

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
16 August 2007 (16.08.2007)

PCT

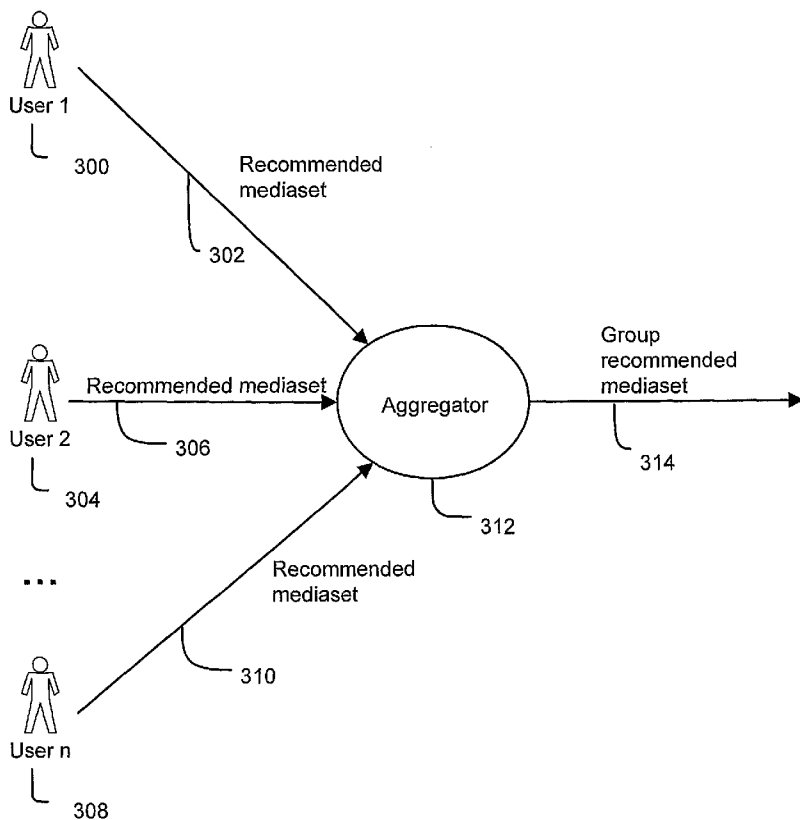
(10) International Publication Number
WO 2007/092053 A1

- (51) International Patent Classification:
G06F 17/30 (2006.01)
- (21) International Application Number:
PCT/US2006/034218
- (22) International Filing Date: 31 August 2006 (31.08.2006)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
60/772,502 10 February 2006 (10.02.2006) US
60/774,072 15 February 2006 (15.02.2006) US
60/796,724 1 May 2006 (01.05.2006) US
- (71) Applicant (for all designated States except US): **MUSIC-STRANDS, INC.** [US/US]; 760 SW Madison, Suite 106, Corvallis, Oregon 97333 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **MARTIN, Francisco J.** [ES/US]; 760 SW Madison, Suite 106, Corvallis, Oregon 97333 (US). **SHUR, Jim** [US/US]; 760 SW Madison, Suite 106, Corvallis, Oregon 97333 (US).

- TORRENS, Marc** [ES/US]; 760 SW Madison, Suite 106, Corvallis, Oregon 97333 (US). **HANGARTNER, Rick** [US/US]; 760 SW Madison, Suite 106, Corvallis, Oregon 97333 (US). **CADEVILLA-LALIENA, Guillermo** [ES/US]; 760 SW Madison, Suite 106, Corvallis, Oregon 97333 (US). **DEL SER BARTOLOME, David** [ES/US]; 760 SW Madison, Suite 106, Corvallis, Oregon 97333 (US). **ROWLEY, Craig** [US/US]; 760 SW Madison, Suite 106, Corvallis, Oregon 97333 (US).
- (74) Agent: **STOLOWITZ, Micah D.**; Stoel Rives LLP, 900 SW Fifth Avenue, Suite 2600, Portland, Oregon 97207 (US).
- (81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU,

[Continued on next page]

(54) Title: DYNAMIC INTERACTIVE ENTERTAINMENT



Aggregating individual recommended mediasets into a recommended mediaset suitable for the whole group of users.

(57) Abstract: Disclosed are various embodiments of systems and methods for generating composite mediasets from mediasets, each comprising media items, associated with a plurality of users. In some embodiments, individual and/or group recommendations are provided for creating a group playlist by aggregating user taste data for a plurality of users in a group. In other embodiments, systems and methods are provided which allow for sharing and playing of a group playlist by users in a group, each of which has a media playback device. In still other embodiments, systems and methods are provided for dynamically building a composite playlist that may be updated as users enter and/or leave the group. One example of an interactive entertainment application also is described for use at a venue to entertain live guests with music selections that are dynamically influenced by the guests' individual tastes.

WO 2007/092053 A1



SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR,
TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA,
GN, GQ, GW, ML, MR, NE, SN, TD, TG).

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT,

Published:

— *with international search 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.

Patent Cooperation Treaty

PATENT COOPERATION TREATY

DYNAMIC INTERACTIVE ENTERTAINMENT VENUE

Priority Claims

[0001] This application claims priority based on U.S. Provisional Patent Application No. 60/772,502 filed February 10, 2006, and titled "System and Method for Building and Sharing a Composite Playlist from Collective Group Tastes on Multiple Media Playback Devices." This application also claims priority based on U.S. Provisional Patent Application No. 60/774,072 filed February 15, 2006, and titled "Mediaset Recommendations for a Group of Users." This application also claims priority based on U.S. Provisional Patent Application No. 60/796,724 filed May 1, 2006, and titled "Dynamically Building Composite Playlist for Merging Collective User Tastes". The right of priority is claimed pursuant to the Paris Convention for the Protection of Industrial Property of March 20, 1883, as revised and amended.

Brief Description of the Drawings

[0002] Understanding that drawings depict only certain preferred embodiments of the invention and are therefore not to be considered limiting of its scope, the preferred embodiments will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

FIG. 1 is a diagram of one embodiment of a system for synthesizing a user's taste by analyzing the user's playlist(s) and playcounts.

FIG. 2 is a diagram of one embodiment of a system for using taste analyses from a plurality of users to generate a plurality of recommended mediasets.

FIG. 3 is a diagram of one embodiment of a system for using recommended mediasets from a plurality of users to generate a recommended mediaset for a group of users.

FIG. 4 is a system diagram of a client-server embodiment of the invention.

FIG. 5 is a system diagram of a peer-to-peer embodiment of the invention.

FIG. 6 depicts the main functional components of a Device Manager according to one embodiment.

FIG. 7 depicts a Member Status screen of a Session Manager, which presents the relevant details of the data structures in the Session Manager subcomponent of the Device Manager of one embodiment.

FIG. 8 is a top-level flow diagram of the operation of the Session Manager subcomponent of the Device Manager of one embodiment.

FIG. 9 is a more detailed flow diagram of the "Prune Session Members" step in the Session Manager flow diagram of FIG. 8.

FIG. 10 is a more detailed flow diagram of the "Update Session Members" step in the Session Manager flow diagram of FIG. 8.

FIG. 11 is a more detailed flow diagram of the "Service Proxy Requests" step in the Session Manager flow diagram of FIG. 8.

FIG. 12 depicts a Playlist Queue screen of a Playlist Manager, which presents the relevant details of the data structures in the Playlist Manager subcomponent of the Device Manager of one embodiment.

FIG. 13 is a flow diagram showing the operation of the Playlist Manager subcomponent of the Device Manager of one embodiment.

FIG. 14 is a flow diagram showing the operation of the Playlist Play subcomponent of the Playlist Manager of a Media Player Client embodiment.

FIG. 15 depicts the relevant data structures in one embodiment of the Media Player Client supporting the Playlist Builder component of the system.

FIG. 16 presents a conceptual diagram of the playlist building function of one embodiment.

FIG. 17 is a flow diagram of the operation of one embodiment of the Playlist Builder component of the system.

FIG. 18 is a diagram of a process for generating a list of media items that define a user taste.

FIG. 19 is a diagram of a process for computing aggregate playlist goals and a list of desired categories.

FIG. 20 is a diagram of a process for computing relevant media items for each user from total available media items.

FIG. 21 is a diagram of a process for categorizing relevant media items according to desired attributes.

FIG. 22 is a diagram of a process for computing the status of the current group playlist's achievement of compositional goals.

FIG. 23 is a diagram of a process for choosing a set of media items to add to a group playlist based on assessing needs according to compositional goal achievement assessment.

FIG. 24 represents a step of a user addition process for detecting new users.

FIG. 25 represents another step of a user addition process in which new users are added to the system and source device information is stored.

FIG. 26 is a diagram of a process for removing users from the system.

FIG. 27 is a simplified flow diagram illustrating one example of a process for a commercial establishment or venue to tailor its offerings to the tastes of an individual customer.

FIG. 28 is a simplified flow diagram illustrating one example of a process for recommending a commercial establishment or venue to a potential customer or group.

FIG. 29 is a conceptual software diagram illustrating a recommender core engine, various algorithms that may employ the core engine, and illustrative applications that may employ one or more of the various algorithms.

FIG. 30 is a simplified diagram illustrating some of the features and equipment that may be found at a venue and employed in accordance with aspects of the present invention to provide dynamic, interactive entertainment at the venue that is responsive to user tastes in real time.

FIG. 31 is a simplified illustration of one example of venue display screen content including an advertisement, host welcome message, pictures and instructions for a user to join the party using a wireless telecom messaging service.

FIG. 32 is a simplified illustration of one example of venue display screen content including a current guest list that identifies each guest currently logged in by a corresponding nickname and symbol or avatar.

FIGS. 33-37 are simplified illustrations of additional examples of venue display screen content including various aspects and features more fully described below.

FIG. 38 is a simplified diagram that summarizes some examples of venue display screen content.

FIG. 39 is a simplified diagram illustrating one example of connections and dataflow paths involved in implementing dynamic interactive entertainment at a venue.

FIG. 40 is a simplified diagram illustrating one example of software distribution, licensing and revenue sharing business methods in accordance with one embodiment of the present invention.

FIG. 41A is a representation in matrix form of a metric describing the similarity values between collections of media items.

FIG. 41B provides a weighted graph representation for the associations within a collection of media items. Each edge between two media items is annotated with a weight representing the value of the metric for the similarity between the media items.

FIG. 42 is a block diagram of one method for selecting a set of media items corresponding to an initial set of media items in accordance with an embodiment of the invention.

Detailed Description of Preferred Embodiments

[0003] In the following description, certain specific details of programming, software modules, user selections, network transactions, database queries, database structures, etc., are provided for a thorough understanding of the specific preferred embodiments of the invention. However, those skilled in the art will recognize that embodiments can be practiced without one or more of the specific details, or with other methods, components, materials, etc.

[0004] In some cases, well-known structures, materials, or operations are not shown or described in detail in order to avoid obscuring aspects of the preferred embodiments. Furthermore, the described features, structures, or characteristics may be combined in any suitable manner in a variety of alternative embodiments. In some embodiments, the methodologies and systems described herein may be carried out using one or more digital processors, such as the types of

microprocessors that are commonly found in PC's, laptops, PDA's and all manner of other desktop or portable electronic appliances.

[0005] Disclosed are embodiments of systems, methods, and apparatus for generating mediasets comprising a plurality of media data items. As used herein, the term "media data item" is intended to encompass any media item or representation of a media item. A "media item" is intended to encompass any type of media file which can be represented in a digital media format, such as a song, movie, picture, e-book, game, etc. Thus, it is intended that the term "media data item" encompass, for example, playable media item files (e.g., an MP3 file), as well as metadata that identifies a playable media file (e.g., metadata that identifies an MP3 file). It should therefore be apparent that in any embodiment providing a process, step, or system using "media items," that process, step, or system may instead use a representation of a media item (such as metadata), and vice versa.

[0006] In one embodiment, a system to provide recommendations of mediasets for a given group of users is provided. Embodiments of such a system may comprise a mechanism to store playlists and/or playcounts of each member in a community of users. Playlists and playcounts may be used to define the taste of each user and may therefore be used in performing taste analyses for each of the respective users.

[0007] In accordance with the general principles set forth above, embodiments of the invention may be used to address the problem of recommending a mediaset or group playlist to a group of users in a community or network. In some embodiments, a mediaset recommender may be provided where the input is a set of media items, and the output is a mediaset of weighted media items. Two illustrative methods for providing group recommendations of mediasets include: 1) building a common profile that expresses the taste of the group of users as a whole and applying that profile to the recommender; and 2) considering individual recommendations of each member taste, and aggregating the results.

[0008] Media players are typically capable of reproducing all types of media items and collecting playcounts and playlists. Playcounts are the number of times a media item has been played in the media player. Playlists are groupings of media items that users create to organize their libraries of media items. A

system (e.g., a server) may be used to collect playcounts and playlists of media items of a community of users.

[0009] Playcounts and playlists of a user may be used to synthesize her or his taste or perform a taste analysis. In that sense, user's taste may be considered a collection of the most relevant taste data considering that user's playcounts and playlists.

[0010] The task of certain embodiments of this system is to recommend a mediaset for a group of users. A mediaset recommendation for a group of users may be the result of an aggregation process of the different mediasets that are recommended to each user of the group. Thus, some embodiments of this system may include a component that recommends a mediaset from another mediaset (by performing an analysis of the taste of each user, for example). The aggregation process may apply, for example, a voting schema and/or an optimization schema.

[0011] Mediaset recommendations for a group of users may be useful in a variety of scenarios. An example is a party where a group of people want to enjoy music together. Instead of playing the music that may be recommended to a particular individual in the group, it may be desirable to play the music that would be recommended to the group as a whole.

[0012] As such, in certain embodiments, the system's task is to find a mediaset or playlist to be recommended to a group of users. In embodiments wherein the system is collecting playcounts and/or playlists from the users, such a recommender system may be composed of three main steps:

- 1) Synthesizing user tastes;
- 2) Producing recommended mediasets for each user taste; and
- 3) Aggregating the set of recommended mediasets into a single

mediaset to be recommended to the whole group of users.

[0013] It should be understood that numerous variations on the content, scope, and order of these steps are contemplated. For example, the step of producing recommended mediasets for each user taste may be optional.

[0014] FIG. 1 depicts how the taste 108 of a user 100 may be built from the playcounts 104 and playlists 102 of that user 100 through a process 106 called Synthesize Taste. It should be understood that process 106 is one example of a taste analysis. Process 106 produces a taste 108, which may be encoded as a

mediaset. As part of process 106, media items with higher playcounts and/or those that appear more in playlists may be selected. The selected media items may include those that are played more often by the user and/or those that are used in more different contexts and situations (because they appear in many different playlists). In some embodiments, playcounts may also be associated with play dates. In such cases, the process may weigh the playcounts with the play dates and therefore give more relevance to recent playcounts than to older playcounts.

[0015] Process 106 may comprise a ranking process of the media items of a user where items with higher playcounts, more recent plays, and/or more playlist appearances get a higher ranking. The process may select the top m ranked media items as the encoding of the user's taste 108. Note that this process may produce different results over time for the same user. This may be a desirable feature in embodiments in which the goal is to encode the taste of a user as it evolves over time.

[0016] In some embodiments, the system may provide individual mediaset recommendations. For example, with reference to FIG. 2, given a group of n users 200, 208, and 216, the system may compute the individual taste (202, 210, and 218, respectively) for each user with, for example, the process described above. The system may provide a recommended mediaset for each user using a recommender engine (204, 212, and 220, respectively) that processes user tastes (mediasets 202, 210, and 218) to produce recommended mediasets 206, 214, and 222. It should be understood that a separate recommended engine may be used for each user (recommender engines 204, 212, and 220), as shown in FIG. 2 or, alternatively, a single recommender engine may be used for the entire group. Thus, in total the depicted system may compute n mediasets of m media items.

[0017] In some embodiments, the system may also provide for aggregating individual recommended media sets into a group recommended mediaset or group playlist. For example, with reference to FIG. 3, once the system has computed the n mediasets of m media items (one for each user 300, 304, 308 of the group), an aggregation step may be used to produce the final mediaset of p ($p \leq n*m$) media items 314 to be recommended to the group of users. As shown in FIG. 3, the system may provide an aggregator 312, which takes as input a

recommended mediaset (302, 306, and 310, respectively) from each of the users in the group and provides therefrom a group recommended mediaset 314.

[0018] The aggregation step may be performed, for example, by following different approaches that serve different goals. As previously mentioned, in two preferred embodiments, the system may follow a) a voting schema; or b) an optimization schema. A voting schema may serve the goal of finding a mediaset that the majority of users would be happy with, without considering the degree of dislike by the rest of the members of the group. On the other hand, an optimization schema may produce a mediaset that minimizes the dislike (or maximizes the like) of all the members of the group. In order to apply an optimization schema, the media items in the n recommended mediasets 302, 306, 310 may each be linked with an associated weight. The weight of a media item in a mediaset for a user may be used to indicate the relevance of that media item for the user.

[0019] With a voting schema, the aggregation process may take the p media items that appear the most in the n mediasets. If there are items that appear the same number of times in the n mediasets (tie-break), then those items may be picked randomly.

[0020] Some recommender engines may produce mediasets having weighted media items. In such embodiments, when a tie-break situation happens, instead of picking the items randomly, the process may pick the media items with highest weights. For example, considering the following mediasets:

$$ms1 = \{s3, s7, s8, s10\}$$

$$ms2 = \{s2, s3, s4, s10\}$$

$$ms3 = \{s3, s4, s7\}$$

then, following the voting schema described above, the media items (s#) in the mediasets (ms#) would be ranked as follows:

$$s1= 0, s2= 1, s3= 3, s4 = 2, s5= 0, s6= 0, s7= 2, s8 = 1, s9= 0, s10= 2$$

[0021] Thus, the media items for the recommended group mediaset would be selected in the following order: s3, s4, s7, s10, s2, s8. Media items s7 and s10 are in a tie-break situation, so they could be ordered in accordance with their respective weights within their mediasets, if any. If the media items do not have associated weights, then the order of s7 and s10 may be randomized. The same would apply for items s2 and s8.

[0022] With an aggregation schema, the aggregation process may take the p media items that optimize some utility function considering all the users of the group, that is, considering all n mediasets 302, 306, 310. In order to apply the optimization schema, the media items in the n mediasets 302, 306, 310 may have an associated weight, for example, in the range from 0 to 1, where 0 means that the item is not relevant at all and 1 means that the item is the most relevant. For a given mediaset j , a media item i may therefore have a weight $w(j,i)$. If a media item i is not in a mediaset j , then it may be considered to have a weight 0. The following example illustrates the weights associated with media items for a collection of mediasets:

$$ms1 = \{s3, s7, s8, s10\}$$

$$ms2 = \{s2, s3, s4, s7, s10\}$$

$$ms3 = \{s3, s4, s7\}$$

$$w(1) = [0, 0, 0.1, 0, 0, 0, 0.3, 0.2, 0, 0.9]$$

$$w(2) = [0, 0.2, 0.5, 0.4, 0, 0, 0.3, 0, 0, 0.9]$$

$$w(3) = [0, 0, 0.1, 0.8, 0, 0, 0.3, 0, 0, 0.5]$$

[0023] A number of different utility functions may be chosen in order to aggregate the media items. For example, a utility function may be selected to maximize the sum for all p selected items of the highest weight in any of the n mediasets. If it is desired to select $p=3$ items in accordance with this utility function, $s10$, $s4$, and $s3$ would be selected. The sum of the highest weights for these items is $0.9+0.8+0.5 = 2.2$, which is the maximum we can get with the above example.

[0024] Alternatively, a utility function may be selected to maximize the sum for all p selected items of the lowest weight in any of the n mediasets. If it is desired to select $p=3$ items in accordance with this utility function, $s10$, $s7$, and $s3$ would be selected. The sum of the lowest weights for these items is $0.5+0.3+0.1 = 0.9$, which is the maximum we can get with the above example.

[0025] As still another alternative, a utility function may be selected to maximize the sum for all p selected items of the mean weight of all of the n mediasets. If it is desired to select $p=3$ items in accordance with this utility function, $s10$, $s4$, and $s7$ would be selected. The sum of the average of weights

for these items is $2.3/3+1.2/3+0.9/3 = 1.16$, which is the maximum we can get with the above example. Of course, other utility functions may be employed, as will be apparent to one having ordinary skill in the art.

[0026] A recommender engine may be provided in some embodiments. In embodiments that do not include a recommender engine, the mediasets that encode the user tastes may be directly aggregated to form the mediaset that would be recommended to the whole group of users.

[0027] A system using the recommender engine may propose mediasets to discover new music, whereas a system that does not provide a recommender engine may propose mediasets with media items already known by at least one of the users in the group.

[0028] As should be apparent, the aforementioned systems and methods may produce mediaset recommendations for a group of n users so as to enable proposing a mediaset that can be enjoyed simultaneously by a group of users. The system may analyze user tastes from playlists and/or playcounts, so as to allow the user tastes to be represented as mediasets. These n user tastes can be then the input of a recommender engine that may suggest another n mediasets. An aggregation process that takes these mediasets and produces a single group mediaset may also be provided. Such a process can be done using, for example, a voting schema or an optimization schema. Similar systems may operate without a recommender engine. In such embodiments, the aggregation process may operate with the mediasets that represent the tastes of the n users, and the result may comprise a mediaset that can be recommended to the whole group of users.

[0029] Additional embodiments are disclosed and described with reference to FIGS. 4-14. In some of these embodiments, composite playlists may be built of media items in individual libraries on multiple media devices in a common location and/or through a communication network. The media devices in the group may include portable and/or non-portable devices. A means for playing the composite playlist may step through the playlist in sequence and cause the media player with the indicated media item to stream it to one or more of the other players where possible. Such embodiments may operate under, for example, a client-server architecture or a P2P distributed architecture. These embodiments may therefore be used to coordinate a set of media players to

transmit media items on the specified group playlist in streaming fashion to each of the individual media players, or to a specified subset of the media players, in the sequential order they are listed on the group playlist. This may be accomplished without transferring standalone copies of the media items between the various media players.

[0030] Those skilled in the art will recognize that systems incorporating the features of one or more of the above-described embodiments may be realized as a collection of media devices whose design embodies the disclosed functional behavior, as a collection of layered protocols in the 7-level ISO Open Systems Interconnection Reference Model, or as an application task in media devices communicating using standard networking protocols.

[0031] Some preferred implementations may comprise three major components. The first component is a plurality of Session Managers which collectively coordinate the information needed to designate and manage the status of each media player with regard to a plurality of other media player devices engaged in a period of collaborative activity referred to as a "session". In a preferred implementation, one Session Manager may be associated with each media player device, although this need not be the case in all implementations. The Session Managers may include means for verifying the eligibility of each device to join the session. This may be accomplished by virtue of the eligible users being subscribed to a service, which may provide legal access to the media items to be collectively enjoyed.

[0032] The second component of the aforementioned implementation is a Playlist Builder, which may reside, for example, on one of the media players, on a server, on a network access controller included in the system, or on a third-party server accessible to the media player devices through a communication network. The Playlist Builder may use information, such as taste data, available about the users in the group, and the collective set of media items available to the media players, to build a group playlist compatible with the collective tastes of the group.

[0033] The third component of the aforementioned implementation is a plurality of Playlist Managers, one associated with each media player device, which collectively communicate with the Playlist Builder to provide the information needed to build the group playlist and to play the media items on the group playlist. The Playlist Manager associated with each media player device may

include functionality for communicating the availability of media items on the media player device, either in a local library or from a media streaming service accessible by a media player, for example. The Playlist Manager may also include a Playlist Play subcomponent, which may work in coordination with the counterpart subcomponents on the other media players in the session to step through each item on the group playlist, and may be configured to cause the media player associated with a media item to stream it to one or more of the other players as each media item is encountered.

[0034] With reference again to the drawings, further aspects of certain embodiments will now be described in greater detail. Two such alternative embodiments are shown in FIGS. 4 and 5. FIG. 4 is a system diagram of a client-server embodiment 400 and FIG 5 is a system diagram of a peer-to-peer embodiment 500. Both embodiments of the invention may have the same essential components. However, the client-server embodiment 400 of FIG. 4 will be described in greater detail herein than the peer-to-peer embodiment. Nevertheless, It should be understood that the disclosure provided herein may apply equally to the peer-to-peer system with the appropriate system organizational differences, as depicted in the figures.

[0035] The key components of a client-server embodiment 400 are the server functional unit 402 and one or more Media Player Device clients 404. The server 402 may include three basic functional components: 1) A Session Access Controller 408, which may be used to grant permission for an individual media player to join the collaborative interaction between devices; 2) A Playlist Builder 410, may be used to construct the list and sequence of media items to be played in the group playlist during the session; and 3) A Device Manager 406 for each Media Player Device 404 in the session, which may be used to control the session-related functions of the device.

[0036] The peer-to-peer embodiment 500, as shown in FIG. 5, also includes a plurality of media player devices 504. Each of the media player devices 504 may include a device manager 506. A network access controller 502 may also be provided, which includes a session access controller 508 and a playlist builder 510 for receiving media item data from the plurality of media devices and for building a composite playlist of media data items from the media item data.

[0037] The device manager for each media player device may include a session manager and a playlist manager. For example, FIG. 6 depicts a device manager 602 having a session manager 604 and a playlist manager 606. Session manager 602 may be configured to manage the status of the media device with which it is associated.

[0038] FIG. 7 depicts a Member Status screen 702 that is associated with one embodiment of a session manager. As shown in FIG. 7, the session manager may track a series of categories of information for each of the media players in the session, such as ID, proxy, status, and time period information.

[0039] Playlist manager 606 may be configured to communicate with a playlist builder to send the media item data of the media device with which it is associated to the playlist builder and to play the media data items on the composite playlist on the media device with which it is associated.

[0040] As shown in FIG. 4 for the server-client configuration, and FIG. 5 for the peer-to-peer configuration, each media player device 404 may have an associated instance of the device manager 602 of FIG. 6. In addition, the server 402 and network access controller 502 may include a single instance of the session access controller 408, 508 and the playlist builder 410, 510.

[0041] The session access controller 408, 508 may interact, as described in greater detail later, with the session manager 604 in the device manager 602 of each media device 404 in a system to define the set of media devices comprising a session, and to enable communications between them. Similarly, the playlist builder 410, 510 may operate with the playlist manager 606 in the device manager 602 to define and perform a sequence of media items in a session.

[0042] FIG. 8 presents a top-level flow diagram of one embodiment of a session manager 604 in a device manager 602. The session manager of an individual media player device may be used to negotiate the introduction, operation, and departure of the device in a media playing session (henceforth "session"). This may include maintaining knowledge of all other media players devices in the session.

[0043] The session manager 604 may assume the presence of a single session access controller 408 on a server system 402 in the client-server configuration, or on a privileged peer-to-peer network access system 502 in the peer-to-peer configuration. In peer-to-peer embodiments, the network access

system could be implemented on the system found in some peer-to-peer networks, which hosts network-level functions, while in other embodiments it could be implemented on one peer system in the network.

[0044] The session access controller's primary function in some embodiments is to serve a session *sID* code in response to a "*request-to-enter*" session message from devices seeking admission into a session as an indication permission has been granted to the device. In some embodiments, that permission may be granted if the media player *mID* code from the device supplied with the "*request-to-enter*" message is recognized as an *mID* eligible for admission to a session. In other embodiments, the session access controller may instead supply a dynamic *mID* back to the requesting media player device to serve as a unique identifier for the device in the context of the session along with the returned session *sID* code. In some embodiments, the session access controller may largely be functional in the networking protocol for the underlying communications network linking the media player devices, and the *mID* and *sID* may be codes derived from parameters in the network protocol that identify devices and communication sessions or transactions.

[0045] The session manager may include two major subfunctions 802 and 806, as shown in FIG. 8. Subfunction 802 mediates the admission of the media player into a session while subfunction 806 implements the process by which the media player maintains cognizance of the other media players in the session via the communication network linking.

[0046] The starting step for the session manager flow diagram of FIG. 8 assumes the media player is not participating in a session. As shown in the subfunction of 802, the session manager broadcasts a "*request-to-enter*" message with an optional "*mID*" code. If the media device successfully communicates with a session access controller that grants the media player access to a session, the media player will receive an "*enter-session-sID*" message indicating that the media player has been granted permission to enter the session identified by the *sID* code. In embodiments having only one session, the *sID* code would be superfluous and may be omitted or a default value returned.

[0047] If an "*enter-session-sID*" response message is not received, the session manager waits a random amount of time before transmitting another

"*request-to-enter*" message. In some embodiments, the session manager may just wait a random amount of time after transmitting a "*request-to-enter*" message before transmitting another "*request-to-enter*" message if an "*enter-session-sID*" response is not received. In other embodiments, the session manager may wait a fixed amount of time after sending the "*request-to-enter*" message and then, if no "*enter-session-sID*" response is received, wait a random amount of time before transmitting the next "*request-to-enter*" message. In yet other embodiments, the session manager may wait until some external event occurs after sending the "*request-to-enter*" message, rather than receiving an "*enter-session-sID*", and then wait a random amount of time before transmitting the next "*request-to-enter*" message.

[0048] Upon receipt of an "*enter-session-sID*" response from the session access controller, the session manager broadcasts the "*in-session-mID-mID-status-sID*" message, as indicated at 804. By broadcasting this message, the session manager indicates its presence in the session to the session managers in all of the other devices in the session. The parameters of this message (*mID*, *mID*, *status*, *sID*), are the *mID* of this media player device, the *mID* of another media player device in the session that knows about this device by proxy (set here to the *mID* of this media player device because no proxy is involved), the *status* of the playlist manager component 606 of the device manager 602, and the session *sID*.

[0049] After broadcasting the "*in-session-mID-mID-status-sID*" message, the session manager initiates the subfunction 806, which maintains knowledge of the other media players in the session identified by a particular *sID*. Subfunction 506 is a polling loop that maintains the information in a session manager state data structure, such as that shown in FIG. 7. The Device Manager 602 in each media player maintains a copy of the session manager state autonomously for each session, as identified by a unique *sID*, in which the media player is a participant. This state information may include several elements for each of the other media players in the session known to the subject media player: One such element is the *mID* of the media player. Another is the *mID* (referred to as the *pmID*) of the proxy media player in the session known to the subject media player, which knows the media player identified by *mID* is in the session, even if that media player is not known directly to the subject media player. A third example is the

status of the media player. Still another example is the amount of *time* since this media player was last affirmatively known to be in the current session by the subject media player.

[0050] The polling loop in Subfunction 806 may be executed a number of times "n" as determined by the implementation. This number is relatively arbitrary, and typically is selected to achieve a desired "liveliness" criteria for the session manager and the session maintenance protocol. "Liveliness" here refers to how often the subject media player indicates its presence in the session to the other media players in the session. As the flow diagram indicates at 804, the session manager may broadcast an "*in-session-mid-mid-status-sid*" message to the other media players to indicate the subject media player is still active in the session and make its status known to those media players if the session manager state data table 702 includes entries for other media players. If the session manager state data table 702 is empty, the session manager may instead revert to searching for a session to join by initiating the session join subfunction 802.

[0051] Subfunction 806 may maintain knowledge of the other media players in the session by repetitively executing the three processes 810, 812, and 814 shown in FIG. 8, each of which are shown in greater detail in FIGS. 9-11, respectively. The "Prune Session Members" process 810 scans the session manager state data structure 702 and removes entries corresponding to media players from which the subject media player has not received an affirmative indication that they are still in the session, and therefore are inferred to have left the session. The "Update Session Members" process 812 processes messages from other media players that are therefore inferred to have joined or remained in the session. Finally, the "Service Proxy Requests" process 814 responds to requests from other media players if the session manager state data structure 702 includes an entry for a media player known directly to the subject media player that has been only known to another media player by proxy via the subject media player.

[0052] Describing each of the processes 810, 812, and 814 in turn, the "Prune Session Members" process 810 (FIG. 9) may perform one of the following three operations on the entry for each media player *i* in the session manager data structure 702.

[0053] 1) The entry is removed if the *time[i]* that the media player was last affirmatively known to the subject media player exceeds an implementation specified *timeout* value. A "*proxy-session-mID-sID*" query is then broadcast to request if the media player corresponding to the removed entry is still known to another media player in the session.

[0054] 2) A "*proxy-session-mID-sID*" query is broadcast to request if the media player is still known to another media player in the session, and the media player entry status has been marked as "cued" (meaning that the media player was scheduled to perform a track by the Playlist Manager process described below), and if one of the two following conditions exist: a) If the *time[i]* that the media player was last affirmatively known to the subject media player exceeds an implementation specified *cued timeout* value, meaning that the subject media player did not receive an indication that the media player corresponding to the entry transmitted a state change out of the cued state; b) If the *time[i]* that the media player was last affirmatively known to the subject media player does not exceed an implementation specified *cued timeout* value, but the media player corresponding to the entry is only known to the subject media by proxy via another media player in the session. In this case, the subject media player would only learn of a state change by the media player corresponding to the entry if that state change is broadcast by the proxy media player in response to the "*proxy-session-mID-sID*" query.

[0055] 3) Nothing is done to the entry for the media player in the event neither of the above conditions apply.

[0056] The "Update Session Members" process 812 (FIG. 10) receives the "*in-session-mID-pmID-status-sID*" messages broadcast by the session manager and the "Service Proxy Requests" process 814 and updates the session manager state data structure 702. The process assumes that the protocol of the network over which the media players communicate buffers all "*in-session-mID-pmID-status-sID*" messages until they can be processed. Each received message in which the message *sID* matches the ID of the session is processed in one of the five following ways:

[0057] 1) An entry corresponding to the media player with the *mID* of the message is added to the Session Manager state data structure if no entry with *mID* as the ID already exists.

[0058] 2) If the *mID* and the *pmID* of the message match, implying that this message was transmitted by a media player that is now directly known to the subject media player, the entry in the data table for the media player with *mID* is updated with the *status* from the message. The time since the media player with *mID* was last affirmatively known to the subject media player is reset to 0 seconds.

[0059] 3) The entry in the data table for the media player with *mID* is updated with the proxy media player ID *pmID* and the *status* from the received "*in-session-mID-pmID-status-sID*" message, if the received message is a proxy message (*mID*, *pmID* differ), the media player referenced by the message is currently known to the subject media by proxy (*mID*, *pmID* for the entry in the Session Manager state data structure differ) or the time since the media player with *mID* was last affirmatively known to the subject media player has exceeded an implementation *timeout* value, and either the *status* of the referenced media player is unknown or the *status* in the message is idle. The time since the media player with *mID* was last affirmatively known to the subject media player is reset to 0 secs.

[0060] 4) The entry in the data table for the media player with *mID* is updated with just the proxy media player ID *pmID* from the received "*in-session-mID-pmID-status-sID*" message, if the received message is a proxy message (*mID*, *pmID* differ), the media player referenced by the message is currently only known to the subject media by proxy (*mID*, *pmID* for the entry in the Session Manager state data structure differ) or the time since the media player with *mID* was last affirmatively known to the subject media player has exceeded an implementation *timeout* value, and the *status* of the referenced media player known and the *status* in the message is idle. The time since the media player with *mID* was last affirmatively known to the subject media player is reset to 0 secs.

[0061] 5) Nothing is done if the received "*in-session-mID-pmID-status-sID*" is a proxy message (*mID*, *pmID* differ), the media player referenced by the message is currently known to the subject media player (*mID*, *pmID* for the entry in the session manager state data structure match), and the *time* since the referenced media play was last affirmatively known to the subject media player does not exceed an implementation-defined *timeout* value.

[0062] The "Service Proxy Request" process 814 (FIG. 11) provides information about any media players the subject media player affirmatively knows to the other media players in the session in response to the "*proxy-session-mID-sID*" request messages, which may have been broadcast by the "Prune Session Members" process 810 of any media player in the session identified by *sID*. As with the "Update Session Members" process, this process assumes that the protocol of the network over which the media players communicate buffers all "*proxy-session-mID-sID*" messages until they can be processed. Each received "*proxy-session-mID-sID*" message in which the message *sID* matches the ID of the session may be processed in one of the four following ways:

[0063] 1) An "*in session-mID-pmID-status-sID*" message is broadcast, where *pmID=mID* and the *status* is that of the subject media player, if the subject media player corresponds to the *mID* of the request.

[0064] 2) An "*in session-mID-pmID-status-sID*" message is broadcast, where *pmID=mID*, and the *mID* and *status* parameters are those in the session manager state data structure 702 for the media player referenced by the query, if the media player *mID* of the query is known affirmatively to the subject media player (*mID* of the query and *pmID* in the state manager state data structure match).

[0065] 3) An "*in session-mID-pmID-status-sID*" message is broadcast, where the *pmID* and *status* parameters are those in the session manager state data structure 702 for the media player referenced by the query, if the media player *mID* of the query is only known to the subject media player by proxy (*mID* of the query and *pmID* in the State Manager state data structure differ).

[0066] 4) Nothing is broadcast if the "*proxy-session-mID-sID*" message does not reference the subject media player or a media player in the Session Manager state data structure 402 known to the subject media player.

[0067] The Playlist Manager of an individual media player device may consist of two major subfunctions: First, the "Playlist Queue Updater" process 1300, as depicted in FIG. 13, which negotiates the sharing of playlist information between the media playing devices in the session. And, second, the "Playlist Play Sequencer" process 1400, as depicted in FIG. 14, which controls performance of the media items by the subject media player.

[0068] The Playlist Manager 1200 (FIG. 12) assumes the presence of a single Playlist Builder on a server system in the client-server configuration, or on a

privileged peer-to-peer network access system in the peer-to-peer configuration. In peer-to-peer embodiments, the playlist builder could be implemented on the system found in some peer-to-peer networks, which hosts network-level functions, while in other embodiments it could be implemented on one peer system in the network. The "Playlist Queue Updater" may provide the data from the media player to the Playlist Builder that is used in the playlist building process. The Playlist Builder will be described in detail subsequently.

[0069] The Playlist Manager 1200 also assumes that the media player makes available several data items relevant to the playlist building process depicted conceptually in Fig. 15. In some embodiments, the media player device 1502 may store media items in a local library 1506, and provide a mechanism for querying the local directory 1508 to determine whether a specific media item is available for performance by the media player. In other embodiments, the media player 1502 may download or stream media items from a remote service over a communication network on demand, and provide a mechanism for querying a directory 1504 of media items available from the remote service for performance by the media player. FIG. 15 also depicts several other data items maintained by the media player relevant to the playlist building process, including indication 1510 of a media item that should be preemptively added to the playlist, and lists of items 1512, 1514, and 1516 of items that user does not want added, added preferentially, and added if at all possible, respectively, to the playlist.

[0070] FIG. 13 presents a top-level flow diagram of the "Playlist Queue Updater" process 1300. As the figure illustrates, this may be a repetitive process performed ad infinitum. The "Playlist Queue Updater" first determines if the user of the subject media player has requested that a particular media item be added to the playlist being built, and sends a "*force-iid-pid-length-sid*" message to the Playlist Builder if so. This message specifies the media item by *iID*, the *pID* of the media player (this is the same value as *mID* used by the Session Queue Updater described previously), the *length* in time units of the media item, and the *sID* for the session.

[0071] After any user request that a specific media item be added to the playlist has been processed, the "Playlist Queue Updater" then checks if a "*have-iid-sid*" request message has been received from the Playlist Builder inquiring whether the subject media player can provide a specific media item for

the playlist. The process assumes that the protocol of the network over which the media players communicate buffers all *"have-iiD-sID"* request messages until they can be processed. Each received message in which the message *sID* matches the ID of the session may be processed in one of three following ways:

[0072] 1) A *"have-iiD-pID-length-sid"* message is broadcast if the subject media player *pID* has access to the requested media item *iiD* with length *"length"* from a local library of media items.

[0073] 2) A *"have-iiD-pID-length-sid"* message is broadcast if the subject media player *pID* does not have access to the requested media item *iiD* with length *"length"* from a local library of media items, but does have access to the requested media item from a remote service.

[0074] 3) No response is broadcast if the media player does not have access to the requested media item.

[0075] In some embodiments, the "Playlist Queue Updater" may take into account the user's preferences with regard to media items, as indicated by the lists 1512, 1514, and 1516 in determining whether to supply a *"have-iiD-pID-length-sid"* message in response to a *"have-iiD-sID"* request message. For example, the "Playlist Queue Updater" may not respond to the *"have-iiD-sID"* message even if the requested item is in the catalog 1504 or 1506 of the device if it is also on the "no play" list 1512. Similarly, the "Playlist Queue Updater" may respond optionally according to some statistical or other criteria if the item is on the "preferred" list 1514. And the "Playlist Queue Updater" may always respond if the item is on the "must play" list 1514.

[0076] The last step in an iteration of the "Playlist Queue Updater" flow diagram processes at least one *"queue-iiD-pID-length-sid"* message from the Playlist Builder, if any have been received. The process assumes that the protocol of the network over which the media players communicate buffers all *"queue-iiD-pID-length-sid"* messages until they can be processed. In some embodiments, the "Playlist Queue Updater" may process only a single *"queue-iiD-pID-length-sid"* message per iteration by adding an entry to the Playlist Queue data structure in the Playlist Manager, consisting of the *iiD*, *length*, and *pID* items from the message. In other embodiments, it may process multiple or all pending *"queue-iiD-pID-length-sid"* messages.

[0077] . In one embodiment, performance of the playlist is, in effect, directed by the Playlist Builder. As described later, the Playlist Builder may broadcast a "*queue-ild-pID-length-sid*" message to all the media players in the session requesting that media item be added to the Playlist Queue data structure 1202 in the Playlist Manager 1200. The Playlist Builder sends this message at the actual time the media item should be performed and the media player accepts that message as a command to perform the specified media item. In a variant of this embodiment, the Playlist Builder may send this message just sufficiently before the time the media item should be performed to allow the media item to perform any processing required to initiate the performance by the time the performance is to start.

[0078] In another embodiment, the "Playlist Play Sequencer" process 1400 of the Playlist Manager 1200 shown in FIG. 14 locally coordinates the performance of the items on the playlist that are actually performed by the subject media player with the performance of the other media items on the playlist by the appropriate media players. All media players in the session may have a copy of the Playlist Queue 1202. This copy may be imperfect: It could be missing items at the head of the queue that were put on the playlist before the subject media player joined the session, or in the middle of the queue due to communication failures. The Playlist Play Sequencer may cause the subject media player to synchronize its performances of individual media items with the performance of media items by other media players in the session in the presence of these possible gaps in the playlist.

[0079] The "Playlist Play Sequencer" 1400 may be an iterative process which achieves the synchronized performance in the presence of gaps by processing the item at the head of the Playlist Queue 1202 in, for example, one of the three following ways:

[0080] 1) If the *pID* of the media item at the head of the queue is not the *pID* of the subject media player, corresponding to the left branch of the flow diagram, the "Playlist Play Sequencer" essentially just idles, monitoring the *status* of the media player with *mID=pID* in the Session Members state data structure 702 until it is inferred that the media item has been performed. The subject media player infers the media item has been performed when either a transition from *played* to

idle is observed, or the value of the local *playtime* timer exceeds the performance *length* of the media item.

[0081] 2) If the *pID* of the media item at the head of the queue is the *pID* of the subject media player, and the Session Members state data structure 702 does not include an entry for another media player with the *status* value *playing*, corresponding to the middle branch of the flow diagram, the "Playlist Play Sequencer" plays the media item. The *status* of the subject media player is set to *playing* while the item is being performed, and then set back to *idle* after the performance is finished to signal the performance to the other media players in the session.

[0082] 3) If the *pID* of the media item at the head of the queue is the *pID* of the subject media player, but the Session Members state data structure 702 includes an entry for another media player with the *status* value *playing*, corresponding to the right branch of the flow diagram, the performance of the media item is postponed. The "Playlist Play Sequencer" repeatedly traverses this branch of the flow diagram until no other media player has the *status* value *cued*, and then sets the *status* value for the subject media player to *cued*. On the next iteration the "Playlist Play Sequencer" takes the middle branch of the flow diagram and performs the media item as described above.

[0083] As previously described, the Playlist Manager of the Device Manager in each media player device may assume the existence of an autonomous Playlist Builder on the server system in the client-server configuration, or on a privileged peer-to-peer network access system in the peer-to-peer configuration. In peer-to-peer embodiments, the playlist builder could be implemented on the system found in some peer-to-peer networks which hosts network-level functions, while in other embodiments it could be implemented on one peer system in the network.

[0084] FIG. 16 depicts how the Playlist Builder 1606 draws on a knowledge base 1604 of information about media items to compose a playlist of media items in the playlist queue 1608 from the pool of media items 1602 available to the media playing devices in the session.

[0085] FIG. 17 is a flow diagram of the playlist builder 1606 process. The process assumes the availability of a method for generating the IDs of media items that are candidates for inclusion in the playlist. Candidates may be proposed based on many criteria, including information in the knowledge base

1604, and knowledge of the preferences of the users whose media players are participating in the session.

[0086] . The "Playlist Builder" may be an iterative process that adds a single media item to the playlist per iteration. Each iteration may include three steps:

[0087] 1) A candidate media item with *ciID* is generated based on information in the knowledge base 1604 and/or other criteria. A "*who-has-ciID-sID*" query is broadcast to all of the media players in the session to determine if any of them has access to the proposed media item.

[0088] 2) If the Playlist Queue Manager of any of the media players in the session has previously broadcast an unprocessed "*force-iID-pID-length-sID*" message, the requested media item *iID* is added to the playlist 2108. The Playlist Builder may broadcast a "*queue-iID-pID-length-sID*" message to all of the media players in the session, directing that they each add the requested media item to their local playlist that their Playlist Queue Manager is maintaining.

[0089] 3) After an implementation-determined delay, a determination is made if the Playlist Queue Manager of any of the media players in the session has broadcast an unprocessed "*have-iID-pID-length-sID*" response message, indicating that a media player in the session has access to the requested media item *iID*. As one or more of the media players may have access to the requested item, one of those media players is selected either at random, or according to some other criteria, as the media player that will perform that media item and the item is added to the playlist. The Playlist Builder may then broadcast a "*queue-iID-pID-length-sID*" message to all of the media players in the session directing that they each add the requested media item to their local playlist maintained by their Playlist Queue Manager 1100.

[0090] The process assumes that the protocol of the network over which the media players communicate buffers all "*force-iID-pID-length-sID*" request messages and "*have-iID-pID-length-sID*" until they can be processed. In any particular embodiment, one or more of each type of message may be processed per iteration. In addition, as the flow diagram indicates, steps 2) and 3) may be repeated a number of times limited by a *timeout* value to increase the responsiveness of the communications between the Playlist Builder and the media player devices.

[0091] The Playlist Builder iterations may be repeated ad infinitum. The playlist is a non-terminating sequence of media items to be performed so long as there is at least one media player in the session. Furthermore, some embodiments may support building playlists consisting solely of media items suggested by users of the media devices, and communicated to the Playlist Builder with the "*force-iID-pID-length-sID*" request message, by providing an option for setting a option flag to "false" so that the "auto build?" tests in the flow diagram fail.

[0092] Still other implementations are disclosed and described with reference to FIGS. 18-26. In these embodiments, playlists of selected media items may be dynamically built from a collection of media items based on the preferences of a temporally-defined group of users with media player devices. Compositional goals and the collection of media items available to meet those goals may be dynamically computed using, for example, the composite taste data for the group as a whole as individual users enter and leave the group.

[0093] Some embodiments therefore relate to methods for dynamically creating a playlist of media items responsive to the collective tastes of a temporally-defined group of individuals. Some embodiments also provide for dynamically diversifying the group playlist so that it does not in whole, or in part, unduly reflect the taste of a single member of the group, or a particular subgroup of users within the whole group.

[0094] Additional embodiments may provide for a system and method for dynamically building a playlist of media items by using the collective taste preferences of the members of a group to determine compositional goals of the playlist, and then building a group playlist that achieves those compositional goals. The system may derive the compositional goals by analyzing the taste preferences of the current members of the group. Media items available to achieve those goals are typically a subset of the media items that are identified in response to analyzing taste data and may be selected from a collection of media items available to the system. The collection of available media items may be the aggregate of the sub-collections of media items provided by the users or, alternatively, may be a pre-existing set of media items stored, for example, in a central database or jukebox.

[0095] In a preferred implementation of the system, three primary processes are provided. The first process keeps track of users as they enter and/or leave the group by starting or ending communications with the system using, for example, individual networked communication devices. Example embodiments include Bluetooth® devices and other devices communicating in an ad-hoc network of Internet or other network-connected devices using, for example, the Apple Bonjour protocol. Users with individual communication devices may be added to and/or removed from the group by the system as they implicitly or explicitly connect and disconnect from the communication network, which links the individual devices to the computational means for building the group playlist.

[0096] As a user enters the group, the system may retrieve the user's taste data. Taste data may be retrieved by, for example, accessing a database of taste data from users known to the system or by requesting taste data directly from the users' communication devices and adding it to the pool of taste data for the group. As a user leaves the group, the system may also be configured to remove that user's taste data from the pool of taste data for the group. In some embodiments in which the media items available for inclusion in a playlist are provided by users and not centrally maintained by the system, the system may maintain a pool of media items available for the current group.

[0097] Using the pool of taste data for the group, and the pool of media items available for inclusion on a playlist, the first process may derive compositional goals for the playlist, such as requiring that the values for the selected media items of a specific attribute have a specified distribution. In some embodiments, the first process may also involve selecting a subset of media items from the total pool of media items to be used to build the playlist. This may be accomplished using a media item "recommender," as further described below with reference to Figs. 41-42.

[0098] The second process may build the composite playlist by selecting media items from the total pool of media items in a manner which causes the evolving playlist to more closely approximate the specified compositional goal as the selected items are added to the playlist. Some implementations of the system may therefore be responsive to the constantly changing group membership. In particular, as users in the group continuously enter and leave the group, the compositional goals and/or the pool of media items available to

achieve those goals may continually change. Some embodiments may remove media items from the dynamic playlist as users depart from the group, particularly in those situations where the group members actually contribute the media items to the pool. In such situations, the media items in the pool could be removed from the pool as users leave, such as by physically leaving a proximity or by logging out of a system. It should be understood, however, that such a feature is not necessary in all implementations, since a media item can be skipped if it is no longer available when it is to be performed.

[0099] The third process may involve diversifying the group playlist. The diversification process may involve shuffling media items on the list as necessary to ensure that no segment of the playlist is dominated by media items representative of the taste of one or more group members. In some embodiments, additional information about aesthetic properties of the media items might be used to rearrange the order of the media items in the group playlist to achieve specific aesthetic goals. Finally, in cases where there are few users, and therefore for each user the playlist includes a large number of items responsive to the taste of just that user, some embodiments may replace some media items with additional media items. These additional or supplemental media items may not be provided by any of the members in the group, and may be selected according to a diversifying criteria to bring more variety to the group playlist.

[00100] One embodiment of the system may implement the processes detailed in FIGS. 18-20. This embodiment may be analogized to the automated evolution of a jukebox. Whereas a jukebox supplies local music and requires explicit user interaction from the user to pick desired music, these embodiments automatically build playlists based on previously collected data about users that are proximal (either physically or virtually) to the system.

[00101] In describing the embodiment of FIGS. 18-20 below, a process of adding users to the system will be described first, then a process for removing users, and finally the overall playlist construction process is described, including how the process uses information to build a targeted playlist that leverages knowledge of proximal users for building a playlist.

[00102] In order for the system to “narrowcast” (*i.e.*, to target content to a specific set of users) a playlist for proximal users, a mechanism may be provided

to allows for discovery/detection of proximal users. In one embodiment, a server process 2400 (as shown in FIG. 24) may be provided using, for example, Bluetooth® technology to allow users to “log in” to the system. Users who are “logged in” to the system are considered proximal users, regardless of their physical proximity to the other users or server. In some embodiments, all proximal users immediately begin to affect the playlist being constructed, which will be discussed in greater detail later. It will be obvious to one of ordinary skill in the art that Bluetooth® is just one of many data discovery/entry mechanisms that could be used to add users to the system. Other embodiments may include explicit text entry from users, or even be a zero-configuration protocol such as Apple's Bonjour protocol.

[00103] In step 1 of one implementation of a Bluetooth® user addition process, a Bluetooth® server with a predetermined service UUID (Universally Unique Identifier) is provided. The Bluetooth® specification uses UUIDs to identify services uniquely across many devices. By using a UUID, a Bluetooth® client is able to detect a specific service on a remote server.

[00104] In step 2, once a client Bluetooth® process has connected with the system's Bluetooth® server 2400, the client transmits user information to the server process 2400, as shown in FIG. 25 at 2500. The server process 2400 will then attempt to validate the user information, as indicated at 2501. If the user validation process 2501 fails, then the user is rejected and is not added to the proximal user list 2503 or the proximal device list 2504 by the user list modification process 2502. If the user validation process 2501 succeeds, then the user is added to the proximal user list 2503 and the user device is added to the proximal device list 2504. The means for entry of the user and the user device to the appropriate lists may vary, as would be understood by one of ordinary skill in the art.

[00105] In order to provide real-time narrowcasted playlist content, the system preferably updates and maintains the contents of the proximal user list 2503 and proximal device list 2504 on a regular basis. At the same time, the system may be configured to reduce the chance that users are erroneously moved from either list. These concerns may be addressed by using a User/Device Proximity Detection Process 2600, as shown in FIG. 26, to identify which users are proximal to the system. Process 2600, like various other processes, may be implemented

using Bluetooth® technology. The system may be able to determine when users are no longer proximal by, for example, detecting proximal devices and mapping the proximal device list 2504 to the proximal user list 2503.

[00106] After a user has been discovered to no longer be proximal, a User/Device Inactivity Detection Process 2601 may be used to begin considering whether a user should be removed from the proximal user list 2503 and the user's device removed from the proximal device list 2504. The User Device User/Device Inactivity Detection Process 2601 may be implemented as a time-based process. The system may be configured to remember the time when a user first became classified as non-proximal. Then, for example, if a specific time limit has been reached and the non-proximal user is identified as still being non-proximal, the user may be removed from the proximal user list 2503, along with the user's device from the proximal device list 2504, by the User/Device Removal Process 2602. If a user is found to be proximal by the User/Device Proximity Detection Process 2600 before the time limit of the User/Device Inactivity Detection Process 2601 is reached, then the user and their device will be left on the appropriate lists 2503/2504. It will be apparent to one skilled in the art that specific implementations of the aforementioned system may, but need not, rely on the use of Bluetooth® or a time-based user removal strategy.

[00107] The Composite Playlist Builder process may rely on the User Addition/Removal processes described above in order to determine which users to which a playlist is to be narrowcasted. In one implementation, the Composite Playlist Builder process may first generate a list of media items that define a user taste, and then repeat this step for all proximal users in the system.

[00108] As shown in FIG. 18, the process of generating a list of media items that define user taste may be accomplished by examining the users 1800 logged into the system and using information—*i.e.*, media item taste data—to synthesize user taste, as shown at 1802. The media item taste data may be gathered by the system contemporaneously or, alternatively, it may be compiled into a list or database from previously gathered media item taste data. Either of the foregoing may be represented by step 1801 in FIG. 18. A list of media items and/or media item metadata (collectively "media data items") that contains information about the music that should be narrowcasted to user 1800 may then be generated, as indicated at 1803. It should be noted that it is not necessary to have any users

logged into the system in order for a group playlist to be generated. Preconfigured goals can be used when no/few users or otherwise insufficient data is available to the system. This process may then be repeated for each user 1800 who is logged into the system.

[00109] The data in one embodiment may be cached to enable quick lookup. As will be demonstrated in other steps, the data which represents user taste may also provide the foundation from which other steps derive information. In some embodiments, user taste may be synthesized by first obtaining a set of the tracks that a user has listened to recently and/or those that a user has ranked highest. Of course, it will be obvious to those skilled in the art that there are many different ways to synthesize user taste based on collected user taste data.

[00110] As a second process of one implementation, aggregate playlist goals may be computed based on results of the first process and/or on a list of desired categories. Note that there will be typically be one set of input media data items for each user.

[00111] In order to compute aggregate playlist goals, each set of media items and/or metadata that indicates user taste 1900, and/or a predetermined or computed set of categories of interest 1901, may be used as input to a User Taste Aggregation Process 1902, as shown in FIG. 19. User Taste Aggregation Process 1902 may be used to create a set of optimal playlist characteristics 1903 for a group playlist that is intended to be narrowcasted to the proximal users. In certain embodiments, media items and/or metadata that are representative of more than one user's taste may not be viewed as duplicative. This feature may allow media items and/or metadata that are common between multiple users to be given increased levels of importance by the User Taste Aggregation Process 1902. The User Taste Aggregation Process 1902 may accept both user taste 1900 and desired categories of interest 1901 as input. Categories of interest 1901 are often (but not required to be) metadata indicators that indicate the set of metadata to be used in computing the optimal playlist characteristics 1903 for a narrowcast playlist targeted at a particular group of users.

[00112] One embodiment uses a category set 1901 that comprises a genre. Such a system may compute the frequency distribution of genres in the play histories retrieved for each connected user. The percentage of each genre may then be used as the optimal genre distribution for the generated group playlist.

Of course, a genre is not the only category which could be used for generating a playlist. It should also be understood that a raw frequency distribution is not the only method for computing statistics about any given category. Other similar implementations are not limited to, but could employ, a weighting strategy or voting strategy to determine desired levels of each feature.

[00113] After the aggregate playlist goals have been computed, relevant media items for a user may be selected from the pool of available media items. This process may then be repeated for each proximal user in the system.

[00114] For example, in FIG. 20, the set of relevant media items 2002 that are associated with a given user may be computed/selected. To do this, a process 2001 for each user may be executed. Process 2001 may accept as input the set of media items and/or metadata that indicates a user's taste 1803 and/or the set of all available media items 2000. Process 2001 may also produce a subset of available media items that are relevant media items for a single user.

[00115] In one embodiment, process 2001 may be implemented by using a media item recommender, such as those described in U.S. Patent Application Serial No. 11/346,818 titled "Recommender System for Identifying a New Set of Media Items Responsive to an Input Set of Media Items and Knowledge Base Metrics," which was previously incorporated by reference. Process 2001 may also be augmented by providing the media item recommender with the complete scope of recommended media items from which to recommend a subset of media items. In practice, this may be used to make sure that the recommended media items are available for use by the composite playlist builder application. For example, if there are one-thousand available media items 2000 to choose from, process 2001 may ensure that the relevant media items 2002 are within the one-thousand available media items 2000. This is analogous to a jukebox that has a limited set of media from which to produce a playlist. It should be apparent that the media items available to process 2001 do not need to be resident on the same machine that is executing process 2001. Any media item which is programmatically obtainable via any protocol may be considered an available media item.

[00116] After the relevant media items for each user have been selected from the pool of available media items, the media items may be categorized according to a set of desired attributes. For example, as shown in FIG. 21, the relevant

media items 2100 may be categorized via a categorization process 2101 according to the categories of interest 1901 used previously in the user taste aggregation process 1902. Categorization process 2101 may be used to produce a categorized list of relevant media items 2102. This list may enable the overall system to pick one or more media items based on one or more categories of interest 1901.

[00117] After the media items have been categorized according to a set of desired attributes, the status of the current group playlist's achievement of goals may be assessed/computed. For example, with reference to FIG. 22, a goal assessment process 2201 may be used, along with categorization process 2101, to produce a prioritized list of goals needing to be fulfilled 2202. The categorization process 2101 may accept the current playlist 2200 and categories of interest 1901 as input. The categorization process 2101 may use the categories of interest 1901 to categorize the media items on the current playlist 2200. The output of the categorization process 2101 may be sent to the goal assessment process 2201 that may be used to determine how close the current playlist 2200 characteristics are to the optimal playlist characteristics 1903. The output of the goal assessment process 2201 may be a prioritized list of goals needing to be fulfilled 2202.

[00118] In one embodiment, a frequency distribution of genres may be computed for the categorization process. The goal assessment process may comprise subtracting the achieved genre percentages for the current playlist from the target percentages for the optimal playlist. It can then take the largest value difference as the needed upcoming genre for the playlist. Of course, a variety of other approaches can be used for computing playlist needs.

[00119] After the status of the current group playlist's achievement of goals has been assessed/computed, a set of tracks may be selected to add to the group playlist based on assessing current needs for the playlist according to the goal achievement assessment process previously performed. For example, with reference to FIG. 23, a set of one or more media items 2301 are selected for addition to the current playlist 2200 by a playlist addition process 2302. The media item selection process 2300 may accept as input the prioritized list of goals needing fulfilled 2202, the categorized list of relevant items 2102, and the current playlist 2200. The media item selection process 2300 may attempt to

fulfill the goals in a prioritized manner in some embodiments, such that the most important goal is fulfilled by picking a media item from the list of relevant items 2102 that best fulfills the goal. If multiple media items could help fulfill a goal, then a media item may be selected at random from the list or in another fashion, such as by some form of tiebreaker process.

[00120] The selected media item may also be subjected to a diversification step, which may be used to ensure that the media items being added to the playlist are not too similar to the current playlist contents. In some embodiments, the diversification step may comprise shuffling media items in the group playlist to diversify at least one segment of the group playlist that includes media data items that are overly representative of the tastes of one or more users. The system may be configured such that, in response to determining that the group playlist is dominated by media items representative of the taste of one or more users, removing at least one media data item from the group playlist that corresponds with the taste of the one or more users.

[00121] In some embodiments, in order for a media item to be selected and added to the current playlist 2200, it must pass through the diversification process (unless the diversification process eliminates all media items in the pool). Once a sufficient set of media items have been selected, the set of media items 2301 may be sent to a playlist addition process 2302 for addition to the active playlist. The playlist addition process 2302 may have the capability of creating/updating the current playlist 2200 and, in some implementations, initiating playing the media items in the current playlist 2200.

[00122] In one embodiment, a media item may be selected from the most needed genre for the current playlist by choosing at random from relevant tracks categorized in the appropriate genre. The diversification step may be used to ensure that no media item is repeated in the group playlist for a predetermined period of time. In other embodiments, the diversification step may also, or alternatively, be used to ensure that no subset of media items—such as artists, albums, genres, etc.—are repeated within a given subset of the group playlist. For example, the diversification step may be used to prevent media items from a particular artist from being repeated within a seven track window within the group playlist. If it is impossible to fulfill the requirements of the diversification process, then a media item may be chosen at random from all media items within the

appropriate genre. If multiple genres are identified with equal priority, then the selected media item may be selected from any one of the identified genres, or by some other tie-breaking selection procedure.

Venue Recommendations

[00123] Turning now to FIG. 27, another application of the technologies described earlier is to enable a venue to tailor its offerings to the tastes of a user or customer. In this context, a “venue” can be an actual physical space such as a hotel, restaurant, bar, etc. or it can exist in virtual space, such as a venue on the Internet. Moreover, in the present context, a “user” can be an individual user, a user group, or a group of users.

[00124] A hotel is used as an example to illustrate this aspect of the invention. Referring now to FIG. 27, a customer would sign up as a hotel user. This could be, for example, an Internet-based service. It is known that many hotels currently have on-line registration systems, as well as “loyalty” or preferred customer arrangements whereby they store customers’ preferences such as nonsmoking room and queen size bed. What they have not done in the past is to collect user preferences such as lists of one or more favorite movies, favorite foods and beverages, music (songs, albums or artists, etc.). Thus the present process begins with a customer signup step 2702 in the figure, followed by a collection of user preferences 2704. The hotel can then employ recommender software of the type described earlier, to identify recommendations that would be appropriate to this particular customer based upon the individual preferences collected, step 2706. Finally, these recommendations can be provided to the customer through various channels. For example, these recommendations might simply be delivered to the user via recorded telephone message, through a hotel room television system, email or wireless telecom messaging. Alternatively, the user, at his or her convenience, might use a device such as a mobile phone to access the hotel’s recommendations as desired. For example, when the user is ready to go to lunch, he could simply access the hotel’s recommendations for a restaurant for lunch, and in all likelihood the recommendations will be consonant with his personal preferences.

[00125] In another aspect of the invention, recommender technology can be employed to recommend one or more venues to an individual user based on that

user's profile. A method for recommending a venue to a user is summarized in the simplified flow diagram of FIG. 28. FIG. 28 begins with the user login step 2802. This step may comprise logging into an existing system where the user is already a "member" such that a preexisting profile can be accessed.

Alternatively, a new user can provide profile information at the time of login. In the case of music or other media items, user preference can be expressed by identifying one or more favorite songs or artists, as explained previously. Step 2804 in FIG. 28 illustrates the case of accessing an existing user profile.

[00126] Next, the method calls for accessing venue profile information, step 2808. Again, in the case of media items, a venue profile may consist of the current media items offered along with the current users logged into the system. This refers to users or "guests" who are currently in attendance at the venue, as further explained below. The media items offered by a venue may be ordered in time resulting in a sequence of media items. For example, in the case of music, the current media items offered by a venue would be a playlist. Additionally, a venue profile may contain general attributes like location, style, price range, opening hours, and other properties that describe the static profile aspects.

[00127] Finally, the system recommends a venue, step 2810, based upon the user profile and the venue profile. The recommendations could be done in real time as the venue profile changes in response to guests entering and leaving the venue, and the media items offered, as further explained below. The system may weight the importance of media items offered with respect to their associated time. In this way, older items would be less weighted than recent items.

[00128] For example, if a group of users wanted to go dancing to disco music, they could use an appropriate device to access the nearest club that was currently playing the kind of music that they liked. In real time, the users could check the current play list. After they arrive, the same users can begin to influence the music played at the venue, as explained below.

[00129] FIG. 29 is a simplified diagram illustrating software programs at three different conceptual levels. At the bottom, a recommender core engine 2900 may comprise a recommender system of the type described above, i.e. one for identifying a new set of media items in response to an input set (or "query set") of media items. Such a system in the preferred embodiment uses a knowledge base comprising a collection of mediasets.

[00130] At the intermediate conceptual level, FIG. 29 illustrates several examples of algorithms that may employ the recommender core 2900. For example, a user to user affinity algorithm 2902, a group recommender 2904, a composite play list generator 2906, and a dynamic group management algorithm 2908 are shown by way of examples. These algorithms were described above in detail. Finally, along the top level in FIG. 29 are some examples of applications that preferably employ one or more algorithms of the type illustrated in the drawing figure. Such applications can include an interactive venue application 2910, a venue recommender 2912, an application for a venue to tailor its offerings to a customer 2914, and other applications that are responsive to one or more user's personal preferences or interests. A venue recommender was described above with reference to FIG. 28. A method for tailoring a venue's offerings to a particular customer was described above with reference to FIG. 27. An application for providing dynamic, interactive entertainment at a given venue, 2910, is described in greater detail in the following example.

Real-time Party Session

[00131] The following example begins with a summary description of a venue 3000 which may be a disco, bar or a nightclub, selected elements of which are illustrated in FIG. 30. In FIG. 30, the venue 3000 includes a sound system with several speakers 3004, at least one display screen 3002, at least one kiosk or terminal 3006, each of which are described in greater detail below. A venue might optionally include one or more cameras 3012 for taking photographs of guests 3008 during the course of a party or a session, as well as one or more video cameras 3014 for a similar purpose. The venue 3000 often includes a bar 3010 for dispensing beverages. The venue 3000 might also include a wifi access point 3016 for wireless communications.

[00132] As noted, a venue includes one or more display screens 3002 which are arranged for displaying content to the guests there assembled. The number and dimensions and the display screen or screens are not critical, but they should be arranged for easy viewing by the guests in general. As discussed below, the display screen 3002 provides the principal medium for interactive communications between the venue proprietor or "host" and the guest, as well as among the guests.

[00133] FIG. 31 is a simplified illustration of one example of venue display screen content. In FIG. 31, the display screen 3002 is currently showing an advertisement 3102 ("Bella Terra Bar and Grill"); and a welcome message from the host 3104 ("Welcome Partygoers. We're here with you tonight until 3:00 a.m."). The display screen example of FIG. 31 shows additional messages 3106 instructing a user on how to "join the party" i.e., how to log into the interactive system described herein. Hereinafter, we refer to a user who is currently logged into the venue system as a guest". FIG. 31 indicates two methods for user to log in and become a guest. The first is by sending an SMS message from a wireless device such as a cell phone or other multipurpose device such as a PDA. In fact, the messaging can be done from any wireless telecom messaging enabled device. The screen display message instructs the user to send a message having the format "PartyStrands J - nickname" to a specified telephone number. The format in this example is generically described as **[server name, party identifier, nickname]**. The server name, in this example "PartyStrands," refers to a remote server (in the presently preferred embodiment, one accessed via the Internet, as described later with reference to FIG. 39). Partystrands is a trademark of Mediastrands, Inc. The party identifier (in this example "J") identifies a particular party or venue that the user wishes to join, and finally, the user's nickname or user name. The telephone number provided corresponds to a wireless telecom gateway or the like where short messages can be received and routed as appropriate. In this case, the message will be routed to the PartyStrands server, further discussed below, and it will then log in the user "nickname" into the "Party J". In another embodiment, a dedicated telephone number can be provided for the specific venue, thus simplifying the messaging protocol. Details of the selected messaging format are not critical, but some examples follow.

A. Illustrative Message Formats.

[00134] The SMS messages to the system for each operation have a specific format. In addition, a response message is sent by the system to the user to acknowledge receipt of the user's SMS message by the system. As a practical matter, it is the response message back to the user for which the carrier bills the user and therefore for which the operator receives revenue.

[00135] (1) Join party:

Log into a specific party and local system, and provide an initial artist preference to the system for the system to include in building the playlist currently being performed.

Message: <party_name> <user_nickname> <artist_name>

where the <party_name> is an identifier for the venue or event that the user sees displayed on the screen and the <user_nickname> may be:

<nickname>	- join as anonymous user
<mystrands_alias> : mystrands	- join using MyStrands profile info
<mystrands_alias> : ms	- join using MyStrands profile info

(alternate form)

<last_fm_alias> : lastfm	- join using LastFM profile info
<myspace_alias> : myspace	- join using MySpace profile info

Response: <text_response>

where the <text_response> is a text message that states whether the user has successfully joined the party or not, and which may optionally provide additional explanatory information. Once the system indicates to the user that the user has joined the party, the user may send additional SMS messages to interact with the system and communicate with other users, for example:

[00136] (2) Text party:

Send a text message to be displayed by the system on the screen.

Message: <party_name> t <text_message>

where <text_message> is any text message the user would like to send. In some embodiments, the system scans the text to remove any offensive language before displaying the message on the screen.

Response: <text_response>

where <text_response> is a text message that states whether the user's message has been accepted by the system or not, and which may optionally provide additional explanatory information.

[00137] (3) Add artist:

Suggest a new artist to the system for the system to include in the process for building the playlist being performed.

Message: <party_name> a <artist_name>

where <text_message> is any text message the user would like to send. In some embodiments, the system scans the text to remove any offensive language before displaying the message on the screen.

Response: <text_response>

where <text_response> is a text message that states whether the user's suggested has been accepted by the system for inclusion in the playlist building process or not, and which may provide additional explanatory information.

[00138] (4) SMS another user: Send a private SMS message to another user logged into the party, as indicated by the presence of the user's avatar or alias appearing on the screen.

In one embodiment, the user sends a message for immediate delivery to the intended recipient with the form

Message where the <user_nickname> is the name the intended recipient provided to the system during login and which the system displays on the screen. The <text_message> is any text message the user would like to send to the intended recipient. In some embodiments, the system scans the text to remove any offensive language before displaying the message on the screen.

Response: <text_response>

where <text_response> is a text message that states whether the user's message has been accepted by the system for attempted delivery to the recipient, and which may optionally provide additional explanatory information.

[00139] In an another embodiment, the user sends a message as described, but a system operated by the Software/IP Owner 4000 "holds" the message for delivery until the intended recipient sends a message requesting delivery. The intended recipient would receive a message from the system indicating a message is waiting, and then text back a message requesting delivery of the message:

Message: <party_name> g

where <party_name> may be optional if the System/IP Owner system is configured to expect a response of this type from the intended recipient.

Response: <text_response>

where the <text_response> is the <text_message> sent initially by the user for delivery to the intended recipient.

[00140] In the former scheme, the Software/IP Owner may receive revenues from the mobile carrier linked only the SMS message sent by the initiating user. In the latter case, the Software/IP Owner may receive revenues from the mobile carrier linked to both the SMS message sent by the initiating user, and the message sent by the intended recipient requesting delivery of the initiating message. In addition, the avatars 3210 (FIG. 32) of users participating in the party may display a number indicating how many messages have been sent to them and/or how many messages are waiting for them for which they have requested delivery.

[00141] (5) Vote:

Send a vote in response to a question displayed on the screen.

Message: <party_name> v <vote>

where <vote> is the user's vote such as "y", "yes", "n", "no". <vote> may also be the identifier such as "a", "b", "c", etc. or "1", "2", "3", etc., for questions with multiple possible responses.

Response: <text_response>

where <text_response> is a text message that states whether the user's vote has been accepted for tallying by the system, and which may optionally provide additional explanatory information.

[00142] (6) Send Photo/Video to party:

Send a photo for display on the party screen using MMS, if available through a messaging provider:

Message: <party_name> p <photo_file> or <party_name> f <photo_file>

where the <photo_file> is a digital photo or video file in a format accepted by the MMS service.

Response: <text_response>

where <text_response> is a text message that states whether the user's photo or video has been accepted for tallying by the system, and which may optionally provide additional explanatory information.

B. Optional Messaging Format and Strategy.

[00143] In one preferred embodiment, the Software/IP Owner 4001 (FIG. 40) may contract with mobile phone network operators for a "non-shared short code" that corresponds to a dedicated phone number. In this embodiment, the

messages 3106 (FIG. 31), 3212 (FIG. 32), 3401 (FIG. 34), and 3506 (FIG. 35) may have the form:

"Send <message> to <short_code>"

where <message> is a message formatted as described earlier, and <short_code> is the text string of the short code provided to the Software/IP Owner by the mobile phone provider. On receipt of this message, the mobile phone provider would forward an message that includes the text <message> to the software/IP owner over a communications network such as the internet.

[00144] In another embodiment, the Software/IP Owner 4000 (FIG. 40) may not contract with mobile phone network operators for a "non-shared short code" that corresponds to a dedicated phone number, but instead may use a shared SMS delivery service such as 411Sync, or those provided by carriers such as Ericsson. Those services provide a shared phone number to which anyone may send an SMS message. The SMS message typically begins with an identifier keyword signifying a customer to whom the shared SMS delivery service should deliver the particular message. In this embodiment, the messages 3106 (FIG. 31), 3212 (FIG. 32), 3401 (FIG. 34), and 3506 (FIG. 35) may have the form:

"Send <ip_owner_keyword> <message> to <phone_number>"

where <ip_owner_keyword> is the identifier such as "fiesta" or "partyStrands" for the software/IP owner 4001 service, <message> is a message formatted as described earlier, and <phone_number> is the dedicated phone number of the shared SMS delivery service. On receipt of this message, the shared SMS delivery service would forward an message that includes the text <message> to the software/IP owner over a communications network such as the internet.

[00145] FIG. 36 also illustrates an alternative method for a user to log in to a party at a venue. That is, to access a kiosk or terminal physically located at the venue, where the user can log in. The terminal may be preconfigured to communicate to the "PartyStrands" server and automatically provide identification of the venue, since it is located at the venue. The user needs only to provide a nickname and, optionally, identification of the user's musical taste, for example in the form of a preferred song track, album or artist. If the user has a preexisting music profile stored at the remote server, that profile can be accessed in lieu of a favorite artist, etc. Whether contacted by SMS or through a terminal, the remote server determines the guest's musical taste, logs the user into the specified party,

and delivers that user's musical taste information to the algorithm that creates the play stream for that party. Finally, FIG. 31 also illustrates displaying pictures 3108 which may be taken from time to time at the venue, using a camera 3012 (FIG. 30), actuated automatically, by a guest or by the host bartender. Similarly, a video camera (3014 in FIG. 30) can be used to provide a live video feed for display 3110 on the display screen 3002.

[00146] FIG. 32 is another illustration of a display screen 3002 for use at a venue during a party session. This figure shows the display screen 3002 as divided into three primary regions; a top region 3202, a central region 3204, and a bottom region 3206. This arrangement of the display screen is merely illustrative. The number of operative regions is not critical, nor is their size or arrangement. In this example, by way of illustration, the top region 3202 includes a host message 3212 which again invites the user to log into the party using the SMS system. In the bottom region 3206, the song currently playing is identified. It could also illustrate the last song played and potentially the next song to be played on the play list. Identifications of songs can include associated artwork such as album covers or other graphics.

[00147] In FIG. 32, the central region 3204 illustrates a graphical display of the guests who are currently logged into the system. Preferably, guests are not identified by their real names, but can be identified by nicknames or an alias, optionally together with graphic or avatar. The avatars can be provided automatically by the system upon login of a guest. The system might take the user's gender into account in assigning an avatar, if the gender can be determined from the user's profile or login information.

[00148] Referring now to FIG. 33, the central region of the display screen 3002 is showing a graphical play list. That is, it displays indicia of several songs, including an indication of which song is currently playing, in this case indicated by dash line 3304. The play list 3300 is of course updated dynamically as the play list is executed. This display screen in FIG. 33 also illustrates an advertisement 3308 which can be a source of advertising revenue for the venue host. In the lower region of the display screen in FIG. 33, a message from the host indicating a bartender's special is displayed, 3306. While these various display elements are of course illustrated as static in these drawings, preferably they would be

animated in an actual implementation, and their content changing frequently to maintain the guests' interest.

[00149] FIG. 34 shows another example of a venue display screen content. This figure introduces the concept of one guest sending a message to another. In FIG. 34, a message 3400 in the upper region of the display invites the guests to send a message to another guest, and a further message at the lower region of the display screen, instructs the user to send a message (using SMS or the like). Here, the message has the format: **[server name, party, "MG" and nickname]**. In this case MG is an arbitrary code indicating a request to send the message to the user identified by "nickname". In this regard, when a user logs into the system (to become a guest), he or she may be given the option as to whether or not they wish to receive messages from other guests. The interactive venue application software can be configured to enforce that option or other appropriate rules for brokering messages.

[00150] FIG. 34 also illustrates a welcome message 3404 which can be displayed when a user logs in as a guest. Conversely, a salutation 3402 can be displayed when a guest is logged out of the venue. The logout can be effective by the user explicitly sending a message to that effect. Alternatively, if a user simply leaves a venue, the logout can be effective automatically after the passage of a predetermined period of time without receiving any messages from that user. In other words, the user's exit from the venue can be inferred. As noted earlier with reference to FIG. 32, a listing of the guests currently logged in can be displayed from time to time.

[00151] FIG. 35 introduces the concept of voting by the guests at a venue. Here, a play list 3502 is displayed in the central region of the display screen 3002. In this case, the currently playing song might be 3502, while 3504 is identified as the next song proposed to be played by the algorithm that is generating the play stream. In the upper region of the display screen, message 3500 invites the guests to vote on the next song. In the lower portion of the display screen, message 3506 provides instructions for interested guests to vote on the next song by sending an SMS message to a predetermined gateway, as noted above.

[00152] FIG. 36 shows a message 3600 reminding the guests that they have a limited time left to cast their vote on the proposed song 3504. Optionally, real

time voting results 3604 can be displayed as the votes come in. FIG. 36 also shows a welcome message 3602, reflecting that a new guest recently logged in. The track currently playing is also identified in a message 3606 in the lower region of the display screen (song title 3608).

[00153] In FIG. 37, there is an additional message 3700 again reminding the guests of the remaining time to vote. It should be noted that the various screen display examples discussed above are merely illustrative. The other arrangements can be used for displaying the content and described above, or similar content. Preferably, the actual screen displays are colorful, animated and frequently changed. The reader will appreciate that the screen display, and the selection of music playing at the venue, are driven by the users who are logged into that venue. FIG. 38 summarizes some of the types of content that can be displayed at the venue. These were mentioned earlier and will not be repeated here, as they are apparent in the drawing figure. Referring now to FIG. 39, it shows a simplified diagram illustrating one example of the principal connections and data flow paths involved in implementing dynamic interactive entertainment at a venue in accordance with aspects of the present invention. In FIG. 39, a horizontal dash line divides the components (and people) located at the venue 3000, shown on the lower half of the figure, from the equipment and people that are remote from the venue (shown above the horizontal dash line).

[00154] Near the center of the figure, an on-site host system 3900 is shown. This can be a conventional computer or a special purpose computer dedicated to the purposes described herein. The host system 3900 executes a computer program, namely an interactive venue application, to carry out the functions described herein, i.e. to manage a party session as described. The application may be installed locally or provided in a client-server implementation with respect to the party server 3902. In a presently preferred embodiment, the host system 3900 is connected to the remote party server 3902 in any event, as further described below. It can be so connected via the Internet, as is illustrated. The host system 3900 preferably controls a venue sound system 3910 to play the selections in accordance with the play stream generated by the software algorithms as described. The host system 3900 is also connected to drive a display system 3912 which includes one or more display screens 3002 as described above. The host system 3900 preferably is also coupled to one or

more terminals 3914 which are located at the venue. Such terminals, or kiosks, can provide various functions. For example:

1. A user who may not have a cell phone available can use the terminal to log in to the venue.
2. A user can use the terminal to access a remote Web site where she may update one or more play lists, thereby indirectly adjusting her preferences profile.
3. A terminal or kiosk could be used to download a selected music track to a personal device or player. For example, a guest might especially like one particular track that was recently played at the venue, and may decide to buy it at that time. The kiosk may be programmed to assist in that effort by displaying the play list that was played at the venue up to that time. Some online services may provide free media downloads.
4. The kiosk could be used to send a message to the public display or to upload a photo, video or other content for use potentially in the public display.
5. The kiosk could be used for voting on upcoming selections.
6. The kiosk could be used to access a music site on the Internet to search for music to then input to the local venue system. This input would not be received and then played as in a conventional "request" but instead would be input as an update to that particular user's profile which in turn will change how that guest influences the ongoing selection of tracks to be played at the venue.

[00155] The host system 3900 can also be coupled to one or more cameras 3916 as noted earlier, and/or one or more video cameras 3918 for a similar purpose. A lighting control system 3920 may also be interfaced to the host system 3900 in some applications to coordinate the venue lighting system with the music being played.

[00156] In FIG. 39, a box 3930 summarizes examples of the types of messages that guests can send, preferably using SMS or similar messaging system, for input to the host system 3900. In a presently preferred embodiment, the messages are received at a wireless telecom gateway 3932 and then routed via 3934 to the party server 3902. The party server 3902 processes these messages as appropriate and passes information down to the host system 3900, as indicated by a dash line 3940. In an alternative embodiment, the messages may

be routed directly to a "standalone" configuration of host system 3900. This routing is indicated by dash line 3942.

[00157] FIG. 39 also illustrates remote users 3944 who may be at home or at any other venue where they have access to a computer or wireless device that enables them to communicate via the Internet. The remote users 3944 can communicate with the party server 3902 to log into the venue/host system 3900 prior to arrival. Or, a remote user might wish to see a record of the venue play stream and other interactions by accessing a venue Web page, provided by the party server 3902. In other words, another aspect of the present invention provides for interaction between the on-site venue system and a home or other remote user mediated by a Web site. In the presently preferred embodiment, after joining the party (logging in), guests receive an SMS invitation to go to the party server Web site and review the party at their convenience. Content that appears on the venue's party display system is sent to the party server Web page, and associated with the day of the party. This information may include unique identifiers for all music tracks played, nicknames and avatars of guests, and user-generated party content, such as messages and pictures.

[00158] Guests can sign up as "PartyStrands" members (Web site community) to add comments to the venue Web page, and send messages to other PartyStrands members listed as attendees at the party. In addition, PartyStrands members/guests may have their own PartyStrands/personal Web pages from which they can receive and send messages to other members/guests.

[00159] As noted above with regard to FIG. 34, the system can enable one guest to send a personal message to another guest at the party. When a message is sent, a "message sent" screen can appear on the party display system. The screen display does not include the message content. A guest who receives a personal message can optionally reply via the party server (3902) system. In a presently preferred embodiment, this functionality is implemented through SMS.

[00160] FIG. 40 is a simplified schematic diagram illustrating one example of a business method in accordance with the present invention. In general, this business method seeks to reward venues (hosts) by providing a share of premium SMS and MMS revenues generated when the hosts use the software application to implement a party session as described above. Specifically, venue

hosts will receive a payment each time they initiate a party session and guests interact via SMS and/or MMS in connection with that session.

[00161] Referring to FIG. 40, an entity 4000 shown at the top may be the developer or owner of computer software for implementing aspects of the present invention, and, in any event, is the owner, agent or licensee of the owner of the intellectual property associated with the technologies that are brought to bear in building and using an interactive entertainment system as described herein.

Those technologies where they were described in detail earlier were summarized briefly with reference to FIG. 29. In accordance with one example of a business model, in accordance with the present invention, the software/IP owner 4000 distributes an interactive venue application which is executable on a host system (3900 in FIG. 39) to implement an interactive entertainment session at a venue (aka a "party") consistent with the foregoing description. The IP owner 4000 may distribute such software through one or more affiliates or distributors 4002. The affiliates, in turn, redistribute the software application to a plurality of venue hosts 4004, for example.

[00162] In the diagram of FIG. 40, one particular host (venue) 4006 is shown as receiving the application via a distributor 4002. Alternatively, a venue host may be able to download the application or otherwise acquire it directly from the IP owner 4000. The host then conducts a session (party) at her venue, as described previously, involving a number of party guests 4010 who attend the party session reference number 4012. As discussed above, the guests in attendance have various opportunities to interact with the host and/or with each other via SMS and/or MMS during that session. This is shown as "SMS traffic" 4016.

[00163] The SMS traffic is routed through a wireless carrier gateway 4018 as discussed above. The wireless carrier handling that traffic will in due course render billings 4020 to the guests who send the messages. Those party guests, in turn, will render payment (either directly or through their home carrier, etc.), to the wireless carrier billing and collections facility 4022. In one illustrative embodiment, a portion of that revenue or, more simply, an amount of money that is based upon the number of such messages, is paid by the wireless carrier to the IP owner 4000 as indicated by dash line 4024. The IP owner 4000 may distribute a portion of that revenue to the affiliate 4002 who distributed the

software and otherwise supports the corresponding venue host 4006. If the host 4006 acquired the software directly from the IP owner 4000, it may receive revenue directly from the IP owner 4000, again preferably based upon the number of messages originating at the party session conducted by that host.

[00164] This business model in various implementation allows broad and potentially free distribution of the interactive venue application software, while encouraging businesses (venues) who receive the application to promote its use, because they will share in revenue generated from appropriate use of the software. As noted, in some embodiments, the application software may be distributed through an affiliate. Affiliates may be paid for each copy of the software installed, or for each party session conducted by a host or at a venue to which they distributed the software. This business model encourages distribution of the software, which may be free, but also encourages affiliates to stay in touch with and support their clients (venues) to promote product use. In a presently preferred model, the affiliate would also receive a revenue share based on the number of SMS and MMS messages or based on the revenues generated by that message.

Recommender Example

[00165] The following section provides a detailed description of one example or implementation of recommender technology that can be used to provide individual mediaset recommendations. In addition, this recommender technology can be applied to groups of users as discussed above. For example, in the discussion above with reference to FIG. 2, a system provides a recommended mediaset for each user using a recommender engine (204, 212, and 220, respectively) that processes user tastes (mediasets 202, 210, and 218) to produce recommended mediasets 206, 214, and 222. One embodiment of such a recommender engine is described next. Others, for example the Yahoo! music recommender could be used.

[00166] By way of background, new technologies combining digital media item players with dedicated software, together with new media distribution channels through computer networks (e.g., the Internet) are quickly changing the way people organize and play media items. As a direct consequence of such evolution in the media industry, users are faced with a huge volume of available

choices that clearly overwhelm them when choosing what item to play in a certain moment. This overwhelming effect is apparent in the music arena, where people are faced with the problem of selecting music from very large collections of songs. However, in the future, we might detect similar effects in other domains such as music videos, movies, news items, etc.

[00167] In general, the following disclosure is applicable to any kind of media item that can be grouped by users to define "mediasets". For example, in the music domain, these mediasets are called playlists. Users put songs together in playlists to overcome the problem of being overwhelmed when choosing a song from a large collection, or just to enjoy a set of songs in particular situations. For example, one might be interested in having a playlist for running, another for cooking, etc.

[00168] Different approaches can be adopted to help users choose the right options with personalized recommendations. One kind of approach is about using human expertise to classify the media items and then use these classifications to infer recommendations to users based on an input mediaset. For instance, if in the input mediaset the item x appears and x belongs to the same classification as y , then a system could recommend item y based on the fact that both items are classified in a similar cluster. However, this approach requires an incredibly huge amount of human work and expertise. Another approach is to analyze the data of the items (audio signal for songs, video signal for video, etc) and then try to match users preferences with the extracted analysis. This class of approaches is yet to be shown effective from a technical point of view.

[00169] Our technology instead leverages the subjective judgment of a large collection of users, as reflected in their playlists, to make recommendations. Thus we address the challenge of assisting users in building their mediasets, or simply discovering new music selections, by recommending media items that go well together with an initial (or input) mediaset. The recommendation is computed using metrics among the media items of a knowledge base of the system. This knowledge base comprises collections of mediasets from a community of users. (As explained below, a mediaset is not a collection of media items or content. Rather, it is a list of such items, and may include various metadata.) Preferably, the methods of the present invention are implemented in computer software.

[00170] In commercial applications, features of the present technology can be deployed in various ways. Recommender services can be provided, for example, to remote users of client computing machines via a network of almost any kind, wired or wireless. Here we use “computing machines” to include traditional computers, as well as cell phones, PDA’s, portable music players etc. The knowledge base of the system, a database, can be local or remote from the user. It may be at one location or server, or distributed in various ways. As noted, the recommender engine can be applied to a group of users in connection with forming a “collective playlist,” i.e., a list of media items that a group of users probably will like.

[00171] The recommender technology in one aspect embodies a system for identifying a set of media items in response to an input set of media items. The system requires a knowledge base consisting of a collection of mediasets. Mediasets are sets of media items, which are naturally grouped by users. They reflect the users subjective judgments and preferences. The mediasets of the knowledge base define metrics among items. Such metrics indicate the extent of correlation among media items in the mediasets of the knowledge base. Various different metrics between and among media items can be generated from the knowledge base of mediasets. Such metrics can include but are not limited to the follow examples:

- a) Pre-concurrency (for ordered mediasets) between two items is computed as the number of times a given item precedes the other item in the mediasets of the knowledge base.
- b) Post-concurrency (for ordered mediasets) between two items is computed as the number of times an item follows another item in the mediasets of the knowledge base.
- c) Co-concurrency between two items is computed as the number of times the items appear together in a mediaset.
- d) Metadata similarities may be computed as well by considering keywords associated with the media items such as artist, actor, date, etc.
- e) Combinations of the previous metrics can be useful.
- f) Combinations of the previous metrics applying transitivity.

[00172] Such metrics can be represented in an explicit form that directly associates media items with other media items. For each media item of the input set, the system retrieves n media items with highest metrics. These media items are called candidates. Then, the recommended set of media items is a subset of the candidates that maximize an optimization criterion. Such criterion can be simply defined using the metrics of the knowledge base of the system.

Furthermore, such criterion can also include filters including but not limited to:

- a) Filters that the user expresses to focus the recommendation only on a determined type of items.
- b) Filters that the user expresses to focus the recommendations on items that meet certain keyword-based criteria, such as a specific artist/s, year/s, genre/s, etc.
- c) Filters that personalize the recommendations to the user. This kind of filtering includes recommending only items that the user knows about, or only items that the user does not know about, etc.

Recommender Definitions

[00173] A recommender system preferably comprises or has access to a **knowledge base** which is a collection of mediasets. A **mediaset** is a list of media items that a user has grouped together. A media **item** can be almost any kind of content; audio, video, multi-media, etc., for example a song, a book, a newspaper or magazine article, a movie, a piece of a radio program, etc. Media items might also be artists or albums. If a mediaset is composed of a single type of media items it is called a **homogeneous mediaset**, otherwise it is called a **heterogeneous mediaset**. A mediaset can be ordered or unordered. An **ordered mediaset** implies a certain order with respect to the sequence in which the items are used by the user. (Depending on the nature of the item, it will be played, viewed, read, etc.) Note again that a **mediaset**, in a preferred embodiment, is a list of media items, i.e. meta data, rather than the actual content of the media items. In other embodiments, the content itself may be included. Preferably, a knowledge base is stored in a machine-readable digital storage system. It can employ well-known database technologies for establishing, maintaining and querying the database.

[00174] In general, mediasets are based on the assumption that users group media items together following some logic or reasoning, which may be purely subjective, or not. For example, in the music domain, a user may be selecting a set of songs for driving, hence that is a homogeneous mediaset of songs. In this invention, we also consider other kinds of media items such as books, movies, newspapers, and so on. For example, if we consider books, a user may have a list of books for the summer, a list of books for bus riding, and another list of books for the weekends. A user may be interested in expressing a heterogeneous mediaset with a mix of books and music, expressing (impliedly) that the listed music goes well with certain books.

[00175] A *set* of *media* items is not considered the same as a mediaset. The difference is mainly about the intention of the user in grouping the items together. In the case of a mediaset the user is expressing that the items in the mediaset go together well, in some sense, according to her personal preferences. A common example of a music mediaset is a playlist. On the other hand, a set of media items does not express necessarily the preferences of a user. We use the term *set of media items* to refer to the input of the system of the invention as well as to the output of the system.

[00176] A metric M between a pair of media items i and j for a given knowledge base k expresses some degree of relation between i and j with respect to k . A metric may be expressed as a "distance," where smaller distance values (proximity) represent stronger association values, or as a *similarity*, where larger similarity values represent stronger association values. These are functionally equivalent, but the mathematics are complementary. The most immediate metric is the *co-concurrency* (i, j, k) that indicates how many times item i and item j appear together in any of the mediasets of k . The metric *pre-concurrency* (i, j, k) indicates how many times item i and item j appear together but i before j in any of the mediasets of k . The metric *post-concurrency* (i, j, k) indicates how many times item i and item j appear together but only i after j in any of the mediasets of k . The previous defined metrics can also be applied to considering the immediate sequence of i and j . So, the system might be considering co/pre/post-concurrencies metrics but only if items i and j are consecutive in the mediasets (i.e., the mediasets are ordered). Other metrics can be considered and also new ones can be defined by combining the previous ones.

[00177] A metric may be computed based on any of the above metrics and applying transitivity. For instance, consider co-concurrency between item i and j , $co(i,j)$, and between j and k , $co(j,k)$, and consider that $co(i,k)=0$. We could create another metric to include transitivity, for example $d(i,k) = 1/co(i,j) + 1/co(j,k)$. These type of transitivity metrics may be efficiently computed using standard branch and bound search algorithms. This metric reveals an association between items i and k notwithstanding that i and k do not appear within any one mediaset in K .

[00178] A matrix *representation of metric M*, for a given knowledge base K can be defined as a bidimensional matrix where the element $M(i, j)$ is the value of the metric between the media item i and media item j .

[00179] A graph *representation for a given knowledge base k*, is a graph where nodes represent media items, and edges are between pairs of media items. Pairs of media items i, j are linked by labeled directed edges, where the label indicates the value of the similarity or distance metric $M(i,j)$ for the edge with head media item i and tail media item j .

[00180] One embodiment of the present technology is illustrated by the flow diagram shown in Fig 42. This method accepts an input set 4201 of media items. Usually, this is a partial mediaset, i.e. a set of media items (at least one item) that a user grouped together as a starting point with the goal of building a mediaset. A first collection of candidate media items most similar to the input media items is generated by process 4202 as follows.

[00181] As a preliminary matter, in a presently preferred embodiment, a pre-processing step is carried out to analyze the contents of an existing knowledge base. This can be done in advance of receiving any input items. As noted above, the knowledge base comprises an existing collection of mediasets. A knowledge base includes a plurality of mediasets; each mediaset comprising at least two media items. The presence of media items within a given mediaset creates an association among them.

[00182] Pre-processing analysis of a knowledge base can be conducted for any selected metric. In general, the metrics reflect and indeed quantify the association between pairs of media items in a given knowledge base. The process is described by way of example using the co-concurrency metric mentioned earlier. For each item in a mediaset, the process identifies every

other item in the same mediaset, thereby defining all of the pairs of items in that mediaset. This process is repeated for every mediaset in the knowledge base, thus every pair of items that appears in any mediaset throughout the knowledge base is defined.

[00183] Next, for each pair of media items, a co-concurrency metric is incremented for each additional occurrence of the same pair of items in the same knowledge base. For example, if a pair of media items, say the song "Uptown Girl" by Billy Joel and "Hallelujah" by Jeff Buckley, appear together in 42 different mediasets in the knowledge base (not necessarily adjacent one another), then the co-concurrency metric might be 42 (or some other figure depending on the scaling selected, normalization, etc. In some embodiments, this figure or co-concurrency "weight" may be normalized to a number between zero and one.

[00184] Referring now to FIG. 41A, matrix 4100 illustrates a useful method for storing the metric values or weights for any particular metric. Here, individual media items in the knowledge base, say $m_1, m_2, m_3 \dots m_k$ are assigned corresponding rows and columns in the matrix. In the matrix, the selected metric weight for every pair of items is entered at row, column location x,y corresponding to the two media items defining the pair. In FIG. 41A, the values are normalized.

[00185] Now we assume an input set of media items is received, step 4201. Referring again to process step 4202, a collection of "candidate media items" most similar to the input media items is generated, based on a metric matrix like matrix 4100 of FIG. 41A. For instance, for each media item, say (item m_2) in the input set 4201, process 4202 could add to a candidate collection of media items every media item ($m_1, m_3 \dots m_k$ in Fig. 41A) that has a non-zero similarity value, or exceeds a predetermined threshold value, in the corresponding row 4102 of metric matrix 4100 for the media item m_2 , labeling each added media item with the corresponding metric value (0.7, 0.4 and 0.1, respectively). See the edges in the graph of Fig. 41B. For each media item in the input set of size m , process 4202 selects n media items as candidates; thus the aggregation of all the candidates produces a set of at most $m * n$ media items.

[00186] In one embodiment, a process 4203 receives the candidate set from process 4202 which contains at the most $m * n$ media items. This component selects p elements from the $m * n$ items of the candidate set. This selection can

be done according to various criteria. For example, the system may consider that the candidates should be selected according to the media item distribution that generated the candidate set. This distribution policy may be used to avoid having many candidates coming from very few media items. Also, the system may consider the popularity of the media items in the candidate set. The popularity of a media item with respect to a knowledge base indicates the frequency of such media item in the mediasets of the knowledge base.

[00187] Finally, from the second collection of $[p]$ media items, a third and final output set 4205 of some specified number of media items is selected that satisfy any additional desired external constraints by a filter process 4204. For instance, this step could ensure that the final set of media items is balanced with respect to the metrics among the media sets of the final set. For example, the system may maximize the sum of the metrics among each pair of media items in the resulting set. Sometimes, the system may be using optimization techniques when computation would otherwise be too expensive. Filtering criteria such as personalization or other preferences expressed by the user may also be considered in this step. In some applications, because of some possible computational constraints, these filtering steps may be done in the process 4203 instead of 4204. Filtering in other embodiments might include genre, decade or year of creation, vendor, etc. Also, filtering can be used to demote, rather than remove, a candidate output item.

[00188] In another embodiment or aspect of the invention, explicit associations including similarity values between a subset of the full set of media items known to the system, as shown in graph form in FIG. 41B, may be used. Fig. 41B shows a portion of a directed graph 4300, reflecting various media items (4302 or m_1 , 4304 or m_2 , 4308 or m_k etc.) Edge 4306 shows the similarity value (0.1) from m_1 to m_2 . To illustrate, if the similarity value between a first media item 4302, generally denoted below by the index i , and a second media item, say 4314, generally denoted below by the index j , is not explicitly specified, an implicit similarity value can instead be derived by following a directed path such as that represented by edges 4310 and 4312 from the first media item to an intermediate item, and finally to the second media item of interest, in this example item m_p . Any number of intermediate items can be traversed in this manner, which we call a transitive technique. The list of similarity values $M(i, i+1)$, $M(i+1, i+2)$, ..., $M(i+k$,

j) between pairs of media items along this path through the graph are combined in a manner such that the resulting value satisfies a definition of similarity between media item i and media item j appropriate for the application. For example, the similarity $M(i,j)$ might be computed as:

$$M(i,j) = \min\{M(i,i+1), M(i,i+2), \dots, M(i+k,j)\}$$

or

$$M(i,j) = M(i,i+1) * M(i,i+2) * \dots * M(i+k,j)$$

[00189] Other methods for computing a similarity value $M(i,j)$ for the path between a first media item i and a second, non-adjacent media item j where the edges are labeled with the sequence of similarity values $M(i, i+1)$, $M(i+1, i+2)$, ..., $M(i+k, j)$ can be used. From the user standpoint, this corresponds to determining an association metric for a pair of items that do not appear within the same mediaset.

[00190] The above description fully discloses the invention including preferred embodiments thereof. Without further elaboration, it is believed that one skilled in the art can use the preceding description to utilize the invention to its fullest extent. Therefore the examples and embodiments disclosed herein are to be construed as merely illustrative and not a limitation of the scope of the present invention in any way.

[00191] It will be obvious to those having skill in the art that many changes may be made to the details of the above-described embodiments without departing from the underlying principles of the invention. Therefore, it is to be understood that the invention is not to be limited to the specific embodiments disclosed and that modifications and other embodiments are intended to be included within the scope of the appended claims. For example, one of ordinary skill in the art will understand various aspects of the embodiments disclosed herein could be used in any system for building and sharing a composite playlist from collective group tastes on multiple media playback devices.

[00192] The scope of the present invention should, therefore, be determined only by the following claims.

Claims

1. A method for generating a group playlist from a plurality of individual mediasets of users within a group, each user mediaset comprising a respective plurality of media data items, the method comprising:
 - performing a taste analysis on a first one of the user mediasets to generate a first sub-mediaset from the first user mediaset;
 - performing a taste analysis on a second one of the user mediasets to generate a second sub-mediaset from the second user mediaset; and
 - aggregating media data items from the first and second sub-mediasets to generate a group playlist.
2. The method of claim 1, wherein the media data items comprise playable media files.
3. The method of claim 1, wherein the media data items each comprise metadata that identifies a playable media file.
4. The method of claim 1, further comprising:
 - generating a recommended mediaset for the first user based on the first user taste analysis; and
 - generating a recommended mediaset for the second user based on the second user taste analysis.
5. The method of claim 4, wherein each of the recommended mediasets comprises metadata that identifies a playable media file.
6. The method of claim 1, wherein the step of aggregating the media data items comprises ranking individual media data items by the number of sub-mediasets in which they appear.
7. The method of claim 1, wherein the group playlist includes media data items comprising songs.

8. The method of claim 1, wherein the step of aggregating the media data items comprises selecting a set of media data items that optimizes a utility function for the group.
9. The method of claim 8, wherein the utility function is based on weights associated with individual media data items.
10. The method of claim 9, wherein the step of selecting a set of media data items that optimizes a utility function for the group comprises selecting a set of media data items so as to maximize a sum of selected media data item weights.
11. The method of claim 10, wherein the step of selecting a set of media data items so as to maximize a sum of selected media data item weights comprises selecting the media data items that maximize a sum of the highest weights for each of a plurality of different media data items in a plurality of user mediasets.
12. The method of claim 9, wherein the step of selecting a set of media data items that optimizes a utility function for the group comprises selecting the media data items that maximize a sum of average media data item weights.
13. The method of claim 1, wherein each of the media data items in the group playlist can be found in a user mediaset within the group.
14. A method for generating a dynamic, composite playlist of media data items for a user group, comprising:
 - retrieving media item taste data for a plurality of users, wherein the user taste data for a given user is retrieved in response to the given user joining the user group;
 - aggregating the media item taste data in the user group to create a set of optimal playlist characteristics for the user group; and
 - adding media data items to a group playlist based at least in part on the set of optimal playlist characteristics.

15. The method of claim 14, wherein the user taste data for a given user is automatically retrieved as the given user joins the user group.
16. The method of claim 14, wherein media data items associated with a given user are retrieved in response to the given user joining the user group.
17. The method of claim 14, wherein the media data items added to the group playlist are selected from a central database of media data items, and wherein the content of the database is independent of the users in the user group.
18. The method of claim 14, wherein the user taste data for a given user is removed from the aggregated user taste data in response to the given user exiting the user group.
19. The method of claim 14, wherein media data items in the group playlist that were retrieved from a given user are removed from the group playlist in response to the given user exiting the user group.
20. The method of claim 14, further comprising subjecting the media data items in the group playlist to a diversification process.
21. The method of claim 20, wherein the diversification process comprises ensuring that no media data item is repeated in the group playlist for a predetermined period of time.
22. The method of claim 20, wherein the diversification process comprises shuffling media data items in the group playlist to diversify at least one segment of the group playlist that includes media data items that are overly representative of the tastes of one or more users.
23. The method of claim 20, wherein the diversification process comprises determining whether the group playlist is dominated by media data items representative of the taste of at least one user and, in response to determining that the group playlist is dominated by media data items representative of the

taste of the at least one user, removing at least one media data item from the group playlist that corresponds with the taste of the at least one user.

24. The method of claim 14, wherein the user taste data for a given user is retrieved in response to the given user logging into a network of connected devices to join the user group.

25. The method of claim 24, wherein the step of logging into the network is accomplished implicitly as the network-connected device of a given user enters a given proximity.

26. A system for building a composite playlist from the collective tastes of a plurality of users in a session, each having a separate media device, comprising:

a plurality of session managers, each associated with one of a plurality of media devices, wherein each session manager is configured to manage the status of the media device with which it is associated;

a playlist builder for receiving media item data from the plurality of media devices and for building a composite playlist of media data items from the media item data; and

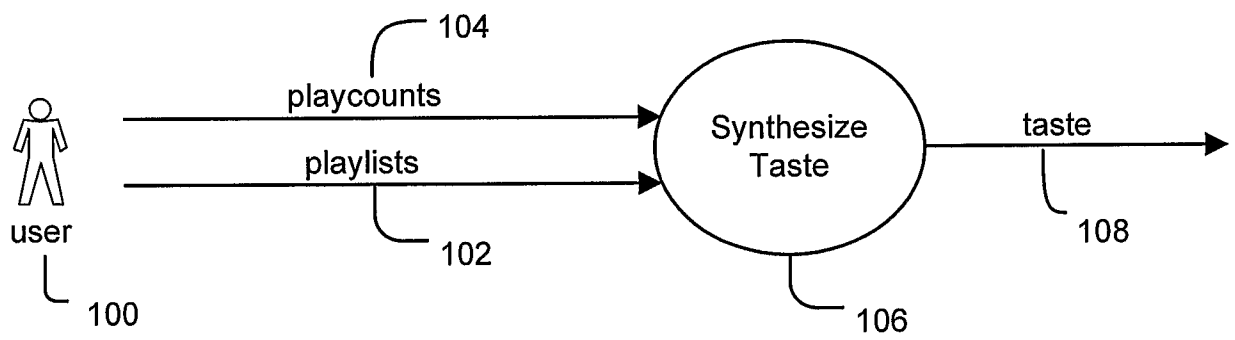
a plurality of playlist managers, each associated with one of a plurality of media devices, wherein each playlist manager is configured to communicate with the playlist builder to send the media item data of the media device with which it is associated to the playlist builder, and wherein each playlist manager is further configured to play the media data items on the composite playlist on the media device with which it is associated.

27. The system of claim 26, wherein the media item data comprises media item taste data.

28. The system of claim 26, wherein the media item data comprises metadata that identifies a playlist of playable media files.

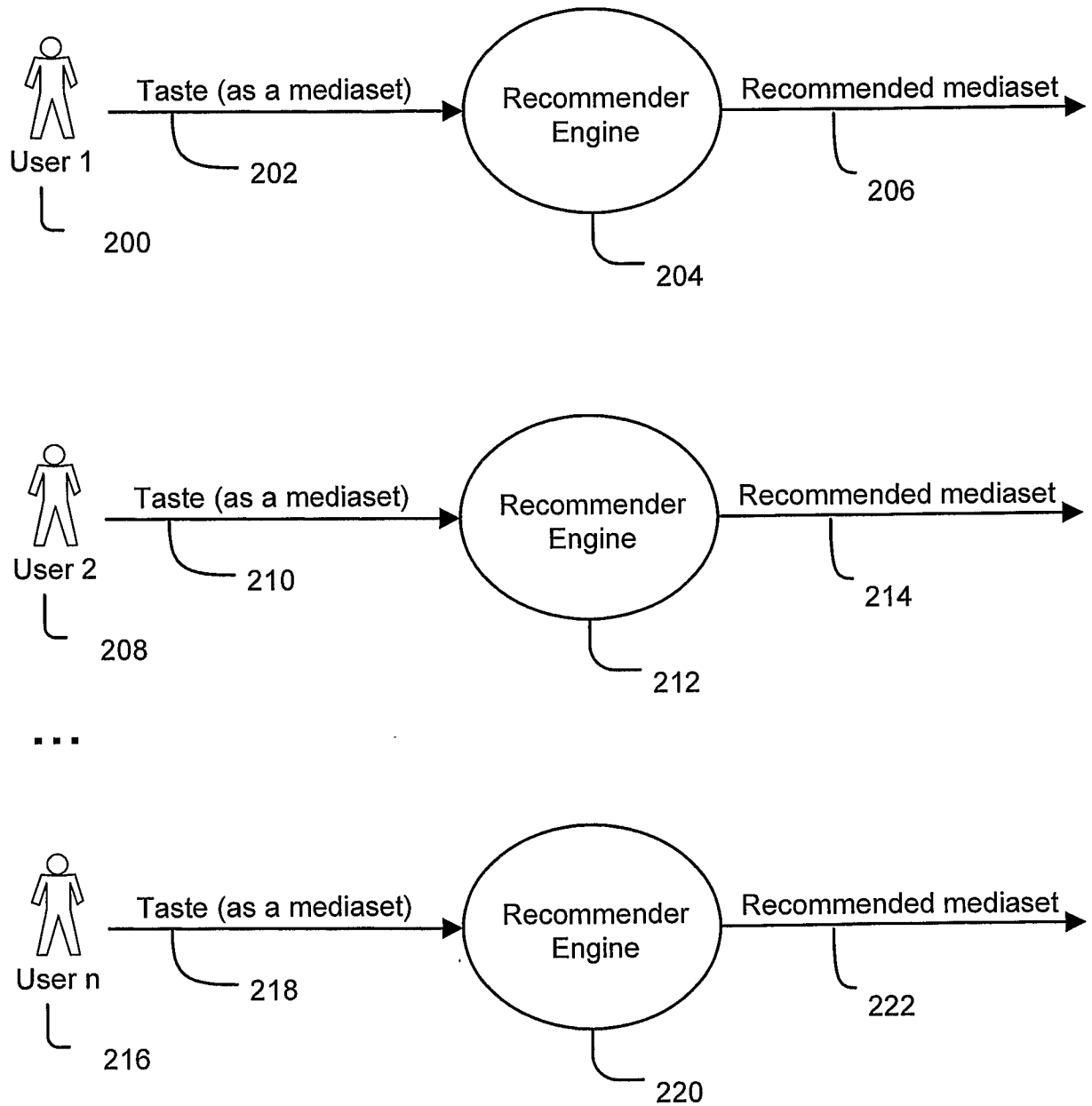
29. The system of claim 26, wherein the playlist builder resides on a central server.

30. The system of claim 29, wherein the plurality of session managers reside on the server.
31. The system of claim 26, wherein the session managers are configured to verify the eligibility of media devices to participate in the session.
32. The system of claim 26, wherein each playlist manager is configured to stream media files to other media players in the session.
33. The system of claim 32, wherein the system is configured such that each media data item in the composite playlist is located on at least one of the media devices in the session, and wherein each individual media data item on the composite playlist is played at least substantially simultaneously on each of the media devices in the session by causing each of the media data items in the composite playlist to be streamed from its corresponding media device to each of the other media devices in the session.



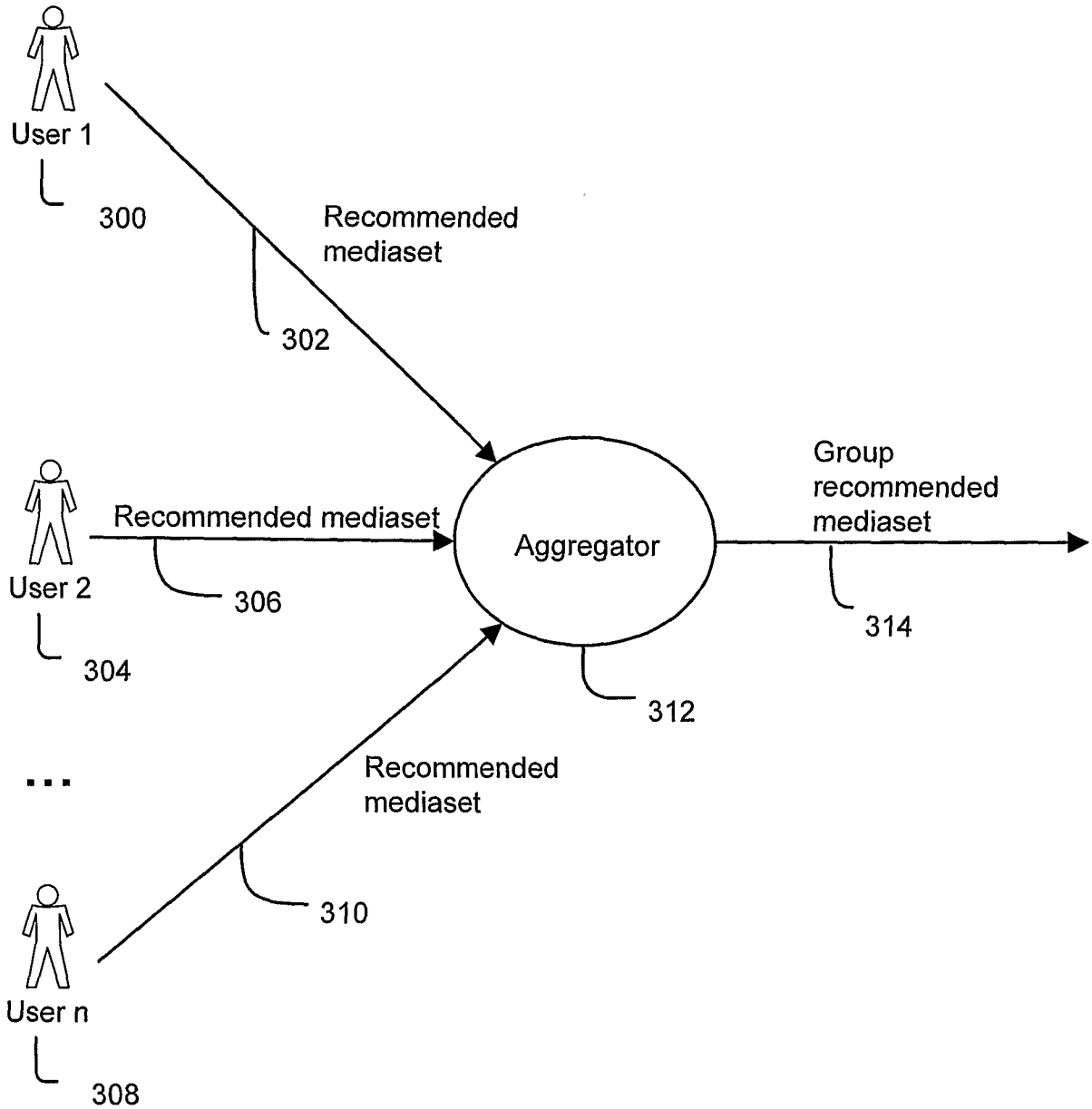
Synthesizing user tastes from playcounts and playlists.

FIG. 1



Getting individual recommended mediasets.

FIG. 2



Aggregating individual recommended mediasets into a recommended mediaset suitable for the whole group of users.

FIG. 3

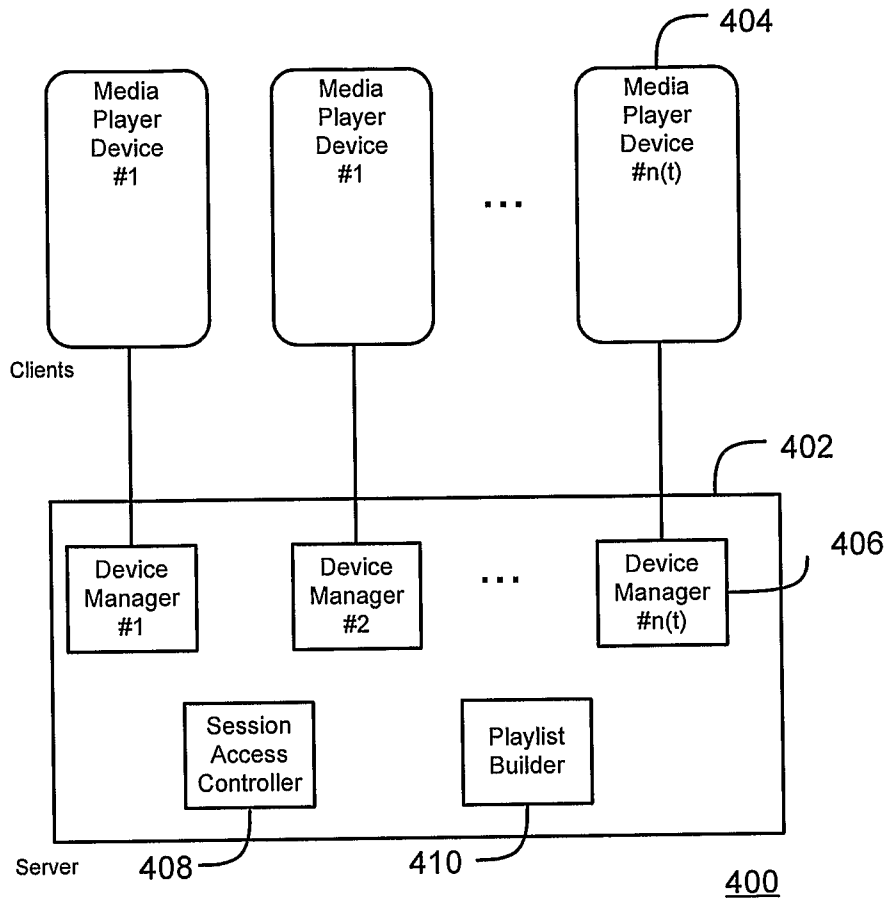


FIG. 4

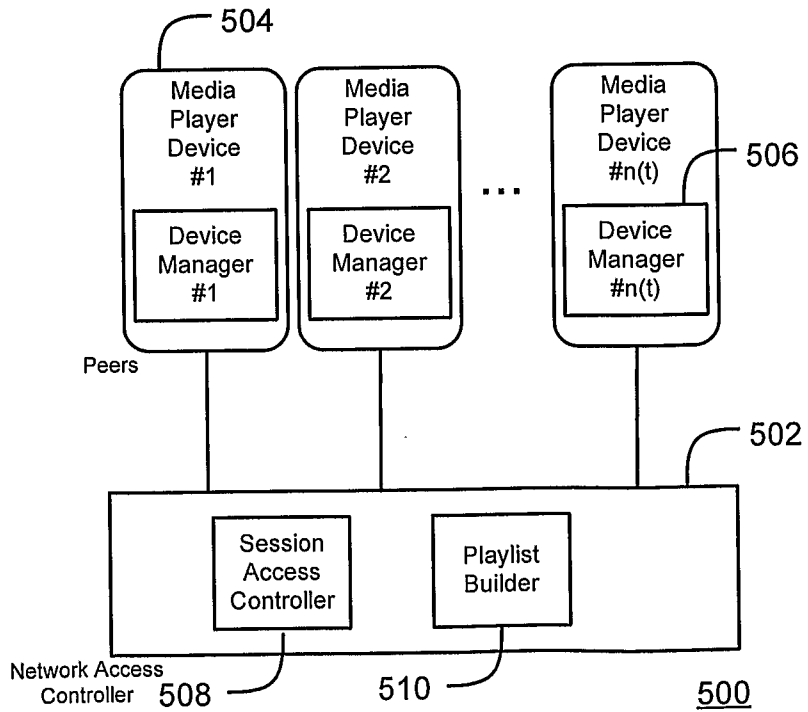


FIG. 5

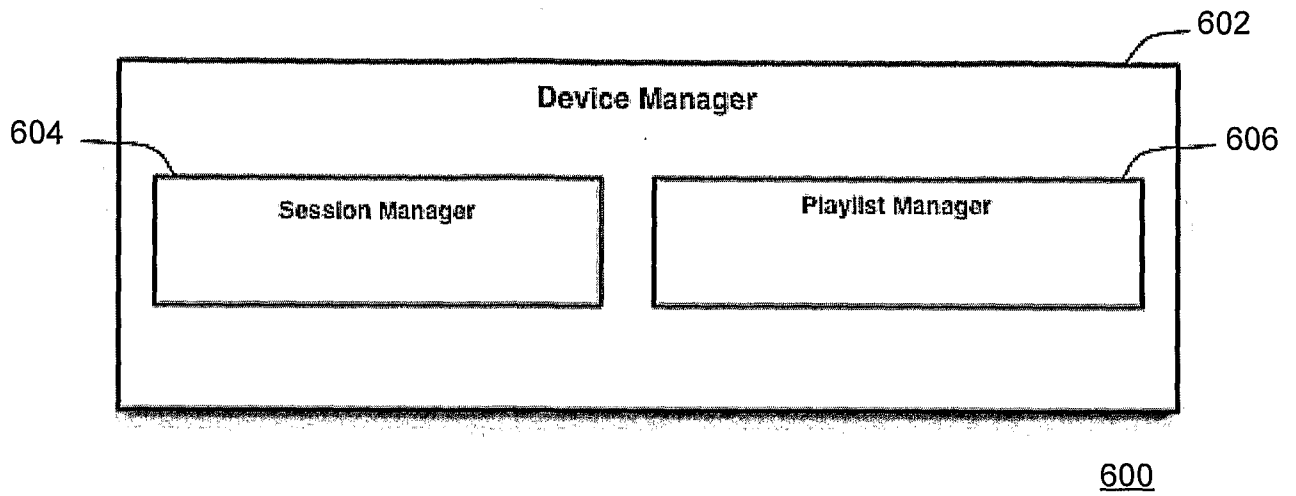


FIG. 6

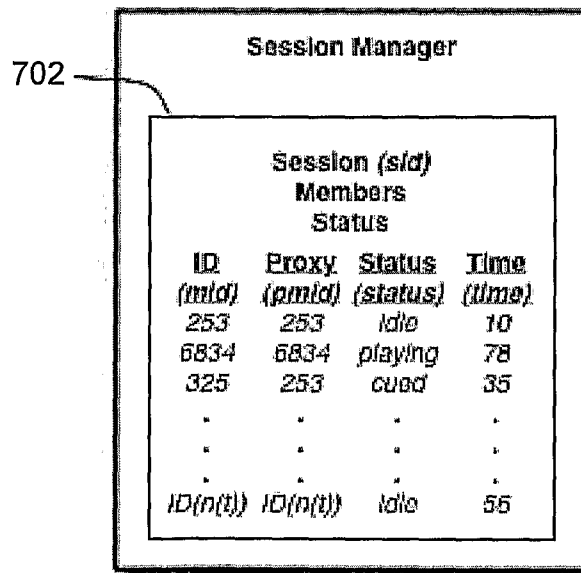
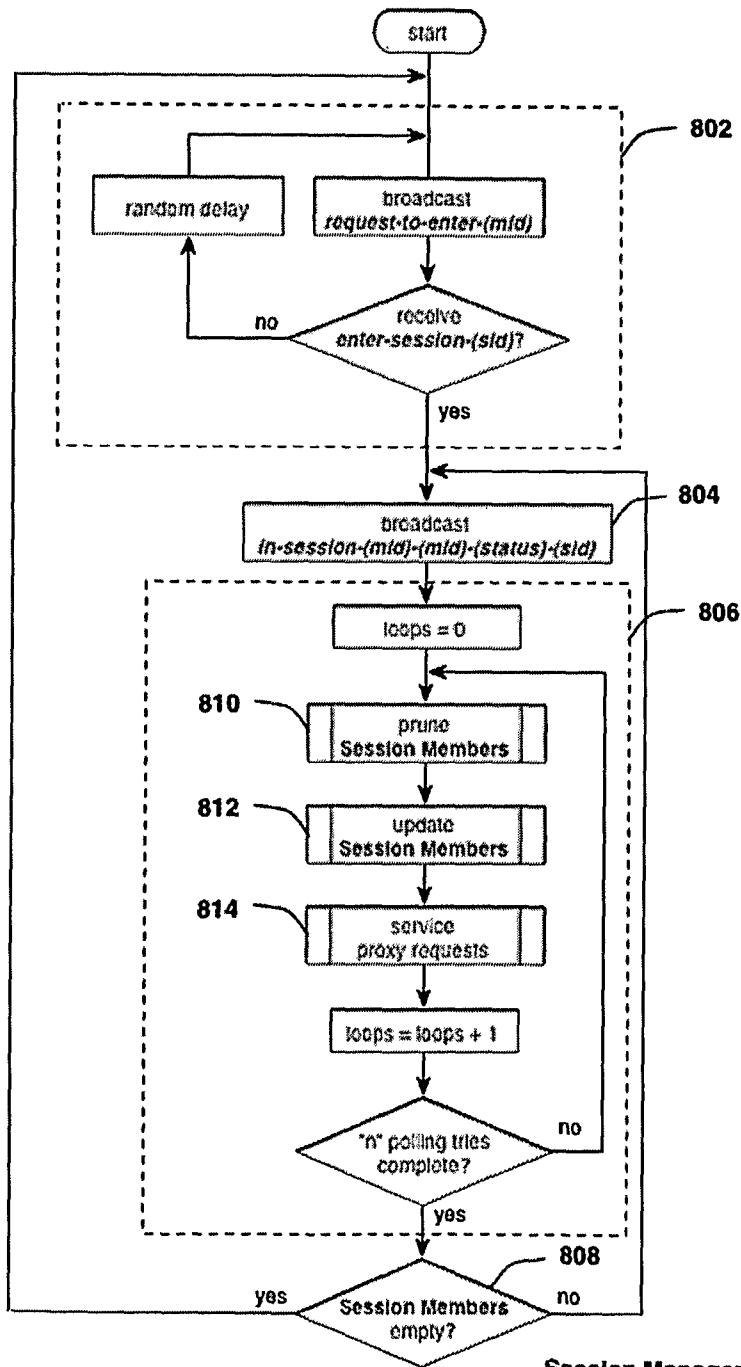
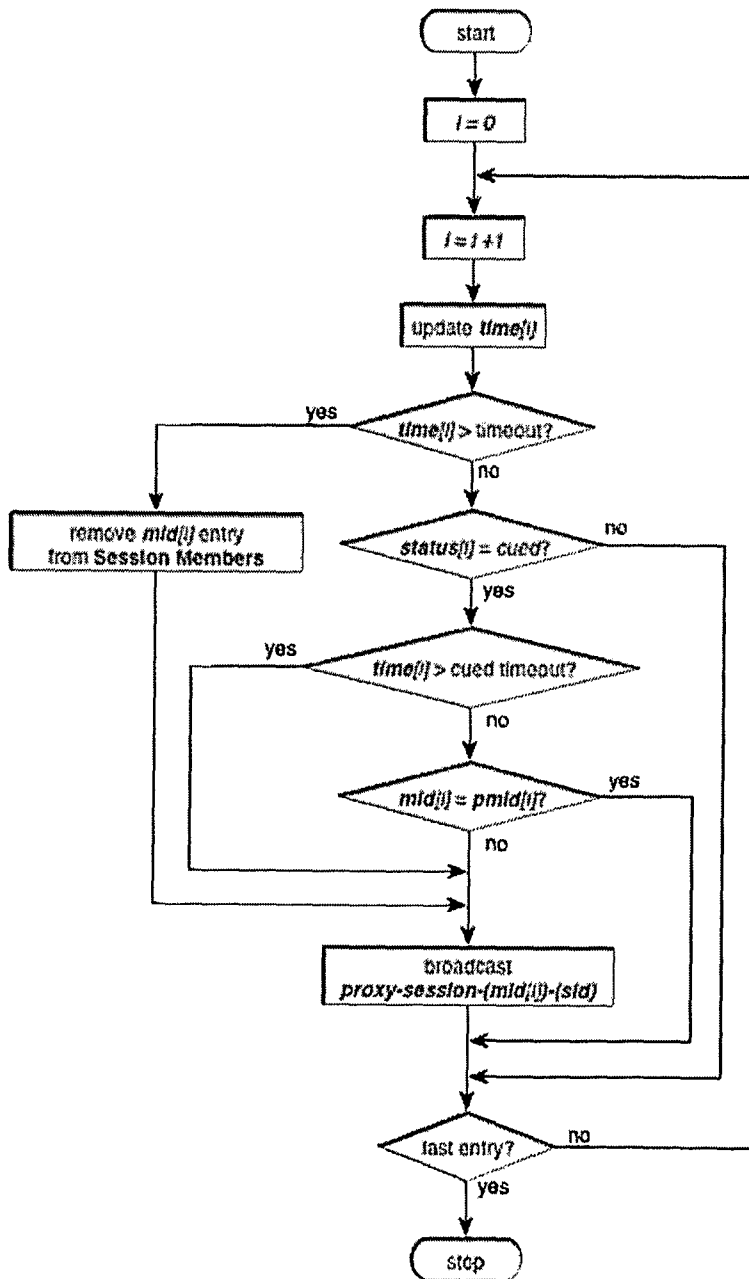


FIG. 7



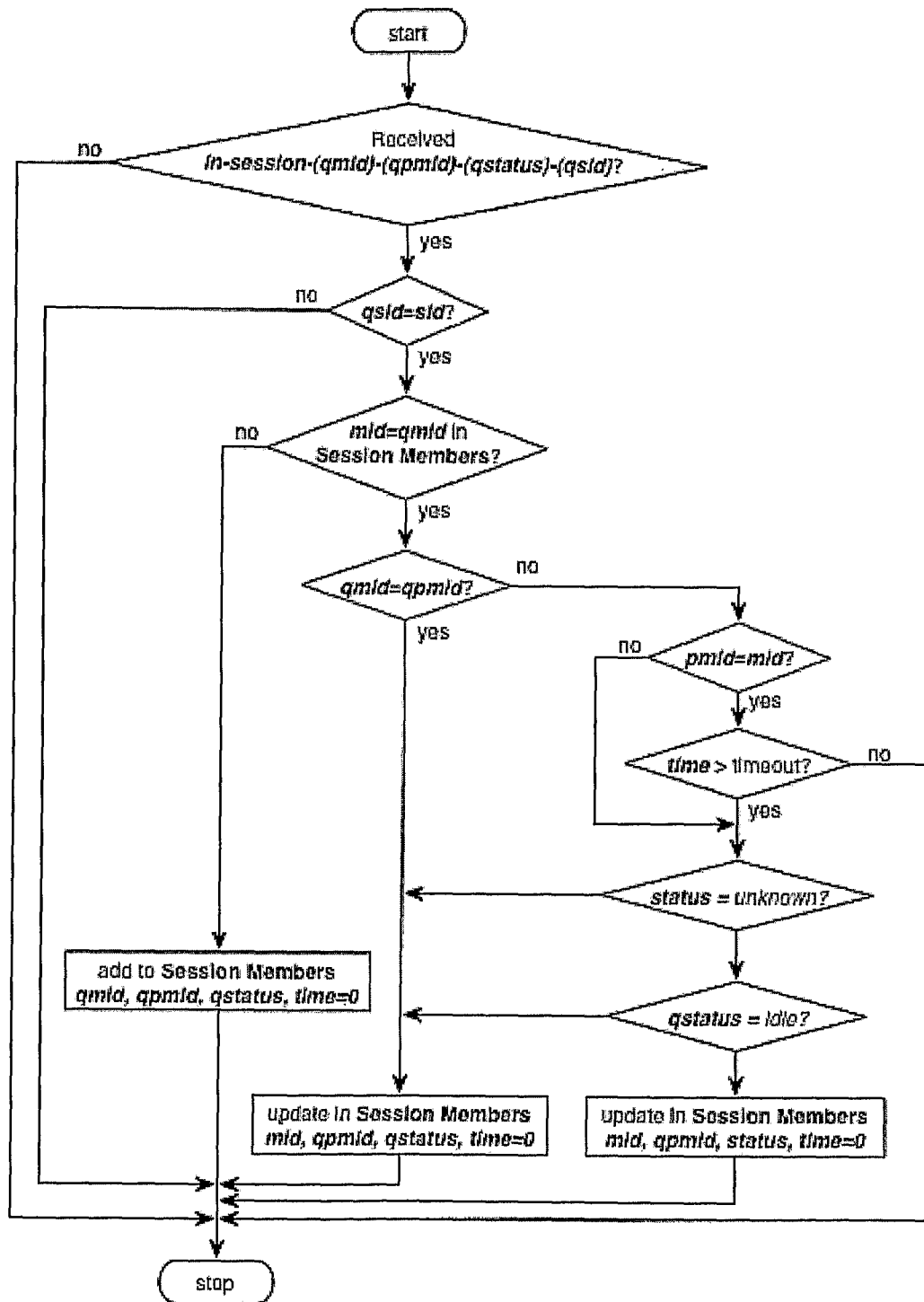
Session Manager Flow Diagram

FIG. 8



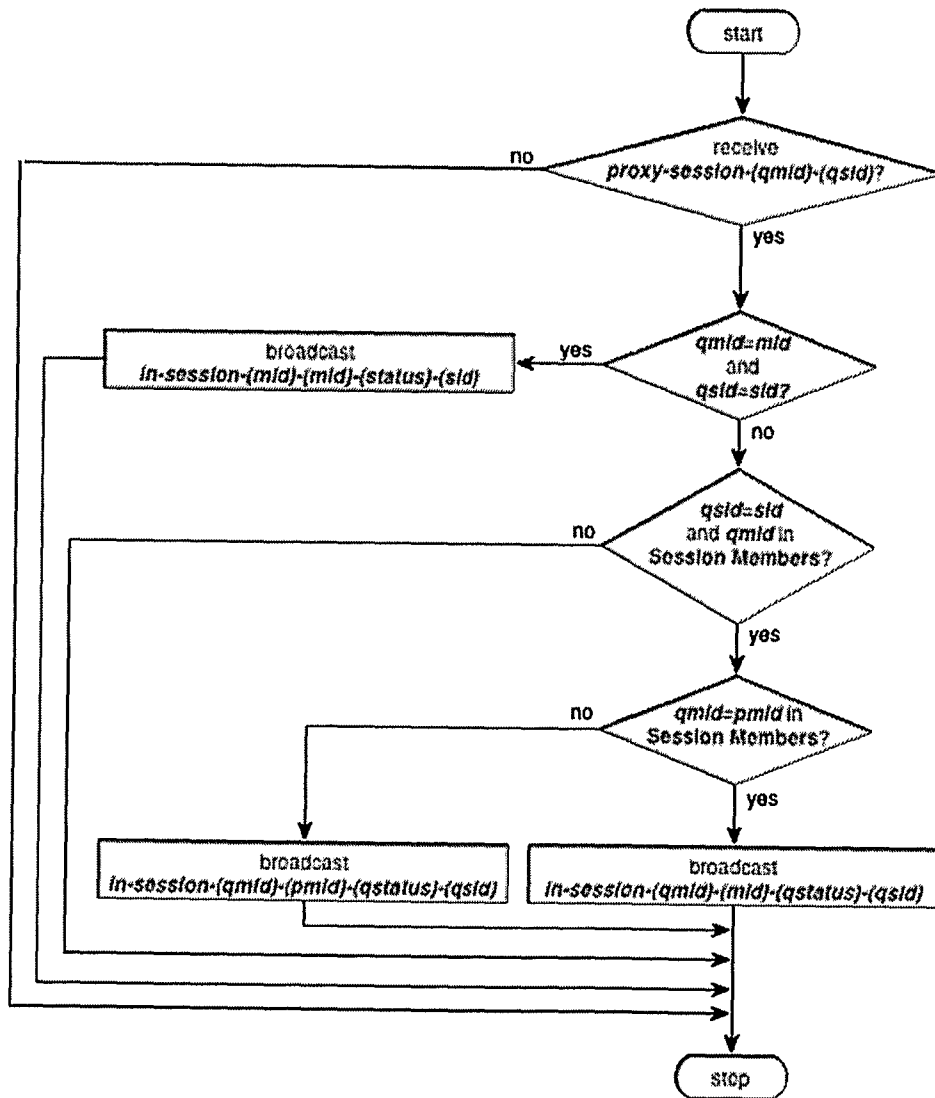
Prune Session Members Flow Diagram

FIG. 9



Update Session Members Flow Diagram

FIG. 10



Service Proxy Requests Flow Diagram

FIG. 11

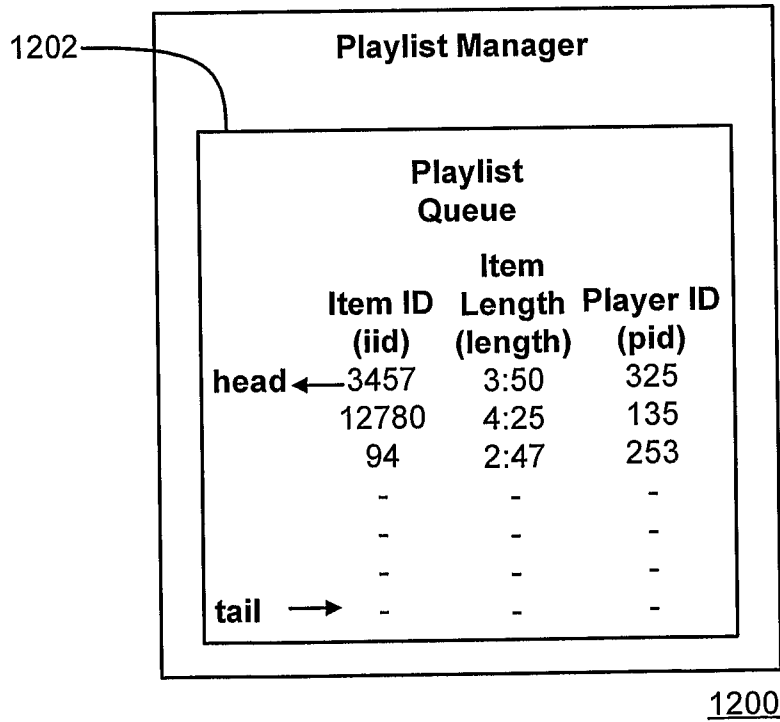
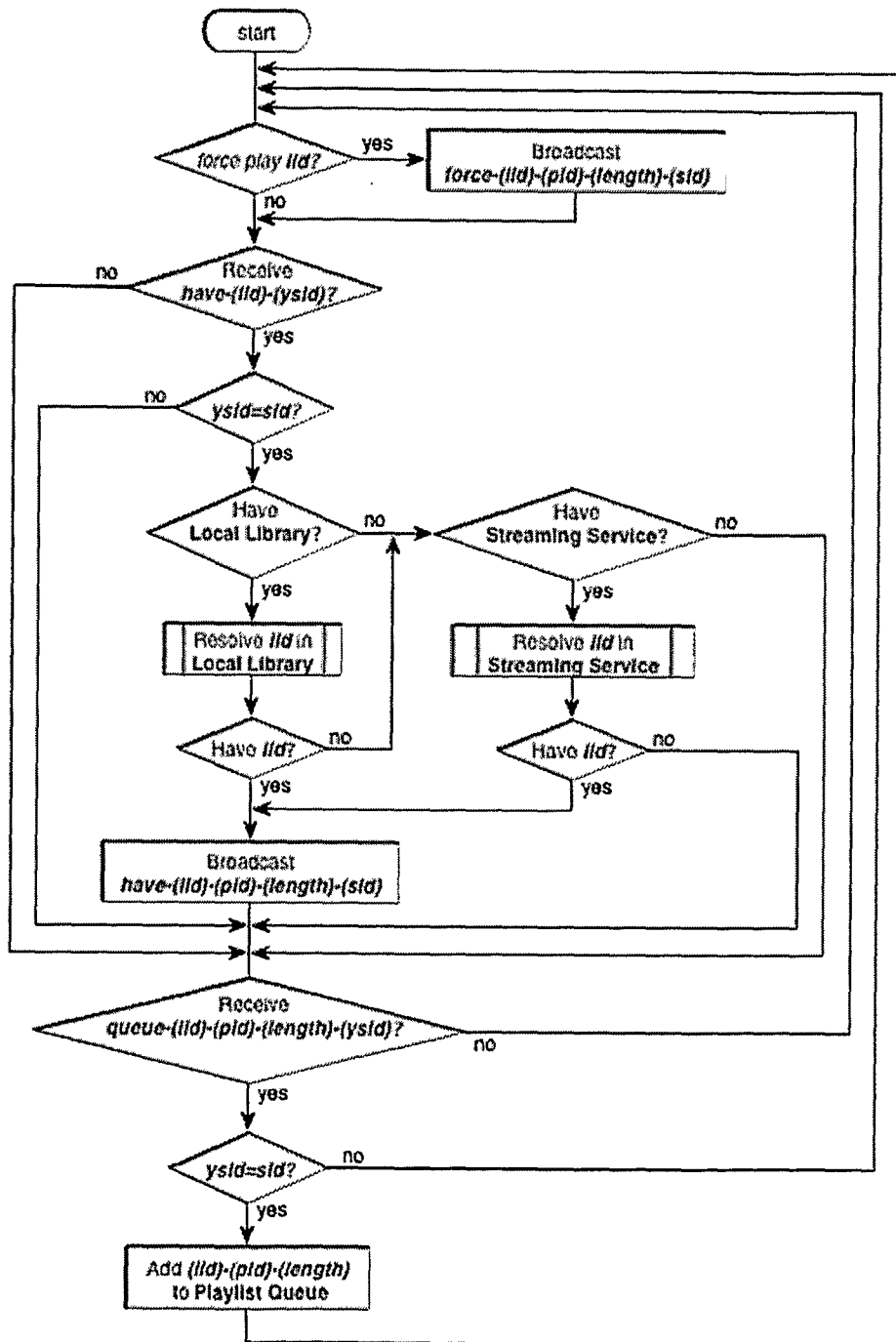


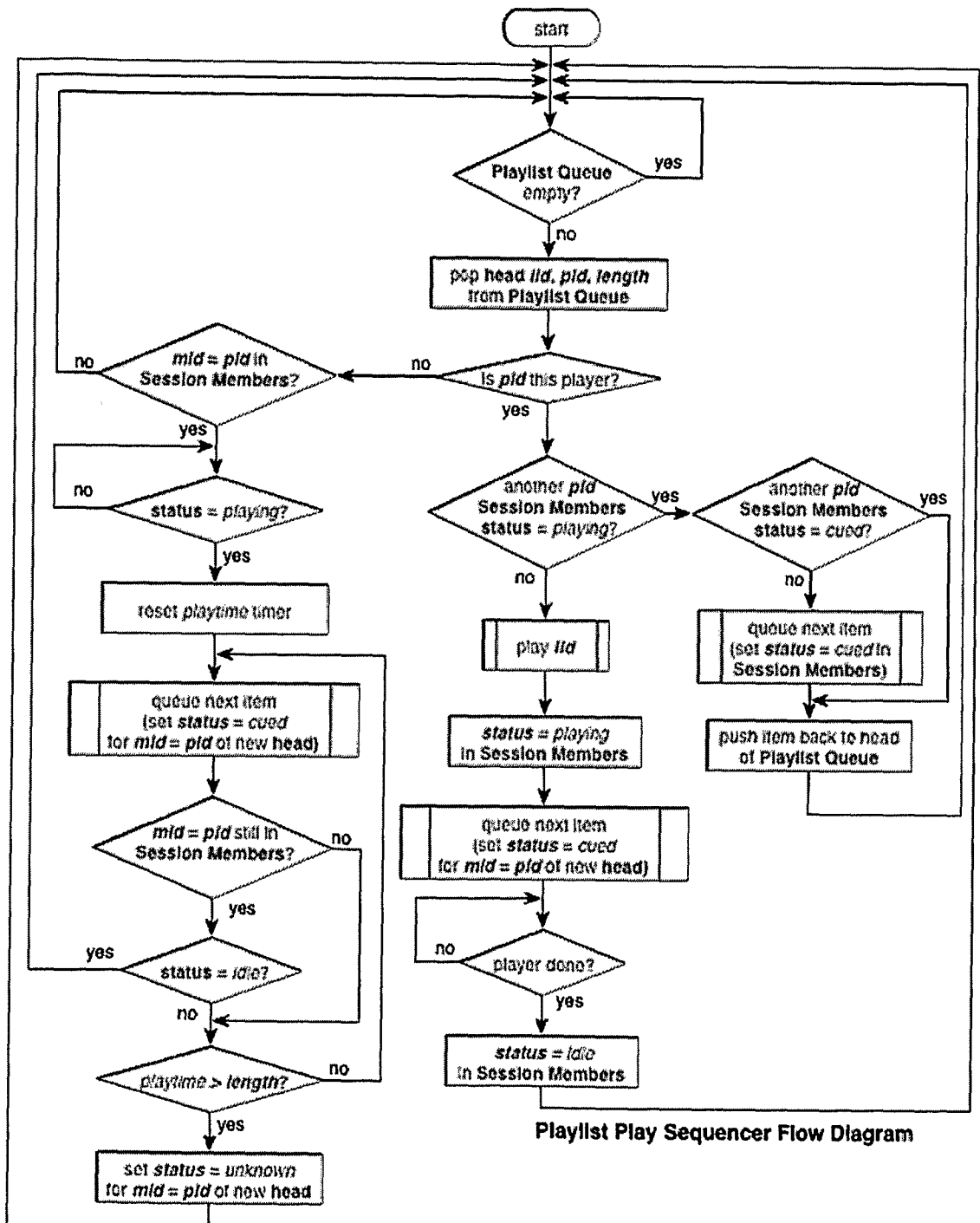
FIG. 12



1300

Playlist Queue Updater Flow Diagram

FIG. 13



Playlist Play Sequencer Flow Diagram

1400

FIG. 14

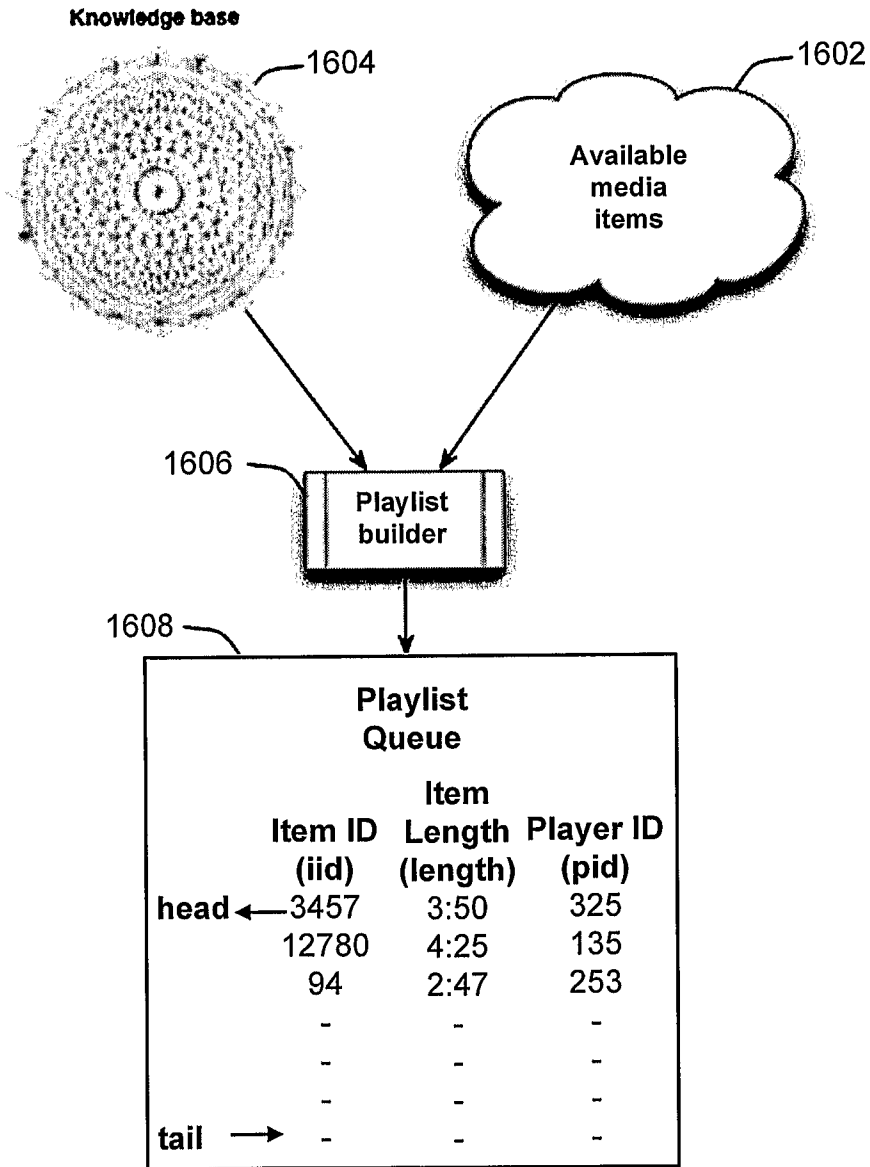
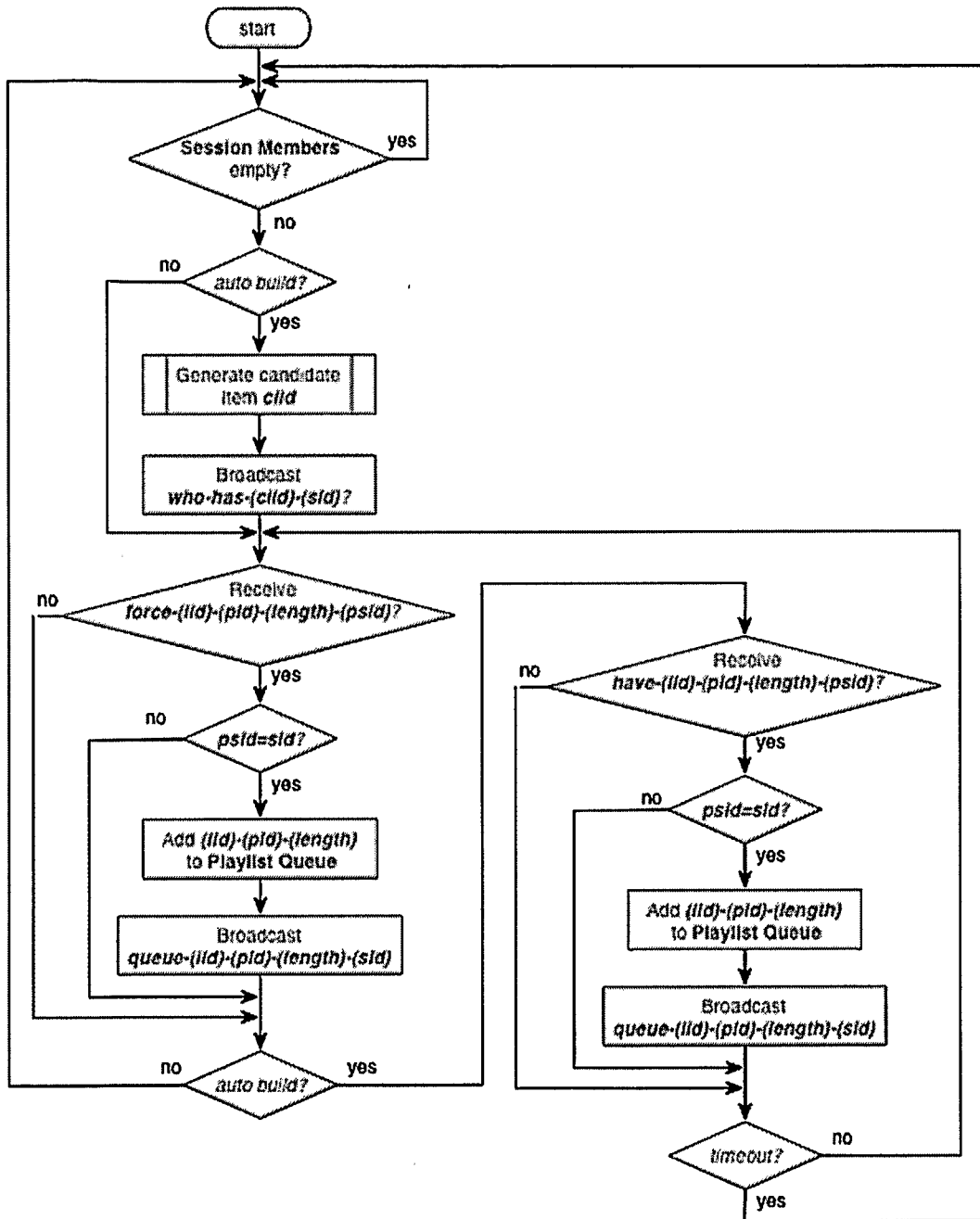


FIG. 16



Playlist Builder Flow Diagram

FIG. 17

Composite Playlist Builder

Step 1: Generate list of media items that define user taste. Repeat for all users in system..

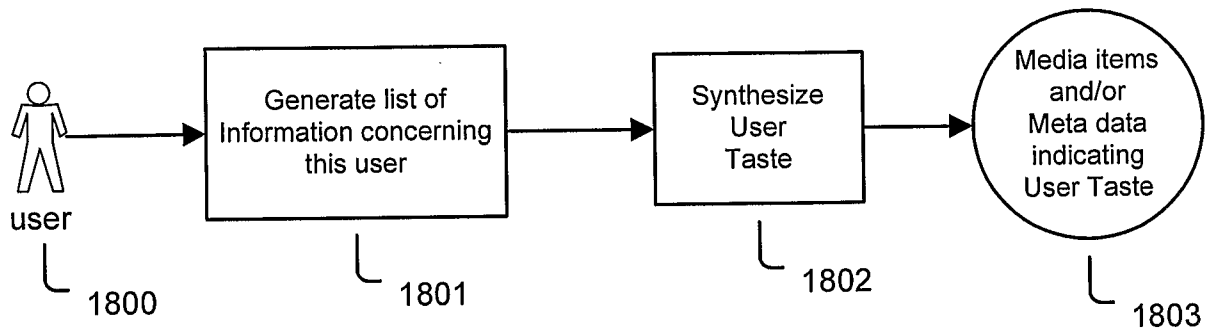


FIG. 18

Composite Playlist Builder

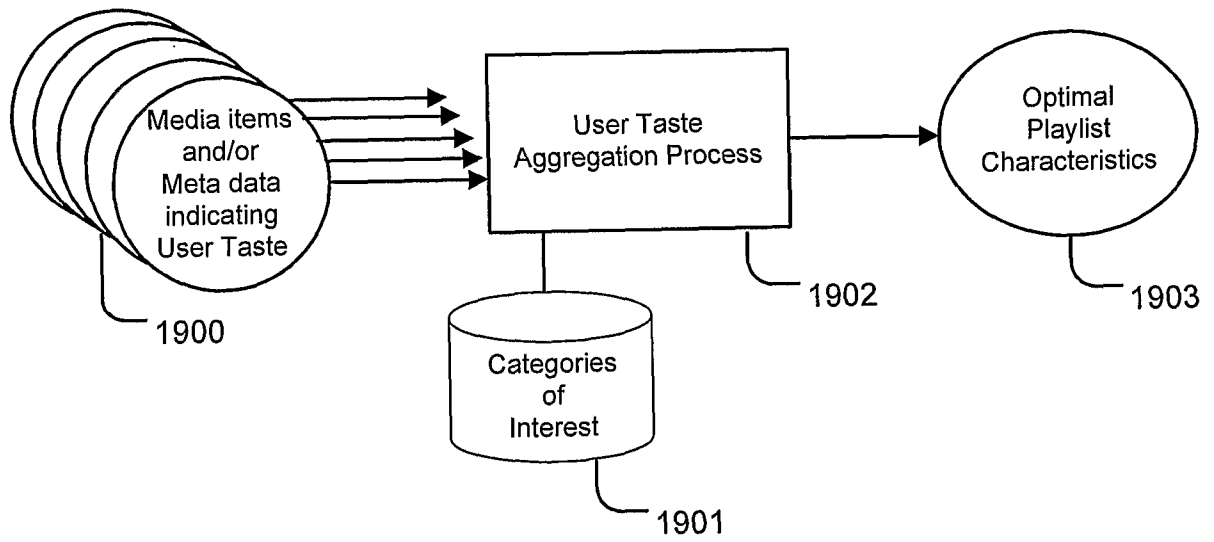


FIG. 19

Composite Playlist Builder

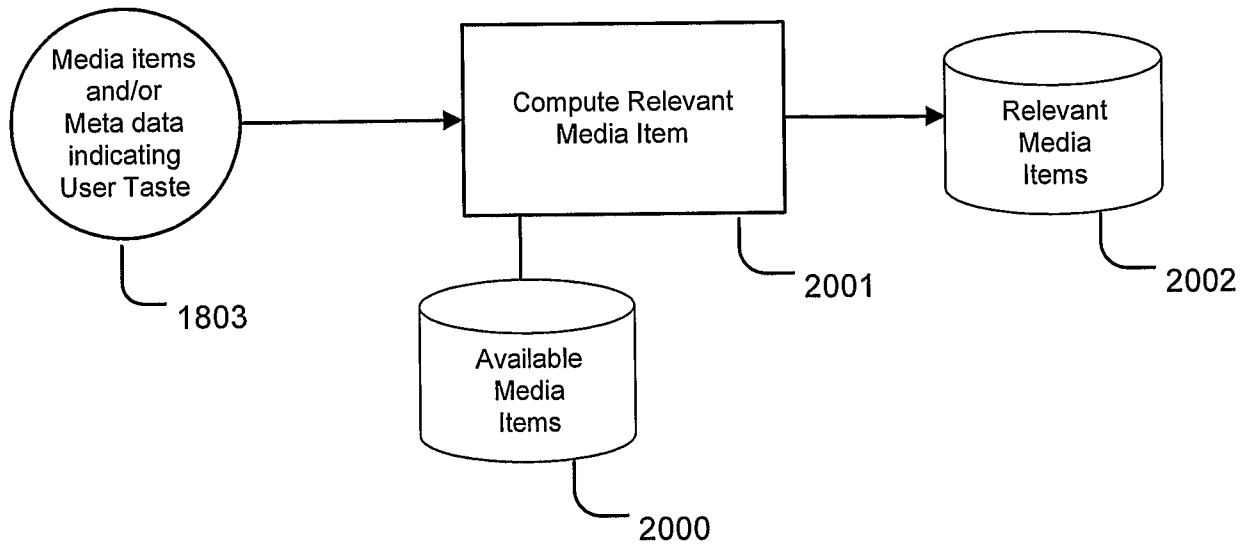


FIG. 20

Composite Playlist Builder

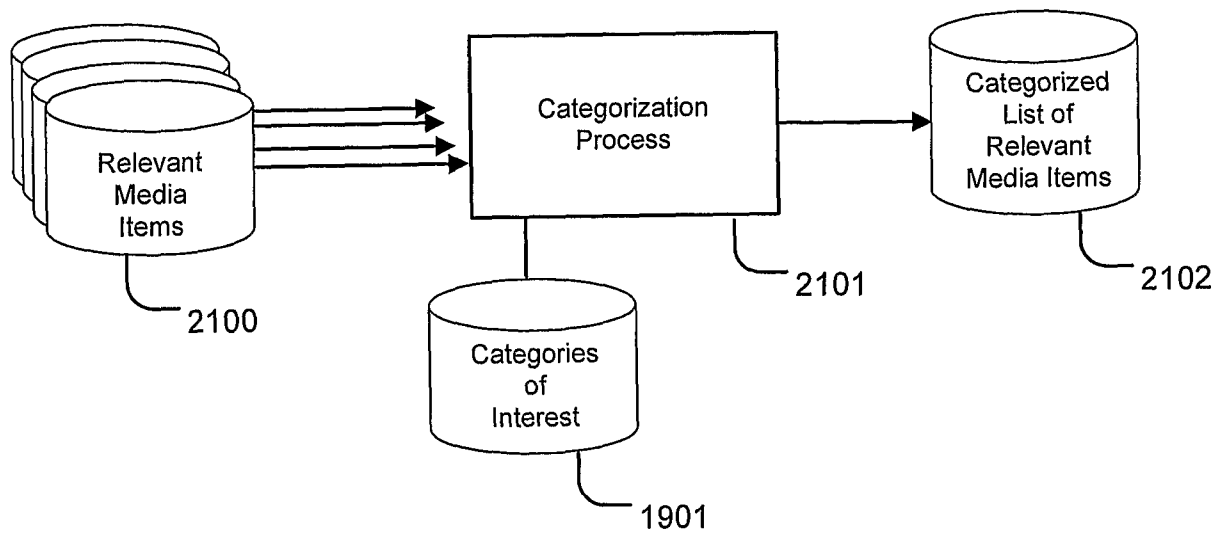


FIG. 21

Composite Playlist Builder

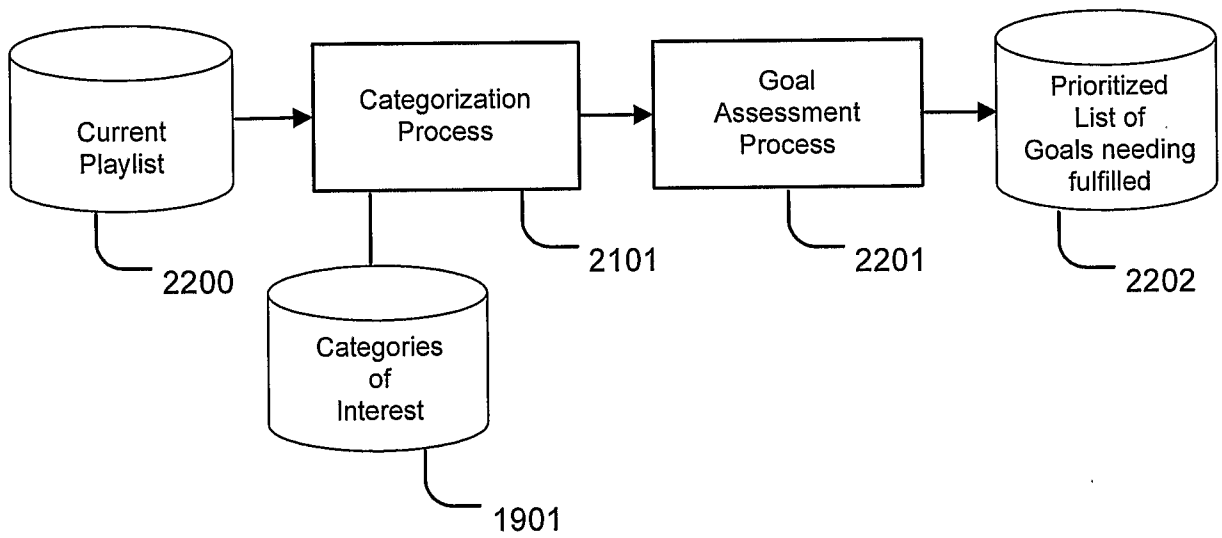


FIG. 22

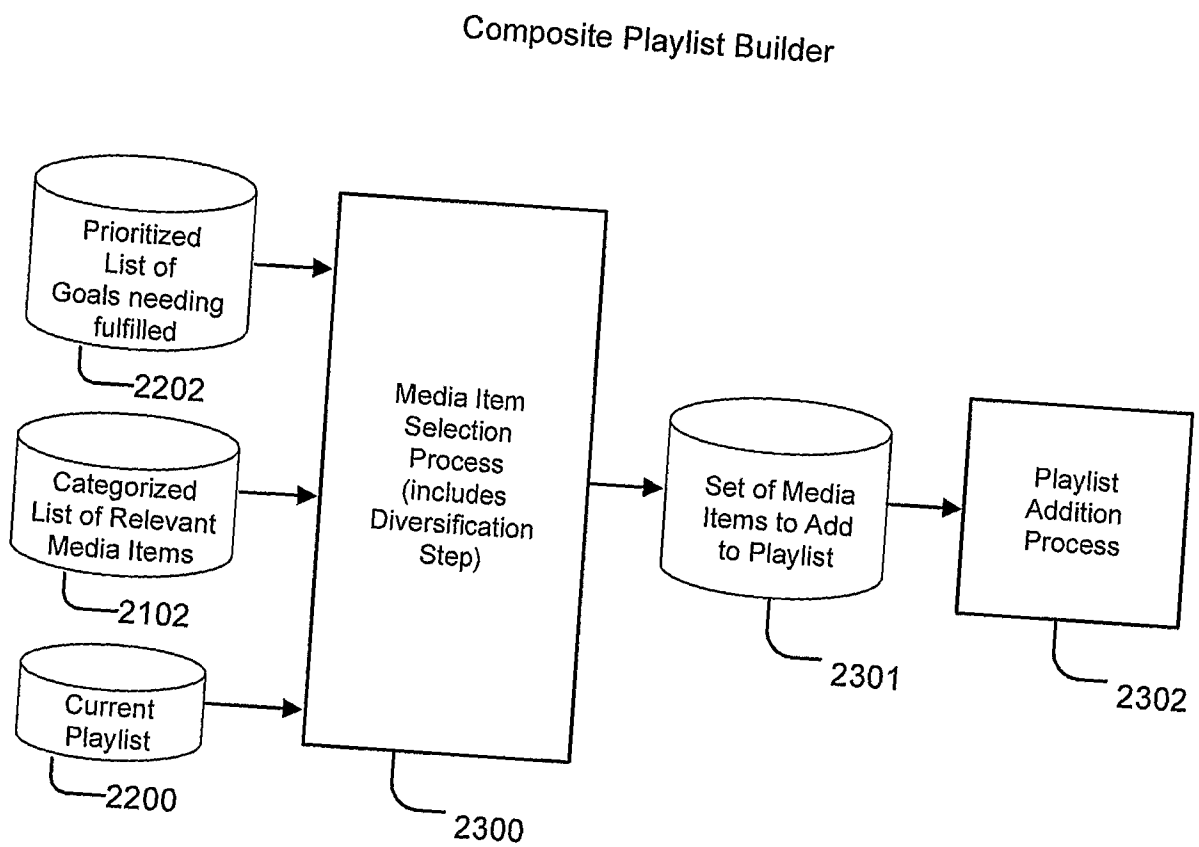


FIG. 23

User Addition Process

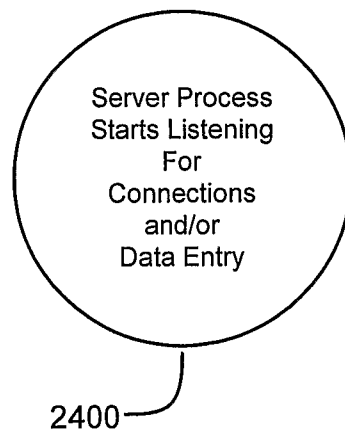


FIG. 24

User Addition Process

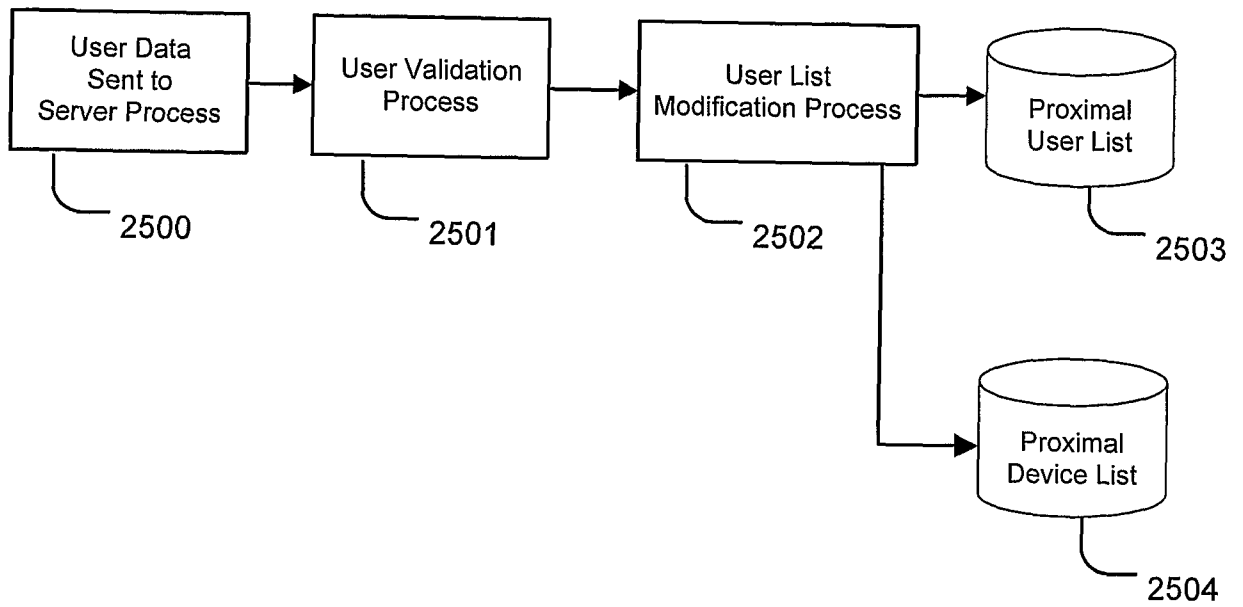


FIG. 25

User Removal Process

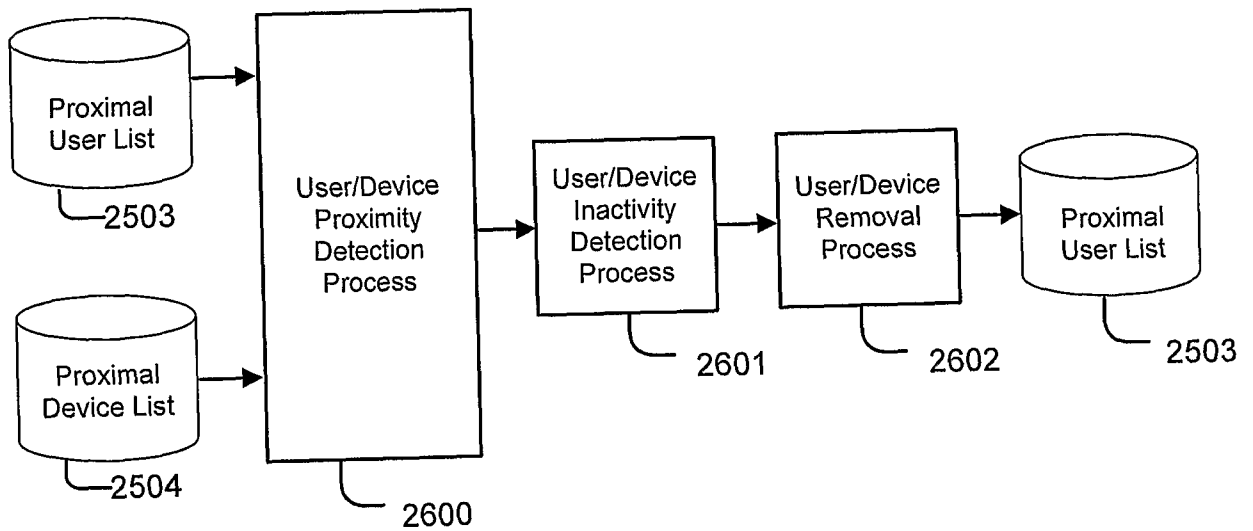


FIG. 26

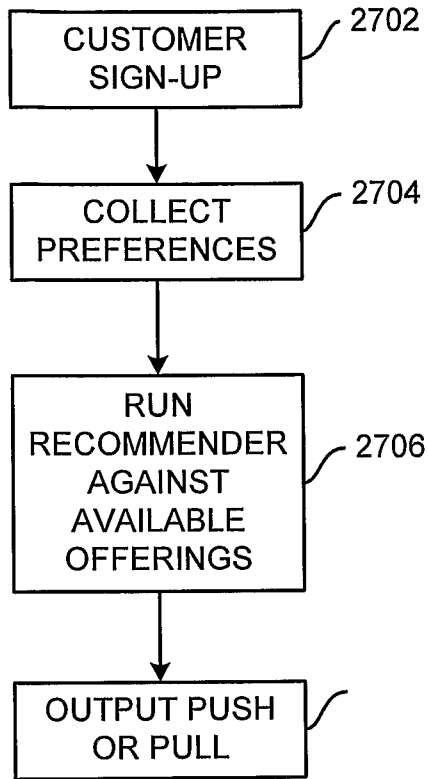


FIG. 27

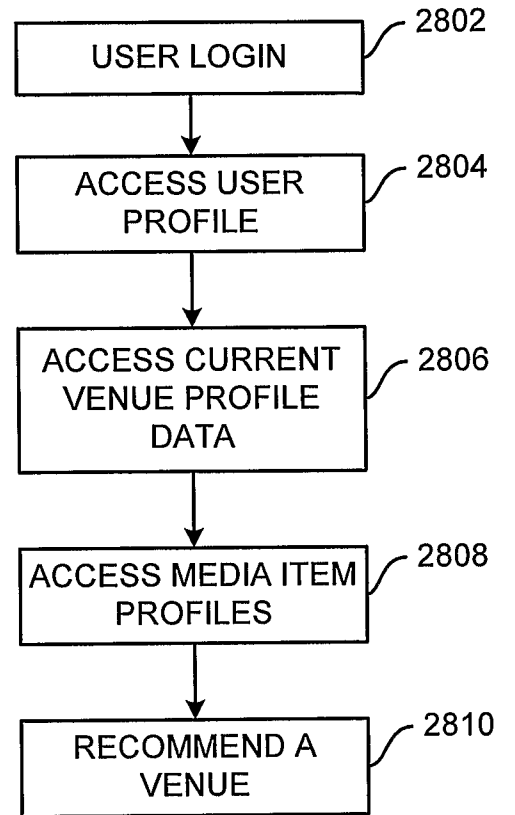


FIG. 28

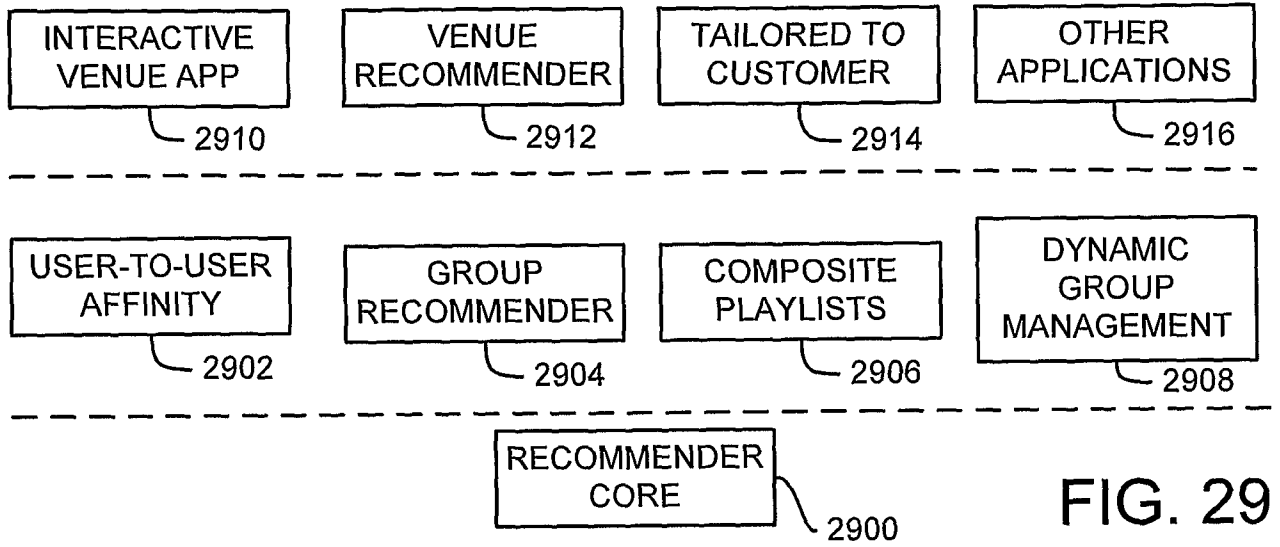


FIG. 29

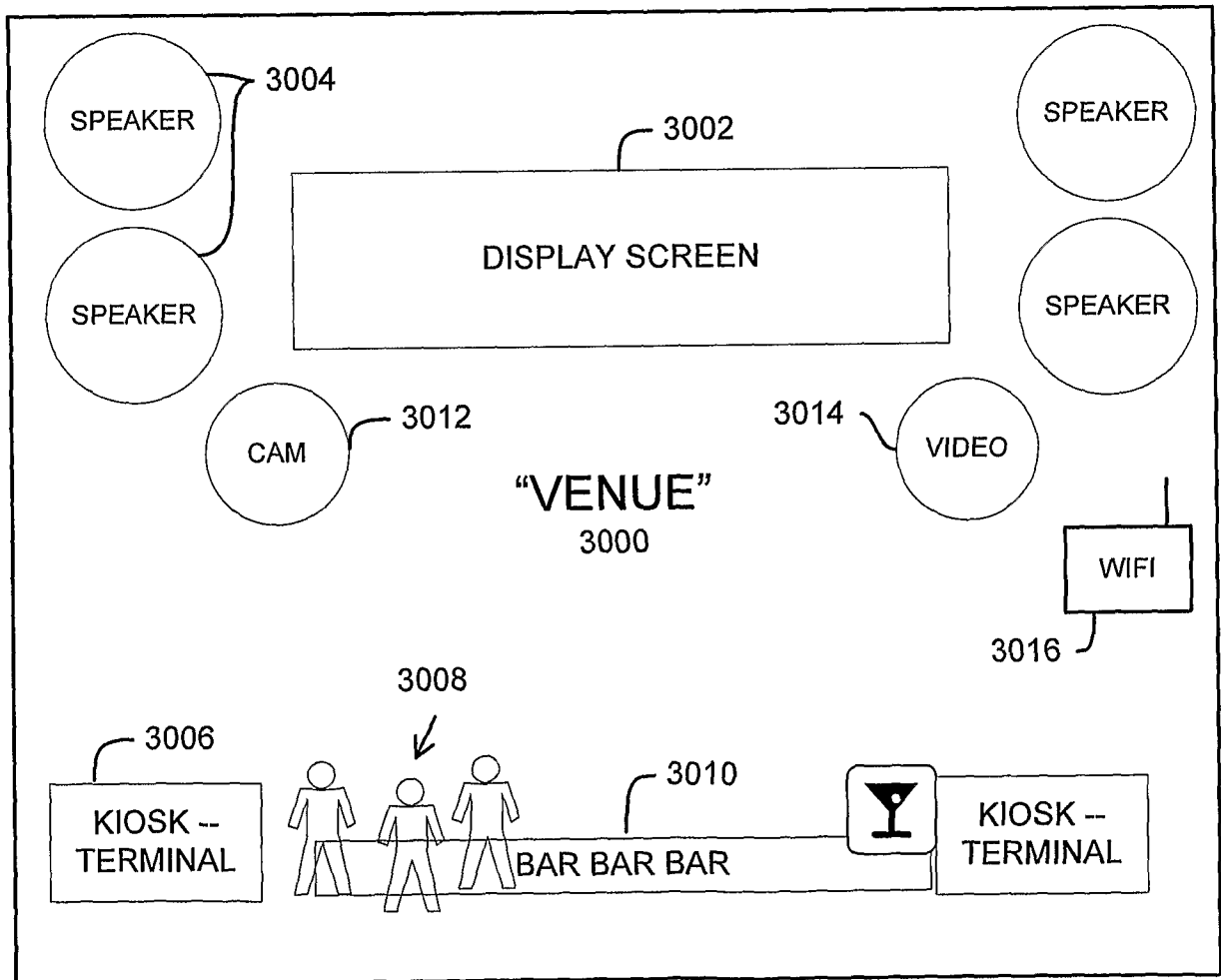


FIG. 30



FIG. 31

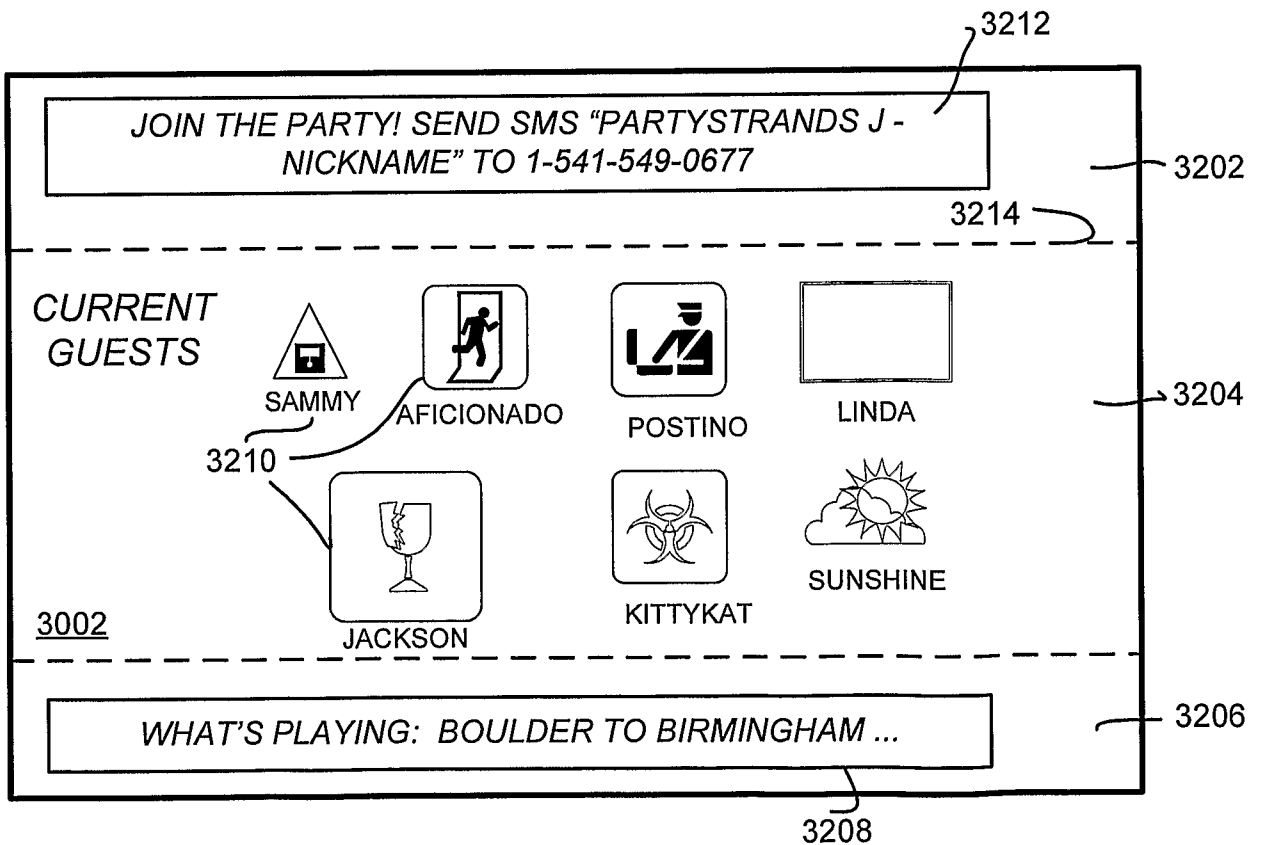


FIG. 32

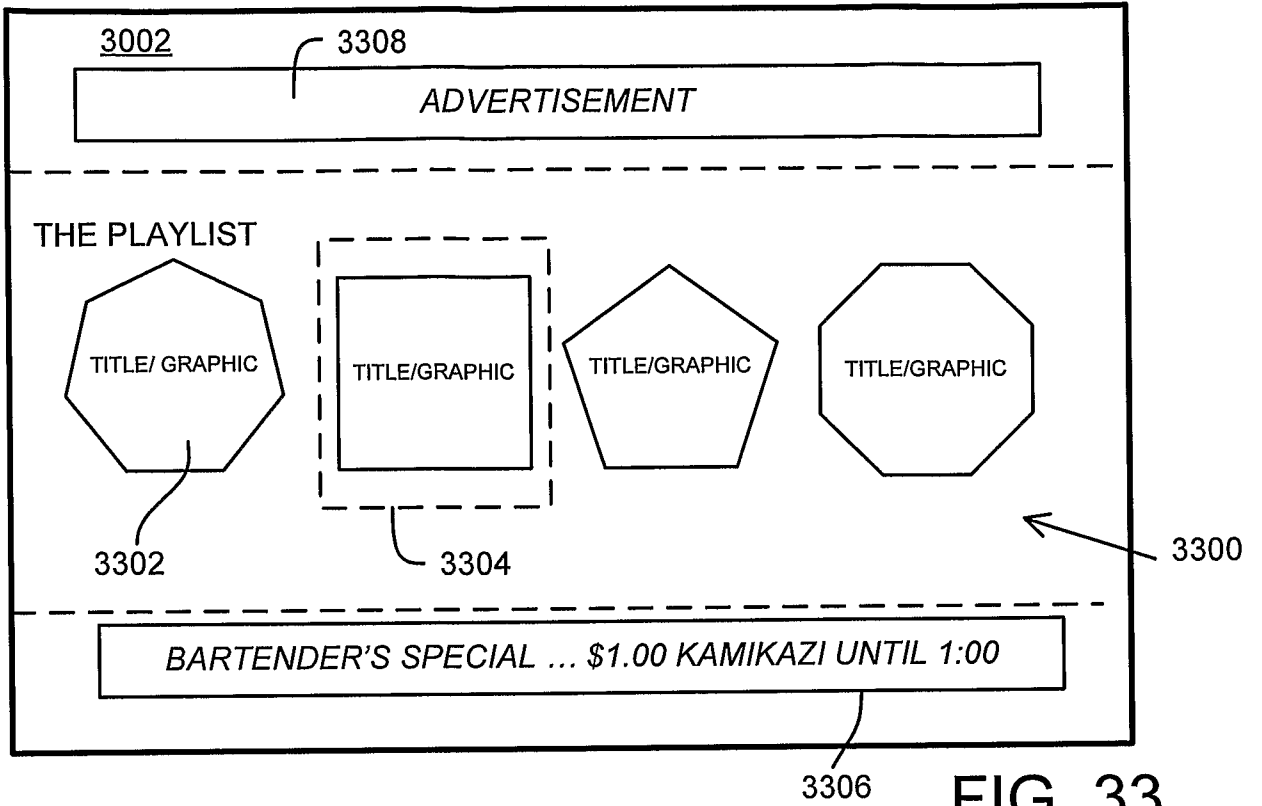


FIG. 33

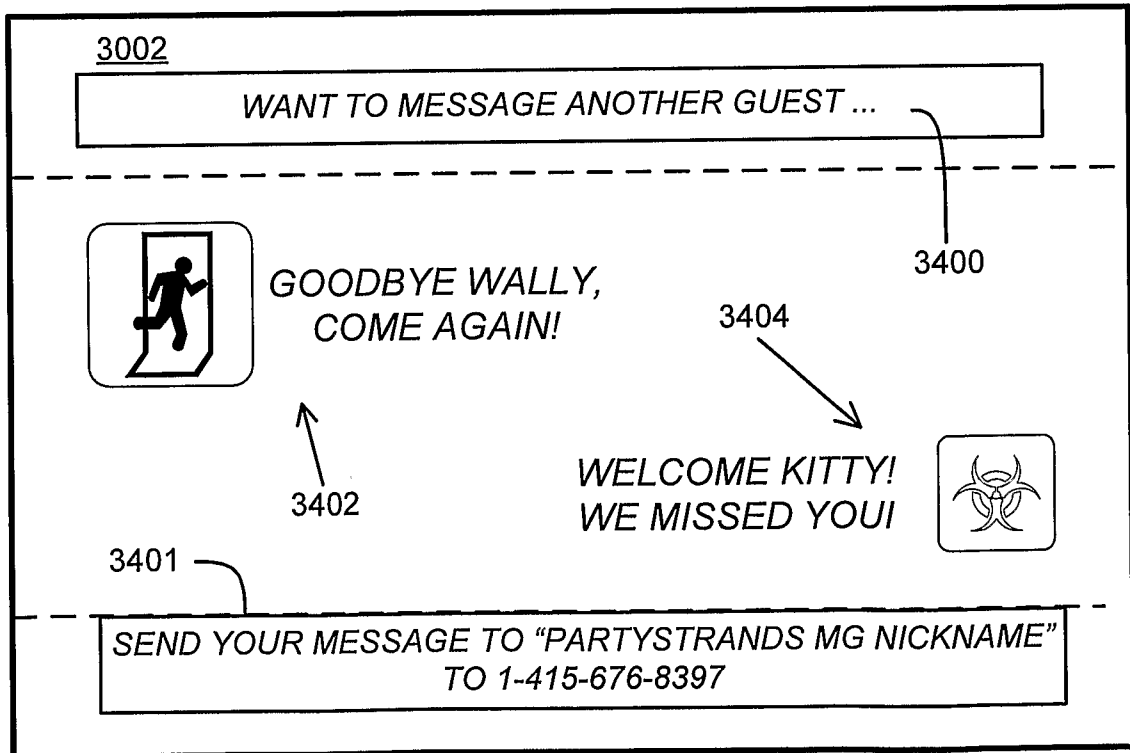


FIG. 34

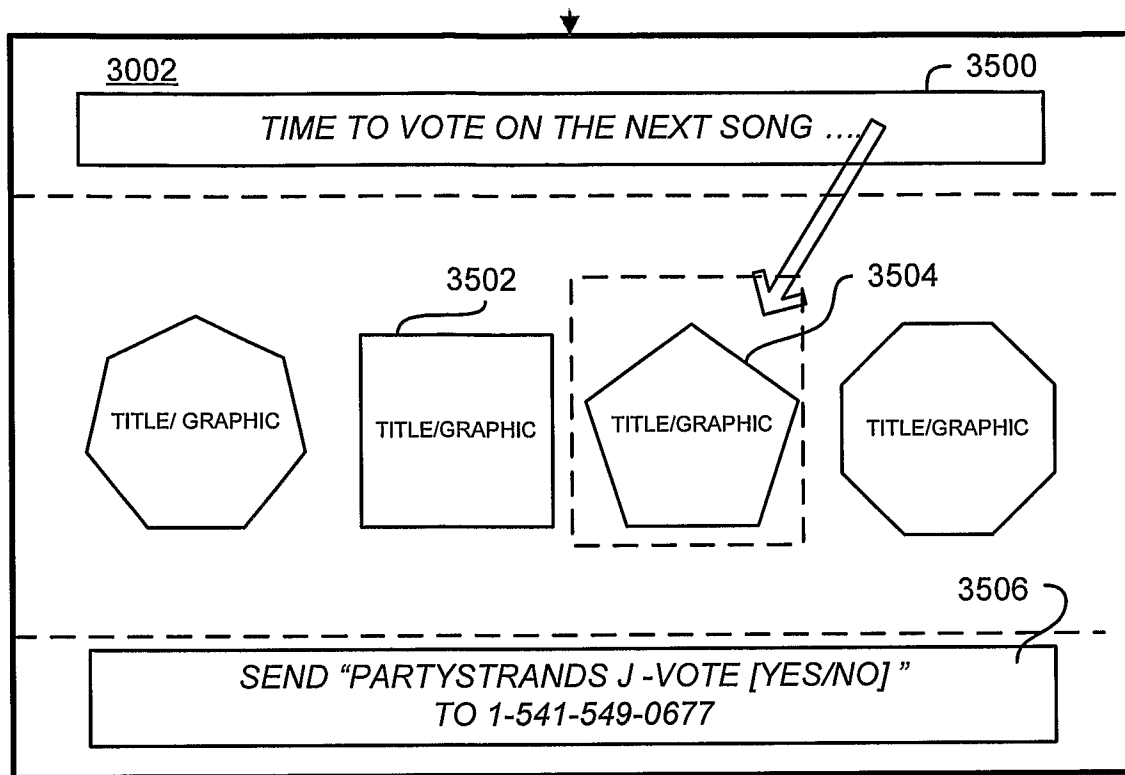


FIG. 35

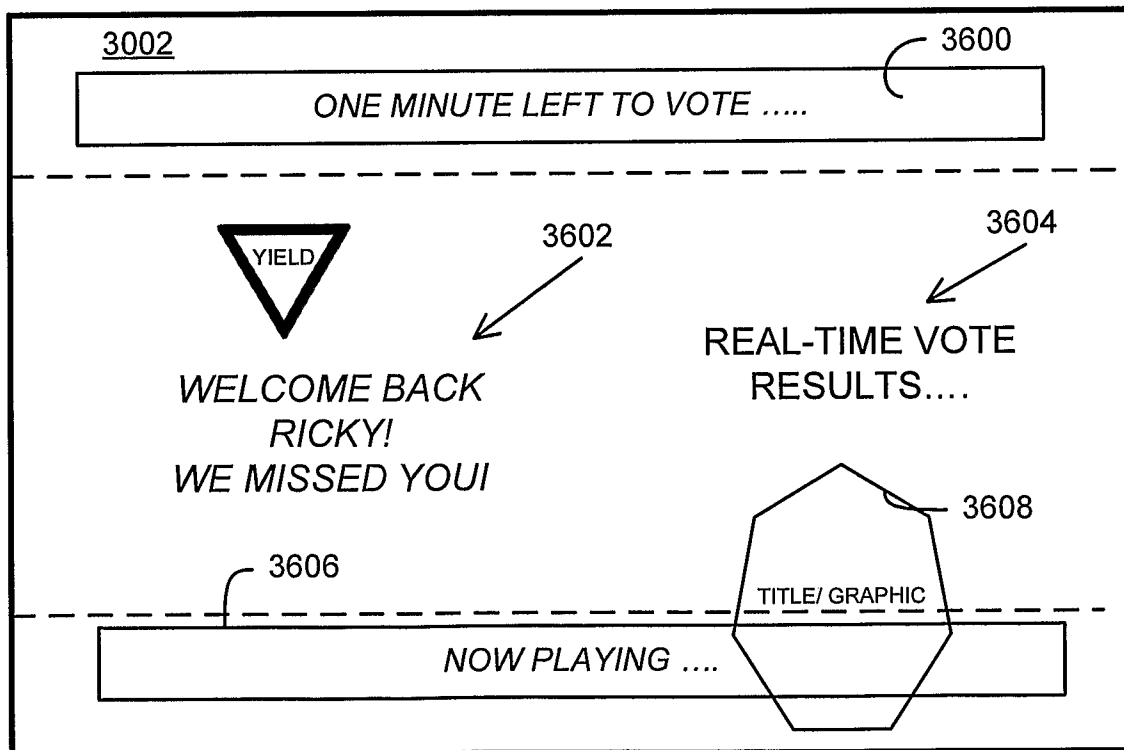


FIG. 36

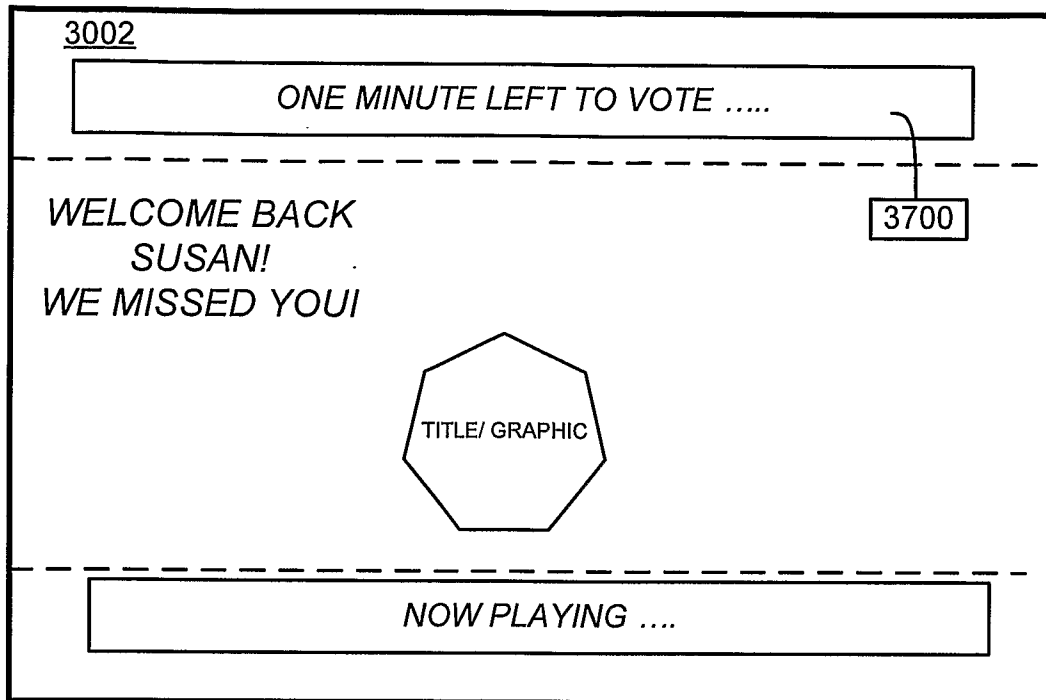


FIG. 37

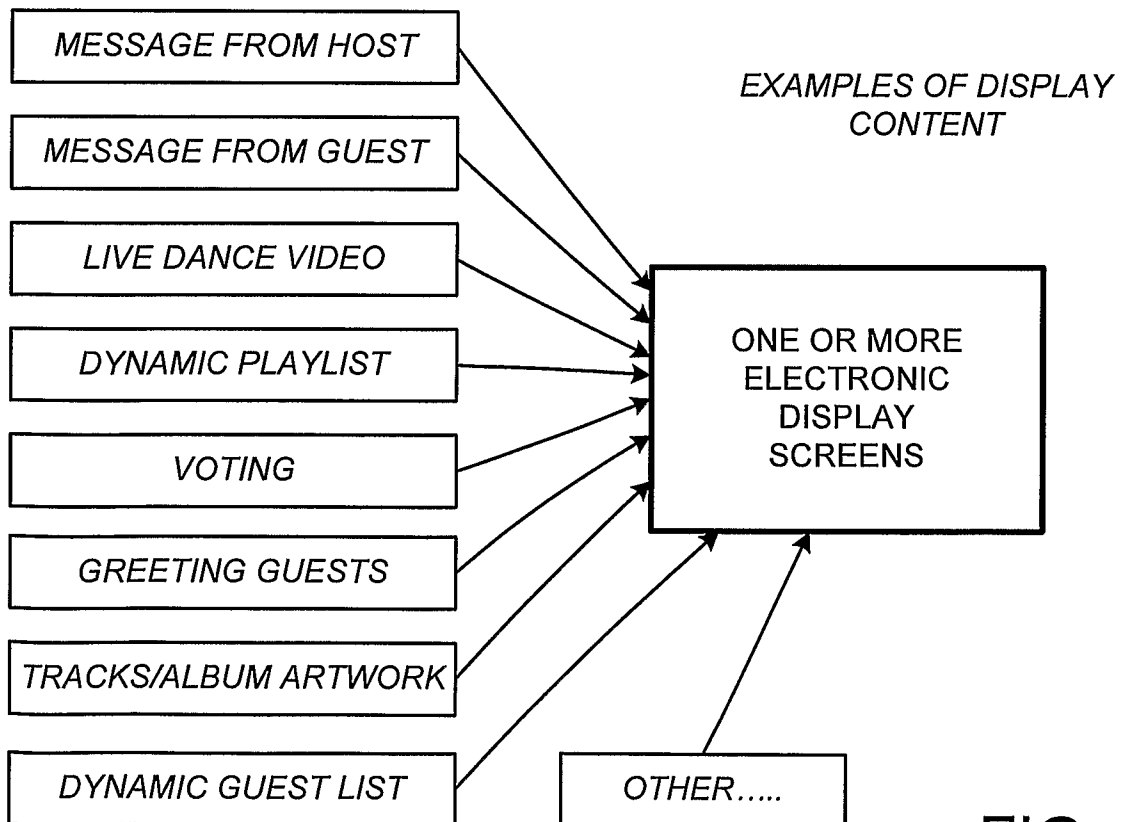
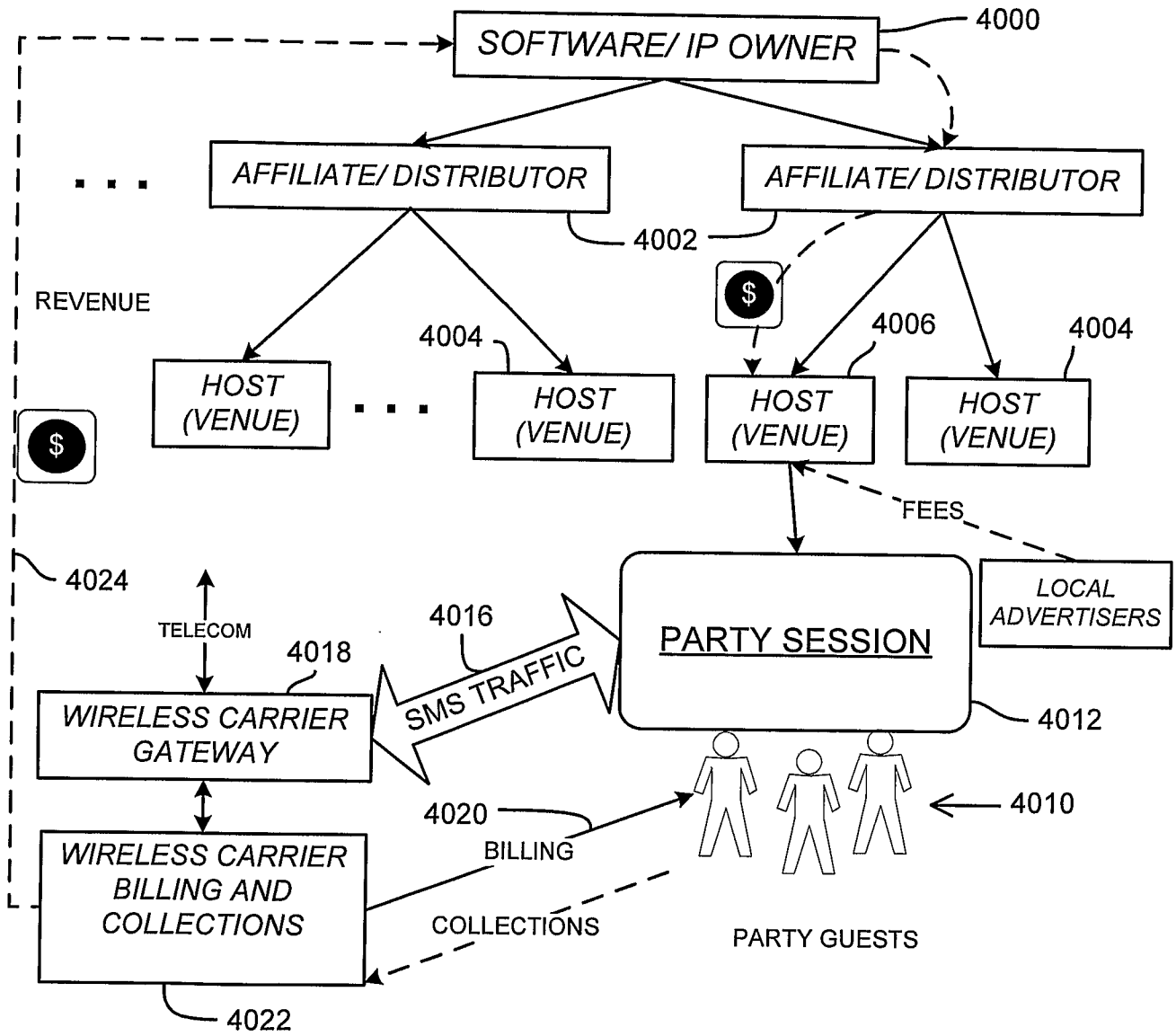


FIG. 38



DISTRIBUTION, LICENSING AND REVENUE SHARING EXAMPLE

FIG. 40

4100

Media Items

	m_1	m_2	m_3		m_k
m_1	-	.1	-		.6
m_2	.7	-	.4		.1
m_3	.5	.1	-		-
m_k	.1	-	.3		-

4102

Media Items

4104

4106

FIG. 41A

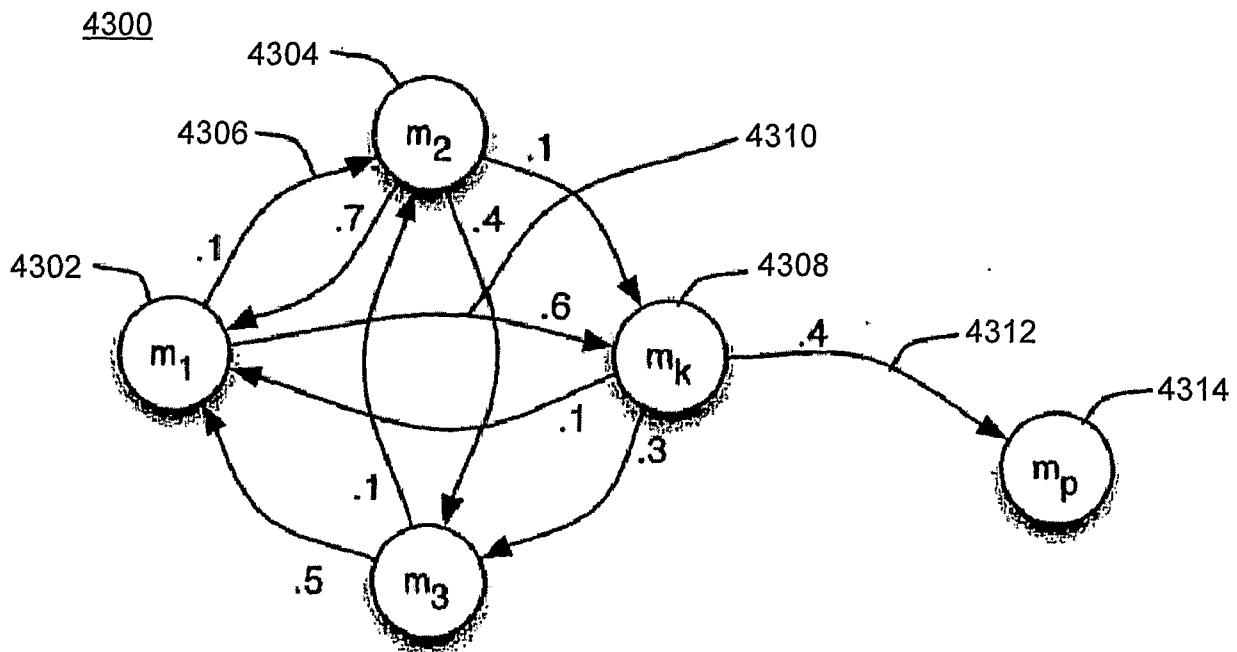


FIG. 41B

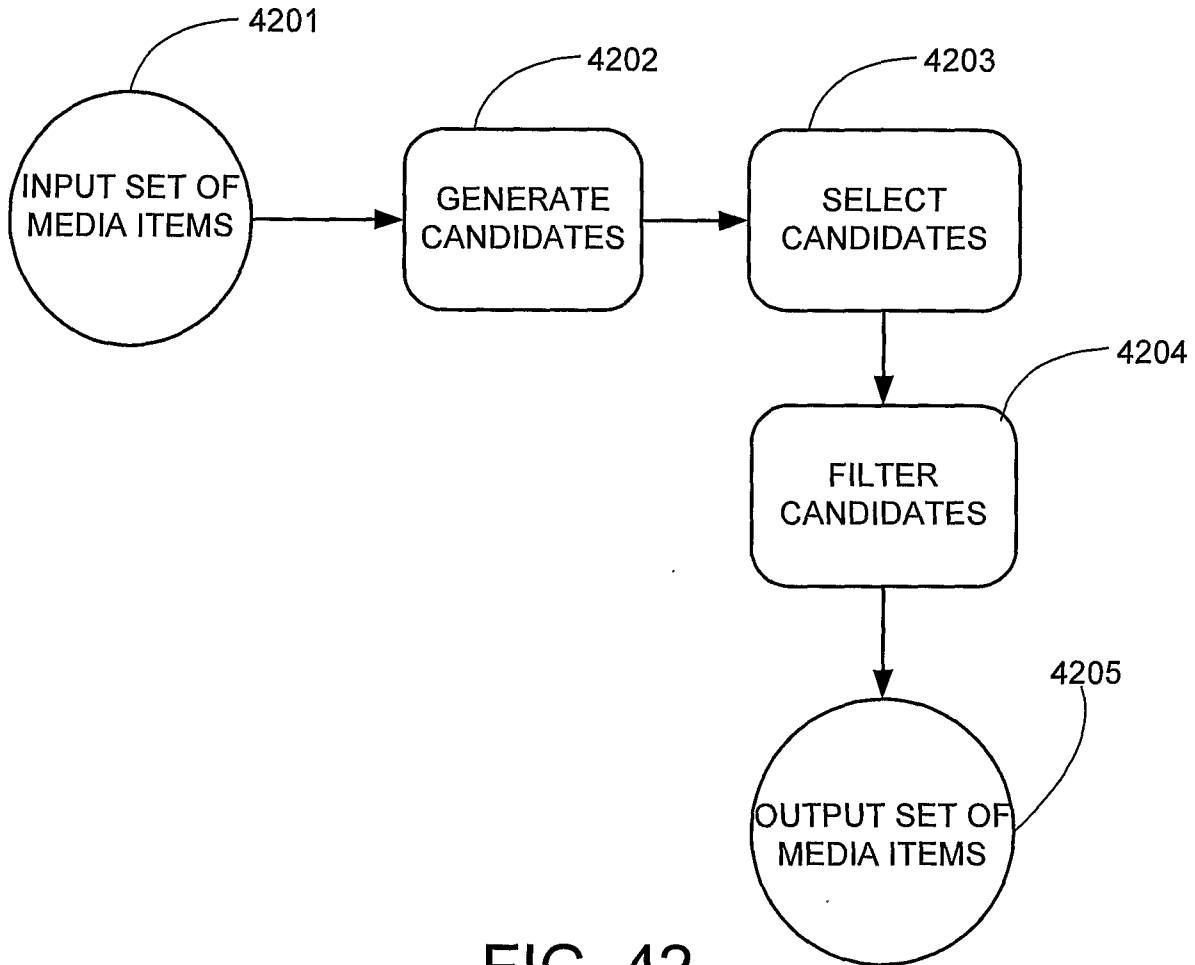


FIG. 42

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2006/034218

A. CLASSIFICATION OF SUBJECT MATTER
INV. 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)
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)

EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2004/073924 A1 (PENDAKUR RAMESH [US]) 15 April 2004 (2004-04-15) abstract paragraph [0017] - paragraph [0018] paragraph [0040] - paragraph [0044] paragraph [0047] - paragraph [0050] paragraph [0057] - paragraph [0058] claims 1-16	1-33
X	EP 1 231 788 A (KONINKL PHILIPS ELECTRONICS NV [NL]) 14 August 2002 (2002-08-14) abstract paragraph [0018] - paragraph [0050]	1-33



Further documents are listed in the continuation of Box C.



See patent family 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.
- *Z* document member of the same patent family

Date of the actual completion of the international search

30 January 2007

Date of mailing of the international search report

09/02/2007

Name and mailing address of the ISA/

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

Authorized officer

Konak, Eyüp

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No PCT/US2006/034218

Patent document cited in search report		Publication date		Patent family member(s)	Publication date
US 2004073924	A1	15-04-2004		NONE	
EP 1231788	A	14-08-2002		CN 1461559 A	10-12-2003
				WO 02065774 A1	22-08-2002
				JP 2004519046 T	24-06-2004
				US 2002124050 A1	05-09-2002