

**SYSTEM AND METHOD FOR INTEGRATING AN ADDRESS BOOK
WITH AN INSTANT MESSAGING APPLICATION IN A MOBILE STATION**

TECHNICAL FIELD

5 The technology described in this patent document relates generally to the field of wireless communication devices. More particularly, the patent document describes a system and method for integrating an address book with an instant messaging application in a mobile station.

BACKGROUND ART

10 Mobile stations are proliferating both in quantity and type. The types of mobile stations may include cell phones, smart phones, PDAs, pagers, phone-enabled laptop computers and a range of other devices. The applications available to a mobile station are also expanding rapidly. One of the newest applications to be ported to a mobile station is the instant messaging applications. In a typical mobile station, however, multiple address or
15 contact databases are not integrated. For instance, addresses used in an instant messaging application (instant messaging "handles") along with other instant messaging information (e.g., presence information) are typically not integrated with contact information in an address book application.

DISCLOSURE OF THE INVENTION

20 In accordance with the teachings described herein, systems and methods are provided for integrating an address book application with an instant messaging application in a mobile station. An instant messaging database may be used to store instant messaging data, the instant messaging data including instant messaging contacts and instant messaging (IM) presence information to indicate whether the instant messaging contacts may currently
25 be accessed by the instant messaging application. An address book database may be used for storing address book data. In one embodiment, an aggregated data viewer may be used to relate instant messaging data from the instant messaging database with address book data from the address book database. In another embodiment, an aggregated data and facilities database may be used to relate instant messaging data from the instant messaging database
30 with address book data from the address book database.

 An instant messaging user interface for use in the mobile station may include a buddy list and a contacts name list. The buddy list may include a list of instant messaging (IM) handles that are stored in the instant messaging database and that may be contacted

using the instant messaging software application. The contact name list may provide address book data corresponding to one or more of the IM handles in the buddy list. In another embodiment, an instant messaging user interface for use in the mobile station may include an instant messaging (IM) group list, a contact name list, and instant messaging presence information. The IM group list may include a list of IM groups, wherein each IM group represents a plurality of instant messaging (IM) handles that are stored in the instant messaging database and that may be contacted using the instant messaging software application. The contact name list may provide address book data corresponding to one or more of the IM handles in each of the IM groups. The instant messaging presence information may indicate a number of IM handles in each group that are currently available to be contacted via the instant messaging application.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of an example system for integrating an address book application with an instant messaging application in a mobile station;

Figs. 2A and 2B include four signal flow diagrams illustrating an example operation of the system of Fig. 1;

Fig. 3 is a block diagram of another example system for integrating an address book application with an instant messaging application in a mobile station;

Figs. 4A and 4B include five signal flow diagrams illustrating an example operation of the system of Fig. 3;

Fig. 5 illustrates one example of an address book entry that has been aggregated with instant messaging fields;

Fig. 6A illustrates one example of an instant messaging user interface in which instant messaging (IM) buddy list information is aggregated with address book contact name information;

Fig. 6B illustrates an example menu function that may be accessed from the user interface of Fig. 6A;

Fig. 7A illustrates one example of an instant messaging user interface in which IM group list and presence information is aggregated with address book contact name information;

Fig. 7B illustrates an example menu for selecting options available from the user interface of Fig. 7A; and

Fig. 8 is a block diagram of an example mobile station.

BEST MODE FOR CARRYING OUT THE INVENTION

With reference now to the drawing figures, Fig. 1 is a block diagram of an example system for integrating an address book application 16 with an instant messaging application 10 in a mobile station 100. In addition to the address book application 16 and the instant messaging application 10, the system includes an instant messaging (IM) library module 12, an instant messaging database 14, an address library module 18, an address book database 20, and an aggregated data and facilities database 22A.

The instant messaging application 10 is a software application that provides a user interface for creating, viewing and managing instant messages and related instant messaging data, such as instant messaging contact information (i.e., IM handles) or other IM data. The buddy/group list and transient state database 14 (the "IM database") stores IM data, such as IM handles (e.g., a buddy list), IM handle group information (e.g., group lists), transient state data relating to one or more IM handles, and/or other instant messaging data. Transient state data may include instant messaging (IM) presence information, which indicates the current status (e.g., online or offline) of an instant messaging contact. The IM library 12 is an application program interface (API) that provides a generic interface to the IM database 14. In addition, the IM library 12 may perform other object-oriented software functions, such as displaying IM data, presenting menu items relating to IM data or other functions.

The address book application 16 is a software application that provides a user interface for creating, viewing and managing address book data, such as contact names, addresses, email addresses, telephone numbers, and other information. The address book database 20 stores address book data. The address library 18 is an application program interface (API) that provides a generic interface to the address book library 20, and may also perform other object oriented software functions, such as displaying address book data, presenting menu items relating to address book data or other functions.

The aggregated data and facilities database 22A may be used to establish a relationship between the address book application 16 and the instant messaging application 10. This relationship also extends between the IM database 14 and the address book database 20. The aggregated data and facilities database 22A enables the mobile station 100 to relate IM handles and other IM data from the IM database 14 with address book data from the address book database 20. Data propagated into the aggregated data and facilities database 22A may, for example, be presented and manipulated by the address book

application 16 and/or the instant messaging application 10. In this manner, the address book application 16 may be used to manage data from all available communication methods (e.g., e-mail, SMS, IM and Phone). In addition, address book data may be accessed and manipulated from the instant messaging application 10.

5 In operation, the address book application 16 may be used to enter both address book data (e.g., name, address, telephone number, etc.) and static instant messaging data (e.g., IM handles) into the aggregated data and facilities database 22. The IM messaging application may be used to enter static instant messaging data (e.g., IM handles) into the aggregated data and facilities database 22, and may also automatically update the database
10 22 with current transient instant messaging data (e.g., IM presence information). The IM library API 12 and the address book library API 18 propagate this information to their respective databases 14, 20. In addition, the IM library API 12 and address book library API 18 may place data from their respective databases 14, 20 into the aggregated data and facilities database 22A to relate IM data with address book information. For example, the
15 IM library API 12 and the address library API 18 may access IM and address book data to build buddy and group lists, as illustrated in Figs. 5-7B. The IM and address book data propagated to the aggregated data and facilities database 22A may then be displayed and/or manipulated using the address book application 16 and/or the instant messaging application 10.

20 Figs. 2A and 2B include four signal flow diagrams illustrating an example operation of the system of Fig. 1. The first signal flow diagram 202, 204 illustrates an example operation for defining the structure of the aggregated data and facilities database 22A. The second signal flow diagram 206-210 illustrates an example operation for entering contact information into the aggregated data and facilities database 22A. The third signal flow
25 diagram 212, 214 illustrates an example operation for integrating address book information from the aggregated data and facilities database 22A into the instant messaging application 10. The forth signal flow diagram 216-218 illustrates an example operation for updating IM presence information in the aggregated data and facilities database 22A.

In the first signal flow diagram 202, 204, the structure of the aggregated data and facilities database 22A is defined by the address book library API 18 and the IM library API
30 12. First, an address book schema 202 is created by the address book library API 18 to define address book data fields for the database 22A, such as a first name field, a last name field, an address field, a telephone number field, an e-mail address field, a company name

field, etc. The address book schema is then extended by the IM library API 12 to add instant messaging data fields, such as an IM identification field (e.g., an IM handle) and an IM presence information field. The schema will preferably include metadata associated with each field that identifies the contents of the field, such that the data fields stored in the database 22A may be generically identified by the applications 10, 16 and APIs 12, 18. For example, the address book application 16 is able to identify an IM identification field in the aggregated data and facilities database 22A from the associated metadata without executing IM-specific software. The metadata associated with each data field may, for example, identify a representation (e.g., numeric, phone format, free form, date, etc.), a range (e.g., 3 lines, 7 digits, etc.) and a label (e.g., "Last Name," "Home Number," etc.)

In the second signal flow diagram 206-210, the address book application 16 is used to enter a new contact record into the database 22A. At step 206, the address book application 16 adds an entry into the database 22A that may include both address book fields (e.g., first name, last name, phone number, etc.) and instant messaging fields (i.e., an IM identification.) When the database 22A is updated to include the new address book entry 206, a notification 207 is sent to the IM library API 12. If the IM library API 12 detects an IM identification field in the new address book entry, then the IM identification field may be used to determine other IM-specific information, such as IM presence information (e.g., currently online or offline). The address book entry may then be updated 208 by the IM library API 12 to include IM presence information, or other IM-specific data. At step 209, the address book application 16 is notified of the IM-specific data added to the address book entry, and the address book application user interface (UI) is updated to reflect the change at step 210.

In the third signal flow diagram 212, 214, address book information (e.g., first name, last name, phone number, address, etc.) is retrieved by the instant messaging application 10. At steps 212 and 214, address book records associated with one or more IM identifications are retrieved from the aggregated data and facilities database 22A by the instant messaging application 10. For example, address book records may be retrieved for each IM identification that is currently listed as online according to the IM presence information. Because the metadata for each address book entry includes information identifying the contents of each field, the IM application 16 may access and display the address book records without executing address book specific software. In this manner, address book records may be accessed, displayed and manipulated from within the instant messaging

application 10. In addition, address book fields may provide links from within the instant messaging application 16 to launch other applications. For instance, a voice communication application may be launched by selecting a displayed phone number field from an address book entry.

5 In the forth signal flow diagram 216-218, IM-specific information, such as presence information, is automatically updated (step 216) in the aggregated data and facilities database 22A when a change is detected by the IM library API 12. For example, if the availability status (i.e., online or offline) of an IM user having an IM identification in the aggregated data and facilities database 22A changes, then the presence information
10 associated with the IM identification is updated at step 216 to reflect the new availability status. Once the database 22A is updated, the address book application 16 is notified (step 217), and the address book application user interface (UI) is updated to reflect the change at step 218.

Fig. 3 is a block diagram of another example system for integrating an address book
15 application 16 with an instant messaging application 10 in a mobile station 100. The example system of Fig. 3 is similar to the system of Fig. 1, except that an aggregated data viewer program 22B is used in place of the aggregated data and facilities database 22A. The aggregated data viewer 22B may be operable to route data from the instant messaging and address book applications 10, 16 to the appropriate database API 12, 18. In addition,
20 the aggregated data viewer 22B may also provide API entry points that may be used by the instant messaging and address book software applications 10, 16 to retrieve and store IM and address book information via the IM library and address book APIs 12, 18, as described below with reference to Figs. 4A and 4B. For example, the aggregated data viewer 22B may enable the software applications 10, 16 to generate buddy or group lists or similar sets
25 of aggregated information, as illustrated in figs. 5-7B.

Figs. 4A and 4B include five signal flow diagrams illustrating an example operation of the system of Fig. 3. The first signal flow diagram 226-231 illustrates an example operation for registering API entry points for accessing and manipulating data in the IM and address book databases 14, 20. The second signal flow diagram 238, 240 illustrates an
30 example operation for requesting data entry elements for the address book application. The third signal flow diagram 242-244 illustrates an example operation for editing and storing contact information from the address book application 16, including both address book fields and IM fields. The forth signal flow diagram 248-250 illustrates an example

operation for contact records on an address book application user interface. The fifth signal flow diagram 254-256 illustrates an example operation for displaying contact records with an IM application 10.

In the first signal flow diagram 226-231, API entry points are registered for the IM
5 and address book databases 14, 20. At step 226 the address library API 18 provides API entry points (e.g., function pointer or object pointers) to the aggregated data viewer 22B specific to the address book database 20. Similarly, at step 227 the IM library API 12 provides the aggregated data view 22B with API entry points specific to the IM database 14. The aggregated data viewer 22B provides an indirection repository of API entry points.
10 Preferably, the indirection repository may be directly supplied by the runtime environment (e.g., by providing object based storage in a Java virtual machine.) If not supplied by the runtime environment, all API registrations and references to library-specific data should preferably be identified with a guide to insure that API calls process their own data. The API entry points supplied by the AB and IM libraries 12, 18 into the indirection repository
15 may include functions, such as library-specific data loading, library specific display, library-specific editing, library implemented actions, and/or other library-specific functions. Preferably, the specific APIs can be negotiated at runtime by defining a base entry point that request further services.

Once the API entry points have been registered with the aggregated data viewer
20 22B, interfaces for data entry may be requested by the IM application 12 and address book application 16 (steps 228 and 230). In response to a data entry interface request, the aggregated data viewer 22B returns opaque references to the address book library 18 and the IM library 12, as shown at steps 229 and 231.

In the second signal flow diagram 238, 240, data entry elements are requested for
25 populating the data entry fields in the address book application user interface with both address book and IM information. The data entry elements enable the address book application 16 to display both address book and IM entry fields on its user interface and receive inputted data for the fields, without requiring the address book application 16 to recognize the content or structure of the fields. That is, the representation of the particular
30 data entry fields is controlled by the aggregated data viewer 22A.

Once contact information has been entered for a new address book entry, the address book application 16 requests edited data elements from the address library and IM library APIs to enable storage of the data fields in either the address book or IM database 14, 20, as

shown in the third signal flow diagram 242-244. The address book and IM fields in the new contact record are then stored in the appropriate database 14, 20 by the aggregated data viewer 22B in step 244.

5 In the forth signal flow diagram 248-250, the data fields for a contact record are loaded from the address book and IM databases 14, 20 for display in a contact record by the address book application 16. In step 248 the address book application 16 sends a request to the aggregated data viewer 22B to load the data fields for one or more contact records. Then, using API information stored with the data record, the address book application 16 may request display elements from the address and IM libraries 12, 18 for displaying the
10 contact record data fields on the address book user interface. Because the library APIs 12, 18 control the information displayed by the address book user interface, changes to transient data (such as an IM presence field) may be immediately reflected in the address book user interface.

In the fifth signal flow diagram 254-256, the data fields for a contact record are
15 loaded from the address book and IM databases 14, 20 for display in a contact record by the IM application 10. In step 254 the IM application 10 sends a request to the aggregated data viewer 22B to load the data fields for one or more contact records. Then, using API information stored with the data record, the IM application 10 may request display elements from the address and IM libraries 12, 18 for displaying the contact record data fields on the
20 IM user interface. Because the library APIs 12, 18 control the information displayed by the IM user interface, the IM application 10 may perform address book related functions without having information regarding the contents of the specific address book field (e.g., linking to a voice communication function from an address book telephone number field.)

Fig. 5 illustrates one example of an address book entry that has been aggregated with
25 instant messaging fields. With this address book entry aggregated from the IM and address book databases 14, 20, the address book application 16 may be used to add, remove, change and access address book entries with IM handles. For example, selecting an IM handle (e.g., JsmithsCurrentHandle) from the address book application 16 may launch an instant messaging communication using the instant messaging application 10.

30 In another example application, an IM handle may be used to run a search using the address book application 16 to locate a contact entry. For instance, with reference to the illustrated example, a mobile device user may be able to enter the IM handle "JsmithsCurrentHandle" into the address book application in order to locate and access all

of the illustrated contact information for "John Smith." In addition, a user who is viewing the IM information within the address book may also see real-time presence or status information (e.g., "IM Online").

Fig. 6A illustrates one example of an instant messaging user interface in which IM buddy list information is aggregated with address book contact name information. This enables a user to select from address book contact names associated with the IM handles from the IM database 14. The list of address book contact names is aggregated from the address book database 20 and includes only contact names that are associated with IM handles. This correlation between address book data and IM data may, for example, be controlled by the IM library API 12 and/or the address library API 18, or by the aggregated data viewer 22B. From the illustrated lists, a device user may scroll and select either an address book contact name or an IM handle to initiate an instant messaging session.

Fig. 6B illustrates an example menu function that may be accessed from the user interface of Fig. 6A. The illustrated menu function may, for example, be accessed from the instant messaging application 10 by selecting an address book contact name or associated IM handle. Options may be selected from the menu to open an instant messaging session, edit an address book entry or IM handle, delete an address book entry or IM handle, create a new address book entry, or to perform other similar functions. The options may, for example, be selected from the user interface using a touch-screen, cursor keys, a mouse, a roller wheel, or some other input device(s). In this manner address book entries may be added, deleted and/or modified from within the IM application 10.

The aggregated contact record illustrated in Figs. 6A and 6B may also enable other inquiries, additions and/or modification to be performed between the instant messaging and address book applications 10, 16. For example, if the user is adding an IM contact with the IM application 10, then the IM application 10 can propagate this information into the address book. During this propagation the IM application 10 may first verify that the new IM contact does not already exist. In addition, presence information may be modified within the address book database on a real-time basis.

Fig. 7A illustrates one example of an instant messaging user interface in which IM group list and presence information is aggregated with address book contact name information. By aggregating the information in this manner, a synergy may be created between the online presence information and the address book contact names associated with an IM group. For instance, the example of Fig. 7A shows that the IM group

"MyFavorites" has 5 members currently present (i.e., online). Options available for the IM group data may be accessed by selecting from a menu, as illustrated in Fig. 7B. For example, options may be selected to open an instant messaging session with the currently present members of a selected group, list all of the members of a group, edit the members of the group, delete a group, create a new group, or perform other IM functions.

Fig. 8 is a block diagram of an example mobile station 100 in which the above-described system may be implemented. The mobile station 100 includes a processing subsystem 338, a communications subsystem 311, a short-range communications subsystem 340, a memory subsystem 324, 326, and various other device subsystems and/or software modules 342. The mobile station 100 also includes a user interface, which may include a display 322, a keyboard 332, a speaker 334, a microphone 336, one or more auxiliary input/output devices 328, a serial port 330, and/or other user interface devices.

The processing subsystem 338 controls the overall operation of the mobile station 100. Operating system software executed by the processing subsystem 338 may be stored in a persistent store, such as a flash memory 324, but may also be stored in other types of memory devices in the memory subsystem, such as a read only memory (ROM) or similar storage element. The flash memory 324 may be segregated into different areas for both computer programs and program data storage 350. For instance, a plurality of PIM software applications may allocate a portion of the flash memory 324 for its database requirements. Flash memory segments 350 may, for example, include configuration information, mobile station state information (e.g., current radio network and region), address book information, calendar profiles, instant message configuration, telephone profiles and other PIM information, such as tasks and todo lists. In addition, the flash memory 324 may include a plurality of software application programs executed by the processing subsystem 328, such as a voice communication module, a data communication module, and/or other software modules.

System software, specific device applications, or parts thereof, may be temporarily loaded into a volatile store, such as a random access memory (RAM) 326. Communication signals received by the mobile station 100 may also be stored to RAM 326.

Communication functions, including data and voice communications, are performed through the communication subsystem 311, and possibly through the short-range communications subsystem 340. The communication subsystem 311 includes a receiver 312, a transmitter 314 and one or more antennas 316, 318. In addition, the communication

subsystem 311 also includes a processing module, such as a digital signal processor (DSP) 320 or other processing device(s), and local oscillators (LOs) 313. The specific design and implementation of the communication subsystem 311 is dependent upon the communication network in which the mobile station 100 is intended to operate. For example, a mobile station 100 may include a communication subsystem 311 designed to operate within the Mobitex™ mobile communication system, the DataTAC™ mobile communication system, a GSM network, a GPRS network, a UMTS network, and/or an EDGE network.

Network access requirements vary depending upon the type of communication system. For example, in the Mobitex and DataTAC networks, mobile stations are registered on the network using a unique personal identification number or PIN associated with each device. In UMTS and GSM/GPRS networks, however, network access is associated with a subscriber or user of a device. A GPRS device therefore requires a subscriber identity module, commonly referred to as a SIM card, in order to operate on a GSM/GPRS network.

When required network registration or activation procedures have been completed, the mobile station 100 may send and receive communication signals over the communication network 319. Signals received by the antenna 316 from the communication network 319 are routed to the receiver 312, which provides signal amplification, frequency down conversion, filtering, channel selection, etc., and may also provide analog to digital conversion. Analog-to-digital conversion of the received signal allows the DSP to perform more complex communication functions, such as demodulation and decoding. In a similar manner, signals to be transmitted to the network 319 are processed (e.g., modulated and encoded) by the DSP 320 and are then provided to the transmitter 314 for digital to analog conversion, frequency up conversion, filtering, amplification and transmission to the communication network 319 (or networks) via the antenna 318.

In addition to processing communication signals, the DSP 320 provides for receiver 312 and transmitter 314 control. For example, gains applied to communication signals in the receiver 312 and transmitter 314 may be adaptively controlled through automatic gain control algorithms implemented in the DSP 320.

In a data communication mode, a received signal, such as a text message or web page download, is processed by the communication subsystem 311 and input to the processing device 338. The received signal is then further processed by the processing device 338 for output to a display 322, or alternatively to some other auxiliary I/O device 328. A device user may also compose data items, such as email messages, using a keyboard

332 and/or some other auxiliary I/O device 328, such as a touchpad, a rocker switch, a thumb-wheel, or some other type of input device. The composed data items may then be transmitted over the communication network 319 via the communication subsystem 311.

5 In a voice communication mode, overall operation of the device is substantially similar to the data communication mode, except that received signals are output to a speaker 334, and signals for transmission are generated by a microphone 336. Alternative voice or audio I/O subsystems, such as a voice message recording subsystem, may also be implemented on the device 100. In addition, the display 322 may also be utilized in voice communication mode, for example to display the identity of a calling party, the duration of
10 a voice call, or other voice call related information.

The short-range communications subsystem 340 enables communication between the mobile station 100 and other proximate systems or devices, which need not necessarily be similar devices. For example, the short-range communications subsystem 340 may include an infrared device and associated circuits and components, or a Bluetooth™
15 communication module to provide for communication with similarly-enabled systems and devices.

This written description uses examples to disclose the invention, including the best mode, and also to enable a person skilled in the art to make and use the invention. The patentable scope of the invention may include other examples that occur to those skilled in
20 the art. For example, in one alternate embodiment a mobile device may include more than one instant messaging system, with each instant messaging system being integrated together and with the address book database as described above.

INDUSTRIAL APPLICABILITY

25 The present invention is directed at a system and method for integrating an address book with an instant messaging application in a mobile station

It is claimed:

1. A system for integrating an address book software application with an instant messaging software application in a mobile station, comprising:

an instant messaging database for storing instant messaging data, the instant messaging data including instant messaging contacts and instant messaging (IM) presence information to indicate whether the instant messaging contacts may currently be accessed by the instant messaging application;

an address book database for storing address book data; and

an aggregated data viewer operable to relate instant messaging data, necessarily the presence information, from the instant messaging database with address book data from the address book database.

2. The system of claim 1, further comprising:

an instant messaging library application program interface (API) operable to interface the aggregated data viewer with the instant messaging database; and

an address book library API operable to interface the aggregated data viewer with the address book database.

3. The system of claim 1, wherein the aggregated data viewer enables both address book data and instant messaging data to be accessed and displayed by the address book software application.

4. The system of claim 1, wherein the aggregated data viewer enables both address book data and instant messaging data to be accessed and displayed by the instant messaging software application.

5. The system of claim 2, wherein the aggregated data viewer is operable to provide API entry points that may be used by the address book software application to retrieve and store address book data via the address book library API and instant messaging data via the instant messaging library API.

6. The system of claim 5, wherein the instant messaging library API is operable to automatically update the address book software application with changes to the IM presence information.

5 7. The system of claim 5, wherein the API entry points include library-specific data loading functions, library-specific display functions and library-specific editing functions.

8. The system of claim 5, wherein the address book library API is operable to register API entry points with the aggregated data viewer that may be used to store and retrieve
10 address book data via the address book library API, and wherein the instant messaging library API is operable to register API entry points with the aggregated data viewer that may be used to store and retrieve instant messaging data via the instant messaging library API.

15 9. The system of claim 8, wherein the API entry points are function pointers.

10. The system of claim 8, wherein the API entry points are object pointers.

11. The system of claim 5, wherein the aggregated data viewer provides an indirection repository for storing the API entry points.

20 12. The system of claim 11, wherein the indirection repository is an object based storage in a virtual machine.

13. The system of claim 2, wherein the aggregated data viewer is operable to provide
25 API entry points that may be used by the instant messaging software application to retrieve address book data from the address book database via the address book library API and to retrieve instant messaging data from the instant messaging database via the instant messaging library API.

30 14. The system of claim 13, wherein the API entry points include library-specific data loading functions, library-specific display functions and library-specific editing functions.

15. The system of claim 13, wherein the address book library API is operable to register API entry points with the aggregated data viewer that may be used to retrieve address book data via the address book library API, and wherein the instant messaging library API is operable to register API entry points with the aggregated data viewer that may be used to retrieve instant messaging data via the instant messaging library API.
16. The system of claim 15, wherein the API entry points are function pointers.
17. The system of claim 15, wherein the API entry points are object pointers.
18. The system of claim 13, wherein the aggregated data viewer provides an indirection repository for storing the API entry points.
19. The system of claim 18, wherein the indirection repository is an object based storage in a virtual machine.
20. The system of claim 13, wherein the API entry points may be used by the instant messaging software application to store instant messaging data in the instant messaging database via the instant messaging library API.
21. A system for integrating an address book software application with an instant messaging software application in a mobile station, comprising:
- an instant messaging database for storing instant messaging data, the instant messaging data including instant messaging contacts and instant messaging (IM) presence information to indicate whether the instant messaging contacts may currently be accessed by the instant messaging application;
 - an address book database for storing address book data;
 - an aggregated data and facilities database for relating instant messaging data from the instant messaging database with address book data from the address book database;
 - an instant messaging library application program interface (API) operable to propagate the instant messaging data to the aggregated data and facilities database; and
 - an address library API operable to propagate the address book data to the aggregated data and facilities database,

wherein the address book software application is operable to access and display both address book data and instant messaging data, necessarily the presence information, by means of the aggregated data and facilities database.

22. The system of claim 21, wherein the aggregated data and facilities database enables both address book data and instant messaging data to be accessed and displayed by the instant messaging software application.

23. The system of claim 21, wherein the address book software application is operable to enter both address book data and instant messaging data into the aggregated data and facilities database;

the instant messaging library API being further operable to interface with the aggregated data and facilities database to store the address book data in the address book database; and

the address book library API being further operable to interface with the aggregated data and facilities database to store the instant messaging data in the instant messaging database.

24. The system of claim 23, wherein the aggregated data and facilities database is operable to notify the instant messaging software application of instant messaging data entered into the aggregated data and facilities database by the address book software application.

25. The system of claim 21, wherein the instant messaging software application is operable to automatically update the aggregated data and facilities database with current IM presence information.

26. The system of claim 25, wherein the instant messaging library API is further operable to interface with the aggregated data and facilities database to store the IM presence information in the instant messaging database.

27. The system of claim 25, wherein the aggregated data and facilities database is

operable to notify the address book software application of the current IM presence information.

28. The system of claim 23, wherein the address book library API is operable to define the structure of the aggregated data and facilities database by creating an address book schema that identifies address book data fields.

29. The system of claim 28, wherein the instant messaging library API is operable to further define the structure of the aggregated data and facilities database by extending the address book schema to identify instant messaging data fields.

30. The system of claim 29, wherein the address book schema includes metadata that identifies the contents of each address book data fields and instant messaging data fields, wherein the metadata is used by the address book software application to identify instant messaging data fields and is used by the instant messaging software application to identify address book data fields.

31. The system of claim 30, wherein the metadata identifies a representation, a name, and a range for each address book data field and instant messaging data field.

32. An instant messaging user interface for use in a mobile station, the mobile station including a system for integrating an address book software application with an instant messaging software application that includes an instant messaging database for storing instant messaging data, an address book database for storing address book data, and a means for relating instant messaging data from the instant messaging database with address book data from the address book database, the user interface comprising:

- a buddy list that includes a list of instant messaging (IM) handles that are stored in the instant messaging database and that may be contacted using the instant messaging software application; and

- a contact name list that provides address book data corresponding to one or more of the IM handles in the buddy list.

33. The instant messaging user interface of claim 32, wherein a menu function may be executed by selecting an entry in the buddy list or the contact name list.

34. The instant messaging user interface of claim 33, wherein the menu function is operable to open an instant messaging session using an IM handle associated with the selected entry.

35. The instant messaging user interface of claim 33, wherein the menu function is operable to modify address book data associated with the selected entry.

36. An instant messaging user interface for use in a mobile station, the mobile station including a system for integrating an address book software application with an instant messaging software application that includes an instant messaging database for storing instant messaging data, an address book database for storing address book data, and a means for relating instant messaging data from the instant messaging database with address book data from the address book database, the user interface comprising:

- a instant messaging (IM) group list that includes a list of IM groups, wherein each IM group represents a plurality of instant messaging (IM) handles that are stored in the instant messaging database and that may be contacted using the instant messaging software application;

- a contact name list that provides address book data corresponding to one or more of the IM handles in each of the IM groups; and

- instant messaging presence information that indicates a number of IM handles in each group that are currently available to be contacted via the instant messaging application.

37. The instant messaging user interface of claim 36, wherein a menu function may be executed by selecting an entry in the IM group list or the contact name list.

38. The instant messaging user interface of claim 37, wherein the menu function is operable to open an instant messaging session using one or more IM handles associated with the selected entry.

39. The instant messaging user interface of claim 37, wherein the menu function is operable to modify address book data associated with the selected entry.