Details of such services and the manners in which they are implemented are described hereafter.

4.7.1 Session Management Services

From the perspective of application programs executing on user workstations 40 or other computers coupled to the CMCE 10, the premises AVSM 160 provides access to services associated with premises, campus, and/or wide-area AVSS resources. Any given application program typically generates one or more graphical windows plus menus and control, list, dialog, and/or other graphical boxes that form a portion of a Graphical User Interface (GUI), as described in detail below. The application program is responsive to

LINKPLAY EXHIBIT 1004
Page 3077 of 4822

selections that a user graphically indicates, in a manner well understood by those skilled in the art.

Particular user selections indicate that the user requires AVSS-related services. In response to such selections, the application program issues service requests to the associated premises AVSM 160. Upon receiving a service request, the AVSM 160 establishes a request processing session to provide or oversee the provision of the required service. During the session, the AVSM 160 may directly perform one or more operations, and/or the AVSM 160 may generate one or more requests directed to premises AVSCs 120, the premises A/V network manager 34, and/or campus or remote AVSMs 160. Additionally, premises, campus, or remote CMCE elements involved in the session may generate requests directed to each other and/or the application program.

Figure 10 is a block diagram illustrating session communication protocols utilized in one embodiment of the present invention. Essentially any number of instances of a wide variety of application programs communicate with their corresponding premises AVSM 160 via an AVSM client protocol. A given premises AVSM 160 communicates with its associated AVSCs 120 via an AVSC client protocol, and the premises A/V network manager 34 via an A/V network client protocol. Cross-premises AVSM-to-AVSM communication is based upon an AVSM peer protocol, while analogous A/V network manager communication is based upon an A/V network peer protocol.

Prior to gaining access to A/V processing resources or services, a workstation user must execute an AVSS access program, which issues a login request to the AVSM 160. As described in detail below, the login request includes a user ID, which establishes the user's identity to the AVSM 160 for purposes of file ownership and access permission checking. The user ID also provides the AVSM 160 with the A/V network port information necessary for establishing any A/V network connections required. In one embodiment, the login request additionally includes password information, which may be encrypted. When a user no longer requires access to A/V processing resources or services, the user may utilize the AVSS access program to issue a logout request to the AVSM 160, which results in removal of the corresponding user ID from the current user list within the AVSS state memory 174.

LINKPLAY EXHIBIT 1004
Page 3078 of 4822

Following a successful login, an application program executing on the user's workstation 40 may issue a request to the AVSM 160 for A/V processing resources and/or A/V file management services. Upon receipt of a request, the AVSM 160 establishes a session by performing any necessary file access authorization operations as described below; generating a session handle that uniquely identifies the session; adding the session handle to the currently-active session list in the AVSS state memory 174; identifying and allocating appropriate AVSC resources; identifying and allocating appropriate A/V network resources; and issuing a session identification message to the requesting client, where the session identification message includes the session handle. After session establishment, the AVSM 160 manages the session by directly performing operations, and/or issuing or routing requests, control messages, and/or status messages to AVSCs 120, the A/V network manager 34, other AVSMs 160, and/or the application program. The AVSM 160 terminates the session by deallocating resources, issuing a termination message or reply to the application program, and deleting the session handle from the currently-active session list. The AVSM 160 supports at least three types of sessions, namely, 1) viewer tool sessions; 2) maintenance tool sessions; and 3) administrative tool sessions. The characteristics of each of these session types, as well as the operations undertaken by the AVSM 160 in support thereof, are described in detail hereafter.

4.7.1.1 Viewer Tool Sessions

Herein, a viewer tool comprises an application program that controls A/V file recording as well as real-time analog or digital streaming A/V file delivery or viewing via one or more graphical windows, menus, and control, dialog, or other boxes that facilitate user input. The AVSM 160 establishes a viewer tool session when a viewer tool application executing on a workstation 40 or other computer issues a request specifying that either real-time (or near-real-time) analog or digital streaming delivery of an A/V or multimedia file is required (i.e., when a user needs to view an A/V file). Real-time analog or digital streaming delivery is required, for example, during A/V file recording, editing, and/or playback, and involves A/V signal delivery to a display device (such as a

LINKPLAY EXHIBIT 1004
Page 3079 of 4822

workstation monitor or television) under the direction of the viewer tool application program.

To initiate a viewer tool session, the AVSM 160 determines whether the request received from the viewer tool application specifies an existing A/V or multimedia file. If so, the AVSM 160 performs file access authorization operations, as described in detail below, to verify whether the viewer tool user should be granted access to the file. If the file access authorization operations are unsuccessful, the AVSM 160 issues a reply to the requesting viewer tool application indicating proper access authorization does not exist, and the viewer tool session ends.

Upon successful completion of the file access authorization operations, the AVSM 160 generates a session handle, and adds it to the currently-active session list. The AVSM 160 next identifies an AVSC 120 that includes A/V processing resources capable of performing the operations required by the viewer tool application. In one embodiment, the AVSC selection is performed in accordance with the resource allocation operations described in detail below. For the creation (i.e., recording) of a new A/V or multimedia file, the AVSM 160 selects an AVSC 120 upon which sufficient storage space and an appropriate type of encoder exist. For the playback or editing of an existing A/V or multimedia file, the AVSM 160 selects an AVSC 120 upon which a copy of the file resides and appropriate decoding or transcoding resources are present.

After identifying an appropriate AVSC 120, the AVSM 160 sends a request to the AVSC 120 to allocate the resources required for performing the operation(s) required by the viewer tool application program. In one embodiment, the AVSM 160 subsequently issues a request to the AV network manager 34 to establish any required A/V network connections. In an alternate embodiment, the viewer tool application program issues a request to the A/V network manager 34 to establish such connections. After AVSC selection/resource allocation and A/V network connection establishment, the AVSM 160 issues a session identification message containing the session handle to the viewer tool application, which indicates that the viewer tool session can proceed.

During the viewer tool session, the AVSM 160 routes control messages received from the viewer tool application to the selected AVSC 120, thereby facilitating AVSC

LINKPLAY EXHIBIT 1004
Page 3080 of 4822

performance of open-file, play, adjust-playback-rate, stop, seek, pause, rewind, fast-forward, record, save-file, or other operations as supported by the viewer tool application program. The AVSM 160 may additionally issue status messages to the viewer tool application, which can include video frame numbers and/or running or elapsed time. In one embodiment, the AVSM 160 polls the AVSC 120 for status information, and forwards it to the viewer tool application program.

A multimedia file comprises one or more types of files and/or references to files, where such files may include textual, graphical, image, audio, and/or video information as well as commands or event sequences for generating or rendering such information. The multimedia file may further include temporal correlation data specifying one or more manners in which its constituent files and/or file references are associated in time. In the context of the present invention, a multimedia file comprises one or more omnifiles, where an omnifile is defined as a metafile plus a corresponding pointer file. The metafile includes one or more of the following types of files:

1) A/V files;
2) audio-only files;
3) video-only files;
4) bitmap files;
5) postscript files;
6) graphics files;
7) text files;
8) application files; and
9) synchronization files.

The synchronization files associate events or commands occurring within the context of one or more application programs with particular time references, which may be A/V file frame numbers. Synchronization files can include application startup event files, text event files, window event files, graphics rendering event files, and shareboard event files, as further described below. In one embodiment, a synchronization file can include a priority level for each command or event specified therein. The various files that comprise the metafile may be stored across one or more workstations 40, AVSCs 120,

LINKPLAY EXHIBIT 1004
Page 3081 of 4822

servers, or other computers coupled to the CMCE 10. The pointer file associated with a metafile contains pointers or references to each element within the metafile.

In one embodiment, the synchronization files are stored on AVSCs 120. In the event that the viewer tool session involves multimedia file recording, the AVSM 160 transfers application program synchronization information or commands received from the viewer tool application to the AVSC 120 such that the synchronization information can be saved in a synchronization file. For multimedia file playback and/or editing, the AVSM 160 requests the synchronization file from the appropriate AVSC 120. To ensure temporal consistency during playback operations, the AVSM 160 polls the AVSC 120 allocated to serving the current viewer tool session for frame number or time-based A/V file data, and issues synchronization messages to the viewer tool application at appropriate times or intervals, in accordance with a currently-specified playback frame rate or speed. The synchronization messages contain event information, commands, and command priority levels specified within the synchronization file. Upon receipt of a synchronization message, the viewer tool application executes commands specified therein, or facilitates the execution of such commands via their transfer to other application programs, thereby regenerating application events and/or the presentation of textual, graphics, image, or other multimedia content information at appropriate times while A/V file playback occurs.

The AVSM 160 could alternatively transfer the synchronization file to the viewer tool application, poll the AVSC 120 for frame-number or time-based A/V file data, and issue frame or time status messages to the viewer tool. The viewer tool application could then execute or issue the appropriate commands to facilitate multimedia synchronization upon receipt of frame or time status messages that correspond to frame numbers or times specified within the synchronization file. In other embodiments, the synchronization files themselves could be stored on the AVSM 160 rather than on AVSCs 120, or the synchronization files could normally reside upon the user workstation 40.

In one embodiment, in the event that the viewer tool session involved A/V file or multimedia file editing and file edits or changes were saved while maintaining the filename, the AVSM 160 performs modification propagation operations as described in

LINKPLAY EXHIBIT 1004
Page 3082 of 4822

detail below to ensure that the file changes propagate to each copy of the A/V file stored on the AVSCs 120.

The operations described above pertain to within-premises viewer tool sessions. Non-premises viewer tool sessions occur in an analogous manner. In one embodiment, in the event that a viewer tool application requires real-time/near real-time analog or digital streaming A/V file delivery from a non-premises AVSS 100, the viewer tool application issues a request to the appropriate non-premises AVSM 160. In response, the non-premises AVSM 160 establishes a viewer tool session in the manner described above, such that analog A/V signals are exchanged via the premises workstation 40, the premises codec gateway 38, the second WAN 39, and the non-premises codec gateway 38 that serves the non-premises AVSS 100.

A viewer tool provides a workstation user with a Graphical User Interface (GUI) for controlling A/V file recording and playback operations. ***Figure*** 11 is a block diagram showing an exemplary recording control GUI 500. The recording control GUI 500 comprises a graphical window that includes a menu bar 502, a title box 504, a recording control panel 506, and a time panel 520. The menu bar 502 provides a workstation user with access to conventional types of submenus, including file, options, and help submenus. User-selectable operations from within the options submenu include login and change-password operations. User-selectable operations from within the file submenu include the following:

1) New – opens a new A/V file for recording, in the manner described below;

2) Open – invokes a browser tool that facilitates opening and subsequent playback of an existing A/V file, as described below;

2) Save – saves an A/V file with a given filename;

3) Properties – facilitates entry and editing of properties associated with the A/V file being recorded, such as encoding type and a natural language title; and

4) Exit – terminates the viewer tool session.

The title box 504 identifies the current title of an A/V file under consideration. The recording control panel 506 provides a set of user-selectable buttons for controlling recording operations. These buttons include a start recording button 510; a pause

LINKPLAY EXHIBIT 1004
Page 3083 of 4822

recording button 512; a stop or end recording button 514, and an erase or discard recorded information button 516. The time panel 520 graphically indicates a current recording length or time via the provision of a slider bar 522 and a time box 524.

Figure 12 is a block diagram showing an exemplary playback control GUI 600. For ease of understanding, common elements shown in ***Figures*** 11 and 12 are denoted with common reference numbers. The playback control GUI 600 comprises a graphical window that includes a menu bar 502, a title box 504, a playback control panel 606, and a time panel 520. The menu bar 502 enables user to select from among the operations described above with reference to ***Figure*** 11. Similarly, the title box 504 indicates the current title of the A/V file under consideration. The playback control panel 606 includes a plurality of user-selectable buttons that facilitate playback control operations, including a rewind to beginning button 610; a skip back button 612 that facilitates moving or jumping backward to an earlier position within an A/V file, where the earlier position corresponds to a predetermined or user-definable time increment, for example, 3 seconds; a playback start button 614; a playback pause button 616; a skip forward button 618 that facilitates moving or jumping backward to a later position within an A/V file, where the later position corresponds to a predetermined or user-definable time increment, for example, 3 seconds; and an erase file button 619. In a manner analogous to that for the recording control GUI 500, the time panel 520 graphically indicates a current playback length or time via the provision of a slider bar 522 and a time box 524. Those skilled in the art will understand that the recording and playback control GUIs 500, 600 provide workstation users with a graphical interface that is visually consistent with controls found on common devices such as Video Cassette Recorders (VCRs).

In response to certain submenu selections, such as Open within the file submenu, the viewer tool transfers control to or invokes functionality within or associated with a browsing or administrative tool to provide a workstation user with a GUI for browsing and selecting A/V files and/or performing A/V file administration operations as described in detail below. ***Figure*** 13 is a block diagram showing an exemplary browsing control GUI 700. The browsing control GUI 700 comprises a dialog box that includes a file source listbox 702 and file source control buttons 704; display format control buttons 706;

LINKPLAY EXHIBIT 1004
Page 3084 of 4822

a file information listbox 710; a file name box 722; a file type listbox 724; and operation control buttons 730, 732, 734. The file source listbox 702 and associated file source control buttons 704 facilitate user-identification of a file source from which A/V files can be selected for consideration. The display format control buttons 706 facilitate user-selection of file information display formats, where available display formats include display-by-icon and display-by-attributes. For each A/V file for which the workstation user is identified as an owner, the file information listbox 710 displays file information in accordance with the selected file parameter display format. Such information can include a filename; a file location; a natural language title; a description; file authorship; and playback duration. The file name box 722 facilitates user-entry of a specific filename, and the file type listbox 724 facilitates browsing operations directed to files of a specific type. Finally, the operation control buttons include an open button 730, the selection of which results in the opening of a specified or selected A/V file for playback operations; a cancel button 732; and a help button 734.

4.7.2 Maintenance Tool Sessions

The AVSM 160 establishes a maintenance tool session in response to a request that specifies that one or more user-based file maintenance operations are required. In the context of the present invention, a maintenance tool comprises an application program executing on a user workstation 40 or other computer, and which generates one or more graphical windows, menus, and/or boxes that facilitate particular A/V file maintenance operations in response to user input. In one embodiment, the user input may be in the form of text, graphical selections, or graphical drag-and-drop sequences.

As described in detail below, the AVSM 160 supports many file management operations, a subset of that comprises file maintenance operations that are directly available to workstation users. In one embodiment, the file maintenance operations include file access list maintenance; file expiration date maintenance; file renaming; file transfer in the form of either copying or moving; file deletion; user-specific file listing; user-specific archival operations; and file attribute queries. Each of these operations is described in detail below in the context of AVSM file management operations. Those

LINKPLAY EXHIBIT 1004
Page 3085 of 4822

skilled in the art will recognize that the file maintenance operations could include additional or fewer types of operations in another embodiment.

The AVSM 160 initiates a maintenance tool session by performing A/V file access authorization operations as described in detail below. If access authorization is unsuccessful, the AVSM 160 terminates the session. Otherwise, the AVSM 160 generates a session handle, adds it to the currently-active session list, and issues a session identification message to the maintenance tool application. The AVSM 160 determines whether access to one or more campus and/or remote AVSMs 160 is required to process the file maintenance request, and, if so, issues requests to allocate the required campus and/or remote resources. Once such resources are allocated, the campus and/or remote AVSMs 160 issue a reply to the premises AVSM 160 specifying AVSC network address information if required. The AVSM 160 subsequently determines what premises AVSCs 120 are required to process the file maintenance request, and issues requests to the premises AVSCs 120 to allocate the appropriate resources and begin performing the required operation.

During the performance of file maintenance operations, the AVSM 160 issues status requests to premises and/or non-premises AVSCs 120, which reply with status information. The AVSM 160 then sends status messages to the file maintenance tool application accordingly. Such status messages may indicate, for example, a percentage of the operation that has been completed or an estimated amount of time remaining until completion. Once the file maintenance operation is complete, the AVSM 160 issues a session termination reply to the maintenance tool.

Some file maintenance operations, such as A/V file renaming or copying, may be requested within the context of a viewer tool session. That is, one or more aspects of maintenance tool functionality may be incorporated into a viewing tool. In such situations, the AVSM 160 performs or oversees the performance of the file maintenance operation within the context of the viewer tool session, in a manner essentially identical or analogous to that described for the file maintenance tool session. Moreover, in one embodiment, a maintenance tool may provide for the invocation of a particular viewer

LINKPLAY EXHIBIT 1004
Page 3086 of 4822

tool in response to user input, in which case the AVSM 160 establishes a viewer tool session as a result.

4.7.3 Administrative Tool Sessions

Herein, an administrative tool comprises an application program that generates, presents, and/or manages administrative information on a workstation-based or other display device in response to user input. The AVSM 160 establishes an administrative tool session in response to a request for user-accessible information stored in the AVSS database. Such information may include, for example, user account and password information, or particular data within the event statistics file.

Upon receiving an administrative information request, the AVSM 160 generates a session handle, and adds it to the currently-active session list. The AVSM 160 then transfers data corresponding to the request to the administrative tool. Upon receiving the data, the administrative tool generates and/or presents administrative information to the user. If administrative data is user-modifiable, as could be the case for user passwords, the AVSM 160 changes the data within the AVSS database in response to a request received from the administrative tool. Some administrative data may be non-user-modifiable, such as current AVSS utilization; AVSS usage history for the user, which may include average session length, by session type; and/or current charges billed to the user.

4.7.4 File Management Services

As indicated above, the AVSM 160 provides or oversees the provision of a variety of file management services. A subset of the file management services comprise the user-accessible file maintenance operations, while other file management services are performed by the AVSM 160 on an internal-only basis. In one embodiment, the file management services include the following:

4.7.4.1 Access Authorization Services

AVSM software objects maintain the access privilege list within each A/V file's parameter table in the AVSS database. The access privilege list comprises a list of user IDs that specifies which users "own" the A/V file, where ownership implies basic access rights to the A/V file. The extent of A/V file access rights may depend upon whether a

LINKPLAY EXHIBIT 1004
Page 3087 of 4822

user falls into a particular category, such as the A/V file's author or other category as detailed below. In one embodiment, an A/V file access request generated by a client application program includes a user ID, a filename, and a password. Upon receiving such a request, the AVSM 160 determines whether the user ID associated with the request is specified within the A/V file's access privilege list. If so, the AVSM 160 proceeds with processing the file access request. If the user ID is not specified within the access privilege list, the AVSM 160 compares the password specified within the A/V file access request with that in the file parameter table. If the passwords match, the AVSM 160 adds the user ID to the access privilege list, and proceeds with request processing operations. Otherwise, the AVSM 160 issues a reply containing an error code to the client application program to indicate that A/V file access has been denied.

In one embodiment, the presence of a reserved user ID in an A/V file's access privilege indicates that the file may be accessed by any user, that is, the file is generally or publicly accessible. One embodiment of the present invention also provides for an administrative "super-user" defined to have essentially unlimited access privileges for any A/V file. In response to a request for A/V file access received from an application program associated with the super-user, the AVSM 160 proceeds with request processing operations.

4.7.4.2 Access List Maintenance

In response to an access list maintenance request received from a client application program, the AVSM 160 selectively adds user IDs to or deletes user IDs from an A/V file's access permission list. The AVSM 160 performs access list maintenance operations only for users having a user ID specified in the access permission list, subject to the whether a user is the A/V file author, an owner, or a super-user. The AVSM 160 allows an A/V file author or a super-user to modify the access permission list as required. The AVSM 160 permits any given A/V file owner that is not the file's author to remove their user ID from the access permission list.

In one embodiment, the AVSM 160 adds the user ID of a user who authors an A/V file to the A/V file's access permission list by default. The AVSM 160 may add other user IDs to the access permission list in response to client application program

requests. For example, an A/V- or multimedia-enabled e-mail application program as described in detail below allows message authors to include an A/V or multimedia file as a MIME-based message attachment. The AVSM 160 adds the user ID corresponding to each message recipient to the A/V file's access permission list, as directed by the e-mail application program. The AVSM 160 may also or alternatively add a message recipient's user ID to the access permission list in response to an A/V file access request specifying a user ID, a filename, and a valid password.

4.7.4.3 User-Specific File Listing

An application program associated with or executing on behalf of a user may issue a file listing request to the AVSM 160. In turn, the AVSM 160 performs a directory sort operation, and replies to the application program with a list identifying each A/V file for which the user is specified as an owner in the file's access permission list. In an alternate embodiment in which the AVSM 160 maintains a user-specific file list in the AVSS database as described above, the AVSM 160 could reply to the application program with data from the user-specific file list.

The application program may subsequently display a set of filenames for which the user is specified as an owner. In one embodiment, the file listing request may indicate that data for publicly-available files is also required. If so, the AVSM 160 provides such data to the requesting application program.

4.7.4.4 User-Specific File Archival

An application program may additionally issue a file archival request to the AVSM 160, where the file archival request identifies one or more A/V files owned by the user associated with the requesting application program for which archival operations are required. The AVSM 160 subsequently manages archival operations in which the identified A/V files or copies thereof are stored in a premises or non-premises user ID-based archive. The archival operations include one or more file transfer services as detailed below.

4.7.4.5 Attribute Query

In response to an attribute request received from an owner of an A/V file, the AVSM 160 provides attribute information for the A/V file, such as file author, size,

LINKPLAY EXHIBIT 1004
Page 3089 of 4822

encoding type, playback duration, current file age, time and date the owner last accessed the file, file expiration date (i.e., access expiration date) for the owner, and/or date of most-recent modification.

4.7.4.6 File Transfer Services

File transfer services supported by the AVSM 160 include a) A/V file copy and/or move (i.e., relocate) operations to and/or from a target destination or source; b) file import or export operations between a workstation 40 or other computer and an AVSC 120; and c) digital file publishing operations between an AVSC 120 and a server external to the AVSS 100 in which the AVSC 120 resides. With regard to A/V file copy or move operations, the target destination or source may be premises-, campus-, or wide-area-based. Since any given AVSS 100 may be linked to another AVSS 100 via a LAN or a WAN, both intra- and inter-AVSS file transfers are possible in the context of the present invention. ***Figure*** 14 is a block diagram of an exemplary networked AVSS organization. In the exemplary networked AVSS organization, a first AVSS 100.1 through kth AVSS 100.2 serve a common campus, and hence are coupled to a common data network 20. A remote AVSS 100.3 is coupled to the common data network 20 via a WAN 29. Those skilled in the art will recognize that the exemplary networked AVSS organization shown in ***Figure*** 14 can be generalized to accommodate essentially any required network organization or geographic scaling.

Each AVSS 100 in ***Figure*** 14 includes a set of AVSCs 120, where each AVSC 120 includes a number of disk drives or other devices for storing A/V files. The number of AVSCs 120 can vary from one AVSS 100 to another, as can the number and type of storage devices from one AVSC 120 to another. Relative to intra-AVSS file transfers, A/V files can be transferred from one storage device to another either a) within the same AVSC 120, that is, "within cell" (WC); or b) across different AVSCs 120 in the same AVSS 100, or "cell to cell" (CC). In terms of A/V file transfers directed from an AVSC storage device within one AVSS 100 to an AVSC storage device within another AVSS 100, such transfers can occur either a) within the common campus, or local area (LA); or b) on a wide area (WA) basis, between a common campus AVSS 100 and the remote AVSS 100.

LINKPLAY EXHIBIT 1004
Page 3090 of 4822

In response to a request that requires a cell-to-cell or within-cell A/V file transfer, the AVSM 160 selects and allocates a source and a destination premises AVSC 120, and issues a request to the one of the selected AVSCs 120 to perform the required transfer. In one embodiment, the request may specify a copy-from, move-from, copy-to, or move-to operation. After the file transfer is complete, the AVSC 120 that fulfilled the request issues a reply to the AVSM 160.

If a file transfer request requires A/V file transfer between a premises and a non-premises AVSS 100, the premises AVSM 160 issues a request to the non-premises AVSM 160 to obtain network address information for an associated non-premises AVSC 120 that can participate in fulfilling the request. Upon receiving such information from the non-premises AVSM 160, the premises AVSM 160 selects and allocates an appropriate premises AVSC 120, and issues a request to the premises AVSC 120 to perform the required file transfer operation. If the file transfer request specifies a copy-to or move-to operation, the A/V file will be transferred from the premises AVSC 120 to the non-premises AVSC 120 accordingly. If the file transfer request specifies a copy-from or move-from operation, the A/V file will be copied or moved from the non-premises AVSC 120 to the premises AVSC 120, respectively. After the file transfer is complete, the premises AVSC 120 issues a reply to the premises AVSM 160. The premises AVSM 160 then issues a notification to the non-premises AVSM 160 to indicate that the file transfer session is complete.

Figure 14 also shows a workstation 40 coupled to the common data network 20. File import operations involve the transfer of a digital file from a workstation 40 or other computer to an AVSC 120, while file export operations involve the transfer of a digital file from an AVSC 120 to a workstation 40 or other computer. In response to a file import request received from a client application program, the AVSM 160 selects and allocates an available AVSC 120 having sufficient storage space, and issues a request to the AVSC 120 to retrieve the file from the workstation 40 or other computer from which the import request originated. Similarly, in response to a file export request specifying an A/V file, the AVSM 160 selects and allocates an AVSC 120 upon which a copy of the file resides, and issues a request to the AVSC 120 to transfer the file to the workstation 40 or

LINKPLAY EXHIBIT 1004
Page 3091 of 4822

other computer from which the export request originated. Upon completion of file transfer operations corresponding to an import or/and export request, the AVSC 120 issues a reply to the AVSM 160.

File publishing operations involve the transfer of a digital file from an AVSC 120 to a server. The server may be essentially any computer system external to the AVSS 100 in which the AVSC 120 resides, for example, an intranet server system 60 or a server coupled to the internet 80. ***Figure*** 14 also shows an intranet server system 60 coupled to the common data network 20. In response to a publish file request that specifies a filename, a target file format, and a network or IP address of a server, the AVSM 160 selects and allocates an AVSC 120 upon which a copy of the file resides. The AVSM 160 then issues a publish request to the AVSC 120, which performs any necessary file format conversion, and transfers the file to the specified server. Upon completing the transfer, the AVSC 120 issues a reply to the AVSM 160.

In general, an AVSC 120 can directly perform both A/V file encoding, decoding, and/or transcoding operations, as well as file transfer operations. Any particular AVSC 120 will have a maximum number of simultaneously-active file transfers it can perform while still delivering high-quality encoding, decoding, or transcoding performance. Thus, file transfer needs must be balanced against encoding, decoding, and/or transcoding needs. Typically, encoding, decoding, and/or transcoding operations are given higher priority than file transfer operations. In one embodiment, an administrative application program facilitates defining a first-priority and a second-priority AVSC bus and disk bandwidth partition. The first-priority bandwidth partition is reserved for performing encoding/decoding/transcoding operations, while the second-priority partition is reserved for file transfer operations. As file transfer requests are allocated to an AVSC 120, it services them immediately using bandwidth within the second-priority partition. If the AVSC 120 receives a file transfer request that it cannot immediately service, the AVSC 120 places the request into the request queue within its state memory 154. Requests in this queue are served as more bandwidth becomes available within the second-priority partition. In one embodiment, the request queue is implemented as a First-In, First-Out (FIFO) queue in a conventional manner. The request queue could alternatively be

implemented as a prioritized queue, and/or as a queue from which requests are serviced in accordance with current resource availability. Alternatively, requests that cannot be immediately serviced could be terminated and a reply sent back indicating resources are not currently available thereby allowing the AVSM 160 and/or client application to look for some other means to service the request.

4.7.4.7 File Replication

An A/V file should be present upon multiple AVSCs 120 to ensure that the file can be independently accessed by a reasonably large number of users at any given time. Thus, after A/V file creation (i.e., after a new A/V file has been created and saved), or after the transfer of a new A/V file onto one of the premises AVSCs 120, the AVSM 160 initiates file replication operations. During the file replication operations, copies of the A/V file are stored on a plurality of premises AVSCs 120. In one embodiment, copies of associated multimedia synchronization files are also replicated.

The particular file replication strategy employed at any given time depends upon several factors, including the number of premises AVSCs 120 present; the performance capabilities of each AVSC 120; AVSS internal network bandwidth; and/or current user demand for AVSC resources. In an AVSS 100 employing a single AVSC 120, file replication may not be performed, or limited file replication may occur between separate storage devices on the AVSC 120. For an AVSS having a small number of AVSCs 120 -- for example, three AVSCs 120 serving approximately 100 users -- full replication may be performed. In this case, the AVSM 160 automatically oversees the copying of a new A/V file to each AVSC 120 within the AVSS 100. The copying operations are performed in the manner described above for cell-to-cell file transfer operations. In general, the A/V file is not replicated to a non-premises AVSC 120 in the absence of an application-level request to do so.

Depending upon the performance capabilities of each AVSC 120, as well as the number of premises AVSCs present, full replication may significantly limit AVSC availability for performing other operations. Thus, for an AVSS 100 employing a medium to somewhat large number of premises AVSCs 120 -- for example, five AVSCs 120 serving a few hundred to several hundred users -- the AVSM 160 manages partial file

LINKPLAY EXHIBIT 1004
Page 3093 of 4822

replication operations. During the partial file replication operations, the AVSM 160 oversees A/V file copying to a subset of the premises AVSCs 120, where the copying operations are performed in the manner described above for cell-to-cell file transfer. The A/V file is not generally replicated to a non-premises AVSC unless an application program issues a request requiring such replication. The selection of a particular AVSC 120 as a member of the aforementioned subset may be based upon the number of A/V files, available file storage space, or the number of decoders and transcoders residing upon the AVSC 120.

At some point, partial replication becomes untenable as the number of premises AVSCs 120 within the AVSS 100 becomes very large. In such situations, a subset of the AVSCs 120 are utilized as streaming A/V file servers, and the other AVSCs function as staging points for encoding, decoding, and/or transcoding operations. In one embodiment, the AVSM 160 manages file replication operations for the streaming A/V file servers in a manner analogous to that described above.

4.7.4.8 File Renaming

In response to a name change request, the AVSM 160 changes an A/V file's name as specified in the corresponding file parameter table, and subsequently oversees modification propagation operations as detailed below to change the name of each copy of the A/V file stored across the premises AVSCs 120. In one embodiment, the AVSM 160 permits a name change only if requested by an application program corresponding to a super-user. The AVSM 160 may further permit a name change request for the A/V file's author in the event that the author is currently the sole A/V file owner.

4.7.4.9 File Deletion

In response to a file deletion request, the AVSM 160 directs the deletion of each copy of the A/V file from the premises AVSCs 120 by issuing a delete request to each AVSC 120 upon which a copy of the A/V file resides. In one embodiment, the AVSM 160 permits file deletion only if requested by an application program associated with a super-user. The AVSM 160 may further permit A/V file deletion by the A/V file's author in the event that the author is currently the sole A/V file owner. In one embodiment, the AVSM 160 additionally deletes or oversees the deletion of any multimedia synchronization

LINKPLAY EXHIBIT 1004
Page 3094 of 4822

files associated with the deleted A/V file. Additionally, the AVSM 160 may issue deletion requests to non-premises AVSM 160 and servers 60 and 80 to which it has transferred the A/V file.

4.7.4.10 Modification Propagation

As previously indicated, multiple copies of any given A/V file may reside upon the premises AVSCs 120. The AVSM 160 may permit a given user, such as a super-user or an A/V file's author, to edit or change the file's name or contents after its creation. Each premises copy of the A/V file must be updated to reflect such modification. Thus, in one embodiment, after an A/V filename or contents change, the AVSM 160 issues a sequence of copy-to requests to the AVSC 120 upon which the modified file resides to ensure name and contents consistency between the A/V file copies stored on the AVSCs 120. In an alternate embodiment, the AVSM 160 could issue copy-from requests. Additionally, the AVSM 160 may issue file modification requests to non-premises AVSM 160 and servers 60 and 80 to which it has transferred the A/V file.

4.7.4.11 Expiration Operations

An A/V file's access privilege list includes for each user ID an expiration date and/or time past which the user ID expires and is no longer valid for obtaining file access. For each A/V file parameter table in the AVSS database, the AVSM 160 determines whether any user IDs specified in the access privilege list have expired. If so, the AVSM 160 removes the expired user IDs from the access privilege list. Once no unexpired user IDs remain in the access privilege list, the AVSM 160 issues a file deletion request to each AVSC 120 upon which a copy of the A/V file resides. After each such AVSC 120 has deleted the A/V file, the AVSM 160 deletes the A/V file parameter table from the AVSS database. Alternatively, the AVSM 160 may log expired files and require deletion by an application program associated with a super-user.

4.7.4.12 Database Consistency Operations

The AVSM 160 periodically performs consistency checks to ensure that the file parameter tables within the AVSS database 136 accurately reflect the current premises AVSC file environment. In one embodiment, the AVSM 160 periodically queries each premises AVSC 120 to obtain data for each A/V file stored thereupon, where such data

LINKPLAY EXHIBIT 1004
Page 3095 of 4822

may include an A/V file's name, size, age, encoding format, storage location or address, or other information. In the event that an AVSC 120 returns a filename that is not specified in the AVSS database, the AVSM 160 generates a file parameter table corresponding to the filename, and adds it to the AVSS database. If no AVSC 120 returns a particular filename specified in the AVSS database, the AVSM 160 deletes the corresponding file parameter table from the AVSS database. If the file data returned by an AVSC 120 indicate a discrepancy in file size, age, encoding format, or other information, the AVSM 160 may update the AVSS database accordingly. The AVSM 160 may additionally log any discrepancies, as well as the type of action(s) undertaken in response to each discrepancy. Finally, if a particular AVSC 120 returns a filename specified in the AVSS database, but for which the file parameter table indicates that the file is not stored on this AVSC 120, the AVSM 160 updates the file parameter table with the storage location of the A/V file upon the AVSC 120 in question.

4.7.5 Device Management Services

The AVSM 160 additionally performs device management services, which include AVSC scheduling and allocation services. AVSS resource scheduling and allocation is subject to several:

interacting resource constraints, which include the following:

1. AVSM multiple-thread resources (i.e., simultaneous use by multiple sessions):
    a. CPU capacity:
        i. encode session management
        ii. decode session management
        iii. file transfer session management
    b. Bus capacity
    c. Networking port capacity
2. AVSC multiple-thread resources (i.e., simultaneous use by multiple sessions):
    a. CPU capacity:
        i. encode session management
        ii. decode session management
        iii. file transfer session management

LINKPLAY EXHIBIT 1004
Page 3096 of 4822

iv. stream session management
v. audio-video stream multiplexing processes
vi. audio-video stream demultiplexing processes
vii. encode session processes
viii. decode session processes
ix. file transfer processes
x. send stream processes
xi. receive stream processes

b. Bus capacity
c. Disk controller/disk bus (e.g. SCSI) capacity
d. Disk read/write/seek capacity
e. Networking port capacity

3. AVSC single-threaded resources (i.e., can be used by only one session at a time):
a. encoder hardware and corresponding encode session(s)
b. decoder hardware and corresponding decode session(s)

Any type of session has a specific number of specific resources required to execute the session. Some embodiments may grant service requests without regard to details of the current utilization of the resources; however, this may result in underutilization of the resources or in unacceptable performance shortcomings or failure modes if the resource capacities are exceeded. One embodiment would base request acceptance on the needs of the request in view of the available unused capacity of the system. An example of how this can be accomplished is described hereafter.

In some cases, resource requirements may be exact. In other cases, resource requirements may be in the form of upper bounds, estimated averages, or averages biased upwards to provide safety margins. These session-type resource usage estimates may or may not include additional "padding" to allow for processing overhead (task switching, paging, etc.). In the absence of padding in the session-type resource usage estimates, overhead can be provide for by either

· reducing the maximum value of available resource to a value that safely accounts for worse-case incurred overhead at full operating capacity, or

LINKPLAY EXHIBIT 1004
Page 3097 of 4822

· introducing a specific linear or nonlinear overhead function dependent on the number of various sessions active or projected within the AVSS 100.

Some of the resource usage estimates could be firm fixed values; for example:

· each "encode" session could be designed to require a fully dedicated encoder hardware subsystem;

· each "encode" session could be designed to allocate a fixed block of disk memory (said size dependent on the application) and attempts to record beyond this would be prevented; or

· every "decode" session will have knowledge of the exact the file size

It may be more likely, however, that resource usage estimates would not be firm; for example:

· all active encode sessions could share a slightly smaller pool of encoders; and

· all encode session requirements could be handled with averages and attempts to record beyond this would be allowed.

To further illustrate an example of resource allocation process, the following table provides an example framework for resource requirements for several different types of sessions. Other embodiments of course may have different session types and resource requirements.

Type of Resource		Type of Session					
		Encode	Decode	Put File	Get File	Send Stream	Receive Stream
AVSM CPU capacity	Encode session management	M^{E}_{SM}					
	Decode session management		M^{D}_{SM}				
	file xfr session management			M^{PF}_{SM}	M^{GF}_{SM}		
	stream session management					M^{SS}_{SM}	M^{RS}_{SM}
AVSM Bus capacity		M^{E}_{B}	M^{D}_{B}	M^{PF}_{B}	M^{GF}_{B}	M^{SS}_{B}	M^{RS}_{B}
AVSM Networking port capacity		M^{E}_{N}	M^{D}_{N}	M^{PF}_{N}	M^{GF}_{N}	M^{SS}_{N}	M^{RS}_{N}
AVSC CPU capacity	encode session management	C^{E}_{SM}					

LINKPLAY EXHIBIT 1004
Page 3098 of 4822

Type of Resource	Type of Session					
	Encode	Decode	Put File	Get File	Send Stream	Receive Stream
decode session management		C^{D}_{SM}				
file xfr session management			C^{PF}_{SM}	C^{GF}_{SM}		
stream session management					C^{SS}_{SM}	C^{RS}_{SM}
multiplexing processes	C^{E}_{MP}					
demultiplexing processes		C^{D}_{MP}				
encode session processes	C^{E}_{CP}					
decode session processes		C^{D}_{CP}				
file transfer processes			C^{PF}_{TP}	C^{GF}_{TP}		
send stream processes					C^{SS}_{SP}	
receive stream processes						C^{RS}_{SP}
AVSC Bus capacity	C^{E}_{B}	C^{D}_{B}	C^{PF}_{B}	C^{GF}_{B}	C^{SS}_{B}	C^{RS}_{B}
AVSC Disk controller/disk bus capacity	C^{E}_{DC}	C^{D}_{DC}	C^{PF}_{DC}	C^{GF}_{DC}	C^{SS}_{DC}	C^{RS}_{DC}
AVSC Disk read/write/seek capacity	C^{E}_{RW}	C^{D}_{RW}	C^{PF}_{RW}	C^{GF}_{RW}	C^{SS}_{RW}	C^{RS}_{RW}
AVSC Networking port capacity	C^{E}_{N}	C^{D}_{N}	C^{PF}_{N}	C^{GF}_{N}	C^{SS}_{N}	C^{RS}_{N}
AVSC hardware encoder hardware	C^{E}_{EH}					
decoder hardware		C^{D}_{DH}				

Each type of resource, or group of resources, has a finite maximum performance value that can be provided by the AVSM 160 or AVSC 120 environments within the AVSS 100. This finite maximum value will be termed a "capacity bound."

5

M^{MAX}_{SM} = AVSM CPU session management maximum capacity

M^{MAX}_{B} = AVSM Bus maximum capacity

M^{MAX}_{N} = AVSM Networking port maximum capacity

C^{MAX}_{SM} = AVSC CPU session management maximum capacity

LINKPLAY EXHIBIT 1004
Page 3099 of 4822

C_{P}^{MAX} = AVSC CPU process maximum capacity

C_{B}^{MAX} = AVSC Bus maximum capacity

C_{DC}^{MAX} = AVSC Disk Controller/disk bus maximum capacity

C_{RW}^{MAX} = AVSC Disk read/write/seek maximum capacity

C_{N}^{MAX} = AVSC Networking port maximum capacity

C_{EH}^{MAX} = AVSC Encoder hardware maximum capacity

C_{DH}^{MAX} = AVSC Decoder hardware maximum capacity

The amount of resource required by a potential combination of granted sessions would be determined, in many cases, by the sum of the total allocated resource estimates. Overhead can be included, for example, via the incorporation of a plurality of overhead constants that include the folloiwng:

M_{SM}^{OH} = AVSM CPU session management overhead

M_{B}^{OH} = AVSM Bus overhead

M_{N}^{OH} = AVSM Networking port overhead

C_{SM}^{OH} = AVSC CPU session management overhead

C_{P}^{OH} = AVSC CPU process overahead

C_{B}^{OH} = AVSC Bus overhead

C_{DC}^{OH} = AVSC Disk Controller/disk bus overhead

C_{RW}^{OH} = AVSC Disk read/write/seek overhead

C_{N}^{OH} = AVSC Networking port overhead

LINKPLAY EXHIBIT 1004
Page 3100 of 4822

Overhead may also be treated as a linear function of the number of various session types. Incorporation of overhead results in the supportable operating states forming a lattice of points within a convex hull defined by the intersection of hyperplanes defined by the resource capacity bound inequalities. For example, with:

n^E = number of encoding sessions
n^D = number of decoding sessions
n^{PF} = number of put file transfer sessions
n^{GF} = number of get file transfer sessions
n^{SS} = number send streaming sessions
n^{RS} = number of receive streaming sessions

then the supportable operating states in an implementation could be given by the following constraints:

For each AVSM 160 (typically one per AVSS 100),

- AVSM CPU capacity (session management)

$$[\, n^E M^E_{SM} + n^D M^D_{SM} + n^{PF} M^{PF}_{SM} + n^{GF} M^{GF}_{SM} + n^{SS} M^{SS}_{SM} + n^{RS} M^{RS}_{SM} \,] + M^{OH}_{SM} \le M^{MAX}_{SM}$$

- AVSM bus capacity:

$$[\, n^E M^E_{B} + n^D M^D_{B} + n^{PF} M^{PF}_{B} + n^{GF} M^{GF}_{B} + n^{SS} M^{SS}_{B} + n^{RS} M^{RS}_{B} \,] + M^{OH}_{B} \le M^{MAX}_{B}$$

- AVSM Networking port capacity:

$$[\, n^E M^E_{N} + n^D M^D_{N} + n^{PF} M^{PF}_{N} + n^{GF} M^{GF}_{N} + n^{SS} M^{SS}_{N} + n^{RS} M^{RS}_{N} \,] + M^{OH}_{N} \le M^{MAX}_{M}$$

For each AVSC 120 (at least one, typically multiple, per AVSS 100):

- AVSC CPU capacity (session management):

$$[\, n^E C^E_{SM} + n^D C^D_{SM} + n^{PF} C^{PF}_{SM} + n^{GF} C^{GF}_{SM} + n^{SS} C^{SS}_{SM} + n^{RS} C^{RS}_{SM} \,] + C^{OH}_{SM} \le C^{MAX}_{SM}$$

- AVSC CPU processes capacity:

$$[\, n^E C^E_{MP} + n^D C^D_{MP} + n^E C^E_{CP} + n^D C^D_{CP} + n^{PF} C^{PF}_{TP} + n^{GF} C^{GF}_{TP} + n^{SS} C^{SS}_{SP} + n^{RS} C^{RS}_{SP} \,]$$

LINKPLAY EXHIBIT 1004
Page 3101 of 4822

$$+ \; C^{OH} \leq C_{CP}^{MAX}$$

- AVSC Bus capacity:

$$[\, n^{E} C_{B}^{E} \; + \; n^{D} C_{B}^{D} \; + \; n^{PF} C_{B}^{PF} \; + \; n^{GF} C_{B}^{GF} \; + \; n^{SS} C_{B}^{SS} \; + \; n^{RS} C_{B}^{RS} \,] \; + \; C_{B}^{OH} \leq C_{B}^{MAX}$$

- AVSC Disk controller/disk bus capacity:

$$[\, n^{E} C_{DC}^{E} + n^{D} C_{DC}^{D} \; + \; n^{PF} C_{DC}^{PF} \; + \; n^{GF} C_{DC}^{GF} \; + \; n^{SS} C_{DC}^{SS} \; + \; n^{RS} C_{DC}^{RS} \,] \; + \; C_{DC}^{OH} \leq C_{DC}^{MAX}$$

- AVSC Disk read/write/seek capacity:

$$[\, n^{E} C_{RW}^{E} + n^{D} C_{RW}^{D} \; + \; n^{PF} C_{RW}^{PF} \; + \; n^{GF} C_{RW}^{GF} \; + \; n^{SS} C_{RW}^{SS} \; + \; n^{RS} C_{RW}^{RS} \,] \; + \; C_{RW}^{OH} \leq C_{RW}^{MAX}$$

- AVSC Networking port capacity:

$$[\, n^{E} C_{N}^{E} \; + \; n^{D} C_{N}^{D} \; + \; n^{PF} C_{N}^{PF} \; + \; n^{GF} C_{N}^{GF} \; + \; n^{SS} C_{N}^{SS} \; + \; n^{RS} C_{N}^{RS} \,] \; + \; C_{N}^{OH} \leq C_{N}^{MAX}$$

- AVSC hardware:

$$[\, n^{E} C_{EH}^{E} \,] \leq \; C_{EH}^{MAX}$$

$$[\, n^{D} C_{DH}^{D} \,] \leq \; C_{DH}^{MAX}$$

Here it is assumed that each encode session is represented with either a fixed or an average amount of disk capacity. Depending upon design choices, one embodiment may enforce a per-session file size limit, while another embodiment may permit file size overruns.

Upon receiving one or more requests for services, the AVSM 160 would, for example, perform the following evaluations:

· serialize any multiple pending requests so the following evaluations are done fully separately in sequence for each pending request;

· check if accepting the new request would not exceed the AVSM's own capacity requirements; and

· check all AVSCs 120 within that AVSS 100 to see if at least one AVSC 120 could accept the new request without exceeding the AVSM's own capacity requirements (this

LINKPLAY EXHIBIT 1004
Page 3102 of 4822

could be done via either the AVSM's own image of the AVSC states or by polling at least one AVSC 120).

- If more than one AVSC 120 can support the request:
- select one according to a conventional procedure such as first available, round robin, or operating lifetime;
- allocate the session to the AVSC 120, ensuring directly or indirectly notification of the requesting application and the invocation of appropriate network connections; or
- otherwise, deny the request.

Clearly there are many other possible implementations. The range of possible implementations could also include methods for ensuring that no single type of application or type of session monopolizes the AVSS 100 at the exclusion of other types of applications or sessions. In the previous illustrative example, this could be accomplished by adding further inequalities to the list above. For example, the total number of file transfer sessions can be limited by adding the constraint

$$n^{PF} + n^{GF} \leq n^{F:MAX}$$

whered $n^{F:MAX}$ is the maximum number of simultaneous file transfers allowed.

The device management services additionally include AVSC validation operations, through which the AVSM 160 queries each AVSC 120 to determine the number, type, and capabilities of the AVSC's storage, encoding, decoding, and transcoding resources. The AVSM 160 uses the query results to update the AVSS database if needed, and further logs any discrepancies as well as the types of operations undertaken in response.

4.7.6 Administrative Services

The AVSM 160 also performs or manages the performance of administrative services that include system performance and utilization monitoring; system diagnostics; event capture operations; billing operations; password maintenance operations; and user maintenance operations.

LINKPLAY EXHIBIT 1004
Page 3103 of 4822

4.8 Software Architecture And System Interface

4.8.1 Communication Class Hierarchy

In the present invention, the manners in which AVSS-external and AVSS-internal clients may access AVSM, AVSC, and A/V network manager functionality are defined by a set of software communication class hierarchies. ***Figure*** 15 is a block diagram showing a high-level AVSS communication class hierarchy 200 for one embodiment of the present invention. As shown in ***Figure*** 15, the base class for this hierarchy is referred to as "Comm," which serves as a basis for AVSS-external and AVSS-internal client communication. The Comm class also serves as a basis for external and internal A/V network manager client communication, and is described in detail in U.S. Patent No. 5,617,539.

Several subclasses derived from the Comm class encapsulate particular aspects of AVSS functionality for AVSS-external and AVSS-internal client access. These subclasses include the following:

1) AvnmComm - the AvnmComm class provides access to functionality supported through its parent class, as well as client access to A/V network manager services.

1) AvsComm - the AvsComm class provides access to AVSS functionality that includes login authorization; A/V encoding, decoding, and transcoding; file transfer between AVSCs 120; file importing and exporting; file and directory management; and AVSC configuration and status reporting.

2) AvscComm - the AvscComm class provides access to the functionality supported through its parent class, and in addition provides client and server access to additional AVSC-related services as detailed below.

3) AvscAppComm - the AvscAppComm class provides a client interface for accessing file import and export services.

4) AvsmComm - the AvsmComm class provides access to the functionality supported through its parent class, and in addition provides client and server access to the AVSM file and device management services described above.

5) AvsmAppComm - the AvsmAppComm class provides event handlers that support client application program access to AVSS services.

LINKPLAY EXHIBIT 1004
Page 3104 of 4822

6) AvsmAdminComm - the AvsmAdminComm class provides AVSS-external and AVSS-internal clients access to AVSS administrative functionality.

4.8.2 AVSM Class Relationship

The classes of AVSM objects resident within the AVSM object memory 172 have a hierarchical class relationship. ***Figure*** 16 is a block diagram of an AVSM class relationship 250 organized in accordance with one embodiment of the present invention. The AVSMManager class comprises comprises a container class for supporting session establishment, session management, and session termination operations. The AVSMManager class further includes parameters and methods for facilitating AVSM operation, i.e., implementing the AVSM 160. The AVSMManager class also contains parameters specifying AVSS operational characteristics, such as AVSC session limits, plus port and connection information for premises AVSCs 120, the A/V network manager 34, and known non-premises AVSSs 100. In one embodiment, the AVSMManager class includes methods that facilitate communication between lower-level classes within the AVSM class hierarchy 250.

An AVSMComm class provides a software infrastructure for implementing a server interface that manages client interactions, where clients include user application programs and/or a non-premises AVSS 100. The AVSMComm class also provides a basis that allows the AVSM 160 to behave as a client relative to non-premises AVSSs 100. The AVSMComm class comprises methods for retrieving requests, building objects that are subclassed according to request type, and dispatching requests. The AVSMComm class further comprises methods for receiving replies, building objects subclassed according to reply type, and issuing replies. The AVSMComm class includes a reference to an AVSMRequestList class and an AVSMReplyList class.

The AVSMRequestList class facilitates the implementation of a queue for requests received from AVSS-external clients, as well as a queue for requests directed to non-premises AVSSs 100. The AVSMRequestList class comprises a reference to an AVSMRequest class, which comprises request parameters as well as methods for accessing data associated with a request. In one embodiment, the AVSMRequest class is subclassed according to request type. Thus, a request queue may include a first reference

LINKPLAY EXHIBIT 1004
Page 3105 of 4822

to an AVSMOpenRequest object that corresponds to an A/V file open request received from a client; and a second reference to an AVSMPlayRequest object that corresponds to an A/V file playback request received from a client.

The AVSMReplyList class facilitates the implementation of a queue for replies directed from the AVSM 160 to external clients, as well as a queue for replies received from non-premises AVSSs 100. The AVSMReplyList class comprises a reference to an AVSMReply class, where the AVSMReply class comprises reply parameters as well as a set of methods for accessing data associated with a reply. In a manner analogous to that for the AVSMRequest class, the AVSMReply class may be subclassed in accordance with reply type.

An AVNMComm class provides a software infrastructure for implementing an A/V network client interface that includes parameters and methods for issuing requests to and receiving replies from the A/V network manager 34. The AVNMComm class includes a reference to an AVNMRequestList class and an AVNMReplyList class. The AVNMRequestList class facilitates the implementation of a queue for requests directed to the A/V network manager 34, and comprises a reference to an AVNMRequest class. The AVNMRequest class includes parameters corresponding to a request directed to the A/V network manager 34, as well as methods for accessing data associated with the request. The AVNMRequest class may be subclassed in accordance with the type of request directed to the A/V network 30. The AVNMReplyList class facilitates the implementation of a queue for replies received from the A/V network manager 34, and includes a reference to an AVNMReply class. The AVNMReply class comprises parameters associated with replies received from the A/V network manager 34, as well as methods for accessing any data associated therewith. The AVNMReply class may be subclassed in accordance with the types of a replies that may be received from the A/V network manager 34.

An AVSCComm class comprises a software infrastructure for implementing an AVSC client interface, and comprises parameters and methods for issuing requests to and receiving replies from AVSCs 120. The AVSCComm class includes a reference to an AVSCRequestList class, which facilitates implementation of a queue for requests directed

LINKPLAY EXHIBIT 1004
Page 3106 of 4822

to AVSCs 120. The AVSCRequestList class includes a reference to an AVSCRequest class, which includes parameters and methods for generating requests directed to an AVSC 120, and which may be subclassed in accordance with the types of requests that may be directed to AVSCs 120. The AVSCComm class also includes a reference to an AVSCReplyList class, which facilitates implementation of a queue for replies received from AVSCs 120. The AVSCReplyList class includes a reference to an AVSCReply class, which comprises parameters and methods associated with receipt of AVSC replies. The AVSCReply class may be subclassed in accordance with AVSC reply types.

A CellList class serves as a basis for a list that describes each premises AVSC 120. The CellList includes a reference to a Cell class, which provides a basis for describing an AVSC 120. The Cell class includes parameters that specify a network address, bandwidth capabilities, and a number of simultaneous requests that the AVSC 120 can process. The Cell class further includes references to a ListDisks class, a ListEncoders class, a ListDecoders class, and a ListTranscoders class.

The ListDisks class facilitates creation of a list that references a Disk class, which supports an AVSC storage device description. The Disk class contains parameters that include such information as device transfer rate, seek time, and available storage space. The Disk class additionally contains methods for accessing and modifying such parameters.

The ListEncoders class facilitates the implementation a list that references an Encoder class, which supports an AVSC encoder description. Encoder class parameters include such information as encoder type, bandwidth and I/O limitations, supported media formats, and input signal characteristics. The Encoder class additionally contains methods for accessing and modifying these parameters.

The ListDecoders class facilitates the implementation of a list that references a Decoder class, which supports an AVSC decoder description. Decoder class parameters include such information as decoder type, bandwidth and I/O limitations, supported media formats, and output signal characteristics. The Decoder class additionally contains methods for accessing and modifying the Decoder class parameters.

LINKPLAY EXHIBIT 1004
Page 3107 of 4822

The ListTranscoders class facilitates the implementation of a list that references a Transcoder class, which supports an AVSC transcoder description. Transcoder class parameters include such information as transcoder type, bandwidth and I/O limitations, supported media formats, and input/output signal characteristics. The Transcoder class additionally contains methods for accessing and modifying the Transcoder class parameters.

A LoginList class provides an implementation basis for a list describing currently logged-in users. The LoginList class references a Login class that supports a user description via parameters that include a user ID, a privilege level, a login date and time stamp, and a set of channel handles currently allocated to the user ID, where each channel handle corresponds to AVSC resources allocated to processing user requests as described in detail below.

A SessionList class serves as basis for a list that describes currently active sessions. The SessionList class references a Session class, which provides a foundation for describing any particular session. The Session class is subclassed according to session type, where the subclasses include CopySession, PublishSession, ImportSession, ExportSession, AVSession, FileSession, and ListSession classes.

The CopySession class provides a framework for describing and tracking a file copy operation between two AVSCs 120. The CopySession class includes parameters such as IP or network addresses for the AVSCs 120 involved, a source and a destination filename, and channel handles corresponding to resources reserved on the AVSCs 120 for performing the file copy operation. In one embodiment, the CopySession class parameters also include a flag that specifies whether a copy operation or a transfer operation is required, where the transfer operation comprises file copying followed by deletion of the file from an origin AVSC 120. The CopySession class further includes methods for accessing its parameters.

The PublishSession class comprises parameters and methods that describe and monitor A/V and/or multimedia document publishing operations, through which A/V and/or multimedia files may be copied to non-AVSS servers. PublishSession class parameters include IP or network addresses plus login data for a target or destination

LINKPLAY EXHIBIT 1004
Page 3108 of 4822

server, as well as a filename. PublishSession class methods includes data conversion and data transfer methods. The PublishSession class may be subclassed in accordance with particular media formats, where such subclasses include parameters such as destination server type and transcoding requirements. Exemplary subclasses include a NetShowPublish class and a RealMediaPublish class.

The ImportSession class comprises parameters and methods that describe and monitor the reading of A/V file data into a target AVSC 120 from a source client application (or source workstation 40 or other computer upon which the client application is executing), and includes parameters specifying the target AVSC's IP or network address and an AVSC channel handle. The ExportSession class provides a basis for writing A/V file data from the AVSC 120 into a target client application program (or target workstation 40 or other computer upon which the client application program is running), and includes parameters specifying the source AVSC's IP or network address and an AVSC channel handle.

The AVSession class comprises parameters and methods for initiating and managing A/V operations, which may include record, play, stream-in, stream-out, pause, resume, status, and stop operations.

The FileSession class comprises parameters and methods that facilitate writing data to and reading data from an AVSC 120. The ListSession class comprises parameters and methods that facilitate the generation of a list that specifies files and file attributes stored upon one or more AVSCs 120.

An AVMFileList class provides a basis for implementing the AVSM file parameter list within the AVSS database. The AVMFileList includes a reference to an AVMFile class, which includes parameters describing an A/V file, such as filename, file author or creator, a file title and description, file size, encoding type, and playback duration. The AVMFile class further includes a reference to an OwnerList class, which in turn references an Owner class. The Owner class provides a basis for describing an A/V file owner, and includes parameters such as user ID, a last access date and time, and an ownership expiration date. The OwnerList class includes methods for adding and deleting owners.

LINKPLAY EXHIBIT 1004
Page 3109 of 4822

Those skilled in the art will understand that the AVSM class hierarchy 250 can be readily modified or extended to provide or accommodate access to additional types of AVSM functionality.

4.8.3 AVSC Class Relationship

In a manner similar to that for the AVSM objects, the AVSC objects resident within the AVSC object memory 152 are hierarchically interrelated. ***Figure*** 17 is a block diagram of an AVSC class relationship 300 organized in accordance with one embodiment of the present invention. In the AVSC class relationship 300, a MediaResource class serves as a container class for implementing multithreaded AVSC functionality, and includes methods for thread management. An AVSCComm class provides a software infrastructure for implementing an AVSC communication port. The AVSCComm class provides a basis for implementing a server that receives requests and issues replies, as well as a client that issues requests and receives replies. Typically, an AVSC 120 will act as a server the majority of the time; however, the AVSC 120 may act as a client during AVSC-to-AVSC file copy or transfer operations. The AVSCComm class comprises parameters and methods for retrieving requests, building objects that are subclassed according to request type, and dispatching requests. In one embodiment, request dispatch occurs via the creation of a thread of execution corresponding to a request. The AVSCComm class includes references to an AVSCRequestList class and an AVSCReplyList class.

The AVSCRequestList class provides a basis for implementing a request queue, and references an AVSCRequest class. The AVSCRequest class facilitates request description, and may be subclassed in accordance with request type. The AVSCReplyList class provides a basis for implementing a reply queue, and references an AVSCReply class, which may be subclassed according to reply type.

A MessageThreadList class facilitates implementation of a list of threads that are currently active within the AVSC 120, and references a MediaThread class. The MediaThread class provides a framework for implementing a thread of execution that processes and fulfills a request. The MediaThread class includes thread parameters such as a thread ID and thread state information. The parameters may further include a

LINKPLAY EXHIBIT 1004
Page 3110 of 4822

filename, a password, and an IP or network address. The MediaThread class references a MediaMessage class. The MediaMessage class comprises parameters and methods that direct thread execution, and is subclassed according to thread type.

In the context of the AVSC class relationship 300, a "channel" comprises a group of resources allocated for performing a particular type of operation, such as an A/V file playback or record operation. A ChannelList class facilitates the implementation of a list that identifies or describes a current set of channels within the AVSC 120. The ChannelList class references a MediaChannel class, which comprises a container class for representing a channel. The MediaChannel class selectively includes references to an Encoder class, a Decoder class, a Transcoder class, a Buffer class, a DiskAccess class, and a NetAccess class. The Encoder class provides a basis for describing and controlling an encoder within the AVSC 120, and includes parameters specifying encoder type, supported quality levels, whether the encoder is busy or available, and methods that communicate with a set of software drivers that control encoder hardware. The Decoder class serves as a basis for describing and directing the operation of a decoder within the AVSC 120, and includes parameters specifying decoder type, supported quality levels, availability, and methods that communicate with software drivers that control decoder operation. Similarly, the Transcoder class serves as a basis for describing and directing the operation of a transcoder within the AVSC 120, and includes parameters specifying transcoder type, supported source and target coding formats, availability, and methods that communicate with software drivers that control transcoder operation. The Buffer class comprises parameters and methods that facilitate the implementation of memory and/or disk buffers, where such parameters may include buffer type, size, and fill level, and such methods may include program instructions for reading from and writing to the buffer. Finally, the NetAccess class serves as a basis for managing data exchange between the AVSC 120 and the AVSS internal network 110. Each of the Encoder, Decoder, Buffer, DiskAccess, and NetAccess classes may be subclassed in accordance with the particular nature of the resources to which they correspond.

The MediaChannel class also includes a reference to a MediaFile class, which serves as a wrapper for providing an interface to particular functionality and AVSC

LINKPLAY EXHIBIT 1004
Page 3111 of 4822

resources allocated for the channel. The MediaChannel additionally includes a reference to a MessageThreadList class, which facilitates implementation of a list that identifies threads currently active within the scope of the channel.

An EncoderList class within the AVSC class relationship 300 serves as a basis for implementing a list that identifies the encoders within the AVSC 120. The EncoderList class includes a reference to the aforementioned Encoder class. Similarly, a DecoderList class serves as a basis for implementing a list identifying the decoders within the AVSC 120. The DecoderList class includes a reference to the Decoder class. A TranscoderList class facilitates implementation of a list identifying the transcoders within the AVSC 120. The TranscoderList class also references the Transcoder class. A BufferList class facilitates implementation of a list that indicates buffer allocation within the AVSC 120, and includes a reference to the Buffer class. A DiskAccessList class facilitates implementation of a list specifying allocation of storage devices within the AVSC 120, and includes a reference to the Disk class. Finally, a NetAccessList class facilitates implementation of a list of allocated network access resources, and includes a reference to the NetAccess class.

As with the AVSM class relationship 250, those skilled in the art will understand that the AVSC class relationship 300 can be readily modified or extended to provide or accommodate access to additional types of AVSC functionality.

4.9 Client And Server Communication

In the present invention, an AVSS-external or AVSS-internal client requiring a service accessed via an instance of a target object calls for the service by issuing a request. In response, the target object may perform one or more portions of the required service, act as a client and issue one or more requests to other objects, and/or issue one or more messages. Upon service completion, the target object issues a reply. Thus, from a hierarchical perspective, the issuance of a high-level request may give rise to the issuance of one or more lower-level requests and replies, followed by the issuance of a high-level reply. Each such request and reply in one embodiment of the present invention is exchanged in accordance with a conventional Internet Protocol (IP) suite, in a manner readily understood by those skilled in the art.

LINKPLAY EXHIBIT 1004
Page 3112 of 4822

4.10 AVSM Request Categories

The requests understood by the AVSM 160 span several categories, which in one embodiment include the following:

4.10.1 External Client to AVSS Requests

Requests that may be used by AVSS-external clients requiring access to AVSS services include the following:

a) Login - initiates a login session with the AVSS 100 and provides user information and associated A/V network service information to the AVSM 160, where the user information includes a user ID and a password.

b) Logout - terminates user interaction with the AVSS 100.

c) GetFileLimits - returns current administered and physical limits on the length of an A/V file to be encoded.

d) AcquireAVChannel - establishes a session by reserving particular AVSC resources for performing specified types of operations, such as encoding, decoding, transcoding, and/or streaming. Specifies a source AVSC resource and at least one destination AVSC resource to flexibly support multiple modes of operation, which include the following:

1) Playback mode - facilitates playback of an A/V file residing upon an AVSC storage device to a user workstation 40, where such playback can occur at a specified speed or playback rate.
Source resource = disk, destination resource = decoder.

2) Preview mode - facilitates recording setup and record preview operations involving a user workstation 40 (i.e., A/V signal "loop through" without storage of A/V signals to disk).
Source resource = encoder, destination resource = decoder.

3) Record mode - facilitates A/V signal recording from a user workstation 40 to generate an A/V file upon an AVSC storage device.
Source resource = encoder, destination resource = disk.

4) Playback-while-recording mode - facilitates A/V signal recording from a user workstation 40 with simultaneous A/V signal playback to the user workstation 40.
Source resource = encoder, destination resources = disk and decoder.

5) Broadcast-to-network mode - facilitates broadcast of A/V signals from a user workstation 40 to a network (i.e., the AVSS internal network 110 or the data network 20).
Source resource = encoder, destination resource = network.

6) Broadcast-and-playback mode - facilitates A/V signal broadcast from a user workstation 40 to a network, with simultaneous A/V signal playback to the user workstation 40.
Source resource = encoder, destination resources = network and decoder.

7) Broadcast-while-recording mode - facilitates broadcast to a network while performing A/V signal recording from a user workstation 40 to an AVSC storage device.
Source resource = encoder, destination resources = disk and network.

8) Broadcast-with-record-and-playback mode - facilitates broadcast to a network while recording A/V signals from a user workstation 40 to AVSC storage device and simultaneously playing A/V signals back to the user workstation 40.
Source resource = encoder, destination resources = network, disk, and decoder.

9) Stream-in-to-disk mode - facilitates receiving a data stream from a network to an AVSC storage device.
Source resource = network, destination resource = disk.

10) Stream-in-to-playback mode - facilitates data streaming from a network to a user workstation 40 via AVSC decoder.

LINKPLAY EXHIBIT 1004
Page 3114 of 4822

Source resource = network, destination resource = decoder or transcoder.

11) Stream-in-to-playback-with-save mode - facilitates data streaming from a network to a user workstation 40, and saves data being streamed in to an AVSC storage device.
Source resource = network, destination resource = decoder or transcoder and disk.

12) Stream-out-to-network mode - facilitates data streaming from an AVSC storage device to a network at a specified streaming rate.
Source resource = disk, destination resource = network.

13) Stream-out-to-network-and playback-mode facilitates data streaming from an AVSC storage device to a network at a specified streaming rate with simultaneous A/V signal playback to the user workstation 40.
Source resource = disk, destination resource = network and decoder

Those skilled in the art will understand that a buffer resource upon an AVSC may be included as an intermediate destination resource for several types of operational modes, for example, those involving streaming. Those skilled in the art will additionally understand that many other modes of operation exist, for which a session may be established and resources may be reserved in a manner analogous to that described above. When appropriate (i.e., depending upon the specified mode of operation), the AcquireAVChannel request also specifies parameters such as a filename and password; A/V file encoding type; and/or a minimum frame count that indicates an amount of storage space that must be available upon a storage device. After successful resource reservation, a reply corresponding to AcquireAVChannel (i.e., an AcquireAVChannelReply) returns a session handle to a requesting client.

e) AcquireDataChannel - establishes a session and reserves AVSC resources for particular types of file transfer operations, including copy-to, copy-from, file query, and file listing operations. This request includes parameters specifying a mode such as read-only or read-write, as well as parameters such as a filename, password, and a minimum

LINKPLAY EXHIBIT 1004
Page 3115 of 4822

amount of storage space that must be present upon an AVSC storage device. After AVSC resource reservation, a reply corresponding to AcquireDataChannel returns a session handle.

f) AcquireAVFileImportChannel - establishes a session for importing an A/V file into the AVSS 100 and returns a session handle and the network or IP address of the AVSC 120 that will initially receive the A/V file.

g) AcquireAVFileExportChannel - establishes a session for exporting an A/V file from the AVSS 100 to the client and returns the network or IP address of the AVSC 120 upon which the file is stored, plus a session handle.

h) AcquirePublishChannel - establishes a session for publishing an A/V file from an AVSC 120 to one or more non-AVSS target servers. Parameters specified include a filename and password, as well as target server type. A reply associated with this request returns a session handle.

i) PublishFile - publishes a file to a non-AVSS target server. Specified parameters include a filename and password, a target server network or IP address, and target server login and password information.

j) RequestPublishLocations - given a specified filename and password, returns known non-AVSS target server identification and network or IP address to which a file has previously been published, as indicated by the AVSS database.

k) DeletePublishedFile - given a specified filename, file password, and non-AVSS target server network or IP address or target server type, deletes file's publication references for target server or target server types from the AVSS database. If target server login, login password, and any delete authorization information is also specified, DeletePublishedFile additionally attempts to delete the published file from the target server itself.

l) ReleaseAVChannel - releases the resource or resources obtained through the AcquireAVChannel, AcquireAVFileImportChannel, AcquireAVFileExportChannel, and AcquirePublishChannel requests, and deletes a corresponding session handle.

m) AVSMStatusRequest - returns status information to a client during file copy, import, export, and publishing operations when a valid session handle is specified, where

LINKPLAY EXHIBIT 1004
Page 3116 of 4822

such status information may include a number of bytes transferred or an estimated amount of time remaining until operation completion.

n) GetAVFileAttributes -returns attributes for an A/V file, including application program-defined attributes. Certain restrictions may exist with regard to which attributes may be returned. For example, only a super-user and the file author may be allowed to view a complete owner list. A user may be allowed to view file attributes if the user is specified as an owner for the file, and any attributes returned are specific to that user.

Super-users and the file authors can access the following operations in one embodiment:

o) SetAVFileAttributes - sets the mutable attributes of an A/V file, including application program-defined attributes.

p) AddOwner - causes the AVSM 160 to add a specified user ID to the access privilege list corresponding to an A/V file.

q) RemoveOwner - causes the AVSM 160 to remove a specified user ID from the access privilege list corresponding to an A/V file.

r) RequestAVFile - requests an A/V file from a non-premises source AVSS 100. If AV file is present within the domain of the AVSM 160 that receives this request, a file transfer procedure retrieves the A/V file from the non-premises source AVSS 100.

4.10.2 Client To AVSC Via AVSM Requests

Many of the requests the AVSM 160 issues to an AVSC 120 are variants of those sent to the AVSM 160 by client application programs. Requests that the AVSM 160 may receive from external clients and forward or pass to an AVSC 120 for servicing include the following:

a) Open - depending upon a mode specifier that indicates whether an A/V file is to be operated upon relative to A/V resource processes (MEDIA mode) or file transfer processes (DATA mode), opens or creates an A/V file and prepares it for encoding, decoding, and/or transcoding, streaming in, or streaming out (MEDIA mode), or data transfer (DATA mode). Upon completion of this request, a reply is issued that includes a session handle.

b) Close - closes a file associated with a specified session handle.

c) Read - for a file opened in DATA mode, reads a specified number of bytes from the file and transfers them to the requesting client.

d) Write - for a file opened in DATA mode, transfers and writes a specified number of bytes to the file.

e) Seek - for a file opened in DATA mode, positions the file at a specified byte. For a file opened in media mode, positions the file at a specified frame.

f) Play - causes the AVSC 120 to signal a decoder to begin processing a file and feeding the output to the A/V network 30. Parameters include playback speed or rate as well as direction and processing length indicators that allow this request to implement fast-forward, rewind, or similar operations.

g) Record - causes the AVSC 120 to signal an encoder to begin saving A/V input to the file. A record length limitation may be specified.

h) StreamIn - causes the AVSC 120 to stream data from the AVSS internal network 110 into a buffer, data storage, and/or decoding resource, in accordance with resources reserved for a particular channel.

i) StreamOut - causes the AVSC 120 to stream data from a buffer, data storage, and/or encoding resource to the AVSS internal network 110 at a specified data rate, in accordance with resources reserved for a particular channel.

j) Stop - causes suspension of a current encoding, decoding, transcoding, or streaming process.

k) Status - returns current file mode and frame number if in MEDIA mode or byte position if in DATA mode.

4.10.3 AVSM To AVSC Requests

Requests the AVSM 160 issues to a target AVSC 120 in relation to A/V file access and file transfer include the following:

a) AcquireChannel - requests the AVSC 120 to create a channel reserving a specified encoder, decoder, and/or transcoder for use by an AVSS client. This request may additionally be used to reserve file transfer bandwidth for impending file transfer or streaming operations. Request completion results in the return of a channel handle that may be used to identify the allocated resources in subsequent operations.

b) ReleaseChannel - indicates to the AVSC 120 that the resources associated with a specified handle are no longer required and shall be released.

c) SetMediaSetup - sets or updates configuration information for the target AVSC 120. This request is typically employed in the event that the AVSC's hardware configuration has been modified.

d) GetMediaSetup - returns configuration and usage information for the target AVSC 120.

e) GetDriveInfo - returns information about a specified storage device.

f) GetMediaStatus - returns current AVSC utilization and state information.

g) Rename - renames a specified file.

h) Delete - deletes a specified file.

i) CopyTo - initiates the transfer of a specified file to a specified destination AVSC 120.

j) CopyFrom - initiates the transfer of a specified file from a specified source AVSC 120.

4.10.4 AVSC To AVSM Requests

An AVSC 120 may send informational requests to its managing AVSM 160 to indicate state changes associated with file transfer operations, A/V operation completion, and/or general hardware status. Such informational requests include the following:

a) PlayEnd - indicates that a current play operation has completed, due to end-of-file or a play-length being reached.

b) RecordEnd - indicates that a current record operation has completed.

c) HardwareError - indicates a hardware failure associated with a specified encoder, decoder, transcoder, or storage device.

4.10.5 AVSM To AVSM Requests

In addition to receiving external requests from client application programs, a premises AVSM 160 may receive requests from a non-premises AVSM 160. Typically, such requests correspond to file transfer operations, and include the following:

a) RequestAVFileSource - issued by the premises AVSM 160 to the non-premises AVSM 160 upon receipt of a "RequestAVFile" from a client. The non-premises AVSM

LINKPLAY EXHIBIT 1004
Page 3119 of 4822

160 returns i) the network or IP address of an AVSC 120 that can serve as a file transfer source; ii) attributes for a specified A/V file; and iii) a channel handle.

b) NotifyAVFileSource - sent upon successful completion of an A/V file transfer between a premises and a non-premises AVSC 120, and includes the user ID and source channel handle, such that the non-premises AVSM 160 can release source AVSC resources that had been reserved for the file transfer.

4.10.6 AVSM Administration Requests

Several requests, some of which may be accessible only to a super-user, may be used to perform AVSS configuration and administrative operations, and include the following:

a) SetPrivileged - establishes a super-user session, through which the super-user can access many administrative functions. This request requires a password.

b) GetAVSMLogInfo - returns information stored in the AVSM event log.

c) ClearAVSMLogInfo - clears the AVSM event log.

d) SetAVSMLogLevel - specifies types of requests and events to log, as well as a level of detail for logging, and a maximum AVSM event log size.

e) ListAVSCs - returns a list of AVSCs 120 managed by the AVSM 160.

f) GetAVSCInfo - returns configuration and utilization information for a particular AVSC 120.

g) GetAVSCLogInfo - returns information stored in the AVSC event log, message queue.

h) ClearAVSCLogInfo - clears a specified AVSC's event log.

i) SetAVSCLogLevel - indicates to a specified AVSC 120 types of requests and events to log, as well as a level of detail for logging, and a maximum AVSC event log size.

j) SetAVNMInfo - updates A/V network manager information maintained within the AVSM 160.

k) AddAVSC - adds configuration information corresponding to a newly-added premises AVSC 120 to the AVSS database. Configuration information for the new AVSC 120 is obtained by querying the AVSC 120.

LINKPLAY EXHIBIT 1004
Page 3120 of 4822

l) RemoveAVSC - removes information corresponding to an AVSC 120 no longer in the premises group from the AVSS database.

m) ListAVSSs - returns a network or IP address for each non-premises AVSS 100 about which the AVSM 160 is aware.

n) GetAVSSInfo - returns routing and/or connection setup information for a particular non-premises AVSS 100 known to the AVSM 160.

o) SetAVSSInfo - sets routing and/or connection setup information for a specified non-premises AVSS 100. In one embodiment, this information pertains to the host AVSM 160 for the specified AVSS 100.

p) AddAVSS - adds a specified non-premises AVSS 100 to those known to the AVSM 160. In one embodiment, routing and connection information for the newly-added AVSS 100 is specified through SetAVSSInfo.

q) RemoveAVSS - removes a non-premises AVSS 100 from those know to the AVSM 160.

r) SetAVFileLimits - sets a maximum A/V file length allowed during encoding operations.

Those skilled in the art will recognize that an AVSS could support additional or fewer requests in an alternate embodiment.

4.11 AVSC Request Categories

The AVSC 120 provides support for a variety of requests, including those in the following categories:

4.11.1 Allocation and Authorization Requests

a) AuthorizationResource - ensures that a client requesting services provided by the AVSC 120 is either an AVSM 160 or a client providing a valid channel handle.

b) AllocateChannel - creates a channel to which specified resources are allocated, and returns a channel handle.

c) ChangeChannel - for a channel identified by a channel handle, adds or deletes specified resources to or from channel, respectively.

LINKPLAY EXHIBIT 1004
Page 3121 of 4822

d) ReleaseChannel - for a channel identified by a channel handle, frees channel resources (i.e., channel resources are returned to "not busy" or "not allocated" status), removes channel from channel list, and releases channel handle.

4.11.2 File Management Requests

a) CopyFromChannel - establishes a copy session with a source AVSC 120 and performs file copy operations.

b) CopyToChannel - establishes a copy session with a target AVSC 120 and performs file copy operations.

c) RenameChannel - renames a specified file.

d) DeleteChannel - deletes a specified file.

e) FindFilesFirstChannel - performs file directory operations, returning first k files.

f) FindFilesNextChannel - performs file directory operations, returning next n files.

g) GetFileInfoChannel - opens a specified file and returns information about the media content of the file; this request may be employed, for example, to validate file import operations or for consistency checking.

h) PublishDigitalFileChannel - performs format conversion, login operations, and file copy operations to publish a specified file to another server in a specified format.

4.11.3 Media Requests

a) OpenChannel - opens a specified A/V file on a specified channel.

b) ReadChannel - reads from an A/V file on a specified channel.

c) WriteChannel - writes to an A/V file on a specified channel.

d) PlayChannel - plays an A/V file on a specified channel at a specified frame rate.

e) StreamInChannel - streams an A/V file into a specified channel.

f) StreamOutChannel - streams an A/V file from a specified channel at a specified data rate.

g) RecordChannel - records an A/V file on a specified channel.

h) PauseChannel - pauses A/V file operations on a specified channel.

i) StopChannel - stops A/V file operations on a specified channel.

LINKPLAY EXHIBIT 1004
Page 3122 of 4822

j) ResumeChannel - resumes A/V file operations on a specified channel.

k) SeekChannel – moves to a given position within an A/V file on a specified channel.

l) StatusChannel - returns information about current status of operations on a specified A/V file, including current frame or position within the file, on a specified channel.

m) CloseChannel - closes an A/V file on a specified channel.

4.11.4 Administrative Requests

a) InitializeResource - resets hardware, rebuilds lists, and reallocates objects.

b) GetMediaSetupResource - returns information about encoders, decoders, transcoders, and storage devices within the AVSC 120.

c) SetMediaSetupResource - sets information about encoders, decoders, transcoders, and/or storage devices within the AVSC 120.

d) GetMediaStatusResource - returns current status of AVSC resources, including a number of channels open; a number of encoders, decoders, transcoders in use; storage device usage and available storage space; and buffer and internal network usage.

e) HangupResource - performs failure recovery or system violation recovery operations.

f) ShutdownResource - performs shutdown operations, after which AVSC 120 can be reinitialized.

4.12 Request Sequence Examples

The description that follows details the flow of requests and replies generated in response to exemplary AVSS-external requests corresponding to several operational categories, which are defined as follows:

4.12.1 Session Establishment and Resource Reservation

Figure 18 is an exemplary request sequence diagram for the AcquireAVChannel request described above. In ***Figure*** 18, and AVSS client (i.e., a client application program) issues an AcquireAVChannel request specifying a mode and mode-specific parameters to its premises AVSM 160. The AVSM 160 determines a set of AVSC resources required to fulfill the channel acquisition request, and identifies an appropriate

LINKPLAY EXHIBIT 1004
Page 3123 of 4822

AVSC 120. The AVSM 160 then issues an AcquireChannel request to the AVSC 120, specifying a set of resources to be reserved for a channel. Upon reserving the resources, the AVSC 120 issues an AcquireChannelReply to the AVSM 160, which includes a channel handle and a return code that indicates whether resource reservation was successful. The AVSM 160 subsequently issues an AcquireAVChannelReply that includes a session handle to the client. The client may subsequently use the session handle to request particular operations in accordance with the types of AVSC resources associated with the channel handle. For any given client, the AVSM 160 may map one or multiple session handles to a single channel and handle. The AVSM 160 is responsible for mapping session handles to the appropriate channel handle during subsequent operations. Those skilled in the art will understand that analogous operations occur for each particular type of channel acquisition request described above.

4.12.2 A/V File Management

Since the AVSM 160 is responsible for maintaining the AVSS database, it must perform some processing in response to client requests affecting A/V files. Since the A/V files are physically stored upon the AVSCs 120, the AVSM 160 passes or forwards requests to the appropriate AVSC(s) 120 following such processing. In general, for AVSS clients requiring access to an A/V file, the AVSM 160 examines the AVSS database to determine whether a specified file exists, and upon which AVSCs 120 the file resides. When an A/V file is created, the AVSM 160 adds a file parameter list to the AVSS database, and selects an available AVSC 120 to support an encoding session. The AVSM 160 issues a Record request to the selected AVSC 120, updates its internal data, and upon completion of the recording operation issues a RecordReply to the client.

Figure 19 is an exemplary request sequence diagram for the "Open" request described above. In ***Figure*** 19, an AVSS client (i.e., a client application program) issues an Open request specifying a session handle, a filename, and a mode (i.e., MEDIA, DATA read/write or read only) to its premises AVSM 160. The AVSM 160 subsequently maps the session handle to the appropriate AVSC channel handle, and forwards the open request to the AVSC 120 corresponding to the channel handle. After opening the A/V file, the AVSC 120 issues an OpenReply to the AVSM 160, which includes a channel handle

LINKPLAY EXHIBIT 1004
Page 3124 of 4822

and a return code that indicates whether the open operation was successful. The AVSC 160 in turn replies to the AVSS client with a session handle and a return code.

Figure 20 is an exemplary request sequence diagram for the above-mentioned "Close" request. In ***Figure*** 20, an AVSS client issues a Close request that specifies a session handle and a filename to the AVSM 160. The AVSM 160 maps the session handle to the appropriate channel handle, and issues or forwards the close request specifying the channel handle and the filename to the appropriate AVSC 120. Upon completing the close operation, the AVSC 120 issues a close reply to the AVSM 160, which includes the channel handle and a return code. The AVSM 160 maps the channel handle to the appropriate session handle, and forwards the close reply to the AVSS client.

Figure 21 is an exemplary request sequence diagram for an A/V file "Delete" request. In response to receipt of a delete request that specifies a session handle and a fully-qualified filename from an AVSS client, the AVSM 160 determines upon which AVSCs 120 copies of the A/V file reside. The AVSM 160 then maps the session handle to a first channel handle, and forwards the delete request to a first AVSC 120 upon which such a copy resides. Once the A/V file copy is deleted, the first AVSC 120 issues a DeleteReply to the AVSM 160. The AVSM 160 then maps the session handle to a second channel handle, and forwards the AVSS client's delete request to a second AVSC 120 upon which a copy of the A/V file resides. The second AVSC 120 performs the required delete operation, and issues a DeleteReply to the AVSM 160. The AVSM 160 continues this process by mapping the session handle to a next channel handle, and forwarding the AVSS client's delete request to a next AVSC 120 storing a copy of the A/V file after receiving a DeleteReply from an AVSC 120 that had just completed the delete operation, and so on, until copies of the A/V file no longer reside upon the AVSCs 120. After the relevant AVSCs 120 have deleted the appropriate A/V file copies, the AVSM 160 updates the AVSS database 176 to reflect the deletion, and issues a DeleteReply to the AVSS client.

4.12.3 Encoding, Decoding, Or Transcoding Control

Figure 22 is an exemplary request sequence diagram for a "Record" request. Upon receiving a Record request from an AVSS client specifying a session handle and possibly a filename depending upon operational mode, the AVSM 160 maps the session

handle to the appropriate channel handle. The AVSM 160 forwards the mapped record request to the AVSC 120, which issues commands to the encoder associated with the channel handle. When the recording session is complete, the AVSC 120 issues a RecordReply that specifies the channel handle and a return code to the AVSM 160. The AVSM 160 then maps the channel handle to a session handle, and forwards the RecordReply to the AVSS client, where the RecordReply includes the session handle and a return code.

4.12.4 File Transfer

Externally-generated file transfer requests may result in the generation of multiple intra-AVSS requests and replies, depending upon the location of the requested file. ***Figure*** 23 is an exemplary request sequence diagram corresponding to a request for a file transfer from a non-premises AVSS 100. To take advantage of the file transfer capabilities of the AVSS 100, a client must have a source AVSM name or address as well as a filename. The client supplies these parameters to its premises AVSS 100 via a RequestAVFile request.

The premises AVSM 160 determines whether the required file is locally available, that is, whether it is stored upon one of the premises AVSCs 120. If not, the premises AVSM 160 issues a RequestAVFileSource request to the non-premises AVSS 100. Via an AcquireChannel request, the non-premises AVSM 160 allocates resources in an AVSC 120 that can serve as a source for the file transfer. After the non-premises AVSC 120 resources have been allocated, it replies to the non-premises AVSM 160 with an AcquireChannelReply. The non-premises AVSM 160 then replies to the premises AVSM 160 by issuing a RequestAVFileSourceReply that specifies the location or network or IP address and channel handle of the allocated non-premises AVSC 120.

The premises AVSM 160 issues an AcquireChannel request to allocate a channel and corresponding resources on a premises AVSC 120 for performing the file transfer operation, and in response receives an AcquireChannelReply that specifies a channel handle. The premises AVSM 160 then issues a CopyFrom request to the premises AVSC 120, where the CopyFrom request includes the source AVSC's location or address and channel handle. The premises AVSC 120 initiates the file transfer by issuing an Open request to the source AVSC 120, where the Open request specifies that the file is to be

LINKPLAY EXHIBIT 1004
Page 3126 of 4822

opened in DATA mode. Upon receiving an OpenReply from the source AVSC 120, the premises AVSC 120 issues a series of Read requests to the source AVSC 120, which result in the file transfer. For each such Read request, the source AVSC 120 performs a read operation and issues a ReadReply to the premises AVSC 120.

After the file has been transferred, the premises AVSC 120 issues a Close request to the source AVSC 120. The source AVSC 120 closes the file, and issues a CloseReply to the premises AVSC 120, which in turn issues a CopyFromReply to the premises AVSM 160. The premises AVSM 160 next issues a ReleaseChannel request to the premises AVSC 120 to release resources associated with the copy operation, after which the AVSC 120 issues a ReleaseChannelReply. The premises AVSM 160 subsequently updates the AVSS database 176 to reflect the presence of the new file. The premises AVSM 160 next sends a NotifyAVFileSource message to the non-premises or source AVSM 160 to indicate to the non-premises AVSM 160 that the transfer is complete. The non-premises AVSM 160 issues a ReleaseChannel request to the source AVSC 120, which releases resources reserved for the copy operation and generates a ReleaseChannelReply directed to the non-premises AVSM 160. The non-premises AVSM 160 then issues a NofityAVFileSourceReply to the premises AVSM 160. Finally, the premises AVSM 160 sends a RequestAVFileReply to the AVSS client that required the file transfer operation.

4.12.5 File Replication

Once a new A/V file resides upon an AVSC 120, the AVSM 160 may issue a series of requests to perform A/V file replication operations. In one embodiment, A/V file replication is required in the event that more than one owner is specified in the file's access privilege list. File replication would thus be performed, for example, after or as part of processing an "AddOwner" request received from a client application program. In another embodiment, A/V file replication is performed even if an A/V file has only one owner, such that a given A/V file resides upon at least two premises AVSCs 120. This approach would enhance system reliability or fault tolerance. In such an embodiment, file replication would occur, for example, following a file transfer operation as described above with reference to ***Figure*** 23.

LINKPLAY EXHIBIT 1004
Page 3127 of 4822

Figure 24 is an exemplary request sequence diagram corresponding to file replication operations. The AVSM 160 could perform file replication operations, for example, following issuance of a RequestAVFile reply to an AVSS client indicating completion of file transfer operations as previously described. To initiate file replication operations, the AVSM 160 issues a first AcquireChannel request to the AVSC 120 upon which the file resides, that is, the source AVSC 120. After receiving an AcquireChannelReply, the AVSM 160 issues a second AcquireChannel request to a destination AVSC 120 to which the file will be copied. Once the AVSM 160 receives an AcquireChannelReply from the destination AVSC 120, it issues a CopyFrom request to this AVSC 120. The destination AVSC 120 then sends the source AVSC 120 an Open request. Upon receiving an OpenReply, the destination AVSC 120 send the source AVSC 120 a series of Read requests, which result in transfer of the file from the source to the destination AVSC 120. After the source AVSC 120 has responded to a given Read request, it issues a ReadReply to the destination AVSC 120. Upon responding to a last Read request and issuing a last ReadReply, the destination AVSC 120 sends a Close request to the source AVSC 120, which closes the file and issues a CloseReply to the destination AVSC 120. The destination AVSC 120 subsequently issues a CopyFromReply to the AVSM 160. The AVSM 160 then issues a ReleaseChannel request to the destination AVSC 120, and in turn receives a ReleaseChannelReply. According to the particular file replication strategy employed, the AVSM 160 may repeat the destination AVSC 120 selection, AcquireChannel, CopyFrom, and ReleaseChannel procedures described above for one or more other destination AVSCs 120. The AVSM 160 could also select either AVSC 120 upon which a copy of the file currently resides as the source AVSC 120. Once any given AVSC's role as a file source is completed, the AVSM 160 issues a ReleaseChannel request to that source AVSC 120, which in turn releases resources reserved for performing the copying operations and generates a ReleaseChannelReply directed to the AVSM 160. Those skilled in the art will understand that the file replication operations could be performed via CopyTo requests rather than CopyFrom requests.

LINKPLAY EXHIBIT 1004
Page 3128 of 4822

4.12.6 AVSS Administration

The AVSM 160 maintains data corresponding to AVSC 120 resources and their capabilities. The AVSM may query an AVSC 120 when requested, or as part of an AVSS database update when a new AVSC 120 has been added to the premises group. ***Figure*** 25 is an exemplary request sequence diagram corresponding to new AVSC addition and AVSC query requests. In response to an authorized client's issuance of an AddAVSC request specifying an AVSC 120 and a network or IP address, the AVSM 160 issues a GetMediaSetup request to the AVSC 120. The AVSC 120 generates a GetMediaSetupReply, and provides data describing its resource types, capabilities, and characteristics to the AVSM 160. The AVSM 160 then issues an AddAVSCReply to the requesting client. In response to a GetAVSCInfo request from the client, the AVSM 160 provides data describing the AVSC's resource types, capabilities, and characteristics to the client via a GetAVSCInfoReply.

4.13 Application Program Interface

Application programs executing on user workstations 40 or other computers act as clients relative to the AVSM 160, and may further act as clients relative to the A/V network manager 34. ***Figure*** 26 is a block diagram showing client application programs communicating with the AVSM 160 and A/V network manager 34. In one embodiment, a typical application program comprises a user interface plus a set of software objects based upon or derived from the AVSMAppComm class hierarchy 250 described above. The application program thus serves as a software wrapper that can act as an AVSM client to provide access to particular types of AVSS functionality.

In response to particular user selections or actions, an application program issues requests to the AVSM 160. Based upon replies received from the AVSM 160, the application program selectively updates information presented to the user. Since a portion of the application program comprises objects corresponding to the AVSMAppComm, communication between the AVSM 160 and the application program occurs in a manner analogous to AVSM-to-AVSM communication. Additionally, since the AVSMAppComm can be readily extended or modified to reflect the evolution of AVSM functionality over

LINKPLAY EXHIBIT 1004
Page 3129 of 4822

time, new types of functionality can be readily accommodated with minimal application program modification.

4.14 Process and Data Primitives

The video storage server can support several types of applications. One of the ways that the several supported applications can share information, as well as cutting down on the coding required to embody the applications, is to share a set of common "primitive" elements between the several applications.

Primitives of the same class are combinable to achieve a specified result. One or more primitives may be combined to form new higher level primitives, which may in turn define yet another even higher level primitive, substantially *ad infinitum*. There are two broad classes of primitives: process primitives and data primitives.

Process primitives are invoked to perform an action. In turn, they may invoke other actions, including invoking other primitives. Typically process primitives establish AVSS sessions of some type, (for instance at least one of playing a file, getting file lists), present the user with a graphical interface, establish network connections, as well as other operations such as rendering synchronized graphics. Examples of process primitives taught by the principles of the present invention are viewers, browsers, and administrative processes.

Process primitives can be organized in a number of manners. A first manner is as a stand-alone process directly launched by a user. Alternatively, they may be launched *from within* another process, for instance a dynamic linked library or DLL. In this case, their thread of execution exists with the thread of execution of the launching process, and also terminates therewith. As a further alternative, process primitives may be launched *by* another process, as a *separate* process, whose thread of execution is not tied to the launching process. In this latter case, the termination of the launching process has no effect on the life of the process primitive.

There are several broad mechanisms for executing process primitives. One process primitive execution methodology implements the direct invocation of the process primitive by the application user. Another process primitive execution methodology is utilized where a target application, including but not limited to third-party software applications,

LINKPLAY EXHIBIT 1004
Page 3130 of 4822

has previously been taught how to display or act upon the receipt of a data primitive in the form of an attachment, MIME type, file type and so forth. The receipt of the data primitive causes the target application, for instance the MIME compliant software application or the operating system, to launch a process primitive associated with the data primitive. Yet another methodology for executing process primitives, especially those process primitives directly accessed by third-party software, is simply to install the process primitive as a plug-in to that software.

Data primitives refer to a number of specifically formatted file types in the context of their usage by process primitives or other related processes. Data primitives include, but are not necessarily limited to: audio-video files; audio-only files, video-only files; bitmap files; application files; postscript files; graphics files; text files; and synchronization files. Included in synchronization files are a number of time-stamped event file types including, but again not limited to: graphics event files; shareboard event files; window event files; application startup event files; and text event files. The time-stamps in the file signal when to take certain defined actions, thereby enabling the temporal synchronization of the information of one file with respect to another.

A metafile is an abstraction representing a combination of one or more externally stored data primitives which together comprise all externally stored ("out-of-band") information comprising a particular AV or multimedia segment (such as a message). Associated with the metafile is a pointer file, which includes all referencing information for the various component files of the metafile (file names, permission keys, file system, etc.). One or more pointer files and their associated metafiles can be conceptually grouped to form a virtual aggregate file called an omnifile. The omnifile therefore must be in some fashion transferred successfully from one environment to another in order for authored information to be completely replayed in the new environment. This can be done in a number of ways as will be clear to one skilled in the art. One such embodiment is shown at ***Figure*** 27.

In common practice, the term "multimedia file" as been used to refer to an aggregate of all types of media files and external reference file pointers. Multimedia files, like metafiles and indeed like other primitives, are hierarchical: they are combinable into

LINKPLAY EXHIBIT 1004
Page 3131 of 4822

other files to perform a certain task, for instance as a multimedia file in any of a number of standard formats. Metafiles, as defined herein, are allowed to be distributed across several file systems and do not include pointer files. In contrast, multimedia files can include pointer files and all of their component media files are stored together. Omnifiles do include pointer files but again are a virtual object spanning potentially several vile systems and thus also differ from multimedia files.

An application implementing the principles of the present invention specifically takes some of the file types of interest, and creates a MIME type associated with them. It then enables the teaching of some third party browsers including, but not limited to, Netscape® or Microsoft Internet Explorer®, or some e-mail packages including Eudora® how to handle the MIME types. Accordingly, many of the third-party application types supported by the present invention are taught how to handle the multimedia files, omnifiles, metafiles, audio-video files, and pointer files of the present invention. Therefore, when a third party application receives one of these file types taught by the present invention, it knows how to invoke the proper process primitive.

Referring now to ***Figure*** 28, an exemplar data primitive 3904 is shown which requires a viewer or browser process to display or act upon it. The application has been taught, as described above, how to respond to the receipt of the data primitive, it "knows" when and how to launch process primitives 3912 and 3906. In the viewer/browser exemplar herein presented, receipt of data primitive 3904 by an application causes the application to launch, at 3910 or 3914 either browser 3912 or viewer 3906. Where browser 3912 is launched, it may invoke a plurality, not shown, of viewers 3906. Viewer 3906 invokes, in turn a at least one data primitive 3908 for access to one or more files.

Referring now to ***Figure*** 29, data primitive 3904 is shown embodied as an attachment, for instance a MIME attachment, to a message 3902. By way of illustration but not limitation, message 3902 could be an e-mail message, word processing document, text document, and so forth. Depending on the application, the attachment may be implemented as the previously discussed MIME attachment, or as a file type. When the application is instructed to act upon the attachment, it invokes the process primitive associated with the attachment type, as previously discussed.

LINKPLAY EXHIBIT 1004
Page 3132 of 4822

It will be understood that the immediately preceding discussion of the viewer/browser process primitive is presented for purposes of illustration and not limitation. The principles of the present invention comprehend an almost limitless number of process primitive configurations, of which the previous exemplar is but one. The teachings of the present invention specifically contemplate all such configurations.

4.14.1 Viewer

Video record and playback capabilities can separately be embedded in each application by stand alone means, but preferentially a standardized video record playback utility used across several applications is desirable. Thus a viewer is a good candidate for a process primitive and is explained here as one example of the use of process primitives as taught by the principles of the present invention.

The viewer process primitive sets up the necessary connections and sessions, as previously described, prepares the files for viewing, and provides the necessary viewer interface to enable the user to access and manipulate the files, as permitted. Because the viewer is a standard video call, it can be treated like any other video call. This enables it to be merged with other video calls, conferences, and the like, in similar fashion to any other session enabled by the principles of the present invention.

The implementation of this feature the present invention is disclosed in ***Figure*** 29. Having reference to that figure, the present invention contemplates the utilization of a shell document 3902 invoked for instance by an application in the MCG (not shown). An example of such an application would be the launching of a video mail session, video conference session, or substantially any of the other applications discussed herein. Embedded in shell document 3902 is a MIME attachment 3904. MIME attachment 3904 invokes viewer 3906, which enables the user to access the AVSS through the network as needed to record or playback video files 3908. Permissions to record or playback specific video files may, or to the user accessing the applications. The shared connection, at 3910, presents the novel advantage of enabling video recording or playback and combining it with other applications running concurrently in a separate session. In some applications there is provided a browser, 3912, which enables the user to select between a number of

LINKPLAY EXHIBIT 1004
Page 3133 of 4822

applications 3906. Alternatively, browser 3912 may be embedded in, or part of viewer 3906.

4.14.2 MIME Attachment

Because the viewer is common across all the applications, and because the files it invokes are accessible by the several applications, a first application can use the viewer to record or playback a file, and a second application can utilize the viewer for video mail or substantially any other application. Accordingly, the video files are sharable at the file level, the attachment or MIME level, or at the connection level. The utilization of MIME (Multipurpose Internet Mail Extensions) attachments enables the invocation of the viewer and its associated networking connections within a standardized attachment which is accepted by third parties in accordance with standard MIME protocols. Thus the several applications taught in the present invention are interoperable by means of file sharing by the copying of MIME attachments, or by connection sharing.

4.14.3 Synchronized Shareboard

One example of such a concurrent session would be enable the concurrent implementation of share board graphics and a video conference session.

The addition of multimedia synchronization capability to the previously discussed viewer and MIME implementation enables the several video applications previously discussed to include animated graphics files which are synchronized with the video. Share capabilities currently implemented by the principles of the present invention "grabs" a window and stores it as a bitmap file, enabling the user to draw on top of the bitmap file. This enables the document to be used by two or more users which includes overlay graphics, to be stored and synchronized for later synchronous playback.

The synchronization is explained as follows: when the two or more users are conducting a video conference, a first window is opened, and a bitmap superimposed thereon. Any files read into the window or the bitmap may be time stamped. Any animations imposed on the bitmap are draw list events which may also be time stamped. Accordingly the principles of the present invention contemplate the capture of the several times stamps invoked during multimedia recording. This enables the later synchronization, during playback, of the several files to present a synchronous view of the entire session.

LINKPLAY EXHIBIT 1004
Page 3134 of 4822

Where a video conference includes N users, the recorded version of the synchronized session may be regarded as the N + 1 user. At playback, the recorded version of the session is again treated like any other user.

The recording process previously discussed is substantially the sequential recording of a sequence of events. If it is desirable that the previously recorded events be capable of reversal, it is necessary to render the graphics in such a way that they are reversible. Similarly, where the system enables "go-tos" it is necessary to render the graphics in such a manner that the rendered image is capable of retrieval at specified "go-to" points.

Now we have the capability to grant any screen in the window and share it with any user or any storage device enabled.

Shared applications (not understood). Recording of shared applications is accomplished by a similar means to bitmap capture.

Where annotations that draw overlays are occurring on top of the video window, the workstations must have the capability for performing graphics overlay on top of video.

4.14.4 Browser

The browser aforementioned can also be adapted by those skilled in the art to act as a process primitive in a manner similar to that described thus far.

4.15 Applications Overview

The present invention teaches a number of novel software applications enabled by, and for use in conjunction with the video server system, or AVSS, previously discussed. As the present invention enables the implementation of its features in a scalable fashion, the programmatic elements of the present invention provide, among other benefits, the ability to scale the features and advantages presented herein over a wide range of hardware implementations. An overview of one such hardware implementation is shown at ***Figure*** 30.

The present invention provides two broad classes of storage applications. Those which utilize only the basic audio and video recording, storage, browsing, and playback capabilities of the current AVSS will be referred to hereinafter as "video" applications. This is typically a reduced-capability, lower cost implementation. A second class of applications utilize audio and video along with a synchronized data sharing capability, e.g.,

LINKPLAY EXHIBIT 1004
Page 3135 of 4822

synchronized-shareboard/T.120recording, storage, browsing, and playback features. Such capabilities are essential for recording conferences, messages, or presentations where information cannot be conveniently conveyed by audio and video alone. This class of storage applications will be referred to hereinafter as "multimedia" applications. The additional functionality provided by this class of applications may require a somewhat higher implementation cost.

The principles of the present invention contemplate the AVSS supporting several types of applications. One way in which the several applications can share information is to share certain common "primitive" elements. This has the added advantage of cutting down on the coding required to embody those applications which utilize primitives. Accordingly, one or more of the applications taught by the present invention may utilize shared primitives to facilitate the interchange of information of between applications or applications elements. Alternatively, one or more of these applications may be implemented as a "stand-alone", or "one-off" implementation which does not utilize shared primitives.

4.16 Video Applications

As previously mentioned, an AVSS can support several types of applications. By way of illustration, but not limitation, some of these applications include a video mail application which combines, in a novel manner, video clips, or files, with e-mail messages whereby one user can leave a detailed audio-video message on another user's e-mail system. Video mail is like e-mail, but with either a video attachment, or with a video file replacing normal e-mail text altogether. The richness and depth of communication afforded thereby enables significantly greater information interchange than is available by simple voice mail messaging. The Video Answering System taught by the present invention answers a user's e-mail when the user is away or otherwise occupied. This application presents the user with the option of greeting callers in a number of manners, and receiving their video messages in response. One alternative of this application contemplates its implementation using the Video Mail application previously discussed. Video Conference Recording enables a plurality of users to record in real time an

LINKPLAY EXHIBIT 1004
Page 3136 of 4822

established video conference. This application may be invoked at the user's command or may, alternatively be invoked automatically when certain conference parameters are met.

In addition to these oriented to electronic meetings applications, the architecture of the AVSS system described herein renders, with facility, several additional capabilities. One such capability is the utilization of the present invention as a general-purpose audio-video storage device. This capability enables users to store and forward audio-video files, in a variety of formats, to users throughout the LAN or WAN. The connectivity of such networks to the Internet provides the additional capability of the system as an intranet or Internet gateway for audio, video, audio-video, or multimedia file transfer. Another application is Video Publishing which enables any of a number of system users to create, edit, store and disseminate complex, informationally rich, audio-video documents to a wide variety of recipients by means of LAN, WAN, or to a world-wide audience utilizing the Internet.

4.16.1 Video Mail

A first video application enabled by the principles of the present invention is video mail. In a first embodiment of this application a "video attachment" capability is provided to existing MIME-based mail systems. Alternatively, video mail may also be implemented by other attachment strategies, without attachments, or as a fully developed stand-alone application which does not rely on any underlying commercial e-mail package.

In a first embodiment of the present invention, a video attachment utility utilizes the generic video storage and session establishment method hereinafter described. Moreover, this utility utilizes the generic MIME video attachment methodology hereinafter discussed. As opposed to the video mail methodology taught in U.S. Patent No. 5,617,539 which starts a mode control GUI (or "MCG") directly at the user's workstation, an alternative embodiment presented by the present invention sets a "MCG" flag so that the AVSS starts a MCG on the user's workstation. Mail authoring invokes the request for a session with an encoder/decoder pair, while mail reading invokes a request for a session with a decoder only.

The principles of the present invention contemplate the implementation thereof on a wide variety of hardware implementations. In the simplest case, a single AVSS serves

both the message authoring user and the message recipient. In this implementation, there are no file transfers required out of the AVSS. Files may be transferred from the AVSS for other reasons, but there is no functional reason for the files to reside on another AVSS in this case. This could be considered a LAN implementation of the principles of the present invention.

Where a given organization is sufficiently large or so geographically distributed that a single AVSS cannot handle the traffic for all the users thereof, multiple AVSSs may be implemented. This case has two sub-cases: the first sub-case is a WAN implementation, and the second is where the implementation at a given site, for instance a large campus, is so large it requires more than one AVSS. In the second sub-case, the multiple AVSSs are interconnected by trunking 16 between switches 32 and/or the local data LAN 20. Where a file is required on a different AVSS than the recording AVSS in the second sub-case, it is simply transferred over the local interconnection environment 16 and 20. In the first sub-case, require the transfer of relatively large files from one AVSS to another, this implementation may use of switched WAN services, frame relay, or one or more of their functional equivalents. Moreover, because there is may be so much bandwidth required for these transfers, it may be necessary to provide bandwidth management solutions. Examples of such network bandwidth management solutions include, but are specifically not limited to conducting file transfers at non-peak times, conducting file transfers in a preemptible manner, and other bandwidth conservation methodologies known or apparent to those having ordinary skill in the art. When file transfers are conducted in a preemptible manner, if a user initiates a higher priority action, for instance she initiates a video conference, the file transfer may be preempted; i.e. aborted, halted and restarted after the high priority application, etc.

Referring now to ***Figure*** 31, the transaction flow among application elements which facilitate a local or LAN implementation of video mail is shown, where a mail message incorporating a video message is both recorded and read from the same AVSS. Having continued reference to ***Figure*** 31, when the user, at 1404, initiates a new e-mail message, the source e-mail system 1402 queries, at 1406, whether the user intends to attach a video attachment to the e-mail. If the user intends to submit a video attachment with the e-mail

LINKPLAY EXHIBIT 1004
Page 3138 of 4822

message, a video authoring request is initiated at 1412 to video mail application 1420. Video mail application 1420 creates a unique video file name at 1422 and requests of the recording AVSS 1430, at 1424, an encode session. The audio-video file is then recorded on AVSS 1430. AVSS 1430 returns the video attachment at 1432 to video mail application 1420. Video mail application 1420 in turn sets the file name pointer at 1434 for e-mail program 1402. Once the e-mail message and video attachment have been completed, the user sends the e-mail, for instance using SMTP protocol 1410, by initiating the send procedure at 1408. Once send procedure 1408 is initiated, e-mail program 1402 initiates a notification at 1414 to video mail application 1420 that the message has been sent. Video mail application 1420 then instructs the AVSS 1430 to release the encode session at 1426.

At this point, the e-mail message itself, not shown in this Figure, has been sent to the user in the normal manner, and the recorded A/V file resides in AVSS 1430. When the e-mail program 1402 receives the e-mail message at 1492, it sends, at 1460, a receipt target and information request to video mail application 1420. Responsive to the receipt target and information request 1460, video mail application 1420 extracts a source AVSS address at 1472. Video mail application 1420 then makes a file transfer request 1480 to AVSS 1430.

Responsive to request 1480, AVSS 1430 confirms it already has the requested A/V file, and confirms, at 1448, that the video file transfer is completed. Responsive to file transfer confirmation 1448, video mail application 1420 at 1474 releases the mail message after the A/V file is received. Video mail application 1420 then releases the mail message to e-mail system 1402 at 1462.

At this point, e-mail system 1402 makes the mail available to the user at 1494. Steps 1494 and 1460, in operative combination, ensure that the user is not notified of a message with an associated video attachment before the video attachment arrives. When the user opens the video attachment at 1496, e-mail system 1402 initiates an attachment open event and sets a file pointer at 1464 to video mail application 1420. In response to set file pointer 1464, video mail application 1420 opens a viewer and prepares to decode the AV file at 1476. Video mail application 1420 in turn requests a decode session from

LINKPLAY EXHIBIT 1004
Page 3139 of 4822

AVSS 1430 at 1482. Decoding of the A/V file by AVSS 1430 renders the file available to the user.

At 1498, once the user closes the message or the video attachment, a view message/attachment and file pointer close event is initiated at 1466. This causes video mail application 1420 to release the view and release the file at 1478. Responsive to release event 1478, video mail application 1420 issues, at 1484, a decode session release command to AVSS 1430. When, at 1493, the user deletes either the message or its video attachment, e-mail system 1402 initiates a message/attachment delete event 1468 to video mail application 1420. Responsive to this event, video mail application 1420 deletes, at 1473, one holder of the file, and at 1486 releases user ownership of the file to AVSS 1430. Alternatively, of course, a video file may be set to expire after a pre-assigned lifetime.

It should be noted that receipt target and information request 1460, release mail message 1462, view message/attachment and file pointer close event 1466, and message/attachment delete event 1468 as taught by this invention are novel concepts and as such are not supported by any known widely available e-mail domain servers. Study of the principles herein disclosed will make obvious to one having ordinary skill in the art that additional dialogs and displays in the video mail application can provide the information supplied by the previously listed messages. Further, a variety of alternate implementations are of course possible and the teachings of the present invention render them clear to those skilled in the art. These alternatives include workarounds for the absence of any or all of 1460, 1462, 1466, and 1468; for example, the unavailability of 1468 could be handled by a file lifetime monitor which takes action to delete a file after a designated period of inactivity.

Referring now to ***Figure*** 32, the transaction flow among application elements which facilitate wide-area video mail is shown, where a mail message incorporating a video message is transmitted from a recording, or source, AVSS 1430 to a target AVSS 1450.

When the user, at 1404, initiates a new email message, the source email system, 1402, queries at 1406 whether the user intends to attach a video attachment to the email. If the user intends to submit a video attachment with the email message, a video authoring request is initiated at 1412 to the video mail application program 1420. Video mail

LINKPLAY EXHIBIT 1004
Page 3140 of 4822

application program 1420 creates a unique video filename at 1422 and requests of the recording AVSS 1430, at 1424, an encode session. The video attachment is then recorded on the recording AVSS, 1430. AVSS 1430 then returns video attachment 1432 to video mail application 1420. Video mail application program 1420 in turn sets the video filename pointer at 1434 for source email program 1402. Once the email program and video attachment have been completed, the user sends the email, for instance using SMTP protocol 1410, by initiating the send procedure at 1408. Once send procedure 1408 is initiated, source email package 1402 initiates a notification to the video mail application 1420, at 1414, that the email message has been sent. Video mail application 1420 then instructs the recording AVSS 1430 to release the encode session, at 1426.

At this point the email message itself, not shown in this figure, has been sent to a remote AVSS utilizing normal SMTP methodology, while the encoded A/V file resides in recording AVSS 1430. When the email message is received, at 1492, at target email domain 1490 a receipt target and information request 1460 is sent to target video mail application 1470. Responsive to the receipt target and information request 1460 target video mail application 1470 extracts the source AVSS address at 1472. Target video mail application 1470 then sends target AVSS 1450 a file transfer request 1480. This in turn initiates the request 1481, by target AVSS 1450, to recording AVSS 1430 for the A/V file.

The recording AVSS, 1430, transfers the A/V file, at 1436, to the target AVSS 1450. Target AVSS 1450 confirms to the target video mail application 1470, at 1448, that the video file transfer is completed. Responsive to file transfer confirmation 1448, target video mail application, 1470, at 1474 releases the mail message after the A/V file is received. Target video mail application 1470 then releases the mail message to the target email domain 1490 at 1462.

At this point target email domain 1490 makes the mail available to the user at 1494. Steps 1494 and 1460, in operative combination, ensure that the user is not notified of a message with an associated video attachment before the video attachment arrives. When the user opens the video attachment at 1496, target email domain 1490 initiates an attachment open event and sets a file pointer, at 1464, to target video mail application 1470. In response to the set file pointer 1464, target video mail application 1470 opens a

LINKPLAY EXHIBIT 1004
Page 3141 of 4822

view and prepares to decode the file for the user at 1476. Target video mail application 1470 in turn requests a decode session from the target AVSS 1450 at 1482. Decoding of the A/V file by the target AVSS 1450 renders the file available to the user.

At 1498, once the user closes the message or the video attachment, a view message/attachment and file pointer close event is initiated at 1466. This causes the target video mail application 1470 to release the view and release the file at 1478. Responsive to release event 1478, target video mail application 1470 issues, at 1484, a decode session release command to target AVSS 1450. When, at 1493, the user deletes either the message or its video attachment, target email domain 1490 initiates a message/attachment delete event 1468 to target video mail application 1470. Target video mail application 1470 deletes, at 1473, one holder of the file, and at 1486 releases user ownership of the file to target AVSS 1450. Alternatively of course, video files can expire after a pre-assigned lifetime.

It should be noted that receipt target and information request 1460, release mail message 1462, view message/attachment and file pointer close event 1466, and message/attachment delete event 1468 taught by this invention are novel concepts and as such are not supported by any known widely available e-mail domain servers. Study of the principles herein disclosed will make obvious to one having ordinary skill in the art that additional dialogs and displays in the video mail application can provide the information supplied by the previously listed messages. Further, a variety of alternate implementations are of course possible and the teachings of the present invention renders them clear to those skilled in the art. These alternatives include workarounds for the absence of any or all of 1460, 1462, 1466, and 1468; for example, the unavailability of 1468 could be handled by a file lifetime monitor which takes action to delete a file after a designated period of inactivity.

The exemplar discussed above and illustrated in ***Figure*** 32 demonstrates a wide-area implementation of the present invention utilizing two systems, 1400 and 1440. Shown in the figure is a first methodology whereby two or more instances of a conventional electronic mail system, two or more AVSSs, and two or more instances of a simple video mail software application "middleware" can be used to create a multi-AVSS, WAN-

LINKPLAY EXHIBIT 1004
Page 3142 of 4822

capable video mail system. This relatively simple network implementation is presented herein for clarity. Study of the principles herein disclosed will make obvious to one having ordinary skill in the art that a number of target systems 1440 may be similarly implemented. All such implementations are specifically contemplated by the principles of the present invention.

The novel concepts of file ownership and video message life cycle are shown in ***Figure***s 33 and 34. In several of the applications taught herein, including video mail, file ownership evolves through the different phases of a message's lifetime. At message authoring, for example, the file is owned by the message author, while at message review the file is owned by the message recipient. This notion of changing file ownership is presented in ***Figure*** 34, which more clearly points out both ownership and readability of the video files associated with a video mail attachment over the message lifetime. Not shown in the figure are the effects of messaging forwarding, but this is simply a second authorship/recipientship transaction exactly like that shown in the figure.

Having reference now to ***Figure*** 33, the life cycle of an exemplar video message file is discussed as follows:

When a video file is created, at 1510, associated with a message by means of a reference pointer 1512, then transmitted, a mailing list 1514 is obtained from the conventional email system, 1513, and presented to the recording AVSS at 1516. Recording AVSS 1516 determines the recipients' addresses from mailing list 1514, which are then associated with the name of their serving AVSS by means of directory services. The directory services provide a mechanism whereby names, e-mail address or login identities of users may be looked up and associated with their video address in a LAN or WAN environment. The AVNM server in the present embodiment could provide such directory services, or these services may be provided by well known directory protocols such as LDAP.

At 1518 the programmatic elements of the present invention make a determination as to whether or not the mailing list 1514 associated with the video mail message is empty. If, at 1520, it is determined that the mailing list is not empty, a further determination is made for each recipient in the list if the current recipient of the message is served by the

LINKPLAY EXHIBIT 1004
Page 3143 of 4822

recording AVSS at 1522. If, at 1524 the determination is made that the current recipient is served by the recording AVSS the file is retained on the AVSS and the next recipient is chosen from the mailing list at 1526. Responsive to the selection of the next recipient at 1530, step 1518 is again invoked until the mailing list is empty. Responsive to a determination of 1528 that the current recipient is not on the recording AVSS the file is transferred to a target AVSS 1580. The completion of the copy at 1529 returns to step 1526 to select the next recipient from the mailing list. Responsive to the selection of the next recipient, at 1530, step 1518 is again invoked until the mailing list is empty.

It is a principal feature of the present invention that the relatively large video files associated with multimedia and video communications be distributed where required and maintained thereat, but only for so long as required. In this manner the principles of the present invention present a heretofore unattained level of systems economy with respect to mass storage, bandwidth, and other system delimiters. Accordingly, the principles of the present invention specifically contemplate the automated deletion of video files once certain deletion criteria have transpired. Examples of these criteria include, but are specifically not limited to: the reading of a giving file by all of its intended recipients, the passage of a certain period of time, a certain number of invocations of the message, and other message accounting parameters well known to those having ordinary skill in the art. Having this deletion criteria in mind, a determination is made at 1582 if one or more deletion criteria has been met. In the event that the deletion criteria has not been met the system loops back, at 1586, and returns to wait for the deletion criteria to be met at 1582. Responsive to a determination, at 1584, that the deletion criteria has been met at 1588 the file is deleted from the target AVSS at 1588.

Returning now to the loop through the mailing list at step 1518, if, at 1540 a determination is made that the mailing list is empty a query is made a 1542 as to whether any file recipients are served by the recording AVSS. Responsive to a determination at 1540 that no file recipients are served by the recording AVSS, the file is deleted from the recording AVSS at 1550 and this function of the present invention is completed at 1552. Responsive to a determination by step 1542 at 1544 that file recipients are served by the recording AVSS, a determination is made at 1545 at to whether file deletion criteria has

LINKPLAY EXHIBIT 1004
Page 3144 of 4822

been met. If a determination is made at 1546 that the file deletion criteria has not been met the system loops back and returns to wait for the deletion criteria to be met at 1545. Responsive to a determination by step 1545 at 1560 that the file deletion criteria has been met the file is deleted from the recording AVSS at step 1550 which action completes this function of the present invention at 1552.

From the foregoing, several novel aspects of the present invention are made manifest.

Within each AVSS, copies of the file can be distributed to one or more additional disks with the same AVSC or multiple AVSCs so as to diminish blocking to file access. This improves both system responsiveness and system reliability. In smaller implementations of the present invention, this file distribution utilizes full replication of the video files. In larger implementations, a "hashing" mapping is utilized, and is explained as follows: where a file may be in high usage, rather than leaving it on only one AVSS, for instance the AVSS on which the file was authored, thereby leaving it only on the one disk which may result in unwanted delays in access time, the principles of the present invention contemplate the distribution of high usage files to one or more storage devices within an AVSC, or indeed to a number of different AVSCs within the system. This may be accomplished by means of any of a number of hashing schemes known to those having ordinary skill in the art. This of course improves the availability of the file, and thus the reliability of the system which relies on the file.

The life cycle advantages presented by an exemplar embodiment of the present invention include, but are not necessarily limited to, the following:

(A) Video files are maintained only on those AVSSs serving a specified recipient for the video file, unless stored at another location for purposes of reliability and data redundancy.

(B) Video files are deleted automatically once all messages which include the video file have been deleted; and

(C) Where a recipient is located on an AVSS other than the recording AVSS, the video file is automatically transferred to the target AVSS. The transfer function may be implemented as "copy" or "reliable move".

LINKPLAY EXHIBIT 1004
Page 3145 of 4822

These advantages ensure that relatively large video files are transferred only where needed, and retained only on those AVSSs which require them, and then only for as long as the files are required. Unneeded files are automatically deleted from any AVSS when no longer required by any recipient served by that AVSS.

Referring now to ***Figure*** 34, during the recording and review processes, 1602 and 1604, a given file is both readable and owned by its author. Once the author sends the file, at 1606, it is still owned by the author but is unreadable. This condition continues through SMPT and video file transfer 1608 and the receive phase, 1610. Once the recipient is notified, at 1612, of the arrival of the email message and video file, both file readability and file ownership pass to the recipient. This condition continues through the message read and delete phases, 1614 and 1616 respectively. Of course, once the file is deleted it is neither owned nor readable.

The author could select to be a recipient or simply remain an owner. The file would therefore remain readable by the author during the send 1606, transfer 1608, received 1610 and notified 1612 steps. The author would participate in the read 1614 and delete 1616 phases.

Certain AVSS capabilities are required depending on the conference client implemented at individual desktops or rooms. When the previously discussed video mail system is implemented inside an enterprise incorporating desktops or rooms outfitted with workstation conference clients as described in U.S. Patent No. 5,617,539, the following AVSS features are assumed:

1. Generic control protocol interface/API.

2. A generic multi-platform audio-video recording/playback utility (similar to vfstool) referred to herein as the "mode control GUI" or "MCG".

3. Multi-session recording/playback capability.

4. Analog A/V I/O.

5. An AVSC decoder is co-allocated when an encoder is allocated. This ensures that the decoder is always available during the record session for review.

6. Variable capacity scaling to match the needs of supported applications.

LINKPLAY EXHIBIT 1004
Page 3146 of 4822

7. File transfer capability (pull model) among AVSSs driven by specific file transfer requests made by applications.

8. Effective "loopback" at the AVSC of incoming video to outgoing video during record mode.

9. A video editing capability. This capability may be internally implemented as part of an applications program, or in the alternative, may be capable of invoking third party video and/or audio editors.

Where desktops or rooms are not outfitted with the workstation conference clients taught in U.S. Patent No. 5,617,539, but rather are fitted with standard MPEG or other encoder/decoders, the following additional capabilities are required:

10. The capability to accept and playback appropriate digital files from third parties as controlled by applications.

11. The capability to transfer appropriate digital files to third parties as controlled by applications.

Sending video email naturally implies the creation of a video file. The principles of the present invention contemplate numerous video file creation scenarios. Video files may be pre-recorded, edited, and stored using the AVSS as a video file storage repository. They may be created at the same time the user is sending the video email. Video files may be imported from other video sources as described elsewhere herein, and attached to video mail messages. Finally, a video mail attachment may completely obviate the need for text with the message. In this latter embodiment of the present invention, the video attachment becomes the message, to the exclusion of text. Accordingly, it will be appreciated that substantially any methodology used to create a video file may be implemented to create video attachments for video mail, and are all comprehended by the teachings of the present invention.

When a user wishes to send a new audio-video attachment to a video mail message, this is accomplished by creating a new audio-video file by selecting FILE: NEW from the viewer main menu and then recording the video. Video recording may comprehend, in one or more embodiments, several levels of editing capability. The most rudimentary of these enables a user to delete a file and then re-record it. Only slightly

LINKPLAY EXHIBIT 1004
Page 3147 of 4822

more sophisticated is an embodiment which enables a user to first review his recording, and then delete it if needed

A further improvement is an editing scheme which enables the user to insert/delete/import material into a file. Finally, the video file is saved as an audio-video file on the local file system. The present invention contemplates each of these editing strategies in each of the applications taught herein.

While the present invention contemplates the incorporation therein of the above-described editing features, it also contemplates the utilization of a third-party video editor for any and all video editing. The third-party editor may be invoked from the present invention by means of primitive, plug-in, invocation button, scripts, or other invocation methodologies well known to those having ordinary skill in the art.

Where desired, the user then uses her normal email package to create an e-mail text message. She then attaches the AV file pointers to the email and then transmits or sends the email. An alternative embodiment contemplates a more closely integrated mail package which includes an Attach Video menu item in the mail package compose dialogue. To slightly simplify the previously discussed process, the Attach Video menu item would launch the viewer with the -n switch to indicate the viewer should automatically create a new video file and open an encode session.

When a user receives an email with an audio-video attachment, he opens the attachment using his email reader. This launches the viewer to open the audio-video attachment. If the video is password protected, the user is prompted the supply the password. If the password is incorrect, the file is not opened. Alternatively, the viewer will in turn, if necessary launch the conference mode of operation. In the case where the file is stored on the AVSS to which the viewer is connected, when conference successfully connects to the AVSS, the VCR controls for playback are enabled on the viewer. Utilizing the play, seek, and rewind buttons of those controls the user plays the video and sees and hears the output at the workstation.

In the case where the file is not stored locally, the AVSS will initiate a file transfer to receive the MPEG video from the remote AVSS. The user is notified that the transfer is taking place. When the file transfer is complete, VCR type controls for playback will

LINKPLAY EXHIBIT 1004
Page 3148 of 4822

be enabled on the viewer. Alternatively, the user could choose to connect to the remote AVSS in a manner similar to connecting to the local AVSS.

The preceding discussion has centered on a first embodiment of the present invention. Alternative embodiments contemplated by the teachings herein disclosed include: the attachment of A/V files by means of pointing to file locations as opposed to direct use of MIME systems; invocation of one or more of the previously discussed processes at the opening or closing of an attachment, or the opening or closing of a message; A/V network and server socket management schemes; video viewer and location and close management methodologies; and screen cleanup and connection management methodologies.

4.16.2 Video Answering System

The video answering application utilizes the AVSS to record an audio and video message from an incoming caller whose call attempt is either not answered or is refused by the recipient. The present invention also contemplates the incoming caller simply wishing to leave a message without connecting to the recipient. One implementation of this feature of the present invention supports the case were no A/V codec trunks are available at that the time when a call is placed to a user. The video answering system includes several major components: an answering module; a browser module; a playback module; and a module wherein file transfers are potentially handled among AVSSs.

The principles of the present invention contemplate several methodologies whereby the video answering application is invoked. One way of invoking video answering is to provide an option to a caller at any time during the "ring" cycle to leave a message for recipient. Additionally, a caller may be presented the option for leaving the message in any of the following cases: after a specified duration of a ring cycle, i.e., a ringing time-out; when a call is refused; and when the caller receives a busy signal due to an excessive number of waiting calls.

When this application is invoked, the caller may be given the option of leaving a message in a number of ways. Such notification methodologies include, but are specifically not limited to: simple text notification; simple audio tone; an audio-only greeting; a video-only greeting; an audio-video greeting, and a multimedia greeting. The

LINKPLAY EXHIBIT 1004
Page 3149 of 4822

decision to implement any of these notification methodologies depends on several factors including available storage capacity, bandwidth, desired system response parameters, desired system "look and feel," as well as other system constraints well known to those having ordinary skill in the art.

Several implementations contemplated for WAN installations include alternatives for placement of the user greetings, which alternatives include user greetings located on the user's AVSS, or alternatively be distributed to each AVSS on the WAN to be stored locally thereat. These alternatives depend on many systems installation factors including the number of users, system usage, bandwidth, and so forth. With respect to message authoring, the message may be recorded locally and sent to a remote site as in video mail, previously discussed. Alternatively the system may be configured such that it establishes a direct link to the receiving AVSS and the message is recorded on the received side. This later implementation results in more assuredly prompt delivery of the message, but requires more immediate bandwidth.

The caller may explicitly choose to not to leave a message by means of either a specific response to a dialog box, or simply by activating the hang-up button on the call viewer.

One implementation of this feature the present invention provides a caller with the opportunity to leave a message automatically either as part of an existing call status, e.g.: a user busy pop-up, with the automatic playing of a pre-recorded video greeting, or other automatic message system invocation methodologies. Once the user accepts the option to leave the message, a record session request is passed to the AVSS. When the request is granted, an MCG is provided, and an AVNM connection is established. In the event that the recipient attempts to answer the call when the caller is recording the message, a non-destructive escape is provided which enables the caller to complete the message and then connect the call to the called party. A similar non-destructive escape is utilized when the called party attempts to call back when the message recording is a process. The principles of the present invention further contemplate a destructive escape if the caller wishes to abandon the message during the recording session. Finally, in the event that the AVSS has insufficient resources to grant the record request, a fall back mode presents the caller

LINKPLAY EXHIBIT 1004
Page 3150 of 4822

with the option to leave a non-video message. Such a non-video message is provided by the "Leave Word" function of the incorporated reference.

In one embodiment of the present invention, recipients are notified when there are messages present in their respective queues. Since there could be more than one message in queue at any given time, the video answering system provides a browser to enable the recipient to review the messages waiting in her queue. The browser can display information about the video messages, including, but not limited to: caller's name, time and date of call, video file name, playback duration, description, text note from caller, and creation time of video. Finally, a playback mode is provided to enable the recipient to play the messages selected for viewing.

File ownership in the video answering application is as follows: When a message is being authored, the file is owned by the message's author. When the message is being reviewed, file ownership passes to the recipient, as identified by the AVNM.

Referring now to ***Figure*** 35, an overview of one embodiment of the video answering system of the present invention is discussed. When a caller, for instance at workstation 1802, calls a recipient, for instance the user at workstation 1804, and the video answering system is invoked as previously discussed, a request is sent from workstation 1802 to AVNM 1702. AVNM 1702 then forwards a request at 1710 to video answer module 1704, which in turn submits a session request to AVSS 1708 at 1714. Responsive to session request 1714, AVSS 1708 provides a reply, 1716, to the video answer module.

Responsive to reply 1716, the video answer module 1704 establishes, at 1712, a connection control with AVNM 1702 and notifies video playback module 1706 of the establishment of a video answering session at 1718. Responsive to this notification, video playback module 1706 submits its own session request to AVSS 1708 at 1720. Responsive to this session request, AVSS 1708, at 1722, establishes a second connection control with AVNM 1702 at 1722.

A data flow diagram detailing the operation of one embodiment of the present invention is given at ***Figure*** 36. Having reference to that figure, a caller initiates a video call request at 3502. When the video call request is not accepted at 3504, the recipient's

LINKPLAY EXHIBIT 1004
Page 3151 of 4822

greeting is invoked at 3506. As previously discussed, the video call request may not be accepted for a number of reasons, including but specifically not limited to: the caller exercising her option at any time during the "ring" cycle to leave a message for the recipient; after a specified duration of a ring cycle, i.e., after a ringing time-out; when a call is refused; and when the caller receives a busy signal due to an excessive number of waiting calls.

If the video call is not accepted, the recipient's greeting is played at 3506 and the caller is presented, at 3508, with the option of leaving a message for the recipient. This option may again be presented in a number of manners, by way of example but not limitation including: a simple text notification; a simple audio tone; an audio-only greeting; a video-only greeting; an audio-video greeting; and a multimedia greeting. If the user, at 3510, exercises her option not to leave a video message, the video answering session terminates at 3512. In the alternative, if the user exercises her option, at 3514, to leave a video message, she may create the message at 3516 in any of the manners previously discussed for the creation of video messages.

Responsive to the creation, by the caller, of the video message at step 3516, at 3518 the message is sent to the appropriate AVSS. At some later point in time, the message is delivered to the recipient at 3520. Subsequent to the recipient invoking a browser, reader, or other file inspection methodology for the purpose of reading the message, not shown in this view, the user is presented with the option, at 3522, of deleting the message.

In the event the recipient elects, at 3524, to delete the message, it is deleted from the AVSS at 3526. In the alternative, where the recipient elects, at 3528, not to delete the message, it retained on the AVSS at 3530.

The principles of the present invention contemplate two broad strategies for implementing the video answering system taught herein. A first broad strategy implements a "layered" methodology, integrating a number of applications, some of which may be pre-existing, into the video answering system of the present invention. One such layered implementation contemplates the use of the previously discussed video mail application as a message handling vehicle. This embodiment inherently defines many of

LINKPLAY EXHIBIT 1004
Page 3152 of 4822

the major file ownership issues. Further, the previously discussed authoring, browsing, playback, and notification functions are all provided by the video mail application itself. Another version of a layered implementation contemplates the utilization of a dedicated instance of a conventional email system program to provide dedicated video answering system message notification and browsing.

A second broad strategy contemplates the creation of a unique "stand-alone" application to perform the previously defined functions. This strategy, which does not utilize video mail or other program as a message handler, requires the implementation of user interfaces to present the caller with the option of leaving a message. Moreover, such an implementation requires non-mail software and new user interfaces to accomplish the tasks of message notification, message browsing, and message playback.

4.16.2.1 Layered Implementation of Video Answering System

This implementation contemplates the use of the previously discussed video mail application, or other discrete application as a message handler or other system element for the video answering system. According to this strategy, after a video connection has been established responsive to a video call not being accepted, video mail can be authored on either the transmit side, i.e., the caller's AVSS, or on the receive side, i.e., the recipient's AVSS. Which implementation constitutes the embodiment is an economical decision, based on a number of system-specific factors.

Video mail created responsive to the video answering system can be transmitted to the recipient off-line, i.e. in a non-real-time mode. This implementation would likely have minimal impact on system bandwidth in most embodiments. One option for this implementation is to author on the send side, and utilize a priority queue to transmit the message during a period of low bandwidth utilization to the recipient's AVSS. This moves the message out of real time and transmits it at a lower bit rate than authoring on the receive side, which would necessitate full-bandwidth connectivity. The alternative to this latter embodiment is of course to author video mail messages responsive to video answering system invocation on the receive side or transmit the message at the same baud rate as in video conferencing. Either embodiment is a systems administrator issue:

LINKPLAY EXHIBIT 1004
Page 3153 of 4822

conservation of bandwidth against a degree of time delay in the message responses. Where messages are needed almost immediately, the expense of bandwidth may be necessitated.

Where the video message system is invoked responsive to the fact that no communications trunks are available between sites, it may still be desirable to leave a message in any event, for later transmission when communications become available. This functionality may be enabled by a script provided by the system administrator or other automated invocation methodology known to those having ordinary skill in the art.

Where the enterprise is not too large, all greetings files from all users could be stored at all sites. Greetings files from all users at all sites enables a rich environment for video answering, where, even when there are no communications trunks available, a caller is greeted with an audio-visual greeting from any recipient. Again, this is an issue of capability versus storage capacity. Accordingly, the system administrator can determine where greetings are stored in accordance with the principles of the present invention.

Where an enterprise generally experiences low communications volumes, messages can be recorded on the receive side to good effect. This embodiment minimizes time delays in getting mail messages to the recipients. Even in this embodiment, where trunks become unavailable, or encounter periods of high use, the message can be recorded on the callers side under control of a script, and then sent as a mail message when bandwidth is available.

Other modifications to this embodiment include giving the caller the option, on the GUI, of prioritizing her message. When sending a file, e.g. an MPEG file, use of video mail enables a caller to transmit the file as a response to an answering machine response.

There are video mail systems that are entirely separate from video conferencing systems. For example, when a caller is recording a video mail message, she may be recording it in MPEG-1 video format. When a caller is talking to a recipient in real time, she may be sending video in, for example, H.320 format. The principles of the present invention contemplate the implementation of video answering by invoking video mail in such a way that it doesn't use H.320. Alternatively, where H.320 is recorded, the principles of the present invention specifically contemplate the use of this same "live"

LINKPLAY EXHIBIT 1004
Page 3154 of 4822

format for both recorded and real time video applications. This implementation further contemplates the utilization of bandwidth management techniques, recording schemes, and video compression algorithms to minimize the bandwidth impact of the relative large video messages engendered by the use of H.320 format video, as well as some other video formats.

The use of video mail as the message handler for the video answering application presents an additional complication in that it can be recording on the caller's side or on the recipient's side. The latter option ties the application more closely to the act of calling the recipient, thereby ensuring prompt delivery. In one embodiment of the present invention, a call-back feature is implemented, which automatically initiates a call back to the sender to talk to them in real time, or a reply-back feature, to which automatically invokes video mail to enable the recipient of a video message to reply with a video message of her own.

Where video mail is utilized, an attachment, for instance a MIME attachment constructed as herein after discussed, may be included in a caller's message to the recipient, the activation of which enables the recipient to return the call of the original caller. Such a MIME attachment could initiate a dialog box or button which, when activated, launches a reply call and/or message response. This provides a replacement for the "Leave Word" function taught in U.S. Patent No. 5,617,539. Alternatively, the principles of the present invention contemplate having the "Leave Word" function implemented, using it to notify the recipient of the receipt of a message, and where it can be found, e.g., the recipient's mail queue. Alternatively, the "Leave Word' function could enable querying the message from the mail queue, or it could have an entirely separate account only for video mail.

4.16.2.2 Stand-alone implementation of Video Answering System

In contrast to the previously discussed layered implementation of this application, an alternative embodiment of the present invention is characterized by being a stand-alone implementation of the video answering system of the present invention. This embodiment does not utilize any other programs, such as video mail, to provide the requisite message handling or other features of the video answering system. Many of the previously

LINKPLAY EXHIBIT 1004
Page 3155 of 4822

discussed engineering choices pertain to this embodiment, including but not limited to greetings used; location of stored greetings; how the greetings are transmitted; and whether messages are recorded remotely or locally. Because this embodiment does not utilize the functionality of other programs, these system engineering choices are not dependent on the message handling system of such an application to implement the answering machine function.

Additionally, the GUI can be better tailored to its function as an answering machine when a stand-alone version is implemented. The stand-alone strategy obviates the need for the recipient to go to the mail system to check her mail file, or any other file specified for answering machine messages. This points out an inconvenience in those versions of video answering which implement video mail as a message handler: these implementations act more like mail systems and less like an answering machines. The stand-alone version has an increased potential for being more responsive to the user with fewer intermediate steps for the user to perform in order to access, review, and process her video answering system messages. A video mail-based system is set up as a stand-alone messaging environment, while an answering machine could be more easily tuned to the fact that the recipient had been called and been left a message. This makes the answering machine function much easier to use, and, in most implementations, has at least one less layer than mail system-based implementations. Moreover, the use of video mail-based systems may present limitations on greetings, file, and recording locations, etc.

Another consideration is that, in some organizations, mail and answering machine functions may have very different urgencies. Accordingly, a call message may have a corporate priority above video mail messages. Where this is the case, answering system messages should be easier to get to, easier to review, and easier to respond to than mail messages in these organizations. In contrast, mail messages are more like memos: they are generally more thoughtfully produced documents. Moreover, mail messages not only have mail browsers associated therewith, but they typically also have one or more text fields. Accordingly, use of video mail as a message handler would likely present the recipient with one or more blank text fields when an answering machine message is displayed.

LINKPLAY EXHIBIT 1004
Page 3156 of 4822

As the use of a stand-alone answering system removes the extra layers inherent in the previously discussed video mail-based application, the recipient can see a listing of those persons who had called, like in the Leave Word function of the incorporated reference. This presents the user with the option of simply replying to the message by returning the call or by clicking on the entry to see what had been left in the answering machine messages: video; audio; or a combination thereof. In any case, response can be by message or by call. Accordingly, this embodiment contemplates the inclusion, in the message transmitted responsive to the video answering machine, of a broad array of video, audio, text, and multimedia attachments therein.

4.16.3 Video Conference Recording

The Video Conference Recording application allows users to record the audio-video portion of audio-video conferences.

In conference recording for a call or conference of *N* participants, there are potentially *N+1* possible viewpoints to record from: the conference as viewed by each individual and the conference as viewed as a sort of composite "global view", which comprehends the combined views, audio, video and multimedia, of all the conference participants. The most comprehensive form of a "global view" would be provided by recording each participant's video and audio separately in multiple concurrent record sessions, for later combination and playback.

Although this extremely comprehensive embodiment would enable a reviewer to freely look back at each participant in full detail at any time, this approach is very resource and disk-space intensive, as it requires a separate record for each participant. While the principles of the present invention specifically contemplate such an embodiment, for all but the most massively implemented hardware suites it will be appreciated that combining the several participants' views into a single global view file is generally preferred, as it is far less resource intensive. Accordingly, it will be appreciated that there is a strong overhead cost basis for recording only one "global view" video and audio signal.

Referring now to ***Figure*** 37, an overview of the video conference recording application of the present invention is discussed. When the user, for instance 1802,

LINKPLAY EXHIBIT 1004
Page 3157 of 4822

desires to invoke the video conference recording feature taught herein, a request is sent to the AVNM 1810. AVNM 1810 then forwards a request at 1812 to video conference recording application 1830 to initiate a conference recording session. Video conference recording application 1830 then submits a session request, 1814, to AVSS 1816. AVSS 1816 establishes connection control at 1818 with AVNM 1810. Video conference recording application 1830 also allocates a conference bridge at 1820 to AVNM 1810.

A user invokes, at 1806, a browser 1821 which in turn invokes a video playback application 1822. Invocation of video playback application 1822 initiates a session request at 1824 to AVSS 1816. Responsive to the session request, AVSS 1816 establishes connection control at 1820 with AVNM 1810.

Referring now to ***Figure*** 38, an overview of several of the components of the present invention required to implement video conference recording is discussed. Having reference to that figure, a first workstation 1802 can comprise any of a number of the previously discussed workstation implementations including camera 2302, microphone 2304, monitor/video card 2306, and speaker 2308. These components may optionally be connected by means of cabling, or other connection means known to those having ordinary skill in the art, to additional hardware element add-on box 2310 which in turn is connected with MLAN switch 2206. While the present invention contemplates the incorporation of a number of workstations to MLAN switch 2206, only a second workstation, 1804, is shown in this figure for purposes of illustrational clarity. Those having ordinary skill may art will recognize that a number of workstations may be so connected.

MLAN switch 2206 is connected with AVNM 1810, not shown in this view, at 2332. A conference bridge 2208 and AVSS 1816 are severally connected to MLAN switch 2206. Conference bridge 2208 further comprises a transceiver set 2336 in operative combination with MLAN switch 2206. Connected to transceiver set 2336 are video switch 2340 and audio mixer 2342. Further connected to video switch 2340 is a video mosaic generator 2346.

AVSS 1816 includes a further transceiver 2352, an encoder 2354, and a decoder 2356. A storage device, 2408, is coupled with encoder 2354 and decoder 2356.

LINKPLAY EXHIBIT 1004
Page 3158 of 4822

When a point to point, or two-party, video conference is established, point-to-point connectivity is established between workstations 1802 and 1804 through MLAN switch 2206, as shown in ***Figure*** 39. A video signal 2902 is transmitted from camera 2302 of workstation 1802 to monitor 2320 of workstation 1804. Moreover, an audio signal 2904 is transmitted from microphone 2304 to speaker 2322 of workstation 1804. Similarly, camera 2316 of workstation 1804 transmits its video signal 2906 to monitor 2306 of workstation 1802, while microphone 2318 transmits its audio signal 2908 to speaker 2308. Having continued reference to that figure, the simple implementation of a conference between two users does not invoke either conference bridge 2208 or AVSS 1816.

The inclusion of a third party, or the implementation of the video conference recording request, invokes a conference bridge 2208 as shown in ***Figure*** 40. While this figure may be used to illustrate the principles of a multi-party conference call, only two workstations are illustrated herein for purposes of clarity. Having continued reference to the figure, the invocation of conference bridge 2208 is shown. In this case audio signals 3002 and 3004 are transmitted by means of MLAN switch 2206 through transceiver 2336 to audio mixer 2342, and thence to their respective recipients. In similar fashion video signals 2508 and 2502 are transmitted through MLAN switch 2206 through transceiver set 2336 and video switch 2340 to video mosaic generator 2346. Mosaic generator 2346 transmits a signal including a mosaic video, 2510, showing all users to video switch 2340. Video switch 2340 then splits this video signal into signals 2506 and 2506'.

When a user initiates a video conference recording request, the signal processing is substantially similar to that shown in ***Figure*** 40, with the additions illustrated in ***Figure*** 41. Having reference to the latter figure, a third video signal 3104, again comprising video mosaic 2510, is split at video switch 2340 and transmitted through MLAN switch 2206 to AVSS 1816. Similarly the audio signals 3002 and 3004 are combined by the audio mixer 2342 producing a summed audio signal 3102. Summed audio signal 3102 is transmitted through MLAN switch 2206 to AVSS 1816. The video mosaic 2510 and summed audio signal are received at

LINKPLAY EXHIBIT 1004
Page 3159 of 4822

transceiver 2352, encoded at encoder 2354, and stored on storage device 2408. A subsequent request from a user for information stored on storage device 2408 is transmitted by means of decoder 2356 through MLAN switch 2206 and thence to the requesting users workstation.

The present invention contemplates that a user may select between any of a number of video representations made available by mosaic generator 2346. By way of illustration, but not limitation, three such alternatives are shown in ***Figures*** 42, 43, and 44. The previously discussed scenario is shown at ***Figure*** 42 wherein user 1, user 2, and the AVSS all see a mosaic of both users. Referring now to ***Figure*** 43, the view presented to each user, at their selection, is that of a close-up of the other user. Despite this selection, the AVSS continues to receive a mosaic of all users. Referring now to ***Figure*** 44, user 1 in this view sees a mosaic of himself and user 2, and user 2 sees a close-up of user 1. As before, the AVSS continues to receive a mosaic of all users being recorded.

The logic required to implement the previously discussed video conference recording application is presented in the data flow diagram of ***Figure*** 45. This logic may be implemented as software, hardware, firmware, or any combination of the foregoing. Having reference to that figure, a video conference recording request is initiated at 3402. The principles of the present invention specifically contemplate a number of methodologies for initiating this request. One such methodology has been previously discussed, wherein a given user initiates a request for video conference recording. Such a request makes usable the recorded video conference to any user having permission thereto. An alternative to this embodiment contemplates the situation wherein, for legal or record purposes, it is desired that *all* video conferences be recorded. In this alternative, the administrator having administrative responsibility for the system taught and disclosed herein may mandate, at her option, that video conference recording be implemented for all video conferences conducted, or for certain specified video conferences, for instance between a specified set of users.

Once the video conference recording request 3402 is initiated, video conference recording application 1830, not shown in this view, is initiated at 3404. When a

LINKPLAY EXHIBIT 1004
Page 3160 of 4822

conference bridge was not previously allocated, such a conference bridge is now allocated at 3406. The initiation of the conference recording application initiates an AVSS session request at 3408 which in turn establishes connection control with AVNM 1810 at 3410. This in turn enables AVNM connectivity with the users at 3412.

Conference bridge allocation 3406 enables the establishment of a network A/V connection from a MLAN switch 2206, not shown in this figure, at 3414. Video signals are transmitted to video switch 2340 at 3416, which in turn transmits the signals to mosaic generator 2346 at 3420. Mosaic generator 2346 in turn transmits the mosaic video back to video switch 2342, at 3416, from whence the mosaic video 2510 is transmitted to the user's and AVSS 1816 at 3418.

Referring again to step 3414, audio signals are transmitted to audio mixer 2342 at 3422. Audio mixer 2342 sends the summed audio signals at 3426 to AVSS 1816. As shown at step 3424, audio mixer 2342 further transmits to each user a copy of the summed audio signal, less that user's own audio signal.

At this point mosaic video and summed audio are stored on storage device 2408 at step 3430. When a user, having permissions, desires to access recorded video conference information, he may do so by means of browser 1821 invoked by MCG, not shown in this view, or other user interface at 3450. Browser 1821 then accesses video playback application 1822 at 3452. Step 3452 in turn initiates another AVSS session request at 3408, enabling multiple concurrent recording and playback sessions by authorized users.

Referring now to ***Figure*** 46, the playback of recorded Video conference calls is explained. A previously recorded conference call is stored on storage device 2408 of AVSS 1816. When playback of the conference call is invoked, it is transmitted at 4170 through decoder 2356, which separates the recorded conference call into its composite video and audio signal components, 4180 and 4182 respectively. In the exemplar presented in ***Figure*** 46, only two conference participants, at workstations 1802 and 1804, are shown for purposes of illustrational clarity. Of course the principles of this embodiment of the present invention contemplate a larger plurality of user workstations.

Video signal 4180 is transmitted through MLAN switch 2206 to transceiver 2336,

LINKPLAY EXHIBIT 1004
Page 3161 of 4822

and thence through video switch 2340 to mosaic generator 2346. Note that mosaic generator has inputs for each N conference participant plus one for video signal 4180. In this manner, previously recorded video conferences are treated as additional conference participants during playback. The output from mosaic generator 2346 is a mosaic 4190 having $N+1$ frames: one for each participant, plus one for the recorded conference being played back. The output from mosaic generator 2346 is transmitted to video switch 2340, where it is split into two video signals, 4188 and 4188' and transmitted to workstations 1802 and 1804 as shown.

A corresponding audio signal 4182 is simultaneously transmitted through MLAN switch 2206 and transceiver set 2336 to audio mixer 2342. Audio mixer 2342 sums the audio inputs, 3004 and 3002, from users at workstations 1802 and 1804 respectively with audio signal 4182 in the following manner: a first summed audio signal, 4184, is transmitted to a first user at workstation 1802, and is composed of the sum of audio signals from a second user at workstation 1804 and from the recorded video signal 4182. A second summed signal 4186 is transmitted to a second user at workstation 1804, and is composed of the sum of audio signals from the first user at workstation 1802 and from the recorded video signal 4182. In this embodiment, note that in each case the workstation's own signal is not transmitted back to that workstation for purposes of improving audio clarity, but that the summed signal contains only audio from other "participants", which includes the signal from the recorded conference. The system may also maintain a summed audio signal for all participants.

From the foregoing, the following features and advantages of a first embodiment the present invention are shown:

1. The "global view" of the sum of all user's audio signals is used by the Video Conference application so as to capture all parts of the conversation.

2. For multi-point conferences, the "global view" for both audio and video is readily obtained from the conference bridge hardware in the following manner:

A. Global view video is naturally produced by the video mosaic box.

B. Global view audio is provided by a special mix in the audio matrix mixer. This "global" mix is exactly the one used in ganging distributed

LINKPLAY EXHIBIT 1004
Page 3162 of 4822

conference bridges. In other words, the "global" mix is the audio signal recorded. A subset of the global mix is sent to each of the conference users. This subset includes the global mix less the user's own audio signal. This is necessary in order to preclude "echoing" and other adverse audio effects.

Each user is provided with a choice between global view video and a selected close-up view. This feature is enabled by the standard MCG.

3. For point-to-point calls (i.e., a two-user conference), a signal capturing global view is generally not inherently available from within the hardware layer and is synthetically created as described.

In multi-party conferences, a conference bridge is automatically allocated. Where a single global viewpoint is to be recorded, two-user calls mandate that a conference bridge be allocated by the video conference application. This conference bridge is most efficiently used if it comes from the same pool used for real-time multi-party conferences.

In the case of protracted conference recordings it may be desirable to "tag" points in time where certain discussion events occurred. One embodiment of the present invention contemplates the use of such tags. Tags are preferably given a unique identifier, for instance a searchable text label or letter, whereby a specific discussion event is uniquely identified.

A user interface is provided to invoke and name a conference recording, and to invoke and operate the tag interface, where implemented. Additional user interfaces are implemented on the MCG for message notification; message browsing; and invocation of message playback.

It should be further noted that the conference bridge is not sensitive as to the nature of the input source. Accordingly, by way of illustration but not limitation, input can be by means of security camera, VCR, or substantially any other video or audio source, and so forth. Moreover, a conference call can specifically include a previously recorded video file All these inputs can be recorded as they are discussed. From this, and as previously discussed, it follows that the system enables multiple simultaneous source sessions: e.g.: one or more recording sessions and one or more playback sessions, and

LINKPLAY EXHIBIT 1004
Page 3163 of 4822

they are all completely independent. Each is connected to the conference bridge, and the user interfaces are available to one or more users during the video meeting.

The Video Conference Recording application previously discussed utilizes and implements the following capabilities:

1. Generic control protocol interface/API.

2. A generic multi-platform audio-video recording/playback utility referred to herein as the "mode control GUI" or "MCG".

3. Multi-session recording/playback capability.

4. Analog audio-video input and output.

5. A decoder co-allocated when an encoder is allocated. This ensures that the decoder is always available during the record session for review.

6. Variable capacity scaling to match the needs of supported applications, and hardware configurations.

Moreover, the following capabilities may be required or advantageous, depending on how wide-area calls are handled, or where architecturally convenient.

7. File transfer capability (pull model) among AVSSs driven by specific file transfer requests made by applications.

8. Effective "loopback" at the AVSC of incoming video to outgoing video during record mode.

9. A video editing capabilities. This capability may be internally implemented as part of an applications program, or in the alternative, may be capable of invoking third party video and/or audio editors.

4.16.4 Video Documents

The Video Documents application allows stored video files to be included in an on-line electronic document such as Word ™ or Framemaker ™. This can be used for many enterprise-wide applications including, but not necessarily limited to: training, corporate memory, procedures, references, marketing, and so forth. There are also many potential overlaps with the Intranet and Internet, as discussed below.

One embodiment of this feature of the present invention is shown in ***Figure*** 19. Having reference to that figure, substantially any archival video are combinable with text

LINKPLAY EXHIBIT 1004
Page 3164 of 4822

or documents. Moreover, the principles of the present invention comprehend video overlay graphics, which overlay the video, as well as a compendium of a plurality of video files which further include additional text or graphics, standalone videos, and MIME attached video files. Accordingly, the video document may bring up a video window alongside the document as viewed by the user, or overlaying the document, which enables the user to play the video while viewing the document. Video can come from any previously discussed source. Videos in a video document are sharable during a conference call or by video mail or messaging.

An embodiment of the present invention contemplates attaching the viewer to a message as a MIME or other functional attachment. A given call may be expanded to include other participants and other mechanisms to merge the document into a running conference. In other words, by the establishment of another concurrent session. This is more than simple video as it enables synchronized graphics either overlaying the video or accompanying the video.

Of course, any such implementation which necessitates the interface with third-party software raises some interoperability issues. Some of these issues include:

The necessity, in some implementations, to make available an additional document type for the supported package using MIME extensions.

The workstation Conference client video window and the MCG may be used for video delivery. However, in some applications of the principles of the present invention, it may make more aesthetic sense to include the video window and some associated controls, e.g., the playback control, in the document itself. One means of implementing this feature would be as a graphics type within an anchored frame.

The necessity, in some implementations, to capture both "video type" delete events and "entire document" delete events so as to not fill the file system with unused video files.

The necessity, again, in some implementations, for an automatic file transfer or user-warning mechanism to ensure that video files are transferred when the document is transferred to a domain not served by the authoring AVSS.

LINKPLAY EXHIBIT 1004
Page 3165 of 4822

Each of these issues is, of course, highly application specific. The implementation of specific solutions thereto is within the purview of those having ordinary skill in the art, in light of the teachings herein enumerated.

The system can support video document applications either using conventional third-party document systems or document systems enhanced to offer the ability to obtain and or file copy events from the document system.

The present invention can use word processing GUIs and the MCG. Accordingly, additional user interfaces are not required.

The Video Documents application requires the following capabilities:

1. Generic control protocol interface/API.

2. A generic multi-platform audio-video recording/playback utility (similar to vfstool) referred to herein as the "mode control GUI" or "MCG".

3. Multi-session recording/playback capability.

4. Analog A/V I/O.

5. An AVSC decoder is co-allocated when an encoder is allocated. This ensures that the decoder is always available during the record session for review.

6. Variable capacity scaling to match the needs of supported applications.

7. File transfer capability (pull model) among AVSSs driven by specific file transfer requests made by applications.

8. Effective "loopback" at the AVSC of incoming video to outgoing video during record

Additionally, the following feature is required to include other video file sources:

9. The capability to accept and playback appropriate digital files from third parties as controlled by applications.

Finally, in a first embodiment of the present invention, in order to support Internet access, the following feature is required:

10. The capability to transfer appropriate digital files to third parties as controlled by applications.

LINKPLAY EXHIBIT 1004
Page 3166 of 4822

12. A video editing capabilities. This capability may be internally implemented as part of an applications program, or in the alternative, may be capable of invoking third party video and/or audio editors.

4.16.5 General Purpose Video Storage

In many cases there are needs for less sophisticated "raw" video storage capabilities reachable from workstations over the video distribution network. These "raw" needs amount to the functionality of a network-accessible VCR or writable laserdisk with its information organized according to file system conventions. Sources for the stored video content may be, for example, clips from broadcast news programs, copies of camcorder footage, captured segments from videotapes, video files transferred from the CDs, DVD, or the Internet. Although user workstations are likely involved with the acquisition, capture, or transfer of such video information (via auxiliary audio/video input jacks at the desktop workstation, file transfer actions, etc.), in general the genesis of this class of video is in contrast video captured from video calls, video conferences, message authoring, or video document authoring. Further, the storage of this class of "raw" video information would not support supplementing the video with any annotations, text, shared graphics, etc.

Current trends in computing lead users to treat such "raw" video clips as any other file in their networked computing environment, freely copying individual copies of large video files and storing them on local disk or in standard data-network file servers. Such an approach has several disadvantages:

- such video only available at workstations with decoding capabilities for the needed video protocol and file format
- data networks and standard data file systems get heavily loaded given even a modest amount of video usage
- large numbers of private copies of large video files load local disks and file servers even further by huge multiples.

It is obvious to one skilled in the art that the current invention easily addresses these shortcomings by offering:

- centralized shared encoding, decoding, and transcoding

LINKPLAY EXHIBIT 1004
Page 3167 of 4822

- moving video over appropriate video-engineered networks rather than data networks
- sharing of a small number (one to a few) of these large-size files across the entire community of users who need access to the file
- still permitting file-transfer and digital streaming where required or desired.

To do this, the video file formats used must match those supported by the AVSS.

By appropriately designing the invention's handling of this class of video information, the video information can be freely exchanged with other types of applications. For example, subject to file permissions, video information can be exchanged between "raw" video clips of the general purpose video storage application and other AVSS applications:

- a "raw" video clip can be incorporated into a videomail message, video document, video publishing applications, video web pages, etc.
- a "raw" video clip can be viewed within a video call or conference
- any video captured within a recorded video call, recorded video conference, video message, or other networked video applications could be captured as a "raw" video clip.

In some situations such exchange of video information between applications would involve an entire unaltered video file. The actual methods for the exchange for this case can be accomplished in at least one of two ways:

- creating a new application-owner of the existing file
- playing back the file as rendered video which is then re-recorded, possibly during viewing, by another networked video application

Other methods are also possible for one skilled in the art.

In other situations, only segments of an original video file are to be transferred between applications, or other edits of the original video file may be required. For these cases, the actual methods for the exchange could include at least one of the two following ways:

LINKPLAY EXHIBIT 1004
Page 3168 of 4822

- use of a video file editor on the existing file to create a new (edited) file, and assigning the appropriate application-owner to this new file
- playing back the file as rendered video and, during viewing, re-record selected segments by means of another networked video application.

Other methods are also possible for one skilled in the art.

The General Purpose Video Storage application requires, in most implementations, the following capabilities:

1. Generic control protocol interface/API.

2. A generic multi-platform audio-video recording/playback utility (similar to vfstool) referred to herein as the "mode control GUI" or "MCG".

3. Multi-session recording/playback capability.

4. Analog A/V I/O.

5. An AVSC decoder is co-allocated when an encoder is allocated. This ensures that the decoder is always available during the record session for review.

6. Variable capacity scaling to match the needs of supported applications.

7. File transfer capability (pull model) among AVSSs driven by specific file transfer requests made by applications.

8. Effective "loopback" at the AVSC of incoming video to outgoing video during record

9. The capability to accept and playback appropriate digital files from third parties as controlled by applications.

10. The capability to transfer appropriate digital files to third parties as controlled by applications.

12. A video editing capabilities. This capability may be internally implemented as part of an applications program, or in the alternative, may be capable of invoking third party video and/or audio editors.

In some limited domain cases, only features 1-8 may be required.

4.16.6 Intranet Video Storage Utility

Intranet refers to the use of Internet utilities, such as web pages, web browsers and web sites, as a means for distributing enterprise-internal information. In that web page

LINKPLAY EXHIBIT 1004
Page 3169 of 4822

authoring tools commonly accept MIME attachments, clearly the AVSS video MIME attachment service primitive can be used in a web page within an enterprise outfitted with the invention. As shown in ***Figures*** 3 and 4, the AVSS can readily be connected with the Internet to facilitate the use of its capabilities with web pages and other Internet utilities.

Use of the AVSS video MIME attachment allows not only the attachment of video clips to arbitrary intranet web pages but also any other features that could be included in the MIME attachments, such as synchronized shareboard sessions, discussed later). Further, as any AVSS video file or AVSS MIME attachment can, subject to permissions, be transferred easily from AVSS-application to AVSS-application, the intranet can serve as another useful enterprise-internal video publishing method.

It should be noted that other types of AVSS-based video applications could be written to serve as web page building blocks by those skilled in the art. Thus, the AVSS advantages described above can be obtained by those skilled in the art via means other than the AVSS MIME attachment.

The Intranet Video Storage application enables raw audio-video files as well as video documents be stored, searched, and accessed within the enterprise, in accordance with Intranet browser interfaces and other conventions.

The General Purpose Video Storage application requires, in most implementations thereof, the following capabilities:

1. Generic control protocol interface/API.

2. A generic multi-platform audio-video recording/playback utility (similar to vfstool) referred to herein as the "mode control GUI" or "MCG".

3. Multi-session recording/playback capability.

4. Analog A/V I/O.

5. An AVSC decoder is co-allocated when an encoder is allocated. This ensures that the decoder is always available during the record session for review.

6. Variable capacity scaling to match the needs of supported applications.

7. File transfer capability (pull model) among AVSSs driven by specific file transfer requests made by applications.

LINKPLAY EXHIBIT 1004
Page 3170 of 4822

8. Effective "loopback" at the AVSC of incoming video to outgoing video during record

9. The capability to accept and playback appropriate digital files from third parties as controlled by applications.

12. A video editing capabilities. This capability may be internally implemented as part of an applications program, or in the alternative, may be capable of invoking third party video and/or audio editors.

In some limited domain cases, only features 1-8 may be required.

4.16.7 Internet Gateway

Current trends in desktop computing technology, networking, and Internet usage has just started the emergence of video files and streaming over the Internet. These few early video exchanges are essentially all low resolution and usually low frame-rate, hence the resulting impact on limited Internet bandwidth, limited corporate network bandwidth, and limited enterprise file system space has been noticeable but manageable. As video rapidly increases in importance to business, limited Internet bandwidth, limited corporate network bandwidth, and limited enterprise file system space will be immediately taxed. Also to be noted is that increasing business use of video will demand increasing resolution and frame-rate, ballooning the bandwidth and file size requirement for each second of video information. Finally, current trends continue to encourage a plethora of video standards and protocols; to support all of them and their respective evolutions at each desktop workstation demands costly computing power, applications software, dedicated hardware, and administration; in many enterprises, this is completely intractable.

As shown in ***Figures*** 3 and 4, the AVSS can readily be connected with the Internet. With this connectivity, the AVSS can be used as a gateway for incoming Internet video streams (which it can read into and thus convert into a file) as well as a proxy-server for incoming video file transfers from the Internet. Once these resulting video files are in the AVSS, perhaps initially captured as "raw" video (as described earlier), they then can be viewed at user workstations or used in other AVSS applications. The AVSS can provide any transcoding operations that may be needed. This approach permits, at no appreciable additional cost, an enterprise outfitted with the invention to pull

LINKPLAY EXHIBIT 1004
Page 3171 of 4822

in large amounts of video information from the Internet, store it economically, and make it available to any user with a workstation fitted with the invention's low-cost audio-video networking hardware and software.

The AVSS can also be used as a repository for video files accessed outside the enterprise via the Internet. The simplest implementation of this nature would simply make "raw" video files available on the network. This could be extended to include annotation animations, generated for example by shareboard or a similar data-sharing application; such annotation animations could in fact be available stand-alone or temporally synchronized to audio, video, or audio-video material. Software viewers and players (with software-only video decoding, for example) could be created as application software for use with, or incorporation into, Internet browsers. The software viewers and players can be implemented and deployed in various ways in a manner readily understood by those skilled in the art, for example:

- an applet downloaded from the server over the Internet
- a full application downloaded from the server over the Internet, said application such that it can be installed on the Internet-user's personal computer
- a stand-alone application product bought and sold in the marketplace

It is noted that all of the above functionality could further be extended in scope to include the capability of handling the full AVSS MIME attachments.

The Internet Gateway application implements at least one of the following functions: the reception of incoming audio-video information from the internet; and the making of internal audio-video information digitally accessible to third parties on the Internet.

Implementation of a first embodiment of this part of the present invention requires the following AVSS features:

8. Effective "loopback" at the AVSC of incoming video to outgoing video during record

9. The capability to accept and playback appropriate digital files from third parties as controlled by applications.

LINKPLAY EXHIBIT 1004
Page 3172 of 4822

12. A video editing capabilities. This capability may be internally implemented as part of an applications program, or in the alternative, may be capable of invoking third party video and/or audio editors.

Where a remote user is outside the enterprise, the principles of the present contemplate the utilization of third-party browsers including, but specifically not limited to Netscape, Microsoft Internet Explorer, and/or FTP.

Embodiments of the Internet Gateway application would typically require the following capabilities:

1. Generic control protocol interface/API.

2. A generic multi-platform audio-video recording/playback utility (similar to vfstool) referred to herein as the "mode control GUI" or "MCG".

3. Multi-session recording/playback capability.

4. Analog A/V I/O.

5. An AVSC decoder is co-allocated when an encoder is allocated. This ensures that the decoder is always available during the record session for review.

6. Variable capacity scaling to match the needs of supported applications.

7. File transfer capability (pull model) among AVSSs driven by specific file transfer requests made by applications.

8. Effective "loopback" at the AVSC of incoming video to outgoing video during record

Additionally, at least one, and in some implementations both, of the following capabilities are required by some embodiments:

9. The capability to accept and playback appropriate digital files from third parties as controlled by applications.

10. The capability to transfer appropriate digital files to third parties as controlled by applications.

12. A video editing capabilities. This capability may be internally implemented as part of an applications program, or in the alternative, may be capable of invoking third party video and/or audio editors.

4.16.8 Video Publishing

Video publishing is accomplished by a user posting video on the AVSS for viewing within the desktop collaboration system or throughout the standard data-based Internet/Intranet.

Desktops (analog or digital) can access stored video documents, either over the LAN, the WAN, or the Internet. They can also be made available to CMW users, as well as outside workstations which have reduced graphics capability. Video publishing enables the publishing of the documents to the outside world, as well as the importing, from the outside world, of additional material, again including text, graphics, audio, video or multimedia files. This latter feature comprehends the import and export of a number of disparate video and other formats including, but specifically not limited to, MPEG1 – 7, Motion JPEG, Video for Windows, Quicktime, DVI, recorded H.320, wavelets, and wavelet-compressed fractals.

The implementation of video publishing on the AVSS comprehends a transcoding methodology embodied as hardware, software or firmware to enable the transcoding among the several formats required, thereby making the video files substantially format transparent to the receiver thereof. Any user with a workstation and permissions from the system administrator can accordingly publish documents to any user on the net or any other person outside the net who has access to the files. This enables high quality video publishing from analog equipment without the expense of implementing digital equipment.

At some point, it becomes cost-prohibitive to maintain all the video files ever created by an enterprise on rapid online storage, such as magnetic or magneto-optical hard drives. At some point long term or mass storage must be implemented for outdated or seldom-accessed video files. The present invention contemplates the inclusion therein of an automated routine which determines the relative time importance of files and either moves them to and from one or more types of long term storage device, including mass storage, magnetic tape, or other long term storage technology well known to those having ordinary skill in the art, or which deletes the files from the system. This algorithm either presents the user with the option of archiving her video files, or does so automatically. This may be done by user script from the System Administrator, or other archival storage

LINKPLAY EXHIBIT 1004
Page 3174 of 4822

technology well known to those having ordinary skill in the art.

4.17 Multimedia Applications

The present invention enables the establishment of real time data sharing sessions between two workstations using shareboard or other synchronized data sharing technologies. The implementation of shareboard enables a given application to be invokable on each workstation involved in the video conference, video mail or other application taught herein. Any user having permission to access the file may then import onto his screen, as another window, a copy of the application onto that screen. Independent scrolling of each screen is enabled. On top of this application in the window the users may then independently point, draw, type text, and perform other graphics functions by means of graphics overlay files. The user can further grab other file programs and so forth and draw them into the conference window. Further, the users may independently telepoint the items on one or more of these applications. Each of these shareboard features are further explained in the incorporated reference.

The present invention enables the shareboard features previously discussed to be extended to the use of stored applications. This enables a user to simultaneously narrated and telepoint, for instance point into the subject of interest and using language like "this one here and that one over there". Accordingly, it will be appreciated that what is required is minimally the user's voice being synchronized with the actions of the pointer. The present invention not only enables this feature in real time, but further enables that this synchronized speech and graphics capability is storable and retrievable such that when the stored file is invoked the data, speech, and graphics synchronization remains accurate. This enables the file to be recorded, stored, and sent to another user as video mail, as an input in response to an answering system greeting, as a document, or substantially in any other file format taught or made obvious by the teachings of the present inventions. It should be noted that this storage includes underline bitmaps of any application so invoked.

The shareboard reference, fully discussed in the incorporated reference, is used in a first embodiment of the present invention. This provides for a uniform user interface. As shareboard is an intricate feature of the video conference methodologies taught herein, the extension of its features to the other applications taught herein enables like

LINKPLAY EXHIBIT 1004
Page 3175 of 4822

functionality without the implementation of additional recording tools or user interfaces. In a very real sense the utilization of shareboard not only means that a user must learn only one interface, usable in real time or otherwise, but it also helps with the users ideation that he is enabled, by the principles of the present invention, to share data across the several applications forming the application suite herein. The combination of multimedia synchronization, previously discussed, combined with shareboard functionality solves a number of compatibility, data reuse, and user interface problems. Moreover, the user shareboard in operative combination with the multimedia synchronization previously discussed enables the implementation of the principles of the present invention utilizing very specific data storage formats as well as the sharing of bitmaps.

A slightly more general case in the use of shareboard is application sharing. When an application is invoked and a plurality of conference users can simultaneously utilize it, for instance each entering data in a single shared spreadsheet program. The present invention is no longer operating with only a snapshot of a bitmap image, but rather is actually working in the process of the application and the several users are simultaneously changing files together.

4.17.1 Multimedia Mail

The Multimedia Mail application adds synchronized datashare, e.g., synchronized-Shareboard/T.120 recording, storage, browsing, and playback capabilities to the Video Mail application described in the previous section.

This inclusion of this feature provides significant value in that mail messages can now include the same media options as real-time desktop video conferencing. Thus a user can telepoint to one number in a spreadsheet, and then circle another number in the spreadsheet, as she first smiles and then says "How on earth could this number be half of this number?" and the entire cross-media message gets captured and conveyed as it would have been in a live desktop video conference, including all the subtle communications included by gesture, tone of voice, body language, and so forth.

The full content of other multimedia storage applications with synchronized datashare, e.g., synchronized-Shareboard T.120 recording and playback capabilities. Thus a user can forward a multimedia answering application message, multimedia

LINKPLAY EXHIBIT 1004
Page 3176 of 4822

conference recording, and so forth with other authors or authoring venues in multimedia e-mail or other multimedia application taught herein. This application requires a new AVSS feature:

11. Synchronized datashare, e.g., synchronized-shareboard/T.120 recording, storage, browsing, and playback. This event list information is best stored in a separate file from the MPEG A/V file; the pair of this event list and the MPEG A/V file could be called a "multi-media meta-file."

4.17.2 Multimedia Answering System

The Multimedia Answering application adds synchronized datashare, e.g., synchronized-Shareboard/T.120 recording, storage, browsing, and playback capabilities to the Video Answering application described in the previous section.

Utilization of this feature provides significant value in that a user can leave a more complete message in comparatively very little time should the call recipient themselves or the necessary connectivity not be available at the time a call attempt is made. This is because all relevant graphic overlay can be included in either the recipient's greeting or the caller's message.

Again, this application, like all other multimedia applications discussed herein utilizes the previously discussed feature 11.

4.17.3 Multimedia Conference Recording

The Multimedia Conference Recording application adds synchronized datashare, e.g., synchronized-Shareboard/T.120 recording, storage, browsing, and playback capabilities feature 11 to the Video Conference Recording application described in the previous section. This ensures that all transactions spanning the full spectrum of media used in desktop teleconferences of two or more participants can be captured for later review.

4.17.4 Multimedia Documents

The Multimedia Documents application adds synchronized datashare, e.g., synchronized-Shareboard/T.120 recording, storage, browsing, and playback capabilities of feature 11 to the video documents application described in the previous section. This ensures that procedural descriptions leveraging video can also include synchronized

LINKPLAY EXHIBIT 1004
Page 3177 of 4822

telepointing and annotation. This is extraordinarily valuable in on-line training and reference material applications.

4.17.5 General Purpose Multimedia Storage

The General Purpose Multimedia Storage application adds synchronized datashare, e.g., synchronized-Shareboard/T.120recording, storage, browsing, and playback capabilities feature 11 to the General Purpose Video Storage application described in the previous section. The advantages presented by such an implementation mirror those discussed above in the section entitled "Multimedia Documents".

4.17.6 Intranet Multimedia Storage Utility

The Intranet Multimedia Storage application adds synchronized datashare, e.g., synchronized-Shareboard/T.120recording, storage, browsing, and playback capabilities of feature 11 to the Intranet Video Storage application described in the previous section. Again, the advantages presented thereby are similar to those discussed above in the section entitled "Multimedia Documents".

The principles of the present invention have been discussed herein with reference to certain embodiments thereof. Study of the principles disclosed herein will render obvious to those having ordinary skill in the art certain modifications thereto. The principles of the present invention specifically contemplate all such modifications. The present invention may be practiced without any element disclosed herein.

LINKPLAY EXHIBIT 1004
Page 3178 of 4822

CLAIMS

1. A networked multimedia system comprising:

A) a plurality of workstations, each including

i) video and audio reproduction capabilities, and

ii) video and audio capture capabilities;

B) at least one converter configured to

i) convert audio/video signals,

(a) originating at any one of the workstations,

(b) into a form suitable for storage;

C) at least one storage cell configured to

i) store,

(a) the converted audio/video signals

- for later retrieval; and

D) at least one signal path,

i) interconnecting

(a) the workstations, the converter(s) and the storage cell(s),

wherein system is configured to

i) store converted signals,

(a) onto the storage cell,

(b) such that the stored converted signals

- can later be retrieved

i. under direction of one or more of the workstations.

2. A networked multimedia system comprising:

A) a plurality of workstations, each including

i) video and audio reproduction capabilities, and

LINKPLAY EXHIBIT 1004
Page 3179 of 4822

(a) video and audio capture capabilities;

B) at least one converter configured to

i) convert audio/video signals,

(a) originating at any one of the workstations,

ii) into a form suitable for storage;

C) at least one storage cell including

i) at least one storage cell,

ii) each storage cell being configured to

(a) store,

- for later retrieval,
- signals

i. converted by at least one of the converters, and

(b) facilitate transfer

- of previously stored audio/video signals
- to another storage cell
- whether included in that or another storage cell system; and

D) at least one signal path,

i) interconnecting

(a) the workstations, the converters and the storage cells,

wherein the system is configured

ii) to co-operate with the storage cell

iii) to manage

- the storage and
- transfer,

i. of the converted audio/video signals,

ii. among storage cells.

3. A networked multimedia system comprising:

A) a plurality of workstations, each including

i) video and audio reproduction capabilities, and

ii) video and audio capture capabilities;

B) at least one storage server

i) the storage server including

(a) at least one storage cell having

- at least one storage disk,
- a storage disk controller
  - i. associated with each disk, and
- at least one converter; and

(b) a storage cell manager (AVSM), and

ii) configured to

(a) store,

- for later retrieval,
- audio/video signals
- converted by the converter(s); and

C) at least one signal path,

i) interconnecting,

(a) the workstations and the storage server,

wherein the networked multimedia system is configured to

ii) store the converted audio/video signals by

LINKPLAY EXHIBIT 1004
Page 3181 of 4822

(a) selecting any one of the group consisting of disks, disk controllers, converters, storage cells and storage cell managers (AVSM), according to the following approach:

- if a first converter
  - **i.** of a first cell
  - **ii.** has reached its capacity or bandwidth limit,
  - **iii.** then a second converter, of the first cell, is selected;
- if a first disk controller
  - **i.** of the first cell
  - **ii.** has reached its capacity or bandwidth limit
  - **iii.** then a second disk controller of the first cell is selected;
- if a first disk,
  - **i.** associated with a first of the disk controllers
  - **ii.** has reached its capacity or bandwidth limit,
  - **iii.** then a second disk associated with the first disk controller is selected;
- if the first storage cell
  - **i.** associated with a first storage cell manager (AVSM)
  - **ii.** has reached its capacity or bandwidth limit,
  - **iii.** then a second storage cell associated with the first storage cell manager (AVSM) is selected; and
- if the first storage cell manager (AVSM) has reached its capacity,
  - **i.** then a second storage server, is selected, and

(b) storing the converted signals using each of the selected ones of the group.

LINKPLAY EXHIBIT 1004
Page 3182 of 4822

4. A networked multimedia system comprising:

A) a plurality of workstations, each including

i) video and audio reproduction capabilities, and

ii) video and audio capture capabilities;

B) at least one converter arranged to

i) convert audio and/or video information,

(a) originating at a workstation,

(b) into a form suitable for storage;

C) a least one storage cell, each including

i) at least one storage disk

(a) configured to receive and store,

- for later retrieval,
- the converted audio/video signals,

D) at least one storage cell manager (AVSM); and

E) at least one signal path,

i) interconnecting

(a) the workstations, the converter and the storage cells,

wherein the system is configured to perform any one of the group of functions consisting of:

ii) transferring

(a) the stored converted signals

(b) between any two storage disks of any storage cell;

iii) transferring

(a) the stored converted signals

(b) between any two storage cells,

LINKPLAY EXHIBIT 1004
Page 3183 of 4822

- under control of a single storage cell manager (AVSM); and

iv) transferring

(a) the stored converted signals

(b) between any two storage cells

- respectively under control of different cell managers (AVSM).

5. The networked multimedia system of any one of claims 1, 3, or 4 wherein

A) the system is configured to

i) facilitate transfer

(a) of converted audio/video signals

(b) previously stored by a first storage cell

(c) to another storage cell

6. The networked multimedia system of any one of claims 1 to 4, wherein

A) the converter is at least one of

i) a transcoder, an encoder and a combination encoder/transcoder.

7. The networked multimedia system of any one of claims 1 to 4, wherein

A) retrieved signals can

i) result in audio/video reproduction

ii) at one or more of the workstations.

8. The networked multimedia system of any one of claims 1 to 4, wherein

A) the converted signals are stored

i) in at least one file, which can be accessed by

(a) more than one workstation at the same time and/or

(b) more than one application type.

9. The networked multimedia system of claims 8, wherein

LINKPLAY EXHIBIT 1004
Page 3184 of 4822

A) multiple copies of each such file can exist.

10. The networked multimedia system of claim 9 further comprising:

A) a storage server defined by at least the following

i) at least one of the storage cells,

ii) at least one of the converters and

iii) at least one controller configured to do at least one of the group of

(a) determine whether signal conversion is to occur;

(b) if the system comprises more than one converter,

- determine which of a plurality of converters will accomplish the conversion;

(c) if the system comprises more than one storage cell,

- determine which of the storage cells will store the converted signals;

(d) control the subsequent retrieval of the stored signals; and

(e) determine which copy of the file is accessed for retrieval.

11. The networked multimedia system of claim 10, wherein

A) the storage server

i) is physically distributed or decentralized across the system.

12. The networked multimedia system according to claim 8, wherein the system is further configured to

A) generate a sequence of graphics rendering events

i) associated with computer generated images and/or overlay graphics at a workstation, and

B) save the generated sequence.

13. The networked multimedia system according to claim 12, wherein

A) the generated sequence of graphics rendering events

LINKPLAY EXHIBIT 1004
Page 3185 of 4822

i) can be played back to render computer generated images and/or overlay graphics

(a) at a workstation,

(b) together with a workstation-user related audio and video

(c) in a determinable temporal relationship, and

B) the system can

i) save the event sequence

(a) related to the displayed computer generated and/or overlay graphics images and/or the audio and video

ii) such that the event sequence can later be reproduced

(a) at a workstation

(b) in the determined temporal relationship.

14. The networked multimedia system according to claim 13, wherein

A) the at least one file includes

i) the saved sequence of graphics rendering events.

15. The networked multimedia system according to claim 6, further comprising:

A) at least one decoder, configured to

i) decode signals retrieved from storage.

16. The networked multimedia system of claims 8, wherein

A) the file is a metafile including

i) at least one of the group of an audio-video file, audio only file, video only file, a graphics event file, a window event file, an application startup event file, a bit map file, a postscript file, a graphics file, a synchronization file and a text file.

17. The networked multimedia system of claim 16, wherein the metafile further includes

i) at least one pointer referencing

(a) a predetermined file or group of files

LINKPLAY EXHIBIT 1004
Page 3186 of 4822

- on one or more of the storage cells

18. A method of using a networked multimedia system comprising the steps of:

A) capturing audio and video at

i) one or more workstations, each including

(a) video and audio reproduction capabilities, and

(b) video and audio capture capabilities;

B) converting the captured audio/video signals,

(a) into a form suitable for storage;

C) storing,

(a) the converted audio/video signals

(b) within a storage cell,

(c) such that the stored converted signals

- can later be retrieved

i. under direction of one or more of the workstations.

19. A method of using a networked multimedia system comprising the steps of:

A) capturing audio and video at

i) one or more workstations, each including

(a) video and audio reproduction capabilities, and

(b) video and audio capture capabilities;

B) converting captured audio/video signals,

i) into a form suitable for storage

ii) by using at least one converter;

C) storing

i) for later retrieval,

LINKPLAY EXHIBIT 1004
Page 3187 of 4822

(a) signals

(b) converted by at least one of the converters,

ii) onto at least one storage cell;

D) transferring

i) previously stored audio/video signals

(a) to another storage cell

- whether included in that or another storage cell;

E) managing

i) the storage and

ii) transfer,

(a) of the audio/video converted signals,

(b) among storage cells.

20. A method of using a networked multimedia system comprising:

A) capturing audio and video at

i) one or more workstations, each including

(a) video and audio reproduction capabilities, and

(b) video and audio capture capabilities;

B) converting the captured audio and video

i) into a form suitable for storage;

C) storing

i) for later retrieval,

ii) the converted audio/video signals

iii) by using

(a) at least one storage cell having

- at least one storage disk,

LINKPLAY EXHIBIT 1004
Page 3188 of 4822

- a storage disk controller

  i. associated with each disk, and

- at least one converter; and

(b) a storage cell manager (AVSM); and

D) storing the converted audio/video signals,

i) by selecting any one of the group consisting of disks, disk controllers, converters, storage cells and storage cell managers (AVSM), according to the following approach:

(a) if a first converter

- of a first cell
- has reached its capacity or bandwidth limit,
- then a second converter, of the first cell, is selected;

(b) if a first disk controller

- of the first cell
- has reached its capacity or bandwidth limit
- then a second disk controller of the first cell is selected;

(c) if a first disk,

- associated with a first of the disk controllers
- has reached its capacity or bandwidth limit,
- then a second disk associated with the first disk controller is selected;

(d) if the first storage cell

- associated with a first storage cell manager (AVSM)
- has reached its capacity or bandwidth limit,

LINKPLAY EXHIBIT 1004
Page 3189 of 4822

- then a second storage cell associated with the first storage cell manager (AVSM) is selected; and

(e) if the first storage cell manager (AVSM) has reached its capacity,

- then a second storage server, is selected, and

ii) storing the converted signals using each of the selected ones of the group.

21. A method of using a networked multimedia system comprising the steps of:

A) capturing audio and video at

i) one or more workstations, each including

(a) video and audio reproduction capabilities, and

(b) video and audio capture capabilities;

B) converting captured audio and/or video,

i) into a form suitable for storage;

C) providing a least one storage cell, each including

i) at least one storage disk

(a) configured to receive and store,

- for later retrieval,
- the converted audio/video signals,

D) providing at least one storage cell manager (AVSM); and

E) performing any one of the group of transferring functions consisting of:

i) transferring

(a) the stored converted signals

(b) between any two storage disks of any storage cell;

ii) transferring

(a) the stored converted signals

LINKPLAY EXHIBIT 1004
Page 3190 of 4822

(b) between any two storage cells,

- under control of a single storage cell manager (AVSM); and

iii) transferring

(a) the stored converted signals

(b) between any two storage cells

- respectively under control of different storage cell managers (AVSM).

22. The method of any one of claims 19, 21 or 22 further comprising the step of:

A) transferring

i) the stored audio/video signals

ii) to another storage cell.

23. The method of any one of claims 19 to 22, further comprising the step of:

A) reproducing

i) audio and/or video

ii) at one or more of the workstations

iii) based on the retrieved signals.

24. The method of any one of claims 19 to 22, wherein

A) converted signals are stored

i) in at least one file, which can be accessed by

(a) more than one workstation at the same time and/or

(b) more than one application type.

25. The method of claim 24, wherein

A) multiple copies of each such file can exist.

26. The method of claim 24, further comprising the steps of:

A) generating a sequence of graphics rendering events

i) associated with computer generated images and/or overlay graphics at a workstation, and

B) saving the generated sequence.

27. The method of claim 26, wherein

A) the generated sequence of graphics rendering events

i) can be played back to render computer generated images and/or overlay graphics

(a) at a workstation,

(b) together with a workstation user related audio and video

(c) in a determinable temporal relationship

the method comprising the step of:

ii) saving the event sequence

(a) related to the displayed computer generated and/or overlay graphics images and/or the audio and video

(b) such that the event sequence can later be reproduced

- at a workstation
- in the determined temporal relationship.

28. The method of claim 27, wherein

A) the at least one file includes

i) the saved sequence of graphics rendering events.

29. The method of claim 25, further comprising the step of:

A) decoding signals retrieved from storage.

30. A networked multimedia system comprising:

A) a plurality of workstations, each including

LINKPLAY EXHIBIT 1004
Page 3192 of 4822

i) video and audio reproduction capabilities, and

ii) video and audio capture capabilities;

B) at least one signal converter configured to

i) convert audio and/or video information

(a) originating at a workstation,

(b) into a form suitable for storage

C) at least one storage cell configured to

i) store,

(a) audio and video information

- captured at a workstation

(b) as at least one file,

- for later retrieval;

D) at least one signal path,

i) interconnecting

(a) the workstations, the converter(s) and the storage cell(s); and

E) at least one pointer referencing

i) a predetermined file or group of files,

(a) on one or more of the storage cells,

ii) wherein the system is configured to transmit,

(a) at least the pointer

(b) to a receiving application program

iii) render the contents

(a) of the file or group of files

(b) at a workstation

LINKPLAY EXHIBIT 1004
Page 3193 of 4822

(c) by retrieving data

- from the file or group of files

31. The networked multimedia system of claim 30, wherein

A) the pointer file

i) has at least one function selected from the group consisting of

(a) identifying a data file,

(b) identifying at least one storage cell on which the data file is stored,

(c) identifying a media type of the data file,

(d) providing access authorization to the data file,

(e) providing a natural-language title of the data file,

(f) providing a natural-language description of the data file,

(g) identifying a creator of the data file, and

(h) noting a time of creation of the data file.

32. A networked multimedia system comprising:

A) a plurality of workstations, each including

i) video and audio reproduction capabilities, and

ii) video and audio capture capabilities;

B) at least one signal converter configured to

i) convert audio and/or video information

(a) originating at a workstation,

(b) into a form suitable for storage

C) at least one storage cell configured to

i) store,

(a) audio and video information

LINKPLAY EXHIBIT 1004
Page 3194 of 4822

- captured at a workstation

(b) as at least one file,

- for later retrieval;

D) at least one signal path,

i) interconnecting the workstations, the converter(s) and the storage cell(s);

E) a viewer configured to enable a workstation user to do at least one of

i) setting up connections and sessions;

ii) preparing files for viewing, and

iii) providing a viewer interface for the user to access and manipulate files.

33. The system of claim 32, wherein

A) the viewer is

i) invoked as a standard video call, and

ii) is capable of being merged with at least one of

(a) video calls, video conferences, video messages, multimedia calls, multimedia conferences, and multimedia messages.

34. The system of claim 33, wherein

A) the viewer is implemented as a process primitive.

35. The system of claim 33 wherein

A) the viewer includes a plurality of viewers,

the system further comprising

B) a browser for selecting at least one of the plurality of viewers.

36. A networked multimedia system comprising:

A) a plurality of workstations, each including

i) video and audio reproduction capabilities, and

LINKPLAY EXHIBIT 1004
Page 3195 of 4822

ii) video and audio capture capabilities;

B) at least one signal converter configured to

i) convert audio and/or video information

(a) originating at a workstation,

(b) into a form suitable for storage

C) at least one storage cell configured to

i) store,

(a) converted audio and video information

- captured at a workstation

(b) as at least one file,

- for later retrieval;

D) at least one signal path,

i) interconnecting

(a) the workstations, the converter and the storage cell(s);

E) at least one stored-video application program

i) on at least one of

(a) the storage cell

(b) a workstation; and

F) at least one process primitive

i) that can be shared

ii) across at least two of

(a) the stored-video application programs.

37. The networked multimedia system of claim 36, wherein

A) at least one of the stored-video application programs

i) is one from the group consisting of:

LINKPLAY EXHIBIT 1004
Page 3196 of 4822

(a) videoconference recording; video mail; video answering system; video documents; and video publishing.

38. The networked multimedia system of claim 37, wherein

A) the process primitive

i) is one from the group consisting of:

(a) viewer; browser; MIME attachment; an annotation authoring tool; and video editor.

39. The networked multimedia system of claim 38, wherein

A) the system is further configured to

i) share

(a) at least one data primitive

(b) across at least two of

- the plurality of stored-video application programs.

40. The networked multimedia system of claim 39, wherein

A) the data primitive

i) is from the group consisting of

(a) a video file, a graphics event file, and a metafile.

41. The networked multimedia system of claim 40, wherein

A) the metafile includes at least one of the group of

i) an audio-video file, audio only file, video only file, a graphics event file, a window event file, an application startup event file, a bit map file, a postscript file, a graphics file, a synchronization file, and a text file.

42. The networked multimedia system of claim 41, wherein

A) the metafile includes

i) an associated pointer file

B) to define an omnifile..

LINKPLAY EXHIBIT 1004
Page 3197 of 4822

43. The networked multimedia system of claim 39, wherein

A) the system is configured to

i) invoke of at least one of the process primitives and the data primitives as

(a) an attachment to the file,

(b) the attachment being acceptable by third parties according to a standard data interchange protocol.

44. The system of claim 43 wherein

A) the standard data interchange protocol

i) is a MIME protocol, and

B) the attachment

i) is a MIME attachment.

45. The system of claim 44, wherein the MIME attachment

A) invokes one of a viewer and a browser, and

B) initiates signal transfer along the signal path.

46. The system of claim 44, further comprising:

A) a plurality of applications

i) accessible by at least one of the workstations,

ii) the applications being rendered interoperable by at least one of

(a) file sharing by the copying of MIME attachments and

(b) connection sharing.

47. The networked multimedia system of any one of claims 31 to 46 wherein the system is configured to

A) establish real time data sharing sessions

i) between at least two workstations

ii) using a synchronized data sharing process

LINKPLAY EXHIBIT 1004
Page 3198 of 4822

(a) as a shared conference window.

48. The system of claim 47, wherein

A) the synchronized data sharing process is shareboard, and

B) the use of shareboard renders

i) an application program simultaneously invocable on a plurality of workstations

(a) whereby a workstation user can

- access a file,
- import data as a window copy from the application program onto the video reproduction capability of a workstation,
- scroll through screens of the application program independent of other invocations of the applications on other workstations, and
- perform graphics functions utilizing graphics overlay files on the application program

i. independent of other invocations of the application program on other workstations.

49. The system of claim 48, further comprising at least one of the group of:

A) an authoring tool for at least one of

i) drawing objects on the shared conference window, and

ii) typing text on the shared conference window;

B) a grab tool for

i) grabbing another file program, and

ii) drawing at least a portion of the file program

(a) into the shared conference window; and

C) a telepointing tool

i) for independently pointing at objects

LINKPLAY EXHIBIT 1004
Page 3199 of 4822

(a) on the shared conference window.

50. The networked multimedia system of any one of claims 31 to 46, further comprising:

A) at least one video conference application for

i) establishing a video conference session

(a) among the interconnected workstations, and

B) a video conference recording application for

(a) recording at least a portion

- of the established video conference session
- on at least one storage cell.

51. The system of claim 50, wherein

A) the video conference recording application

i) is implemented at the option of at least one user.

52. The system of claim 50, wherein

A) the video conference recording application

i) is automatically implemented as at least one of

(a) an element of a video conference call initiator, and

(b) an element of a video conference call control routine.

53. The system of claim 50, wherein

A) the signal path includes at least

i) a conference bridge

(a) which is invoked at the initiation of the video conference recording application.

54. The system of claim 50, wherein

A) the signal path includes at least one

i) conference bridge

LINKPLAY EXHIBIT 1004
Page 3200 of 4822

(a) previously invoked at the initiation of the video conference session.

55. The system of claim 50, wherein

A) the signal path includes

i) a conference bridge port

(a) providing a global view of the video conference session participants.

56. The system of claim 50, wherein

A) the signal path includes

i) a normally configured conference bridge port

(a) treated as a standard user port on the conference bridge.

57. The networked multimedia system of any one of claims 31 to 46 further comprising:

A) at least one video mail application configured to do at least one of

i) author a video mail message for transmission to the storage cell and

ii) read a video mail message previously stored on the storage cell.

58. The system of claim 57, wherein

A) the video mail application includes

i) a MIME-based video attachment capability

(a) provided to an existing MIME-based mail application.

59. The system of claim 57, wherein

A) the video mail application is configured to

i) share

(a) a video mail message

- across a plurality of workstations as

(b) a broadcast video e-mail,

LINKPLAY EXHIBIT 1004
Page 3201 of 4822

- thereby requiring the storage of fewer copies of the message than the number of the plurality of workstations receiving the broadcast video e-mail.

60. The networked multimedia system of any one of claims 31 to 46, wherein the system is configured to

A) to record

B) on a storage cell, at least one of

i) an audio and video message, and

ii) a multimedia message

C) from an incoming caller.

61. The system of claim 60 further configured to

A) record the video message

i) on one of

(a) the recipient's storage cell, and

(b) the caller's storage cell such that

ii) the message is stored to optimize at least one of

(a) message availability,

(b) system overhead for system network configuration, and

(c) system data storage facilities.

62. The system of claim 60, further configured to

A) respond to a call

i) to a workstation user

ii) by producing a recorded greeting selected from the group consisting of

(a) a stored video greeting,

(b) a text greeting, and

(c) a graphics greeting.

LINKPLAY EXHIBIT 1004
Page 3202 of 4822

63. The system of claim 62, wherein

A) the recorded greeting

i) is stored on a storage cell

ii) associated with the workstation user.

64. The system of claim 62, wherein

A) at least one recorded greeting

B) is stored on a plurality of storage cells.

65. A networked multimedia system comprising:

A) a plurality of workstations, each including

i) video and audio reproduction capabilities, and

ii) video and audio capture capabilities;

B) at least one signal converter configured to

i) convert audio and/or video information

(a) originating at a workstation,

(b) into a form suitable for storage

C) at least one storage cell configured to

i) store,

(a) audio and video information

- captured at a workstation

(b) as at least one file,

- for later retrieval;

D) at least one signal path,

i) interconnecting the workstations, the converter(s) and the storage cell(s)

E) wherein the system is configured to

LINKPLAY EXHIBIT 1004
Page 3203 of 4822

i) enable files

(a) stored on a storage cell

(b) to be included in an on-line electronic document.

66. The system of claim 65 where the system

A) provides access to stored files

i) by utilizing a standardized attachment methodology.

67. The system of claim 66, wherein

A) the standardized attachment methodology is MIME.

68. The system of claim 66, wherein

A) the files are selected from the group consisting of

i) archival video, video overlay graphics, and a compendium of a plurality of video files which further include at least one of additional text or graphics, standalone videos, and MIME attached video files.

69. A networked multimedia system comprising:

A) a plurality of workstations, each including

i) video and audio reproduction capabilities, and

ii) video and audio capture capabilities;

B) at least one signal converter configured to

i) convert audio and/or video information

(a) originating at a workstation,

(b) into a form suitable for storage

C) at least one storage cell configured to

i) store,

(a) audio and video information

- captured at a workstation

(b) as at least one file,

LINKPLAY EXHIBIT 1004
Page 3204 of 4822

◆ for later retrieval;

D) at least one network providing

i) a signal path,

(a) interconnecting the workstations, the converter(s) and the storage cell(s)

E) wherein the system defines a proxy server configured to do at least one of

i) reducing the total number of copies of video files by file sharing,

ii) reducing network load by transporting video files over a separate audio-video network,

iii) transcoding among multiple audio-video formats, and

iv) allowing any one of the workstations to serve as a video publisher for the system.

70. The system of claim 69, wherein

A) the workstation serves as a video publisher

i) subject to user permissions.

71. The system of claim 69, wherein the system is configured

A) as an intranet video storage server for at least one of

i) reducing the total number of copies of video files in the enterprise by file sharing,

ii) reducing enterprise network load by transporting video files over a separate audio-video network,

iii) transcoding among multiple audio-video formats, and

iv) allowing any audio-video workstation in the enterprise to serve as a video publisher for the enterprise.

72. The networked multimedia system of claim 69, wherein

A) the system is configured as a proxy server

i) enabling the implementation of at least one animated annotation on the internet.

LINKPLAY EXHIBIT 1004
Page 3205 of 4822

73. The system of claim 72, wherein

A) at least one of the animated annotations

i) is synchronized with respect to one or more of

(a) an accompanying video file,

(b) an accompanying audio file.

74. The system of claim 71, further configured to

A) enable the authoring of an animated annotation utilizing a window-sharing tool.

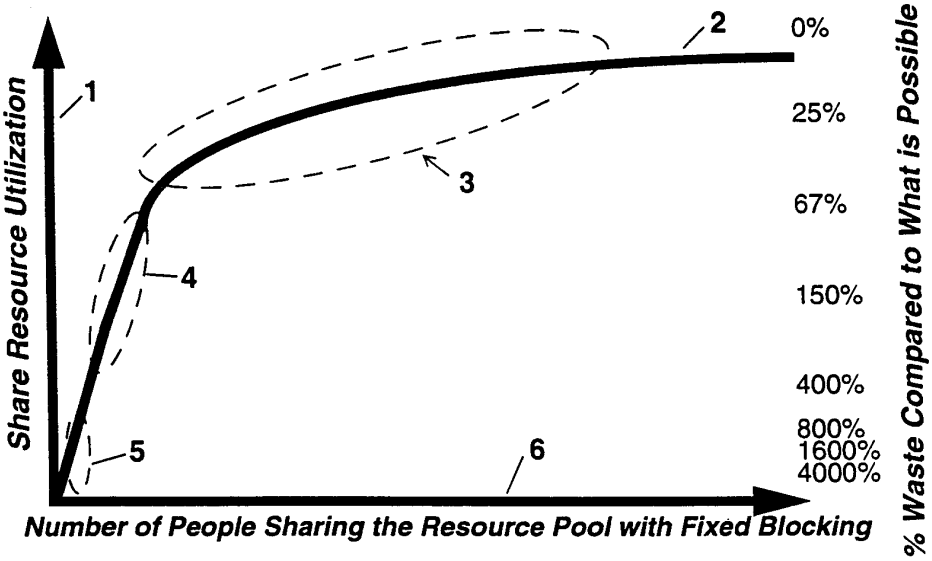
75. The system of claim 74, wherein

A) the window-sharing tool is selected from the group consisting of

i) a shareboard application and an interactive meeting enabling computer application.

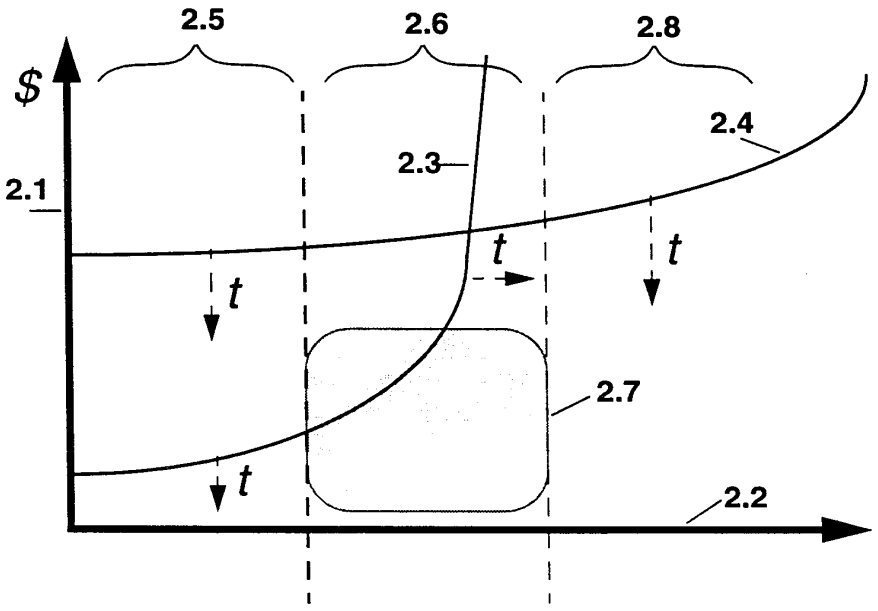
LINKPLAY EXHIBIT 1004
Page 3206 of 4822

Figure 1



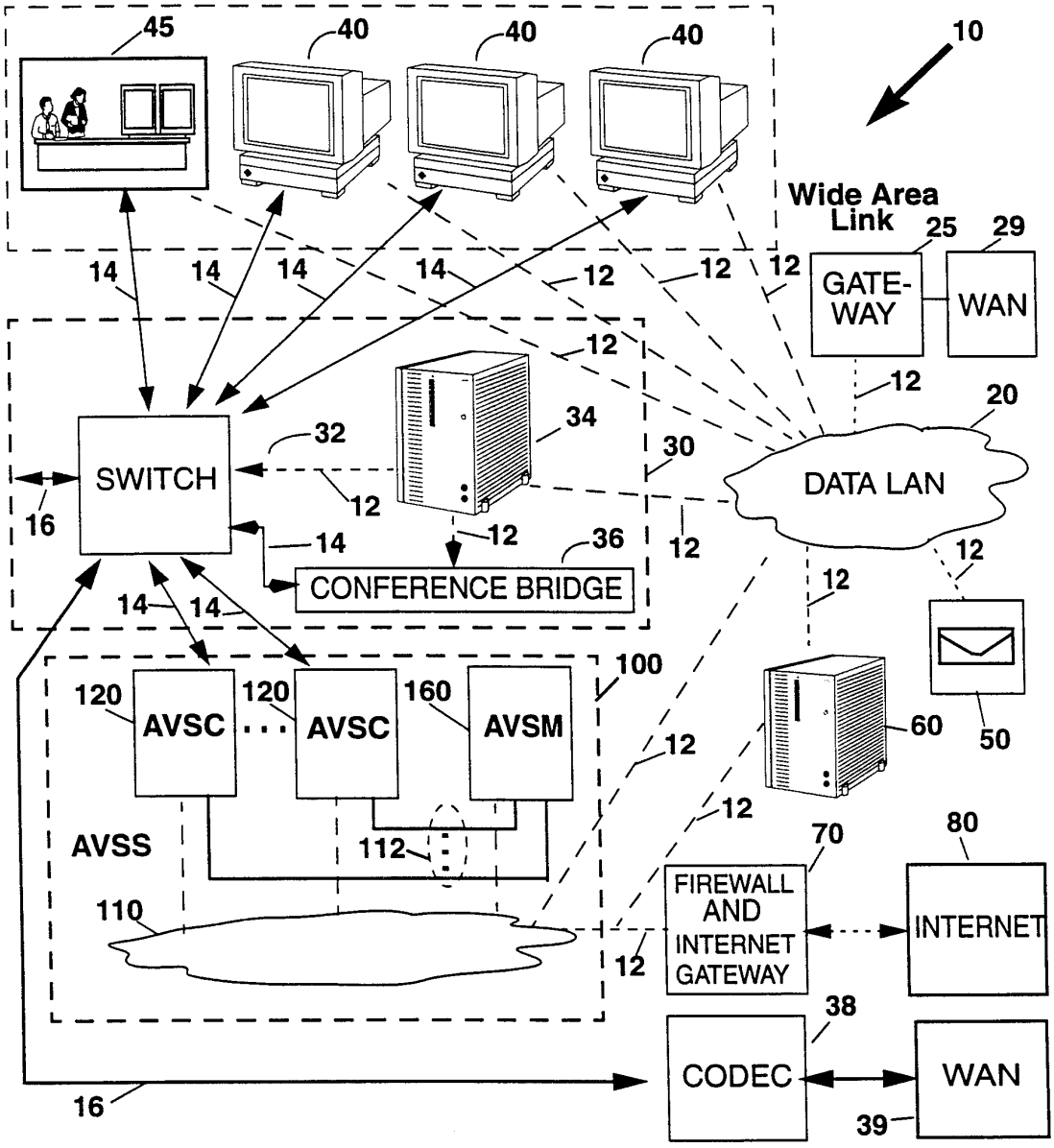
LINKPLAY EXHIBIT 1004
Page 3207 of 4822

Figure 2



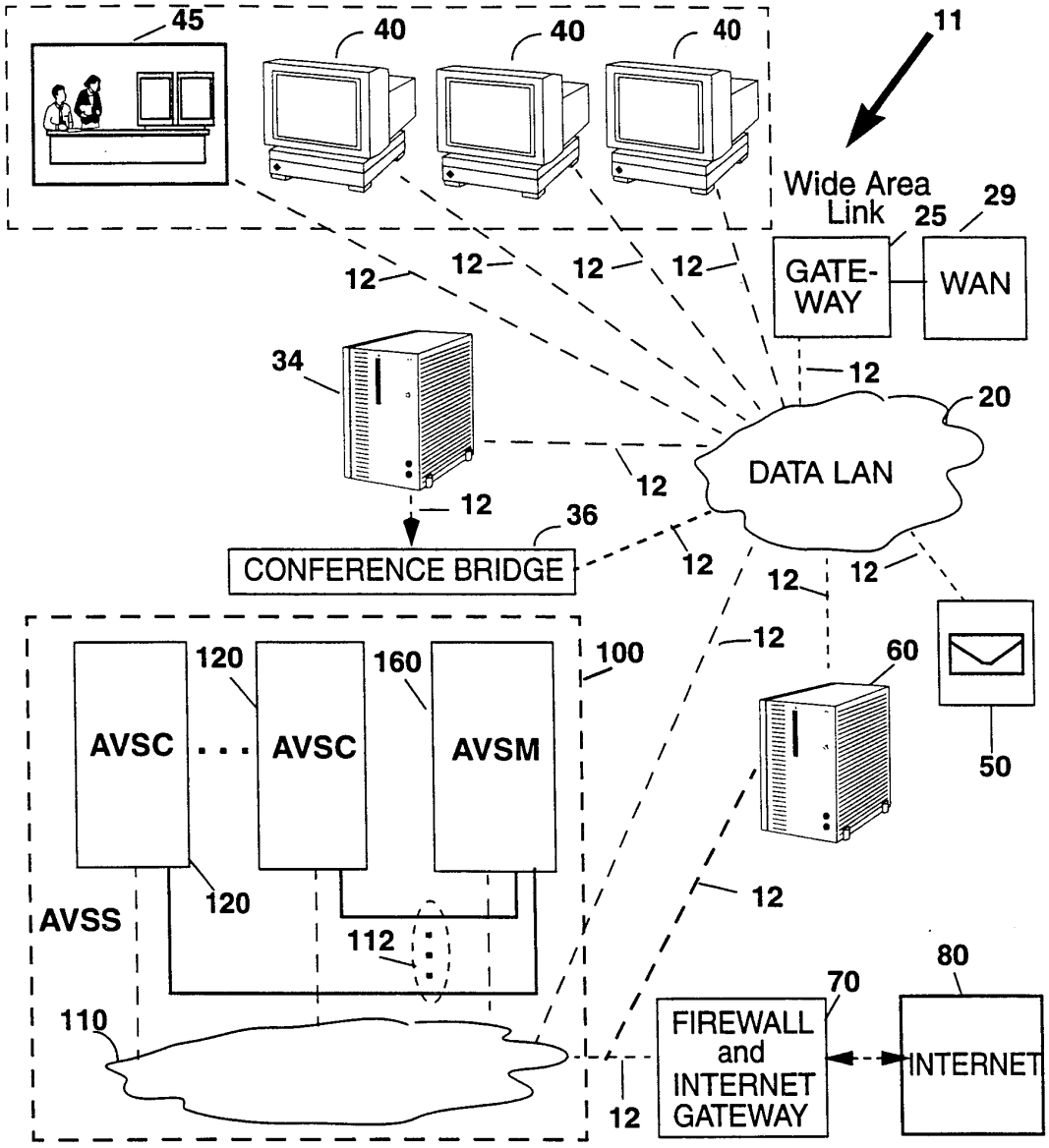
LINKPLAY EXHIBIT 1004
Page 3208 of 4822

Figure 3


45
40
40
40
10
Wide Area Link
25
29
GATE-WAY
WAN
14
14
14
14
12
12
12
12
12
20
34
32
30
SWITCH
DATA LAN
16
12
14
12
36
12
CONFERENCE BRIDGE
12
12
14
14
12
100
120
AVSC
120
AVSC
160
AVSM
60
50
12
12
AVSS
112
70
80
FIREWALL AND INTERNET GATEWAY
INTERNET
110
12
38
CODEC
WAN
16
39

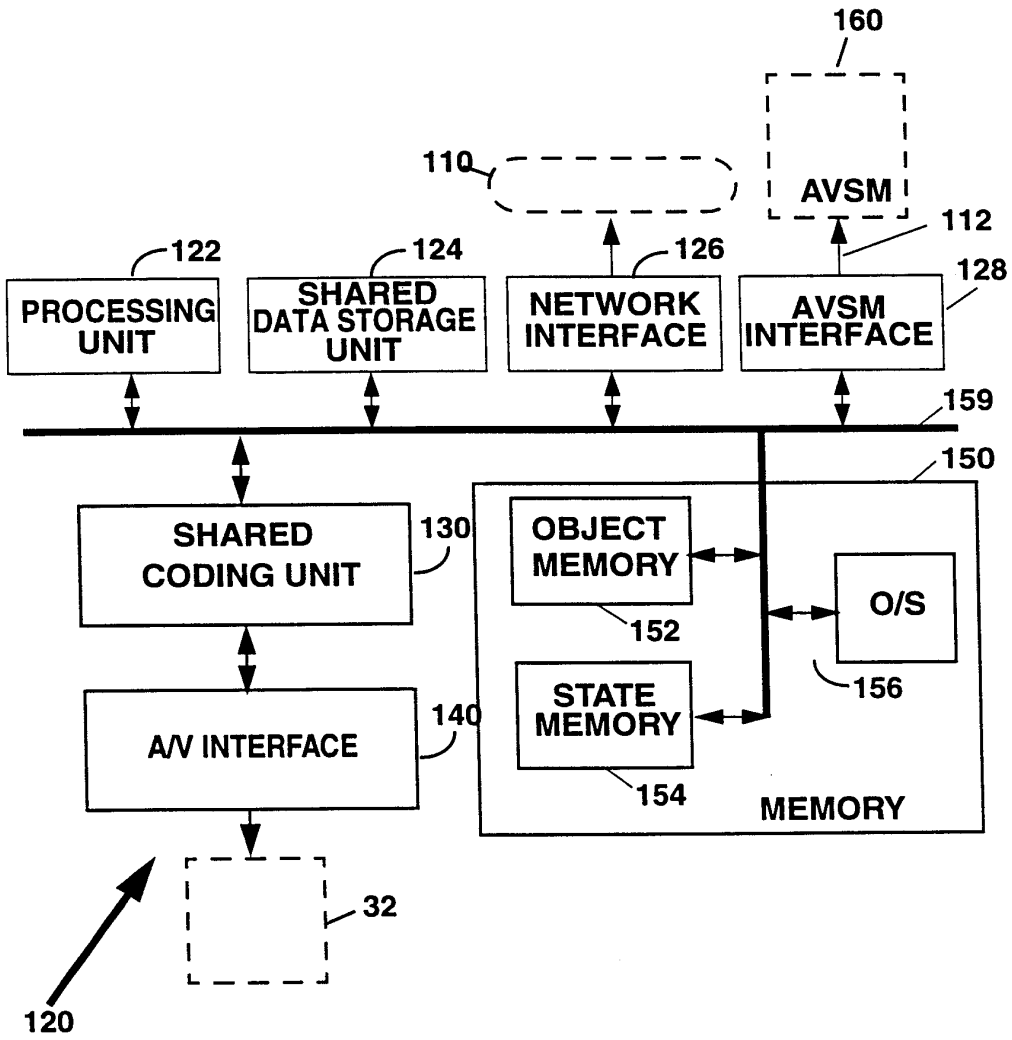
3/46

Figure 4


45
40
40
40
11
Wide Area Link
25
29
GATE-WAY
WAN
12
34
20
DATA LAN
36
CONFERENCE BRIDGE
AVSC
AVSC
AVSM
120
160
100
60
50
AVSS
112
110
70
80
FIREWALL and INTERNET GATEWAY
INTERNET

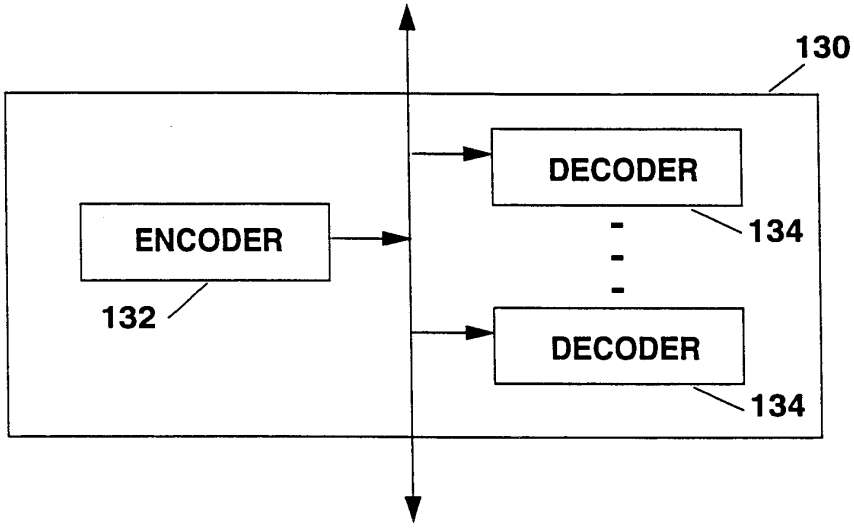
4/46

Figure 5



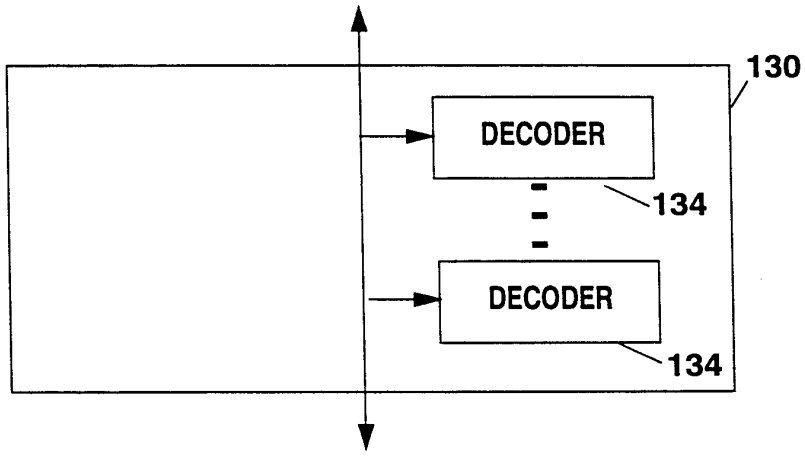
LINKPLAY EXHIBIT 1004
Page 3211 of 4822

Figure 6



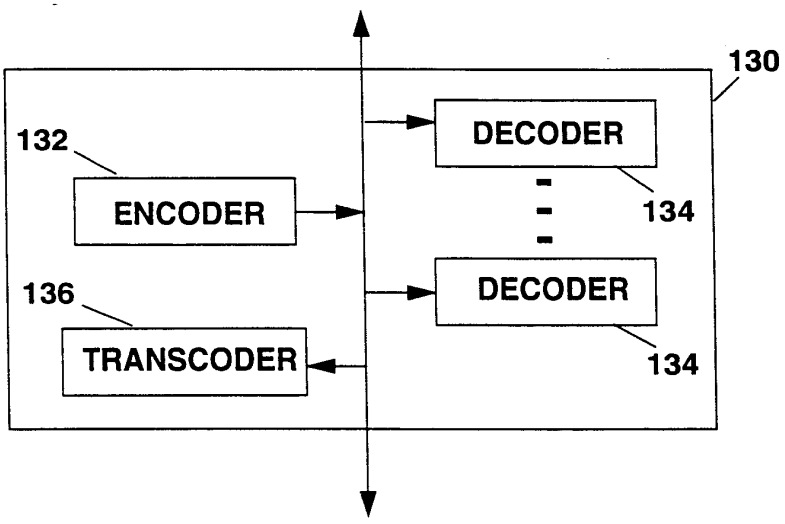
LINKPLAY EXHIBIT 1004
Page 3212 of 4822

Figure 7



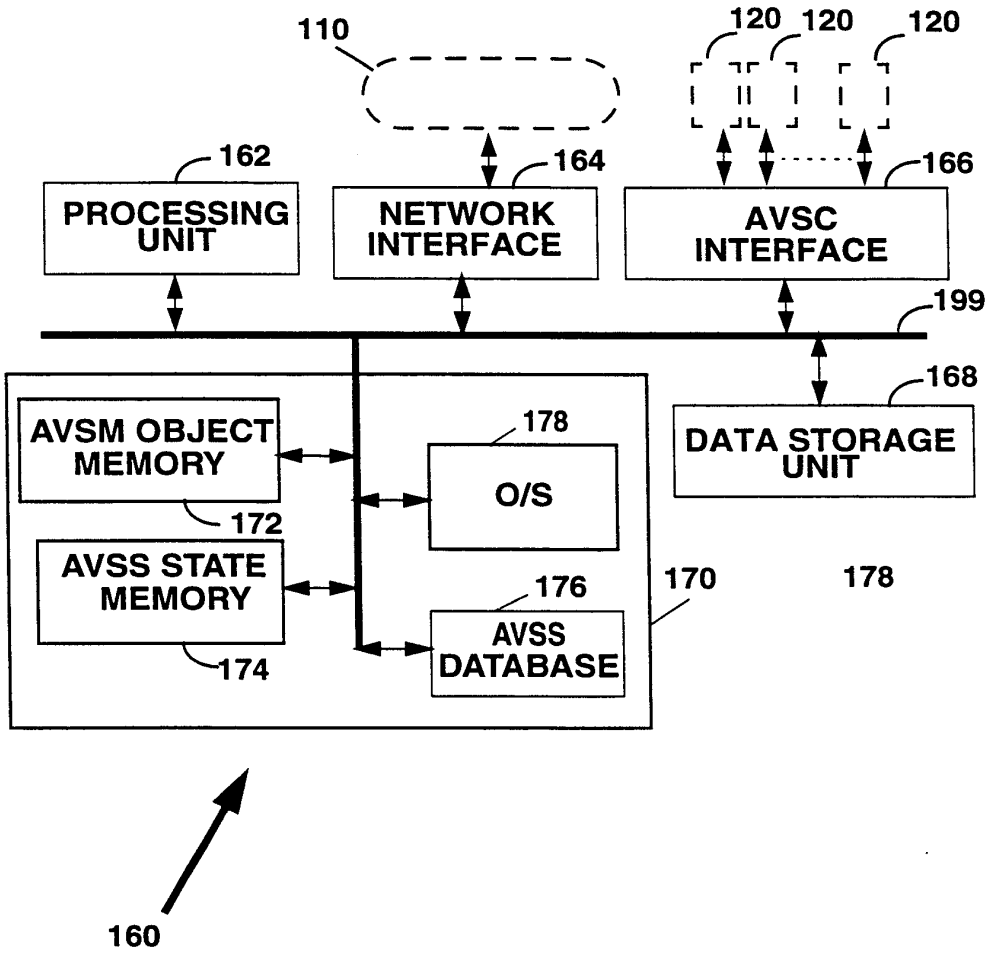
LINKPLAY EXHIBIT 1004
Page 3213 of 4822

Figure 8



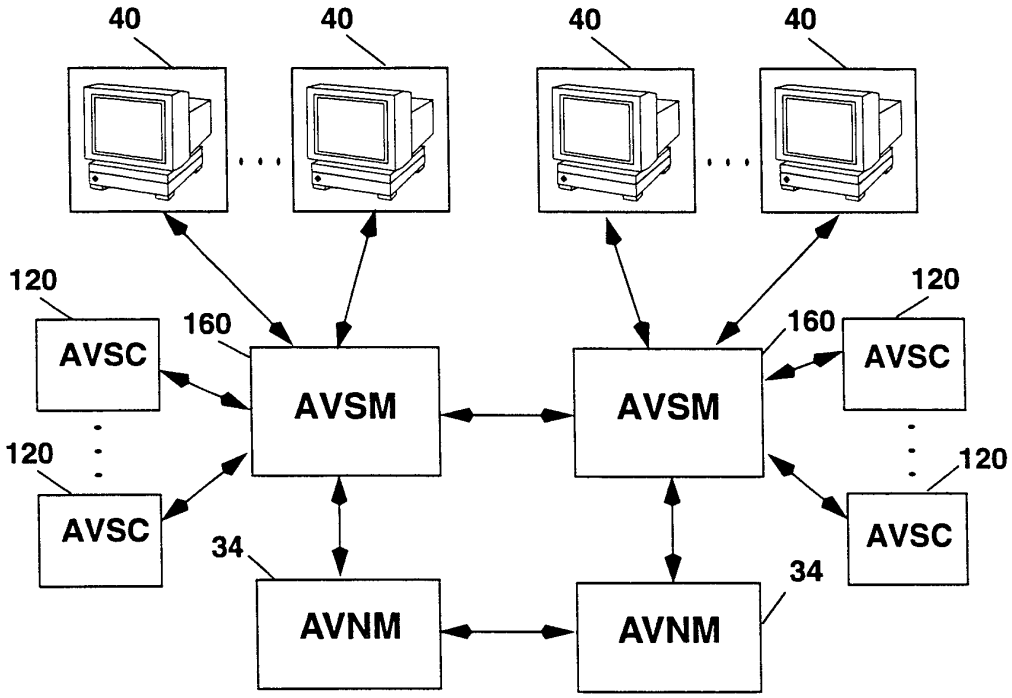
LINKPLAY EXHIBIT 1004
Page 3214 of 4822

Figure 9



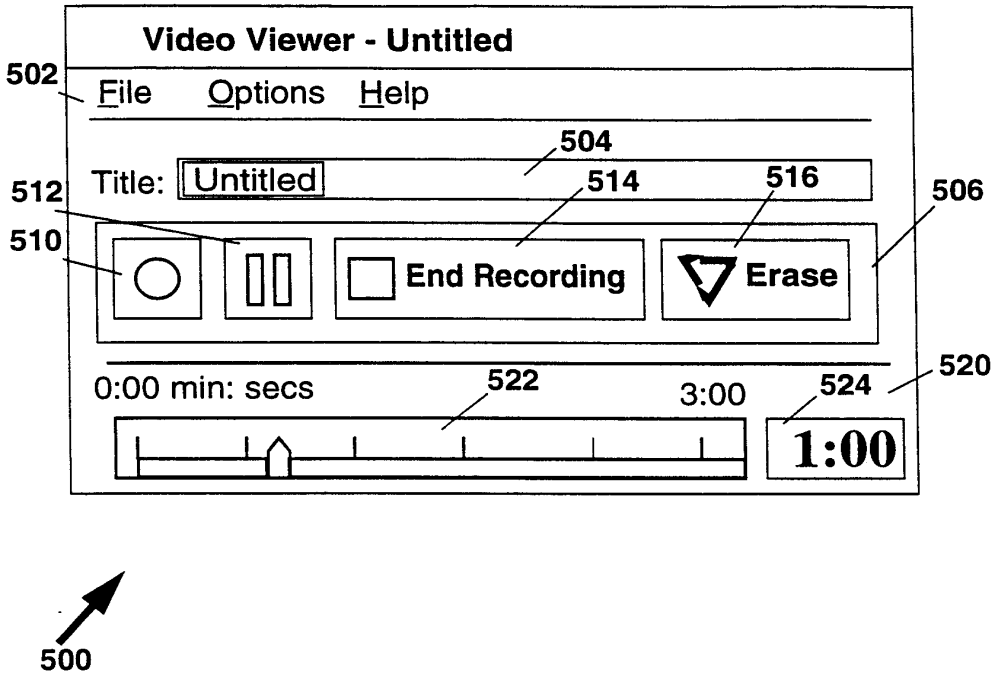
LINKPLAY EXHIBIT 1004
Page 3215 of 4822

Figure 10



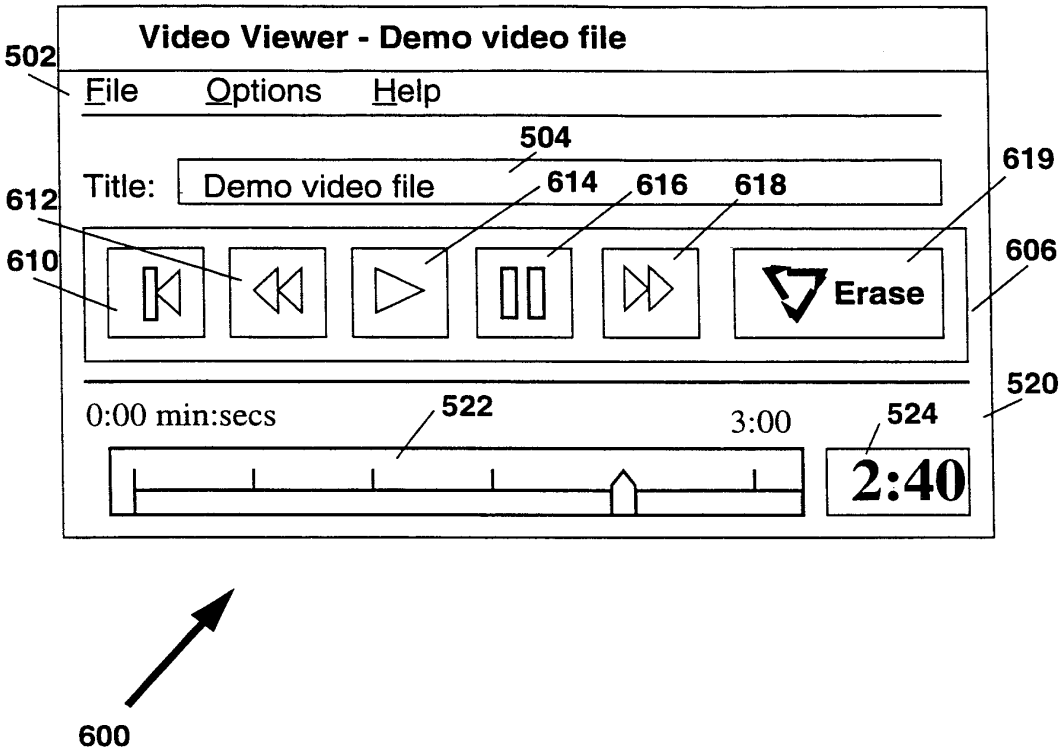
LINKPLAY EXHIBIT 1004
Page 3216 of 4822

Figure 11



LINKPLAY EXHIBIT 1004
Page 3217 of 4822

Figure 12



LINKPLAY EXHIBIT 1004
Page 3218 of 4822

Figure 13

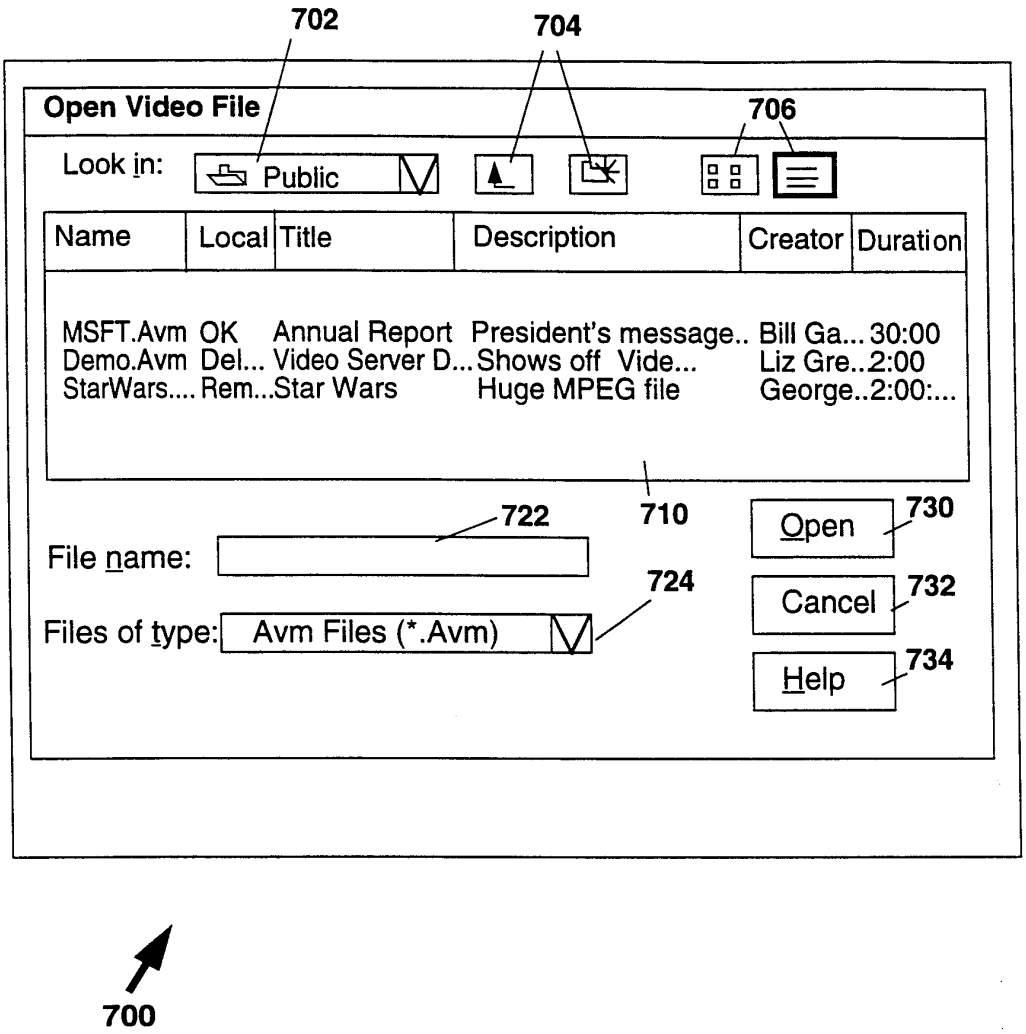
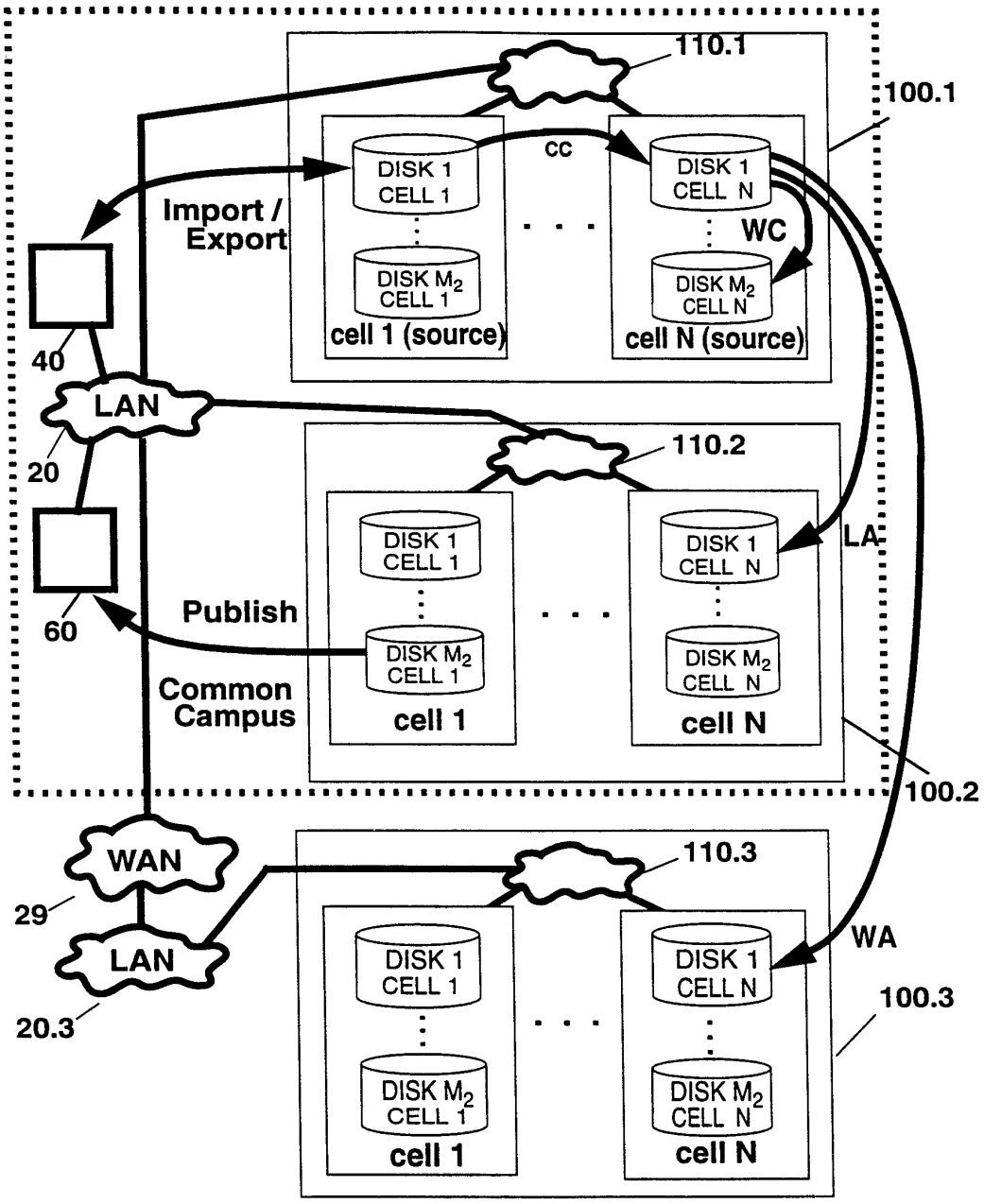
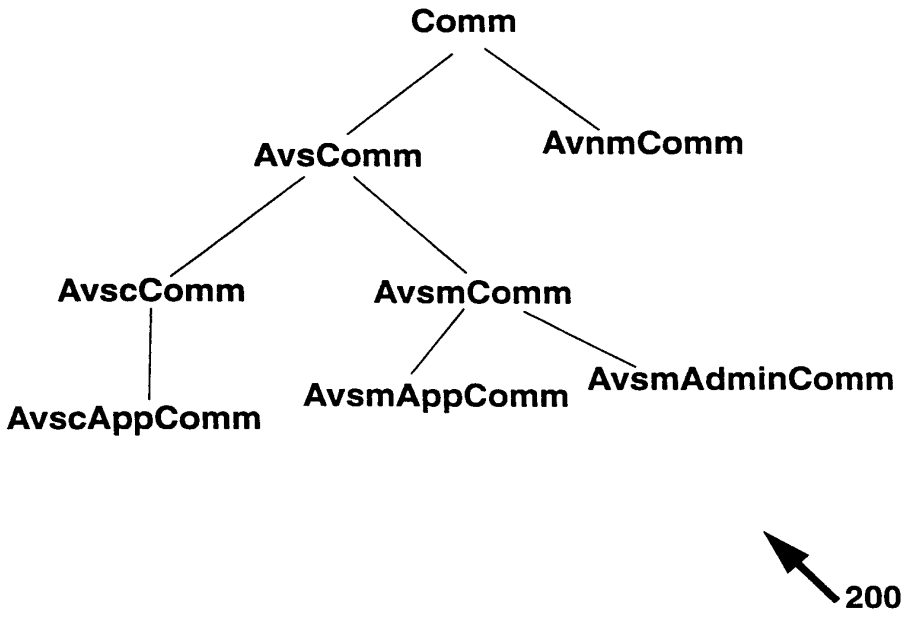

702
704
706
Open Video File
Look in:
Public
Name
Local
Title
Description
Creator
Duration
MSFT.Avm OK Annual Report President's message.. Bill Ga... 30:00
Demo.Avm Del... Video Server D... Shows off Vide... Liz Gre..2:00
StarWars.... Rem...Star Wars Huge MPEG file George..2:00:...
710
722
File name:
724
Files of type:
Avm Files (*.Avm)
Open
730
Cancel
732
Help
734
700

Figure 14



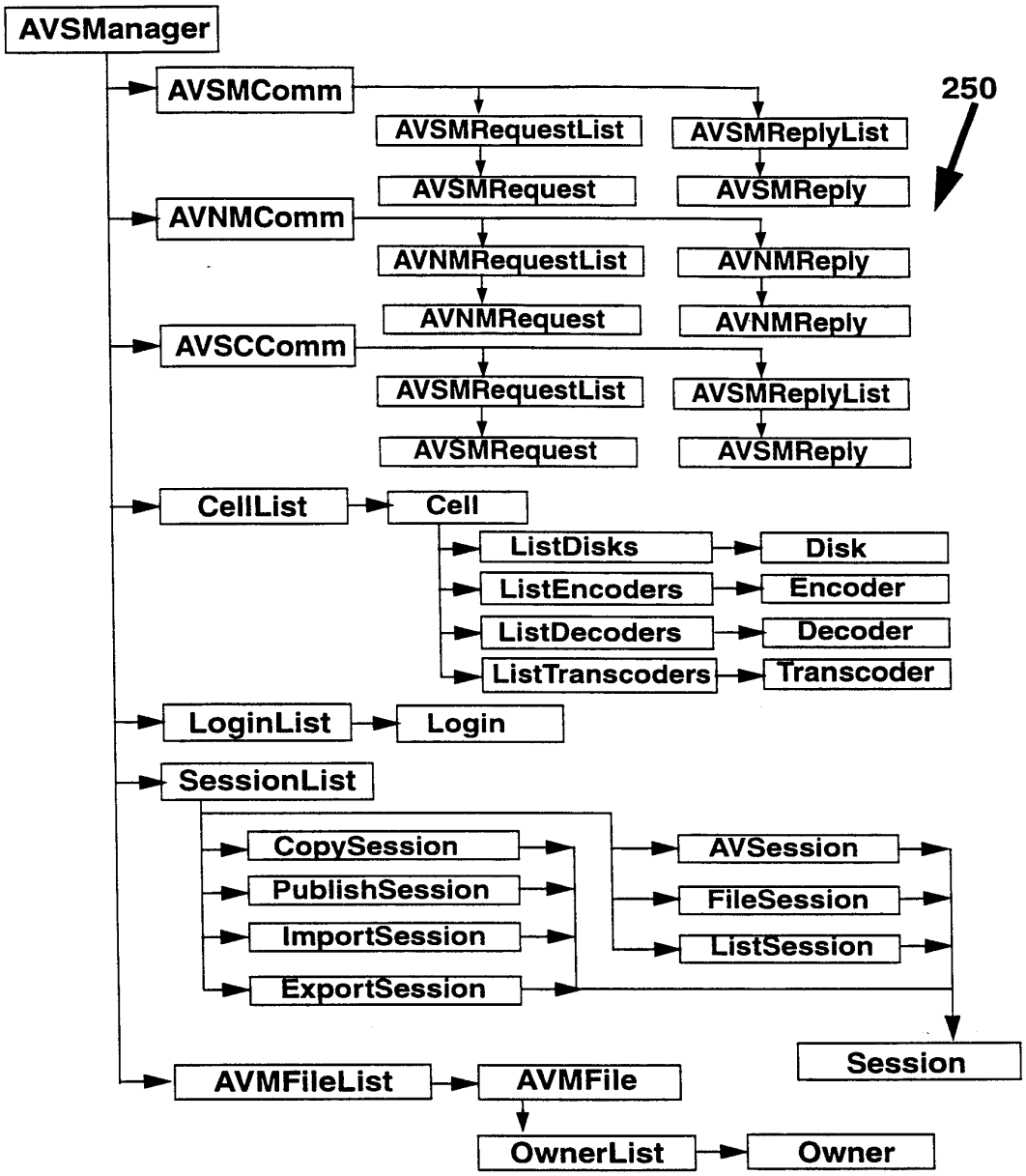
14/46

Figure 15



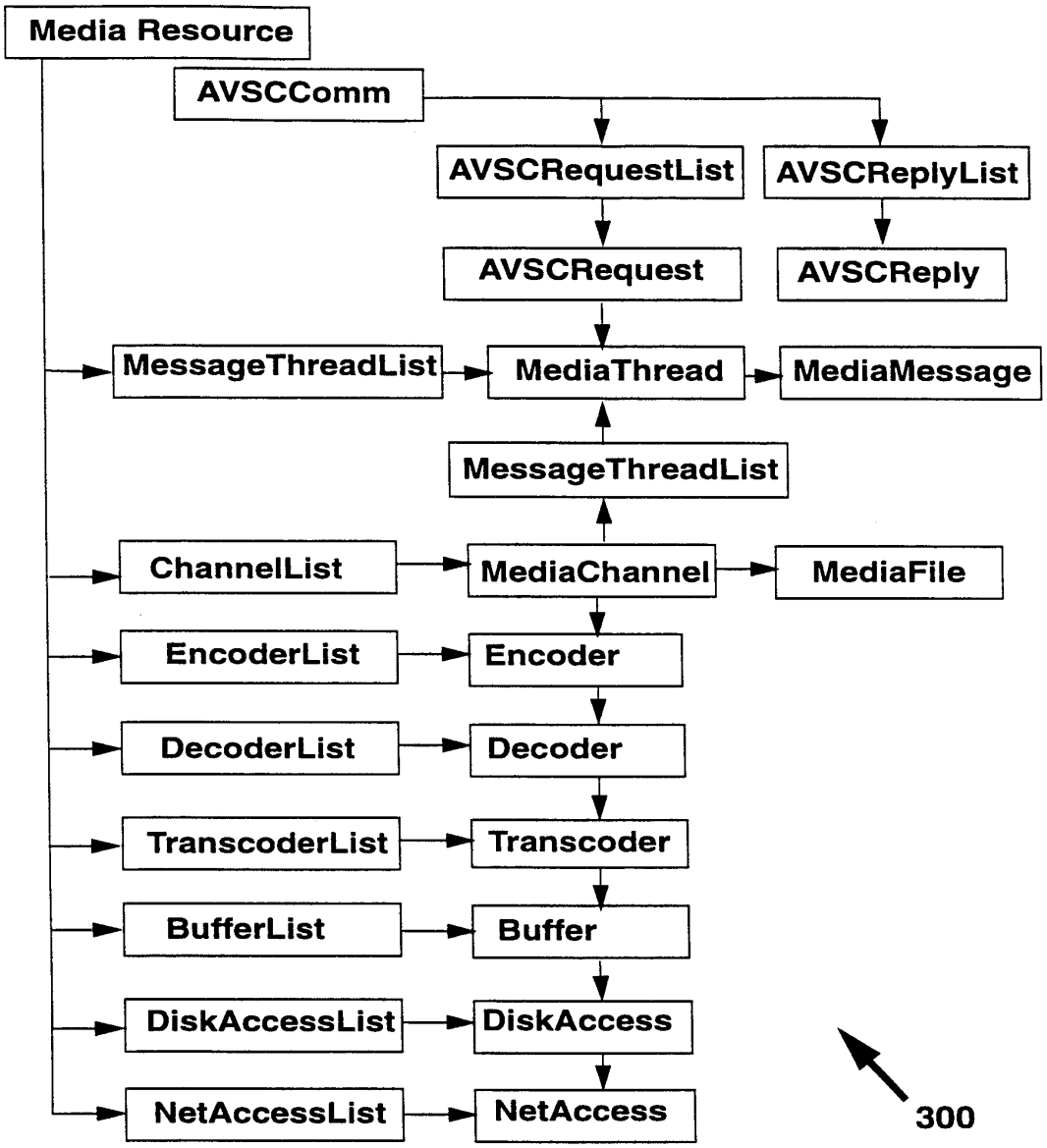
LINKPLAY EXHIBIT 1004
Page 3221 of 4822

Figure 16



LINKPLAY EXHIBIT 1004
Page 3222 of 4822

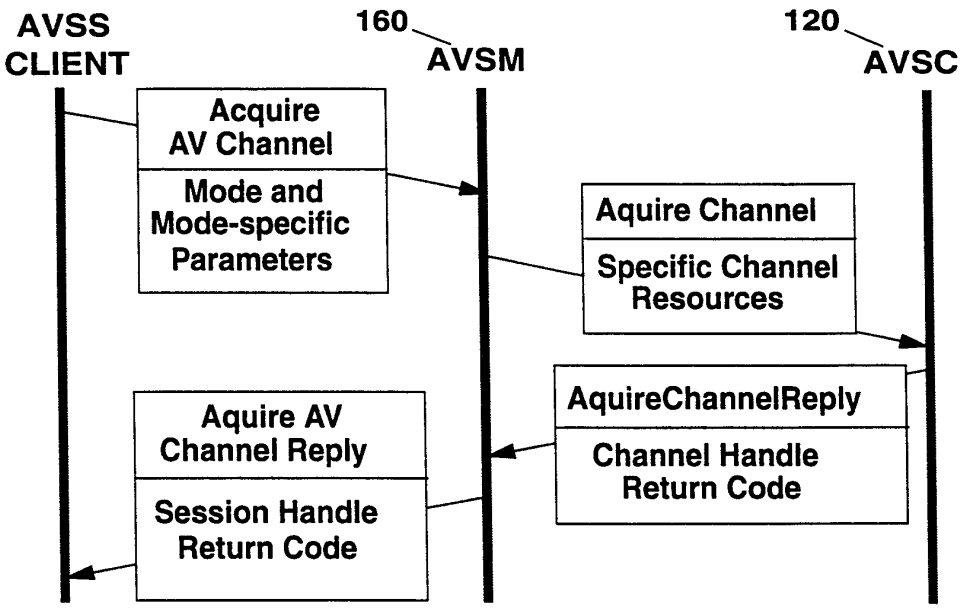
Figure 17



17/46

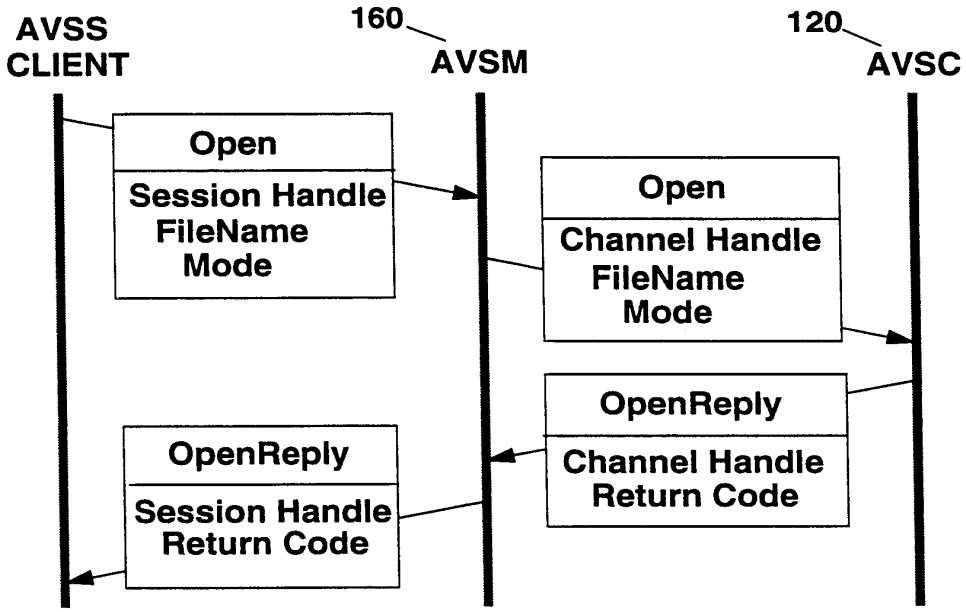
LINKPLAY EXHIBIT 1004
Page 3223 of 4822

Figure 18



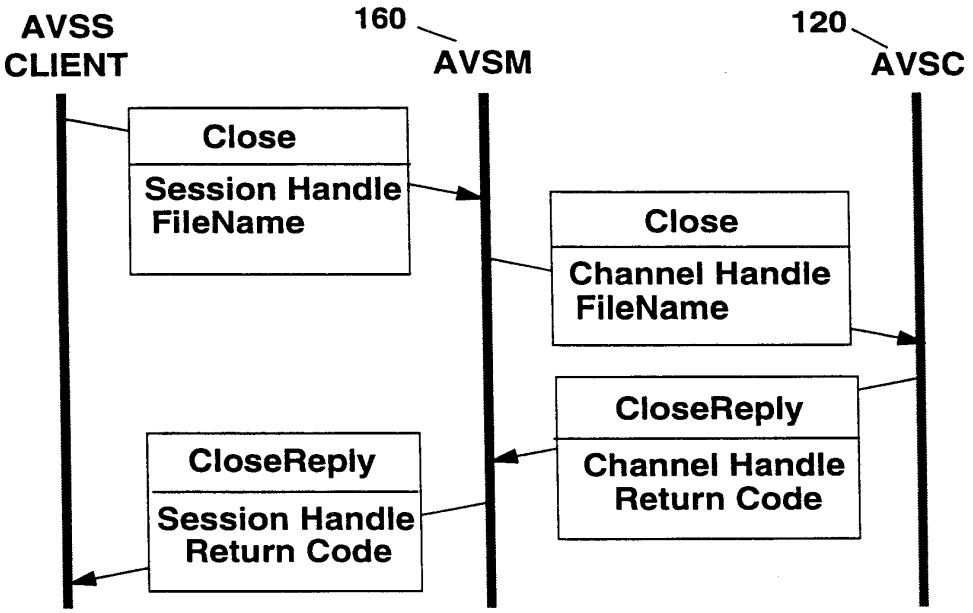
LINKPLAY EXHIBIT 1004
Page 3224 of 4822

Figure 19



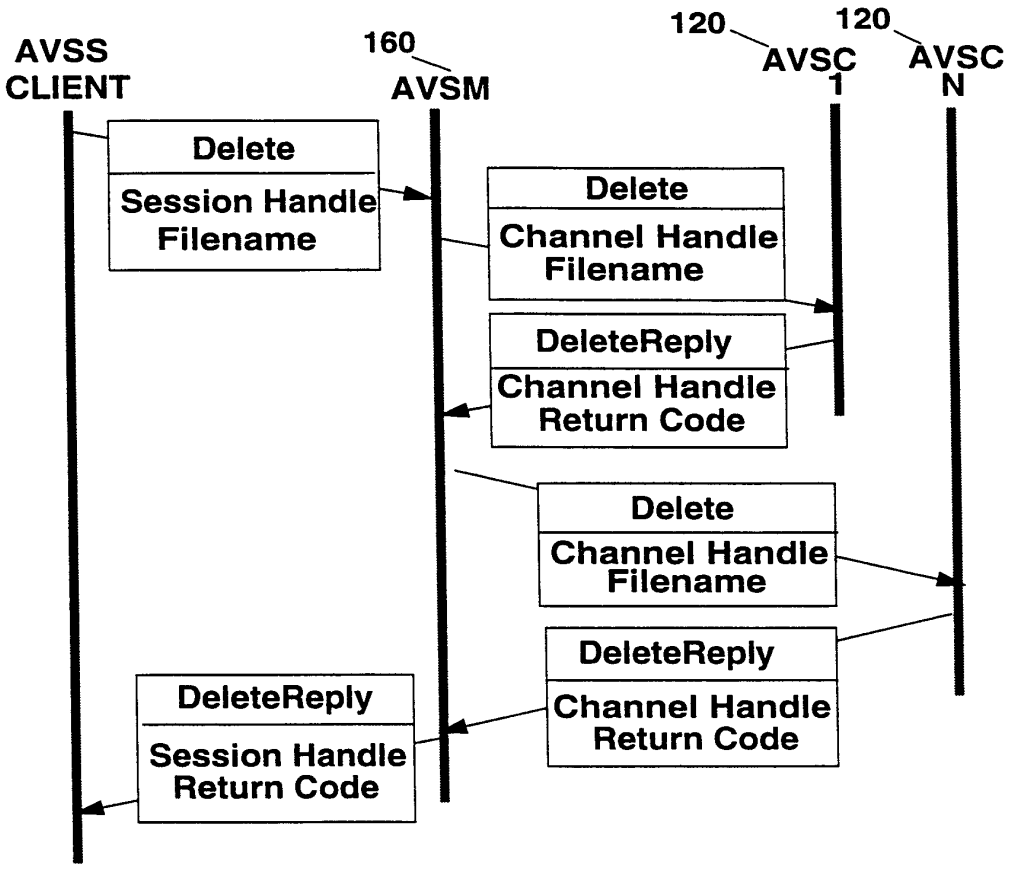
LINKPLAY EXHIBIT 1004
Page 3225 of 4822

Figure 20



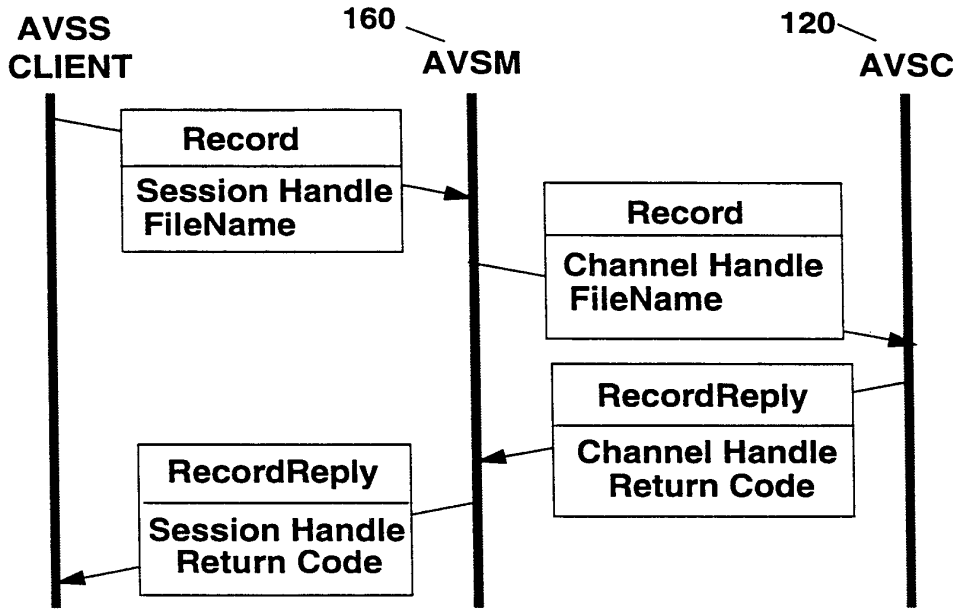
LINKPLAY EXHIBIT 1004
Page 3226 of 4822

Figure 21



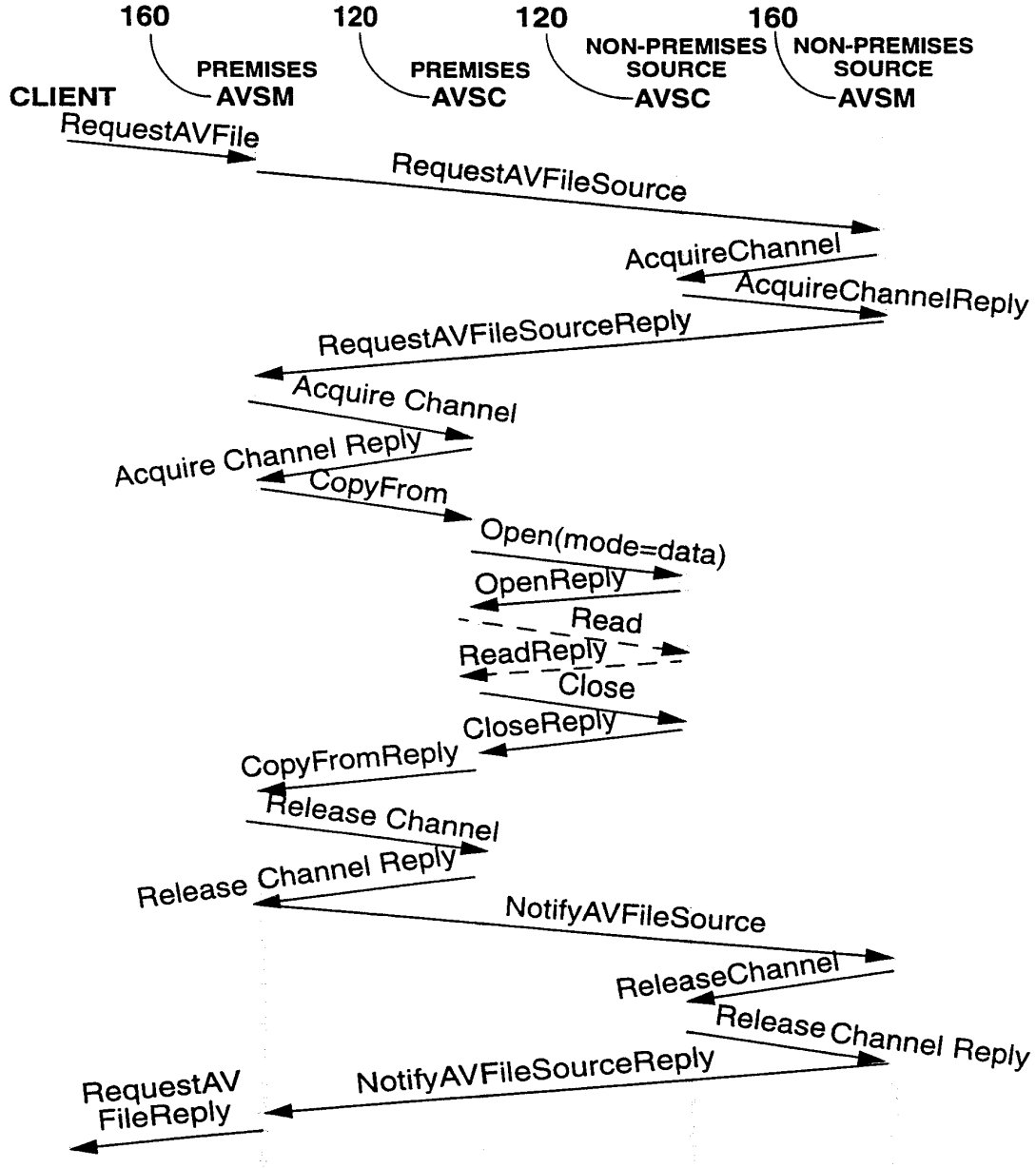
LINKPLAY EXHIBIT 1004
Page 3227 of 4822

Figure 22



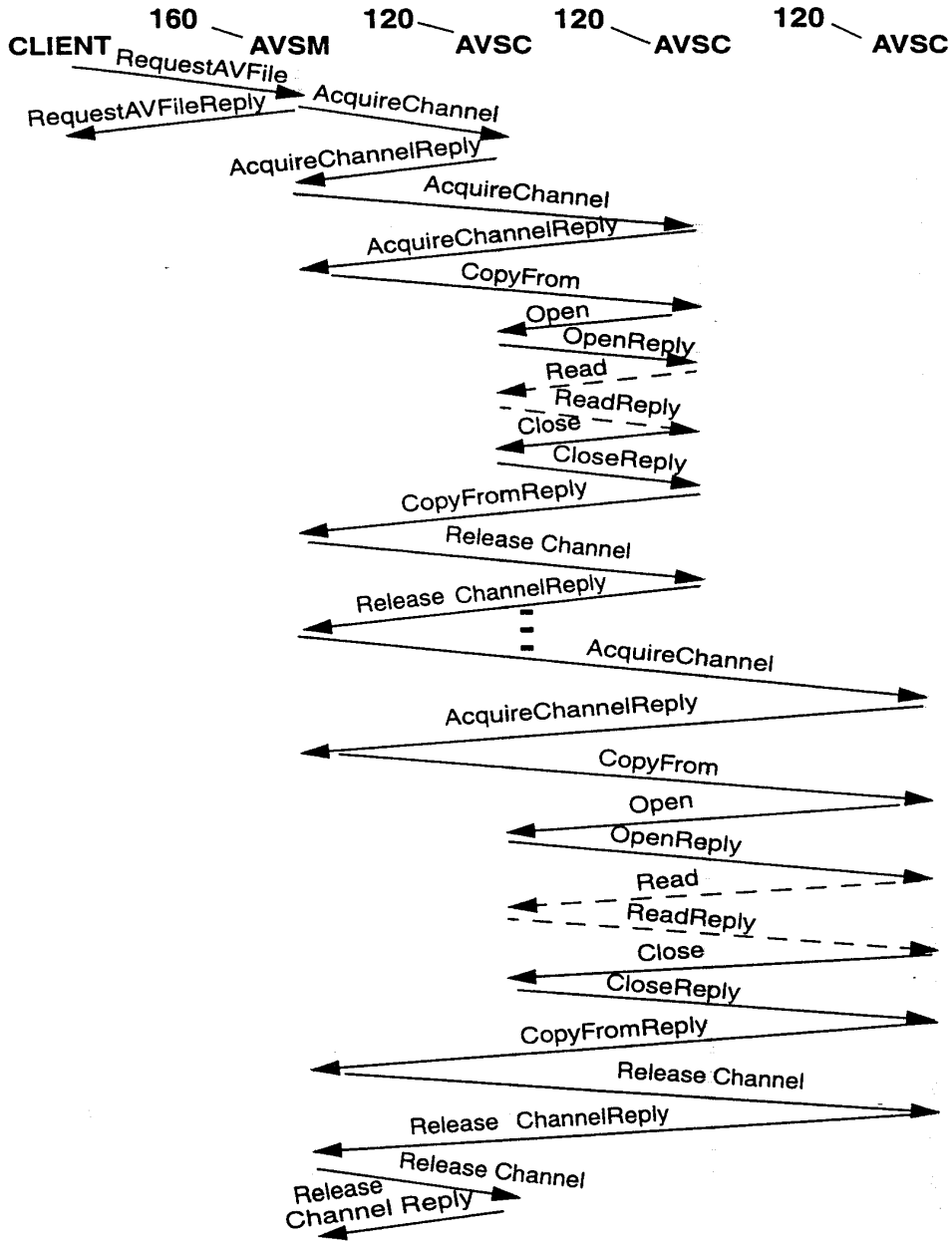
LINKPLAY EXHIBIT 1004
Page 3228 of 4822

Figure 23



LINKPLAY EXHIBIT 1004
Page 3229 of 4822

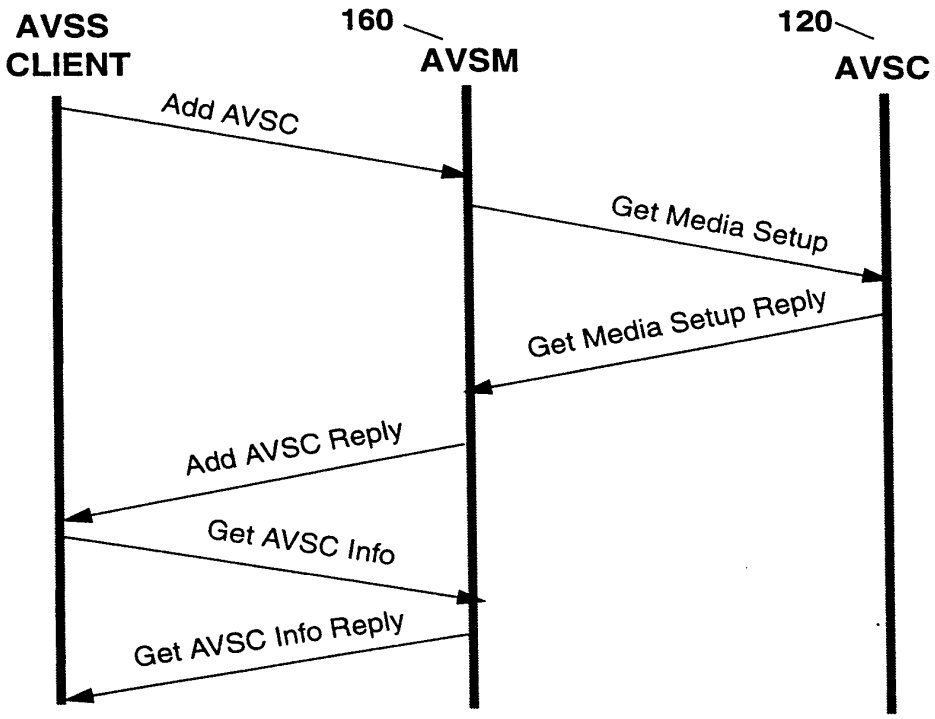
Figure 24



24/46

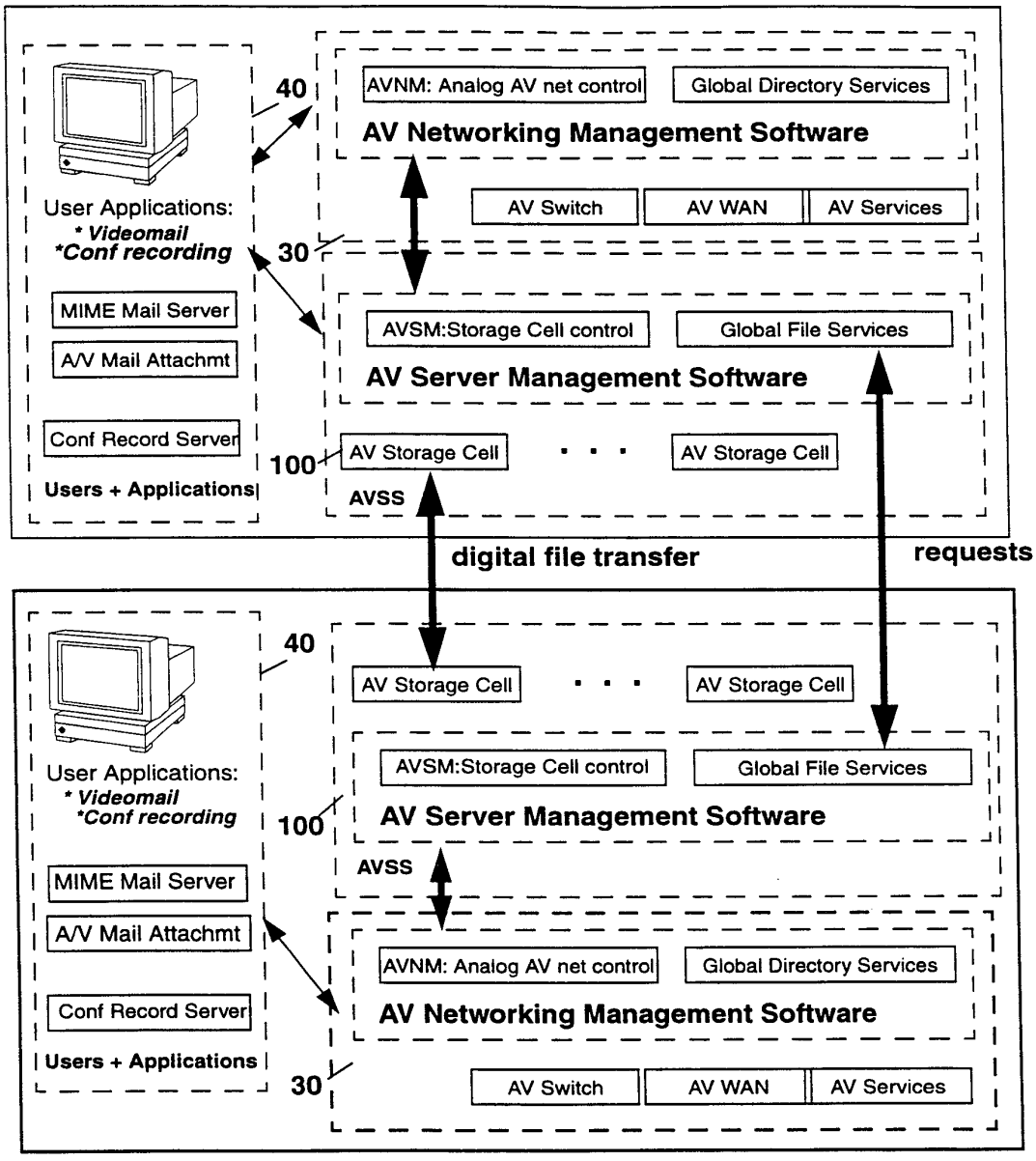
LINKPLAY EXHIBIT 1004
Page 3230 of 4822

Figure 25



LINKPLAY EXHIBIT 1004
Page 3231 of 4822

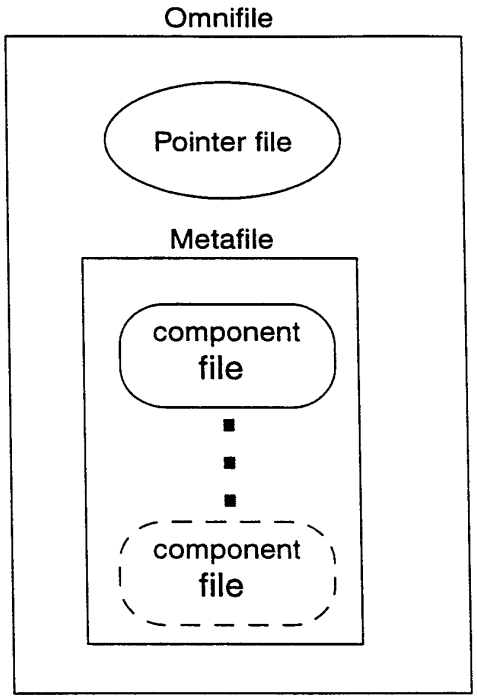
Figure 26



26/46

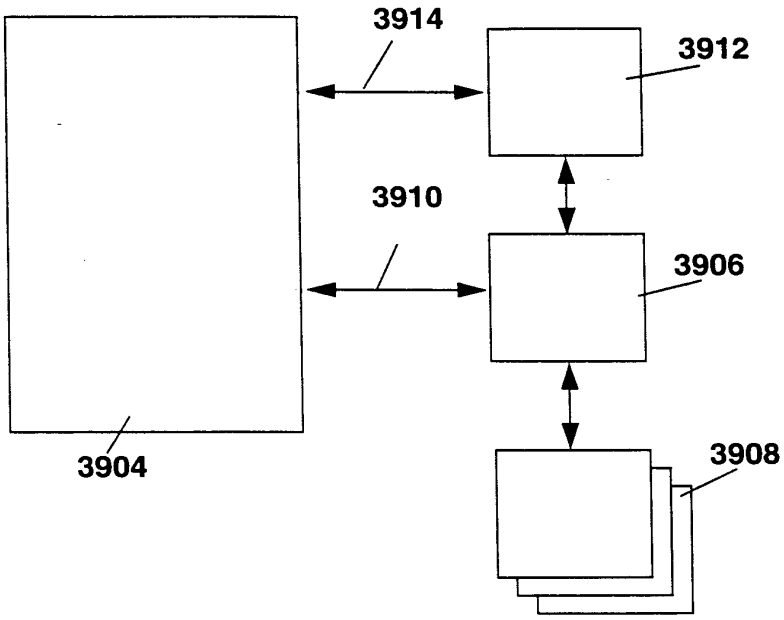
LINKPLAY EXHIBIT 1004
Page 3232 of 4822

Figure 27



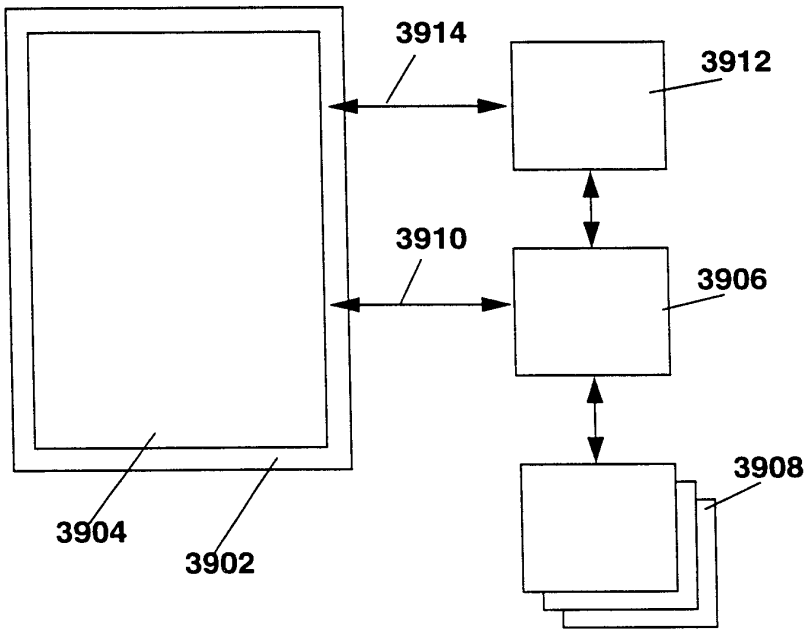
LINKPLAY EXHIBIT 1004
Page 3233 of 4822

Figure 28



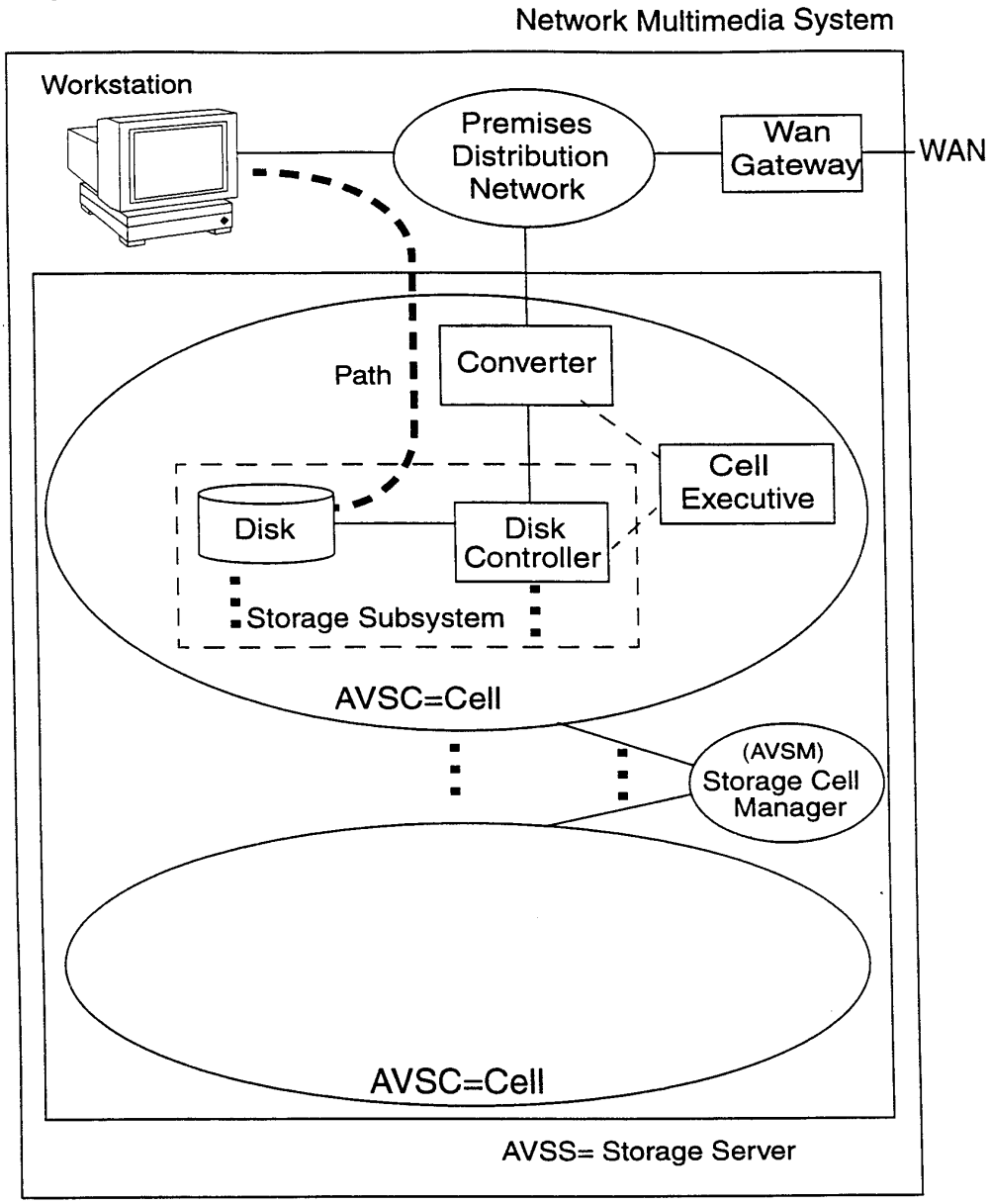
LINKPLAY EXHIBIT 1004
Page 3234 of 4822

Figure 29



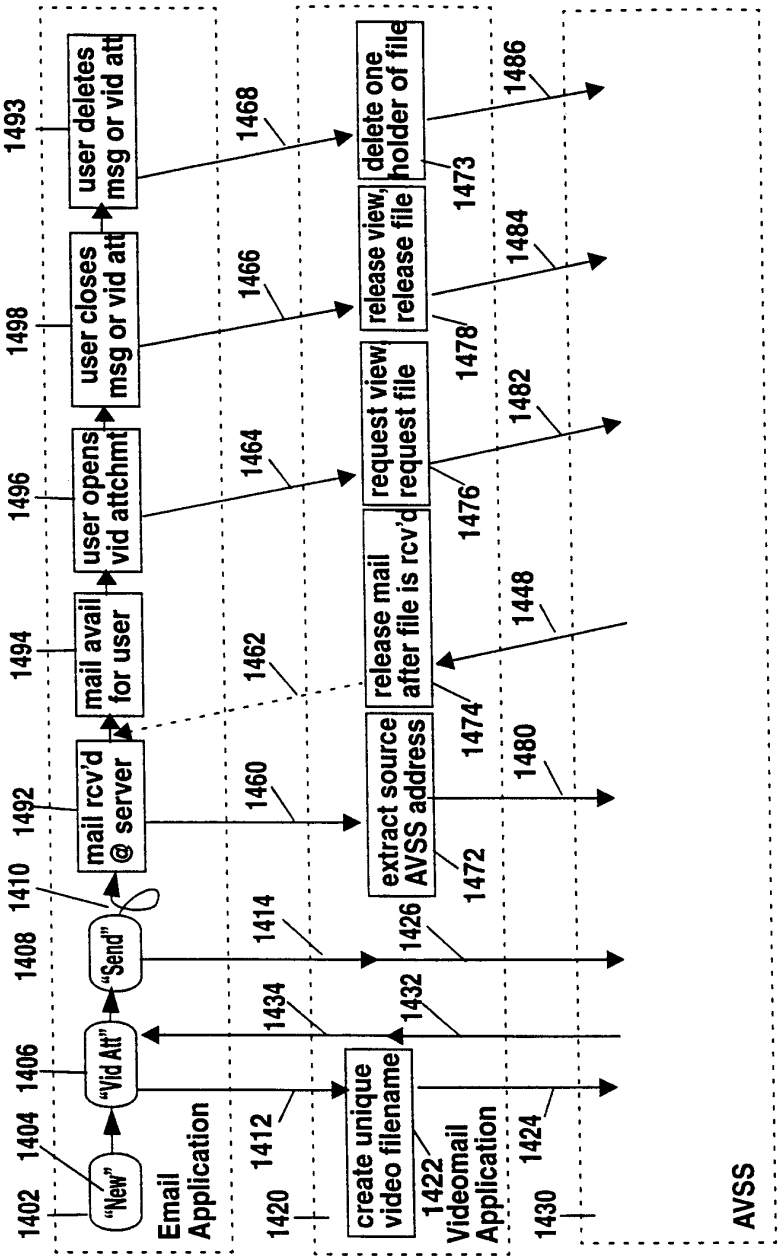
LINKPLAY EXHIBIT 1004
Page 3235 of 4822

Figure 30



LINKPLAY EXHIBIT 1004
Page 3236 of 4822

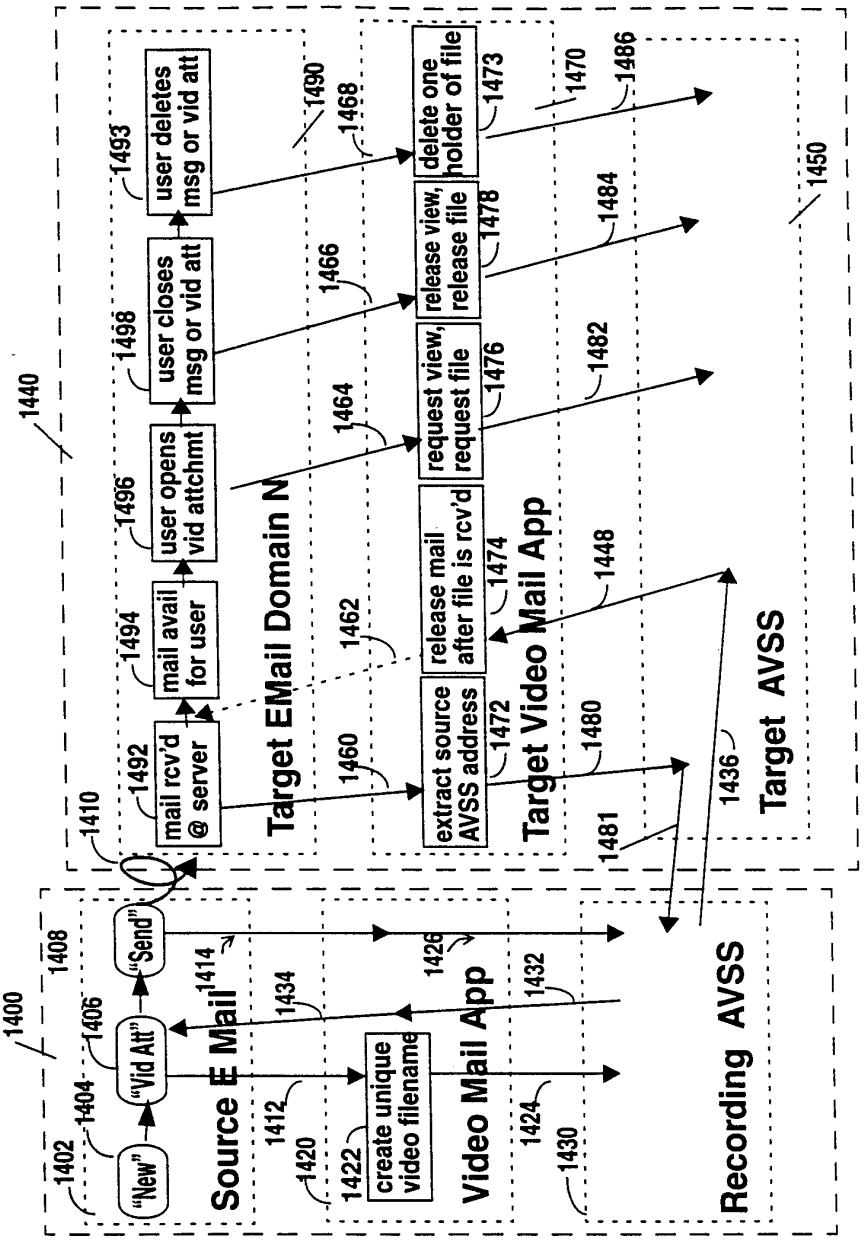
Figure 31



31/46

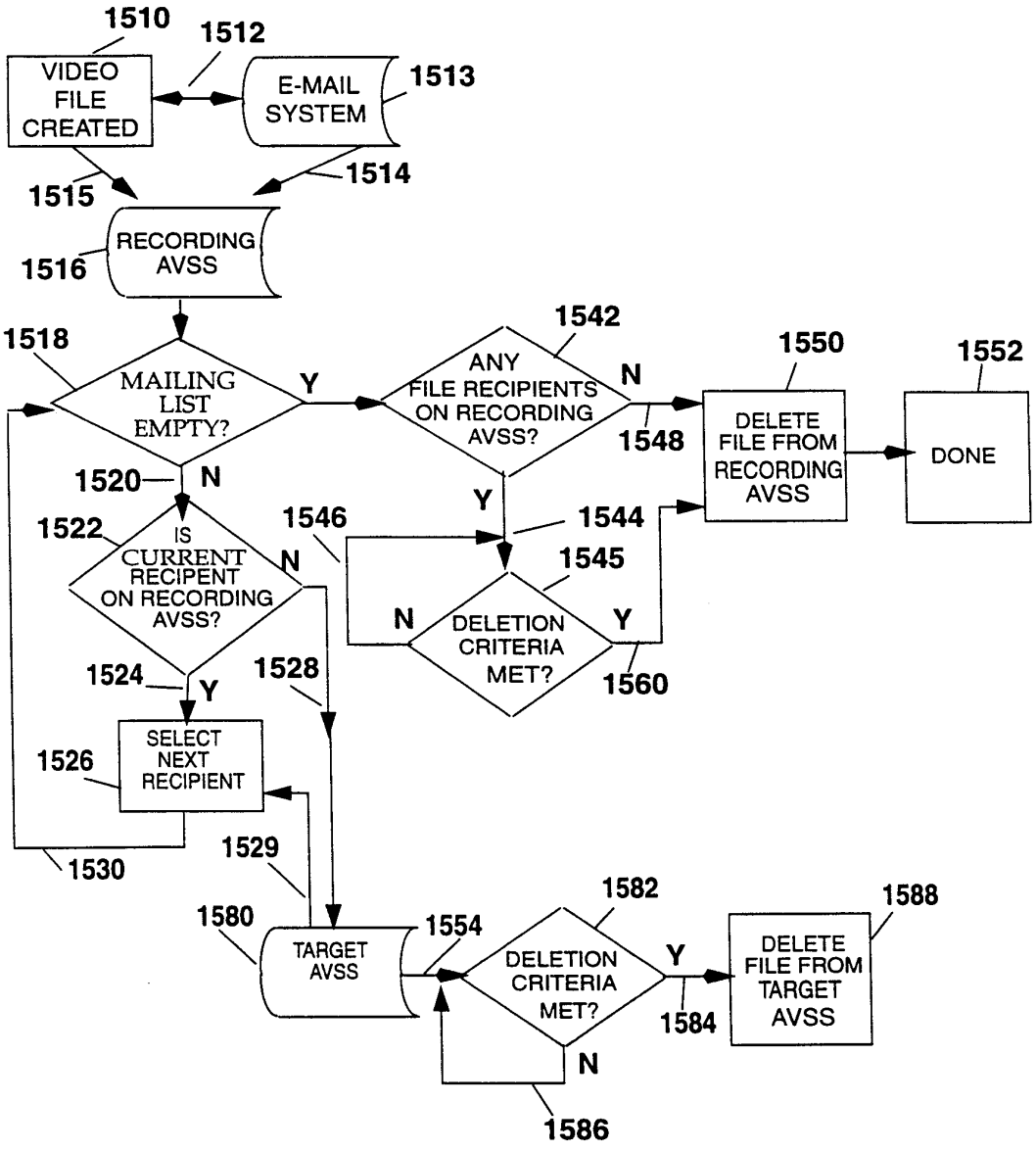
LINKPLAY EXHIBIT 1004
Page 3237 of 4822

Figure 32



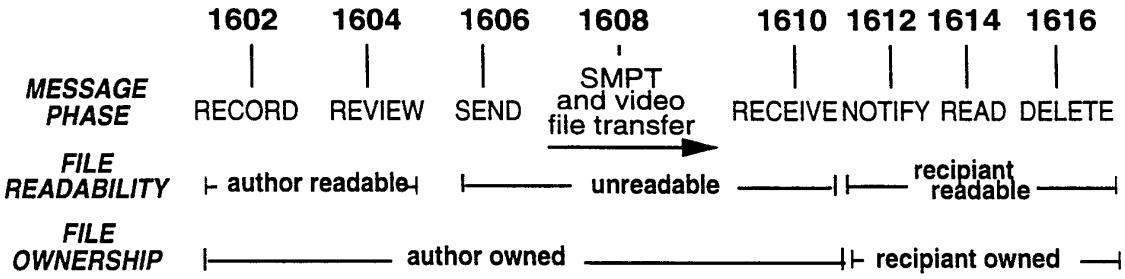
32/46

Figure 33



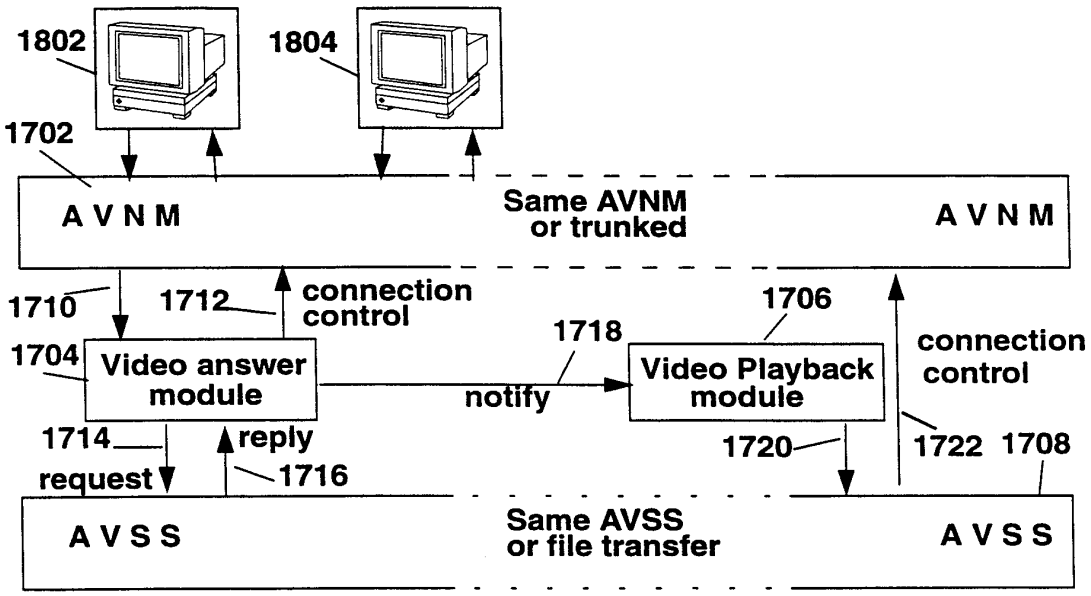
33/46

Figure 34



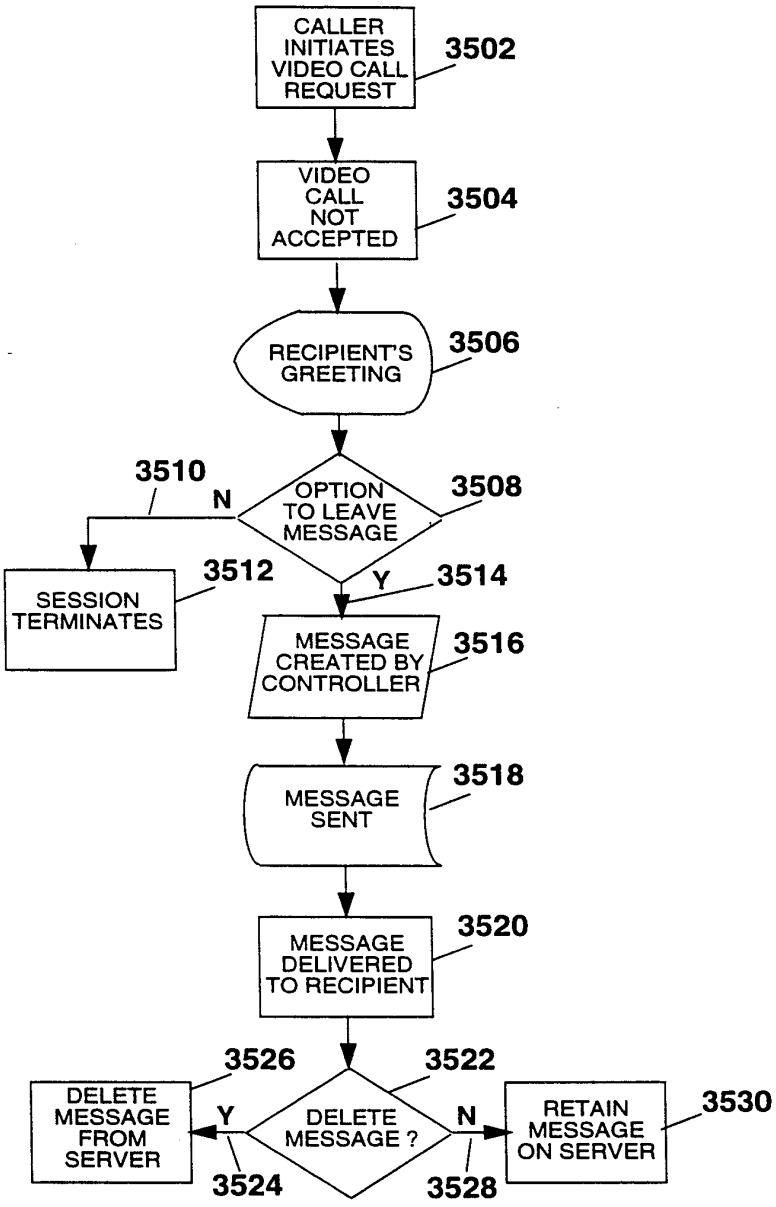
LINKPLAY EXHIBIT 1004
Page 3240 of 4822

Figure 35



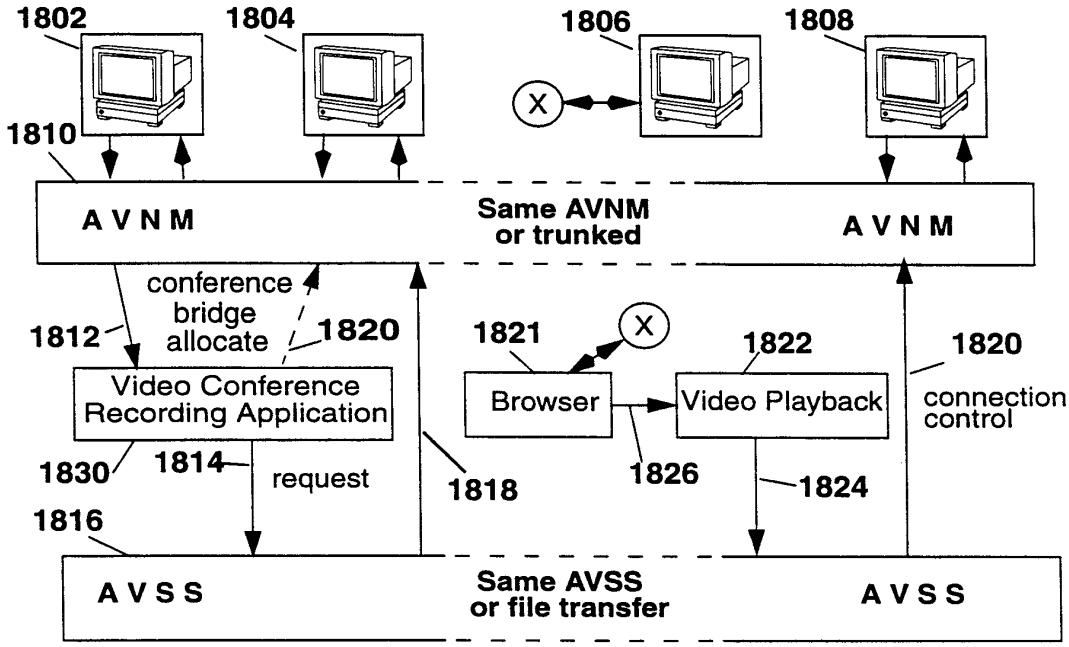
LINKPLAY EXHIBIT 1004
Page 3241 of 4822

Figure 36



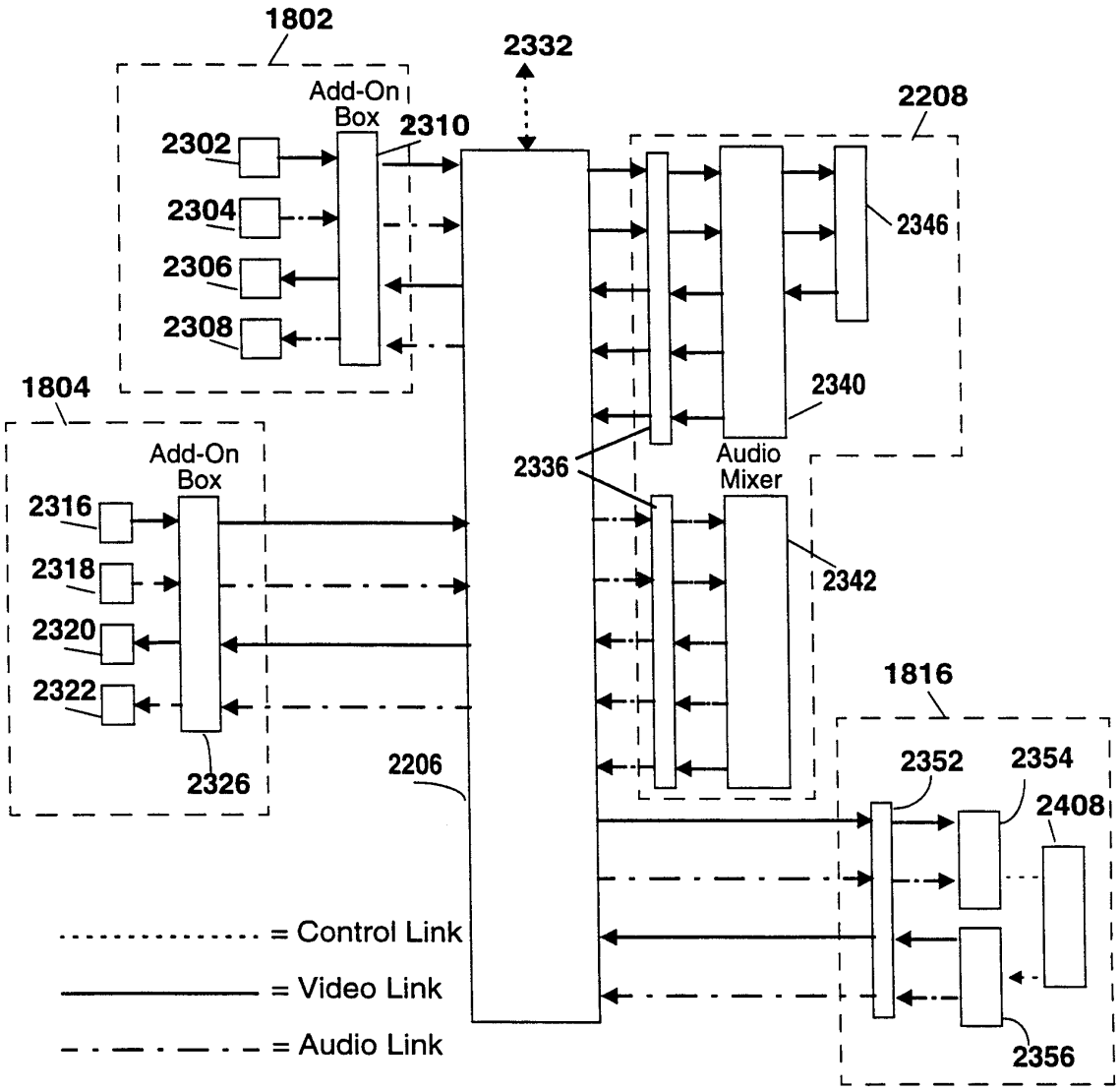
36/46

Figure 37



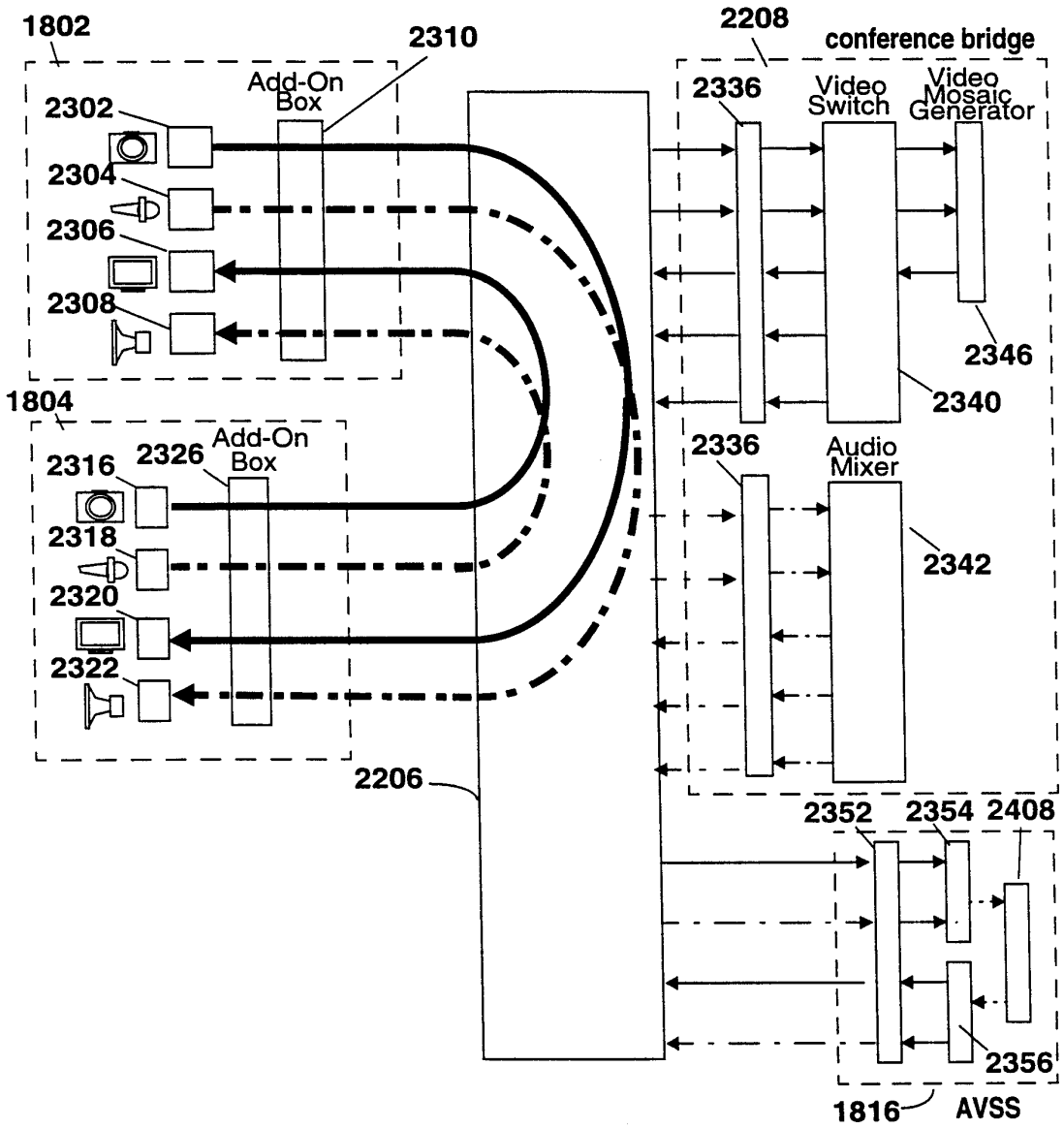
LINKPLAY EXHIBIT 1004
Page 3243 of 4822

Figure 38



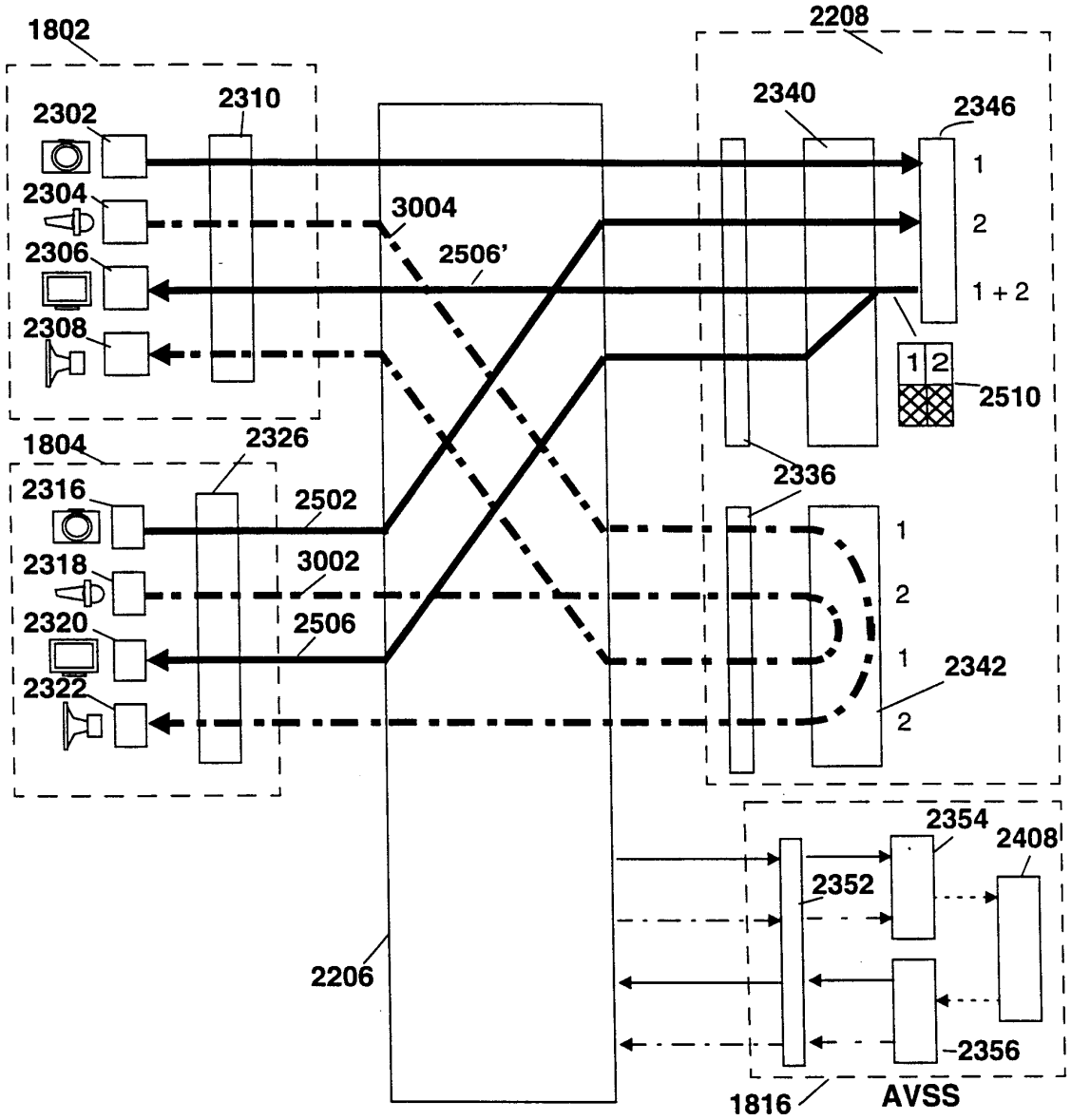
38/46

Figure 39



LINKPLAY EXHIBIT 1004
Page 3245 of 4822

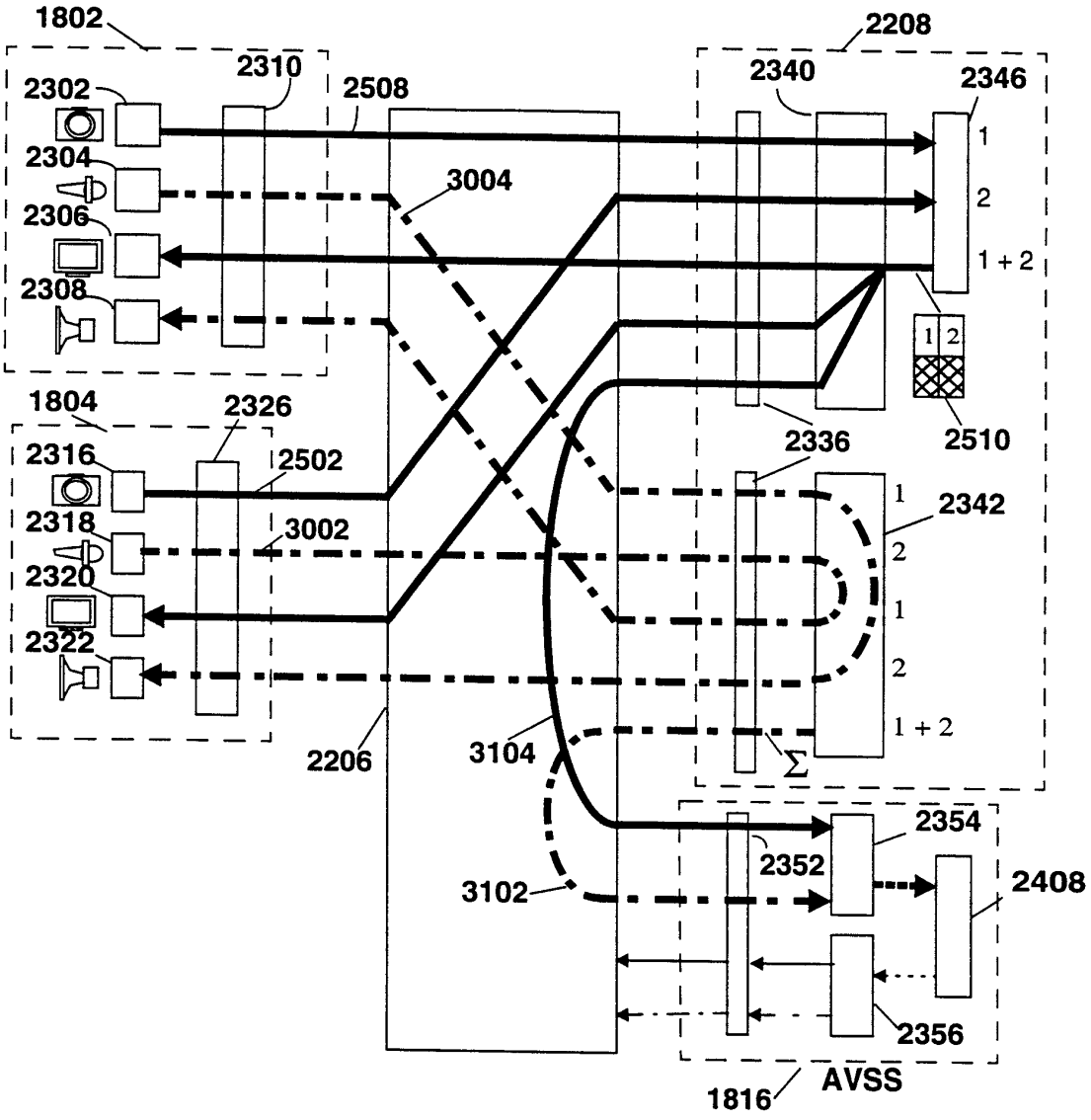
Figure 40



40/46

LINKPLAY EXHIBIT 1004
Page 3246 of 4822

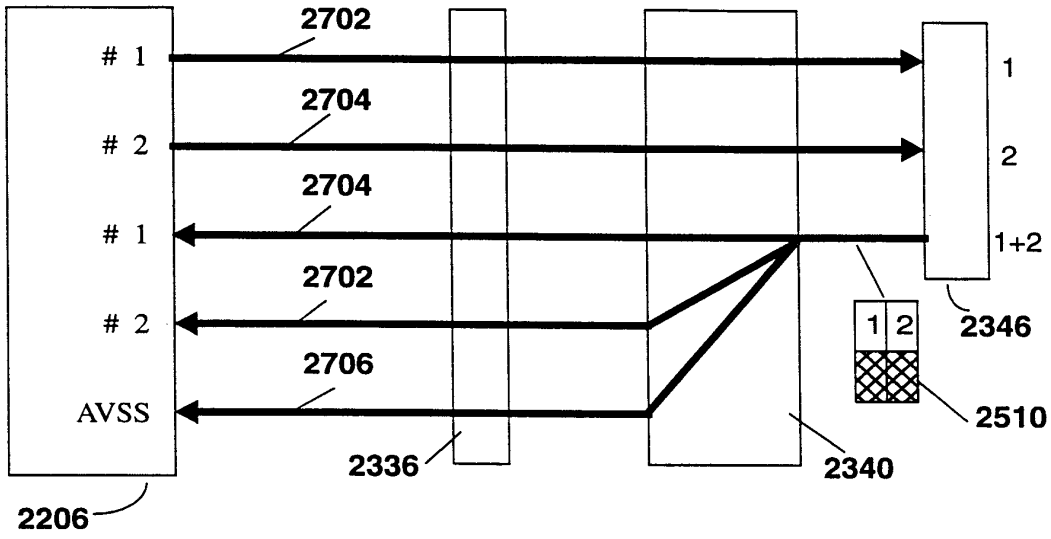
Figure 41



41/46

LINKPLAY EXHIBIT 1004
Page 3247 of 4822

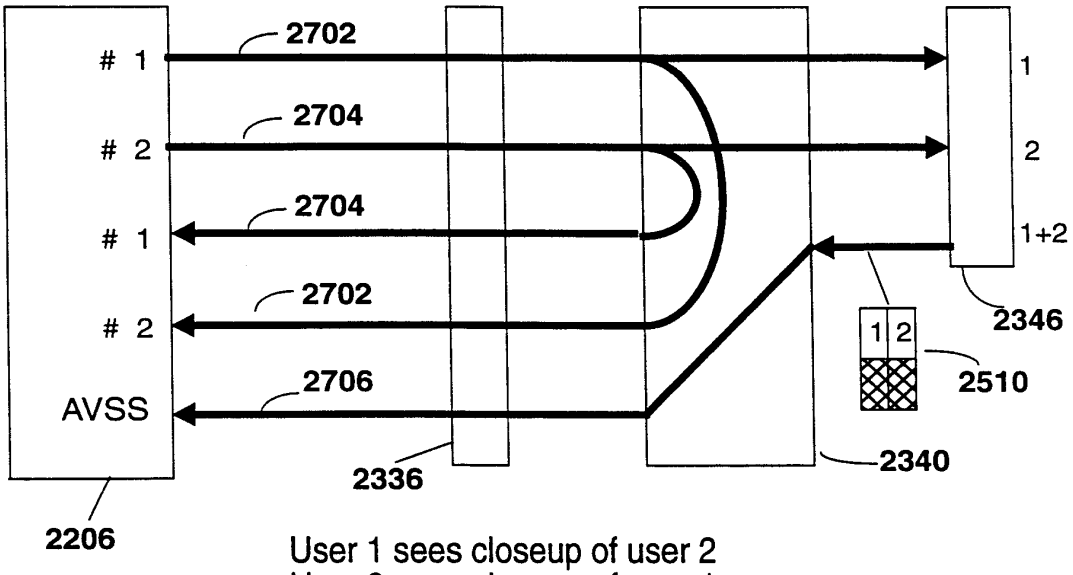
Figure 42



User 1 sees mosaic of users 1 and 2
User 2 sees mosaic of users 1 and 2
AVSS receives mosaic of users 1 and 2

LINKPLAY EXHIBIT 1004
Page 3248 of 4822

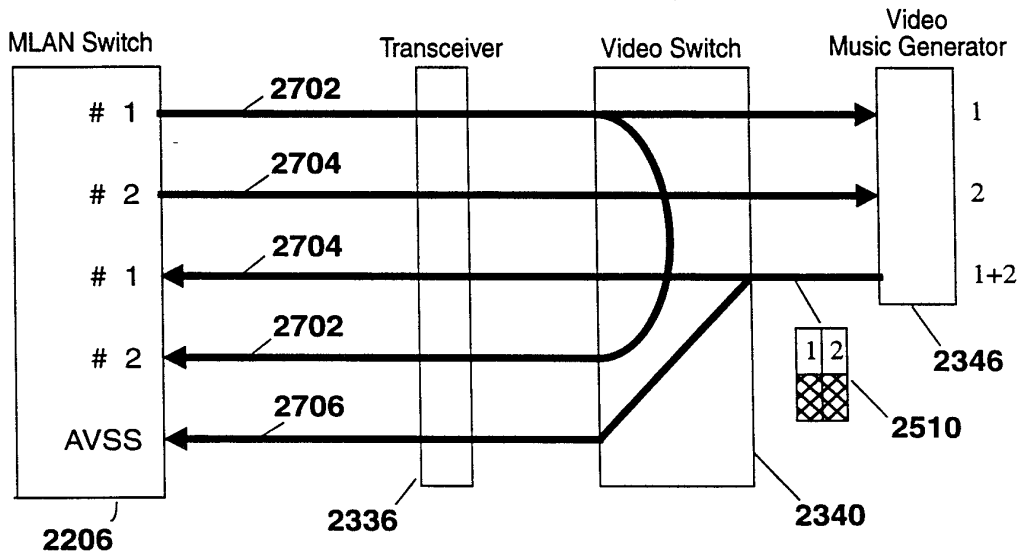
Figure 43



User 1 sees closeup of user 2
User 2 sees closeup of user 1
AVSS receives mosaic of users 1 and 2

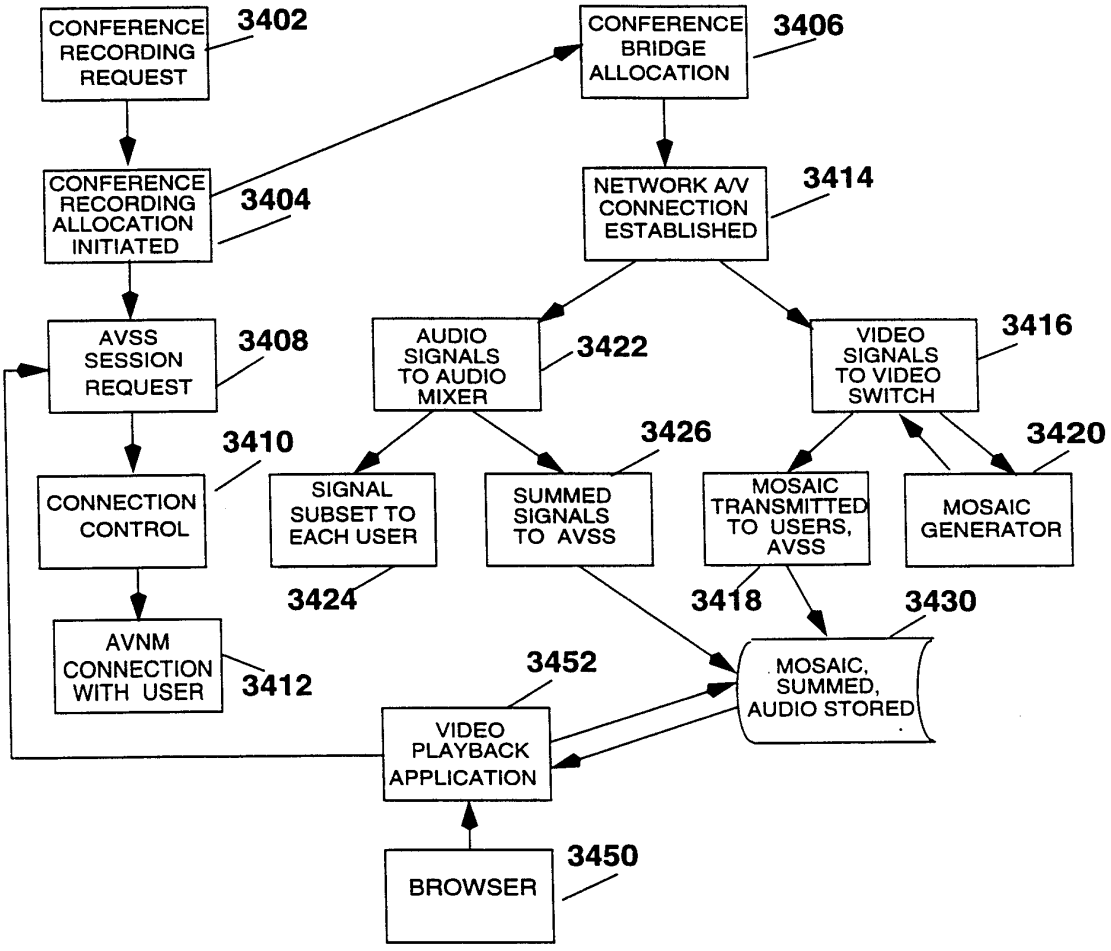
LINKPLAY EXHIBIT 1004
Page 3249 of 4822

Figure 44



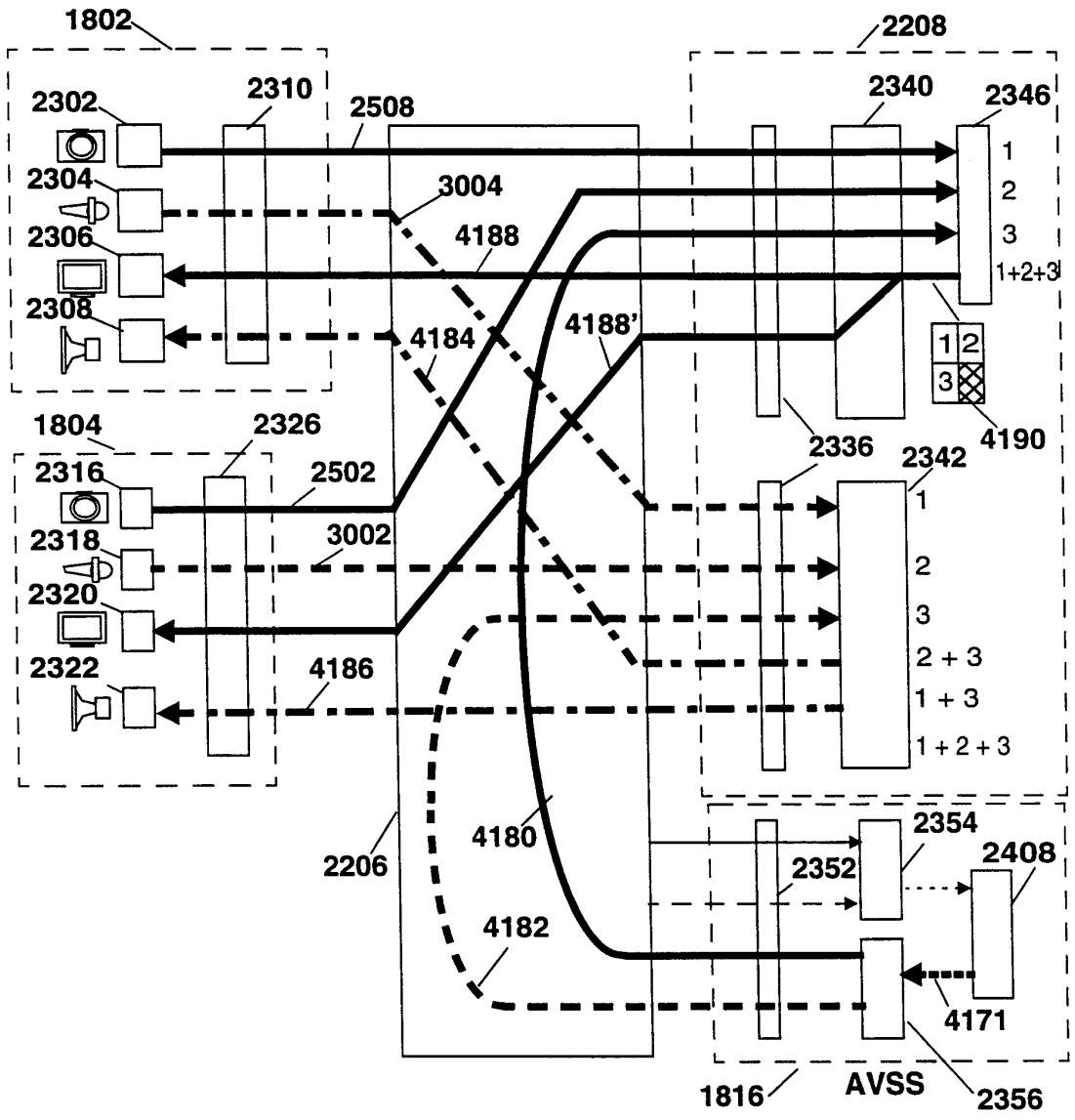
User 1 sees mosaic of users 1 and 2
User 2 sees closeup of user 1
AVSS receives mosaic of users 1 and 2

Figure 45



LINKPLAY EXHIBIT 1004
Page 3251 of 4822

Figure 46



46/46

LINKPLAY EXHIBIT 1004
Page 3252 of 4822

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/23596

A. CLASSIFICATION OF SUBJECT MATTER
IPC(6) : G06F 9/46; H04L 12/28, 12/56, 12/40
US CL : 370/432, 440; 395/200.02, 200.49, 800
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 370/432, 440; 395/200.02, 200.49, 800

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y,P	US 5,740,176 A (GUPTA et al) 14 April 1998, abstract.	1-75
Y	US 5,673,265 A (GUPTA et al) 30 September 1997, abstract	1-75
Y	US 5,617,539 A (LUDWIG et al) 01 April 1997, abstract	1-75
Y	US 5,574,934 A (MIRASHRAFT et al) 12 November 1996, see entire document	1-75
A	US 5712976 A (FALCON, JR et al) 27 January 1998 , abstract	1-75

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 03 FEBRUARY 1999	Date of mailing of the international search report 18 MAR 1999
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 308-5357	Authorized officer HIEU C. LE Telephone No. (703) 306-3101

Form PCT/ISA/210 (second sheet)(July 1992) ★

Electronic Acknowledgement Receipt	
EFS ID:	30217231
Application Number:	90013959
International Application Number:	
Confirmation Number:	9548
Title of Invention:	OBTAINING CONTENT FROM REMOTE SOURCE FOR PLAYBACK
First Named Inventor/Applicant Name:	9213357
Customer Number:	107361
Filer:	John Gerrard Tolomei/Jeffrey P. Armstrong
Filer Authorized By:	John Gerrard Tolomei
Attorney Docket Number:	04-0401-REEXAM (14-1800-R
Receipt Date:	31-AUG-2017
Filing Date:	16-JUN-2017
Time Stamp:	18:35:21
Application Type:	Reexam (Patent Owner)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Foreign Reference	CA2320451A1.PDF	2348016 886d48985fac09d9e164460844f6ffb62cbae8d3	no	61
Warnings:					

Information:					
2	Foreign Reference	CN101292500A.PDF	1601396 6f657a01929ebdf1dc7a0c4073a4c485b75e835a	no	27
Warnings:					
Information:					
3	Foreign Reference	CN1598767A.PDF	20019126 3d9b02d6e3a548c43ff97598ffd71e5975399d48	no	18
Warnings:					
Information:					
4	Foreign Reference	EP0251584A2_EFS.PDF	1875628 562f34c374299cd2d58cde1b3a7742c11b64cfdc	no	33
Warnings:					
Information:					
5	Foreign Reference	EP0672985A1_EFS.PDF	1719412 a5d39219e710078ac354114bd60bec9719392bc8	no	31
Warnings:					
Information:					
6	Foreign Reference	EP0742674B1.PDF	5626511 44cbbf2e5b4b7766e25423a7c4c4272ce46db063	no	107
Warnings:					
Information:					
7	Foreign Reference	EP0772374A2.PDF	350320 d0540f032eddeb881bde140e8cffe91866034bd3	no	6
Warnings:					
Information:					
8	Foreign Reference	EP0895427A2.PDF	1317370 9d64634828f09470c11e866285a88b57b4466126	no	20
Warnings:					
Information:					

9	Foreign Reference	EP1111527A2_EFS.PDF	1892700 8d496606f1392fd1baacd48b1e92c6f1c98ffc90	no	33
Warnings:					
Information:					
10	Foreign Reference	EP1122931A2_EFS.PDF	698059 295927d9d246e375ba624f211e9c49a919eab5a9	no	14
Warnings:					
Information:					
11	Foreign Reference	EP1312188B1.PDF	1175411 76ee008b5907a0b079de8b9791646286 2bf68759	no	22
Warnings:					
Information:					
12	Foreign Reference	EP1389853A1.PDF	1133104 5b5f94b59082db3faa5f61e6a03a2576c8a47399	no	21
Warnings:					
Information:					
13	Foreign Reference	EP1410686.PDF	1218656 0186846328be480cb409b6ba6d441972ef64c66f	no	17
Warnings:					
Information:					
14	Foreign Reference	EP1416687B1.PDF	2226988 615f24870bcdb2a5fdf9aced55c38ec73addе684	no	39
Warnings:					
Information:					
15	Foreign Reference	EP1517464-V2.PDF	14927003 46e64310c4749690168bd3e1b8de42d30326d154	no	37
Warnings:					
Information:					

16	Foreign Reference	EP2043381A2.PDF	500649 cfc9c3ebd465cd227af4d42f9c2ae879e4bd6622	no	11
Warnings:					
Information:					
17	Foreign Reference	EP2161950A2.PDF	438198 2055aa8a99d8f8a574f4bbd38868c6446f7d6b3d	no	10
Warnings:					
Information:					
18	Foreign Reference	EP2591617B1.PDF	1046822 af13974cacabcc4384fbf2614f669a4c46855494	no	17
Warnings:					
Information:					
19	Foreign Reference	EP2713281A1_EFS.PDF	6287993 e0bad8f9ccdf164cd3b1e5c4be7514c8122a5421	no	84
Warnings:					
Information:					
20	Foreign Reference	GB2284327A_EFS.PDF	1407480 8994fe8680cdb304fb1ce712869ba65bb6bd97e4	no	44
Warnings:					
Information:					
21	Foreign Reference	GB2338374_EFS.PDF	570090 3f631e460ac8630289b34fec71d695ffa1250fe1	no	16
Warnings:					
Information:					
22	Foreign Reference	GB2379533A.PDF	1396801 a26439e2db6f6c183a7fb12c7e8b2074853a297b	no	40
Warnings:					
Information:					

23	Foreign Reference	GB2486183A.PDF	680286 e007f2f1707f277d675a1db46b358fa7a14aa0af	no	19
Warnings:					
Information:					
24	Foreign Reference	JP07210129_EFS.PDF	1047465 c0f7b2f5445519535515cb21b06d7dca1b83a6b7	no	13
Warnings:					
Information:					
25	Foreign Reference	JP2000149391.PDF	502149 26b08e237f101717605ce2a98520fa4c7b63d0fb	no	10
Warnings:					
Information:					
26	Foreign Reference	JP2001034951.PDF	1088543 8cd2aeed4220f83c04a42ec8ac7f7540347d3100	no	22
Warnings:					
Information:					
27	Foreign Reference	JP2002-111817_EFS.PDF	1257794 3d40587cd556374f08d7a24dce1cd5341f42768c	no	23
Warnings:					
Information:					
28	Foreign Reference	JP2002123267A.PDF	1142650 d973c970470e2f10394d4a717e65dc8a5bbbcd31	no	20
Warnings:					
Information:					
29	Foreign Reference	JP2002358241A.PDF	12952206 9d0f58947bcbca587920aee8f529848415f85a18	no	10
Warnings:					
Information:					

30	Foreign Reference	JP2003037585_EFS.PDF	1725819 96f567820ee48e199f4071ab50ede187e7c6b1be	no	25
Warnings:					
Information:					
31	Foreign Reference	JP2003101958_EFS.PDF	950993 17257ed4f8a187cb6546ab4fe66fa716732fb265	no	12
Warnings:					
Information:					
32	Foreign Reference	JP2003169089.PDF	1819191 9d9f707f6897394efd823384531ccbee8c02e9ff	no	22
Warnings:					
Information:					
33	Foreign Reference	JP2003506765.PDF	2102143 28262bc3a8cb4c09b8b12c5eb5a6a3609d9297b6	no	46
Warnings:					
Information:					
34	Foreign Reference	JP2005108427_EFS.PDF	1075788 d14308cea9e61af4294aa93c28d700e7116a11a2	no	22
Warnings:					
Information:					
35	Foreign Reference	JP2005-136457_EFS.PDF	500771 6bd05fde776373b1e0285d145896f1c124e01ac4	no	13
Warnings:					
Information:					
36	Foreign Reference	JP2007241652A.PDF	757694 b11416f3d9a7b4362836f219a73539d2234a7da8	no	19
Warnings:					
Information:					

37	Foreign Reference	JP2009075540.PDF	9109282 c2051b858c053ba138ed00fd9df37a4fcd801a20	no	202
Warnings:					
Information:					
38	Foreign Reference	JP2009-135750_EFS.PDF	898894 56c3b9f4768ca88451df9e194ed0014a034bc1dd	no	18
Warnings:					
Information:					
39	Foreign Reference	JP2009506603.PDF	1735260 c271278bf877a4edd6299ffa3ca942efd1b82069	no	26
Warnings:					
Information:					
40	Foreign Reference	JP2009535708.PDF	1158271 a225780e8ab7472c0ae89ca2c32fd69881f5cea9	no	26
Warnings:					
Information:					
41	Foreign Reference	JP2009538006A.PDF	780949 58eb6213f43d027560b2f620c1b0c79f46fff66c	no	32
Warnings:					
Information:					
42	Foreign Reference	JP2011-130496_EFS.PDF	558338 0e00194241fac1e7fd7510c43a4b4c3704790df9	no	12
Warnings:					
Information:					
43	Foreign Reference	JP63269633.PDF	873462 b98faac4f170b698f2d534c3fc18503e58233d80	no	7
Warnings:					
Information:					

44	Foreign Reference	TW439027.PDF	924565 5941c824259d4ff874f61a57547d58c5dbb16dad	no	21
Warnings:					
Information:					
45	Foreign Reference	WO2000019693.PDF	1693361 46a45ae609ad520061da42ce819a15f7600ee6d1	no	38
Warnings:					
Information:					
46	Foreign Reference	WO0110125A1.PDF	1576645 b65f9570bb587ec501b672b075480c4fd45448b5	no	38
Warnings:					
Information:					
47	Foreign Reference	WO02073851A1_EFS.PDF	917956 d2d3113bf7994616b0d3066b9e4ffefb65109070	no	26
Warnings:					
Information:					
48	Foreign Reference	WO03093950A2_EFS.PDF	7403788 45672595a62d0867cd00ade157f38623a4f6a9e8	no	147
Warnings:					
Information:					
49	Foreign Reference	WO9525313A1_EFS.PDF	768644 6c073cc6a1ae2db78fa5e45e3af71a8b8121cdd3	no	25
Warnings:					
Information:					
50	Foreign Reference	WO1999023560.PDF	8501071 052d2aeca1b4d36613c68b08b4b83fcbdbcfaa46	no	208
Warnings:					
Information:					
		Total Files Size (in bytes):	134281711		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Electronic Acknowledgement Receipt	
EFS ID:	30230889
Application Number:	90013959
International Application Number:	
Confirmation Number:	9548
Title of Invention:	OBTAINING CONTENT FROM REMOTE SOURCE FOR PLAYBACK
First Named Inventor/Applicant Name:	9213357
Customer Number:	107361
Filer:	John Gerrard Tolomei/Jeffrey P. Armstrong
Filer Authorized By:	John Gerrard Tolomei
Attorney Docket Number:	04-0401-REEXAM (14-1800-R
Receipt Date:	31-AUG-2017
Filing Date:	16-JUN-2017
Time Stamp:	18:38:52
Application Type:	Reexam (Patent Owner)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Non Patent Literature	INTEGRA_AUDIO_NETWORK_SERVER_NAS2_3_INST_MANUAL.PDF	4573788 9ab726c4dd2095fb64df2a45f69048a5e584a6b2	no	32

Warnings:

Information:					
2	Non Patent Literature	INTELDESIGNING_SONDM0001 15117-62_46PAGES.PDF	21487917 3818acfd58a1844f3bdcf875b8fabe056bead49c	no	46
Warnings:					
Information:					
3	Non Patent Literature	INTELMEDIARENDERERDEVICE_ 2002_62PAGES.PDF	230873 93af56f7dc27e60f1dc2bd2053b9b4b32d02c4f7	no	62
Warnings:					
Information:					
4	Non Patent Literature	INTEL_SDK_UPNP_PROGRAMM ING_GUIDE_2002_30PAGES. PDF	287348 9f944b5fb8ff224578ed477d527f670591bcb16e	no	30
Warnings:					
Information:					
5	Non Patent Literature	PCTUS2013046372_IPRP_01-08 -2015.PDF	240252 933fa91104ab0e5235038bfe50f0eb26bf3364a9	no	6
Warnings:					
Information:					
6	Non Patent Literature	PCTUS2013046386_INTL_PRELI MINARY_REPORT_1-8-15.PDF	308278 d483d48dbb01747b5d9fb4710b31c32aadf7ba53	no	8
Warnings:					
Information:					
7	Non Patent Literature	PCTIB2012045894_IPRP_01_30 _2014.PDF	203266 7f11126a42ea767ff6a1811b6b35feb0afa162ef	no	6
Warnings:					
Information:					
8	Non Patent Literature	PCTUS2004023102_ISR_08-01- 2008_5PAGES-V2.PDF	175067 21658fe9b3b1e5a3e3ab0b8e852b40eec0b0693b	no	5
Warnings:					
Information:					

9	Non Patent Literature	PCTUS2013046372_ISR_8-26-13_3pages_V2.PDF	128104 0769238d2762e920b9ee249390eda2ad4b808e63	no	3
Warnings:					
Information:					
10	Non Patent Literature	PCTUS2012045894_ISR_12-26-12.PDF	94148 1b81a8f19df487d09e193d7b87d8badc0f7f6a52	no	3
Warnings:					
Information:					
11	Non Patent Literature	2013-09-30_PCT-US2013-046386_ISR_V2.PDF	128182 c6f72734ee8a1a36ff8e4c9a1ba8a2c1bea74ef0	no	3
Warnings:					
Information:					
12	Non Patent Literature	PCTUS2013046372_WRITTEN_OPINION_8-26-13_4PAGES-V2.PDF	165836 f2b5bb1025b1d673c414b2acaf395d35cd76466c	no	4
Warnings:					
Information:					
13	Non Patent Literature	PCTUS2012045894_WO_12-26-12.PDF	99988 b7a29f654e3f813d6d525b45740c4010bb46c87e	no	4
Warnings:					
Information:					
14	Non Patent Literature	2013-09_30-PCT-US2013-046386-WRITTEN_OPINION_V2.PDF	232176 7587dabb5ed76d0c2f5bf6c754d4b4813788a6c1	no	6
Warnings:					
Information:					
15	Non Patent Literature	ISHIBASHI_A_COMPARISON_OF_MEDIA_2001_8PAGES-V2.PDF	1894249 ccee33b847a24fde66fd399f584014692666a871	no	8
Warnings:					
Information:					

16	Non Patent Literature	ISHIBASHI_IEEEGLOBALTELECOMMCONF_1997_2_746.PDF	825931 bdd8f8aacd41f15aa5026386084f8f648a4c296b	no	7
Warnings:					
Information:					
17	Non Patent Literature	ISHIBASHI_IEEEINFORMATIONREVCOMM_1997_2_692.PDF	796585 354fa7cf7b9d8a0bea4697452b433dcaec936fe1	no	9
Warnings:					
Information:					
18	Non Patent Literature	ISSUES_AVAGO0058-V2.PDF	1377564 793144f58091f94dbcf170c917620b5928427a3a	no	5
Warnings:					
Information:					
19	Non Patent Literature	JP2012-17811_DECISION_OF_REJECTION_07-08-2014-V2.PDF	82294 40ba3e50b4e68e0dbb29983c90d257088a31eee6	no	3
Warnings:					
Information:					
20	Non Patent Literature	JP2015520286_FOA_11-08-2016.PDF	179629 f2bead36683aa94386ef45cd26d365a00dfcadfc	no	5
Warnings:					
Information:					
21	Non Patent Literature	JAPANESE_NO_2014-521648_REJECTION_NOTICE_2015-02-03-V2.PDF	316632 6a0630bc1eb25d14c002919312a755dde9c91a17	no	7
Warnings:					
Information:					
22	Non Patent Literature	JP2014-220704_NOTICEOFREJECTION_9-15-15.PDF	155605 4c4284e08a78c78c60e19290456dcb307dc22059	no	7
Warnings:					
Information:					

23	Non Patent Literature	JAPANESE_NO_2015-520288_OFFICE_ACTION_11-22-16.PDF	255674 d01c8753f526ab1cb1abb20c843002598e00116b	no	6
Warnings:					
Information:					
24	Non Patent Literature	JAPANESE_NO_2014-220704_OFFICE_ACTION_5-24-16.PDF	205657 ac13737727671ca89f2dba0cbc7ee7dd13ba7500	no	7
Warnings:					
Information:					
25	Non Patent Literature	JP2015520288_OA_03-29-2016.PDF	304956 5082659c564f8039e97018b78697d9b8277be286	no	12
Warnings:					
Information:					
26	Non Patent Literature	JAPANESE_NO_2015-516169_OFFICE_ACTION_11-29-16.PDF	113875 280ee9b72c5452c5197db65d34743ea88fa8ecde	no	4
Warnings:					
Information:					
27	Non Patent Literature	JAPANESE_NO_2015-520286_OFFICE_ACTION_2-2-16_V2.PDF	173880 9eea259a42f3e253969f04f31376eb78933907b0	no	6
Warnings:					
Information:					
28	Non Patent Literature	JP2012-17811_OFFICE_ACTION_SUMMARY_11-19-13-V2.PDF	145813 f8f6f1db643276c13b47333f5c39d8169b43913d	no	5
Warnings:					
Information:					
29	Non Patent Literature	JO_PROCEEDINGSSPIE_2002_4861_71.PDF	623348 7456c250e9a0d9d14bdab77ee04ae019146d379f	no	11
Warnings:					
Information:					

30	Non Patent Literature	JONES_2000.PDF	2767920 6ae528491dacb165a4f1fea5caedb649a2843c7d	no	2
Warnings:					
Information:					
31	Non Patent Literature	KOU_RENDERINGCONTROL_JUNE_25_2002.PDF	6537880 8a0b7039632d8bb1148c28b89d8224415cd68db7	no	63
Warnings:					
Information:					
32	Non Patent Literature	LAKE_SERIES_DIGITAL_2011.PDF	4370645 ebc9f43f81c5da0201e69e37a86799d91bb6d594	no	71
Warnings:					
Information:					
33	Non Patent Literature	LEVERGOOD_1993-V2.PDF	11929228 8e9763c73298c3a088449113b0993fd96a7e61b0	no	109
Warnings:					
Information:					
34	Non Patent Literature	LG_RJP-201M_REMOTE_2010.PDF	20283650 eca9aa83452c739668fc8df7c3d64dd5aaf5ac8f	no	24
Warnings:					
Information:					
35	Non Patent Literature	LIENHART_2002.PDF	210917 6d3471bc568caeea931144e41573f34e43a38e74	no	1
Warnings:					
Information:					
36	Non Patent Literature	LINKSYS_DMRW1000_DATASHEET_2PAGES.PDF	184015 c726064357019eaf402cbcbf09c2b66f4c1f829f	no	2
Warnings:					
Information:					

37	Non Patent Literature	LINKSYS_DMRW1000_USERGUIDE_64PAGES.PDF	9044320 d1e893540b3ffe5baa8b3f6092c9d6142b504c73	no	64
Warnings:					
Information:					
38	Non Patent Literature	LINKSYS_DMP100_QUICK_INSTALLATION_GUIDE_32PAGES.PDF	3385349 4835be784438429e1012fa331258a424ee5eb8a4	no	32
Warnings:					
Information:					
39	Non Patent Literature	LINKSYS_DMP100_USERGUIDE_65PAGES.PDF	8612733 4a53cbef176372759de60c8372502b098f9b8dff	no	65
Warnings:					
Information:					
40	Non Patent Literature	LINUX_SDK_FOR_UPNP_2002_101PAGES.PDF	330243 4b9250fccdc7f98de95e3ab422d6b6b393b99145	no	101
Warnings:					
Information:					
41	Non Patent Literature	LINUX_SDK_2003_102PP.PDF	6580264 c2b61fc0541f45fa70dcc3a572c54b876e930e38	no	102
Warnings:					
Information:					
42	Non Patent Literature	LIU_ETAL_A_SYNCHRONIZATION.PDF	14664014 05bfb6e6ebcecfb9188988a5e2de2b2c3e6fd4dc	no	10
Warnings:					
Information:					
43	Non Patent Literature	LIU_ETAL_ADAPTIVE.PDF	3753450 41e262c0ef1ab95b2b0d3858dd003cba164212e3	no	4
Warnings:					
Information:					

44	Non Patent Literature	LOUDERBACK_2000.PDF	82505 401ceed7630aa3dc4edcf829b55770d9f673c17f	no	2
Warnings:					
Information:					
45	Non Patent Literature	MANIACTOOLS_2010.PDF	287375 61babf16b81c514a1120bfd0913483a1794523f8	no	2
Warnings:					
Information:					
46	Non Patent Literature	MEDIARENDERER_1_2002_12PAGES.PDF	314359 1c7959498a37654aae2ed1b35a5e50d8542cbe9f	no	12
Warnings:					
Information:					
47	Non Patent Literature	MEDIASERVER_2002_12PAGES.PDF	325151 ed2a3d0adb5d617bf15be8b107dd791ecfc6a266	no	12
Warnings:					
Information:					
48	Non Patent Literature	MICROSOFT_UNIVERSAL_PLUG_AND_PLAY_DM_0402007-24_18PAGES.PDF	8990496 19c9666824691f55156ab318725527384d0d79cf	no	18
Warnings:					
Information:					
49	Non Patent Literature	MICROSOFT_WINDOWS_XP_REVIEWERS_GUIDE_DM_0402225-85_61PAGES_PART1.PDF	18506638 dd4b0688f6be457f3109865418be53d77684650d	no	10
Warnings:					
Information:					
50	Non Patent Literature	MICROSOFT_WINDOWS_XP_REVIEWERS_GUIDE_DM_0402225-85_61PAGES_PART2.PDF	24811494 d946404e54742cc85dfcab70f7f0c4d8227f1b94	no	10
Warnings:					
Information:					

51	Non Patent Literature	MICROSOFT_WINDOWS_XP_REVIEWERS_GUIDE_DM_0402225-85_61PAGES_PART3.PDF	20456310 eeb8bdb7eaeeddb5104825f80697c6d7380ab779	no	10
Warnings:					
Information:					
52	Non Patent Literature	MICROSOFT_WINDOWS_XP_REVIEWERS_GUIDE_DM_0402225-85_61PAGES_PART4.PDF	23455812 7f103bb5c53a7a1b0117035ea3c2f2baae3932fb	no	10
Warnings:					
Information:					
53	Non Patent Literature	MICROSOFT_WINDOWS_XP_REVIEWERS_GUIDE_DM_0402225-85_61PAGES_PART5.PDF	18816327 b7e09cf675982370f73b214306c787cb6ece271b	no	10
Warnings:					
Information:					
54	Non Patent Literature	MICROSOFT_WINDOWS_XP_REVIEWERS_GUIDE_DM_0402225-85_61PAGES_PART6.PDF	16028477 088845aa2e1533f1abf4927423ecb2588d6abd98	no	11
Warnings:					
Information:					
		Total Files Size (in bytes):	260536487		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.
<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.
<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 9/46	A1	(11) International Publication Number: WO 99/61985
		(43) International Publication Date: 2 December 1999 (02.12.99)

(21) International Application Number: PCT/US99/06634

(22) International Filing Date: 26 March 1999 (26.03.99)

(30) Priority Data:
60/086,737 26 May 1998 (26.05.98) US

(71) Applicant: THE DOW CHEMICAL COMPANY [US/US]; 2030 Dow Center, Midland, MI 48674 (US).

(72) Inventors: WOODS, Randy, D.; 3788 Old Pine, Midland, MI 48642 (US). JACHIM, David, M.; Apartment L, 3063 Signature Boulevard, Ann Arbor, MI 48103 (US). DUPREE, Wayne, P.; 5301 Stony Creek, Midland, MI 48640 (US). VERNIERS, Gerrit, H.; Heirweg 88, B–9180 Moerbieke–Waas (BE). CHURCHILL, Stephen, G.; 2588 Waskevich, Midland, MI 48642 (US). FERNANDEZ, George, P.; 449 North 5 Mile, Midland, MI 48640 (US).

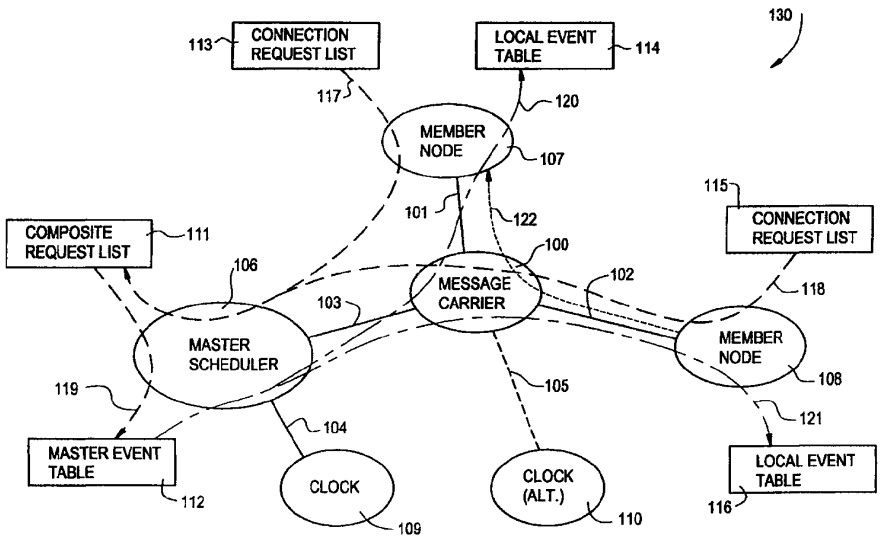
(74) Agent: SCHULTZ, Dale, H.; Patent Dept., P.O. Box 1967, Midland, MI 48641–1967 (US).

(81) Designated States: CA, CN, JP, MX, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: DISTRIBUTED COMPUTING ENVIRONMENT USING REAL–TIME SCHEDULING LOGIC AND TIME DETERMINISTIC ARCHITECTURE

(57) Abstract



A system is presented for a plurality of computer nodes in mutual data communication to execute logic in a coordinated manner. The approach features aligning the definition of the time in the nodes so that each node has essentially the same definition of time,defining a schedule of communications for each computer node, with the schedule having future moments defined for initiating each communication, communicating each schedule to its respective node, and initiating each communication at the time defined in the schedule. The computer network has a message carrier supporting multiple communication paths; a plurality of computers coupled to the message carrier, a clock system for maintaining time synchronism among the computers; and a scheduler coupled to the message carrier for determining the sequence of communication between the computers. The message carrier establishes communication between selected computers based on the time–deterministic tasks being performed by the computers. A computer node in the plurality of computer nodes executes logic in coordination with the other nodes by defining a communication request list of each communication needed by the computer node with any other computer node respective to a future time period, sending the communication request list to the master scheduler, receiving back a communication schedule back from the master scheduler, defining a local task schedule from the communication schedule, and then executing the communication at the scheduled time.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

DISTRIBUTED COMPUTING ENVIRONMENT USING REAL-TIME SCHEDULING LOGIC AND TIME DETERMINISTIC ARCHITECTURE

This invention is directed to the field of data processing control systems, particularly to integrated systems where a plurality of computers execute their logic in a coordinated manner within a network. The primary application of the system described in this specification is for computerized control of machines deployed on a large physical scale (for example a chemical manufacturing facility), although benefits to endeavors related to plural computer processors, parallel computer processors, and networked computer processors are also enabled.

When computational demands or geographical constraints require the physical distribution of computing facilities (for example such as for process control in a large, geographically dispersed chemical plant), implementation factors (such as product quality) may benefit from centralized coordination of the computing tasks. This invention describes a system and method for enabling a plurality of geographically dispersed computers to be networked for enabling time coordinated inter-activity. This resultant network enables task coordination and facilitates the implementation of other qualities beneficial to the task being performed by the system - qualities such as redundancy, security, and high throughput.

One embodiment of a system showing such substantive real-time networking to a set of physically distributed remote field units for use in a process control system is described the United States Patent 5,428,769 issued on Jun. 27, 1995 and entitled "Process control interface system having triply redundant remote field units" to Glaser, Robert S.; Hoy, Robert S.; Fernandez, G. Paul; and Grai, Timothy J. While the system is adequate for many process control situations, the networking methodology effectively addresses each system in each time period on the presumption that there is a need to do so; however, many of these addressing cycles are unnecessary if all variables for exchange are stable -- a more elegant scenario from the standpoint of network efficiency would provide for communications only when the status of either the inputs or the outputs require change.

Physical distribution of computing facilities combined with centralized coordination of the computing tasks is also a driving concept in high performance processors characterized by parallelism. Vector processors, as discussed in sections 7.1 and 7.2 ("Vector Processors") of Computer Architecture A Quantitative Approach by John L. Hennessy and David A.

LINKPLAY EXHIBIT 1004
Page 3274 of 4822

Patterson (Morgan Kaufmann; San Mateo, California; 1990; pages 351-363) are one manifestation of such systems, and processor-memory pair approaches such as the Cm* design discussed from pages 11 to 18 of "Parallel Processing - The Cm* Experience" by E. F. Gehringer, D. P. Siewiorek, and Z. Segall (Digital Press via Editorial Inc.; Rockport, Massachusetts; 1987) provide a second example of these types of approaches.

Centralized coordination of computing tasks in a scheduled context for the future also draws on concepts from traditional project scheduling such as is discussed from pages 569 to 584 of the text Productions and Operations Management by J. William Gavett (Harcort, Brace, and World; New York; 1968).

Process control systems development is characterized by an ever-increasing incorporation and management of issues at large in the system being controlled, requiring that a comparably larger amount of logic on a comparably larger amount of data be executed in a time frame which is not permitted to expand. This can be achieved to some degree by hardware which processes logic more quickly and holds more data in an addressed physical store, but there are substantial costs in pursuing this solution. What is needed to resolve the ability of future systems to effectively expand their comprehensive treatment of the control situation is to truly provide a solution path for enabling a very large number of distributed computers to execute massive amounts of logic in a mutual manner which is time-synchronous across an entire network. The present invention provides such a method and system.

The invention provides a computer implemented method for enabling a plurality of computer nodes in mutual data communication to execute logic in a coordinated manner by

(a) aligning the definition of time between the computer nodes so that each computer node has essentially the same "understanding" of time;

(b) defining a schedule of communications for each computer node, where the schedule has a different time in the future defined for initiating each respective communication;

(c) communicating each schedule to its respective computer node; and

LINKPLAY EXHIBIT 1004
Page 3275 of 4822

(d) initiating each communication from each node at the time defined in its schedule.

The described embodiment provides a computer network architecture, having a message carrier with a plurality of computers coupled to it; a clock system (accessible by all computers in the network) for maintaining time synchronism among the computers; and a scheduler for determining the sequence of communication between the computers by causing the message carrier to establish communication between selected computers based on time-deterministic tasks being performed by the computers.

The described embodiment further provides a global time system for aligning the sense of time in the set of computers to a globally defined reference.

The described embodiment further provides a computer implemented method for enabling a plurality of member nodes to execute logic in a coordinated manner.

The described embodiment further provides for concurrency in communications between computers via the message carrier.

The described embodiment further provides provides a computer implemented method for scheduling tasks for resource objects in a plurality of resource objects functioning as a collective group.

The described embodiment further provides a method by which different computers can align their definition of time by adjusting their oscillator's rate of oscillation.

The computer network architecture is enabled with a message carrier supporting multiple communication paths; a plurality of computers coupled to the message carrier, with each computer being programmed to perform time-deterministic tasks by cycling through a plurality of different operating states, including a communication state; a clock system accessible by all the computers for maintaining time synchronism among the computers; and a scheduler coupled to the message carrier for determining the sequence of communication between the computers by enabling the message carrier to establish communication

LINKPLAY EXHIBIT 1004
Page 3276 of 4822

between selected computers based on the time-deterministic tasks being performed by the computers.

A computer node (element for computing with an independent central processing unit) in the plurality of computer nodes can execute logic in coordination with the other nodes by being connected to a message carrier along with the other computer nodes where there is also a provision for (a) master scheduling capability in one computer node to establish a master scheduler; (b) defining, in each computer node, a communication request list of each communication needed by the computer node with any other computer node respective to a future time period; (c) sending of the communication request list from each computer node to the master scheduler via the message carrier; (d) receiving of, in each computer node, a communication schedule from the master scheduler via the message carrier; and (e) defining, in each computer node, a local task schedule from the communication schedule.

The above approach enables a plurality of member nodes in a system to execute a large amount of logic in a coordinated manner on a group of distributed platforms.

In performing the above, the step of defining a reference moment time value is performed in one embodiment by the master scheduler.

In performing the above, the step of defining a communication request list in each member node in one embodiment limits the communications to a maximum number in the future time period and each communication has an duration time attribute and the value of the duration time attribute is less than a fixed maximum value in order to achieve efficient operation.

Use of a global time system in one embodiment enables a universal understanding of time across a large number of different computer nodes (member nodes) in an extended system.

The method for scheduling involves a process of scheduling tasks for resource (member) objects in a plurality of resource objects functioning as a collective group; and this computer implemented process (called an "apparent critical resource" process) involves the steps of:

defining a set of future tasks for the collective group;

LINKPLAY EXHIBIT 1004
Page 3277 of 4822

defining each resource object needed for each future task in the set of future tasks;

defining each interaction needed between any two resource objects respective to the set of future tasks;

defining a use instance for each resource object in achieving each interaction where the use instance has an use duration attribute value;

defining an accessing resource object and an accessed resource object respective to each use instance;

determining a summed accessing time value for each accessing resource object as a sum of all use duration attribute values for all use instances respective to that resource object when it is the accessing resource object;

determining a summed accessed time value for each accessed resource object as a sum of all use duration attribute values for all interactions between the accessed resource object and the accessing resource object;

defining a set of sequential future time increments within a scheduling period in a schedule;

scheduling, into the earliest available time in the first future time increment, a future resource interaction event based on the first available future task in the set of future tasks between

(a) the accessing resource object having the highest summed accessing value respective to the set of future tasks and

(b) the accessed resource object having the highest summed accessed value respective to the set of future tasks (if a use instance exists between the accessed resource object and the accessing resource object -- the future resource interaction event in the schedule record has an initiation time equal to the earliest available time in the

LINKPLAY EXHIBIT 1004
Page 3278 of 4822

schedule, an accessing resource object identifier attribute respective to the accessing resource object, an accessed resource object identifier attribute respective to the accessed resource object, and the aforementioned use duration attribute value)

removing from scheduling candidacy the use instance respective to the accessed resource object in the scheduled future resource interaction event and the accessing member object identifier attribute in the scheduled future resource interaction event to redefine the set of future tasks to those future tasks for which interactions have not been scheduled;

repeating the steps of determining a summed accessed value, determining a summed accessing value, scheduling a future communication task, and removing all use instances respective to scheduled resource objects until all use instances have been scheduled which can be scheduled into the first future time increments; and

recurrently executing the steps of determining a summed accessed value, determining a summed accessing value, scheduling a future resource interaction event, removing all resource objects respective to scheduled resource objects, and repeating such steps until all resource interaction events have been scheduled which can be scheduled into each future time increment and until all resource interaction events have been scheduled into the scheduling period.

In one embodiment, the message carrier has a plurality of channels for permitting concurrency in the communication events, the step of scheduling being respective to the plurality of channels as a group wherein the earliest available time is respective to any channel in the message carrier.

In some embodiments, each member node has a variable speed oscillator for adjusting a local clock to align with a global clock on the network. Time alignment logic and oscillator rate adjustment logic for execution by the computer circuitry are also provided to achieve rapid time alignment of the particular member node (computer node).

LINKPLAY EXHIBIT 1004
Page 3279 of 4822

The message carrier used in the described embodiment is enabled with a network topology which can be manifested (in the preferred embodiment) with a crossbar switch, or, in an alternative embodiment, with a radio frequency communications system.

Additional features and advantages of the present invention will become more fully apparent from a reading of the detailed description of the preferred embodiment and the accompanying drawings.

Figure 1 presents a diagram of the data flows between the various elements in a distributed network.

Figure 2 depicts a distributed network of computing elements which are resident in physically different locations throughout the world. This figure includes a network of computing elements, a master scheduler, and a common source of time.

Figure 3 presents a diagram of a network in a bus structure.

Figure 4 shows a distributed computing environment where the network is implemented in a ring topology.

Figure 5 presents a diagram of a network in a switched star topology which is a preferred network architecture embodiment.

Figure 6 presents a diagram of data structures communicated between the computing elements and a master scheduler.

Figure 7 depicts a flowchart of the process implemented in the master scheduler in the scheduling of communications between two separate computing elements.

Figure 8 presents a simplified diagram of the scheduling logic (software in execution by electrical circuitry) implemented in the master scheduler in the scheduling of communications between two separate computing elements.

LINKPLAY EXHIBIT 1004
Page 3280 of 4822

Figure 9 presents a detailed flowchart of the scheduling logic which is followed at the master scheduler.

Figures 10 presents a bar chart depicting the first scheduled communications between two separate computing elements after the first connection is scheduled.

Figure 11 presents a bar chart depicting the scheduled communications between computing elements after all connections are scheduled which can be scheduled in the first step.

Figure 12 presents a bar chart depicting the scheduled communications between computing elements after all connections have been scheduled.

Figure 13 provides a detailed composition of the Message Carrier, the Master Scheduler, the Clock, and the Member Nodes.

Figure 14 shows the relationships between the technologies which have been combined in the preferred embodiment, with emphasis on those most critical to the application.

Figure 15 presents a flowchart of the logic which executes in each of the computing elements in order to facilitate communication between the member nodes.

Figure 16 depicts a diagram of the data flows between the various elements in a distributed network where communications are facilitated by radio frequency.

As an introduction to this part of the specification, the system and method of the present invention implement centralized scheduling of cooperatively performed tasks, time deterministic event operations, and effective real-time synchronization and alignment of oscillators respective to the plurality of computers (also referenced as "member nodes") on the network. In this regard, the time deterministic network involves communicating between separate computing elements based on time rather than on some event (examples of events are the reception of a token or the detected availability of a particular bus). Centralized scheduling of cooperatively performed tasks, time deterministic event operations, and effective real-time synchronization and alignment of oscillators respective to the plurality of computers on the network requires that the computers share an essentially common view of the datalogical representation of an actual moment in real-time. This means, for instance

LINKPLAY EXHIBIT 1004
Page 3281 of 4822

and assuming for this example (without limitation to the scope of the present invention) a system wide-precision requirement of one millionth of a second , that two different computers in the system essentially perceive that, when the moment of time in real-time defined by an undisputed source respective to the network is 08:00:0000000 hours, that the first of the two computers has a datalogical representation of the moment in real-time between 07:59:9999995 and 08:00:0000005 and that the second of the two computers also has a datalogical representation of the moment in real-time time between 07:59:9999995 and 08:00:0000005; as can be appreciated, the difference in the datalogical representation of the moment in real-time between the two computers is no more than 1 millionth of a second.

When the computers in a distributed computing network share an essentially common view of the datalogical representation of an actual moment in real-time, they can then be programmed to perform specific tasks based on time. Tasks which involve multiple computers can then be scheduled centrally and a basis is established for time-dependent activities to be invoked on geographically dispersed computers in a deterministic manner. In this regard, determinism in real-time operations characterizes the occurrence of a predefined event at a predefined moment of real-time.

The use of such time coordination in communications between computers also facilitates the employment of active redundancy. In active redundancy, there are two or more computers which are each executing essentially identical logic, exchanging the results from these executions, optionally arbitrating the independently derived results to come to a commonly understood mutual conclusion for further use, and acting to concurrently effect the result. This process proceeds most effectively when data received by each computer is shared with the other computer(s) executing the same logic prior to arbitrating the results, so there must be some mechanism by which the time to begin the arbitration process is mutually determined. In the present invention, the mechanism for providing time synchronization between the redundant computers and to schedule the tasks related to a communication instance is (1) to provide an essentially identical datalogical representation of the moment in real-time time among the computers; (2) to use a scheduling approach to define in each computer the moment in time when the communication instance in that computer will occur; and (3) to then to implement the appropriate sending, receiving, and linkage tasks in each system involved in the communication instance at the time defined for the communication instance. In this way, the time at which the referenced arbitration task will be performed is

LINKPLAY EXHIBIT 1004
Page 3282 of 4822

pre-determined after the communications tasks have been scheduled (as opposed to the approach of waiting to determine the start time of the arbitration task after the communication tasks have actually executed).

The scheduling approach is implemented by execution of scheduling logic. The concept of logic relates to software (a computer program) whose purpose is achieved in execution by an enabling system of electrical circuitry (usually denoted by the term "computer"). The scheduling logic used for the scheduling of communications is based on analysis of demands for the resources (for example specific computers, data space within each computer, execution time in each computer, and specific circuits within each computer) which are being scheduled. This scheduling logic is used in the scheduling of network communications, but it has further applicability in the scheduling of any resources which require interaction among one another, and it is especially suited for real-time applications. The scheduling logic is further designed to be efficient respective to the resources it is responsible for scheduling and to be predictable as to the amount of time required to generate its output, a schedule, from the inputs (that is a collection of connection requests between any two computers on this network). In achieving this task, the scheduling logic schedules the available resources with the highest demand for the time that the schedule is being created. Reflecting the nature of (1) evaluating the resources with the highest demand at each point in the schedule and (2) giving precedence to the resources which have the highest demand, the particular scheduling logic method described in this specification is also referred to herein as Apparent Critical Resource, or ACR, scheduling logic. In this regard, in the preferred embodiment, the scheduling logic is not iterative in the sense that, once a particular task is scheduled, it is NOT removed from the schedule in order to pursue an alternative to its placement in the schedule. It is this quality which enables the preferred ACR scheduling logic type of scheduling logic to be well-suited for real-time applications.

In the application of scheduling logic to the scheduling of network communications, there are transmission resources and reception resources which are to be considered. Each time a network communication is scheduled, a transmission resource is associated with a reception resource.

In the scheduling of network communications, there are responsibilities which are handled by the master scheduler and those which are handled by the member nodes.

LINKPLAY EXHIBIT 1004
Page 3283 of 4822

Figure 1 presents a data flow diagram 130 of the general elements and associated data flows between the various general elements in a distributed network and shows a message carrier 100, a first member node 107 and a second member node 108, a master scheduler 106 for coordinating communications between member nodes 107, 108 via data communication path 101 and data communication path 102, clock alternatives (as clock 109 and alternative clock 110) for achieving synchronization between member nodes 107, 108 (via data communication path 104, data communication path 105, data communication path 101, and data communication path 102), and the primary data tables and exchanges required to interact effectively in this distributed computing environment (via data communication path 118, data communication path 117, data communication path 119, data communication path 120, and data communication path 121).

The clock 109 is distributed to the member nodes through the message carrier from the master scheduler (alternatively, the message carrier 100 could distribute an alternative local clock 110 to the member nodes 107, 108 and to the master scheduler 106). This permits all nodes attached to the message carrier to share a common view of time. In this regard, clock 109 represents a global clock for use around the physical globe (for example in North America, in Europe, and in Asia) where the alternative clock 110 represents a clock which has been designated for the network referenced only by message carrier 100 and where a computer executing aspects of message carrier 100 arbitrarily defines time for its affiliated computers in the distributed network. Clock 109 or clock 110 periodically output a common clock signal via either linkage 104 or 105, respectively.

Each member node 107, 108 is responsible for receiving the common clock signal; effectively "synchronizing" (aligning the definition of time in the node to a close tolerance with the definition of time respective to the common clock signal to create a system time base operating in synchronism with an established clock frequency and anchor point) to that common clock signal; receiving an appropriate local event table 114, 116 from the master scheduler 106; formulating a respective task execution list based on the received local event table 116, 114 and any additional tasks which are not described in the local event table 114, 116; executing the task execution list; generating the request list of communications 113, 115; and transmitting the request list of communications to the master scheduler 106. Each member node 107, 108 generates a connection request list 113, 115 based on the tasks required of the node 107, 108, and forwards these requests at a prescribed time through the

LINKPLAY EXHIBIT 1004
Page 3284 of 4822

message carrier 100 to the master scheduler 106. The master scheduler 106 combines the connection request lists 113, 115 from the member nodes into a composite request list 111, which it uses to generate a communications schedule represented in the master event table 112. This master event table 112 is then broken apart into sections applicable to each of the member nodes 107, 108 and sent through the message carrier 100 (also at prescribed times) to the member nodes 107, 108 for storage in local event tables 114, 116. The local event tables 114, 116 are used in the member nodes 107, 108 to control timing for applicable time dependent tasks, including communications with other member nodes.

Figure 2 depicts a distributed network 220 of computing elements which are resident in physically different locations throughout the world to show an instance of the situation abstracted in the data flow diagram 130 of Figure 1. This figure includes a network of computers 201-214 and two sources of time in clock 203 and clock 214 representing time sources respective to the data flow diagram clocks 109, 110; in this regard, Figure 2 shows clock 203 as the basis for time definition for all systems in North America and Eurasia while clock 206 defines a basis for time definition for the less globally dispersed domain of computers 206, 207, 208, 209, and 214 in Africa. It should be noted that clock 203, then, "maps" to clock 109 in the data flow diagram context of Figure 1 and that clock 214 "maps" to clock 110 in the data flow diagram context of Figure 1. Figure 2 thereby depicts two possible configurations for distributed computing environments (facilitating the associated message carriers 100, master schedulers 106, and clocks 104, 105). A distributed computing environment can be located in a geographically limited area or dispersed throughout the world. When spread across large geographical areas, the clock 203 is preferably a satellite timing distribution system such as the Global Positioning System.

Figure 3 presents a diagram of a network in a bus structure 330. Figure 3 shows the fundamental components of Figure 1 except that the message carrier 100 is a bus 300 connecting the member nodes 107, 108, to the master scheduler 106.

Figure 4 shows a distributed computing environment where the network is implemented in a ring topology 430. Figure 4 shows the fundamental components of Figure 1 except that the message carrier 100 is a ring enabled by ring segments 402, 403, 405, and 401 to effectively connect the member nodes 107, 108, to the master scheduler 106.

LINKPLAY EXHIBIT 1004
Page 3285 of 4822

Figure 5 presents a diagram of a network in a switched star topology 530 which is a preferred network architecture embodiment. Figure 5 also shows the fundamental components of Figure 1 except that the message carrier 100 is explicitly a switched network implemented in a star topology which connects the member nodes 107, 108 and the master scheduler 106, and that this master scheduler 106 is embedded into the network switch 501. This diagram shows the preferred embodiment as it could be used in a large manufacturing process (a manufacturing process referencing the effecting of goods manufacture in an apparatus whose movable components are modified in real-time using control signals from a computer control system executing logic reading measurements from the apparatus and outputting the control signals) through a large number of input/output processing devices and computing platforms in a potentially geographically distributed arrangement. The star topology network 530 enables the reading of measurements and outputting of control signals respective to the apparatus for manufacture through I/O (input signal/output signal) interface 510, I/O interface 511, I/O interface 522, I/O interface 519, I/O interface 512, I/O interface 513, I/O interface 520, and I/O interface 521. I/O interface 510, I/O interface 511, I/O interface 512, and I/O interface 513 are all joined via ring breakout 508 and ring breakout 509 to control computer 504 and historian computer 505 through local network 515. In a similar manner, I/O interface 522, I/O interface 519, I/O interface 520, and I/O interface 521 are all joined via ring breakout 518 and ring breakout 517 to control computer 506 and historian computer 507 through local network 514. I/O interface 510, I/O interface 511, I/O interface 512, I/O interface 513, ring breakout 508, ring breakout 509, control computer 504, historian computer 505, and local network 515 operate with local logical integrity and high real-time datalogical cohesion. In a similar manner, I/O interface 522, I/O interface 519, I/O interface 520, I/O interface 521, ring breakout 518, ring breakout 517, control computer 506, historian computer 507, and local network 514 operate with local logical integrity and high real-time datalogical cohesion. Cohesion between

(a) I/O interface 510, I/O interface 511, I/O interface 512, I/O interface 513, ring breakout 508, ring breakout 509, control computer 504, historian computer 505, and local network 515

and

LINKPLAY EXHIBIT 1004
Page 3286 of 4822

(b) I/O interface 522, I/O interface 519, I/O interface 520, I/O interface 521, ring breakout 518, ring breakout 517, control computer 506, historian computer 507, and local network 514

via messaging is enabled by network switch 501 which implements (1) communications for provision of an essentially identical datalogical representation of the moment in real-time time among the computers; (2) master scheduler 106 processes to define for control computer 506, historian computer 507, control computer 504, historian computer 505, user interface 503, network support 502, and network switch 501 each moment in time when each communication instance between any two of computer 506, historian computer 507, control computer 504, historian computer 505, and user interface 503 VIA THE USE OF network switch 501 will occur; and (3) appropriate sending, receiving, and linkage tasks respective to components involved in the communication instance at the time defined for the communication instance. Network support 502 is also coordinated as a node into switched star topology 530 as a single source of information for defining acceptable communication partners in switched star topology 530 and also for accumulating and providing access to diagnostic information in switched star topology 530. Network switch 501 references network support 502 for a listing of acceptable communication partners in switched star topology 530 and confirms communication instance requests against this listing in defining sending, receiving, and linkage tasks.

In one embodiment, I/O interface 522, I/O interface 519, I/O interface 520, and I/O interface 521 are facilitated by a process control system such as is described in United States Patent 5,555,424 (24Sederlund, et. al.) issued on Sept. 10, 1996 and entitled "Extended Harvard architecture computer memory system with programmable variable address increment" to Sederlund, Edward R.; Lindesmith, Robert J.; Root, Larry A.; Dupree, Wayne P.; and Thomas, Lowell V. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

In a more preferred embodiment, I/O interface 522, I/O interface 519, I/O interface 520, and I/O interface 521, are facilitated via a redundant process control computer system in a general process control system using two process control computers (such as the process control computer described in 24Sederlund, et.) as is described in United States Patent 5,583,757 (Baca, Jr., et. al.) issued on Dec. 10, 1996 and entitled "Method of input signal resolution for actively redundant process control computers" to Baca, Jr., Eloy; Dupree,

LINKPLAY EXHIBIT 1004
Page 3287 of 4822

Wayne P.; Grinwis, Donald J.; Kanse, Johannes C.; Pelletier, Douglas P.; and Schulze, Oscar E. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

An embodiment of a system for achieving data access for the process control computer described in 24Sederlund, et al. is described in United States Patent 5,568,615 (15Sederlund, et. al.) issued on Oct. 22, 1996 and entitled "Stealth interface for process control computers" to Sederlund, Edward R.; Thomas, Nadene T.; Lindesmith, Robert J.; and Cowles, Russell W. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

An embodiment of a system providing a remote field unit for use with the process control computer described in 24Sederlund, et al. is described the United States Patent 5,428,769 (69Glaser, et. al.) issued on Jun. 27, 1995 and entitled "Process control interface system having triply redundant remote field units" to Glaser, Robert S.; Hoy, Robert S.; Fernandez, G. Paul; and Grai, Timothy J. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

An embodiment of a system providing an interface for reading electrical current in power distribution systems for use with the field unit described in 69Glaser, et. al. is described in United States Patent 5,151,866 (66Glaser, et. al.) issued on Sept. 29, 1992 and entitled "High speed power analyzer " to Glaser, R. Steven and Bade, Jeffrey M. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

An embodiment of a system providing a high speed gateway for use with the redundant process control computer system described in Baca, et al. and the system for achieving data access for the process control computer described in 24Sederlund, et al. is described in

(a) United States Patent 5,519,603 (Allbery, Jr., et. al.) issued on May 21, 1996 and entitled "Intelligent process control communication system and method having capability to time align corresponding data sets" to Allbery, Jr., James D.; Troisi, Peter A.; Johnson, Susan J.; Cullen, James H.; Butler, Richard L.; Ferreira, James P.; Ellison, Joseph ; Patel, Chiman L.; Uban, James E.; and Schultz, Dale H.;

LINKPLAY EXHIBIT 1004
Page 3288 of 4822

(b) United States Patent 5,428,745 (45de Bruijn, et. al.) issued on Jun. 27, 1995 and entitled "Secure communication system for re-establishing time limited communication between first and second computers before communication time period expiration using new random number" to de Bruijn, Ronny P.; Verboven, Marc L. K.; van Weele, Leonardus A.; Vermeire, Roger R.; Schulze, Oscar E.; Schultz, Dale H.; and Bell, Brian G.; and

(c) United States Patent 5,561,770 (70de Bruijn, et. al.) issued on Oct. 1, 1996 and entitled "System and method for determining whether to transmit command to control computer by checking status of enable indicator associated with variable identified in the command" to de Bruijn, Ronny P.; van Weele, Leonardus A.; Verboven, Marc L. K.; Vermeire, Roger R.; Schulze, Oscar E.; Bell, Brian G.; and Schultz, Dale H.

These patents are expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

An embodiment of a system providing human interfacing for use with the redundant process control computer system and interfaces described in the above patents is described in United States Patent 5,631,825 (van Weele, et. al.) issued on May 20, 1997 and entitled "Operator station for manufacturing process control system" to van Weele, Leonardus A.; de Bruijn, Ronny P.; Vermeire, Roger R.; Zemering, Christo; and Lenting, Ben. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention. While this human interface system could be used as one user interface in each or any one of I/O interface 522, I/O interface 519, I/O interface 520, and I/O interface 521, it should be noted that user interface 503 is preferably designed for interface via network switch 501 to the entire set of systems in a particular instance of switched star topology 530.

Embodiments of systems providing real-time interpretation of application program code executing in the above systems are described (a) in United States Patent 5,491,625 (Pressnall, et. al.) issued on Feb. 13, 1996 and entitled "Information display system for actively redundant computerized process control" to Pressnall, Dana W.; Polishak, Jeffery T.; Felix, Bradley K.; Durisin, Michael J.; and Ellison, Joseph. and (b) in United States Patent 5,408,603 (Van de Lavoir, et. al.) issued on Apr. 18, 1995 and entitled "Global process

control information system and method" to Van de Lavoir, Ronny; Follon, Marinus (Neerpelt, BE); and Ravenscroft, Ian. These patents are expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

A large graphical overview system providing interface to humans is deployed in the preferred embodiment along with the Operator Station. This graphical overview system is described in United States Patent 5,726,668 (Clement) issued on Mar. 10, 1998 and entitled "Programmable graphics panel" to John L. Clement. This patent is expressly incorporated herein by reference in the present application for showing a manner of making and using the present invention.

In a conceptualized most preferred embodiment, I/O interface 510, I/O interface 511, I/O interface 522, I/O interface 519, I/O interface 512, I/O interface 513, I/O interface 520, I/O interface 521, ring breakout 508, ring breakout 509, control computer 504, historian computer 505, ring breakout 518, ring breakout 517, control computer 506, historian computer 507, I/O interface 522, I/O interface 519, I/O interface 520, I/O interface 521, ring breakout 518, ring breakout 517, control computer 506, network support 502, network switch 501, and historian computer 507 are implemented using a dedicated context cycling computer as is described in US patent application number 08/797,967 which was filed on February 12, 1997 by Wayne Dupree, Jeffery Lucas, Larry Root, Gerrit Verniers, and Stephen Churchill entitled "A Dedicated Context-Cycling Computer" with the system of 69Glaser, et. al. facilitating interface to field instrumentation. This application for patent is expressly incorporated herein by reference in the present application for showing the preferred contemplated manner of making and using the present invention.

The context cycling processor provides for a substantial number and variety of input and output circuits, with each of the specialized circuits having at least one dedicated register for retaining the process and configuration state of pertinent resources, attributes, or aspects associated with either the circuit or function while another context is active; with the combination of specialized circuit and dedicated register(s) establishing the electrical circuitry base for a context. Contexts also include affiliated logic specific for the circuitry. A plurality of contexts share common assets within the CPU.

As needed, a co-processor for use with the context cycling computer is described in United States Patent 5,655,133 (Dupree, et. al.) issued on Aug. 5, 1997 and entitled "Massively

LINKPLAY EXHIBIT 1004
Page 3290 of 4822

multiplexed superscalar Harvard architecture computer" to Dupree, Wayne P.; Churchill, Stephen G.; Gallant, Jeffrey R.; Root, Larry A.; Bressette, William J.; Orr, III, Robert A.; Ramaswamy, Srikala; Lucas, Jeffrey A.; and Bleck, James. This patent is expressly incorporated herein by reference in the present application for showing the preferred contemplated manner of making and using the present invention. Network switch 501 also preferably uses a cross-bar (Xbar) switch 516 capable of implementing at least 99 parallel and simultaneous communication linkages.

The master scheduler 106 is implemented in network switch 501 to provide a common clock signal (as an access to either clock 109 or a manifestation of clock 110) and is further responsible for the distribution of the common clock signal; the collection of the requests from each of the member nodes 108, 107 (in this instance of network 530, control computer 506, historian computer 507, control computer 504, historian computer 505, user interface 503, network support 502, and network switch 501 are all instances of member nodes 107 and 108 in the data flow context of Figure 1); the scheduling of the communication tasks for each of the member nodes 108, 107; the transmission of the schedule to each of the member nodes 108, 107; and the implementing of the connections between the member nodes 108, 107.

The purpose of distributing a common clock signal is so that each member node 108, 107 on the network 530 has the same understanding of time. This aspect of having the same understanding of time is crucial in this invention since the scheduling of tasks is done based on time. Thus, the common clock signal is used to effectively synchronize and time-align all member nodes 108, 107 to a common time. On a pre-defined periodic basis, the master scheduler 106 in network switch 501 sends a message that is propagated to each of the member nodes 108, 107 to define the beginning of a time period.

The collection of the requests from each member node 108, 107 (any one of control computer 506, historian computer 507, control computer 504, historian computer 505, user interface 503, network support 502, and network switch 501) occurs once each scheduling period. The master scheduler 106 in network switch 501 sets aside the time to receive requests from each member node 108, 107 (any one of control computer 506, historian computer 507, control computer 504, historian computer 505, user interface 503, network support 502, and network switch 501) each scheduling period. Requests originate from the member nodes 108, 107 which will need the communications requested. The master

LINKPLAY EXHIBIT 1004
Page 3291 of 4822

scheduler 106 in network switch 501 receives each of these requests and considers them in its scheduling logic. Each request contains information on the source, destination, and length of the communication. In addition, each request may contain a priority indicator as well as information needed to deliver its message to a particular application which is executing on the destination node. After the task of collecting the connection requests from all of the member nodes 108, 107 is completed, the result is collected into a composite request list 111. Note that it is also possible for entries in the composite request list 111 to originate inside of the master scheduler 106 in network switch 501 itself; it is also possible for particular connection requests to have a repetitive nature such that multiple connections in a set of future time periods are invoked from a single connection request.

The master scheduler 106 in network switch 501 then schedules the communications which are to occur in the future. The method which is used in order to perform the scheduling of the communications is not critical in the functioning of the time deterministic network, although the Apparent Critical Resource approach discussed in this specification is the preferred approach. Logic implements to assign times to the requested communications. The determination of when to stop the scheduling process during the scheduling period can occur in one of three ways: first, all of the communication requests which can be scheduled are scheduled and therefore, there is no more scheduling work remaining; second, there is no more time remaining in which to perform the scheduling (the processor must stop the scheduling process in order to continue on with its next task); and third, there have been more connections requested than could be scheduled in the available scheduling window of time increment (an exhaustion of bandwidth), in which case the excess requests are retained as a queue for the next scheduling attempt.

The resulting schedule is stored in a master event table 112. This master event table 112 contains the start time 621 of the scheduled communication in addition to some of the same information similar to that in the composite request list 111. Note that it is also possible that certain tasks are pre-loaded into the master event table 112 in order to bypass the scheduling process, such as is required to enforce certain timing requirements of a task.

The master scheduler 106 in network switch 501 then communicates the local event table 114, 116 (that portion of the schedule which pertains to a particular member node 107, 108) to each appropriate member node 107, 108. For each event appearing in the master event table 112 which includes an appropriate member node 108, 107 as either the source or the

LINKPLAY EXHIBIT 1004
Page 3292 of 4822

destination, an entry exists in the local event table 114, 116. Each member node 108, 107 (any one of control computer 506, historian computer 507, control computer 504, historian computer 505, user interface 503, network support 502, and network switch 501) which is involved in any communication receives its own local event table 114, 116.

The master scheduler 106 in network switch 501 then implements the connections according to the times which are assigned and appear in the master event table 112. The processing of these connections when using the preferred embodiment of the non-blocking switch leads to the situation whereby there may be multiple connections which are implemented concurrently.

The Composite Request List 111 residing in the master scheduler 106 in network switch 501 may contain both continuously repetitive connections to be implemented without contingencies, or dynamically requested connections (implemented when requested, usually initiated by one of the nodes on the network). This Connection Request List contains, at a minimum, a Source specifier, a Destination specifier, and a Duration specifier (see Figure 6 showing record details 600). Additional parameters pertinent to scheduling optionally include such specifiers as Priority 617.

Figure 6 shows the composition of the primary tables and messages 600 used in the scheduling and task coordination process. The connection request list 115 has Source 614, Destination 615, Length 616, and possibly Priority 617 fields 610 in both the member nodes 108, 107 and when combined into the composite request list 111 in the master scheduler 106 in network switch 501. The scheduler logic 611 in master scheduler 106 uses the data in these fields 610 to construct a schedule in the form of a master event table 112 with fields 612 for the Source 618, Destination 619, Length 620 and Start Time 621 for each of the connections. Note that the scheduler logic 611 takes the Source 614, Destination 615, and Length 616 from the connection request list 111, while the Start Time 621 is a product of the scheduling process itself.

The local event tables 114, 116 are simply subsets of the master event table 112, broken into those portions applicable to the respective member node 108, 107 (any one of control computer 506, historian computer 507, control computer 504, historian computer 505, user interface 503, network support 502, and network switch 501) - no manipulation of the fields 613 is required. Virtual paths 601, 602, 603, 606, 607, and 608 show the flow of the

LINKPLAY EXHIBIT 1004
Page 3293 of 4822

information from the connection request list fields 610 to the local event table fields 613. Virtual path 604 shows the input of priority 617 to the scheduler; and virtual path 609 shows the start time 621, 625 as communicated from scheduler logic 611.

This data is then used by the scheduler logic 611 to derive a schedule for servicing the requested connections. The outcome of the scheduling process of the scheduler logic 611 is recorded in the Master Event Table 112, which is in turn parsed up into respective Local Event Tables 116 for distribution to the network nodes (see Figure 1). The Event Tables contain Source 622, Destination 623, Duration 624, and Start Time 625 fields 613.

Within the network nodes, the Local Event Tables 114, 116 are merged together with the Local Task Lists and logic executes to ensure that all the necessary work can be completed and that all timing corresponds to (or doesn't conflict with) that specified by the scheduler logic 611.

Figure 7 summarizes the general scheduling process 710. Connection requests are collected during a defined request acceptance period. After a defined period of time, the collection of connection requests must be processed. Additional connection requests are not prohibited, but a cutoff point must be established. Figure 7 shows the basic steps involved in the master scheduler 106 in network switch 501 node which are required in scheduling the connection requests. The first three steps 701-703 provide for aligned time definition in all nodes . In step 704, connection requests are received from all member nodes 107, 108 (the nodes sending their requests as a communication at predefined moments in their local task schedules from prior scheduling operations). In step 705, the connection requesting process ends so that a schedule can be defined for the future as the scheduling process applies to those requests which were collected from the member nodes 108, 107 during a particular window of time increment which is about to conclude. Schedule connections requests step 706 can begin once the requests have been collected (more detail in Figure 8). Once the schedule is complete, the applicable portions of the schedule are extracted from the master event table 112 in the creation of the local event tables 114, 116, in step 707 which are then conveyed to the nodes 107, 108 in step 708 for later implementation in step 709.

For distributed computing environments in general, connections between nodes 107, 108 may be required at any time, meaning that connection requests may originate any time, and some service time must be tolerable in handling the connections. There are four processes,

LINKPLAY EXHIBIT 1004
Page 3294 of 4822

then, that repetitively execute at a rate conformant to the desired response in the network 530: collecting connection requests, scheduling connection requests, distributing the resulting schedule to the network nodes, and communicating through the network 530 via the established requested connections (see Table 1).

Table 1 - Four Scheduling Steps

Request Connections
Schedule Connections
Convey Schedule
Communicate

Table 2 shows the timing relationship between these four processes which, as stated, are always in progress.

Table 2 - The Scheduling Pipeline

Request Connections			
Request Connections	Schedule Connections		
Request Connections	Schedule Connections	Convey Schedule A	
Request Connections	Schedule Connections	Convey Schedule B	Communicate A
Request Connections	Schedule Connections	Convey Schedule C	Communicate B
Request Connections	Schedule Connections	Convey Schedule D	Communicate C
Request Connections	Schedule Connections	Convey Schedule E	Communicate D
Request Connections	Schedule Connections	Convey Schedule F	Communicate E

Note in the rightmost column of Table 2 that communications are always taking place after an initial few iterations establish a basis for communications. For any particular communications in a window of time increment, the schedule for the communications being implemented was conveyed in a previous window of time increment, the requests were scheduled in a window of time increment prior to the conveyance of the schedule, and the requests were requested even prior to scheduling. Timing variations can exist in this pipeline, but this resembles the pipeline used in the preferred embodiment.

The logic used in the currently preferred embodiment has been designed with two objectives in mind: an efficient schedule must be the outcome, and the number of cycles required to generate that schedule must be bounded at a reasonable level. Efficient scheduling involves a simple process based on the premise that the efficiency of the resultant schedule is primarily a function of the efficiency to which the busiest resources have been utilized in the schedule. This amounts to making sure that, at all times in the scheduling process,

LINKPLAY EXHIBIT 1004
Page 3295 of 4822

scheduling precedence is given to the resources with the highest remaining demands. In order to derive a schedule in an acceptable amount of time, the logic was designed to schedule each request once and only once (trial and error logics deemed unacceptable), and with a moderate handling requirement for each connection request.

The basic process in scheduling 806, as shown in Figure 8, is to

1) define a criteria for measuring resource demand;
2) sort the resources according to criticality;
3) service those connections which can be serviced in this step and in the order of resource criticality;
4) step forward to the completion of the previously serviced requests thereby freeing up the applicable resources;
5) repeat steps 2-4 until all requests have been serviced or until there's no opportunity to service more requests.

Any requests which weren't able to be serviced in this scheduling pass can then be resubmitted during the next scheduling pass.

The flowchart of Figure 8 outlines the process used in the Apparent Critical Resource scheduling logic. The first step is to define the criteria by which resource usage is to be measured 800. For the preferred embodiment, transmit and receive channels are the resources which are scheduled, and the usage duration for each of these resources is the criteria chosen.

Once the resources are evaluated for usage, they are sorted by usage 801. This allows the resources which are used most heavily to be scheduled in step 802 first. When all resources which can be scheduled have been scheduled (the existing requested connections may not require or permit the simultaneous use of all resources), the scheduling window of time increment is advanced in step 803 until applicable resources become available. If there are more connection requests to be serviced and more time in the schedule to service these requests, this process is repeated after step 804 resolves the question respective to the need for another iteration through steps 801 to 803). The sub process 806 of the scheduler logic proceeds to step 805 to exit upon resolution by step 804 that the scheduling instance has been completed.

LINKPLAY EXHIBIT 1004
Page 3296 of 4822

Figure 9 presents a more detailed overview of the Apparent Critical Resource (ACR) preferred embodiment sub-process 950 of the scheduling logic. This diagram encompasses the steps mentioned in Figure 8 and some additional actions required for conditional operation and exception handling as the preferred implementation of the process shown in Figure 8. Step 901 represents the initiation point for the preferred ACR process. In step 902, the ACR table is constructed, and a determination is made in step 903 respecting the presence of unscheduled requests based on the values of the SAV in the ACR table (a rapid exit to step 918 is executed if the values indicates that no requests are in the ACR table). If requests exist, the ACR sub-process 950 proceeds to calculate window size in step 904, look for a specific request in step 905, and check on the existence of the specific request and parties for the communication in step 906. If the request and parties exist, the process proceeds to add the event to the event table in step 912 and modify the ACR path in step 913 before proceeding to the next request in step 914. If the high value is not zero in step 915, the process proceeds again to step 904. If the high value is zero in step 915, the definition of (future) time in the ACR is advanced in step 916, the bandwidth is checked in step 917 for sufficiency in time to enable further scheduling in this scheduling operation, and the process returns to step 914 for the next request. If in step 906, a receiver for the communication did not exist, an alternative receiver is defined in step 907 and evaluated in step 908. If step 908 does not define an alternative receiver, an alternative transmitter is selected in step 919 and evaluated in step 920; if acceptable (step 920), step 921 selects the receiver with the highest ACR value and returns the process to step 908; and, if not acceptable (step 920), the process advances to step 916. When an alternative is defined in step 908, the process proceeds to schedule via steps 909, 910 and 911 to step 912. The virtual datalogical process being executed by these steps can be best described by considering an example respective to the list of communication requests for a hypothetical network consisting of five nodes shown in Table 3:

LINKPLAY EXHIBIT 1004
Page 3297 of 4822

Table 3: Initial Request List

Source	Destination	Length	Priority
1	2	15	1
1	4	23	1
1	5	2	1
2	1	24	1
3	1	16	1
3	1	12	1
3	2	20	1
3	3	8	1
3	4	19	1
3	4	16	1
3	5	4	1
3	5	9	1
4	1	13	1
4	5	9	1
5	1	21	1
5	1	8	1
5	2	8	1
5	3	1	1
5	4	18	1
Total		246	

Assume that the length is measured in some arbitrary units of time, perhaps in milliseconds, and that there are n different priority levels, 1 through n, where 1 is the most urgent and n is the least.

First, note that with a broadcast type network (where all transactions are required to share the bandwidth, such as a Carrier Sense Multiple Access with Collision Detection or CSMA/CD network), service time required to complete the above requests would require a theoretical minimum of 246 units of time.

The incorporation of a non-blocking switch permits the simultaneous operation of the 5 transmit resources and the 5 receive resources. In analyzing the benefit of implementing a non-blocking network, the scheduler logic of the example first assesses the demands on each of the resources by constructing a table such as Table 4:

Table 4: Initial Summed Access Values

X / R		1	2	3	4	5		SAV
1		-	15	-	23	2		40
2		24	-	-	-	-		24
3		28	20	8	35	13		104
4		13	-	-	-	9		22
5		29	8	1	18	-		56
SAV		94	43	9	76	24		246

The summed access values, or SAVs, are simply the row or column totals for the transmitter (X-in the rows of the table) and receiver (R-in the columns of the table) interaction times and reflect the total demands on the particular resource, as measured in the arbitrary units of length. Note now that the theoretical minimum time required to service the requests has dropped from 246 units of time to 104 by selecting a non-blocking switch solution. The minimum service time for these requests is probably constrained by transmitter 3 since transmitter 3 has the highest SAV and the other transmitters can typically be serviced while transmitter 3 is being serviced. While the theoretical lower limit may be 104 units of time, the actual minimum service time may be greater and may even be dependent upon the order in which the connections are serviced. The next objective, however, is determining the optimum scheduling method in order to maximize the capabilities of the existing network (minimizing the required service time).

Assume for this particular example that the resultant schedule, regardless of the number or composition of the requests, applies to a future window of time increment of time with a duration of 150 units of time. The example begin by looking at the connections to be made at

LINKPLAY EXHIBIT 1004
Page 3299 of 4822

Time = 0 for that 150 unit schedule. The primary objective, judging by the SAVs, is to schedule some request involving Transmitter 3 and Receiver 1. Referring back to the initial Request List, the scheduler logic of the example finds a request involving Transmitter 3 and Receiver 1 which has a length of 16 units of time. Since the scheduler logic of the example is currently scheduling connections at Time = 0 and the scheduler logic of the example has 150 units of time, the scheduler logic of the example concludes that the 16 word request fits into the scheduling window of time increment , and records the transaction into the Event Table of Table 5.

Table 5: Intermediate Event Table

Source	Destination	Length	Priority	Start Time
3	1	16	1	0
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

Pictorially, Table 5 may be shown as schedule chart 1001 in Figure 10. Transmitter scheduled element 1002 and receiver scheduled element 1003 denote the scheduled tasks.

The Request List must then somehow reflect the fact that this last request has been serviced. This modified Request List may now resemble that shown in Table 6.

Table 6: Intermediate Request List

Source	Destination	Length	Priority	Start Time
1	2	15	1	-
1	4	23	1	-
1	5	2	1	-
2	1	24	1	-
3	1	16	1	0
3	1	12	1	-
3	2	20	1	-
3	3	8	1	-
3	4	19	1	-
3	4	16	1	-
3	5	4	1	-
3	5	9	1	-
4	1	13	1	-
4	5	9	1	-
5	1	21	1	-
5	1	8	1	-
5	2	8	1	-
5	3	1	1	-
5	4	18	1	-
Total		246		

Since Transmitter 3 and Receiver 1 are now busy for Time = 0, their status must indicate this, so the updated SAV Table, following some reformatting, may now resemble Table 7:

Table 7: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	40	78
2	24	43
3	88	9
4	22	76
5	56	24

Note that the SAVs for Transmitter 3 and Receiver 1 have been adjusted to reflect the serviced request. This is because the scheduler logic continues to give scheduling precedence to the resources with the highest impact on the schedule. This happens at the expense of the other resource needs, and at some point the inefficiencies injected into the resources with low SAVs may cause those resources to have a higher impact on the remaining portion of the schedule than those resources which had the highest impact at the start. For this reason, the protocol is referred to as Apparent Critical Resource scheduling (scheduling precedence is always given to resources which appear to be on the critical path at the time of consideration), or ACR scheduling.

It is now evident that the scheduler logic of the example should try to schedule a request from Transmitter 5 to Receiver 4, the next two concurrent resources in regards to scheduling impact which remain available for scheduling. There is such a request, so the scheduler logic of the example executes so that the tables now resemble Tables 8, 9, and 10.

LINKPLAY EXHIBIT 1004
Page 3302 of 4822

Table 8: Intermediate Request List

Source	Destination	Length	Priority	Start Time
1	2	15	1	-
1	4	23	1	-
1	5	2	1	-
2	1	24	1	-
3	1	16	1	0
3	1	12	1	-
3	2	20	1	-
3	3	8	1	-
3	4	19	1	-
3	4	16	1	-
3	5	4	1	-
3	5	9	1	-
4	1	13	1	-
4	5	9	1	-
5	1	21	1	-
5	1	8	1	-
5	2	8	1	-
5	3	1	1	-
5	4	18	1	0
Total		246		

LINKPLAY EXHIBIT 1004
Page 3303 of 4822

Table 9: Intermediate Event Table

Source	Destination	Length	Priority	Start Time
3	1	16	1	0
5	4	18	1	0
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

LINKPLAY EXHIBIT 1004
Page 3304 of 4822

Table 10: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	40	78
2	24	43
3	88	9
4	22	58
5	38	24

Note that the Event Table is effectively a (chronologically) sorted list of the scheduled requests from the Request List. For this reason, intermediate stages are not shown for each scheduling step.

Repeating the process as before, the scheduler logic of the example schedules a 1-to-2 communication instance request (that is a connection from Transmitter 1 to Receiver 2), and the tables are modified to Tables 11 and 12 as follows:

LINKPLAY EXHIBIT 1004
Page 3305 of 4822

Table 11: Intermediate Request List

Source	Destination	Length	Priority	Start Time
1	2	15	1	0
1	4	23	1	-
1	5	2	1	-
2	1	24	1	-
3	1	16	1	0
3	1	12	1	-
3	2	20	1	-
3	3	8	1	-
3	4	19	1	-
3	4	16	1	-
3	5	4	1	-
3	5	9	1	-
4	1	13	1	-
4	5	9	1	-
5	1	21	1	-
5	1	8	1	-
5	2	8	1	-
5	3	1	1	-
5	4	18	1	0
Total		246		

LINKPLAY EXHIBIT 1004
Page 3306 of 4822

Table 12: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	25	78
2	24	28
3	88	9
4	22	58
5	38	24

The scheduler logic of the example next looks for a 2-to-5 communication instance request, but such a request was not made. Several alternatives exist in determining the next combination to be attempted, and the optimum solution is chosen based on the desired tradeoffs between bandwidth efficiency and processing costs. (With minimal bandwidth impact, a simple rule set can be chosen, and for the preferred embodiment, an alternative receiver is selected for the existing transmitter until all receiver choices have been exhausted, after which that particular transmitter is ignored and the next transmitter is selected, and the highest impact receiver is re-considered - see Figure 9.)

Using this logic, a 2 to 3 communication instance request is searched for but not found, then a 4 to 5 communication instance request is searched for, found, and selected. The SAV Table is updated, and it is determined that no other connections can be made at Time = 0. The Event Table now is shown in Table 13:

LINKPLAY EXHIBIT 1004
Page 3307 of 4822

Table 13: Intermediate Event Table

Source	Destination	Length	Priority	Start Time
3	1	16	1	0
5	4	18	1	0
1	2	15	1	0
4	5	9	1	0
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-
-	-	-	-	-

Again, these four connections can be represented pictorially in schedule chart 1101 in Figure 11. Transmitter scheduled element 1002 and receiver scheduled element 1003 from Figure 10 have been denoted as scheduled elements 1102 and 1103, respectively, and have been joined by scheduled elements 1104 and 1105 (the 5-to-4 communication), 1106 and 1107 (the 1-to-2 communication), and 1108 and 1109 (the 4-to-5 communication).

The logic involved in Advancing Time amounts to advancing to the next chronological finish. In this case, the 4 to 5 communication instance connection is the next chronological finish,

LINKPLAY EXHIBIT 1004
Page 3308 of 4822

so the example advances time to Time = 9, at which time Transmitter 4 and Receiver 5 are again enabled. The Request List and SAV Table are now as shown in Tables 14 and 15.

Table 14: Intermediate Request List

Source	Destination	Length	Priority	Start Time
1	2	15	1	0
1	4	23	1	-
1	5	2	1	-
2	1	24	1	-
3	1	16	1	0
3	1	12	1	-
3	2	20	1	-
3	3	8	1	-
3	4	19	1	-
3	4	16	1	-
3	5	4	1	-
3	5	9	1	-
4	1	13	1	-
4	5	9	1	0
5	1	21	1	-
5	1	8	1	-
5	2	8	1	-
5	3	1	1	-
5	4	18	1	0
Total		246		

LINKPLAY EXHIBIT 1004
Page 3309 of 4822

Table 15: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	25	78
2	24	28
3	88	9
4	13	58
5	38	15

After searches for a 2-to-5 communication instance request, a 2-to-3 communication instance request, a 4-to-5 communication instance request and a 4-to-3 communication instance request, the scheduler logic of the example concludes that the example must again advance time - nothing could actually be scheduled at Time = 9. Now, the next finish is at Time = 15, at which time the 1-to-2 communication instance connection finishes and the SAV Table is now Table 16.

Table 16: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	25	78
2	24	28
3	88	9
4	13	58
5	38	15

The scheduler logic of the example now looks for a 1-to-2 communication instance request, but the only 1-to-2 communication instance has already been scheduled, so the example look for a 1-to-5 communication instance request, which the example finds and schedules. The new SAV Table is then Table 17.

LINKPLAY EXHIBIT 1004
Page 3310 of 4822

Table 17: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	23	78
2	24	28
3	88	9
4	13	58
5	38	13

The scheduler logic of the example then looks for a 2-to-2 communication instance request, a 2-to-3 communication instance request, a 4-to-2 communication instance request and a 4-to-3 communication instance request before determining that it's time again to advance time. The example advances time from 15 to 16, at which time the SAV Table looks like Table 18.

Table 18: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	23	78
2	24	28
3	88	9
4	13	58
5	38	13

The scheduler logic of the example then schedules a 3-to-1 communication instance request, which turns out to be the only request which can be scheduled at Time = 16. The tables, after the scheduler logic of the example advances time to Time = 17, look like Tables 19, 20, and 21.

LINKPLAY EXHIBIT 1004
Page 3311 of 4822

Table 19: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	23	66
2	24	28
3	76	9
4	13	58
5	38	13

LINKPLAY EXHIBIT 1004
Page 3312 of 4822

Table 20: Intermediate Request List

Source	Destination	Length	Priority	Start Time
1	2	15	1	0
1	4	23	1	-
1	5	2	1	15
2	1	24	1	-
3	1	16	1	0
3	1	12	1	16
3	2	20	1	-
3	3	8	1	-
3	4	19	1	-
3	4	16	1	-
3	5	4	1	-
3	5	9	1	-
4	1	13	1	-
4	5	9	1	0
5	1	21	1	-
5	1	8	1	-
5	2	8	1	-
5	3	1	1	-
5	4	18	1	0
Total		246		

Nothing can be scheduled at Time = 17, so the example advance to Time = 18 where the 5-to-4 communication instance request finishes up.

Table 21: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	23	66
2	24	28
3	76	9
4	13	58
5	38	13

After unsuccessfully searching for a 5-to-4 communication instance request, a 5-to-2 communication instance request is found, a 1-to-4 communication instance request is found, and the tables now look like Tables 22 and 23:

Table 22: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	66
2	24	20
3	76	9
4	13	35
5	30	13

Table 23: Intermediate Request List

Source	Destination	Length	Priority	Start Time
1	2	15	1	0
1	4	23	1	18
1	5	2	1	15
2	1	24	1	-
3	1	16	1	0
3	1	12	1	16
3	2	20	1	-
3	3	8	1	-
3	4	19	1	-
3	4	16	1	-
3	5	4	1	-
3	5	9	1	-
4	1	13	1	-
4	5	9	1	0
5	1	21	1	-
5	1	8	1	-
5	2	8	1	18
5	3	1	1	-
5	4	18	1	0
Total		246		

The scheduler logic of the example now advances time to Time = 26 for which the scheduler logic of the example defines Table 24:

Table 24: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	66
2	24	20
3	76	9
4	13	35
5	30	13

The scheduler logic of the example now schedules a 5-to-3 communication instance, which is all that can be scheduled at this time. Time is then advanced to 27, then to 28, and the SAV Table looks like Table 25.

Table 25: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	66
2	24	20
3	76	8
4	13	35
5	29	13

Since there are no 3-to-1 communication instance requests, the scheduler logic of the example schedules a 3-to-2 communication instance request and a 5-to-1 communication instance request, and advances time to 31 and then to 48, at which time the SAV Table becomes Table 26.

Table 26: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	45
2	24	-
3	37	8
4	13	16
5	8	13

The scheduler logic of the example then schedules the 3-to-4 communication instance request and advances time to 49, giving Table 27 as a result.

Table 27: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	45
2	24	-
3	37	8
4	13	16
5	8	13

The scheduler logic of the example then books a 2-to-1 communication instance request and advances time to 67, giving Table 28.

LINKPLAY EXHIBIT 1004
Page 3317 of 4822

Table 28: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	21
2	-	-
3	29	8
4	13	16
5	8	13

The scheduler logic of the example now books the 3-to-3 communication instance request and advances time to 73 as shown in Table 29.

Table 29: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	21
2	-	-
3	29	-
4	13	16
5	8	13

The scheduler logic of the example then books the 4-to-1 communication instance request, and advances time to 75, resulting in Table 30.

LINKPLAY EXHIBIT 1004
Page 3318 of 4822

Table 30: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	8
2	-	-
3	29	-
4	-	16
5	8	13

This allows the remaining 3-to-4 communication instance request to be scheduled by the scheduler logic of the example, after which the scheduler logic of the example advances to Time = 86, resulting in Table 31 data.

Table 31: Intermediate Summed Access Value Table

Channel	Transmitter SAV	Receiver SAV
1	-	8
2	-	-
3	13	-
4	-	-
5	8	13

The scheduler logic of the example now schedules the remaining 5-to-1 communication instance request, advances time to 91, schedules the first 3-to-5 communication instance request, advances time to 94, then to 95, and schedules the final request, another 3-to-5 communication instance request. The resultant Event Table then looks like Table 32.

LINKPLAY EXHIBIT 1004
Page 3319 of 4822

Table 32: Final Event Table

Source	Destination	Length	Priority	Start Time
3	1	16	1	0
5	4	18	1	0
1	2	15	1	0
4	5	9	1	0
1	5	2	1	15
3	1	12	1	16
5	2	8	1	18
1	4	23	1	18
5	3	1	1	26
3	2	20	1	28
5	1	21	1	28
3	4	19	1	48
2	1	24	1	49
3	3	8	1	67
4	1	13	1	73
3	4	16	1	75
5	1	8	1	86
3	5	4	1	91
3	5	9	1	95

Note that the entire list of request completes in 104 units of time - the theoretical minimum amount of time required to service these particular requests.

Pictorially, the final schedule is shown in as schedule chart 1201 in Figure 12. Transmitter scheduled element 1002 and receiver scheduled element 1003 from Figure 10 have been denoted as scheduled elements 1202 and 1203; scheduled elements 1104 and 1105, 1106 and 1107, and 1108 and 1109 from Figure 11 are denoted as scheduled elements 1204, 1205, 1206, 1207,1208, and 1209; and new elements 1216-1217 for the 1-to-4 communication instance, 1214-1215 for the 5-to-2 communication instance, 1212-1213 for the 3-to-1 communication instance, 1220-1221 for the 3-to-2 communication instance, 1222-1223 for the 5-to-1 communication instance, 1226-1227 for the 2-to-1 communication

LINKPLAY EXHIBIT 1004
Page 3320 of 4822

instance, 1224-1225 for the 3-to-4 communication instance, 1228-1229 for the 3-to-3 communication instance, 1230-1231 for the 4-to-1 communication instance, 1232-1233 for the 3-to-4 communication instance, 1234-1235 for the 5-to-1 communication instance, and 1238-1239 for the 3-to-5 communication instance are shown. Three of the interactions are shown without drawing element identifiers (the 1-to-5 communication having a duration of 2, the 5-to-3 communication having a duration of 1, and the 3-to-5 communication having a duration of 4.

Note that transmitter 3, the critical resource, is booked solid until Time = 104, a criteria which must be met in order to reach the theoretical minimum scheduling time. Note also that receiver 1, the second most critical resource, was booked solid until Time = 94. While this logic doesn't always produce the theoretically optimum schedule, it is designed to be efficient with the resources having the highest demands.

If the scheduling process had applied to a 100 unit window of time increment of time rather than a 150 unit window of time increment , the very last request would have been precluded from being scheduled in this window of time increment , but would have been queued until the scheduling process was resumed for the next time window of time increment .

The above example showed a list of requests where all of the requests had priority levels of 1. In one embodiment, four different priority levels are supported. All of the highest level requests are scheduled according to the logic before any of the other priority requests are even considered for scheduling.

Once all of the guaranteed requests have been scheduled, connection requests from the other three priorities are scheduled around the guaranteed requests already in the schedule. When there are already connections in the schedule, the process of calculating the window of time increment size is a little more complex, but the logic is essentially the same. Though the ACR value (or SAV) is still the predominant factor in deciding what gets scheduled when, connection requests between the same two concurrent resources are selected based on priority. This means, for example, that the ACR logic may elect to schedule a low priority 4-to-5 communication instance request before it schedules a medium priority 4-to-2 communication instance request, but never before it schedules a medium priority 4-to-5 communication instance request.

LINKPLAY EXHIBIT 1004
Page 3321 of 4822

Since there are requirements for certain connections to happen at a particular time rather than simply in some window of time increment of time, accommodations can be made for having these requests bypass scheduling. These fixed time connections are entered into the Event Table before scheduling begins and the scheduled connections must be worked in around the fixed connections. These fixed time connections obviously have the highest priority of any of the requests.

In the above example, advancing time was simply a matter of finding the next chronological finish of an already scheduled connection. However, the process of advancing time consumes cycles which might be better spent performing other functions. The network scheduler in the preferred embodiment uses a minimum time increment of 256 units (with a scheduling window of time increment of over 25,000 units rather than the 150 unit window of time increment described above) to reduce the processing requirements. While this does have an obvious negative impact on scheduling efficiencies, there is also a less obvious positive impact. On the average, more resources are candidates for scheduling at each time increment as a result of having processed more finishes between increments. With more resources contending simultaneously, those resources with higher critical impact values are less likely to be preempted by some other resource which had simply finished earlier. The end result of this compromise is that compute cycles are saved, but bandwidth efficiencies aren't substantially impacted.

In the above example, all requests were scheduled in their entirety - there was no need to break up the requests. Had any one of the requests been of a length longer than the defined window of time increment (150 units of time in the above example), however, it would have been segmented into at least two portions to enable scheduling. If a long message cannot be avoided by design, by breaking up the message and using a low priority attribute in this situation, long lower priority messages are less likely to preclude higher priority messages contending for the same resources.

In the preferred embodiment, requests greater than 1023 units are broken into 1000 unit "full" blocks and an optional "partial" block up to 1023 units in length.

In the preferred embodiment, the Itineraries (the particular messages used to convey the schedule) are themselves scheduled messages due to their varying, and potentially large,

LINKPLAY EXHIBIT 1004
Page 3322 of 4822

size. The Agendas (the messages used to collect the connection requests), however, are small and handled as fixed time connections (they are not literally scheduled). The scheduling pipeline used in the preferred embodiment involves collecting the agendas and scheduling the Itineraries in the first window of time increment, where the second, third, and fourth window of time increments are used as described (for scheduling the messages, conveying the schedule, and implementing the connections, respectively).

Turning now to Figure 13, a more detailed representation of components respective to a member node 107, 108 is depicted. While Figure 13 reprises a number of elements of Figures 1 and 5, details of logic types are shown enabling concurrent transmission and receipt of messages respective to the other nodes. In this regard, Transmit Logic 1301 and Receive Logic 1302 are shown for the Master Scheduler 106, with companion Transmit Logic 1319 and Receive Logic 1320 on member node 107 and further companion Transmit Logic 1321 and Receive Logic 1322 on member node 108. Pathways 1311, 1305, and 1306 show vectored message paths outbound from message carrier 100 (with crossbar switch 516). Such messaging approaches are achievable with a context cycling computer such as has previously been discussed, enabling a particular node 107, 108, 106 to be engaged in highly interleaved transmission and receipt of messages. Further detail showing the local scheduler 1307, oscillator 1308, CPU 1309, and VCXO (Voltage Controlled Crystal Oscillator) 1309 is shown for node 107 along with the previously discussed local task list 1303 and local task schedule 1304 which have also been discussed in general.

Turning now to Figure 14, focus is directed to three technologies which, when combined, illustrate a unique and valuable abstracted characterization of considerations in networking distributed, real-time computers to further put the nature of the present invention into full perspective. In this regard key technologies relate to Tight Synchronization between computers; Time Deterministic operation, including network communications; and Centralized Scheduling of distributed tasks, particularly those involving network communications. This diagram shows the interrelationships 1400 between the various technologies implemented in the preferred embodiment, with emphasis on those technologies considered critical to the implementation. The fundamental technologies which, when combined, provide an architecture amenable to real time computing in a distributed computing environment are (1) Time Deterministic Event Control 1422; (2) Tight Synchronization 1423 between systems; and (3) Centralized Scheduling 1421 of the tasks critical to the operation of the real time systems in the distributed environment.

LINKPLAY EXHIBIT 1004
Page 3323 of 4822

The relationships between the technologies are shown using arrows, where the arrowheads point from the enabling technology to the enabled technology, and where the term "enabling" is loosely used here to describe an improvement in the utility of that technology, or in the simplicity of applying the technology. Note that many of the technologies are mutually beneficial, as depicted with a double headed arrow.

The following table explains the relationships between the implementation characteristics as virtual linkages of Figure 14 between the features and goals of tight synchronization 1423, prioritization 1424, time deterministic event control 1422, security 1426, limit checking 1425, centralized scheduling 1421, physical circuit switching 1434, non-blocking tree topology 1429, reliability 1428, active redundancy 1427, and low cost 1435 are interrelated in an abstracted characterization.

Table 33 Interactions of key technical features

Connection In drawing	Quality
1401	Centralized scheduling assures that all network interactions occur at prescribed times. The first requirement for Time Determinism, by this description, is having the nodes on the network in tight synchronization. None of the network nodes can know the needs of the entire network, so in the absence of the central scheduler, the alternatives are to either remove all dynamics or to implement a solution tolerant of the variability. By incorporating centralized connection scheduling, prioritization, and limit checking, the (high priority) time critical interconnections can be time deterministic while the (low priority) high volume interconnections can exist through variability tolerant designs.
1402	In a distributed computing environment, deterministic networking can be achieved either with tightly synchronized systems, or by providing enough excess bandwidth to realize timely delivery in a worst case timing and bandwidth demand situation. Because such an analysis of worst case situation is difficult and because providing the excess bandwidth can be costly, synchronization is an enabling technology for Time

LINKPLAY EXHIBIT 1004
Page 3324 of 4822

	Determinism. Additionally, the difficulty in achieving a desired level of synchronization is usually proportional to the extent that the communications are indeterminate.
1403	Active Redundancy enables Time Determinism primarily in regards to failure recovery. It is difficult to manage the possible failure scenarios in the absence of redundancy, yet it is also difficult to manage failure recovery in a Time Deterministic system with passive redundancy. Time Determinism facilitates the use of Active Redundancy in that one of the biggest challenges to active redundancy is in assuring that all sides operate identically. Furthermore, with a Time Deterministic environment, a redundant element of a system can be diagnosed as dysfunctional simply by noting that the appropriate actions aren't occurring at the prescribed times. The time deterministic network involves communicating based on time between separate computing elements. The fact that the communications are scheduled based on time facilitates the employment of active redundancy. In active redundancy, there are two or more computers which are each executing identical logics via its own sequence of program instructions, exchanging the results of this execution, and then arbitrating to come to a result. Due to the fact that the data received by each computer must be shared with the other computer(s) executing the same logic prior to arbitrating this data, there must be some mechanism by which the time to begin the arbitration process in determined. The mechanism which has been chosen is to provide time synchronization between the redundant computers and schedule the tasks related to the communications based on time. In this way, there is known the time at which the arbitration task can begin.
1404	One of the most difficult challenges in implementing active redundancy is in handling the variation between sides. All sides of a redundant system must be expected to match at defined checkpoints, and discrepancies must be resolved in order to retain the value in redundancy. All sides must know when a checkpoint has been reached, and rules must be defined to allow differences to be resolved. With centralized scheduling,

LINKPLAY EXHIBIT 1004
Page 3325 of 4822

	variations in connection requests, either in content or in timing, can be reduced or even eliminated, thereby increasing the probability that all sides can maintain operational equivalence. On the other hand, centralized scheduling places a tremendous dependency on the availability of that scheduling element. Centralized scheduling becomes much more viable in the presence of redundancy.
1405	Centralized scheduling requires that all connection requests be assembled together for analysis. This permits limit checking to verify that (1) certain undesirable node-to-node combinations are prohibited and (2) traffic levels (possibly at a particular priority) between certain combinations of nodes doesn't exceed predefined levels.
1406	Limit checking enhances the security of the distributed computing system by assuring that certain restrictions aren't violated. These limit checks help against malicious attacks where certain systems shouldn't be allowed to interact with others, and against accidental complications such as overloading a system with legitimate low priority data or unintentional overloading of high priority data.
1407	Prioritization of the messages enhances the utility of the limit checking. Not only can the highest priority data be made time deterministic without excessively restricting the lower priority messages, but security measures can be put in place which take message priority into account.
1408	Limit checking improves the overall reliability of the system by preventing certain security risks and by retaining the time determinism. In the absence of limit checking, certain network faults could cripple the network, such as when a node were to accidentally request an excessive amount of traffic at the highest priority level.
1409	In the implementation of Limit Checking, the highest priority traffic is testing against throughput thresholds to assure that no errors on one channel preclude the highest priority messaging to take place between any nodes not including the node with errors.
1410	Non-blocking network architectures allow the network to realize throughputs higher than the throughput available to any particular node on the network. This translates to higher bandwidth at any particular

LINKPLAY EXHIBIT 1004
Page 3326 of 4822

	price when compared to blocking architectures.
1411	Security can often be implemented more easily when a centralized security agent can be entrusted with the administration of the security policies. Specifically in the preferred embodiment, the central scheduler is given the security tables and the interconnecting systems can delegate the security task to the central authority, allowing the administration to be done in one place rather than in two. Furthermore, the same security policies don't have to be re-implemented in as many environments as can exist on the star.
1412	By introducing a centralized scheduler, the scheduling logic can be implemented so as to enforce prioritization on the received connection requests.
1413	Prioritization simplifies the task of implementing Time Determinism. This is because large but infrequent tasks such as program loading can be set to operate at a low priority, eliminating the analysis of this high impact operation would have on the simpler yet more critical task of peer to peer communications.
1415	On networks employing a broadcast protocol, all network nodes have the opportunity to view the data going to all other nodes (though encryption can make this viewing rather futile). Blocking protocols aren't always implemented in broadcast protocols, but non-blocking protocols preclude the possibility of broadcasting, thereby avoiding the obvious security risks.
1416	One of the obvious ways of achieving reliability is in the introduction of redundancy.
1417	High throughput, the primary objective of high bandwidth, is positively impacted in two ways through the introduction of prioritization. First, there is the perception of high throughput in that the data that has the most value is delivered in a more timely fashion by invoking prioritization. Secondly, higher volumes of data can be permitted knowing that the high priority data, which often must be delivered in a timely fashion, won't be sacrificed by permitting large volumes of low priority traffic to be requested.
1418	Active Redundancy requires that a failed element be capable of

LINKPLAY EXHIBIT 1004
Page 3327 of 4822

	recovering from downtime. This often involves the exchange of historical data from the elements which did not fail. In order to minimize the impact of this exchange, prioritization can be introduced assuring that the recovering element can recover and that it can do so without a substantial detrimental impact on the healthy elements.
1419	Certain architectures and topologies are inherently at risk of network outages in the presence of system outages. Use of a non-blocking architecture and physical circuit switching, while not the only means to achieve a higher reliability, offers a level of isolation from network node faults.
1420	Centralized scheduling is an enabling technology for effectively using the bandwidth available to non-blocking networking architectures such as those implementing certain crossbar switches. In the absence of scheduling, fixed connection patterns must be implemented, thereby reducing the practical throughput available to the nodes.
1431	Physical Circuit Switching technology can be difficult to implement for several reasons, including challenges associated with determining what circuits should be put in place at any particular time. By knowing the needs ahead of time, the connection patterns for the circuit switch can be determined a priori and implemented as needed.
1432	Circuit switching of systems configured in a tree topology can be implemented with a separate circuit, or pair of circuits, for each system on the network. This allows each channel to have its own path for communications, permitting a non-blocking architecture.
1433	Because physical circuit switching can reduce the requirement for active circuitry, high line speeds can often lead to higher network bandwidths.
1436	The simplicity of the circuitry associated with physical circuit switching often permits a given set of capabilities to be implemented at a lower cost.

Turning now to Figure 15, the general process 1500 executed respective to scheduling in each member node 108, 107 (any one of control computer 506, historian computer 507, control computer 504, historian computer 505, user interface 503, network support 502, and network switch 501) is discussed. Each member node 108, 107 is responsible for receiving

the common clock signal; synchronizing to that common clock signal; receiving the local event table 114, 116 from the master scheduler 106 in network switch 501; formulating the task execution list based on the received local event table 114, 116 and merging in any additional tasks which are not described in the local event table 114, 116; executing the task execution list; generating the request list of communications; and transmitting the request list of communications to the master scheduler 106 in network switch 501.

Steps 1501 and 1502 - Since the network is so highly dependent on a global time base, the first step 1501 at a member node 108, 107 is to receive the common clock signal. The clock signal is communicated as a message from the master scheduler 106 in network switch 501 to each of the member nodes 108, 107. The member node 108, 107 receives this clock signal in step 1501, stamps the actual time of reception, and validates the integrity of this message. The validated message is used as an input to the process of synchronization and, in step 1502, the difference between the expected and actual times of time message receipt is optionally determined (optionally if the nature of the time message is a pulse instead of a data bearing message).

Steps 1503 and 1504 - Each member node 108, 107 must now synchronize to the common clock signal. The process of performing this synchronization can be broken up into two separate tasks: first, in step 1503, the adjustment of a voltage controlled oscillator (1308) to a frequency which matches the source of the common clock signal and second, in step 1504, the alignment of the time such that the start of each time period which is defined by the common clock signal occurs at the same time in the member nodes 108, 107 as it does in the master scheduler 106 in network switch 501.

Step 1505 - Once the synchronization of a member node 108, 107 to the master scheduler 106 in network switch 501 is complete, the member node 108, 107 must receive the local event table 114, 116 from the master scheduler 106 in network switch 501. The local event table 114, 116 contains the schedule for the communications which are involving this member node. Note that the reception of the local event table 114, 116 itself is scheduled by the master scheduler 106 in network switch 501. Therefore, the first time that the node is to receive the local event table 114, 116 it must be available to receive that message at any possible time that the local event table 114, 116 might be received.

LINKPLAY EXHIBIT 1004
Page 3329 of 4822

Step 1506 - The next step for the member node 108, 107 is to formulate the task execution list. The task execution list is formulated from the local event table 114, 116 and the additional tasks which the member node 108, 107 must do which are not described in the local event table 114, 116. The local event table 114, 116 which is communicated from the master scheduler 106 in network switch 501 provides the timing associated with the transmission and reception tasks. In order to formulate the task execution list, the following sub-steps must be taken:

Substep 1506A: Adjust the start time received from the master scheduler 106 in network switch 501 to account for the propagation delay. The timing which is received from the master scheduler 106 in network switch 501 is relative to the master scheduler 106 in network switch 501. Although the member node 108, 107 is synchronized with the master scheduler 106 in network switch 501, there is still an inherent delay in the propagation through the media being communicated. Either this delay must be accounted for by the master scheduler 106 in network switch 501, the member node, or the communication window of time increment must be sufficiently large that the inherent delay is included as overhead in the time allotted to the connection. In the preferred embodiment, this delay is accounted for in the member node.

Substep 1506B: Schedule the tasks which are dependent on the communication tasks. There are tasks which must be scheduled prior to a transmission (for example, the transfer of the data to be transmitted from memory); there are also tasks which must be scheduled after a reception (for example, the arbitration of the data between the multiple, different sources in a redundant computing environment and the transfer of data to memory). The member node 108, 107 is responsible for the scheduling of these tasks.

Substep 1506C: Schedule the tasks which are neither described by the local event table 114, 116 nor are dependent on the communications. There are tasks which the member node 108, 107 must accomplish in addition to communication related tasks. These tasks may be for the operation of other I/O, localized display, etc.

Step 1507 - The next step for the member node 108, 107 is to execute the task execution list. Each task in the list contains the task which is to be executed, the start time for the task, and any parameters which are necessary in order to successfully complete the task.

LINKPLAY EXHIBIT 1004
Page 3330 of 4822

Step 1508 - The member node 108, 107 can now formulate the communication request list, which is a list of the member nodes 108, 107 to which this member node 108, 107 is desiring to communicate. The information which is included in this communication request list is the source node identifier, destination node identifier, and the total length of the communication.

Step 1509 - This communication request list is communicated to the master scheduler 106 in network switch 501 for consideration in its scheduling. At this point, the member node 108, 107 has performed all of the necessary functions in a particular time period. The next step 1510 is to determine whether or not it is time to receive the next reference moment message. If it is, then the entire process repeats itself starting with the very first step 1501 of receiving the reference moment message. If it is not time to receive the next reference moment message, then the logic of the member node may forward to step 1505.

As one contemplated embodiment, Figure 16 presents the general data flow diagram 130 of Figure 1 as a data flow diagram 1600 where communications are effected via radio frequency means and transmissions 1601, 1602, and 1603 are effected via radio frequency transmitters and receivers.

The present invention is enabled through use of machines which are considered as a special purpose computers created in one embodiment by combining a computer with computer program code (alternatively termed, for example, software or logic, where logic denotes software in and for execution by an enabling system of electrical circuitry) directed to the particular function of process control so that, when connected to an electrical power source, powered by electricity, and activated to execute its computer program code (software), the special purpose computer executes the particular function related to the present invention pursuant to the instructions from the computer program code. The computer has electrically activated components denoted as a central processing unit, at least one physical memory connected to the central processing unit, an optional input keyboard connected to the central processing unit, an optional output display (which can include, without limitation, a printer, printing terminal where output is printed on paper, cathode ray tube monitor, and/or flat panel monitor) connected to the central processing unit, a computer clock pulse generator, and a connection to electrical power for energizing all of the electrically activated components of the computer. The central processor further has a control bus and specific computer circuits for either temporarily holding data (for example, a register or an

LINKPLAY EXHIBIT 1004
Page 3331 of 4822

accumulator) or for executing a fundamental data processing operation (for example, an addition circuit); the specific computer circuits are connected in communication to the control bus through latching circuits which can individually be in either a conducting (communicating) or non-conducting (non-communicating) configuration; the collective configurations of all the latching circuits at a particular moment define a particular state of the central processor; and the state of the central processor is respectively modified and changed as the individual instructions of the computer program code are sequentially accessed by the central processing unit and output from the clock pulse generator is directed to the latches. The computer optionally also has computer program code in the form of a commercially available operating system which functions as a computer program code platform for enabling the execution of the special purpose computer program code directed to the particular function of the present invention. In overview, the present invention in this embodiment is implemented in the preferred embodiment by and through a plurality of computers which have been programmed to contribute to technology by performing the specifically useful purpose and practical application of the described system; each computer functions as a medium for realizing the functionality provided by the computer program code's functionality. In some embodiments, some communication linkages within an extended form of the special purpose computer may be implemented by electrical, optical, or radio-frequency signal communication interfaces and communication media which collectively use electrically-powered transmitter and receiver components which are directly or indirectly linked to at least one central processing unit.

The computer program code resident within a computer provides at least one component part of that machine. The computer program code may be in the form of, without limitation, inter-linked modules, routines and subroutines, program objects, and/or installed interactive processes. In some emergent computer designs, enabled by modern application specific integrated circuitry (ASIC) technology, computer program code may be at least partially expressed in hardwired circuitry reminiscent of traditional hardwired gate arrangements which execute simplified logical scenarios without need for a clock pulse. The Figures of this description show a number structural relationships between key modules and components within the computer program code and other aspects of the invention which enable the functionality of the machine of the present invention to be realized; in this regard, the computing elements in the system are constructed out of a combination of hardware and software.

LINKPLAY EXHIBIT 1004
Page 3332 of 4822

The present invention has been described in an illustrative manner. In this regard, it is evident that those skilled in the art, once given the benefit of the foregoing disclosure, may now make modifications to the specific embodiment described herein without departing form the sprit of the present invention. Such modifications are to be considered within the scope of the present invention which is limited solely by the scope and spirit of the appended claims.

LINKPLAY EXHIBIT 1004
Page 3333 of 4822

We claim:

1. A computer implemented method for enabling a plurality of computer nodes in mutual data communication to execute logic in a coordinated manner, comprising:

aligning real-time time between said plurality of computer nodes so that each computer node has essentially the same real-time time;

defining a schedule of communications for each computer node, said schedule having a different time in the future defined for initiating each respective communication;

communicating each schedule to its respective computer node; and

initiating each communication from each node at the time defined in its schedule.

2. A computer network architecture, comprising:

a message carrier supporting multiple communication paths;

a plurality of computers coupled to said message carrier, said computers being programmed to perform time-deterministic tasks by cycling through a plurality of different operating states, including a communication state;

a clock system accessible by all said computers for maintaining time synchronism among said computers; and

a scheduler coupled to said message carrier for determining the sequence of communication between said computers by causing said message carrier to establish communication between selected computers based on said time-deterministic tasks being performed by said computers.

LINKPLAY EXHIBIT 1004
Page 3334 of 4822

3. A computer implemented method for enabling a computer node in a plurality of computer nodes to execute logic in coordination with the other nodes in said plurality of computer nodes, comprising:

connecting a message carrier to each computer node in said plurality of computer nodes;

providing master scheduling capability in one computer node to establish a master scheduler;

defining, in each computer node, a communication request list of each communication needed by said computer node with any other computer node respective to a future time period;

sending said communication request list from each computer node to said master scheduler via said message carrier;

receiving, in each computer node, a communication schedule from said master scheduler via said message carrier;

defining, in each computer node, a local task schedule from said communication schedule.

4. A computer implemented method for enabling a plurality of member nodes to execute logic in a coordinated manner, comprising:

connecting a message carrier to each member node in said plurality of member nodes;

providing a master scheduler in data communication with each said member node via said message carrier;

defining a reference moment time value for said plurality of member nodes;

LINKPLAY EXHIBIT 1004
Page 3335 of 4822

communicating said reference moment time value to each member node and said master scheduler via said message carrier;

defining a communication request list in each said member node showing each communication needed by said member node with any other member node in a future time period;

communicating said communication request list from each member node to said master scheduler via said message carrier;

defining, in said master scheduler, scheduled communications between member nodes at communication event times respective to said reference moment time value and also respective to each said communication listed in each said received communication request list;

defining, in said master scheduler, a communication schedule for each member node respective to said scheduled communications ;

sending each said communication schedule from said master scheduler to each respective member node via said message carrier;

defining, in each member node, the difference in time between said reference moment time value and the time when said reference moment time value is communicated into said member node;

defining, in each member node, a local task schedule from said communication schedule and from said difference in time for the member node so that the member node will communicate to other member nodes via said message carrier at said scheduled communication event times;

executing said scheduled communications between said member nodes respective to each scheduled communication via said message carrier according to said respective local task schedules.

LINKPLAY EXHIBIT 1004
Page 3336 of 4822

5. The method of Claim 4 wherein said step of defining a reference moment time value is performed by said master scheduler.

6. The method of Claim 4 wherein said step of defining a communication request list in each said member node limits said communications to a maximum number in said future time period.

7. The method of Claim 4 wherein each communication has a duration time attribute and the value of said duration time attribute is less than a fixed maximum value.

8. The method of Claim 4 wherein the step defining a reference moment time value for said plurality of member nodes is done by a global time system.

9. A computer implemented method for enabling a plurality of member nodes to execute logic in a coordinated manner, comprising:

connecting a message carrier to each member node in said plurality of member nodes, each node having a central processing unit;

providing a master scheduler in data communication with each said member node via said message carrier;

defining a reference moment time value as a moment in time for said plurality of member nodes;

communicating said reference moment time value to each member node and said master scheduler via said message carrier;

defining into a communication request list in each said member node, each access to any other member node needed by said member node in achieving a future task increment respective to a future time period;

communicating said communication request list from each said member node to said master scheduler via said message carrier;

LINKPLAY EXHIBIT 1004
Page 3337 of 4822

defining, in said master scheduler, a communication event data object respective to each said access to any other member node needed by said member node in achieving a future task increment, said communication event data object having an accessing member identification attribute value, an accessed member identification attribute value, and an access duration attribute value;

collecting, in said master scheduler, all defined communication event data objects into a communication event set;

determining in said master scheduler a summed accessed value for each member node in said communication event set as a sum of all access duration attribute values in all communication event data objects having a accessed member identification attribute value respective to said member node;

determining in said master scheduler a summed accessing value for each member node in said communication event set as a sum of all access duration attribute values in all communication event data objects having an accessing member identification attribute value respective to said member node;

scheduling in said master scheduler and into the earliest available time in a schedule, a future communication task respective to said reference moment time value between the accessing member node having the highest said summed accessing value respective to said communication event set and the accessed member node respective to the accessed member identification attribute value in the communication event data object having the highest affiliated access duration attribute value for all communication event data objects respective to said accessing member node, said future communication task having a task initiation time equal to the earliest available time in the schedule, an accessing member node identifier attribute respective to the accessing member, and an accessed member node identifier attribute respective to the accessed member node;

LINKPLAY EXHIBIT 1004
Page 3338 of 4822

removing, in said master scheduler, the communication event data object respective to the accessed member node identifier attribute in the scheduled future communication task and the accessed member node identifier attribute in the scheduled future communication task to redefine the communication event set to those communication event data objects which have not been scheduled; and

repeating, in said master scheduler, respective to the redefined communication event set, the steps of determining a summed accessed value, determining a summed accessing value, scheduling a future communication task, and removing all communication event data objects respective to scheduled member nodes until all communication event data objects have been scheduled into a communication schedule for each member node;

sending each said communication schedule from said master scheduler to each respective member node via said message carrier;

defining, in each member node, the difference in time between said reference moment time value and the time when said reference moment time value is communicated into said member node;

defining, in each member node, a local task schedule from said communication schedule and from said difference in time for the member node so that the member node will communicate to other member nodes via said message carrier at said scheduled communication event times; and

executing said scheduled communications between member nodes respective to each scheduled communication event via said message carrier according to said respective local task schedules.

10. A computer implemented method for scheduling communications between member objects in a plurality of member objects functioning as a collective group, comprising the steps of:

providing a computer;

defining in said computer, for each member object, a communication event data object respective to each access to any other member object needed by said member object in achieving a future task increment, said communication event data object having an accessing member identification attribute value, an accessed member identification attribute value, and an access duration attribute value;

collecting, in said computer, all defined communication event data objects into a communication event set;

determining, in said computer, a summed accessed value for each member object in said communication event set as a sum of all access duration attribute values in all communication event data objects having a accessed member identification attribute value respective to said member object;

determining, in said computer, a summed accessing value for each member object in said communication event set as a sum of all access duration attribute values in all communication event data objects having an accessing member identification attribute value respective to said member object;

defining a set of sequential future time increments within a scheduling period respective to a schedule;

scheduling, in said computer, into the earliest available time in the first of said future time increments, a future communication task based on the first available communication event data object in said communication event set between

the accessing member object having the highest said summed accessing value respective to said communication event set and

the accessed member object having the highest said summed accessed value respective to said communication event set if a communication event data object exists between said accessed member object and said accessing member object,

said future communication task having a task initiation time equal to the earliest available time in the schedule, an accessing member object identifier attribute respective to the accessing member object, an accessed member object identifier attribute respective to the accessed member object, and said access duration attribute value;

removing from scheduling candidacy, in said computer, the communication event data object respective to the accessed member object identifier attribute in the scheduled future communication task and the accessing member object identifier attribute in the scheduled future communication task to redefine the communication event set to those communication event data objects which have not been scheduled;

repeating, in said computer respective to the redefined communication event set, the steps of determining a summed accessed value, determining a summed accessing value, scheduling a future communication task, and removing all communication event data objects respective to scheduled member objects until all communication event data objects have been scheduled which can be scheduled into said first future time increment; and

recurrently executing, in said computer respective to the redefined communication event set and sequentially respective to each of the remaining of said plurality of future time increments, the steps of determining a summed accessed value, determining a summed accessing value, scheduling a future communication task, and removing all communication event data objects respective to scheduled member objects, and repeating such steps until all communication event data objects have been scheduled which can be scheduled into each said future time increment and until all communication event data objects have been scheduled into said scheduling period.

LINKPLAY EXHIBIT 1004
Page 3341 of 4822

11. The method according to 10, wherein said defining is confined to future task increments which are needed within a predefined time period and said scheduling is respective to said predefined time period.

12. The method according to 11, further comprising the step of connecting a message carrier to each member object in said plurality of member objects, said message carrier having a plurality of channels for permitting concurrency in said communication events, said step of scheduling being respective to said plurality of channels as a group wherein said earliest available time is respective to any channel in said message carrier.

13. The method according to Claim 12 wherein each said member object is a member node being in data communication with said message carrier, each member node having a central processing unit.

14. A computer implemented method for scheduling tasks for resource objects in a plurality of resource objects functioning as a collective group, comprising the steps of:

providing a computer;

defining, in said computer, a set of future tasks for said collective group;

defining, in said computer, each resource object needed for each future task in said set of future tasks;

defining, in said computer, each interaction needed between any two resource objects respective to said set of future tasks;

defining, in said computer, a use instance for each said resource object in achieving each said interaction, said use instance having an use duration attribute value;

defining, in said computer, an accessing resource object and an accessed resource object respective to each said use instance;

LINKPLAY EXHIBIT 1004
Page 3342 of 4822

firstly determining, in said computer, a summed accessing time value for each accessing resource object as a sum of all use duration attribute values for all said use instances respective to that resource object when it is the accessing resource object;

secondly determining, in said computer, a summed accessed time value for each accessed resource object as a sum of all use duration attribute values for all said interactions between said accessed resource object and said accessing resource object;

defining, in said computer, a set of sequential future time increments within a scheduling period respective to a schedule;

scheduling, in said computer, into the earliest available time in the first of said future time increments, a future resource interaction event based on the first available future task in said set of future tasks between

the accessing resource object having the highest said summed accessing value respective to said set of future tasks and

the accessed resource object having the highest said summed accessed value respective to said set of future tasks if a use instance exists between said accessed resource object and said accessing resource object,

said future resource interaction event having an initiation time equal to the earliest available time in the schedule, an accessing resource object identifier attribute respective to the accessing resource object, an accessed resource object identifier attribute respective to the accessed resource object, and said use duration attribute value;

removing from scheduling candidacy, in said computer, the use instance respective to the accessed resource object in the scheduled future resource interaction event and the accessing member object identifier attribute in the

LINKPLAY EXHIBIT 1004
Page 3343 of 4822

scheduled future resource interaction event to redefine the set of future tasks to those future tasks for which interactions have not been scheduled;

repeating, in said computer, respective to the redefined future task set, the steps of determining a summed accessed value, determining a summed accessing value, scheduling a future communication task, and removing all use instances respective to scheduled resource objects until all use instances have been scheduled which can be scheduled into said first future time increment; and

recurrently executing, in said computer, respective to the redefined future task set and sequentially respective to each of the remaining of said plurality of future time increments, the steps of determining a summed accessed value, determining a summed accessing value, scheduling a future resource interaction event, removing all resource objects respective to scheduled resource objects, and repeating such steps until all resource interaction events have been scheduled which can be scheduled into each future time increment and until all resource interaction events have been scheduled into said scheduling period.

15. A computer implemented apparatus for enabling a plurality of computer nodes in mutual data communication to execute logic in a coordinated manner, comprising:

means for aligning real-time time between said plurality of computer nodes so that each computer node has essentially the same real-time time;

means for defining a schedule of communications for each computer node, said schedule having a different time in the future defined for initiating each respective communication;

means for communicating each schedule to its respective computer node;

means for initiating each communication from each node at the time defined in its schedule.

LINKPLAY EXHIBIT 1004
Page 3344 of 4822

16. A computer-implemented apparatus having a message carrier, comprising:

a plurality of member nodes, each member node being in data communication with said message carrier, each member node having

a central processing unit,

time phase logic for receiving from said message carrier a reference moment message having a reference moment time value and for defining the difference in time between said reference moment time value and the time of receiving said reference moment message,

communication definition logic for identifying each internodal communication needed by said member node in a future time period and for establishing a communication request list respective to all said internodal communications,

communication request sending logic for communicating said communication request list via said message carrier to a master scheduler,

communication task receiving logic for receiving a communication schedule via said message carrier from said master scheduler,

task integration logic for defining a local task schedule from said communication schedule and said difference in time, and

communication implementation logic for communicating via said message carrier with other member nodes according to said local task schedule; and

a master scheduler in data communication with said plurality of member nodes via said message carrier, said master scheduler

having scheduler logic executed by a central processing unit, said scheduler logic having

reference moment definition logic for defining said reference moment time value as a moment in time,
reference moment communication logic for creating a reference moment message having said reference moment time value and for communicating said reference moment message via said message carrier to each member node,

communication request receiving logic for receiving each said communication request list from each said member node via said message carrier,

communication schedule definition logic for scheduling communications between member nodes at communication event times respective to said reference moment time value and also respective to each said internodal communication listed in each said received communication request list and for further defining a communication schedule for each member node respective to said scheduled communications, and

member node communication schedule response logic for sending each said communication schedule to each respective member node via said message carrier.

17. The apparatus of Claim 16, wherein each member node further comprises a variable speed oscillator.

18. The apparatus of Claim 16, wherein each member node further comprises time alignment logic and oscillator rate adjustment logic.

19. The apparatus of Claim 16, wherein said message carrier comprises a crossbar switch.

20. The apparatus of Claim 16, wherein said message carrier comprises a network topology.

21. The apparatus of Claim 16, wherein said message carrier comprises a radio frequency communications system.

LINKPLAY EXHIBIT 1004
Page 3347 of 4822

SUBSTITUTE SHEET (RULE 26)

FIG. 1

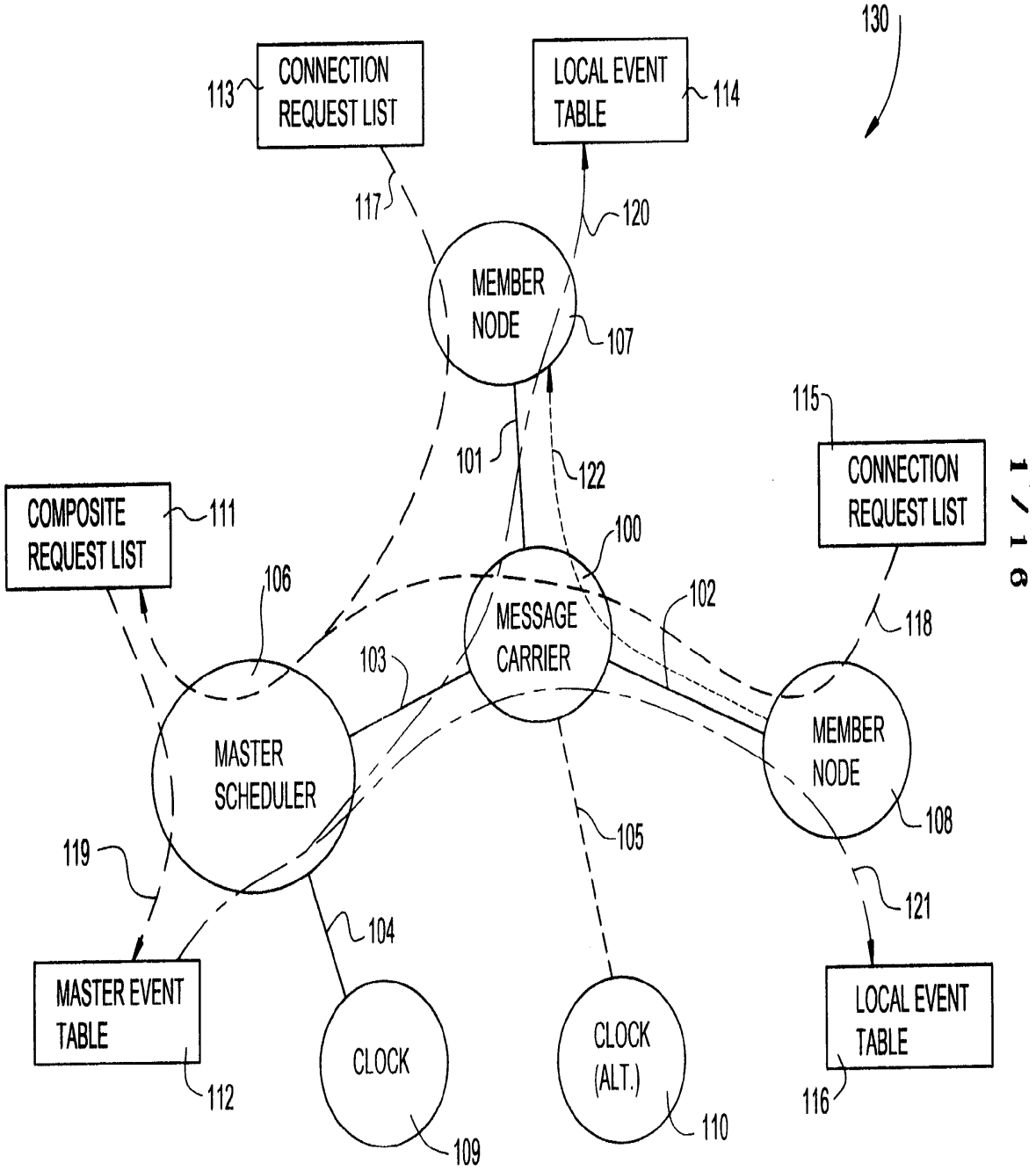
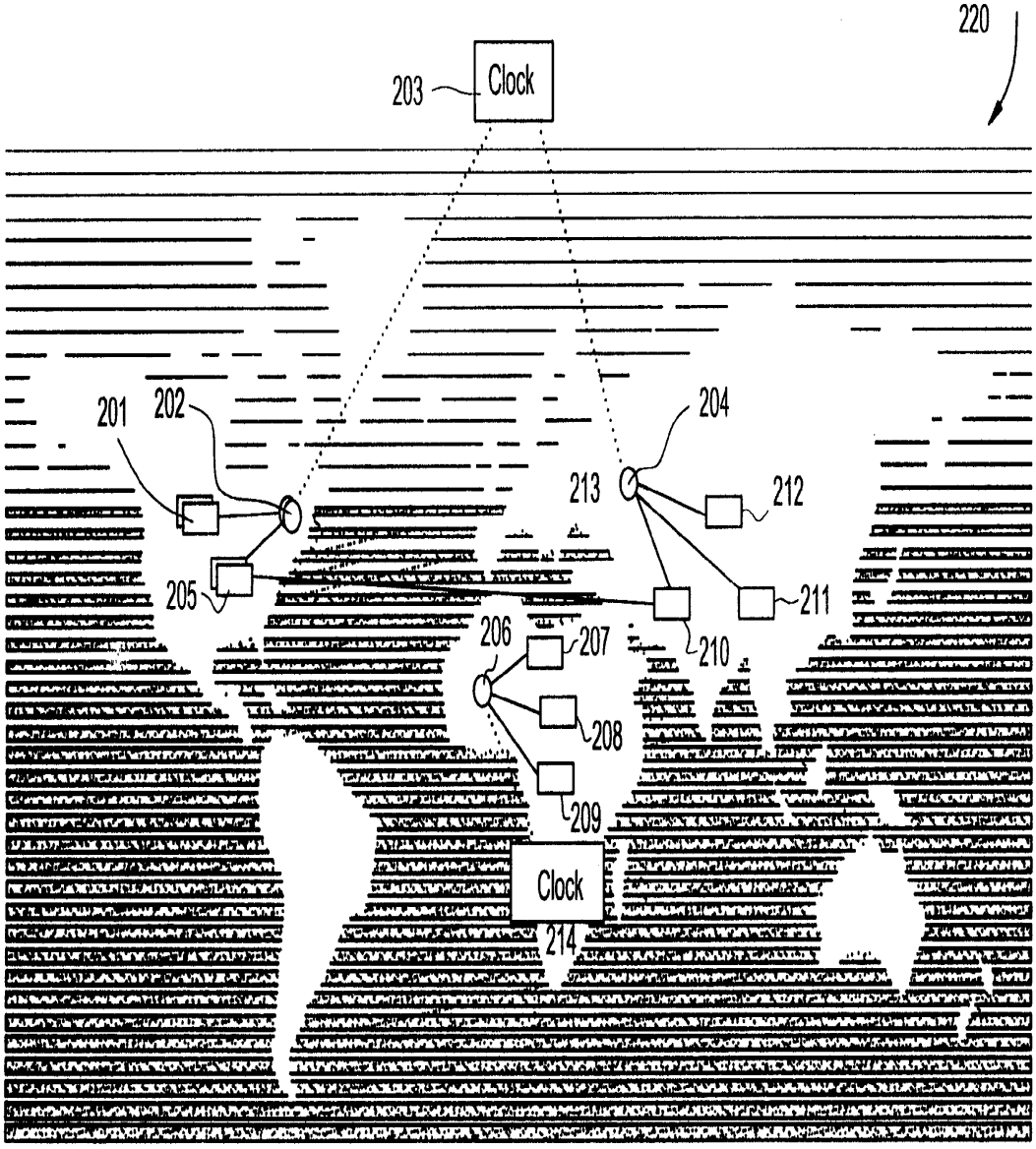


FIG. 2



SUBSTITUTE SHEET (RULE 26)

SUBSTITUTE SHEET (RULE 26)

FIG. 3

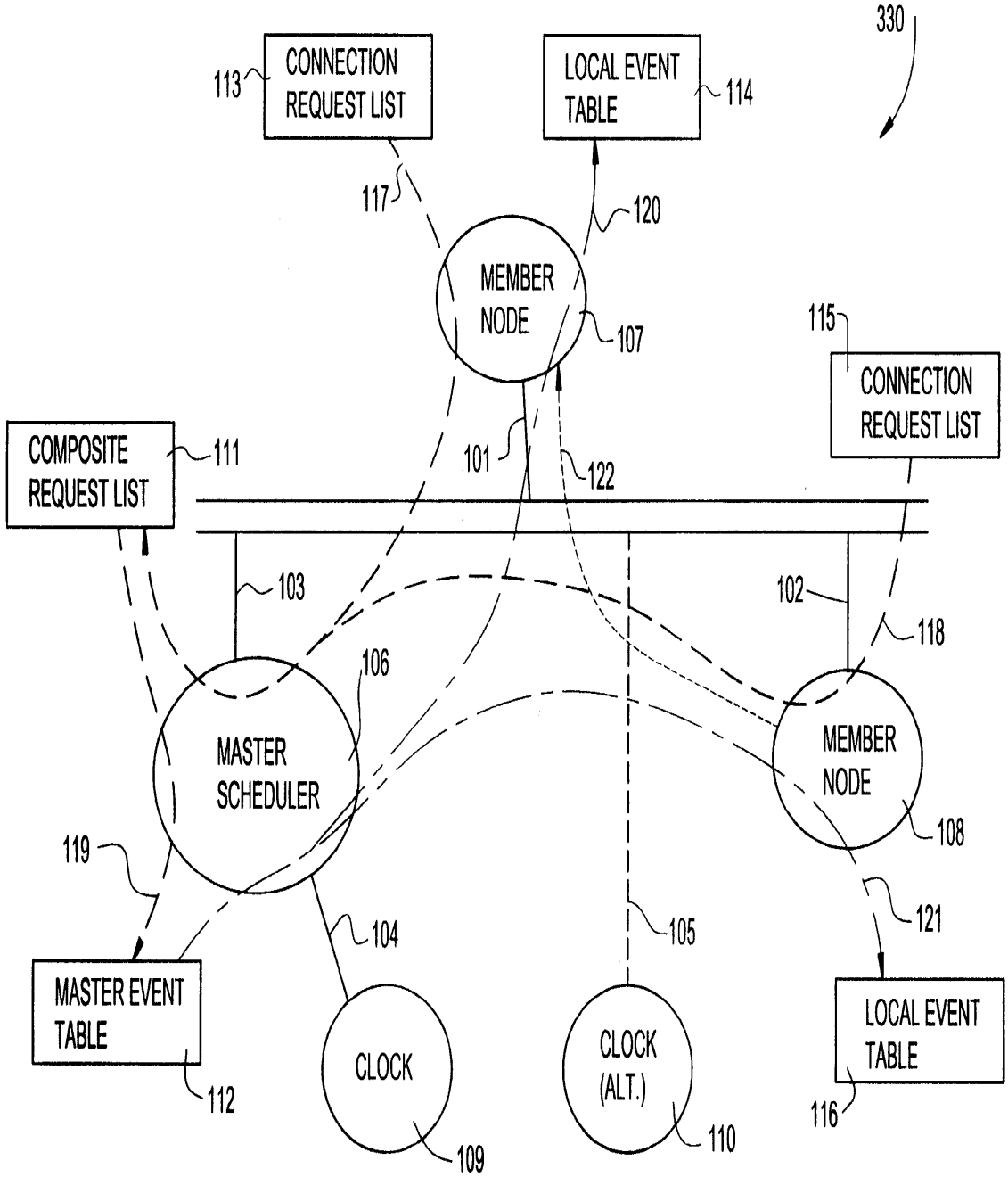
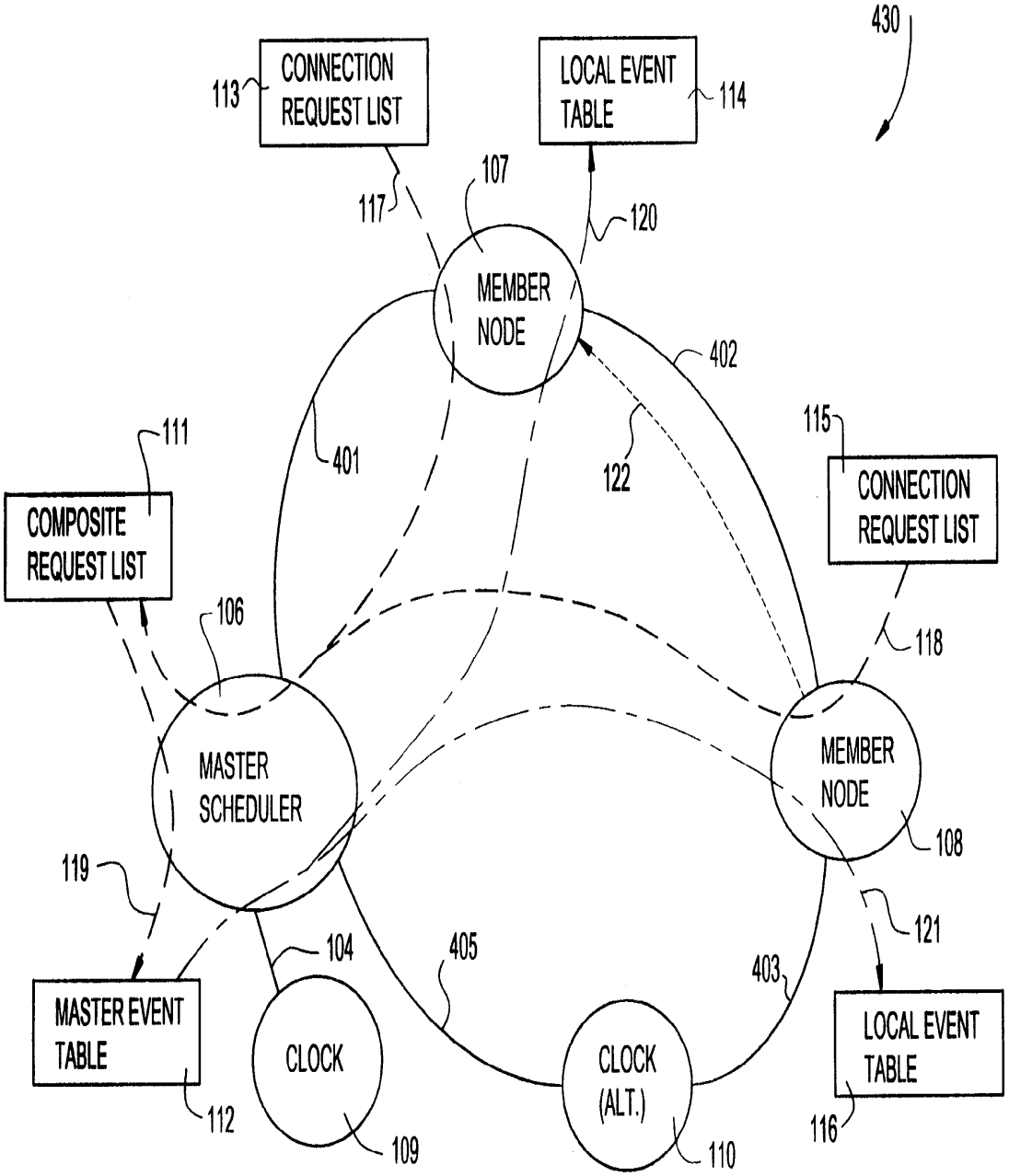


FIG. 4



SUBSTITUTE SHEET (RULE 26)

SUBSTITUTE SHEET (RULE 26)

FIG. 5

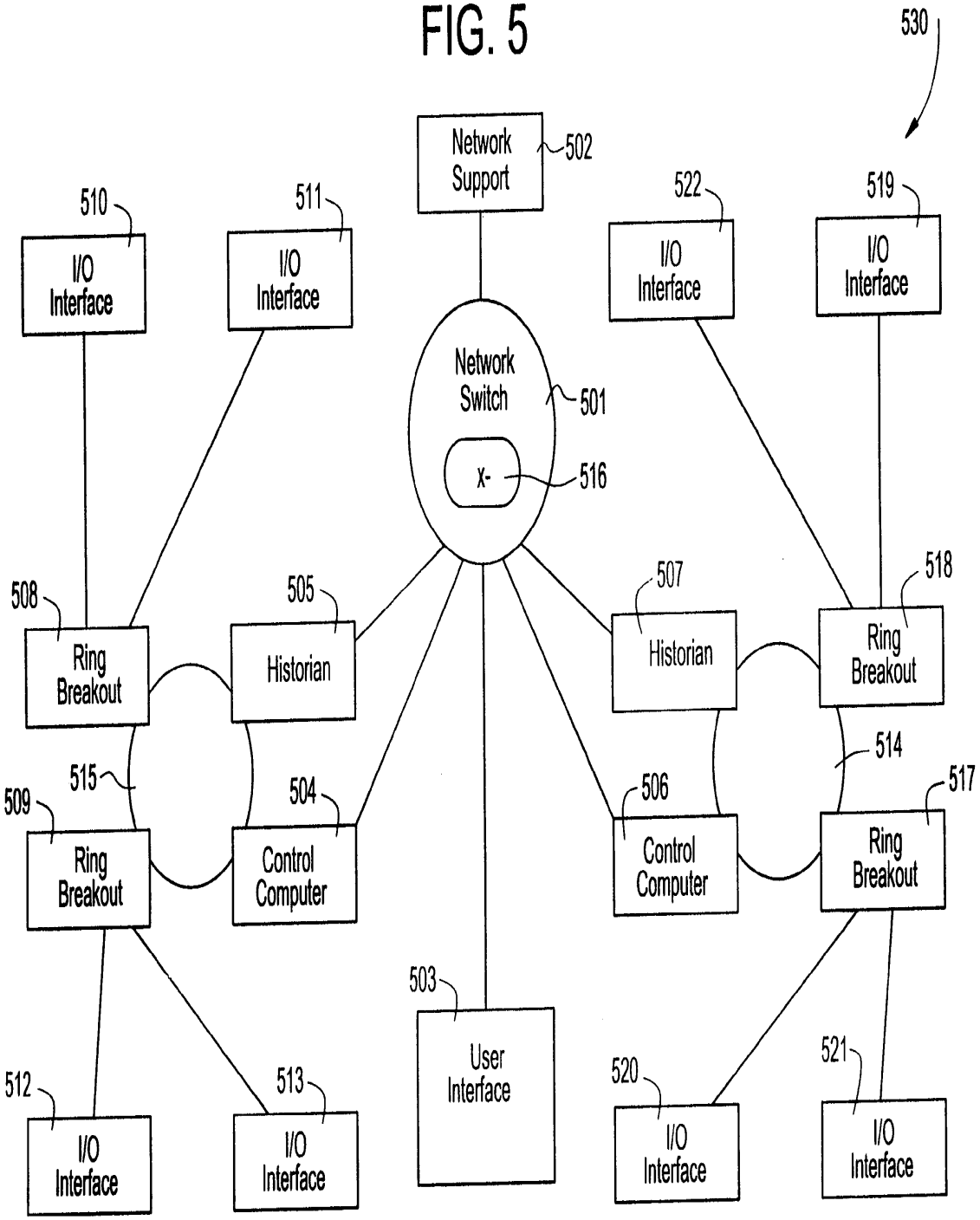


FIG. 6

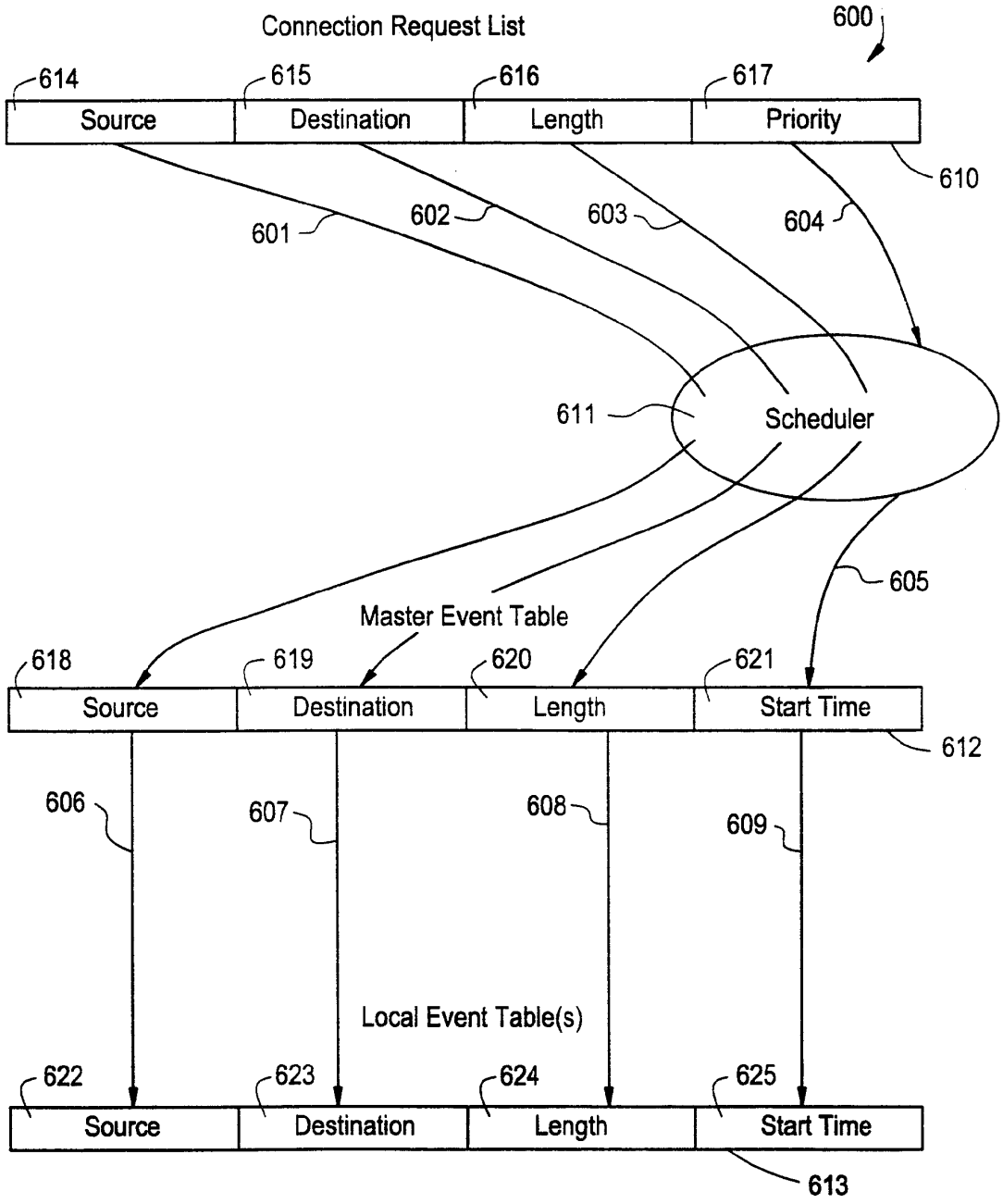

Connection Request List
600
614
615
616
617
Source
Destination
Length
Priority
610
601
602
603
604
611
Scheduler
605
Master Event Table
618
619
620
621
Source
Destination
Length
Start Time
612
606
607
608
609
Local Event Table(s)
622
623
624
625
Source
Destination
Length
Start Time
613

FIG. 7

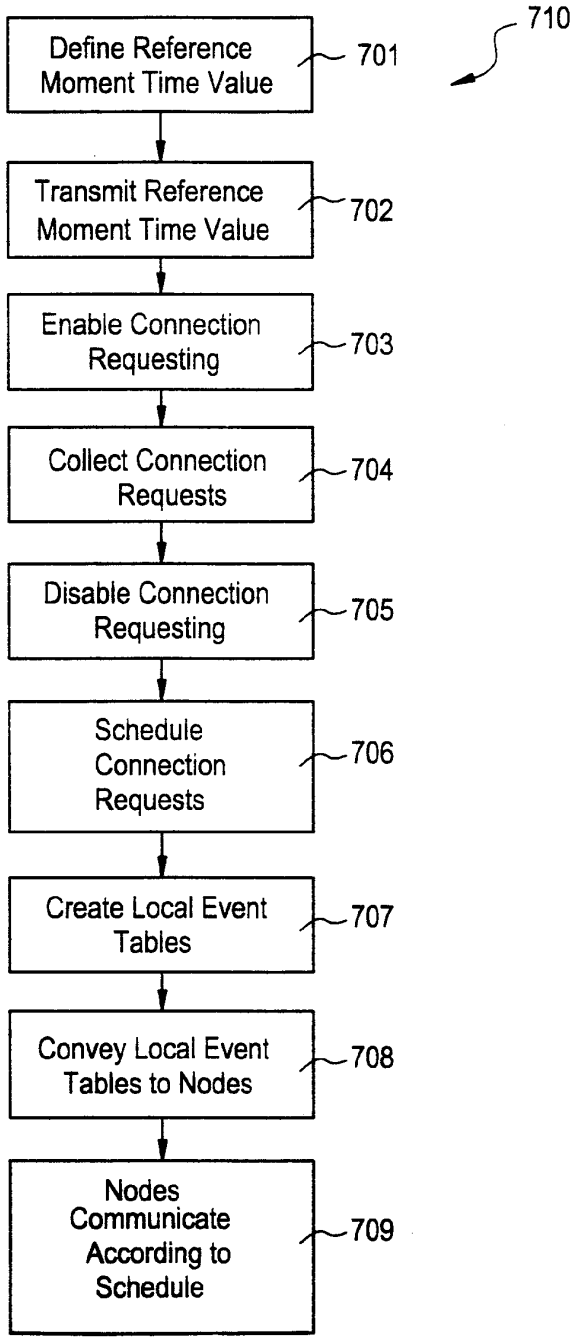

710
Define Reference Moment Time Value
701
Transmit Reference Moment Time Value
702
Enable Connection Requesting
703
Collect Connection Requests
704
Disable Connection Requesting
705
Schedule Connection Requests
706
Create Local Event Tables
707
Convey Local Event Tables to Nodes
708
Nodes Communicate According to Schedule
709

FIG. 8

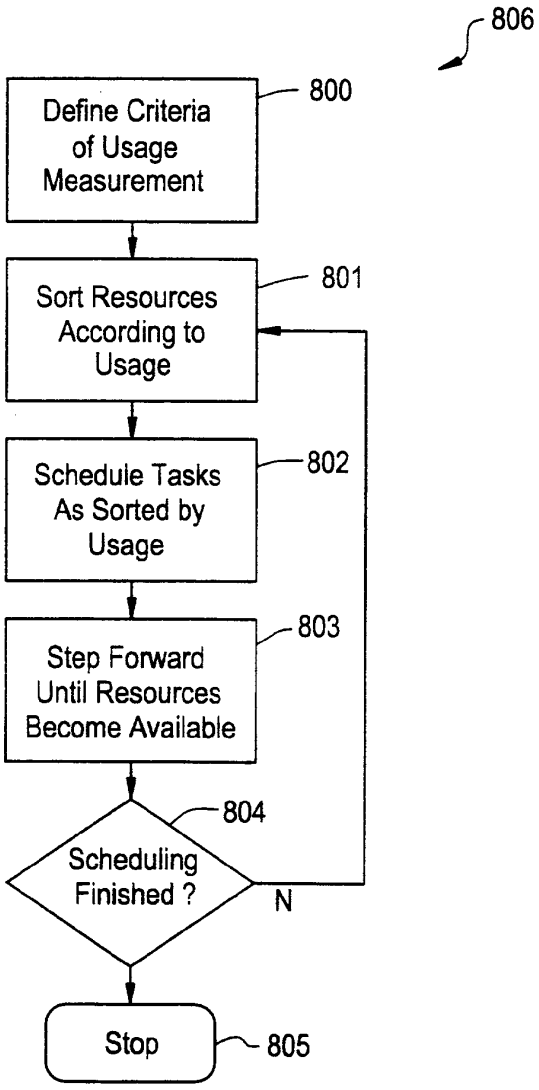

806
Define Criteria of Usage Measurement
800
Sort Resources According to Usage
801
Schedule Tasks As Sorted by Usage
802
Step Forward Until Resources Become Available
803
Scheduling Finished ?
804
N
Stop
805

FIG. 9

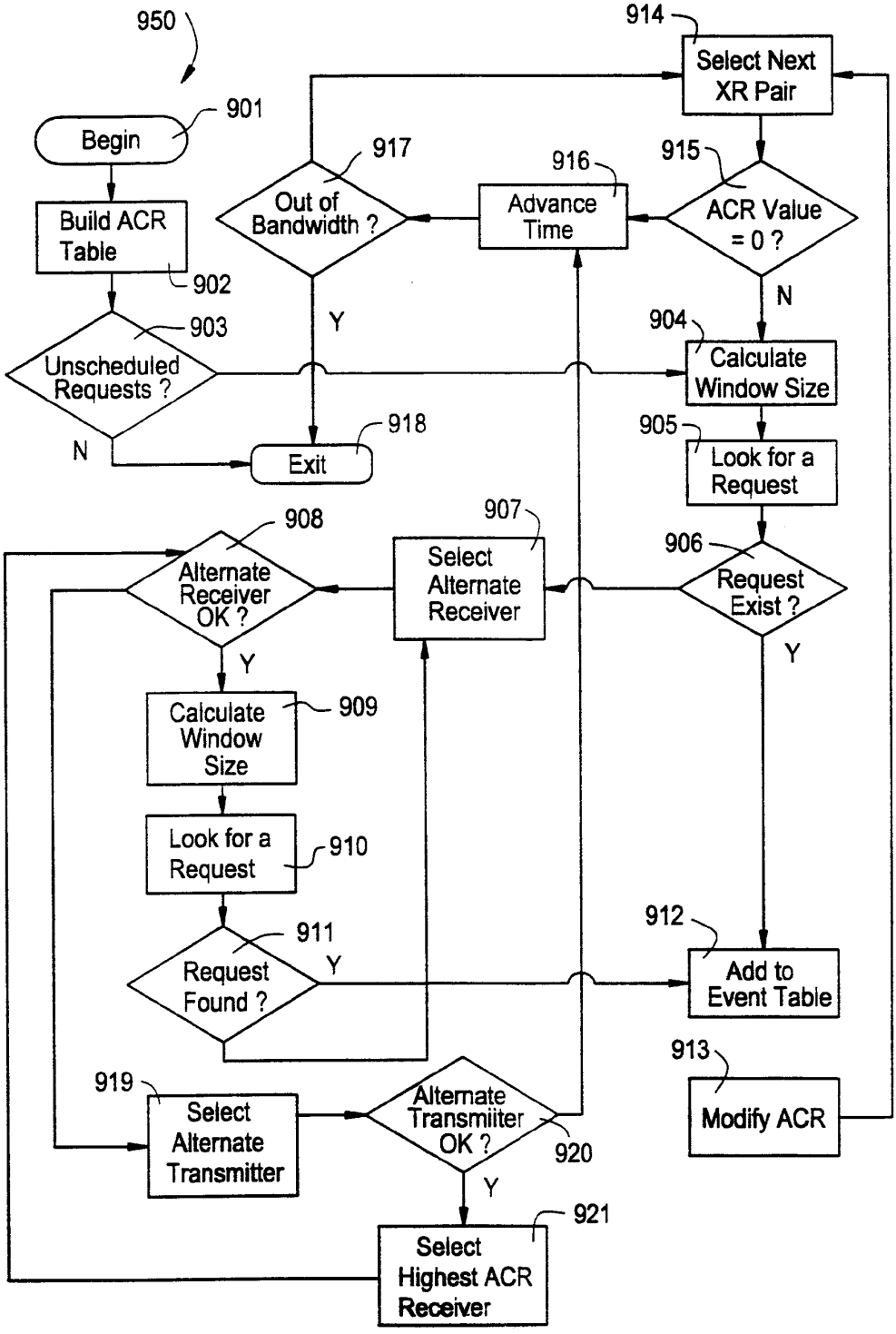
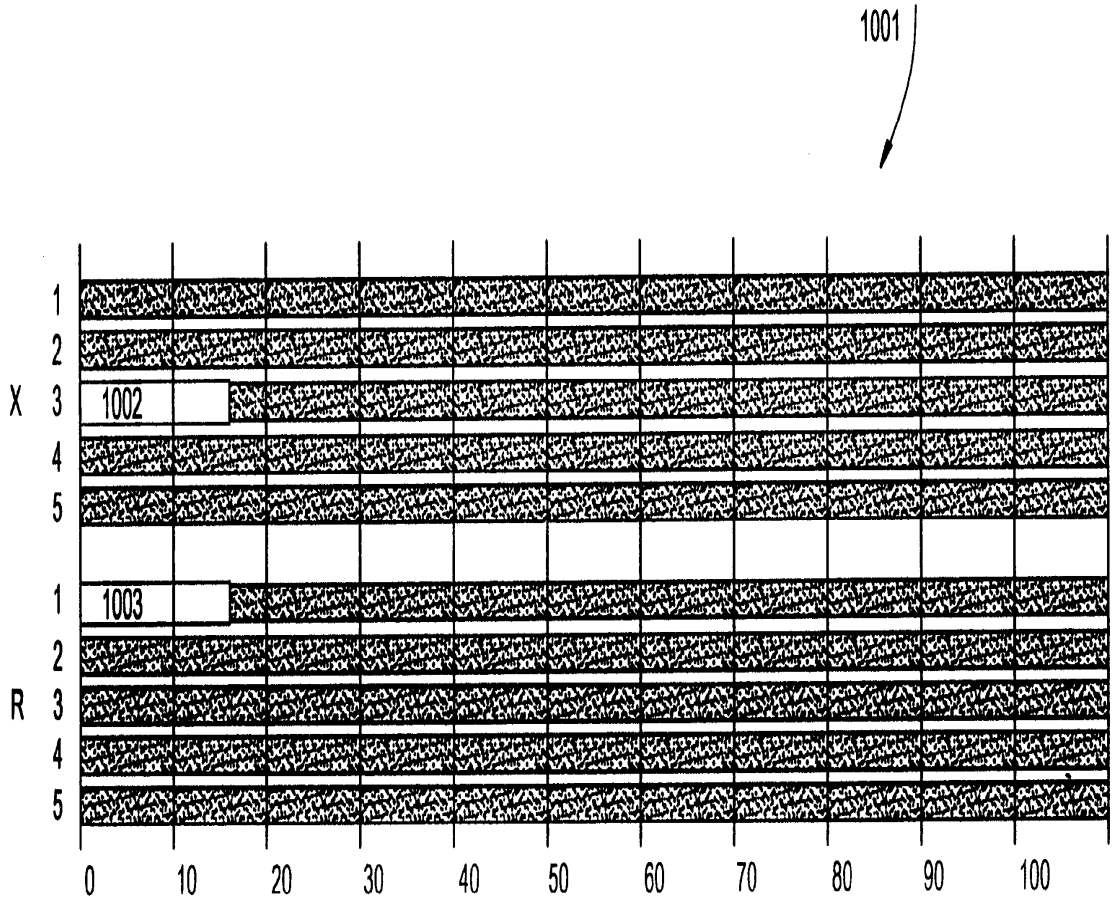

950
914
Select Next XR Pair
901
Begin
917
916
915
Out of Bandwidth ?
Advance Time
ACR Value = 0 ?
Build ACR Table
902
903
Y
N
904
Unscheduled Requests ?
Calculate Window Size
905
N
918
Exit
Look for a Request
908
907
906
Alternate Receiver OK ?
Select Alternate Receiver
Request Exist ?
Y
Y
Calculate Window Size
909
Look for a Request
910
912
911
Request Found ?
Y
Add to Event Table
913
919
Alternate Transmiiter OK ?
Select Alternate Transmitter
Modify ACR
920
Y
921
Select Highest ACR Receiver

FIG. 10

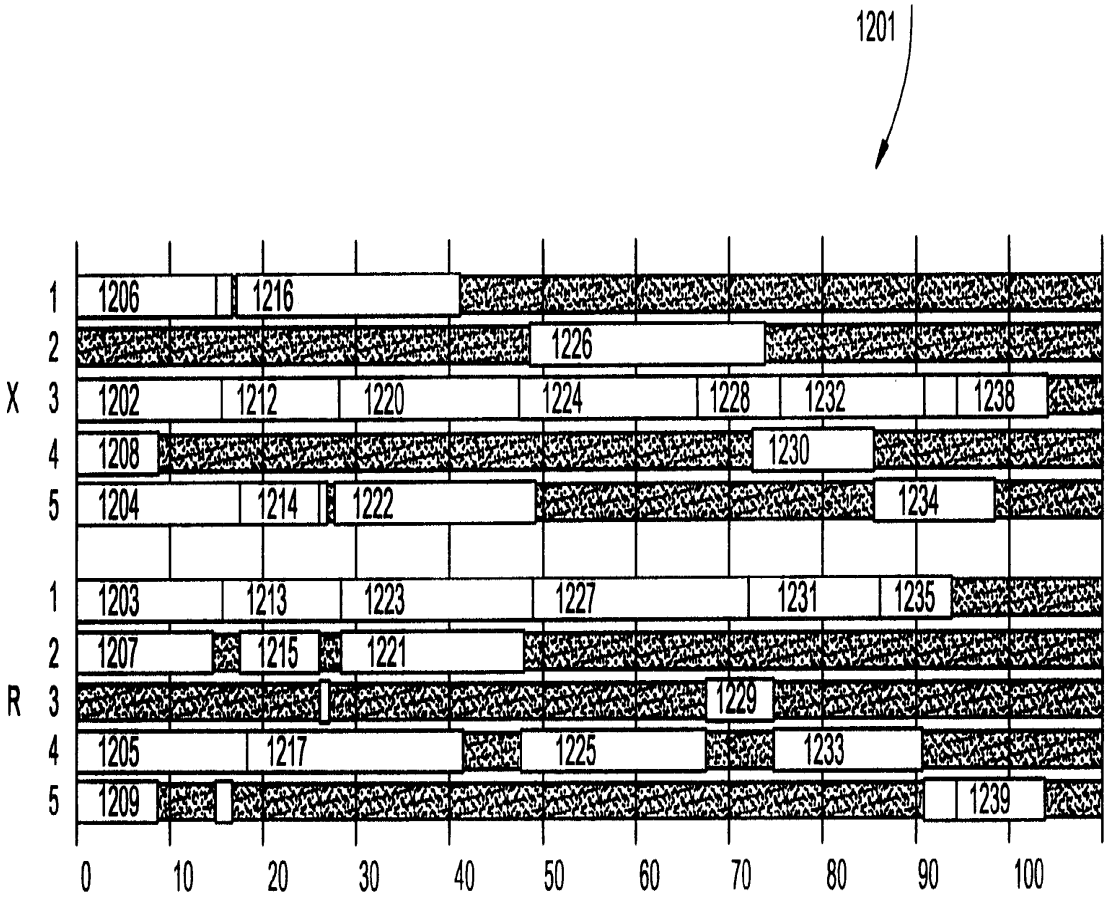


SUBSTITUTE SHEET (RULE 26)

FIG. 11

SUBSTITUTE SHEET (RULE 26)

FIG. 12



SUBSTITUTE SHEET (RULE 26)

SUBSTITUTE SHEET (RULE 26)

FIG. 13

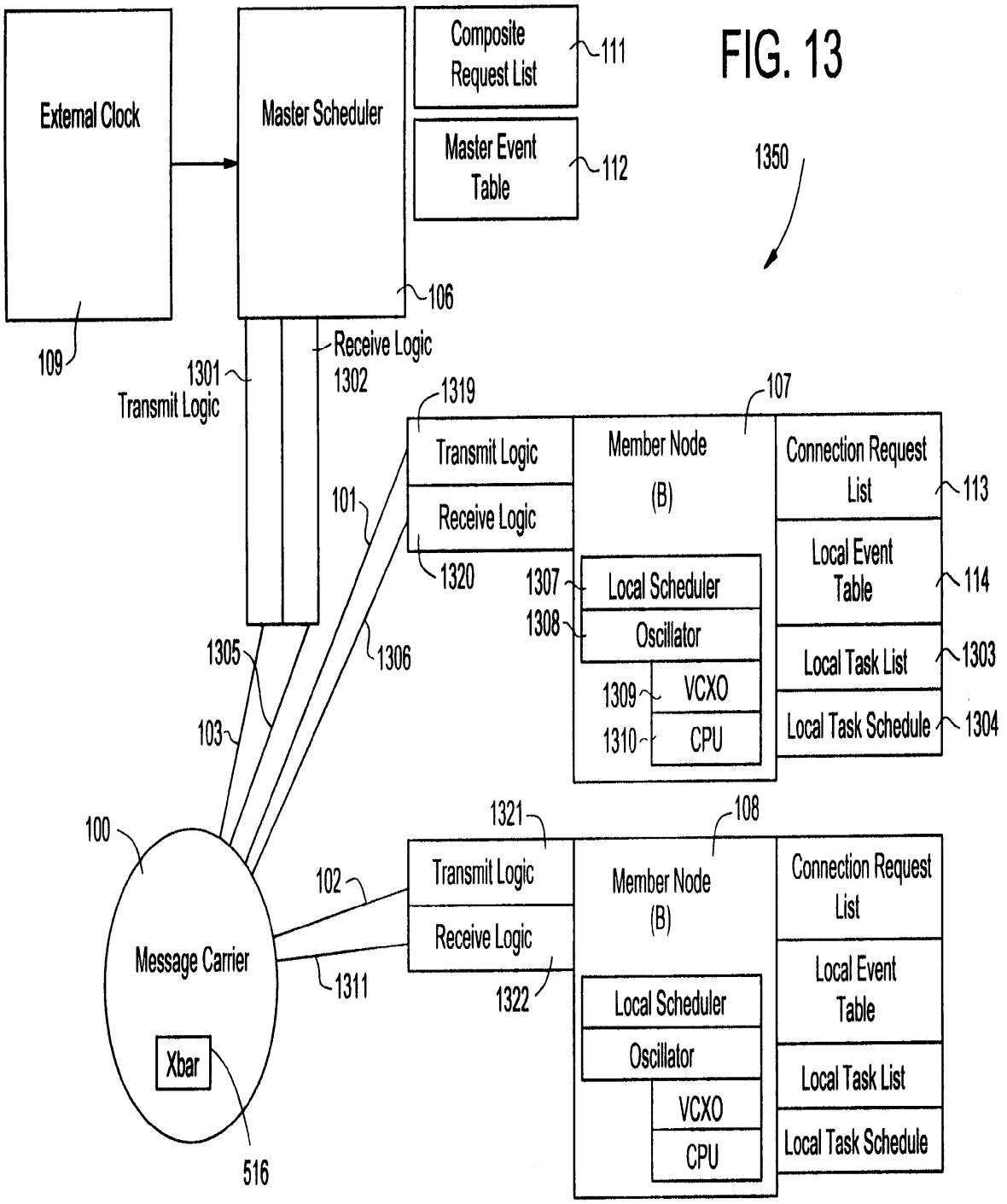
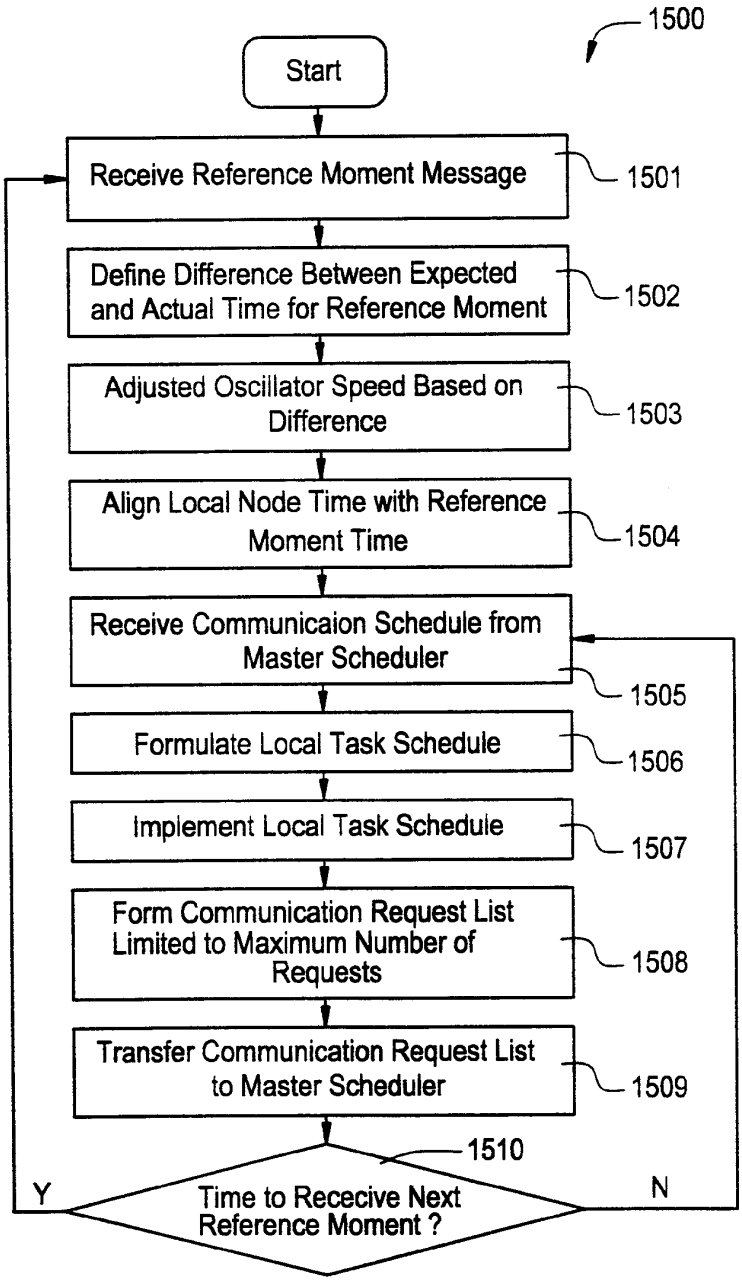


FIG. 14

1400
Tight Synchronization
Prioritization
High Bandwidth
Time Deterministic Event Control
Centralized Scheduling
Physical Circuit Switching
Limit Checking
Non-blocking Tree Topology
Low Cost
Security
Active Redundancy
Reliability
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437

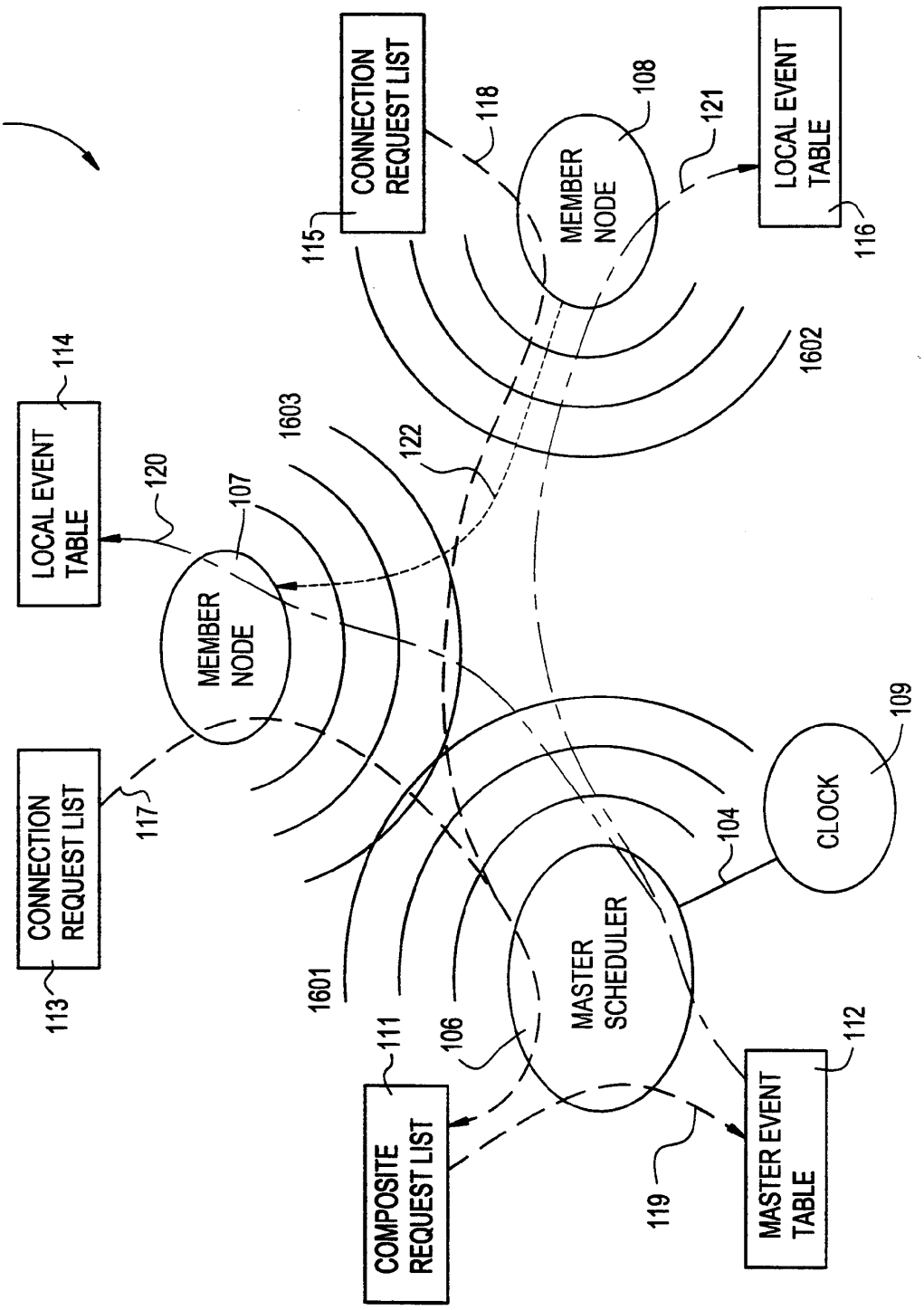
15 / 16

FIG. 15



LINKPLAY EXHIBIT 1004
Page 3362 of 4822

FIG. 16


1600
CONNECTION REQUEST LIST
115
118
MEMBER NODE
108
121
LOCAL EVENT TABLE
116
1602
1603
122
LOCAL EVENT TABLE
114
120
MEMBER NODE
107
CONNECTION REQUEST LIST
113
117
1601
CLOCK
109
104
MASTER SCHEDULER
106
111
COMPOSITE REQUEST LIST
112
119
MASTER EVENT TABLE

INTERNATIONAL SEARCH REPORT

Int ional Application No
PCT/US 99/06634

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F9/46

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	SHIN K G ET AL: "REAL-TIME COMPUTING: A NEW DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING" PROCEEDINGS OF THE IEEE, vol. 82, no. 1, 1 January 1994 (1994-01-01), pages 6-23, XP000435878 ISSN: 0018-9219 page 8, paragraph II - page 17, paragraph VII ---	1-4,10, 11,14-16
A	EP 0 183 877 A (INMOS LTD) 11 June 1986 (1986-06-11) the whole document ---	1-4,10, 11,14-16
A	EP 0 250 011 A (INMOS LTD) 23 December 1987 (1987-12-23) the whole document ---	1-4,10, 11,14-16
	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
27 July 1999	03/08/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Michel, T

1

Form PCT/ISA/210 (second sheet) (July 1992)

LINKPLAY EXHIBIT 1004
Page 3364 of 4822

INTERNATIONAL SEARCH REPORT

Int ional Application No
PCT/US 99/06634

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	RAMAKRISHNAN K V: "AN EVENT SCHEDULER FOR CONTROL APPLICATIONS" EDN ELECTRICAL DESIGN NEWS, vol. 41, no. 12, 6 June 1996 (1996-06-06), pages 155-158, 160, 16, XP000622016 ISSN: 0012-7515 the whole document	9-11

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

LINKPLAY EXHIBIT 1004
Page 3365 of 4822

INTERNATIONAL SEARCH REPORT

nformation on patent family members

Int ional Application No
PCT/US 99/06634

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0183877 A	11-06-1986	NONE	
EP 0250011 A	23-12-1987	DE 3375844 A	07-04-1988
		DE 3376894 A	07-07-1988
		DE 3378083 A	27-10-1988
		DE 3382708 A	16-09-1993
		DE 3382708 T	10-02-1994
		EP 0111399 A	20-06-1984
		EP 0113178 A	11-07-1984
		EP 0110642 A	13-06-1984
		EP 0113516 A	18-07-1984
		EP 0110643 A	13-06-1984
		JP 1028967 B	07-06-1989
		JP 1547176 C	28-02-1990
		JP 59149543 A	27-08-1984
		JP 1789597 C	29-09-1993
		JP 4070652 B	11-11-1992
		JP 59133647 A	01-08-1984
		JP 1982715 C	25-10-1995
		JP 6101042 B	12-12-1994
		JP 59194256 A	05-11-1984
		JP 2111325 C	21-11-1996
		JP 8007739 B	29-01-1996
		JP 59133662 A	01-08-1984
		JP 2502960 B	29-05-1996
		JP 59133650 A	01-08-1984
		JP 2509453 B	19-06-1996
		JP 6342378 A	13-12-1994
		JP 2688320 B	10-12-1997
		JP 6260551 A	16-09-1994
		US 5452467 A	19-09-1995
		US 5506437 A	09-04-1996
		US 5491359 A	13-02-1996
		US 5031092 A	09-07-1991
		US 4680698 A	14-07-1987
		US 4704678 A	03-11-1987
		US 4724517 A	09-02-1988
		US 5243698 A	07-09-1993
		US 4819151 A	04-04-1989
		US 4967326 A	30-10-1990

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
26 July 2001 (26.07.2001)

PCT

(10) International Publication Number
WO 01/53994 A2

(51) International Patent Classification[7]: G06F 17/30

(21) International Application Number: PCT/US01/01946

(22) International Filing Date: 19 January 2001 (19.01.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/177,768	24 January 2000 (24.01.2000)	US
09/563,250	2 May 2000 (02.05.2000)	US
09/563,173	2 May 2000 (02.05.2000)	US
09/562,939	2 May 2000 (02.05.2000)	US
09/563,104	2 May 2000 (02.05.2000)	US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier applications:

US	60/177,786 (CIP)
Filed on	24 January 2000 (24.01.2000)
US	09/563,250 (CIP)
Filed on	2 May 2000 (02.05.2000)
US	09/563,173 (CIP)
Filed on	2 May 2000 (02.05.2000)
US	09/562,939 (CIP)
Filed on	2 May 2000 (02.05.2000)
US	09/563,104 (CIP)
Filed on	2 May 2000 (02.05.2000)

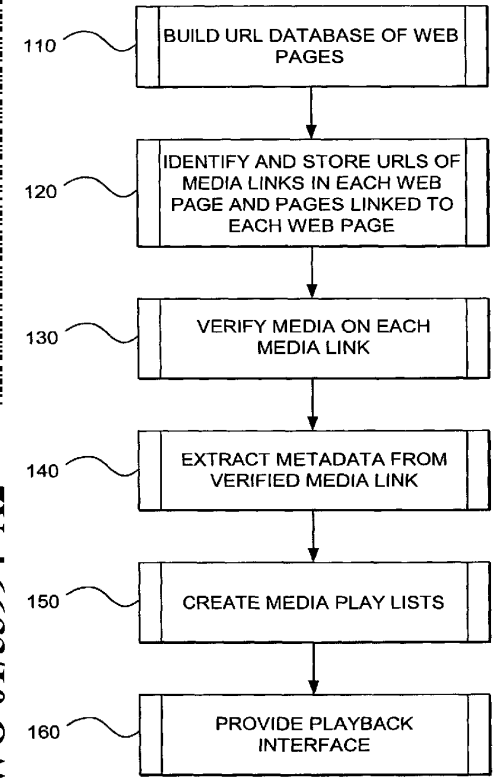
(71) Applicant *(for all designated States except US)*: **FRISKIT, INC.** [US/US]; Suite 124, 699 Mississippi Street, San Francisco, CA 94107 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **EYAL, Aviv**

[Continued on next page]

(54) Title: STREAMING MEDIA SEARCH AND PLAYBACK SYSTEM



(57) Abstract: A method is provided for playing back media from a network. The method includes receiving a search criteria from a network enabled device. The method further includes accessing a database comprising a plurality of network addresses, where the database associates each address with one or more classes of information. The addresses in the database each access a media network resource. The method further includes selecting at least one address in the database using the search criteria, and signaling the selected address to the network enabled device. The method also includes controlling the network enabled device so as to automatically access and play back the media resource of the selected address.

WO 01/53994 A2

[IL/US]; 690 Pennsylvania Avenue #210, San Francisco, CA 94107 (US). **APOSPOROS, George** [US/US]; 1340 Clay Street #25P, San Francisco, CA 94109 (US).

(74) Agent: MAHAMEDI, Van; Wilson Sonsini Goodrich & Rosati, 650 Page Mill Road, Palo Alto, CA 94304-1050 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

STREAMING MEDIA SEARCH AND PLAYBACK SYSTEM

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates to the field of streaming media content search and playback over a network. In particular, the invention relates to a computer system that enables a continuous streaming media playback from a distribution of sites available over a network such as the Internet.

Description of the Related Art

Computers currently can access streaming media on the Internet. Streaming media available on the Internet include, for example, music, video clips such as movie trailers, home movies, and animation.

Users locate streaming media on the Internet by manually selecting links. Typically, users browse the media sites that contain numerous sub-links. Users sometimes select through a chain of links to locate a desired media on a media link. Once located, the desired media link may or may not contain the desired media.

Some services provide media search engine capabilities. Users may enter a search request for selected media creations by an artist. The media search engine then displays links to categories and/or sub-links of media that are determined to match one or more criteria in the search request set forth by the user. The determination of which links should be displayed in response to the search request is dependent on the algorithm used in by the search engine. Typically, links displayed to users of current search engines are not subject to a determination of the quality or availability of the media associated with the media links. Further, the search results are outputted to the user as a display of links for the user's selection.

Many Internet streaming media outlets provide a limited number of source nodes. The sites can be unreliable when the number of users accessing the site become congested.

LINKPLAY EXHIBIT 1004
Page 3369 of 4822

SUMMARY OF THE INVENTION

An embodiment of the invention includes a method for playing back media from network. The method comprises receiving a search criteria from a network enabled device. The method further includes accessing a database comprising a plurality of network addresses, where the database associating each address with one or more classes of information. Each address accesses a media network resource. The method further includes selecting at least one address in the database using the search criteria, signaling the selected address to the network enabled device, and controlling the network enabled device so as to automatically access and play back the media resource of the selected address.

Another embodiment includes a method for playing back media from a network. The method includes receiving a request for media playback from a network enabled device. Further, accessing a database comprising a plurality of network addresses, where each address accessing a media network resource. The method also includes identifying at least two addresses from the database, signaling each identified address to the network enabled device, and controlling the network enabled device to access and automatically play back the media network resources of each of the signaled addresses.

In another embodiment, a computer system is provided for playing back media from a network. The computer system comprises a network enabled device comprising a media playback component. A database is included that comprises a plurality of addresses, where each address locates a media network resource on the network. The database includes one or more classes of information associated with each address in the plurality of addresses. The system also includes a network server module that is coupleable to the network enabled device and to the database. The network server module is able to receive a search request from the terminal that specifies one or more criterias. The network server module selects an address from the database that is associated with a class of information that matches the search criteria. The network server module signals the address to the network enabled device to cause the device to access the media network resource, and to signal media

LINKPLAY EXHIBIT 1004
Page 3370 of 4822

playback component to load the media network resource after the device accesses the media network resource.

In another embodiment, a computer system is provided for playing back media from a network. The computer system includes a network enabled platform comprising a media playback component. A database includes a plurality of addresses, where each address locates a media network resource on the network. Each address accesses a media network resource. The embodiment further includes a network server module coupleable to the network enabled device and to the database. The network server receives a request for media playback from the network enabled device, selects multiple addresses from the database, and signal the multiple addresses to the network enabled device. The network server module control a media playback component on the network enabled device to use the addresses to automatically access and play back the media network resource associated with the addresses.

In another embodiment, a network enabled device is configured to playback media from a network. The network enabled device is coupleable over the network to a database that includes a plurality of addresses. Each address locates a media network resource on the network. The network enabled device includes a user-interface to prompt for a search request. The network interface signals the request to a network server module that is communicatable with the database, and receives one or more addresses in the database that match the search request. The network enabled device includes a media playback component that is configured to be programmatically controlled by the network server module to automatically load the media network resources located by the addresses that match the search request.

In another embodiment, a network enabled device is configured to playback media from a network. The network enabled device is coupleable over the network to a database comprising a plurality of addresses. Each address locates a media network resource on the network. The network enabled device comprises a user-interface including a plurality of user-interactive features, including a first user-interactive feature that prompts to receive a search request for media playback. A network interface signals the request to a network server module upon the first user-interactive feature receiving the

LINKPLAY EXHIBIT 1004
Page 3371 of 4822

search request for media playback. The network interface is communicatable with the database to receive one or more addresses in the database that match the search request. A network playback component is configured to be programmatically controllable by the network server module to automatically load the media network resource associated with each address signaled to the network enabled device upon accessing the media network resource. A playback of the media playback component being controllable by one or more control user-interactive features.

An embodiment includes a system that provides media from a network to a terminal having a media playback component. The system includes a first network site and a second network site, where each network site locates one or more media network resources. Each media network resource is locatable on the network by a corresponding address that accesses the media network resource. A network server module is coupleable to the terminal through the network. The network server module identifies a first media network resource from the first network site and a second media network resource from the second network site. The network server module signals the corresponding address of the first media network resource to the terminal with control signals to cause the playback component to automatically load the first media network resource. The network server module automatically signals the corresponding address of the second media network resource to the terminal with control signals to cause the playback component to automatically load the second media network resource.

Another embodiment provides a media playback system for the Internet. The system includes an end terminal having a media playback component. A web server module is coupleable to the end terminal through the Internet. The web server module has access to one or more media web resources on a first web site, and to one or more media web resources on a second web site. The web server module signals a first link to a first media web resource on the first web site, and a second link to a second media web resource on the second web site. The web server module provides control signals to the end terminal to cause the end terminal to access and load the first media web resource and the second media web resource into the media playback component.

One or more of the embodiments may include a database that stores links to each of the plurality of media web resources, the web server module identifying the first link and the second link from the database.

Another embodiment includes a media playback system for the Internet. The system includes a terminal having a media playback component and a user-interface.

A web server module is coupleable to the user terminal through the Internet. The web server module has access to a plurality of links, where each link locates a media web resource. The plurality of links are accessible on a plurality of web sites. The web server module signals the plurality of links to the user terminal in a designated order to cause the terminal to load the media web resource located by each of the plurality of links into the media playback component. The embodiment also includes a database that stores the plurality of links. The database is accessible to signal the plurality of links to the web server module in the designated order. The user-interface signals one or more inputs from a user to the web server module. The one or more inputs direct the web server to alter the designated order in which the database signals the plurality of links to the web server module.

Another embodiment includes a system that provides media play-back on a network. The system includes a terminal that is coupleable to the network. A play-list module is coupleable to the terminal. The play-list module stores a first play-list signaled from the terminal. The first play-list includes a plurality of network addresses. A first network address locates a first media network resource on a first network site, and a second network address locates a second media network resource on a second network site. A network server module is coupleable to the terminal and to the play-list module. The network server module signals the first play-list to the terminal. The network server module controls the terminal to cause the terminal to access the media network resource associated with each network address in the first play-list, and to automatically load each respective media network resources into the media playback component.

Another embodiment includes a method for providing media to a terminal coupled to a network, where the terminal includes a media playback

component. A terminal is programmatically directed to access a first network site in the plurality of network sites. The media playback component on the terminal is caused to automatically load a first media web resource located at the first network site to playback a first media. The terminal is programmatically directed to access a second network site in the plurality of network sites. The media playback component on the terminal is caused to automatically load a second media web resource located at the second network site to playback a second media.

Another embodiment includes a method to provide media to a terminal coupled to the Internet. A database is accessed that stores a plurality of links, where each link opening a corresponding media web resource. A first link is selected from the database, the first link being located on a first network site. Next, a second link is selected from the database, the second link being located in a second network site. The second network site is external to the first network site. Then, the selected links are signaled to a media playback component on the terminal to sequentially access the media web resources associated with the selected links. The media playback component on the terminal is automatically signaled to load each of the media web resource accessed from the selected links so as to playback a media corresponding to each media web resource.

Another embodiment includes a system to share media playback from a network between a plurality of terminals. The plurality of terminals include a first terminal and a second terminal. The system includes a play-list component locatable on the network by a selectable link. The play-list component identifies a plurality of links to form a play-list, where each link in the play-list locating a media file on the network. The system includes a network server module that signals the plurality of links that form the play-list to the first terminal. The network server module receives a signal to transmit the selectable link to a second terminal to enable the second terminal to locate the play-list module.

In another embodiment, a method is provided to locate web resources on the Internet. A web site is accessed to identify a plurality of links using a web browser component. The web site can be automatically or programmatically accessed. Each of the plurality of links are selectable to open a corresponding

web resource of a specified data type on the web site. The plurality of links are made available to a plurality of Internet enabled devices that select one or more of the links.

Another embodiment includes a system to locating web resources on the Internet. The system includes a web browser component, and a database. A search module controls the web browser component to access at least one web site. The search module controls the web browser component to identify a plurality of links to media web resources at the web site. Each of the plurality of links are selectable to open a media web resource. The search module stores the plurality of links in the database.

Another embodiment includes a method to locate web resources on the Internet. A database that stores a plurality of links is accessed, the plurality of links being selectable to open a corresponding web media resources. Metadata information is programmatically identified about the web media resource corresponding to each of the plurality of links. The plurality of links are made accessible to a plurality of Internet enabled devices. The plurality of Internet enabled devices elect one or more of the links to open the corresponding media web resource.

Another embodiment includes a method to locate web resources on the Internet. A database is accessed that includes a plurality of links to media web resources. Each of the plurality of links are programmatically verified to open a corresponding web media resource. Each verified link is accessible to a plurality of Internet enabled devices that select one or more of the links to open the corresponding media web resource.

Another embodiment includes a system to locate web resources on the Internet. The system includes a first indexed data structure comprising a plurality of links. A media playback component is coupleable to the database. The media playback component loads each of the plurality of links to verify whether the link is selectable to open a media web resource. A second indexed data structure stores each verified link in the plurality of links. The second indexed data structure is available to the plurality of Internet enabled devices.

Another embodiment includes a method to providing links for use in a media search engine. A plurality of internal links on a network site are

identified. The network site makes a network resource of a specific data type accessible for a network enabled device. The internal links that are selectable to open the network resource of the specific data type are extracted. The external link is stored in a database. One or more of the links are automatically signaled to a media playback component in response to receiving a search requests from the network enabled device.

Another embodiment includes a method to provide links for use in a media search engine. The method includes a) receiving from a first indexed data structure a first external link to a first network site; b) initializing a second data structure to be empty;
c) determining if the first network site contains at least one internal link; d) storing the at least one internal link contained on the first network site that is not in the first indexed data structure and not in the second indexed data structure as another external link in the first indexed data structure; e) identifying the internal links contained on the first network site that are selectable to open a network resource of a specific data type or types; f) moving the first external link from the first indexed data structure to the second indexed data structure; and g) repeating steps a) through f) until the first indexed data structure is empty.

Another embodiment includes a computer system to search for links to streaming media playback on a network, the network being accessible to a network enabled device. The system includes a metacrawler to locate one or more media sites in directories containing streaming media. A media search module coupled to be signaled the one or more directories from the metacrawler. The media search module identifies a plurality of media links for the media sites. Each of the plurality of media links are selectable to open streaming media network resource. A metadata extraction module accesses each media link identified by the media search engine to extract metadata about the identified media link. A database comprising the plurality of media links identified by the media search engine, and the metadata is extracted about each identified media link. The database enables the network enabled device to access the plurality of media links.

LINKPLAY EXHIBIT 1004
Page 3376 of 4822

An embodiment includes a rating system for rating media network resources on a network that is coupleable to a plurality of terminals. The rating system includes a database having a plurality of addresses. Each address locates a corresponding media network resource on the network. A network server module is coupleable to the plurality of terminals. The network server module accesses the database to signal one or more addresses from the database to the plurality of terminals. A rating module is coupleable to the plurality of terminals. The rating module receives a rating input from each of the plurality of terminals. The rating module associates the rating input with a selected address in the database.

In another embodiment, a rating system is provided to rate media network resources on a network. The rating system includes a database comprising a plurality of addresses that each locate a corresponding media network resource on the network. The database includes one or more classes of information associated with each of the plurality of addresses. A network server module is coupleable to the plurality of terminals. The network server module communicates with each of the plurality of terminals to receive a search request. The network server module signals the database to retrieve one or more addresses from the database in response to the search request. The retrieved addresses are associated with a class of information matching the search request. A rating module is coupleable to the plurality of terminals. The rating module receives a rating input from each of the plurality of terminals. The rating module associates the rating input with a selected address in the database.

Another embodiment includes a rating system for rating media network resources available over a network. The media network resources are located on the network by a plurality of terminals. The rating system includes a database that stores a plurality of addresses. Each address locates a corresponding media network resource on the network. The database includes a rating associated with each of the plurality of addresses. A network server module is coupleable to each of the plurality of terminals. The network server module accesses the database to signal one or more addresses from the database to the plurality of terminals. A rating module is coupleable to each of the plurality of terminals. The rating module receives a rating input from one of the terminals for each of

LINKPLAY EXHIBIT 1004
Page 3377 of 4822

the plurality of addresses in the database. In response to receiving the rating input from one of the plurality of terminals for a selected address in the database, the rating module accesses the database and reconfigures the rating associated with the selected address. A play-list module accesses the addresses to select one or more combinations of addresses. The play-list module signals the play-list to the network server module as addresses to be signaled to one or more of the plurality of terminals.

In a variation, the address may be selected by the play-list module based on a criteria stored with the address in the database. Examples of criterias include rankings, reflecting preferences of users on terminals after playing back media located by the respective addresses. Other criterias that can be used to select addresses include metadata information, such as artist name and media title. For example, the search request may specify a ranking as one of the criterias. The play-list module then sorts the database for the ranking in selecting the addresses.

Another embodiment includes a method for ranking media sources on a network. The method includes accessing a database that stores a plurality of addresses. Each address locates a media resource on the network and each address is associated with a rating. A selected address from the database is signaled to a terminal coupled to the network. A rating input is received from the terminal after signaling the selected address to the terminal. The rating is associated for the selected address is adjusted in response to receiving the rating input.

Another embodiment includes a method for ranking media sources on a network. A database is accessed that stores a plurality of addresses. Each address locates a media resource on the network and each address is associated with a rating. A combination of addresses are selected to form a play-list. The play-list is signaled to a terminal coupled to the network. A ranking is received from the terminal after signaling the addresses in the play-list to the terminal. The rating is adjusted for each address signaled to the terminal from the play-list in response to receiving the ranking.

Another embodiment includes a method that ranks media sources on a network. A database that stores a plurality of addresses is accessed. Each

address locates a media resource on the network, and each address is associated with a rating. A selected address is signaled from the database to a terminal coupled to the network. A ranking is received from the terminal after the selected address is signaled to the terminal. The rating associated for the selected address is adjusted in response to receiving the ranking.

Another embodiment includes a network enabled device that comprises a media playback component. The media playback component is configured to communicate with a network-side module to receive a first plurality of links. Each of the first plurality of links locate a media file on a network. A web browser component is configured to receive a second plurality of links. Each of the second plurality of links hosts a media file located by one of the first plurality of links. The web browser component displays the web site for each of the second plurality of links when the media playback component plays back media from the media file being hosted by web site being displayed.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a flow process describing an embodiment of the invention.

FIG. 2 is a block diagram illustrating an architecture for use with an embodiment of the invention.

FIG. 3 is a block diagram illustrating a back end architecture, under an embodiment of the invention.

FIG. 4 is a block diagram illustrating a media search and playback system, under an embodiment of the invention.

FIG. 5 is a block diagram illustrating components on an end terminal receiving control information from a server-side module, under an embodiment of the invention.

FIG. 6 is a flow chart illustrating a system for forming a search database of media resources accessible on a network, under an embodiment of the invention.

FIG. 7 is a flow chart illustrating a system for forming a search database of media resources accessible on a network, under an embodiment of the invention.

LINKPLAY EXHIBIT 1004
Page 3379 of 4822

FIG. 8 is a flow chart for verifying records in a search database of media resources, under an embodiment of the invention.

FIG. 9 is a flow chart for extracting metadata about a media resource associated with a site on a network, under an embodiment of the invention.

FIG. 10 is a flow chart for forming play-lists for end users of a system under an embodiment of the invention.

FIG. 11 is a flow chart for receiving user input in response to playing back media resources from a search database, under an embodiment of the invention.

FIG. 12 is a block diagram of a media playback system including a rating feature, under an embodiment of the invention.

FIG. 13 is a flow chart describing user input to a user interface for a media playback system, under an embodiment of the invention.

FIG. 14 is a flow chart describing a rating system, under an embodiment of the invention.

FIG. 15 illustrates an exemplary structure for a database to maintain updated records on ratings for addresses containing media resources, under an embodiment of the invention.

FIG. 16 is a flow chart for creating play-lists using rating information, under an embodiment of the invention.

FIG. 17 is a flow chart for programmatically categorizing media files, under an embodiment of the invention.

FIG. 18 is a flow chart for creating personalized play-lists of streaming media files available in a network, under an embodiment of the invention.

FIG. 19 illustrates a distributed playback architecture, under an embodiment of the invention.

FIG. 20 illustrates a block diagram of a messaging application, under an embodiment of the invention.

FIG. 21 illustrates a user-interface for use with a media search and playback system, under an embodiment of the invention.

FIG. 22 includes another user-interface displaying an instance of the web browser while media is being played back, under an embodiment of the invention.

LINKPLAY EXHIBIT 1004
Page 3380 of 4822

DETAILED DESCRIPTION

A. System Overview

According to an embodiment, a system is provided comprising a media search engine. The media search engine may be used to create a database of links to media files. The links may be structured according to predefined categories and/or user-defined search criteria. A client terminal includes a media player to automatically access one or more media files using the corresponding links. The media player then plays back media contained on the media files.

Among other advantages of the invention, the user terminal accesses media files at various sites on a network, without requiring users to manually select media links. For example, user-terminals may output music to a user by automatically accessing one or more Internet sites containing media files. The music is outputted without requiring users' to view and select links to sites containing the media.

In contrast to embodiments of the invention, using other systems to search for Internet files containing media can be a distracting and time-consuming experience for an end user. In many instances, such a search will yield a series of links on a directory or web search page. A user may have to click on each individual link, one at a time, to play each individual media file. The selected media file may be broken and unavailable to deliver media content. Even if the number of broken links is not high, the user must still click on the links one at a time to activate each media file, providing at best a stop-and-go experience.

In one embodiment of the invention, a user terminal is able to receive continuous media streaming from multiple sites on the Internet. Multiple sites may be accessible to enable the user terminal to receive streaming media without any interaction required from an end user other than signaling a request to receive streaming media. The user terminal automatically accesses media links containing media using a media playback component.

The media playback component may be controlled by one or more server-side modules. In one embodiment, the media playback component on the user terminal interacts with one or more play-lists generated by server side modules. The play-lists contain media links for the media playback application. The media links may be structured or ordered in the play-lists. The play-lists may be generated automatically by back-end modules and/or manually by editors. The play-lists may also be generated by end users.

The media playback component may also interact with one or more server side search modules to access media links on the network. The media links may be automatically selected based on, for example, a search criteria from the end user.

Embodiments of the invention provide a system to search and playback media accessible over a network. In one embodiment, a media search engine is provided to enable users to request media output based on a criteria set forth in a search request. The media search engine is able to efficiently locate streaming media on the network that matches criteria set forth in a search request. The system provides continuous playback of media found on multiple sites of the network. For example, a user may specify a search based on a specified artist. The system locates one or more sites on the Internet containing media files from the specified artist. The system enables the user terminal to automatically and continuously play back media creations available on the Internet sites.

Further, a backend system under an embodiment of the invention minimizes possibilities of broken links and mismatched search results. The backend system may also be used to perform manual and/or programmatic quality check of the media associated with each link.

Further, a search engine under an embodiment of the invention employs an Internet web browser software component on the back-end to perform searches and indexing of web resources. The Internet web browser component may be a configured or modified commercially available web browser component. Server-side modules may combine to control the browser in locating media links and media sites containing media content. As a result, the media search engine under this embodiment is efficiently implemented, using existing resources on the back-end system.

LINKPLAY EXHIBIT 1004
Page 3382 of 4822

Among other advantages, embodiments of the invention enable streaming media from multiple media links to be automatically played to users. Embodiments of the invention also employ a scalable and distributed architecture. Scalability in this sense means that the service is available to a large (thousands or more) audience of simultaneous listeners or viewers while minimizing bottlenecks caused by congestion. Another advantage of a distributed architecture is that the unavailability of one media site, or of one or more media on the media site, does not preclude the user terminal from receiving media from another site. As a result, users are ensured a continuous listening or viewing experience.

Further, streaming media may be continuously outputted to users from multiple sites on the Internet based on personalized criteria set forth by users. The criteria may be set forth in one or more requests by an end user. The end user may experience media continuously outputted from multiple sites, based on only one request from the end user. This allows a user to request media through actions such as clicking requests through a user-interface.

An embodiment of the invention enables users to share streaming media experiences with other end users. For example, users may share play-lists containing links to multiple Internet sites. This enables individuals to create media programs of streaming media using multiple sites on the Internet. For example, play-lists may be shared among end users using a host web site, or e-mails.

B. Search and Playback System

A user terminal may transmit a search request from an end user to one or more modules on a server. A client side playback module, one or more server-side modules, or a combination of client and server side modules combine to access the user terminal to a site on the Internet that contains media content immediately available for loading and playback. The response to the search request is media output through the user terminal. The media content is outputted from the user terminal without any additional action on the part of the end user after the initial search request. Once media from one site is completed, the playback module automatically enables the user terminal to access and

LINKPLAY EXHIBIT 1004
Page 3383 of 4822

playback media located on another Internet site. As a result, an embodiment enables the user terminal to output continuous streaming media to an end user, where the media outputted is accessed from multiple Internet sites.

Embodiments of the invention may be implemented on the Internet. Other embodiments may be implemented on any network that carries digital information, such as local-area networks (LANs), Wide Area Networks (WAN), Extranets, Intranets, Internet, and wireless networks, or networks utilizing wireless transmissions. An example of a network for use with an embodiment of the invention includes a network operating under a transmission control protocol/Internet protocol (TCP/IP). Embodiments of the invention may also be employed on proprietary WANs, such as America Online™. Thus, discussion of embodiments employed on the Internet are exemplary, and equally applicable to other types of networks described above.

A system for use with an embodiment includes a network enabled device, a network server module and a database. The network enabled device includes a device having components to couple to a network such as the Internet. The network enabled device includes a communication port and processor, and may also include memory and a display. The communication port may be a physical port, such as a connector extending a modem connection. The communication port may also be a wireless port, such as those configured to transmit and receive radio frequency data communications. Examples of network enabled devices include personal computers, handheld devices such as those operating Windows CE™ or Palm™ operating systems, and cellular phones with Internet capabilities such as Sprint PCS™ systems. Other examples of network enabled devices include smart appliances, such as systems including speakers and a processor to receive communications from the network.

The network enabled device may include a media playback component. The media playback component includes an application that plays back streaming media files. Examples of commercially available media playback components include Real Network Player™, Apple Quicktime Player™, and Microsoft Windows Media Player™.

LINKPLAY EXHIBIT 1004
Page 3384 of 4822

In an embodiment, network server module includes server-side modules that communicate to the network enabled device through the communication port. The network modules may be coupleable to the network enabled device through a network such as the Internet. Alternatively, the network server module may exist on the terminal. The network server module may, for example, access a database on the network from the terminal. Still further, the network server module may exist on both the terminal and on a server on the network. Specifically, the network server module may comprise network-side code, executed on the terminal through a client application. For example, the network server module may includes applets or Java script delivered to the user terminal for execution of processes and functions, as disclosed herein.

The database stores a plurality of addresses. Each address locates a media network resource. The media network resource includes files that can be loaded into the media playback component to output media. As used herein, media refers to a combination of audio and/or video. Video media may include a collection of images assembled together in an animated fashion to resemble motion or action. Examples of video media include movie clips, recordings from video recorders, and animation such as cartoons. Still further, media may include a collection of still images and graphic presentations that are combined with audio media. Other examples of media include dynamic or animated pictures or text on a web page.

In one implementation, the media files may be loaded and played back to output music or music videos. As another example, media files may include video or animation with story-lines, plots, characters and resemble conventional television or radio programming. Other examples include movie clips, home movies, movie trailers, or highlights from sporting events.

As used herein, a module includes a program, a subroutine, a portion of a program, a software component or a hardware component capable of performing a stated task or function. A module can exist on a hardware component such as a server independently of other modules, or a module can exist with other modules on the same server or client terminal, or within the same program.

LINKPLAY EXHIBIT 1004
Page 3385 of 4822

The network server module is coupleable to the network enabled device to exchange communications, and to access the database. The network enabled device provides a search request, including a search criteria. The search criteria includes any condition specified by the user to identify some of media files from other media files in the database. Examples of search criterias include titles, artist names, data types, user preferential ratings, quality, and duration.

The network server module selects at least one address from the database based on the search criteria. The identified addresses are signaled to the network enabled device. The network server module may communicate with the media playback component to cause the media playback component to playback the media resource located by the address.

FIG. 1 illustrates a process for use with a system to search for and playback Internet streaming media, under an embodiment of the invention. In one application, the process is performed on architectures described and illustrated with FIGS. 2 and 3. While the process is described with reference to an integral system, one or more steps described with FIG. 1 may be performed independently of other steps. Similarly, components and modules used to perform steps in FIG. 1 may also be implemented in different systems and architectures. Further, steps mentioned with FIG. 1 may be performed concurrently with one other, or in an order different than shown in FIG. 1.

In step 110, a system builds a database of addresses. An address may include a Universal Resource Locations (URL) for network and Internet sites. A media site include, for example, a web site that allows web users to access streaming media. In other embodiments, the media site may locate network media resources on other types of networks. The media sites may be located through a media search engine, as described elsewhere in this application. An exemplary process for identifying media sites under an embodiment of the invention is provided with FIG. 4.

Each media site may provide access to media through one or more media links available at the site or through other means. The media links identify web resources having media content. These web resources may include a file of arbitrary type. Examples of file types include Multipurpose Internet Mail Extension (MIME) types such as MOV, JPEG, or RAM. The file is

LINKPLAY EXHIBIT 1004
Page 3386 of 4822

available for loading, browsing or playback on the World Wide web. Each media link may be either an internal or external link relatively to that particular media sites. An internal media link on a web-site may correspond to a URL that identifies a web resource located on the web domain, host, property or server of that site. An external media link on a media site identifies a web resource that is not located on web domain, host, property or server of that media site.

In step 120, the system identifies and stores in a database media links (URLs) for each media site. An exemplary process for identifying and storing media links on individual media site stored in a database of media site is provided with FIG. 5.

In step 130, each media link is verified. The media link is verified to contain media that is available for playback for users. Thus, broken links, inoperational or unavailable media are precluded from being verified.

In step 140, metadata information is extracted from each media link. Preferably, metadata information is extracted from each verified media link. In an embodiment, metadata may also be added to a list or database of extracted metadata. Additional metadata may be added using, for example, manual interactive editing and an editor interface (see for example, editor interface module 275 in FIG. 2). Examples of metadata information include (with an exemplary data structure type associated with each media link in parenthetical): identification (Integer), author (String), duration (String), media URL (URL), source web site (URL), media type (Integer), rating (Real number), number of votes (Integer), verification status (Boolean), edited status (Boolean), genre type (Index into a genre database table), play-list genre status (Boolean), mix (index into mixes database table), play-list mix status (Boolean), mood (index into moods database table), description (String), clip broadcast quality (integer), image size for videos (integer, integer), and play-list mood status (Boolean). One or more of these types of metadata may be extracted from the media links or from the actual media file. For example, a media link to a web resource may be extracted for identification, duration, author, and source web site. Similarly, one or more of these types of metadata may be added to the extracted metadata information. For example, genre type and description information may be added to the extracted metadata information.

LINKPLAY EXHIBIT 1004
Page 3387 of 4822

In step 150, the system creates media play-lists using media database for predefined categories. In an embodiment, verified media links are structured into play-lists, such as described with FIG. 10.

In some embodiments, links to streaming media commercials may be inserted into the play-lists in various locations between media clips. These commercials are targeted to the audience likely to listen to the media available on the play-list. The commercial may be produced and broadcast from distributed sources, or from web server module. Other examples of streaming media that can be included with play-lists includes news items and weather reports.

In step 160, a playback interface is provided. The playback interface causes the media player component on the user terminal to play media associated with media links in each play-list. The playback interface may include features to manipulate play-lists, or to switch between play-lists. For example, the playback interface may allow for a user to skip media or web resources until a preferred media or web resource is located. The playback interface is a software or hardware application that is executed on the user terminal. The playback interface may be packaged as a web application, dynamically accessible through a web server module, or be packaged as a desktop software application.

In an embodiment, a playback interface module includes a streaming media clips rating system that allows users to rate each clip as it is played back. The back-end module rating system uses these votes to generate rated play-lists that are available through the playback Interface for playback.

Further, the playback interface module may include a system to allow users to send Internet e-mail notifications to one or more e-mail addressees regarding a media clip, or to send continuous streaming media programs containing multiple media clips from multiple network sources. Recipients may initiate the playback module by selecting one or more links contained in the e-mail. Selecting a link from the e-mail initiates the play back module on that recipient's terminal, causing the play back module to play back the media clip or the programming referred to by the sender.

LINKPLAY EXHIBIT 1004
Page 3388 of 4822

The playback interface includes user interface elements that allow users to define and execute search criteria for media playback.

FIG. 2 is a block diagram illustrating an architecture of a system 200, under an embodiment of the invention. The system is shown to link a user terminal 210 with media that is accessible on the Internet 220, including the World Wide web 215. Other embodiments of the invention may operate with different types of networks.

The user terminal 210 includes any network enabled multimedia computing platform. In particular, user terminal 210 includes any Internet enabled multimedia computing platform. Examples of computing systems for user terminal 210 include personal computers (PC), personal digital assistants (PDA), smart phones, and Internet enabled televisions and radios, and other devices. The multimedia capability is manifested in the availability of a steaming multimedia playback software and or hardware component. Internet enabling means that the platform can access information over the Internet. In an embodiment, user terminal 210 runs the media location and playback interface module 270 that is accessible over the Internet. A communication channel 212, such as a phone line, wireless medium, or DSL line, is used to couple the user terminal 210 to the Internet. Alternatively, the playback module may be preinstalled on the client terminal. Under both configurations the playback module access media play-lists that are stored on an Internet web server.

A back-end database management system 245 is provided to maintain information used in providing media searching and playback to user terminal 210. The database management system 245 receives information from modules, including server-side modules that communicate with user terminal 210. In an embodiment, modules used to provide media search and playback capabilities to user terminal 210 include a media search module 230, an automatic verification and extraction module 255, an editor module 250, a play-list generator module 260, and a web server module 270. The modules may communicate with an interface of user terminal 210.

Under an embodiment, this communication is implemented using media play-lists on the web server module 270.

LINKPLAY EXHIBIT 1004
Page 3389 of 4822

The modules may also communicate with software applications or components on the user terminal, such as a web browser application or a Streaming Media player component in a manner that will be described below.

In an embodiment, media search module 230 includes a media directories meta-crawler module 234 and a media search engine 238. The meta-crawler module 234 and the media search engine 238 may be operated independently and concurrently of one another. The meta-crawler module 234 conducts a general search of the Internet 220 to locate media sites. Media sites may include web pages that are likely to contain web resources, media links to web resources, or links to other web pages that have such media links and/or web resources. The meta-crawler module 234 adds the address or location of each found media site to a media site table 243 maintained by database management system 245. The media site table 243 may list media sites that identify a URL for each web page located by meta-crawler 234.

In one embodiment, the entire media site table 243 is programmatically generated by meta-crawler module 234, without any manual or interactive human input. In other embodiments, an editor module 232 may interface with database management system 245 to manually input a URL for one or more of the media sites into the media site table 243. Another embodiment may substitute editor module 232 for meta-crawler 234, so that the media subdirectory manually receives a URL for each media site.

The media search engine 238 accesses the media site table 243 maintained by database management system 245. The media search engine 238 identifies media links to web resources on each media site provided in the media site table 243. In an embodiment, media search engine 238 contacts each site in the media site table 243 to locate media links. The media search engine 238 then stores the addresses of each media link in the database management system 245. In an embodiment, a URL of each media link is stored in a portion of a media and metadata table 247.

An automatic media verification and metadata extraction (AMVME) module 240 accesses the portion of media and metadata table 247 that contains URLs to the media or media links. The AMVME module 240 verifies each media link in media and metadata table 247. The media links are verified to

LINKPLAY EXHIBIT 1004
Page 3390 of 4822

contain web resources matching a criteria defining media. For example, each media link may be verified to contain a combination of audio or video, rather than be only a text document. In addition, the media links are verified as available for playback by users, to avoid broken or old links being maintained by database management system 245.

The AMVME module 240 also extracts metadata from the web resource associated with each media link in the media and metadata table 247. Preferably, AMVME module 240 extracts metadata from verified media links. The AMVME module 240 may automatically visit each media link on the Internet to extract metadata information, as well as verification information. The metadata extracted pertains to information available from the web resource or about the web resource on the media link. Examples of metadata that may be extracted by media extraction module 255 include information such as the author, duration, name, description text, broadcasting and playback quality of the media content and frame size and display resolution for images, video and home movie clips. For example, a media link may be associated with a web resource that is an audio media. Metadata that may be extracted from the media creation may include the artist name, the name of the media creation, length and audio/video quality. In an embodiment, media extraction module 255 also verifies that the media is available for playback from the media site. The AMVME module 240 may accesses database management system 245 to store verification and metadata information in media and metadata table 247.

In an embodiment, a metadata editor interface 275 is included in the system 200. The metadata interface 275 accepts manual entry from an editor pertaining to metadata of the web resource associated with each media link. The metadata interface module 275 may access one or more media links in the media and metadata table 247 to allow manual inspection of each web resource for metadata information. An editor operating metadata interface module 275 transmits a media streaming request to have the media of the web resource replayed for inspection on a terminal. The metadata editor interface 275 then allows for additional metadata to be stored in media and metadata table 247. Preferably, the additional metadata information includes metadata that is not programmatically available from the media link containing the web resource.

LINKPLAY EXHIBIT 1004
Page 3391 of 4822

For example, metadata editor interface 275 may be used to add information to media and metadata table 247 information such as genre of the web resource, description of the web resource, and system predefined information, such as mood and mix, that are found applicable by the editor to the web resource.

A play-list generator module 260 generates a plurality of play-lists based on information in the database management system 245. In an embodiment, play-list generator 260 accesses media and metadata list 247 for URLs to media contained on stored media links. The play-list generator module 260 may create play-lists 284 from predefined categories characterized by information stored in the database system for media links and metadata stored in table 247. Play-lists 284 are stored on web server module 270.

Under one embodiment, the web server module 270 includes a media location and playback application. The user terminal 210 interfaces with the media location and playback application through the Internet. For example, web server module 270 makes the media location and playback application available on a web site. The user can launch the media location and playback application by clicking a link on the web site. Under another embodiment, the playback application is pre-installed on the user terminal.

The playback application accesses the web server module 270 to load media play-lists that are stored on it. In an embodiment, the playback application reads Media URLs and Metadata stored in one or more play-lists. This information is used to playback continuous media from the play-lists to the user. A web page or network site hosting the media file being played back may also be displayed as an instance of a web browser on the network enabled device. For example, audio media may be played back while the user is presented with a web page hosting the audio playback (see FIG. 22 and accompanying disclosure).

The media location and playback application may output or playback media processed by the back-end system and stored in the media and metadata table 247 upon receiving a request from user terminal 210. For example, under an embodiment, music may be outputted from user terminal 210 continuously in a manner that resembles a jukebox, Disk Jockey Mix or a radio station.

LINKPLAY EXHIBIT 1004
Page 3392 of 4822

An interface of the user terminal 210 enables users to skip playback of media clips, or to switch categories. For example, a user on user terminal 210 may select to hear Jazz programming, and then switch to a genre of classical music. One or more features of a user-interface may be used to enable users to make selections (see FIG. 21 and accompanying text). The user may also control playback settings such as volume, pause, seek and retrieve additional media clip information, skip songs, or replay certain songs being automatically played. The user may also control and/or customize the creation of play-lists using the interface. For example, one musical play-list may include a combination of genres, such jazz and classical songs.

In an embodiment, the media location and playback application programmatically controls a streaming media multimedia software or hardware component to perform the actual streaming of the media digital bits to the user terminal's multimedia output device (such as video display and speakers hardware) . The media location and playback application contains functionality that responds to software events generated by the streaming media component. For example, a playback error generated by the streaming component may result in the application instructing the component to play another media file. In another example, the application determines and initiates playback of a media clip in response to the component reporting that the currently playing media has finished. The application may contain user interface elements that allow users to issue media playback commands. These commands are dispatched by the application to the component that implements the playback command for the currently played media.

In an embodiment, the media location and playback application works in combination with functional commands provided to the user via a web based software application. A user-interface may be provided to enable the user to select the function commands at the software application. An example of a user-interface is provided below, with FIG. 21 and accompanying text.

In an embodiment, a categorization module 290 accesses media and metadata table 247 to add metadata and to categorize media associated with media links in media and metadata table 247. The automatic process generates metadata such as music genre by consulting with information stored in other

LINKPLAY EXHIBIT 1004
Page 3393 of 4822

records in media and metadata table 247. For example, the module can automatically set the genre metadata information for all media creations available in the table, for a given artists, according to genre metadata entered for one or more media creations by the same artists. This process greatly contributes the efficiency and scalability of the back-end system.

FIG. 3 is high-level system software components diagram for the system 300, under an embodiment of the invention. The diagram shows how software components may be written, deployed and interact to provide the functionality described by system 200. The components of system 200 may be described as a three-tier architecture. Components are written to spec and deployed to a backend tier, a middle tier, and a front tier. The backend tier includes the database management system 245. The database management system 245 includes a database 345 and a backend interface module 355. The backend interface module 355 may be provided with, for example, a Microsoft SQL Server system (MS SQL).

The middle tier includes modules that communicate with backend interface module 355. The middle tier may include a media sites manager 360 and a media manager 365 software components. The media sites manager 360 and the media manager 365 each independently communicate with backend interface module 355. The media sites manager 360 components exposes a programmatic interface 362 to communicate with modules and components in the front tier. The media manager 365 includes a first media manager interface 366 and a second media manager interface 368.

The front tier includes a media site module 330 and a media module 340. The media site module 330 communicates with site interface 362. The media site module 340 communicates with the first and second media manager interfaces 366 and 368. The first and second media manager interfaces 366 and 368 communicate with the media module 340. The media site module 330 includes a front-end interface 332 to a directory meta-crawler 310 and a media search engine 312 modules. The media site module 340 includes a front-end interface 342 to the media search engine 312, an editor interface module 314, and an automatic verification module 316. The directory meta-crawler 310 crawls Internet media directories web sites. The links to media web sites are

LINKPLAY EXHIBIT 1004
Page 3394 of 4822

handed over to the MediaSite module 330 for storage in the database. The media search engine 312 searches for media links on web sites provided by the MediaSite module 330, these links are transferred through Interface 342 on the Media module 340 for storage in the database module 345.

The editor interface module 314 obtains media link for editing from the Media module 340, using Interface 342 and loads the media for editorial playback from the Internet. The editors provide metadata for media that are added to the database by the Media module 340.

The verification module 316 examines media files or web resources accessed through each media link and updates metadata regarding media availability in the database using Media module 340. This module also extracts metadata from Internet media and updates this in the database using Media module 340. The module queries the database for a batch of media records using the Media module 340 and automatically verifies and extracts metadata for the Internet media represented by these records.

With respect to communications from the backend tier to the front-end tier, database management system 345 of the backend tier provides records to the system 300. Each record or record set is disconnected from tables or databases of record(s). Disconnected records are transmitted from the backend tier to the front-end tier as active database objects (ADO) Disconnected record sets.

With respect to communications from the front-end tier to the backend tier, each disconnected record can be updated in the database by any components on any tier. Updated records are transmitted to the database management system 345 in the form of record set update operations. In an embodiment, directory meta-crawler 310 sends URLs to be added to records in database 345 to media site module 330 using an asynchronous method calls. The media search engine 312 transmits to media site module 330 using a get search method call for batch sites of URLs. The media search engine 312 uses an asynchronous method call to add media links and metadata associated with media links.

The components and all tiers expose programmatic interfaces that contain callable methods using the MS DCOM (distributed component object

LINKPLAY EXHIBIT 1004
Page 3395 of 4822

model) software component technology. Communication between the tiers is also implemented using method calls on these components. The components are deployed in front, middle and back tier hardware systems. Alternatively, The components may be developed and deployed using the MS COM+ components technology. Using this technology, a COM+ In Memory Database system (IMDB) proxies and caches tables of the back-end database module 245. This process speeds up the search and editorial process. COM+ services such Queued Components may used to implement asynchronous method calls exposed on Interfaces 362 and 366.

C. Media Search Engine

Embodiments of the invention locate web resources on a network such as the Internet. In one embodiment, a network browser identifies a plurality of links to one or more network sites. The links are each selectable to open a network resource of a specified data type. The identified links are then made available to network enabled devices that can select one or more of the links.

As used herein, a network browser is software that performs core functions that include (i) loading network resources; (ii) parsing, translating and laying out network resources, and (iii) displaying the network resources. The network browser includes an application programmable interface (API). An embodiment of the invention employs the network browser on a back end to locate the network resources of the specified data type. One advantage of this embodiment is that the web browser is employed on the back end programmatically, rather than through manual interaction with an editor or other user.

A network browser may include a shell, an API, and a processing module. A component of the network browser includes the API and the processing module. For Internet applications, the processing module may include, for example, a MSHTM or DLL module. The network browser component performs functions that include loading a network resource, as well as parsing, translating, and laying out the network resource.

In an Internet application, a web browser component may be used to locate resources of a specified data type. The web browser component may be a

LINKPLAY EXHIBIT 1004
Page 3396 of 4822

portion of a commercially available browser. For example, the web browser component for use with an embodiment of the invention may be a reconfigured Netscape Navigator™ or Internet Explorer™ browser.

In an embodiment, the web browser component is programmatically controlled through the API of the web browser to access the web resource for the plurality of links. The web browser may be programmatically controlled to bypass the shell of the web browser. For example, the API may be used to instruct the web browser to ignore the shell, or to detach the functionality of the shell. The remaining web browser component then identifies the links to the specified data types. The result is that the web browser component accesses the web resources of the plurality of links to identify the data types of the resources on the links while ignoring data such as images and sound.

In another embodiment, a search module controls the web browser component to access a web site. The search module controls the web browser component to identify a plurality of links to media web resources at the web site. Each of the plurality of links identified by the web browser component are selectable to open a media web resource. The search module stores the plurality of links in the database.

In another embodiment, a database includes a plurality of links to media web resources. The plurality of links are programmatically verified to determine whether each link opens a corresponding media web resource. The verified link are made available to a plurality of Internet enabled devices that select one or more of the links to open the corresponding media web resource.

The links may be verified on the back end using a media player, including a commercially available media player. For example, each link that needs to be verified may be programmatically loaded through an API of the media player. The response provided by the media player to the link determines whether the links are verified.

In another embodiment, the media player may be programmed to identify metadata from the media web resource of each link. The metadata may then be stored in a database associated with the link.

Among other advantages, embodiments enable network links to files of a particular data type to be rapidly accumulated and stored in a database. Each

LINKPLAY EXHIBIT 1004
Page 3397 of 4822

of the links are selectable to open a file on the network. The files may enable a terminal to play back media. In an embodiment, the addresses access media files that can be loaded into the media playback component of the user terminal. The files can be stored in the database with information that characterizes the files associated with the links. Thus, the links may be characterized by, for example, metadata information, and one or more classes of information.

In addition, embodiments enable each link in the database to be programmatically verified, so that there are no broken or unavailable links in the database. Still further, some metadata information may be programmatically identified from each media file. In contrast, existing systems verify links manually, employing interactive users to perform the manual functions. Existing systems also extract metadata information manually.

FIG. 4 illustrates a block diagram in which system 200 receives a search request 203 and provides a response 209. In an embodiment, system 200 processes the search request 203 using the web server module 270 and the media and metadata database 247. The end terminal 210 signals the search request 203 to web server module 270. The web server module 270 accesses the media and metadata database 247 to retrieve one or more URLs matching the search request. The web server module 270 signals the response containing the retrieved URLs to the media playback component 211 of end terminal 210.

In an embodiment, the search request 203 includes one or more criterias that specify a selection of URLs from media and metadata database 247. The criterias may correspond to parameters in media and metadata database 247. The table 249 illustrates a data structure of media and metadata database 247. The table 249 includes a URL list comprising a plurality of URLs. Each URL provides direct access to a web resource containing media. Each URL is characterized by one or more parameters that correspond to metadata information about the web resource associated with the URL. As an example, table 249 provides parameters as being genre (G), data type (DT), category (C), web resource identity, and one or more play-lists (PLAY1, PLAY2, PLAY3).

The genre data is a broad class identifier of the media creation comprising the web resource of the respective URL. For music, the genre may include rock, classical, and jazz. For movies, the genre may correspond to

LINKPLAY EXHIBIT 1004
Page 3398 of 4822

romance, comedy, horror etc. The genre may be identified either programmatically, or through an editor interface. One genre may be associated with one or more URLs in table 249. In the example provided, URL1-URL7 are in either one of three genres, G1, G2, and G3. Alternatively, several genres may be associated with one URL.

The data type parameter corresponds to the MIME characteristic of the web resource associated with the URL. The category parameter may correspond to a sub-class of a genre. For example, in music, a category may correspond to soft rock. In movies, a category may correspond to the time-period of the movie. The table 249 illustrates an example in which the category is unique to the genre. Thus, a web resource of one genre is not in the same category as a web resource of another genre. As an example, URL1 and URL3 are in the same category, as are URL2 and URL5. However, no other URLs are in the same category.

Other metadata information that may be included in media and metadata database 247 include identifier information. The identifier information identifies the web-resource. The identifier may provide name of a specific media creation, as well as an artist or author of the media creation.

In an embodiment, media and metadata database 247 includes play-list information as parameters of metadata information. The play-lists may be identified in any one of several ways. For example, the play-lists may be identified by a unique name or other identifier. The play-lists may be identified by another parameters, such as genre or category type. A Boolean data type may be associated between each URL and each play-lists.

The criterias of the search request 203 specify one or more parameters to media and metadata database 247. For example, search request 203 may include criterias corresponding to one or more of a genre, category, play-list, or identifier. In an embodiment, web server module 270 accesses media and metadata database 247 for URLs that have all of the parameters set forth in the search criteria.

The web server module 270 retrieves the URLs matching the criterias of the search request. The response 209 is signaled to the media play-back component 211. The response includes one or more URLs. It is noted that

LINKPLAY EXHIBIT 1004
Page 3399 of 4822

when play-lists are requested, additional URLs multiple play-lists are provided. The response 209 may also include metadata information. For example, the response 209 may signal to end terminal 210 the duration of the web resource for each URL, the artist, the history, etc.

The web server module 270 further signals control information 207 to access the URL provided in the response 209. The control information 207 causes end terminal 210 to load the web resource for the media playback component 211. Thus, the media playback component automatically loads the web resources associated with each URL included in response 209. The experience provided to end user 210 is that media is outputted in response to inputting a search request. This is in contrast to other systems in which the user is provided links to media sites containing web resources matching the search criteria.

As an example, a user may specify a media creation from a specific category. For example, the user may input a search request for "nature sounds". The web server module 210 accesses media and metadata database 247 for parameters that match "nature sounds". In one application, a play-list is located that is pre-programmed to provide URLs to web resources containing nature sounds. The response 209 then comprises one or more play-lists, each containing multiple URLs to web resources containing nature sounds. In another application, a category parameter or sub-parameter is searched for "nature sounds". The response 209 may include one or more URLs that are not pre-programmed into play-lists. The response 209 may provide URLs to media playback component 211 one at a time, in groups (such as in play-lists), or all at once.

In an embodiment, categorization module 280 may be used to programmatically create one or more parameters such as illustrated by table 249. The parameters may be determined, by for example, identifying metadata information on the media site hosting the URL.

FIG. 5 is a block diagram illustrating the media playback component 211 being controlled by one or more modules of system 200, under an embodiment of the invention. The web server module 270 signals control information to an application program interface 276 of the media playback

LINKPLAY EXHIBIT 1004
Page 3400 of 4822

component 211. The control information may be provided by the media locator and playback application of the web server module 270. The web server module 270 signals commands, with one or more URLs corresponding to media resources selected to be signaled to the user terminal 210. As an example, commands from web server module 270 may be instructions that use each URL as an arguments. Examples of commands that control the media playback component include play(URL) and pause (URL).

As an optional feature, web server module 270 may also signal control information to a web browser component 213 of user terminal 210. The control information may be in the form of commands to access and display a web site associated with the media resource. The commands may be provided to an application program interface 279 of the web browser component 213. This allows the system 200 to display the web site associated with the media resource selected to be played back on user terminal 210. Thus, user terminal 210 may play back media from the media resource while displaying the web site where the media resource is located. One advantage of this embodiment is that it allows users to receive media playback from the media resource in one medium, such as audio, while providing images, audio text, or media not associated with the media resource. Thus, users can listen to songs from media resources signaled to user terminal 210, while viewing banner ads on the web site where the media resource is located.

Each URL signaled from web server module 270 has a network protocol. For media resources, and specifically audio files, types of protocols include "HTTP" protocol, "PNM" protocol (RealNetworks, having .RM extensions), or "RTSP" protocol (having .RAM extensions). The URLs signaled by web server module 270 include the protocol at an initial portion of the string forming the URL. Preferably, for HTTP protocol files, the string portion corresponding to "HTTP" is replaced with "PNM". This adjustment prevents playback component 211 from failing as a result of a bug in the media playback component, particularly if the playback component 211 is a RealNetworks Player™.

LINKPLAY EXHIBIT 1004
Page 3401 of 4822

The web server module 270 may be either a network-side module, client side module, or a combination of both. In either embodiment, web server module 270 may access the database directly or indirectly.

FIG. 6 illustrates a process for a component of an Internet media search module, under an embodiment of the invention. A process such as described with FIG. 6 may be used to build a database of media sites, where each media site includes media links and/or links to other media sites. In an embodiment, the process of FIG. 6 is applicable to meta-crawler module 234 in system 200 (FIG. 2).

The process of FIG. 6 is a backend operation that is unobservable to a user of the user terminal. Preferably, the flow process of FIG. 6 is an automatic or programmatically controlled process, conducted periodically. For example, media links may be identified and stored under a flow process such as shown by FIG. 6 every few days or weeks. The duration between executions of the flow process of FIG. 6 maybe referred to as an idle period.

The process of FIG. 6 provides for extracting URLs of media sites from a web pages directory. Examples of a web directory for use with an embodiment of the invention includes directories on web sites such as Yahoo.com® and Lycos.com®. The flow process of FIG. 6 is a backend operation that is unobservable to a user of user terminal 210. Preferably, the flow process of FIG. 6 is an automatic or programmatically controlled process that does not require human interaction.

In step 410, a directory home page is added to a searched-pages data structure. The searched-pages data structure maintains. A similarly structured parsed-pages data structure is also maintained to hold URL of pages already processed by the module. The parsed-pages data structure indicates whether a home page web directory was previously parsed by the process. The searched-pages and parsed- pages data structures are keyed or indexed by URL and they support querying for existence of a given URL in them. Examples of keyed or indexed data structures include database tables and hashtables.

In step 410, the parsed-pages data structure is empty, indicating that the directory home pages in the searched-page data structure have not been parsed.

LINKPLAY EXHIBIT 1004
Page 3402 of 4822

In step 420, a determination is made as to whether the searched-pages data structure is empty. If the determination is affirmative, the flow process is done. This occurs when the process has parsed all the Internal web pages in the directory. If the determination is negative, a current page link is called from the searched-pages data structure in step 430. The current page is then loaded into memory and parsed. Parsing means loading and reading the HTML source (or equivalent) code of the web page so its content is accessible and in a machine-readable format.

In an embodiment, the page is parsed using an HTML parser component. An example of an HTML parser is a web browser. Thus, the current page may be parsed using a web Browser component. Specifically, step 440 of the process may be implemented using an application program interface (API) that is exposed by the web Browser component. In this context, the web Browser component is configured and used in a back-end server process with no visible presentation area or end user. It is configured not to load or render media at the web page so that the loading is more efficient. The configuration may occur through the API.

In an embodiment, the web browser is configured to parse web pages efficiently by, for example, automatically excluding a presentation layer from being displayed. Further, the web browser may be programmatically configured to not load or parse information that is not critical to the search function. For example, the web browser component can be configured to not load media data found on web pages.

In step 440, all links to media sites on the currently parsed-page are determined using the parser. The HTML parser API allows access to the page document object model. In step 450, all new external page links are added to the media sites database. An example of a web-page data structure is provided with media site database table 243 (FIG. 2). The database stores the URL of the media sites and not the sites themselves. New external page links implies media sites that are not already indexed or present in the media site database.

In step 460, all URLs also link to internal links found on the currently parsed page are added to the web pages data structure, provided that the URL in question is (i) not already existing in the searched pages data structure and (ii)

LINKPLAY EXHIBIT 1004
Page 3403 of 4822

not already existing in the parsed pages data structure. In step 470, the currently parsed page is moved to the parsed pages data structure, and the flow process returns to step 420. This process adds the URL of all the media sites indexed by the directory to the media sites database.

FIG. 7 illustrates another component of an Internet media search module, under an embodiment of the invention. A process such as described with FIG. 7 may be used to identify and store media links to web resources that are accessible from one or more web site. The process of FIG. 7 may be used in conjunction with a process such as described with FIG. 4. In an embodiment, the process of FIG. 7 is applicable to media search engine 238 in system 200 (FIG. 2).

In an embodiment, the flow process of FIG. 7 is a backend operation that is unobservable to a user of the user terminal. Preferably, the flow process of FIG. 7 is an automatic or programmatically controlled process, conducted periodically. For example, media links may be identified and stored under a flow process such as shown by FIG. 7 every few days or weeks. The duration between executions of the flow process of FIG. 7 maybe referred to as an idle period.

The flow process of FIG. 7 assumes access to a media sites database store. The database includes URLs to each media site. An exemplary database of media sites includes media sites and metadata table 243, described with FIG. 2. Each record contains a URL field for media site and a field indicating the last date, if any that the process described in FIG. 5 lastly processed the web site at the URL in the URL field. Reference to a media site that is parsed implies that the media site was programmatically examined for media links to web resources, and for links to other media sites using a process such as the one described in FIG 5.

In step 510, MIME types are determined for web resources. Examples of MIME types that can be selected for step 510 include JPEG, MOV, RAM and WAVE.

In step 515, a record for a media site is fetched or received from the database. A condition of the media site received is that the media site was not parsed by the process described here during the idle period. This condition may

LINKPLAY EXHIBIT 1004
Page 3404 of 4822

be specified by checking, for example, the date field associated with the record. For example, a date field may indicate when the media site was previously parsed.

A determination is made in step 520 as to whether a record and a URL was received in step 515. If no URL was received, the system interprets that all media sites in the database have already been parsed during the idle period. If a URL is received, the system in step 525 adds the URL of the media site to a URLs data structure of media sites to be processed. The URL data structure of unparsed media sites may be indexed or keyed. For example, the URL data structure may be a list, or a hashtable software data-structure.

In step 526, the last search field of the record fetched in step 515 is updated with the current date to indicate that the media site is parsed. In an embodiment, the field corresponds to a date in which the last parsing occurred.

In step 530, a URL in the date structure of unparsed media sites is fetched or received. If in step 535, a determination is made that the URL data structure is empty, the system returns to step 515. As will be further described, the flow process returns to step 515 only when step 570 is completed. If a determination is made that the URL data structure is not empty. Thus, steps 510-545 allow the flow process to distinguish between when a media site is being parsed for the first time, or has been previously parsed by the process during the idle period.

In step 545, media links on the media site fetched in step 530 are extracted from the HTML code of the page fetched in step 530. The media links are associated with web resources on that media site. In step 545, the media resources may be in any MIME format recognizable as media. In an embodiment, the currently parsed page is parsed using an HTML parser. An example of an HTML parser is a web browser. Thus, the media links may be extracted using a web browser. Specifically, the media links may be extracted using an application program interface (API) provided by a web browser software or hardware component. The web browser may be configured to perform this task efficiently by, for example, excluding a presentation layer. Further, the web browser may be programmatically configured to not load or parse information that is not critical to the search function. For example, the

LINKPLAY EXHIBIT 1004
Page 3405 of 4822

browser component may be configured to not load media data found on web pages.

In step 550, new media links on each media site that match the MIME format specified in step 510 are added to a database. New media links refers to media links that do not already existing in the database from, for example, a previous execution of the flow process. In step 555, metadata is extracted from each new media link found in step 550. The metadata may also be stored in the database, with a reference to the URL the media that the metadata refers to. An exemplary database is provided with media and metadata table 247 (FIG. 2). An example of metadata is the URL of the web Page that provided a link for each media URL.

In step 560, the URLs of all new internal media site links on the media site currently being parsed are added to the URL data structure. New internal media site links refers to URLs of media sites that do not already exit in the URL data structure and that are not in the parsed URL data structure.

In step 570, the currently parsed page is removed from the URL data structure and added to the parsed URL data structure. The flow process returns to step 530.

D. Verification and Extraction Flow Processes

FIG. 8 illustrates a flow process that verifies and extracts metadata from Internet streaming media files. While media links are specified, other embodiments of the invention may employ the flow process of FIG. 8 within a system that incorporates verification and extraction of any content or resource associated with links stored in a database. A specific application employs a process such as described with FIG. 4 in the system 200. In the system 200, flow process of FIG. 8 may be performed by the AMVME module 240.

In step 610, a module operating the flow process of FIG. 8 fetches or receives an unverified URL from a database. An example of such a database is provided by media and metadata table 247 (FIG. 2). The unverified URL corresponds to a media link on a media site stored in a database such as the media site database 243 (FIG. 2).

LINKPLAY EXHIBIT 1004
Page 3406 of 4822

In step 620, a determination is made as to whether a URL for a media link is present. If the URL to the media link is not present, the module assumes all media links have been determined as being verified, and the process is done.

If a URL exists, the module in step 630 loads the URL into an Internet multimedia playback software component and programmatically control the component to provide metadata embedded in the media file. In response to this request, the component loads some or the entire file over the Internet and provides the process with this information. In step 640, a determination is made as to whether the media or web resource associated with the URL was successfully loaded over the Internet by the module. If the determination is that media was not loaded, then in step 650 the URL associated with the media link is marked as unavailable and verified. The media is marked as verified to prevent the process from revisiting it once it already extracted metadata for it and verified it. The availability mark assists the flow processes described in FIG. 9 and FIG. 10. The determination may be in the negative if, for example, the media link is old and no longer contains a media file that is available for playback through its URL, or if the media link contains content other than what is designated as media.

In an embodiment, the process may use an availability rating for each media. Under such schema, each media is assumed to have the maximum availability score. For media that are currently not available for playback, step 650 may lower the score by one. The system may consider the media as unavailable if its score is below a predefined threshold. This process is useful since Internet streaming media availability may vary according to factors such as web server load and the time of day or year.

If the media is loaded, then in step 660 the media metadata is extracted from the playback component. Examples of metadata that can be extracted in this step include-artist name, playback duration, playback quality, frame size etc.

In step 670, extracted metadata is stored in a database with the associated URL of the media link that was presently verified. In step 680, the URL of the media link presently verified is marked as verified in the database. The flow process then returns to step 610.

LINKPLAY EXHIBIT 1004
Page 3407 of 4822

FIG. 9 illustrates a process for interactively adding metadata to URLs of media stored in a database. In an embodiment, the database may correspond to verified URLs of media links determined in FIG. 8. For example, a process such as shown by FIG. 9 may be performed on information stored by AMVME module 240 in media and metadata table 243. The process of FIG. 9 may be performed by a module interface, such as editor interface 275 (FIG. 2).

In step 710, an unedited media record is fetched or received by an interface module from a database. The unedited record may be one or more categories of metadata and other information about a media link, media site, or web resource. In an embodiment, the unedited media information includes a URL to a media link, as well as metadata extracted programmatically, such as described with FIG. 8. In step 720, a determination is made as to whether a record was received. If no record is received in step 720, the flow process assumes there are no unedited records remaining in the database, and the process is done. If a record is received, the in step 730 the interface module is updated with the record received. This may correspond to displaying the record fields to the editor operating the editor interface.

In step 740, the web resource associated with the record received is loaded into an Internet multimedia playback software component. Preferably, the software component is programmatically constructed and controlled by the module interface. The software component plays back the media to the editor. The editor is able to experience the media played back from the web resource associated with the media link. The editor is able to determine metadata information regarding the web resource. For example, the editor may determine mood, genre, quality, appropriate mix name, and description of a web resource such as an audio media creation or a home movie clip. In an embodiment, the editor controls playback of media located by the search engine on the network side, such as by pausing and playing media back through an editor interface.

In step 745, a determination is made as to whether the record received also includes previously determined metadata. The previously determined metadata may be extracted programmatically in another process, such as described with FIG. 8. If the determination is made that the record received does not contain extracted metadata information, then in step 750 the editor

LINKPLAY EXHIBIT 1004
Page 3408 of 4822

interface automatically extract metadata from the web resource associated with the URL of the record. To accomplish this step, the editor interface may access another module that automatically extracts certain types of metadata information. For example, the editor interface may forward the record to AMVME module 240 (FIG. 3) that performs a flow process such as shown by FIG. 8.

Once metadata is included with the record, then a determination is made in step 760 that the editor operating the interface module may choose to save the auto-extracted metadata already included with the record in the database. If the determination to step 760 is negative, then in step 765 the editor updates media information with editor provided metadata using input elements on the editorial software user interface. Once the determination in step 760 is positive, the record is marked as edited and updated in step 770. Then, all the newly added metadata for the media record is updated to the database. The process then continues from step 710 for the next unedited media. The media is marked as edited so it won't be included for editing in the process.

FIG. 10 illustrates a process for generating a play-list. The flow process of FIG. 10 assumes that play-lists names are predetermined and stored in a database. Each play-list is identifiable by its name. For example, classic music, jazz and rock. The play-lists include records of media links. In an embodiment, a record includes at least a URL to a streaming media that is categorized in a database as belonging to the play-list name. In step 810, a play-list name is received from the play-lists database. In step 820, a determination is made as to whether a play-list name was received. If the determination is negative, the system assumes it produced play-lists for all play-lists names, and the flow process is done. This process is routinely executed to add all newly added media to the appropriate play-lists.

In step 830, media records that match search criteria are fetched from the media database. The criteria is that each record play-list record field must match the current play list name obtained in step 810 and that the 'in-play-list' record field for the play list name has False value. In step 840, a play-list is generated to include all media stored at the records fetched is step 830. The new play-list contains one record entry for each fetched media record. Preferably,

LINKPLAY EXHIBIT 1004
Page 3409 of 4822

each play list record includes media URL and metadata information that is obtained from the media database record.

In step 850, the media records called in step 840 are labeled as being in the "in- play-list" for the current play-list name. This is achieved by setting the "in-play-list" value for the media database record to true for the appropriate play-list name.

In step 860, the generated play-list is made available to a server-side module, such as web server module 270 (FIG. 2). As an example, the play-lists may be stored, copied or appended to be made available on the web server module. Once the play-lists are available on a server-side module, media URLs and metadata stored in the play-lists is made available to the user terminal so that the user terminal may customize media output available from the play-list. Specifically, one or more playback applications that run on the user terminal may read the play-lists, access media links on the play-lists, continually play-back streaming media from media URLs in the play-list and present media metadata to users for further interaction. The play-list may also be configured to provide access to a client-side media player component that uses a URL of a media link to load and play the media associated through it. Additionally, users may further modify and edit play-lists to create personalized media programming. Further, play-lists may be dynamically generated by a web application in response to a request for media playback made on the user terminal.

FIG. 11 is a flow process for software or hardware application that enables a user terminal to playback streaming media programming determined by play-lists. An embodiment described below assumes that play-lists are available for the process, and that play-lists are identifiable by play-list names. For example, the play-lists may be dynamically generated by the play-list module, in response to a search criteria signaled from a user terminal. The play-lists may also be manually generated by editors on the network side. The play-lists may be predetermined by, for example, a process described with FIG. 10. The playback component can access the play-list without any direct interaction with server side modules.

LINKPLAY EXHIBIT 1004
Page 3410 of 4822

In an embodiment, the flow process employs a streaming media player component installed on the user terminal. The media player may be preexisting on the user terminal. Examples of media players for use with an embodiment include RealNetwork Player™, Microsoft Windows Media Player™, and Apple QuickTime™. The application described in the process may be web based or installed on the user terminal.

In step 910, an application interface for a media player on the user terminal is provided.

In step 920, a default play-list name is selected from a list of play-lists. In an embodiment, a database of play-lists is stored on or accessible through the web server module 270. The play-list generator module creates and stores play-list on the web server module to provide the interface with the user terminal 210 and media player stored thereon. Each play-list may store two or more media links, and preferably a plurality of media links.

In step 930, one or more play-lists for the current play-list name are loaded. In step 940, media is presented and played-back on the user terminal. To playback media, server-side modules may provide the media player with URLs to media links that are stored in each play-list. Each play-list may include one or more media link URLs and media metadata. The user terminal then accesses the URL and loads the web resource associated with that media link into a streaming media playback component on the user terminal. Media playback on the user terminal includes outputting, for example, audio and/or video stored in digital format on web resources associated with a media link. In embodiments where the web resources include video media, the step described may dynamically adjust the interface size and the playback component that will handle the actual playback according to the media clip metadata.

Under an embodiment of the invention, the media playback component continuously plays back media by (i) accessing a first site on the network and playing back media from the first site, (ii) then automatically accessing a second site and playing back media from the second site. The media sites may be provided by play-lists which that are made accessible to the media playback component. The sequence for automatically and continuously playing back media may be repeated for each media link included in the play-list. The play-

lists may include hundreds of media links, thus allowing the media play back component to automatically and continuously play back media using numerous sites on a network. As a result, a user is able to experience continuous media play back for hours at a time.

The user terminal also includes an interactive interface to affect the media being played back. An end user on the user terminal may choose to manipulate media playback through one or more commands that may be inputted through the application interactive interface or presentation layer.

In step 945, a determination is made as to whether an event occurred. If an event occurred, the flow process determines the event. The flow process may determine the event that occurred sequentially or concurrently. Preferably, the flow process is configured to receive media playback event from the streaming media playback component. If the event is to skip the currently played media, then a determination in step 960 causes a corresponding action of the media being skipped, and the process step returning to step 940. If the process received a playback error event, or an playback error event, then a determination in step 965 causes the flow process to return to step 940 to playback media from the next web resource of the play-list or play-list name.

If a user quits the application, or signals to quit, then a determination in step 970 causes the flow process to be done. In step 975, a user may choose to view a new media web site while media is being played back on the user terminal. Then in step 980, the media web site is opened in a new Internet window, preferably using the client side web browser.

In step 982, a determination is made as to whether the action selected by the user on the user terminal is to send an e-mail message to one or more e-mail addresses that allow the receiver(s) to playback the current media and the current play-list played by the sender at the time of the event. A user may send either an e-mail containing the URL, the play-list, the play-list name or a URL link. When the message receiver clicks on the link, his terminal will execute the media playback application and will start playing back the media and the play-lists played at the time of the message-sending event. If the determination is positive, then in step 984 the user terminal prompts the user for an one or more e-mail addresses, and then prompts the user to transmit the e-mail. The user

LINKPLAY EXHIBIT 1004
Page 3412 of 4822

need not have an e-mail client software application for the operation to succeed. The e-mail may be directed to terminals having a streaming media playback component and Internet access.

Preferably, the e-mail message is directed to a user having a user terminal that communicates with server-side modules so that the recipient user terminal is automatically plays back media from the transmitted media link received upon the e-mail being opened. Thus, server side modules may receive addresses from the terminal and act as the source of the email addressed to one or more recipients.

In step 986, a determination is made as to whether a user wishes to rate a media played back from the media links. If the determination is positive, then the user is prompted to rate the media in step 988. A rating value is transmitted to a backend web rating system application using the Internet. This operation is part of the service rating system that includes server side modules that produce, in combination with this operation, top and bottom rating media in each media category.

Step 990 illustrates other exemplary actions that may be received from a user on the user terminal interfacing with the playback application. For example, user actions may correspond to pausing media playback, adjusting volume, picture controls, size, seeking within the media, etc. In step 992, playback settings are changed according to user input. In an embodiment, the flow process returns to step 945 to check for another event if any of the determinations in step 982-992 are negative.

E. Rating System for Media Play Back

An embodiment of the invention includes a rating system with a media search and playback system. The rating system allows users to listen or view media segments available over the Internet according to a rating. In addition, users participate in determining rating media segments by providing a rating input after listening or viewing a media clip. The ratings may be used as a category, similar to other categories such as genres or categories.

FIG. 12 illustrates an architecture for use with the rating system, under an embodiment of the invention. The rating system 1000 may be employed

LINKPLAY EXHIBIT 1004
Page 3413 of 4822

with, for example, the system 200 (FIG. 2). The rating system 1000 may include or cooperate with components of a system such as described with FIG. 2 to enable the user view and/or select media clips from play-lists. The generated play-lists may contain a list of links to media on the Internet. Selection of media clips by the user causes media to be played back to the user over the user terminal.

The rating system 1000 includes a backend database management component 1045. The database management component 1045 maintains organizational data structures such as tables that describe rating information for media clips. The media clips include Internet streaming audio or video. The rating information may be in the form of values such as, for example, total votes counted. In an embodiment, database management component 1045 maintains records that comprise meta information on each media clip including the URL to the media clip, the current rating of the media clip, and the total votes for that media clip.

A user 1010 on a user terminal interacts with a web-based playback interface 1020. As an example, play-back interface 1020 outputs play-lists 1018, 1022 to the user. The media clips in each play-lists may be outputted automatically, or displayed for the selection of the user. The play-back interface 1020 may also display to the user genre field 1016 or category field 1014 of the selected media clip, or play-list 1018, 1022. The playback interface 1020 includes features to enable a user on the user terminal to make entries or selections regarding preferences and opinions, as well as other types of information. The user may also view ratings stored on backend database 1045. The user may enter selections by, for example, using icons or other display features. The user may make entries by, for example, inputting text or voice. FIG. 12 illustrates a rating selection component 1012 as a feature of play-back interface 1020. As an example, the rating selection component 1012 allows users to rate a media output between a scale of 1 to 5.

A user 1010 may input a rating to play-back interface 1020. The rating is signaled from play-back module 1020 to a rating module 1030. In an embodiment, rating module 1030 maintains a tally for each media clip. The tally compiles ratings received from play-back module 1020. The ratings may

LINKPLAY EXHIBIT 1004
Page 3414 of 4822

be received from more than one user and/or user terminal. The tally may be implemented through a protocol that enables the rating module 1030 to organize media clips according to an order. The organization of the media clips may correspond to a user preferential list, where preferred media clips are, for example, listed together or listed before less preferred clips. The rating module 1030 may also determine a genre, category, or other organization information through selections or entries received from the play-back module 1020. The selections may be tallied through any protocol, such as summation, averages, weighted averages and moving averages. In another embodiment, the rating module 1030 may maintain a text field to store user comments regarding each media clip.

In an embodiment, the rating component 1030 updates the rating information maintained in the database management component 1045. For example, the rating component 1030 may update values of the current rating and total votes for each media link.

A play-list generator 1040 generates play-lists based on rating information maintained in the database management component 1045. The play-list generator 1040 may signal to retrieve or receive records for each media clip. The play-list generator 1040 then automatically generates one or more play-list 1042. As previously discussed, each play-list is a list of media links. In an embodiment, the play-lists 1042 are generated according to the current rating and/or rating for each media clip. The generated play-lists are provided by the play-list generator 1040 for the play-back interface 1030.

The user 1010 may choose to listen to play-lists containing media clips rated according to one or more criteria. The play-lists may also be organized according to other factors, such as genre and category.

FIG. 13 is a flow chart that allows a user to listen to media clips that are rated according to one or more criteria. In step 1110, the user is provided a user-interface that allows users to receive media sorted according to one or more categories. The categories correspond to genres, such as type of music etc.

In step 1120, the user selects a category from the options presented by the user-interface. In response to the selection, the user terminal is provided

LINKPLAY EXHIBIT 1004
Page 3415 of 4822

one or more play-list in step 1130. The play-list received by the user-terminal matches the category or genre selected by the user. Further, play-lists contain predetermined media links to media clips. The media clips in each of the play-lists are determined according to a rating system, using a system such as described by FIG. 12. The predetermined play-list may correspond to a play-list generated by play-list generator 1040 (FIG. 12). Once the play-list is received by the user terminal, the flow process returns to step 1120.

In step 1140, media clips are played back on the user terminal. The media clips are played back consecutively and automatically, so that the user experiences continuous media playback. For example, the play-lists may contain numerous media creations from a selected genre. The media creations may be determined for the play-list according to a rating formula. The user is provided the media creations of the selected genre continuously, so that the user's media experience resembles listening to an album.

FIG. 14 illustrates a flow process for updating a rating of a media clip, under an embodiment of the invention. In step 1210, a module is provided a rating event. The rating event is a rating for a particular media clip, having an associated URL. The rating from the user is predefined from a closed set. For example, the user may provide a rating from 1 to 5.

In step 1220, a record is located for the media clip that was currently rated. The record may be stored in a database, and include the media link for the media clip, the current rating of the media clip, and the votes received for that media clip. In an embodiment, the record may include more than one URL associated with the media clip that was rated. In an embodiment, the record is maintained in database management component 1045 (FIG. 12).

In step 1230, a rating field for the media record is updated. The rating field may correspond to the current rating of a media clip. A module such as the rating module 1030 may update the rating field in database management component 1045 (FIG. 12). The media record is updated to determine a new rating. In one embodiment, the new rating is an averaged based formula. The formula may also be weighted. An example of a formula to determine a rating, under an embodiment of the invention is:

Newrating= 1/(n+1) (N*(old rating)+user provided rating)

N is the total number of votes received, and newrating ranges between 0 and a maximum value.

In step 1240, record for the media clip rated is further updated to add an additional vote to the field for votes received. The flow process then returns to step 1210.

FIG. 15 illustrates an exemplary structure for a database to maintain updated records on ratings and votes tallied. The table may associate values corresponding number of votes, rating, and other information to a media link containing a media clip.

FIG. 16 illustrates a flow process for generating media clips into play-lists according to a rating criteria. Play-lists including a rating criteria are referred to as rated play-lists. The flow process assumes known categories for media clips. The flow process also assumes a rating for rated media clips, and the number of media clips in a rated play-list. The flow process may be used with any of the aforementioned embodiments.

In step 1410, a next category may be fetched from a database containing the different categories of media clips. In step 1420, the system makes a determination as to whether a category was received. If no category is received, the system assumes that there are no more media categories to be rated.

In step 1430, a new play-list is created for a current category. In step 1440, up to N rated media clips from a database of rated media clips are added to the play-list. Preferably, N is a constant in the flow process. Then in step 1450, an old play-list is deleted, and the new play-list is saved. The new play-list may be saved in a format that follows predefined protocol so that the play-list and its contents are accessible to a streaming media play back interface. The flow process then returns to step 1410.

FIG. 17 illustrates a flow process for programmatically categorizing media files. The process assumes a database containing metadata associated with media clips. The metadata includes metadata provided by a human editor. For example, the metadata may pertain to categories such as genre, mood and atmosphere.

LINKPLAY EXHIBIT 1004
Page 3417 of 4822

In step 1510, a record is retrieved from the database. The record is retrieved with metadata information containing a first type of metadata information and a second type of metadata information. As an example, the first type of metadata information may correspond to a genre of music, and the second type of metadata information may correspond to an artist. In step 1520, a determination is made as to whether a record was received. If the determination is negative, then the process assumes that all media clips have been categorized.

If the determination in step 1520 is positive, then all records in the database having the first type of metadata information are retrieved in step 1530. In step 1540, all records retrieved in step 1530 are updated to include the second type of metadata information. As an example, all records belonging to a particular artist (first type of metadata information) or given additional metadata information of a particular genre (second type of metadata information). The process then returns to step 1510 to retrieve another record.

In one embodiment, the second type of metadata information is a genus category, and the first type of metadata information is a species of the first type of metadata information. Once the first record is known to have the a particular species and genus, the genus may be determined and stored for all records having the same species.

F. Personalized Media Playback

FIG. 18 is a flow process to create personalized play-lists of streaming media files available on the Internet (or other networks). The play-lists may be personalized by users on user-terminals.

In step 1610, a user chooses to add a URL of a selected media clip to a personal favorite play-list. In step 1620, the flow process adds the URL (and metadata) of the selected media clip to a user terminal store for user persistent information, such as an Internet cookie. The persistent data store is then accessible for the web-based play back application on the user terminal.

In step 1630, the user selects to play back media clips from that user's favorite play-list. In step 1640, the system reads back the media clips from the persisted data store. In step 1650, the system plays media clips using a URL

LINKPLAY EXHIBIT 1004
Page 3418 of 4822

associated with each media clip. The cookie may also provide additional URLs. Thus, multiple media clips may be played continuously from different sites on the Internet.

The user may edit the play-list, change an order of the play-list, or delete selections from the play-list. The user may designate certain play-lists as personal, so as to identify the play-list with that user's terminal. Alternatively, the play-list may be stored on a network server and accessed using the media location and playback module. Users may access their personal play-lists from any one of a plurality of terminals that have access to the system.

G. Distributed Architecture

An implementation under an embodiment provides a distributed architecture in which a user terminal accesses media resources from a plurality of network sites. In a network such as the Internet, the user terminal accesses multiple web sites to playback media locates as files on those web sites.

A network site includes any network location having internal links. Embodiments of the invention access network sites providing links to media files and/or other network sites. A web site refers to a network site on the Internet. Examples of web sites include web pages, including web pages with HTML links to other web sites, to media files, and to other types of files.

The distributed architecture inverts conventional media distribution paradigms. Numerous streaming media files can be streamed to an individual user terminal continuously from throughout the Internet using the embodiment of the distributed playback architecture. The distribution architecture is scalable to provide thousands or millions of streaming media files to user terminals. The users can then play media files located throughout the Internet in a continuous manner from the numerous Internet sites.

FIG. 19 illustrates a distributed playback architecture, under an embodiment of the invention. A user terminal 1710 has access to N network sites that provide access to media, also referred to here as media sites. The N media sites 1722 via web server module 1770. The media sites 1722 each have one or more links to media web resources. The links are represented by URLs 1-N. The web server module 1770 can load the media resources onto a media

LINKPLAY EXHIBIT 1004
Page 3419 of 4822

playback component of user terminal 1710. Once loaded, the media resources are played back by a media playback component on user terminal 1710.

In an embodiment, the media sites 1722 correspond to different locations on a network such as the Internet. For example, media sites may have different Internet addresses, including different domains. Each media site provides direct access to a media network resource. This implies that a URL (or link) to one of the media sites accesses the media network resource for playback without accessing another internal URL (or link).

In an embodiment, web server module 1770 signals multiple URL links to user terminal 1710. The media playback component of user terminal 1710 accesses each link to playback the media resource. The URLs are selected for media playback so as to output media from user terminal 1710 according to a predetermined program.

In an embodiment, the program is selected or defined from a search request of user. For example, a search request may designate a category for media output, such as a genre and artist. All URLs containing media from that artist and genre may be gathered and signaled from web server module 1770 to user terminal 1710. The URLs may be provided in any order, such as random, etc. or a chronological order of the artist.

In another embodiment, a program may be provided by one or more play-lists. Each play-list in the program may be generated by, for example, a play-list generator 1040 (FIG. 12). The play-lists may be personalized for the user of the end terminal 1710. For example, play-lists may be generated for preferences and profiles specified by the user of user terminal 1710. As another example, a user may couple to the Internet, prompting web server module 1770 to automatically signal one or more play-lists containing the URLs to user terminal 1710. Still, other embodiments provide for URLs to web resources, or play-lists containing the URLs to be randomly provided to user terminal 1710.

Among other advantages, the distributed architecture permits simultaneous playback of, for example, thousands or millions of multiple streams which do not congregate on a single point. This avoids congestion arising under examples of the current media paradigms. This ensures that the embodiment of our distributed architecture may "scale," or permit the

LINKPLAY EXHIBIT 1004
Page 3420 of 4822

simultaneous playback of, for example, thousands or millions of simultaneous streams. Further, the quality of the user experience is not affected by scaling a system under the distributed architecture embodiment.

In contrast, conventional broadcasting employs one radio or television signal to broadcast to listeners or viewers. Media files disseminated over the Internet today may be distributed in a manner which is somewhat similar in that the media file is located on a single server (or small group of servers) which is accessed by potentially large number of Internet users. As a result, the experience of the users may diminish due to the limited ability of current systems to scale.

This distributed playback architecture, the delivery of streaming media through this playback architecture, in combination with the search functionality performed by the back-end module, and the rating and personalization features of the playback client terminal module permits the creation of a broadcasting system that is personalized by an end user. A personal broadcasting system permits each individual user to create media programs which can be sent to, for example, thousands or millions or other users who can simultaneously play different programming combinations using a distribution of Internet (or other network) sites.

An example of a distributed architecture playback system includes wireless devices that are communicatable to a network containing media resources. For example, media playback component 1710 may be loaded onto a wireless access protocol (WAP) enabled device. Examples of WAP enabled devices include handheld computes and cell phones. The WAP enabled device may use a wireless communication network to access network server module 1770. The WAP enabled device may also include output features, such as speakers or a display screen. The WAP enabled device accesses media sites 1722 by control of network server module 1770. The WAP enabled device then plays back media from the media sites 1722. The WAP enabled device may then be used to simulate a portable radio.

As another example, an automobile may be equipped with a wireless device. The wireless device accesses multiple media sites on a network such as the Internet to and provide playback of media clips. For example, the user may

LINKPLAY EXHIBIT 1004
Page 3421 of 4822

select to hear music from a favorite play-list using the WAP enabled device in the automobile.

H. Messaging Applications

FIG. 20 is a diagram illustrating a messaging application, under an embodiment of the invention. The messaging application enables the user to share a media playback experience with other users having access to the network.

FIG. 20 assumes the messaging application is operated on a network such as the Internet. In the embodiment, a network interface or network-side module is used to enable messaging, rather than a client messaging applications. Examples of messaging applications for use with embodiments include e-mails, which are delivered to a folder on a recipient's terminal. Other types of electronic messages include instant messages, which can be displayed or heard on the recipients terminal automatically upon arrival.

A messaging module 2080 receives a messaging request from a first user terminal (sender) 2010. The messaging module 2080 may be an application or portion of network server module 2070. The messaging request is entered by the user through the user-interface 1900, using for example, an e-mail selection field 1990.

The messaging module 2080 receives addresses to deliver messages to recipient terminals (recipient) 2020. The sender 2010 may manually signal the recipients address using entry methods such as keyboards, graphic user selection features, or audio commands. Alternatively, messaging module 2080 has access to network stored addresses for the specific user. The network stored messages are then selectable from a terminal by the user on the sender terminal 2010.

In response to a request from sender 2010, messaging module 2080 generates a message 2085 for the recipient. The request may also include the address of the intended recipient(s). The message 2085 is sent to all recipients 2020 specified by sender 2010. The contents of the message 2085 include a URL to the network server module 2070. In an embodiment, the URL in the message is packaged with arguments or other coding to identify a play-list

maintained on the network server module 2070. The URL may also be packaged with arguments to identify the specific song being played when the sender causes the message to be transmitted to the recipient. Alternatively, the URL may identify to the recipient the search request or criteria used by the sender.

The recipient may choose to return a message 2082 to signal messaging module 2080. In one embodiment, the content of the message is constructed so that once the message is opened, the user can select a link to a module that stores or maintains the play-list 2075. For Internet applications, the link is HTML formatted to include the URL of network server module 2070, and arguments to identify the play-list selected by sender 2010. Once network server module 2070 is accessed by the recipient 2020, arguments 2088 contained with the link identify the specific play-list 2075 experienced by sender 2010 when the request to send the message was made.

If the message is instant, the recipient 2020 can respond immediately to simultaneously share the experience of sender 2010. Alternatively, the recipient 2020 can be made to respond automatically upon the recipient 2020 receiving the message 2085, so as to enable sender 2010 and 2020 to simultaneously or concurrently share the same media playback.

The argument 2088 that is packaged with the link may also identify individual media clips, and/or an entire play-lists. The recipient 2020 is able to experience media playback from individual media clips selected by sender 2010.

In another embodiment, message 2085 is constructed so that once the message is opened, the user is automatically connected to network server module 2070. The message may be an e-mail, stored in a designated folder of recipient 2020. The e-mail may include an HTML formatted URL to cause the recipient's terminal to access and communicate with network server module 2070. The HTML formatted URL may also include code that causes the user terminal to automatically access network server module 2070 upon the e-mail being opened. The HTML formatted link may also include arguments to specify the play-list 2088 and/or media clip identified by the sender, as well as other parameters of the URL. Once the play-list or media clip is identified by

LINKPLAY EXHIBIT 1004
Page 3423 of 4822

network server module 2070, the recipient 2020 is able to share the media playback experience of the sender 2010. The sender 2010 can experience the media playback at the time sender 2010 requests the message to be sent.

In the embodiments shown, sender 2010 can select a media program that is signaled transmitted to recipient 2020 by messaging module 2080. The program may correspond to one or more play-lists, playing back multiple media resources. While sender 2010 is being played back the program, the sender 2010 can specify the address of recipient 2020 to messaging module 2080. The messaging module 2080 generate a message that includes a selectable link to enable the recipient to access the network server module 2070. Arguments or scripting contained with the URL identify the particular play-list being signaled to the sender 2010. Since the play-list is updated after every media resource is played back to sender 2010, recipient 2020 accesses the play-list at the selection being played back to the sender 2010. The play-list is then signaled to the sender 2010 and recipient 2020 simultaneously, or approximately thereabout. Alternatively, recipient 2020 accesses the play-list beginning with the media clip played back to sender 2010 when the sender 2010 selected to transmit the message to recipient 2020.

As an example, sender 2010 transmits a search request causing media, based on search criterias of a specific artist and a ranking. The network server module 2070 identifies URLs matching the search request and forms play-list 2075 for sender 2010. After reviewing the play-list 2075, the sender 2010 decides to share the media playback with a friend, recipient 2020, whom the sender believes would appreciate play-list 2075. The sender 2010 requests messaging module 2080 to transmit message 2085 to recipient 2020 by submitting the recipient's e-mail address to the messaging module 2080. The message 2085 sits on the recipient's terminal until accessed. The recipient 2020 selects message 2085 to access play-list 2075. Unless the messaging is nearly instantáneous, the recipient experiences media playback from the play-list 2075 when sender 2010 request message 2085 to be sent. Alternatively, sender 2010 may request the entire play-list 2075 to be transmitted to recipient 2020. In this way, sender 2010 and recipient 2020 may share a common interest in certain genres, category, artists etc. of media playback.

LINKPLAY EXHIBIT 1004
Page 3424 of 4822

I. User-Interface

FIG. 21 illustrates a user-interface 1900, under an embodiment of the invention. The user-interface 1900 may correspond to the play-back interface 1020, described with FIG. 10. Alternatively, the user-interface may be a terminal side component in communication with one or more server-side modules, such as for example web server module 1070 (FIG. 12).

In an embodiment such as shown by FIG. 12, user-interface 1900 includes a plurality of user-interactive features. The user-interactive features enable users to interact with the system 200 from the user terminal 210. Some of the plurality of user-interactive features allow users to submit search requests and other media requests for playback. Other control user-interactive features allow users to affect the play back of the media resources.

The user-interface 1900 may output to the user information, images, and/or audio that is different than the media resource being played back. For example, user-interface 1900 displays metadata information to the users about the media resource being played back. In addition, the user-interface 1900 enables users to, for example, view advertisement, receive electronic messages, and create and manage play-lists.

In an embodiment such as shown by FIG. 21, the user-interface 1900 includes a first menu field 1910, a second menu field 1920, and a third menu field 1930. The first menu field 1910 allows the user to select a first criteria for media resources that are to be played back. The second menu field 1920 allows the user to select a second criteria from a set of media resources matching the first criteria. The third menu field 1930 allows the user to select a third criteria from the a set of media resources matching both the first and second criteria. Each of the menu fields 1910-1930 may be in the form of click and drag-down menus.

The user-interface 1900 may also include a text field 1940. The text field 1940 allows users to enter a search criteria. The search criteria entered in text field 1940 may be combined with the search criterias of one or more menu fields 1910-1930 using a Boolean operation. Preferably, all search criterias are AND together into a single search criteria. For example, the search criteria

entered in text field 1940 may correspond to an artist name, or a title of a media creation.

In an embodiment, the user-interface 1900 provides features that prompt a user for input, such as for one more search criterias. The web server module 270 receives the search criteria(s) signaled from user terminal 210, and access media and metadata table 247. Each of the menu field 1910-1930 may also allow users to enter text field as the search criterias. The search criteria(s) are matched to URLs containing metadata having the same (or equivalent) criterias. For example, the search request may specify a genre, and a first name of an artist. Then, web access server 270 locates URLs to media resources having associated metadata information that identifies the media resource as being of the same genre, and as containing the same first name in the artist name metadata indexed data structure.

The web server module includes a feedback display portion 1950. The feedback display portion 1950 may signal information, messages, advertisement etc. to the end user. In an embodiment, feedback display portion 1950 displays metadata information about the media resource being played back. For example, a song of a particular genre and category may be played back. The display screen portion 1950 may display the title of the song, the artist name, a play-list associated with the song, a rating component of the song, and the song's duration. Information is read when the media playback component loads the media resource.

Other user-interactive features may also be included in user-interface 1900. In an embodiment, user-interface 1900 includes a play-list feature 1960. The play-list feature 1960 enables users to add a media creation to a play-list. The play-list feature 1960 may, as an example, be a selectable icon. Upon selecting the play-list feature 1960, a pop-up window (not shown) may be displayed allowing a user to name or select the play-list that will include the media resource being played. In this way, a user of user terminal 210 can provide input to create and manage play-lists, using systems such as described with FIG. 12 and 19.

The user-interface 1900 may include one or more control user-interactive features. The control user-interactive features may be in the form of

LINKPLAY EXHIBIT 1004
Page 3426 of 4822

selectable icons. A skip feature 1972 causes, for example, web server module 270 to signal a URL of another media creation to the media playback component. This causes the media playback component to start playing back a new media creation. A pause feature 1974 enables users to pause the media playback component from playing back the media resource. The pause feature 1974 may signal the media playback component directly, or cause the web server module to signal the command to the media playback component. Reselecting the pause feature 1974 then causes the media creation to be played back from the portion where playback was paused. Similarly, a seek feature 1976 may signal to seek or move to a specific instance of playback on the media resource. A volume feature 1978 signals the application program interface 276 (FIG. 5) to raise the volume of the media resource being played back.

The user-interface 1900 may also include a rating feature 1980. The rating feature 1980 may be in the form of multiple selectable icons, where the icons are arranged to correspond to a rating. For example, five icons may be provided to represent best to worst ratings. In an embodiment, the rating feature 1980 enables a user to rate a media resource during or after it is played back on the user terminal 210. With reference to an embodiment such as described with FIG. 12, the rating feature 1980 is used to prompt a user to signal a rating to rating component 1030 (FIG. 12). The rating feature 1980 may be a user response to a media clip played back on user terminal 210. The rating component 1030 receives the rating and modifies rating information associated with the URL that is stored in 1045. The rating information may then be provided to other users or user terminals. For example, the rating information may then be signaled to display portions 1950 of other user terminals 210 who select that media clip for playback.

The user-interface 1900 also includes a personal play-list feature 1985. The personal play-list feature includes iconic selection features, including an add icon 1987 to add a URL to a play-list, and a play icon 1989 to play a personal play-list. The add icon 1987 enables a user to signal play-list generator 1040 (FIG. 12) to add a URL to the personal play-list. The URL being added to the play-list may correspond to a media resource being played back on user terminal 210. The play icon 1989 may be selected to cause web server module

LINKPLAY EXHIBIT 1004
Page 3427 of 4822

1070 (FIG. 12) to signal URLs from the personal play-list to the media playback component of user terminal 210. In this way, user terminal 210 may select to have continuous media output from resources previously selected to be on a play-list.

The user-interface 1900 may also include an e-mail selection feature 1990. The e-mail selection feature 1990 may be iconic, to allow selection by the user upon the media playback. Once selected, an e-mail program on user terminal 210 may be launched. The e-mail program may be directed to open a new message, and attach the URL of the selected media resource.

FIG. 22 illustrates an embodiment in which user-interface 1900 is displayed on the desktop along a second window 2210 showing a web site 2212. The web site 2212 hosts the media file being played back. In this embodiment, web server module 1070 signals a media file URL to the media player component of the terminal. The web server module 1070 concurrently signals the web browser component 213 on the terminal another URL to the hosting web site. The web browser 213 opens the second window 2210 to display the web site while the media from the media file is being played back on the terminal. In this way, users are displayed the web site hosting the media file while media from the media file is played back. This allows the user to view, for example, banner ads, artist name and titles, and copyright information while media from the web site is being played back.

After the playback is complete for one media file, a URL to a next media file is signaled to the media player component on the terminal. The next URL may be determined by a sequence of a play-list, or by a result to a search term inputted from the user. If the URL of the next media file is hosted on a web site that is different than the preceding web site, then web server module 270 signals the URL of the next hosting web site to the web browser. The second window 2210 then displays a second web site 2212' that hosts the media file being played back. In this way, the second window 2210 displays only web sites hosting the media files being played back.

LINKPLAY EXHIBIT 1004
Page 3428 of 4822

J. Conclusion

The foregoing description of various embodiments of the invention has been presented for purposes of illustration and description. It is not intended to limit the invention to the precise forms disclosed. Many modifications and equivalent arrangements will be apparent.

LINKPLAY EXHIBIT 1004
Page 3429 of 4822

CLAIMS

What is claimed is:

1. A method for playing back media from a network, the method comprising:

receiving a search criteria from a network enabled device;

accessing a database comprising a plurality of network addresses, the database associating each address with one or more classes of information, each address accessing a media resource;

selecting at least one address in the database using the search criteria;

signaling the selected address to the network enabled device; and

controlling the network enabled device so as to automatically access and play back the media resource of the selected address.

2. The method of claim 1, wherein selecting at least one address in the database includes matching the search criteria with a parameter in the one or more classes of information associated with that address.

3. The method of claim 1, wherein accessing a database comprising a plurality of network addresses includes accessing a plurality of links to Internet sites, each link containing no other internal link.

4. The method of claim 3, wherein selecting at least one address in the database using the search criteria includes selecting a plurality of links in the database using the search criteria, signaling the selected plurality of links to the network enabled device, and programmatically controlling the network enabled device so as to automatically access and sequentially play back the media resource of each of the plurality of selected links in a designated order.

5. The method of claim 4, wherein controlling the network enabled device so as to automatically access and sequentially play back the media resource includes sequentially playing back the media resource of each of the plurality of selected links after accessing the respective selected link.

LINKPLAY EXHIBIT 1004
Page 3430 of 4822

6. The method of claim 3, wherein receiving a search criteria from a network enabled device includes receiving the search criteria corresponding to an artist name, and wherein selecting at least one address in the database using the search criteria includes selecting a plurality of links that are each associated with a class of information containing a parameter corresponding to the artist name.

7. The method of claim 3, wherein receiving a search criteria from a network enabled device includes receiving the search criteria corresponding to a title of a network media resource, and wherein selecting at least one address in the database using the search criteria includes selecting a plurality of links that are each associated with a class of information containing a parameter corresponding to the title of the network media resource.

8. The method of claim 2, wherein receiving a search criteria from a network enabled device includes receiving the search criteria corresponding to a musical category, and wherein selecting at least one address in the database using the search criteria includes selecting a plurality of links that are each associated with a class of information containing a parameter corresponding to the musical category.

9. A method for playing back media from a network, the method comprising:

receiving a request for media playback entered on a network enabled device;

accessing a database comprising a plurality of network addresses, each address accessing a media resource;

identifying at least two addresses from the database;

signaling each identified address to the network enabled device; and

controlling the network enabled device to access and automatically play back the media resources of each of the signaled addresses.

10. The method of claim 9, wherein controlling the network enabled device to access the media resources of the signaled links includes signaling a uniform resource locator for one or more of the media resources to a media playback

LINKPLAY EXHIBIT 1004
Page 3431 of 4822

component to cause the network media resources of the signaled links to be accessed and played back on the network enabled device.

11. The method of claim 10, wherein receiving a request for media playback includes receiving the request for media playback from an Internet enabled device, and wherein the method further comprises signaling the uniform resource locator of the one or more media resources to a web browser component on the Internet enabled device to cause a web page to be displayed on the Internet enabled device while the media playback component plays back the media resource on the network enabled device.

12. The method of claim 11, wherein receiving a request for media playback from a network enabled device includes receiving a request for a play-list comprising a plurality of addresses, the play-list arranging the plurality of addresses so that the addresses are loaded by the media playback component in a designated order, wherein identifying at least two addresses in the database includes identifying the addresses in the play-list, and wherein controlling the network enabled device to access and automatically play back the media resources of the signaled addresses includes accessing and playing back each media resource in the play-list sequentially in the designated order.

13. The method of claim 12, wherein receiving a request for media playback from a network enabled device includes receiving a request for a program comprising multiple play-lists, each play-list being corresponding to a predetermined selection of media resources, wherein identifying at least two addresses includes identifying the addresses in each of the play-lists in the program, and wherein controlling the network enabled device to access and automatically playback the media resources of the signaled addresses includes accessing and playing back each media resource in a sequential order designated by the play-lists of the program.

14. A computer system for playing back media from a network, the computer system comprising:

a network enabled device comprising a media playback component;

a database comprises a plurality of addresses, each address locating a media resource on the network;

a network server module that communicates with the network enabled device and with the database, the network server module being able to receive a search request entered on the network enabled device that specifies one or more criterias, to select at least one address from the database that is associated with a class of information that matches the search criteria, and to cause the media playback component to load the media resource located by the selected address.

15. The computer system of claim 14, wherein the database stores one or more classes of information associated with each address in the plurality of addresses.

16. The computer system of claim 15, wherein the database is locatable by the network enabled device on the network.

17. The computer system of claim 15, wherein at least a portion of the database resides on the network enabled device.

18. The computer system of claim 14, wherein the network server module signals the selected address to the network enabled device to cause the media playback component to load the media resource located by the selected address.

19. The computer system of claim 18, wherein the network server module communicates with the network enabled device through the network.

20. The computer system of claim 14, wherein the network server module is located at least partially on the network enabled device and communicates with the database through the network.

21. The computer system of claim 14, wherein the network server module comprises a sequence of code that is signaled to the network enabled device from another device on the network, the sequence of code being executable on the network enabled device.

LINKPLAY EXHIBIT 1004
Page 3433 of 4822

22. The computer system of claim 14, wherein the network enabled device comprises the media playback component, and wherein the network server module programmatically controls the media playback component of the network enabled device to playback the media resource.

23. The computer system of claim 14, wherein the network enabled device is a network enabled device selected from a group of devices consisting of a personal computer, a handheld computer, a smart network appliance, and a wireless Internet enabled device.

24. The computer system of claim 14, wherein the database comprises a plurality of links to Internet sites, wherein each link accesses a media resource without requiring another internal link.

25. The computer system of claim 14, wherein all or a portion of the database resides on the network enabled device.

26. The computer system of claim 14, wherein all or a portion of the database is accessible to the network enabled device over the network.

27. The computer system of claim 14, wherein the one or more classes of information for each address in the database include types of metadata selected from the group consisting of genre, title, and data format of the network resource.

28. The computer system of claim 14, wherein the media resource is a streaming media file that outputs a combination of at least video or audio once the media resource is loaded by the media playback component.

29. The computer system of claim 14, wherein the network includes the Internet, the network enabled device includes an Internet enabled device, each of the plurality of addresses includes links to a media web resource, and wherein the network server module signals control information to the Internet enabled device to cause the Internet enabled device to programmatically access each of a plurality of media web resources signaled to the network enabled device.

LINKPLAY EXHIBIT 1004
Page 3434 of 4822

30. The computer system of claim 29, wherein the web server module signals control information to the media playback component and a web browser component of the Internet enabled device, to cause the media playback component of the network enabled device to programmatically load and playback the media web resource while the web browser displays a web page of a web site corresponding to the media web resource being played back.

31. A computer system for playing back media from a network, the computer system comprising:

a network enabled platform comprising a media playback component;

a database comprising a plurality of addresses, each address locating a media resource on the network, each address accessing a media resource; and

a network server module that coupleable to the media playback component and the database, the network server being able to receive a request for media playback from the network enabled device, to select multiple addresses from the database, to signal the multiple addresses to the network enabled device, and to control a media playback component on the network enabled device to use the addresses to automatically access and play back the media resource associated with the addresses.

32. The computer system of claim 31, wherein the network server module resides on the network.

33. The computer system of claim 31, wherein at least a portion of the network server module resides on the network enabled device.

34. The computer system of claim 31, wherein the network server module comprises a sequence of code that is signaled to the network enabled device from another device on the network, the sequence of code being executable on the network enabled device.

35. The computer system of claim 31, wherein the network enabled device is an Internet enabled device comprising the media playback component and a

LINKPLAY EXHIBIT 1004
Page 3435 of 4822

network browser component, the database comprises a plurality of links to media web resources, wherein the network server module programmatically controls the media playback component of the Internet enabled device to access and playback the media web resource of each link signaled to the Internet enabled device.

36. The computer system of claim 31, wherein the network enabled device further comprises a web browser component, and wherein the network server module programmatically controls the web browser component to display a web page associated with the media web resource being played back.

37. The computer system of claim 31, further comprising a play-list module coupleable to the network server module, the play-list module generating a play-list based on one or more criterias designated by a user of the network enabled device, the play-list comprising a plurality of addresses that are signaled to the network enabled device in response to the request for media playback.

38. The computer system of claim 31, wherein the network enabled device is a terminal device selected from a group of devices consisting of a personal computer, a handheld computer, a wireless Internet enabled device, and a handheld computer.

39. The computer system of claim 31, wherein the database comprises a plurality of links to Internet sites, wherein each link accesses a media web resource and contains no other internal link.

40. The computer system of claim 31, wherein all or a portion of the database resides on the network enabled device.

41. The computer system of claim 31, wherein all or a portion of the database is accessible to the network enabled device over the network.

42. A network enabled device configured to playback media from a network, the network enabled device being coupleable over to a database

LINKPLAY EXHIBIT 1004
Page 3436 of 4822

comprising a plurality of addresses, each address locating a media resource on a network, the network enabled device comprising:

a user-interface to prompt for a search request;

a network interface to signal the request to a network server module that is communicatable with the database, and to receive one or more addresses in the database that match the search request; and

a media playback component being configured to be programmatically controllable by the network server module to automatically load the media resource associated with each address signaled to the network enabled device upon accessing the media resource.

43. The network enabled device of claim 42, wherein the user interface is controllable by the network server module after the network server module signals the response, the user-interface providing one or more prompts to enable an end user to control the media played back from the one or more media resources associated with the addresses signaled from the network server module.

44. The network enabled device of claim 42, wherein the user-interface prompts for a user to signal the network server module to immediately access a next address signaled to the network enabled device from the network server module.

45. The network enabled device of claim 44, wherein the user-interface prompts for the end user to signal the media playback component to repeat playing back a previously played media resource.

46. The network enabled device of claim 42, wherein the user interface prompts for a specific media resource from the media resources signaled from the network server module for playback, and to pause playback from the media resources signaled from the web server module.

47. The network enabled device of claim 42, wherein the network access component is a web browser residing on an Internet enabled device.

LINKPLAY EXHIBIT 1004
Page 3437 of 4822

48. The network enabled device of claim 42, wherein all or a portion of the database is accessible to the network enabled device over the network.

49. The network enabled device of claim 42, wherein all or a portion of the database resides on the network enabled device.

50. The network enabled device of claim 42, wherein the network device is Internet enabled, wherein the network interface signals the request to a web server module that is communicatable with the database to receive one or more links to web sites containing media web resources, the one or more links containing no internal links, and wherein the Internet enabled device further comprises a web browser component, the web browser component being controllable by the web server module to automatically access and display a web page associated with the media web resource being played back by the media playback component.

51. The network enabled device of claim 50, wherein the network interface includes a communication port between the network enabled device and the network.

52. A network enabled device configured to play back media from a network, the network enabled device being coupleable over the network to a database comprising a plurality of addresses, each address locating a media resource on the network, the network enabled device comprising:

a user-interface including a plurality of user-interactive features, a first user-interactive feature in the plurality of user-interactive features prompting to receive a search request for media playback;

a network interface that signals the request to a network server module upon the first user-interactive feature receiving the search request for media playback, the network interface being communicatable with the database to receive one or more addresses in the database that match the search request; and

a media playback component being configured to be programmatically controllable by the network server module to automatically load the media resource associated with each address signaled to the network

enabled device upon accessing the media resource, a playback of the media playback component being controllable by one or more control user-interactive features in the plurality of user-interactive features.

53. The network enabled device of claim 52, wherein the first user-interactive feature in the plurality of user-interactive features is selected from a group consisting of selectable icons, text-entry field, and menu items.

54. The network enabled device of claim 53, wherein the control user-interactive features are selected from a group of selectable icons having functions corresponding to skipping to a next address selected to be signaled to the network enabled device, pausing the media resource being played back; repeating the media resource previously played back, and stopping playback from the media playback component.

55. The network enabled device of claim 53, wherein the user-interface includes a plurality of user-interactive features to select a characteristic of a media playback from the media playback component.

56. The network enabled device of claim 55, wherein the user-interface features include the plurality of user-interactive features to select a genre, an artist, or a title of a network media resource for playback .

57. The network enabled device of claim 56, wherein the search request is composed by selecting one or more of the user-interactive features of the user-interface.

58. A method for sharing media playback on a first terminal coupled to a network with other terminals coupled to the network, the method comprising:

signaling a link in a play-list to the first terminal, the play-list including one or more links to media files on the network;

receiving a signal from the first terminal to message a second terminal while the first terminal is playing back media from one of the media files;

LINKPLAY EXHIBIT 1004
Page 3439 of 4822

signaling the second terminal a selectable link to enable the second terminal to receive one or more links in the play-list; and

signaling the one or more links in the play-list to the second terminal upon the selectable link being selected by the second terminal.

59. The method of claim 58, wherein signaling the second terminal a selectable link includes messaging the second terminal the link to the play-list.

60. The method of claim 58, wherein signaling the second terminal a selectable link includes signaling a message to the second terminal that causes the second terminal to access the play-list once the message is opened.

61. The method of claim 58, wherein signaling the one or more links in the play-list to the second terminal includes signaling each link in the play-list to the first terminal and to the second terminal concurrently.

62. The method of claim 58, further comprising receiving the signal from the first terminal to message the second terminal while the first terminal is playing back media located by a link in the play-list, and causing the media playback component of the second terminal to load the media file being played by the first terminal.

63. The method of claim 58, wherein signaling a link to the second terminal that locates the play-list module includes signaling the link to include arguments that locate the address in the play-list corresponding to the media file being played back on the first terminal.

64. A system for sharing media playback from a network between a plurality of terminals, the plurality of terminals including a first terminal and a second terminal, the system comprising:

a play-list component locatable on the network by a selectable link, the play-list component identifying a plurality of links to form a play-list, each link in the play-list locating a media file on the network;

a network server module signaling the plurality of links that form the play-list to the first terminal, the network server module being able

LINKPLAY EXHIBIT 1004
Page 3440 of 4822

to receive a signal to transmit the selectable link to the second terminal to enable the second terminal to locate the play-list module.

65. The system of claim 64, wherein the network server module signals the plurality of links that form the play-list to the second terminal upon the selectable link being transmitted to the second terminal is selected.

66. The system of claim 64, wherein the messaging module transmits the selectable link as a content of a message.

67. The system of claim 64, wherein the play-list module transmits the selectable link with a corresponding message to the second terminal so that the selectable link is selected when the message opened.

68. The system of claim 65, wherein the play-list module identifies the plurality of links to form a plurality of play-lists, and wherein the messaging module transmits the selectable link with identifiers that identify the play-list being signaled to the first terminal.

69. The system of claim 68, wherein the messaging module transmits the selectable link with identifiers that identify the media file being played back on the first terminal.

70. The system of claim 69, wherein the network server module identifies the link in the play-list being played back on the first terminal and transmits the identified link with a remainder of the links in the play-list to the second terminal, so that the first terminal and the second terminal simultaneously playback the same media files while the links in the play-list are signaled to each of the first terminal and the second terminal.

71. A network enabled device comprising:

a media playback component configured to communicate with a network-side module to receive a first plurality of links, each of the first plurality of links locating a media file on a network;

a web browser component configured to receive a second plurality of links, each of the second plurality of links hosting a media file

located by one of the first plurality of links, the web browser component displaying the web site for each of the second plurality of links that corresponds to the media file being played back by the media playback component.

72. The network enabled device of claim 71, further comprising a user-interface including user-interactive features for controlling media output from media files located by the first plurality of links.

73. The network enabled device of claim 72, wherein the user-interface is displayed in a first window, and the web browser component displays the web site for each of the second plurality of links in a second window.

74. The network enabled device of claim 73, wherein the media playback component is configured to sequentially receive the first plurality of links from the network-side module, and the web browser displays the web sites hosting media files for each of the web sites in the sequence.

75. A system for playing back media from a network, the system comprising:

means for receiving a search criteria from a network enabled device;

means for accessing a database comprising a plurality of network addresses, the database associating each address with one or more classes of information, each address accessing a media resource;

means for selecting at least one address in the database using the search criteria;

means for signaling the selected address to the network enabled device; and

means for controlling the network enabled device so as to automatically access and play back the media resource of the selected address

76. A system for playing back media from a network, the system comprising:

means for receiving a request for media playback from a network enabled device;

means for accessing a database comprising a plurality of network addresses, each address accessing a media resource;

means for identifying at least two addresses from the database;

means signaling each identified address to the network enabled device; and

means for controlling the network enabled device to access and automatically play back the media resources of each of the signaled addresses.

77. A method to locate web resources on the Internet, the method comprising:

accessing a web site;

identifying a plurality of links using a web browser component, each of the plurality of links being selectable to open a corresponding web resource of a specified data type on the web site; and

making the plurality of links available to a plurality of Internet enabled devices that select one or more of the links.

78. The method of claim 77, further comprising arranging the plurality of links in a database that is accessible over the Internet to the plurality of Internet enabled devices.

79. The method of claim 78, further comprising identifying one or more classes of information about each of the plurality of links using a media playback component, and storing each of the plurality of links in association with the one or more classes of information.

80. The method of claim 79, further comprising making the one or more classes of information available to the plurality of Internet enabled devices.

81. The method of claim 77, wherein accessing a web site includes accessing a web site identified by a metacrawler.

LINKPLAY EXHIBIT 1004
Page 3443 of 4822

82. The method of claim 77, wherein identifying a plurality of links includes identifying links that only access web resources including audio data types, video data types, or a combination of audio and video types.

83. The method of claim 77, wherein identifying a plurality of links using a web browser component includes reconfiguring a commercially available web browser.

84. The method of claim 77, wherein identifying a plurality of links using a web browser component includes programmatically controlling the web browser component through an application programmable interface of a web browser to access the web resource for each of the plurality of links.

85. The method of claim 84, wherein programmatically controlling the web browser through an application programmable interface includes programming the web browser to bypass a shell of the web browser.

86. The method of claim 85, wherein programmatically controlling the web browser through an application programmable interface includes programming the web browser to reduce functionality of the web browser in displaying and translating data structures on each web site.

87. The method of claim 84, wherein programmatically controlling the web browser component through an application programmable interface further comprises using the web browser component to parse a web page that provides access to each link in the plurality of links to identify the plurality of links programmatically.

88. The method of claim 77, further comprising verifying that each of the plurality of links are selectable to open the corresponding web resource of the specified data type upon being selected.

89. The method of claim 88, wherein verifying that each of the plurality of links are selectable includes verifying that the web resources of each available on the network.

LINKPLAY EXHIBIT 1004
Page 3444 of 4822

90. The method of claim 89, wherein verifying that each of the plurality of links are selectable to open the corresponding web resource of the specified data type includes loading each of the plurality of media web resources into a media playback component.

91. A system for locating web resources on the Internet, the system comprising:

a web browser component;

a database;

a search module that controls the web browser component to access at least one web site, the search module controlling the web browser component to identify a plurality of links to media web resources at the web site, each of the plurality of links being selectable to open a media web resource, the search module storing the plurality of links in the database.

92. The system of claim 91, wherein the search module signals an application programmable interface of the web browser an address of the web site so as to cause the web browser component to load and parse the web site for the plurality of links to media web resources.

93. The system of claim 91, wherein the search module signals an application programmable interface of the web browser an address of the web site so as to cause the web browser component to load and parse the web site for the plurality of links to media web resources without displaying the media web resources.

94. The system of claim 92, wherein the search module signals the application programmable interface of the web browser the address of the web site so as to cause the web browser component to load and parse the web site for the plurality of links to media web resources of a video data type, an audio data type, or a combination of the audio and video data types.

LINKPLAY EXHIBIT 1004
Page 3445 of 4822

95. The system of claim 94, wherein the search module causes the web browser component to load and parse the plurality of links to media web resources of data types including animation and/or still images

96. The system of claim 91, further comprising a metacrawler module to locate a plurality of web sites for the search module.

97. The system of claim 91, wherein the search module controls a media playback component to extract metadata about the media web resources of each of the plurality of links.

98. A method to locate web resources on the Internet, the method comprising:

accessing a database including a plurality of links, the plurality of links being selectable to open a corresponding web media resources;

programmatically identifying metadata information about the web media resource corresponding to each of the plurality of links; and

making the plurality of links accessible to a plurality of Internet enabled devices, the plurality of Internet enabled devices being able to select one or more of the links to open the corresponding media web resource.

99. The method of claim 98, wherein programmatically identifying metadata information about the web media resource corresponding to each of the plurality of links includes signaling each link to a media playback component so as to cause the media playback component to load the corresponding media resource.

100. The method of claim 99, wherein signaling each link to a media playback component includes loading streaming audio media into the media playback component for sound recordings.

101. The method of claim 99, wherein signaling each link to a media playback component includes loading streaming video media in the media playback component for video recordings, including video recording having sound.

LINKPLAY EXHIBIT 1004
Page 3446 of 4822

102. The method of claim 100, wherein signaling each link to a media playback unit includes determining at least one of a duration of the web media resource, and a title of the web media resource.

103. The method of claim 100, wherein signaling each link to a media playback unit includes determining a broadcast quality of a web media resource.

104. The method of claim 100, wherein signaling each link to a media playback unit includes determining an image size of a web media resource having video or image.

105. The method of claim 100, wherein signaling each link to a media playback unit includes determining at least one of an artist, a genre, and a data type of the web media resource.

106. The method of claim 98, further comprising manually extracting the metadata information after programmatically identifying the metadata information.

107. The method of claim 106, further comprising storing the identified and extracted metadata information with each of the plurality of links in a database, and then making the database available to the plurality of Internet enabled devices.

108. A method to locate web resources on the Internet, the method comprising:

accessing a database including a plurality of links to media web resources;

programmatically verifying whether each of the plurality of links opens a corresponding web media resource; and

making each verified link in the plurality of links accessible to a plurality of Internet enabled devices that select one or more of the links to open the corresponding media web resource.

LINKPLAY EXHIBIT 1004
Page 3447 of 4822

109. The method of claim 108, wherein programmatically verifying whether each of the plurality of links opens a corresponding web media resource includes loading each link into a media playback component.

110. The method of claim 109, programmatically verifying whether each of the plurality of links opens a corresponding web media resource includes verifying each of the plurality of links makes available a web media resource for playback on one of the Internet enabled devices.

111. The method of claim 108, wherein programmatically verifying whether each of the plurality of links opens a corresponding web media resource includes loading each link into a web browser component

112. The method of claim 108, further comprising programmatically identifying metadata information about the web media resource corresponding to each of the plurality of links using the media playback component.

113. The method of claim 112, wherein programmatically identifying metadata information includes loading each of the verified links into the media playback component and then receiving editorial input about the web media resource loaded onto the media playback component.

114. A system for locating web resources on the Internet, the system being accessible by a plurality of Internet enabled devices, the system comprising:

a first indexed data structure comprising a plurality of links;

a media playback component coupleable to the database, the media playback component loading each of the plurality of links to verify whether the link is selectable to open a media web resource;

a second indexed data structure stores each verified link in the plurality of links, the second indexed data structure being available to the plurality of Internet enabled devices.

115. The system of claim 114, wherein the media playback component identifies one or more classes of information about the web media resource upon loading the link corresponding to the web media resource.

116. The system of claim 114, further comprising a search module that signals each of the plurality of links in the first indexed data structure to the media playback component, and wherein for each verified link, the search module signals the verified link to the second indexed data structure.

117. The system of claim 112, further comprising an editor interface to receive an editor input, the editor input including metadata information that was not identified from the media playback component loading the link.

118. The system of claim 117, further comprising an editor interface to receive an editor input, the editor input including metadata information identified from the media playback component loading the link.

119. The system of claim 118, further comprising an editor interface to receive an editor input, the editor input including metadata information identified from the media playback component loading the link, wherein the search module signals the metadata information about that verified link to the second indexed data structure, so that each verified link is associated with the corresponding metadata received from the editor input.

120. A method for providing links for use in a media search engine, the method comprising:

identifying a plurality of internal links on a network site;

determining whether the network site makes a network resource of a specific data type accessible for a network enabled device;

extracting the internal links that are selectable to open the network resource of the specific data type;

storing the external link in a database; and

automatically signaling one or more of the links to a media playback component in response to receiving a search request from the network enabled device.

121. The method of claim 120, wherein extracting the internal links that are selectable to open the network resource of the specific data type includes using a web browser component.

LINKPLAY EXHIBIT 1004
Page 3449 of 4822

122. The method of claim 120, wherein determining whether the network site makes a network resource of a specific data type accessible for a network enabled device includes using a media playback component.

123. The method of claim 120, including extracting the internal links that are selectable to open the network resource of the specific data type for playing back media.

124. A method for providing links for use in a media search engine, the method comprising

a) receiving from a first indexed data structure a first external link to a first network site;

b) initializing a second data structure to be empty;

c) determining if the first network site contains at least one internal link; and

d) storing the at least one internal link contained on the first network site that is not in the first indexed data structure and not in the second indexed data structure as another external link in the first indexed data structure;

e) identifying the internal links contained on the first network site that are selectable to open a network resource of a specific data type or types;

f) moving the first external link from the first indexed data structure to the second indexed data structure; and

g) repeating steps a) through f) until the first indexed data structure is empty.

125. The method of claim 124, wherein prior to receiving from a first indexed data structure a first link to a first site, adding one or more of the first plurality of links to the first indexed data structure using an editor interface.

126. The method of claim 125, wherein prior to receiving from a first indexed data structure a first link to a first site, adding one or more of the first plurality of links to the first indexed data structure using a metacrawler.

LINKPLAY EXHIBIT 1004
Page 3450 of 4822

127. A computer system to search for links to streaming media playback on a network, the network being accessible to a network enabled device, the computer system comprising:

a metacrawler to locate one or more media sites containing streaming media;

a media search module coupled to the metacrawler be signaled the one or more media sites, the media search module being able to identify a plurality of media links located by the one or more media sites, each of the plurality of media links being selectable to open streaming media network resource;

a metadata extraction module that accesses each media link identified by the media search engine to extract metadata about the identified media link; and

a database comprising the plurality of media links identified by the media search engine, and the metadata extracted about each identified media link, the database enabling the network enabled device to access the plurality of media links.

128. The computer system of claim 127, wherein the metacrawler locates one or more media sites in media directories, each media directory comprising a plurality of media sites.

129. The computer system of claim 127, where the metadata extraction module comprises a media playback component that loads the streaming media network resource.

130. The computer system of claim 127, wherein the metadata extracted from each identified link is sorted in the database according to one or more classes of information.

131. The computer system of claim 130, wherein the metadata extracted is sorted into one or more classes of information selected from classes consisting of genre, sub-genre, artist, and title.

132. The computer system of claim 131, further comprising a verification module that verifies each media link identified by the media link module opens the corresponding streaming media network resource.

133. A computer system to search for links to streaming media playback on a network, the network being accessible to a network enabled device, the computer system comprising:

means for accessing a web site;

means for identifying a plurality of links using a web browser component, each of the plurality of links being selectable to open a corresponding web resource of a specified data type on the web site; and

making the plurality of links available to a plurality of Internet enabled devices that select one or more of the links.

134. A system for providing media from a network to a terminal, the terminal including a media playback component, the system comprising:

a first network site and a second network site, each network site locating one or more media network resources, each media network resource being locatable on the network by a corresponding address that accesses the media network resource; and

a network server module that communicates with the media playback component, the network server module identifying a first media network resource from the first network site and a second media network resource from the second network site, the network server module signaling the corresponding address of the first media network resource to the media playback component with control signals to cause the playback component to automatically load the first media network resource, the network server module automatically signaling the corresponding address of the second media network resource to the media playback component with control signals to cause the playback component to automatically load the second media network resource.