US 20130298046A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2013/0298046 A1

(43) Pub. Date: Nov. 7, 2013

Taylor et al.

(54) **CONTACT LIST DISPLAY SYSTEM AND METHOD**

(71) Applicant: **eBuddy Holding B.V.**, Amsterdam (NL)

(72) Inventors: **Paulo Taylor**, Amsterdam (NL); **Jan-Joost C. Rueb**, Amsterdam (NL); **Onno Bakker**, Haarlem (NL)

(21) Appl. No.: **13/941,354**

(22) Filed: **Jul. 12, 2013**

Related U.S. Application Data

(63) Continuation of application No. 12/774,700, filed on May 5, 2010, now Pat. No. 8,510,395, which is a continuation of application No. 11/637,316, filed on Dec. 11, 2006, now abandoned.

(60) Provisional application No. 60/748,988, filed on Dec. 9, 2005.

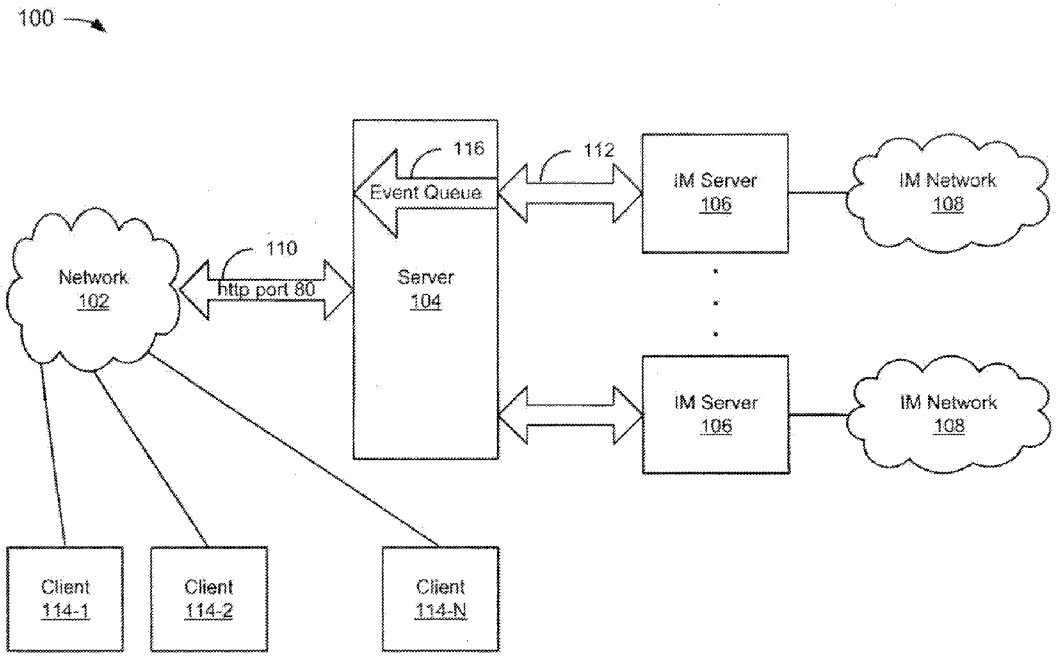
Publication Classification

(51) **Int. Cl.**
H04L 29/06 (2006.01)

(52) **U.S. Cl.**
CPC *H04L 65/403* (2013.01)
USPC **715/753**; 709/204

(57) **ABSTRACT**

A technique for contact list aggregation across networks involves logging into low level networks through a high level network. A system constructed according to the technique may include a network interface coupled to the different low level networks. The system may further include a contact aggregation engine coupled to the network interface and a network contacts database. In operation the system logs into one or more of the low level networks (or facilitates login for a user). To the extent that the data in the network contacts database is not current, the contact aggregation engine updates the networks contacts database contact information, then provides an aggregated contact list including the contact information to a display device. A method according to the technique may include logging into a high level network and displaying contacts from the one or more low level networks in an aggregated contact list. The method may further include logging into the one or more low level networks.



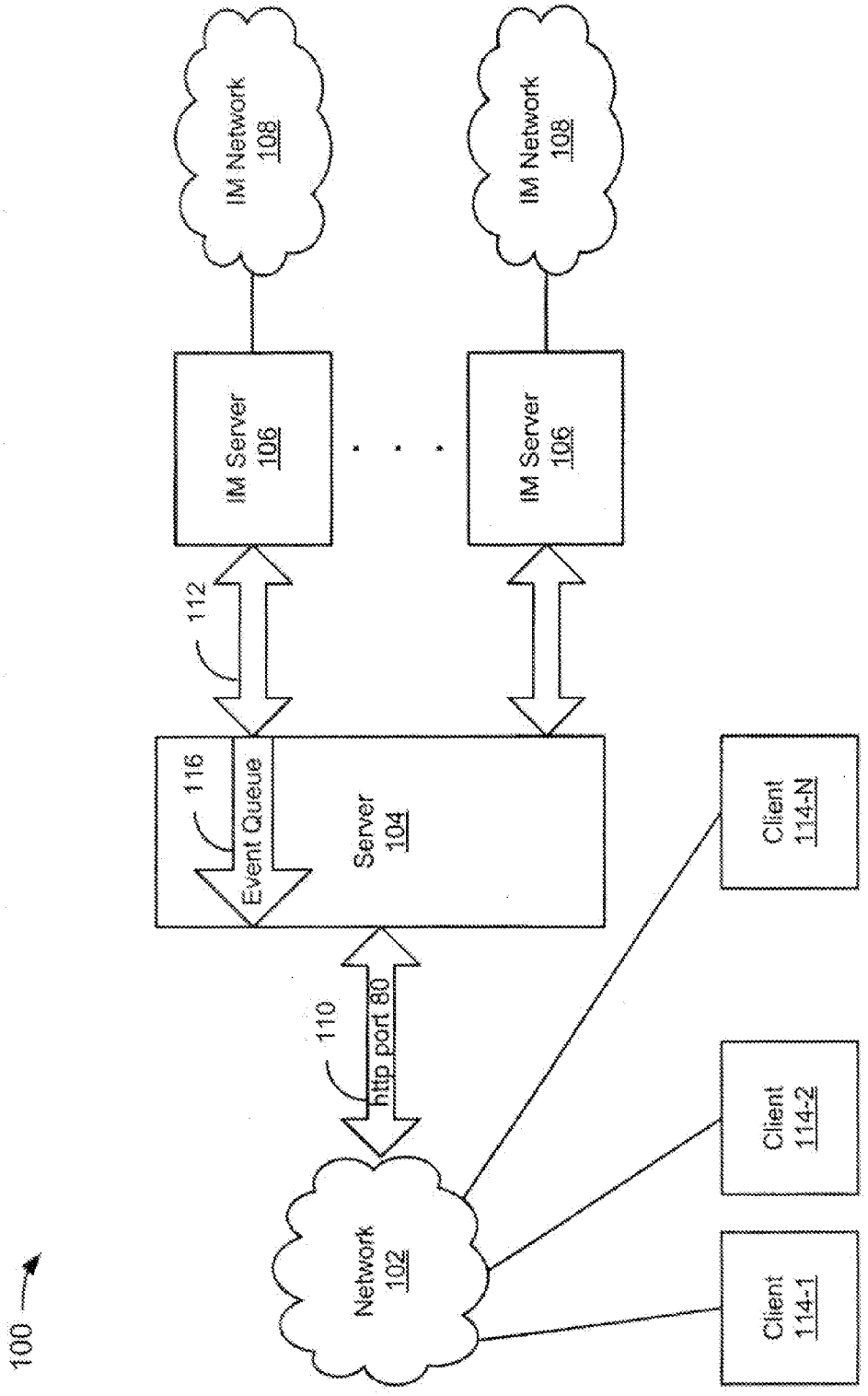

100
Network 102
Client 114-1
Client 114-2
Client 114-N
110
http port 80
Server 104
116
Event Queue
112
IM Server 106
IM Server 106
IM Network 108
IM Network 108

FIG. 1

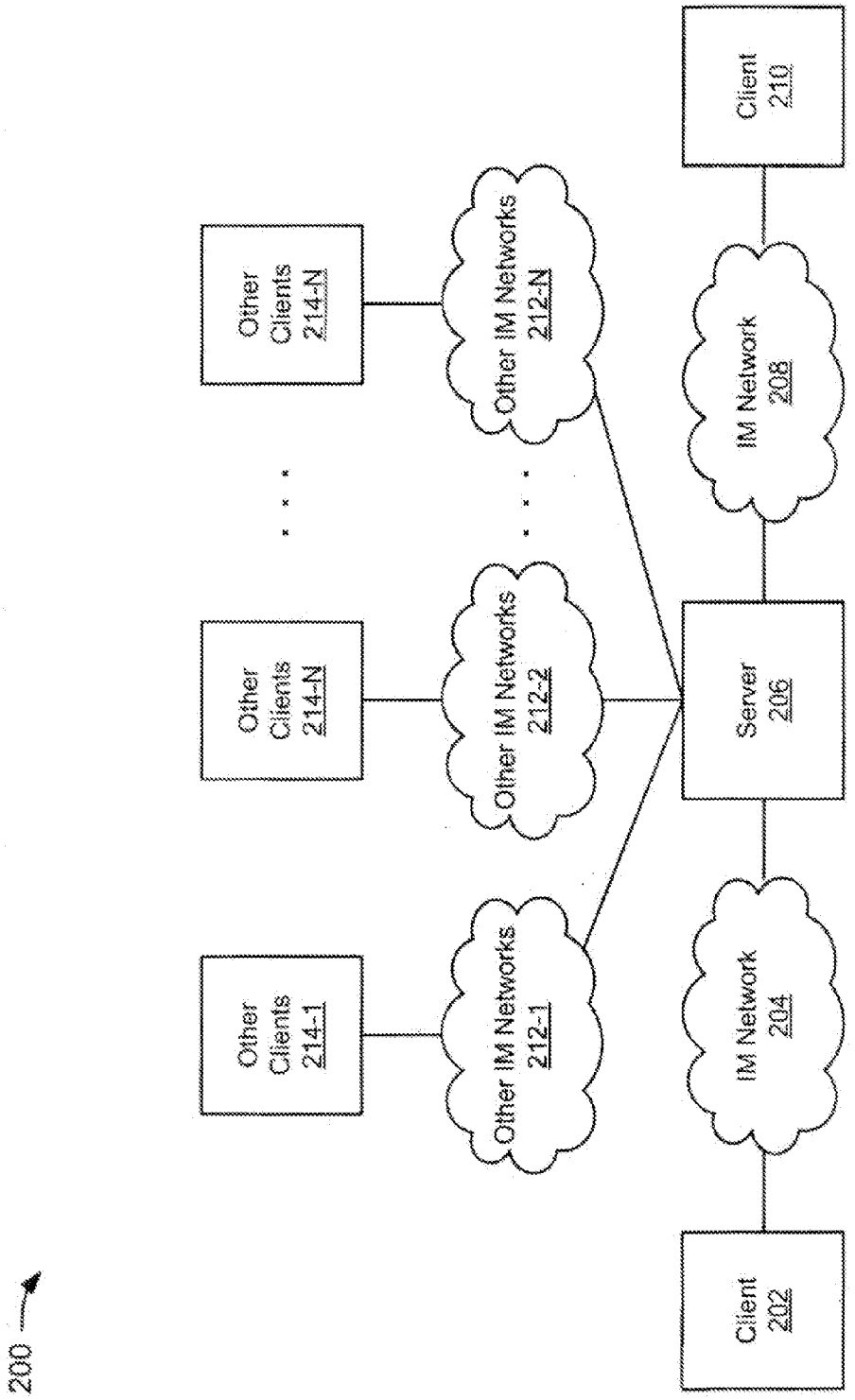

200
Client 202
IM Network 204
Server 206
IM Network 208
Client 210
Other IM Networks 212-1
Other IM Networks 212-2
Other IM Networks 212-N
Other Clients 214-1
Other Clients 214-N
Other Clients 214-N

FIG. 2

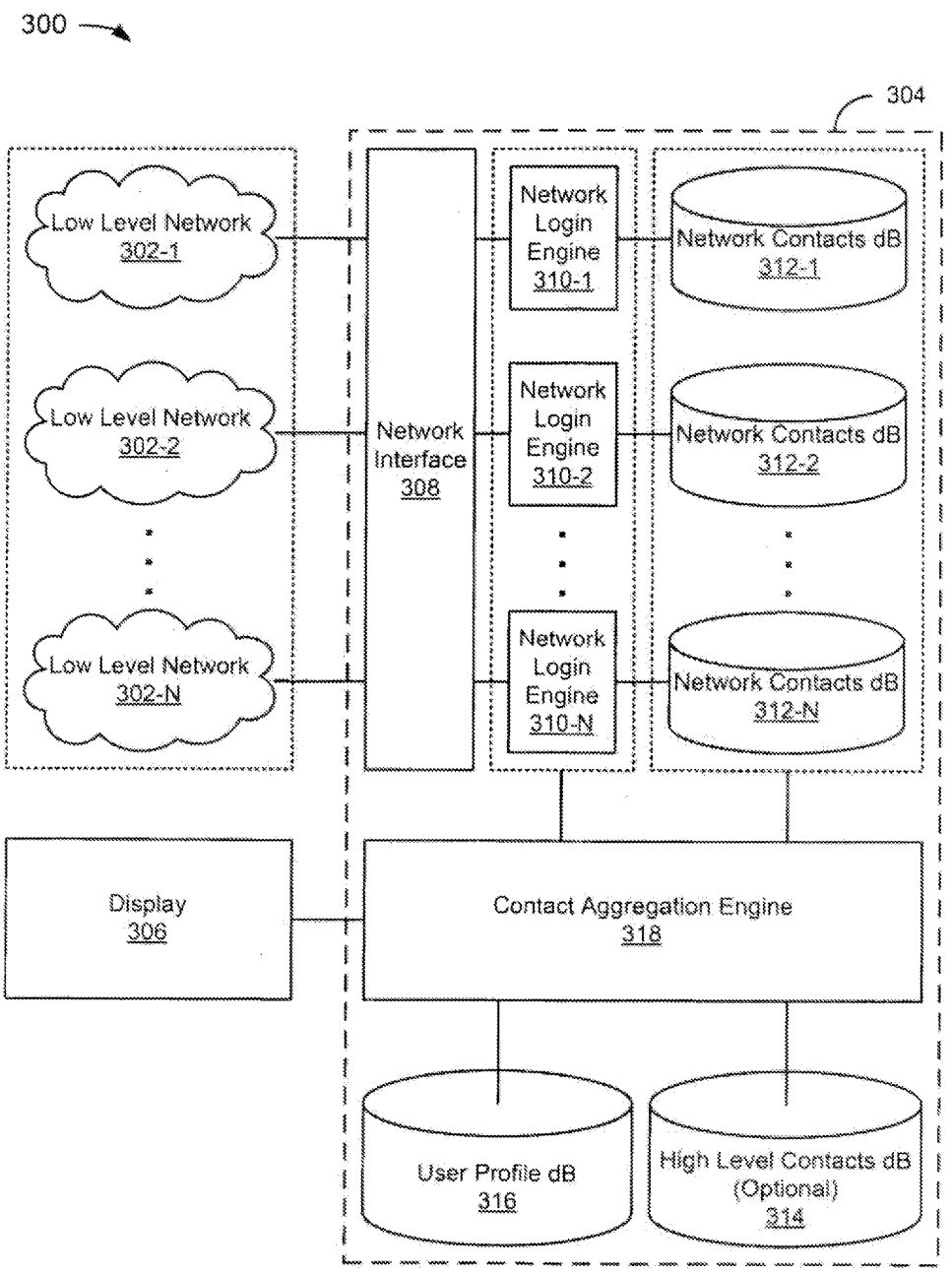

300
304
Low Level Network
302-1
Low Level Network
302-2
Low Level Network
302-N
Network
Interface
308
Network
Login
Engine
310-1
Network
Login
Engine
310-2
Network
Login
Engine
310-N
Network Contacts dB
312-1
Network Contacts dB
312-2
Network Contacts dB
312-N
Display
306
Contact Aggregation Engine
318
User Profile dB
316
High Level Contacts dB
(Optional)
314

FIG. 3

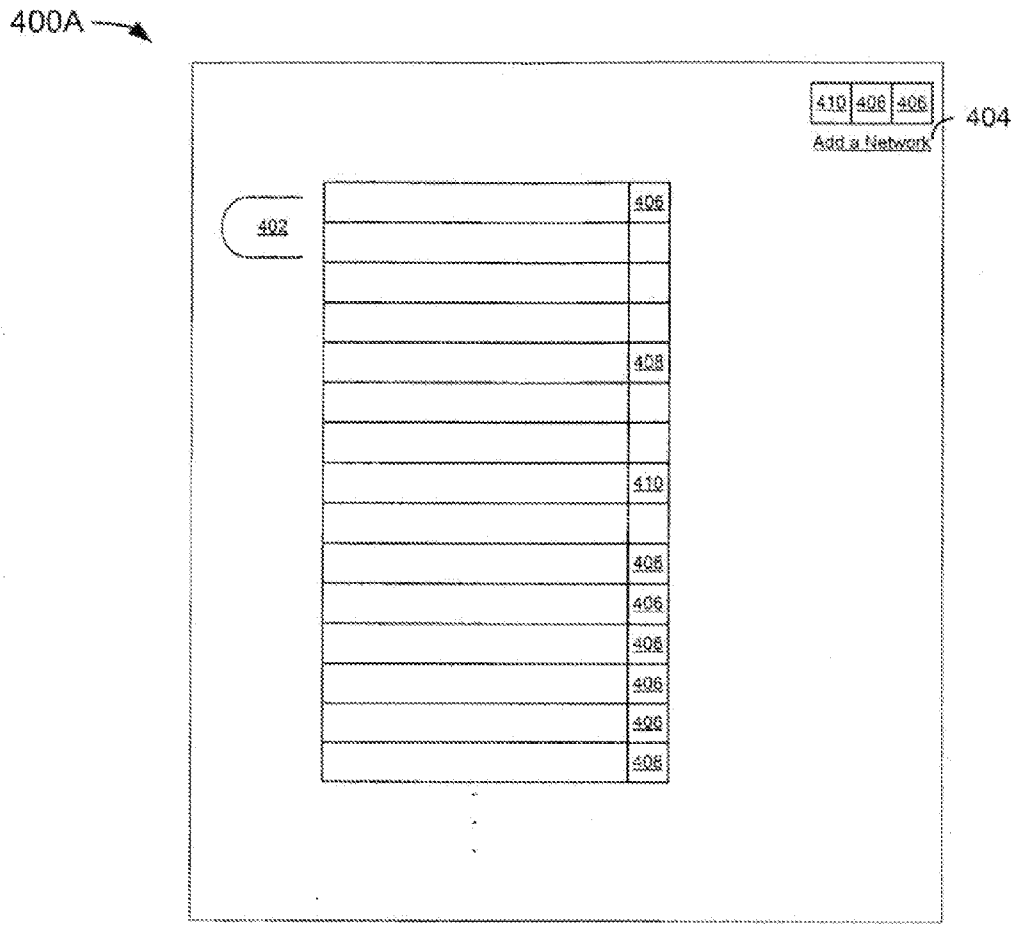

400A
410
406
406
404
Add a Network
402
406
408
410
406
406
406
406
406
406

FIG. 4A

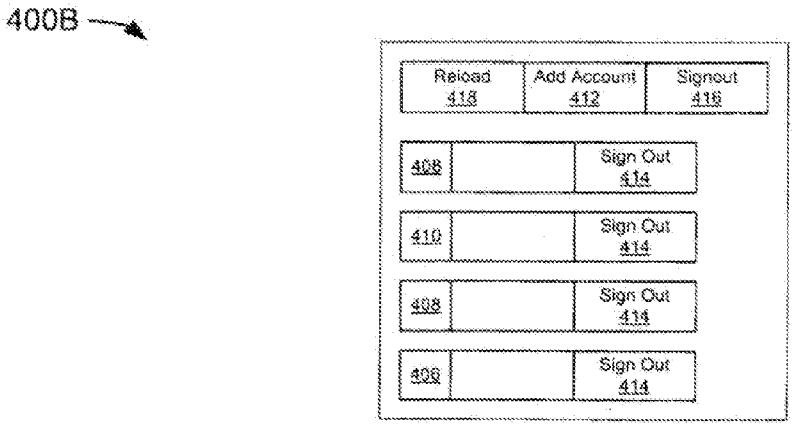

400B
Reload
418
Add Account
412
Signout
416
406
Sign Out
414
410
Sign Out
414
408
Sign Out
414
406
Sign Out
414

FIG. 4B

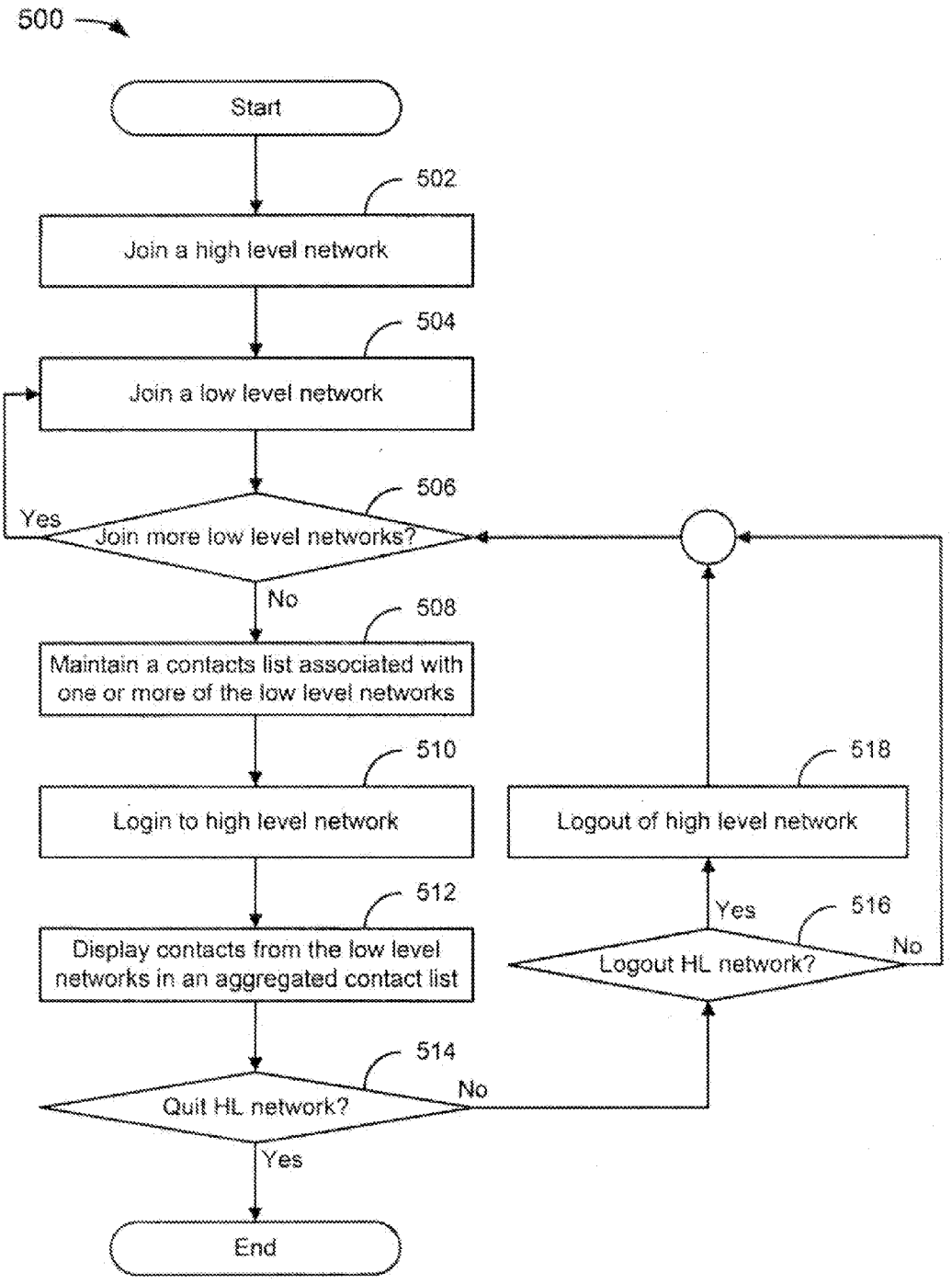

500
Start
502
Join a high level network
504
Join a low level network
506
Yes
Join more low level networks?
No
508
Maintain a contacts list associated with one or more of the low level networks
510
Login to high level network
512
Display contacts from the low level networks in an aggregated contact list
514
Quit HL network?
No
Yes
End
518
Logout of high level network
Yes
516
Logout HL network?
No

FIG. 5

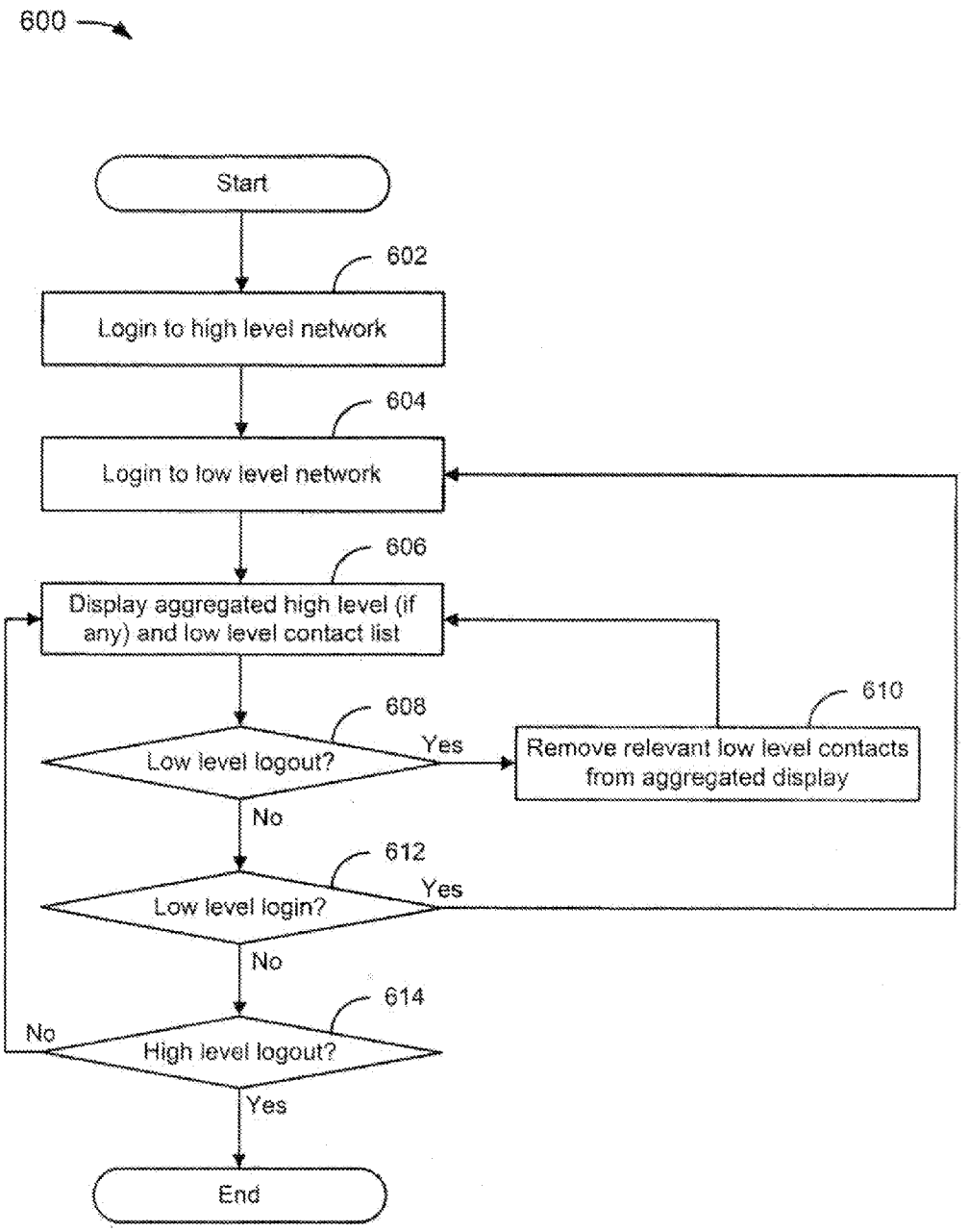

600
Start
602
Login to high level network
604
Login to low level network
606
Display aggregated high level (if any) and low level contact list
608
Low level logout?
Yes
610
Remove relevant low level contacts from aggregated display
No
612
Low level login?
Yes
No
614
High level logout?
No
Yes
End

FIG. 6

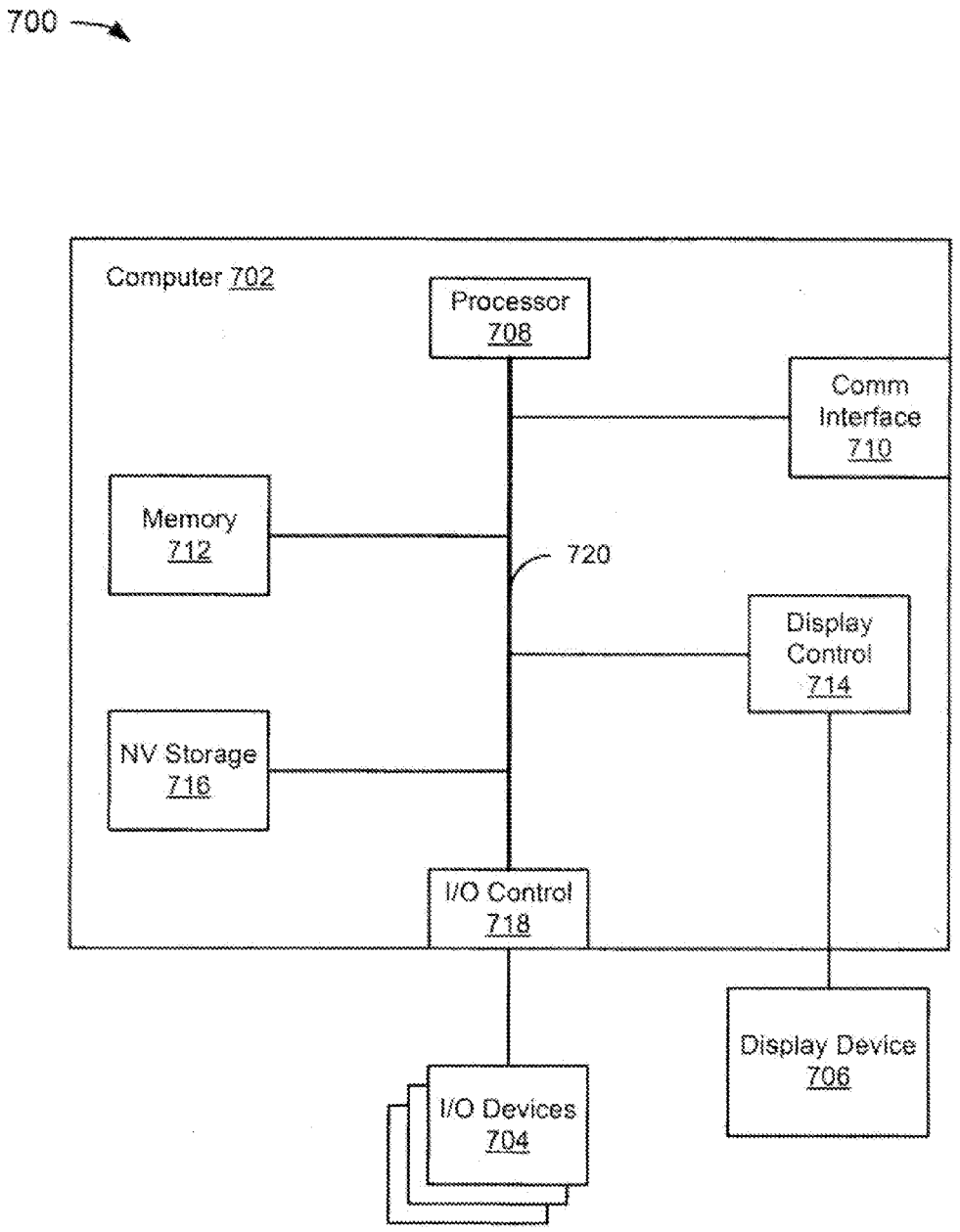

700
Computer 702
Processor 708
Comm Interface 710
Memory 712
720
Display Control 714
NV Storage 716
I/O Control 718
Display Device 706
I/O Devices 704

FIG. 7

CONTACT LIST DISPLAY SYSTEM AND METHOD

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of U.S. Ser. No. 12/774,700, filed May 5, 2010, entitled CONTACT LIST DISPLAY SYSTEM AND METHOD, which is a continuation of U.S. Ser. No. 11/637,316, filed Dec. 11, 2006, entitled CONTACT LIST DISPLAY SYSTEM AND METHOD, which claims priority to U.S. Provisional Patent Application No. 60/748,988, filed Dec. 9, 2005, all of which are incorporated herein by reference.

BACKGROUND

[0002] Instant messaging requires the use of a client program that hooks up an instant messaging service and differs from e-mail in that conversations are then able to happen in real time. Most services offer a presence information feature, indicating whether people on one's list of contacts are currently online and available to chat. This may be called a contact list. In early instant messaging programs, each letter appeared as it was typed, and when letters were deleted to correct typos this was also seen in real time. This made it more like a telephone conversation than exchanging letters. In modern instant messaging programs, the other party in the conversation generally only sees each line of text right after a new line is started. Most instant messaging applications also include the ability to set a status message, roughly analogous to the message on a telephone answering machine.

[0003] Popular instant messaging services on the public Internet include .NET Messenger Service, AOL Instant Messenger, Excite/Pal, Gadu-Gadu, Google Talk, iChat, ICQ, Jabber, Qnext, QQ, Meetro, Skype, Trillian and Yahoo! Messenger. These services owe many ideas to an older (and still popular) online chat medium known as Internet Relay Chat (IRC).

[0004] The foregoing examples of the related art and limitations related therewith are intended to be illustrative and not exclusive. Other limitations of the related art will become apparent to those of skill in the art upon a reading of the specification and a study of the drawings.

SUMMARY

[0005] The following embodiments and aspects thereof are described and illustrated in conjunction with systems, tools, and methods that are meant to be exemplary and illustrative, not limiting in scope. In various embodiments, one or more of the above-described problems have been reduced or eliminated, while other embodiments are directed to other improvements.

[0006] A technique for contact list aggregation across a plurality of different networks involves logging into low level networks through a high level network. A system constructed according to the technique may include a network interface coupled to the different low level networks. The system may further include a contact aggregation engine coupled to the network interface and a network contacts database. In operation the system logs into one or more of the low level networks (or facilitates login for a user). The network contacts database may include some information about contacts associated with the networks from, by way of example but not limitation, previous logins or data explicitly entered by a user. To the extent that the data in the network contacts database is not current, the contact aggregation engine updates the networks contacts database contact information, then provides an aggregated contact list including the contact information to a display device.

[0007] A method according to the technique may include logging into a high level network and displaying contacts from the one or more low level networks in an aggregated contact list. The method may further include logging into the one or more low level networks.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Embodiments of the inventions are illustrated in the figures. However, the embodiments and figures are illustrative rather than limiting; they provide examples of the invention.

[0009] FIG. **1** depicts an example of a system for providing instant messages to clients via a web interface.

[0010] FIG. **2** depicts an example of a system for displaying content from an IM client at an alternative IM client.

[0011] FIG. **3** depicts an example of a system capable of contact aggregation and display.

[0012] FIGS. **4**A-**4**B depict examples of screenshots that depicts a multi-network IM display.

[0013] FIG. **5** depicts a flowchart of an example of a method for contact list aggregation and display.

[0014] FIG. **6** depicts a flowchart of an example of a method for aggregated contact list display.

[0015] FIG. **7** depicts a computer system suitable for implementation of the techniques described with reference to FIGS. **1-6**.

DETAILED DESCRIPTION

[0016] In the following description, several specific details are presented to provide a thorough understanding of embodiments of the invention. One skilled in the relevant art will recognize, however, that the invention can be practiced without one or more of the specific details, or in combination with other components, etc. In other instances, well-known implementations or operations are not shown or described in detail to avoid obscuring aspects of various embodiments, of the invention.

[0017] FIG. **1** depicts an example of a system **100** for providing instant messages to clients via a web interface. In the example of FIG. **1**, the system **100** includes a network **102**, a server **104**, and an Instant Messenger (IM) server **106**, and an IM network **108**. The server **104** is coupled to the network at least by way of port **80**. The two way communication via port **80** is represented in the example of FIG. **1** as an arrow **110**. The server **104** is coupled to the IM server **106** via one or more other ports. The two way communication via the other ports is represented in the example of FIG. **1** as an arrow **112**. The IM server **106** is coupled to the IM network **108** via any known or convenient mechanism. Indeed, the IM server **106** may be thought of as part of the IM network **108**. The network **102** couples a plurality of clients **114-1** to **114**-N (referred to collectively as clients **114**) to the server **104**. In the example of FIG. **1**, the server **104** includes an event queue **116**.

[0018] The network **102** may include by way of example but not limitation LAN, WAN, VLAN, WLAN, Internet, cellular network, phone network, radio network, or some other known or convenient network. The term "Internet" as used herein refers to a network of networks that uses certain pro-

tocols, such as TCP/IP, and possibly other protocols such as the hypertext transfer protocol (HTTP) for hypertext markup language (HTML) documents that make up the World Wide Web (the web). The physical connections of the Internet and the protocols and communication procedures are well known, but any convenient physical connections or protocols could be used.

[0019] The server **104** may include a multiple servers. Indeed, it may be desirable, depending upon details of a particular implementation, to install several servers to cope with the number of simultaneous users the system **100** supports. It may further be desirable, depending upon details of a particular implementation, for the server **104** to have a high CPU throughput, together with large amounts of RAM, to handle a large number of users. It may further be desirable, depending upon details of a particular implementation, to accomplish resource sharing via thread handling where a pool of threads is shared and used by one or more of the clients **114** for client-server communication and between the server **104** and the IM server **106**.

[0020] The server **104** may include one or more of an application server, database server, web server, banners server, and content server, or any combination thereof. To make the most of the techniques described herein, the server **104** should, though is not required to, include at least one application server. The other servers can have supporting roles in, by way of example but not limitation, serving static content or advertising (e.g., banners), storing usage data, or fulfilling some other known or convenient function.

[0021] The server **104** may act as a proxy server between the clients **114** and the IM server **106**. The server **104** receives communications from the clients **114** on http port **80**, and responds to the clients **114** on http port **80**. Communications from the clients **114** that are bound for the IM network **108**, however, must also come through http port **80** to the server **104**, and are then forwarded to the IM server **106**. In this way, the server **104** acts as a carrier of the data from users to the IM network **108** using a mechanism that controls and manages the data (e.g., text messages, display images, emoticons, audio/video streams, etc.) sent between one of the clients **114** and the server **104**, and vice versa.

[0022] The IM server **106** may be any known or convenient IM server that is compatible with IM. Events, messages, or other appropriate data from the IM server **106** are collected in the event queue **116** of the server **104**. The events may be collected in association with a variety of protocols including by way of example but not limitation port **1863**, port **5050**, port **5222**, port **5190**, etc.

[0023] The IM network **108** may include one or a combination of networks selected from MSN Messenger, Yahoo! Messenger, AIM AOL, ICQ, QQ, Jabber, Google Talk, IRC, or some other known or convenient IM network.

[0024] The clients **114** may include any known or convenient device, including by way of example but not limitation, a Web browser, mobile client, PDA, game console, TV box, native application, etc. The clients poll the server **104** for events. The events can be removed from the event queue **116** and translated into text, JavaScript, XML, or some other known or convenient format that one or more of the clients **114** need or expect in order to process data associated with the event.

[0025] To interact with the IM network **108**, the clients **114** send data to the server **104**. The data, which may include commands, is processed and translated into corresponding data that will be sent to the appropriate IM network. In an embodiment, the appropriate IM network may be determinable based upon the protocol encoded in a message.

[0026] Messages or actions from the clients **114** are collected over network protocols such as, by way of example but not limitation, HTTP or plain socket connections. The messages or actions are transformed to an appropriate protocol format to be sent over a compliant port from the clients **114** to the server **104**, with the IM protocol on the application side. In a non-limiting embodiment, the compliant port is http port **80**. However, any port having similar characteristics to those of a typical port **80** could be used.

[0027] The latest available browsers, as of December 2005, enable the use of a technique called AJAX (Asynchronous JavaScript And XML). With AJAX, appropriately configured clients **114** can execute actions and poll for messages or events using only JavaScript. The method is based on using an XMLHttpRequest object to make HTTP requests to the server **104**. The server **104** may reply with messages taken from the queue of the corresponding session in XML (or another) format that are parsed and displayed according to the message content.

[0028] For clients **114** that include a browser, when accessing the server **104** the browser typically uses hidden HTML frames to update information on visible frames. The visible frames display appropriate information while the hidden frames are reloaded in short periods of time. In each refresh that hits the server **104**, the browser identifies the current messaging session and checks if new events or messages associated with the session are in the event queue **116**. When new information arrives and needs to be displayed in some form, the browser makes use of, for example, JavaScript code to update the visible frames and windows with new messages or events keeping the information up to date in the screen. In this way, automatic refreshing can take place in a hidden frame.

[0029] In another embodiment, certain of the clients **114** with browsers may not make use of refreshes. For example, a form of updating the screen without using a refresh technique is to keep one single HTTP socket request alive for the whole period of a messaging session without actually closing the socket connection. In this example, information is initially loaded and displayed in one single visible frame. While events and messages are being received by the server **104**, JavaScript code can be injected into the HTML document through the same HTTP socket kept alive and managed by the server **104**. For each event or message, the browser can interpret the JavaScript code injected and the corresponding parts of the HTML document and windows will be updated.

[0030] In another embodiment, certain of the clients **114** with browsers may make use of manual refreshes. Some relatively unsophisticated browsers, such as WAP and xHTML browsers often available on mobile phones, do not support hidden frames and/or JavaScript (and others may be configured such that they do not support hidden frames and/or JavaScript). In such cases, the information displayed has to be updated manually by the user. Manual updating enables any mobile phone, PDA, TV Set or any device with a browser to connect to the server **104** and use the messaging platforms made available by the server **104** assuring the communication between the clients **114** and the IM server **106**.

[0031] Message history can be stored by most IM clients on a local computer. For alternative web and mobile-based clients local storage may not be possible. In a non-limiting

embodiment, the server **104**, may have the capability to store message history from IM conversations done via one or more of the clients **114**. The message history can be accessed and searched at any time via the server **104** by one or more of the clients **114**

[0032] FIG. **2** depicts an example of a system **200** for displaying content from an IM client at an alternative IM client. In the example of FIG. **2**, the system **200** includes a client **202**, an IM network **204**, a server **206**, an IM network **208**, a client **210**, other IM networks **212-1** to **212**-N (referred to collectively as other IM networks **212**), and other clients **214-1** to **214**-N (referred to collectively as other clients **214**).

[0033] For illustrative purposes, it is assumed that the client **202** has content that is compatible with the IM network **204**. However, the client **210** is capable of reading content formatted to be compatible with the IM network **208**. Thus, in operation, the server **206** collects content from the client **202** (either through the IM network **204**, as shown in FIG. **2**, or directly from the client **202**, such as is shown by way of example in FIG. **1**). The server **206** then formats the content as appropriate for use on the IM network **208**. Once the content is properly formatted, it can be made available to the client **210** (either through the IM network **208**, as shown in FIG. **2**, or directly to the client **210**, such as is shown by way of example in FIG. **1**). Depending upon the embodiment and/or implementation, the content may also be formatted as appropriate for one or more of the other IM networks **212**, to be made available for one or more of the other clients **214**.

[0034] In an embodiment, the server **206** can save the content in one or many formats. In this way, the client **202** could make content available in a first IM format, the server **206** could convert the content into a second IM format, and the server **206** can save the content in at least the second IM format. Thus, the client **210** could receive the data in the second IM format. The server **206** could easily store the content in the first IM format, as well, and make the content available to other clients coupled to the IM network **204**. In addition, the server **206** could convert the content to other IM formats, such as those formats that are associated with the other IM networks **212**, and save the other IM formats. In this way, the other clients **214** may have access to the content.

[0035] In a non-limiting embodiment, the client **202** may be able to view the client **210** and the clients **214** simultaneously. This is advantageous because IM clients typically cannot view IM clients from other IM networks. Advantageously, since the server **206** is used, the client **202** could even include, for example, a mobile device without a client. (Of course, this could be interpreted to mean that the client **202** is not an IM client at all, though the term client is still used for illustrative purposes because the client **202** is served by the server **206** in a client-server-like fashion.) In a non-limiting embodiment, if the client **202** does not or cannot install client software, the client **202** can use a browser for web-based messaging and display.

[0036] FIG. **3** depicts an example of a system **300** capable of contact aggregation and display. The system **300** includes low level networks **302-1** to **302**-N (collectively referred to as the low level networks **302**), a computer **304**, and a display **306**. The low level networks **302** may include, by way of example but not limitation, various IM networks. It may be noted that the computer **304** and the display **306** may be referred to, in certain implementations, as comprising a computer system.

[0037] The computer **304** includes a network interface **308**, network login engines **310-1** to **310**-N (collectively referred to as network login engines **310**), network contacts databases **312-1** to **312**-N (collectively referred to as the network contacts database **312**), a high level contacts database **314**, a user profile database **316**, and a contact aggregation engine **318**.

[0038] The network interface **308** is coupled to the low level networks **302**. In a typical implementation, the network interface **308** also couples the computer **304** to a network such as the Internet and/or a high level network (not shown).

[0039] The network login engines **310** may include logic and storage that facilitates login to the various low level networks **302**. For example, the network login engine **310-1** may include a user name (and perhaps a password, if the password is not requested from the user at each login) associated with the low level network **302-1**. Conceptually, each of the network login engines **310** is intended to represent the capability of login to the low level networks **302** in a general sense (i.e., the data and logic required for any device to connect to the low level networks **302**) and a user-specific sense (e.g., the data provided from a user that enables login to accounts associated with the user).

[0040] The network contacts database **312**, which is embodied in a computer-readable medium, includes contacts data associated with any of the networks into which a user has logged. When a user logs out, the data may or may not be cached (or stored in nonvolatile memory) for future reference, depending upon the implementation and/or user configuration.

[0041] The high level contacts database **414**, which is embodied in a computer-readable medium, is an optional database that includes contacts associated with a high level network. The high level contacts database is optional for at least two reasons. The first reason is that a system need not provide the ability to maintain high level contacts, requiring that a user maintain only low level contacts. The second reason is that even if the system provides the ability to maintain high level contacts, a user may opt not to maintain any high level contacts, opting instead for low level contacts.

[0042] The user profile database **416**, which is embodied in a computer-readable medium, is intended to represent data associated with a user. The amount of data maintained is implementation-specific.

[0043] The contact aggregation engine **418** is coupled to the databases **412**, **414**, **416**, the network login engines **410**, and the display **406**. In operation, the contact aggregation engine **418** controls the network login engines **410** to login to or facilitate login by a user to the various low level networks **402**. The databases **412**, **414**, **416** are accessed in such a way that a list of contacts stored in the network contacts database **412** is aggregated and displayed on the display **406**, as illustrated by the screenshots of FIGS. **4**A-**4**B, and the flowcharts of FIGS. **5**-**6**.

[0044] FIGS. **4**A-**4**B depict examples of screenshots that depicts a multi-network IM display. In many cases, a user will login to a high level account and adjust user configurations such that the system will automatically login the user to various low level accounts. This may require storing login credentials so that the system can automatically login the user to each selected interface. FIG. **4**A depicts an example of a screenshot **400**A for a full screen display. In the example of FIG. **4**A, the screenshot **400**A includes a contacts list tab **402**,

an "add a network" hyperlink **404**, a plurality of MSN icons **406**, a plurality of Yahoo! icons **408**, and a plurality of AIM icons **410**.

[0045] As the name implies, the contacts list tab **402** includes a listing of designated IM client contacts. For an IM client to be included in the listing, a user associated with the contact list typically must explicitly enter a client into the list. This may be accomplished with a single click. For example, when a new IM client contacts the user, the user may be prompted to allow the IM client to be entered into the user's contacts list.

[0046] As is suggested by the folders depicted in the contacts list tab **402**, a user may organize contacts in various folders. A single contact may be listed in no folders, one folder, or multiple folders, depending upon implementation and user configurations. Depending upon the implementation details, all contacts who are not currently online are listed in the "Offline" folder.

[0047] The "add a network" hyperlink **304** can be followed to add networks for display. In a non-limiting embodiment, when a network is added, contacts of the user in the added network are added to the contacts list tab **402**. The added contacts are distinguishable by network using, for example, the icons **406**, **408**, **410**.

[0048] In an alternative embodiment, the contact list could be maintained in a high level network (i.e., all contacts are listed in the contact list, even if the contacts are from a network that has not been added). If a contact is a member of a network that is not added, the contact would presumably be listed in the "Offline" folder because the contact is not known to the user to be online. However, depending upon the implementation, the "Offline" folder could be effectively split into two folders, if desired, one which indicates a contact is offline, and one that indicates a contact is a member of a network that has not been added. In the example of FIG. **4**A, this is implicit because the added networks have icon **406**, **408**, **410** associated with them, and each of the contacts have icons **406**, **408**, **410** associated with them. So, it is apparent from viewing the display whether a contact is or is not a member of a network that is currently added.

[0049] In a non-limiting embodiment, adding a network requires that the user be a member of the added network. For example, if the user wishes to add the MSN IM network, the user must have an IM account with MSN. However, in an alternative embodiment, a system could set up a dummy account for a user who wishes to add an account for a network with which the user is not a member. The dummy account will be associated with the user, but the user might feel more comfortable with not ever logging in, remembering a password, etc. The system could handle all of these details for the user without the user's knowledge, and without exposing the user to security risks such as compromised passwords (since some users use the same password for multiple accounts, but in this case the system would generate a random, and presumably strong, password for use with the account).

[0050] FIG. **4**B depicts an example of a screenshot **400**B for a mobile phone display. In the example of FIG. **4**B, the contact list includes a series of contacts and their associated network-identifying icons **406**, **408**, **410**. The information available on a mobile phone display is less than that of a full screen display, such as is available on, for example, a laptop display.

[0051] In the example of FIG. **4**B, a user can add accounts by clicking on an "add account" hyperlink **412**. Depending upon the implementation, clicking on the "add account" hyperlink **412** could prompt the user to select an account from a list of accounts the user has already designated in a contacts list (or add a new account), to select a list of accounts associated with a particular subset (see, e.g., the folders of FIG. **4**A), to select a particular network (which may include logging the user in to the network or prompting the user to do so), storing login credentials and, when logging in, the system can automatically login the user to each selected interface, or to select accounts in some other manner. Presumably, if an account is selected for a contact that is a member of a network in which the user has not logged in, in a non-limiting embodiment, either the user or the system will have to login to the network.

[0052] In the example of FIG. **4**B, the user can remove accounts from the online list by clicking on a "signout" hyperlink **414** next to the accounts. Depending on the implementation, clicking on the "signout" hyperlink **414** may cause the user to go offline with respect to the account associated with the particular "signout" hyperlink, or with all accounts on the same network as the account associated with the particular "signout" hyperlink. A "signout" hyperlink **416** may be used to sign out of all networks at once or, in a different implementation, cause the user to be prompted regarding the networks from which to sign out.

[0053] In the example of FIG. **4**B, the screenshot **400**B includes a "reload" hyperlink **418**. It may be desirable to occasionally refresh the screen. Some mobile devices may even be incapable of refreshing without an explicit reload.

[0054] FIG. **5** depicts a flowchart **500** of an example of a method for contact list aggregation and display. This method and other methods are depicted as serially arranged modules. However, modules of the methods may be reordered, or arranged for parallel execution as appropriate. In the example of FIG. **5**, the flowchart **500** begins at module **502** where a high level network is joined. A high level network may include, by way of example but not limitation, eBuddy.

[0055] In the example of FIG. **5**, the flowchart **500** continues to module **504** where a low level network is joined. Low level networks may include IM networks, such as MSN Messenger, Yahoo! Instant Messenger, AIM, or any other known or convenient network. It should also work properly if high level networks were joined as if the high level networks were low level. For example, a member of a first high level network could join a second high level network, which in turn is associated with first and second low level networks. The second high level network would function rather like a node in a tree, where the first high level network is the root, and the leaves of the node are the first and second low level networks. Thus, the second high level network could be thought of as a mid-level network. Unless a distinction is needed, mid-level networks are treated as low level networks herein.

[0056] In the example of FIG. **5**, the flowchart **500** continues to decision point **506** where it is determined whether to join more low level networks. If it is determined that more low level networks are to be joined (**506**-Y), then the flowchart **500** loops back to module **504**, as described previously. If, on the other hand, it is determined that no more low level networks are to be joined (**506**-N), then the flowchart **500** continues to module **508** where a contact list associated with one or more of the low level networks is maintained.

[0057] While contact lists are not particularly important in email environments (because you can write an email to anyone whose email address you know), in IM environments,

contact lists are more desirable because users want to see who is online before sending an IM. If a contact is not online, IM is not allowed in many implementations (and in an implementation that "allows" IM with an offline contact, the communication is arguably not an instant message). Accordingly, at least in an IM environment, most users will maintain a contact list (**508**).

[0058] In the example of FIG. **5**, the flowchart **500** continues to module **510** where the high level networks is logged into. When a user logs into a high level network, the user will typically have a number of options, including requesting that a list of contacts be displayed. The list of contacts could also be displayed automatically upon login, depending upon the implementation and user configurations.

[0059] In the example of FIG. **5**, the flowchart **500** continues to module **512** where contacts from each of the low level networks are displayed in an aggregated contact list. Depending upon the implementation or user configuration, the contact list could include high level contacts, too. Also depending upon the implementation and/or user configuration, the user may or may not be required to login to each of the low level networks for which contacts are to be displayed in the aggregated contact list.

[0060] In the example of FIG. **5**, the flowchart **500** continues to decision point **514** where it is determined whether to quit the high level network. If the user does not quit the high level network (**514**-N), then the flowchart **500** continues to decision point **516** where it is determined whether the user logs out of the high level network. If it is determined that the user is to logout of the high level network (**516**-Y), then the flowchart **500** continues to module **518** where the user logs out. Whether the user logs out (**516**-Y, **518**), or not (**516**-N), the flowchart **500** loops back to decision point **506**, as described previously.

[0061] In many cases, a user may never quit the high level network, however it is theoretically possible that a user would quit (or a user would be banned). If the user quits or is banned from the high level network (**514**-Y), then the flowchart **500** ends.

[0062] FIG. **6** depicts a flowchart **600** of an example of a method for aggregated contact list display. In the example of FIG. **6**, the flowchart **600** starts at module **602** where a member logs into a high level network.

[0063] In the example of FIG. **6**, the flowchart **600** continues to module **604** where a user logs into a low level network through the high level network. Depending upon the implementation and/or user configurations, the user may be logged into the low level network automatically when logging into the high level network, explicitly by the user, or the user may be logged into the low level network when the user selects a contact that is associated with the low level network.

[0064] In the example of FIG. **6**, the flowchart **600** continues to module **606** where high level and low level contacts are displayed in an aggregated contact list. It may be noted that a user may not have a high level contact list, either because a high level system does not allow for contact lists, or because the user does not opt to maintain any contact lists at the high level.

[0065] In the example of FIG. **6**, the flowchart **600** continues to decision point **608** where it is determined whether to logout of the low level network. If it is determined to logout of the low level network (**608**-Y), then the flowchart **600** continues to module **610** where the relevant low level contacts are removed from the aggregated display, and the flowchart **600** loops back to module **606**, as described previously. The relevant low level contacts are those contacts that are in the contact list associated with the low level network from which logout was elected. If, on the other hand, it is determined not to logout of the low level network (**608**-N), then the flowchart **600** continues to decision point **612**.

[0066] In the example of FIG. **6**, the flowchart **600** continues to decision point **612** where it is determined whether to login to a low level network. It is possible to login to an implementation-specific number of low level networks (or, in the alternative, practically any number of low level networks). If it is determined to login to a low level network (**612**-Y), then the flowchart **600** loops back to module **604**, as described previously. If, on the other hand, it is determined not to login to a low level network (**612**-N), then the flowchart **600** continues to decision point **614**, where it is determined whether to logout of the high level network. If not (**614**-N), the flowchart **600** loops back to module **606**, as described previously. If so (**614**-Y), then the flowchart **600** ends.

[0067] FIG. **7** depicts a computer system **700** suitable for implementation of the techniques described above with reference to FIGS. **1-6**. The computer system **700** includes a computer **702**, I/O devices **704**, and a display device **706**. The computer **702** includes a processor **708**, a communications interface **710**, memory **712**, display controller **714**, nonvolatile storage **716**, and I/O controller **718**. The computer **702** may be coupled to or include the I/O devices **704** and display device **706**.

[0068] The computer **702** interfaces to external systems through the communications interface **710**, which may include a modem or network interface. The communications interface **710** can be considered to be part of the computer system **700** or a part of the computer **702**. The communications interface **710** can be an analog modem, ISDN modem, cable modem, token ring interface, satellite transmission interface (e.g. "direct PC"), or other interfaces for coupling a computer system to other computer systems. Although conventional computers typically include a communications interface of some type, it is possible to create a computer that does not include one, thereby making the communications interface **710** optional in the strictest sense of the word.

[0069] The processor **708** may include, by way of example but not limitation, a conventional microprocessor such as an Intel Pentium microprocessor or Motorola power PC microprocessor. While the processor **708** is a critical component of all conventional computers, any applicable known or convenient processor could be used for the purposes of implementing the techniques described herein. The memory **712** is coupled to the processor **708** by a bus **720**. The memory **712**, which may be referred to as "primary memory," can include Dynamic Random Access Memory (DRAM) and can also include Static RAM (SRAM). The bus **720** couples the processor **708** to the memory **712**, and also to the non-volatile storage **716**, to the display controller **714**, and to the I/O controller **718**.

[0070] The I/O devices **704** can include a keyboard, disk drives, printers, a scanner, and other input and output devices, including a mouse or other pointing device. For illustrative purposes, at least one of the I/O devices is assumed to be a block-based media device, such as a DVD player. The display controller **714** may control, in a known or convenient manner, a display on the display device **706**, which can be, for example, a cathode ray tube (CRT) or liquid crystal display (LCD).

[0071] The display controller **714** and I/O controller **718** may include device drivers. A device driver is a specific type of computer software developed to allow interaction with hardware devices. Typically this constitutes an interface for communicating with the device, through a bus or communications subsystem that the hardware is connected to, providing commands to and/or receiving data from the device, and on the other end, the requisite interfaces to the OS and software applications.

[0072] The device driver may include a hardware-dependent computer program that is also OS-specific. The computer program enables another program, typically an OS or applications software package or computer program running under the OS kernel, to interact transparently with a hardware device, and usually provides the requisite interrupt handling necessary for any necessary asynchronous time-dependent hardware interfacing needs.

[0073] The non-volatile storage **716**, which may be referred to as "secondary memory," is often a magnetic hard disk, an optical disk, or another form of storage for large amounts of data. Some of this data is often written, by a direct memory access process, into memory **712** during execution of software in the computer **702**. The non-volatile storage **716** may include a block-based media device. The terms "machine-readable medium" or "computer-readable medium" include any known or convenient storage device that is accessible by the processor **608** and also encompasses a carrier wave that encodes a data signal.

[0074] The computer system **700** is one example of many possible computer systems which have different architectures. For example, personal computers based on an Intel microprocessor often have multiple buses, one of which can be an I/O bus for the peripherals and one that directly connects the processor **708** and the memory **712** (often referred to as a memory bus). The buses are connected together through bridge components that perform any necessary translation due to differing bus protocols.

[0075] Network computers are another type of computer system that can be used in conjunction with the teachings provided herein. Network computers do not usually include a hard disk or other mass storage, and the executable programs are loaded from a network connection into the memory **712** for execution by the processor **708**. A Web TV system, which is known in the art, is also considered to be a computer system, but it may lack some of the features shown in FIG. **6**, such as certain input or output devices. A typical computer system will usually include at least a processor, memory, and a bus coupling the memory to the processor.

[0076] The computer system **700** may be controlled by an operating system (OS). An OS is a software program—used on most, but not all, computer systems—that manages the hardware and software resources of a computer. Typically, the OS performs basic tasks such as controlling and allocating memory, prioritizing system requests, controlling input and output devices, facilitating networking, and managing files. Examples of operating systems for personal computers include Microsoft Windows®, Linux, and Mac OS®. Delineating between the OS and application software is sometimes rather difficult. Fortunately, delineation is not necessary to understand the techniques described herein, since any reasonable delineation should suffice.

[0077] The lowest level of an OS may be its kernel. The kernel is typically the first layer of software loaded into memory when a system boots or starts up. The kernel provides access to various common core services to other system and application programs.

[0078] As used herein, algorithmic descriptions and symbolic representations of operations on data bits within a computer memory are believed to most effectively convey the techniques to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of operations leading to a desired result. The operations are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[0079] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "processing" or "computing" or "calculating" or "determining" or "displaying" or the like, refer to the action and processes of a computer that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0080] An apparatus for performing techniques described herein may be specially constructed for the required purposes, or it may comprise a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, such as, by way of example but not limitation, read-only memories (ROMs), RAMs, EPROMs, EEPROMs, magnetic or optical cards, any type of disk including floppy disks, optical disks, CD-ROMs, DVDs, and magnetic-optical disks, or any known or convenient type of media suitable for storing electronic instructions.

[0081] The algorithms and displays presented herein are not inherently related to any particular computer architecture. The techniques may be implemented using any known or convenient programming language, whether high level (e.g., C/C++) or low level (e.g., assembly language), and whether interpreted (e.g., Perl), compiled (e.g., C/C++), or Just-In-Time (JIT) compiled from bytecode (e.g., Java). Any known or convenient computer, regardless of architecture, should be capable of executing machine code compiled or otherwise assembled from any language into machine code that is compatible with the computer's architecture.

[0082] As used herein, the term "embodiment" means an embodiment that serves to illustrate by way of example but not limitation.

[0083] It will be appreciated to those skilled in the art that the preceding examples and embodiments are exemplary and not limiting to the scope of the present invention. It is intended that all permutations, enhancements, equivalents, and improvements thereto that are apparent to those skilled in the art upon a reading of the specification and a study of the drawings are included within the true spirit and scope of the present invention. It is therefore intended that the following

appended claims include all such modifications, permutations and equivalents as fall within the true spirit and scope of the present invention.

What is claimed:

1. A system comprising:
a network interface;
a network login engine coupled to the network interface;
a network contacts database embodied in one or more computer-readable mediums;
a web server coupled to the network contacts database;
a contact aggregation engine coupled to the network login engine and the network contacts database;
wherein, in operation, the contact aggregation engine controls the network login engine to login or facilitate login to one or more networks, updates in the networks contacts database contact information associated with the one or more networks, stores the aggregated contact list in a computer-readable medium at the web server, and provides an aggregated contact list including the contact information to a display device.

2. The system of claim **1** further comprising a high level contacts database embodied in a computer readable medium, wherein, in operation, contacts from the high level contacts database are included in the aggregated contact list.

3. The system of claim **1**, wherein the network login engine includes a plurality of network login engines that include sufficient information to login or facilitate login to a respective plurality of low level networks.

4. The system of claim **1**, further comprising a user profile database, embodied in a computer-readable medium and coupled to the network login engine, wherein, in operation, the user profile database includes a user-specific information that the network login engine uses to login or facilitate login to the one or more networks.

5. The system of claim **1**, further comprising the display device.

6. The system of claim **1**, further comprising the one or more networks.

7. A method comprising: joining a high level network;
joining one or more low level networks;
maintaining a contact list associated with the one or more low level networks;
logging into the high level network;
displaying contacts from the one or more low level networks in an aggregated contact list.

8. The method of claim **7**, further comprising logging into the one or more low level networks.

9. The method of claim **7**, further comprising:
maintaining a contact list associated with the high level network;
displaying contacts from the high level network in the aggregated contact list.

10. The method of claim **7**, further comprising joining an additional low level network, wherein the aggregated contact list does not include contacts of the additional low level network.

11. The method of claim **7**, further comprising:
logging into the one or more low level networks and an additional low level network;
displaying contacts from the one or more low level networks and the additional low level network in the aggregated contact list;
logging out of the additional low level network.

12. The method of claim **7**, further comprising logging out of the high level network.

13. A method comprising:
logging into a high level network; logging into a low level network;
displaying an aggregated contact list that includes contacts associated with the low level network through the high level network.

14. The method of claim **13**, further comprising automatically logging into the low level network in response to logging into the high level network.

15. The method of claim **13**, further comprising facilitating an explicit login to the low level network.

16. The method of claim **13**, wherein the logging into the low level network is in response to a user selecting a contact for display in the aggregated contact list that is associated with the low level network.

17. The method of claim **13**, wherein the low level network is a first low level network, further comprising:
logging into a second low level network through the high level network;
displaying the aggregated contact list, including contacts associated with the second low level network.

18. The method of claim **17**, further comprising:
logging out of the second low level network;
removing low level contacts associated with the second low level network from the aggregated contact list.

19. The method of claim **17**, further comprising:
logging out of the first low level network;
removing low level contacts associated with the first low level network from the aggregated contact list.

20. The method of claim **13**, further comprising displaying the aggregated contact list including contacts associated with the high level network.

* * * * *