(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 February 2006 (02.02.2006)

PCT

(10) International Publication Number
WO 2006/010263 A1

(51) International Patent Classification[7]: G06F 9/44, 3/02

(21) International Application Number:
PCT/CA2005/001176

(22) International Filing Date: 27 July 2005 (27.07.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/592,128 30 July 2004 (30.07.2004) US

(71) Applicant *(for all designated States except US)*: **RESEARCH IN MOTION LIMITED** [CA/CA]; 295 Phillip Street, Waterloo, Ontario N2L 3W8 (CA).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **STOUT, Craig, Allen** [CA/CA]; c/o Research in Motion Limited, 450 March Road, Kanata, Ontario K2K 3K2 (CA). **EMERY, Jeffrey, Kenneth** [CA/CA]; c/o Research in Motion Limited, 450 March Road, Kanata, Ontario K2K 3K2 (CA). **VANDEN HEUVEL, David** [CA/CA]; c/o Research in Motion Limited, 450 March Road, Kanata, Ontario K2K 3K2 (CA). **VARANDA, Marcelo** [CA/CA]; c/o Research in Motion Limited, 450 March Road, Kanata, Ontario K2K 3K2 (CA). **JAIN, Gaurav** [CA/CA]; c/o Research in Motion Limited, 450 March Road, Kanata, Ontario K2K 3K2 (CA).

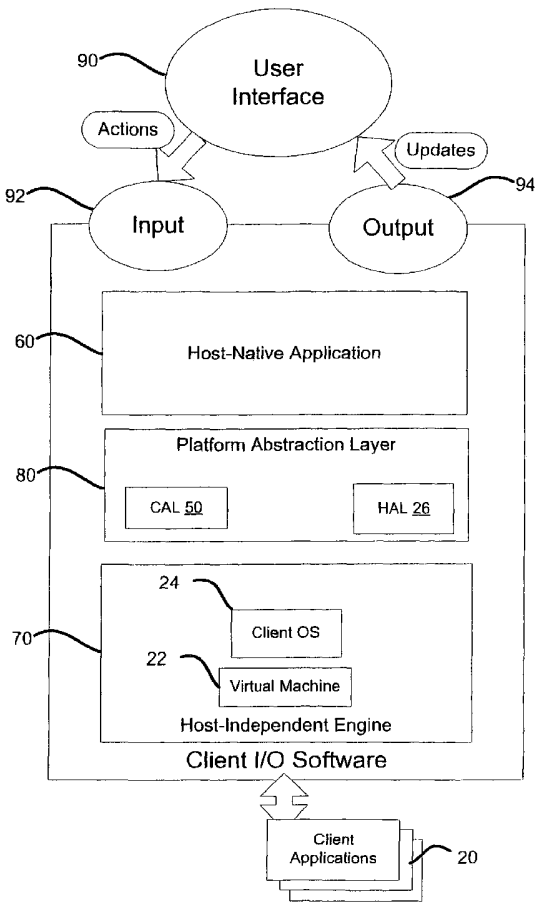
(74) Agent: **MOFFAT & CO.**; 1200 - 427 Laurier Avenue West, Ottawa, Ontario K1R 7Y2 (CA).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH,

[Continued on next page]

(54) Title: CLIENT-HOST DIVIDED ARCHITECTURE FOR INPUT-OUTPUT COORDINATION



(57) Abstract: A system and method for integrating a client with a host device where a client application has access to a user interface of the host device, the system having: a host independent engine adapted to provide an execution environment for the client application; a host native application adapted to provide access to a user interface on the host device; and a platform abstraction layer adapted to isolate the host independent engine and the host native application, the platform abstraction layer being configured to adapt host independent engine device calls to the host native application and adapt host native application calls to the host independent engine.

WO 2006/010263 A1

GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

CLIENT-HOST DIVIDED ARCHITECTURE FOR INPUT-OUTPUT COORDINATION

FIELD OF THE APPLICATION

[0001] The present application deals with a method and system for coordinating input and output between a communications client and its host device and, in particular, to an method and system in which a client is able to gain access through the host to a user interface.

BACKGROUND

[0002] In a host wireless device, it is sometimes desirable to add a client onto the host to perform functionality that the host normally would not include. The host is typically certified with its software and hardware to communicate over a wireless network, whereas a client typically would not be. Further, certification could occur prior to the client being added, especially in the case that the client is integrated after-market onto the wireless device.

[0003] It is further desirable that the client is able to communicate with the native applications on the host and that the host applications are able to communicate with client applications. This communication preferably includes controlling a user interface on the host device from a client application, including registering inputs to the host device for the client application and displaying or outputting from the client application.

[0004] In some cases it is also desirable to be able to use device settings from the host environment in a client setting. Examples of this could include locale information, time zones, display themes or backgrounds. The automatic propagation of a change in host device setting would be preferable in some situations.

[0005] In one embodiment it is also desirable to have symbol inputs to a client correspond with symbol inputs to a host. It is further desirable that the input of symbols be simplified.

[0006] It is further desirable to be able to upgrade or downgrade the provisioning of a client directly from a host device without having to load new software on the host device.

SUMMARY

[0007] The present system and method provides a divided architecture for integrating a client into a host wireless device. One key to the present system is that the host is recognized as the dominant determinant in a divided architecture due to the fact that device type certification efforts (Global Certification Forum (CGF)/ PCS Type Certification Review Board (PTCRB)) may happen prior to the client being integrated onto the host. This necessitates that the host and tightly-tied applications to the host remain unfettered.

[0008] The present system and method provides a virtual machine that is started upon start-up of the host device and is used to run client applications. The virtual machine communicates through a client OS that would normally send client application commands and functions to host dependent features, such as hardware, software, firmware or communications networks. However, since the host dependent features are certified and controlled by the host device, the operating system instead communicates with abstraction layers. The abstraction layers have a native interface for communicating with host applications, allowing client applications to use the host dependent features by utilizing host applications.

[0009] Device setting such as locale, time zone, display themes and backgrounds can be set using a binary variable. In one mode, the client settings are adapted to automatically adjust when host device settings are changed. This change is propagated by either having a listener at the host to signal a change in device settings, or polling when a graphical interface of a client is brought to the foreground. In the other mode the client settings can be fixed at the client and changes at the host device are ignored.

[0010] A client application accesses the user interface of a host device using a host native application, a platform abstraction layer and a host independent engine communicating between the user interface and a client application. The host independent engine is platform independent and relies on the platform abstraction layer to translate and/or map function calls. The host native application depends on the user interface and host device, and is used to control actions and updates to the user interface.

[0011] One example of an input for the host native application is the input of symbols. In a system for inputting symbols to a client where the host has a native system for inputting symbols from a host symbol table by navigating a host cursor to move between adjacent symbols displayed within a host grid and the host further has a keyboard, the keyboard can be taken advantage of to map symbols to one keystroke. A client symbol table is created conforming to the host symbol table, and a grid is made where the indicia of at least one keyboard key is associated with a symbol such that when a user actuates a key in the keyboard, the cursor jumps to the corresponding symbol.

[0012] Provisioning of the device can be accomplished from software that is already loaded onto the device. By following steps from a client application on a host device provisioning of the client can be changed. A host device user is thereby enabled to upgrade or downgrade client service, i.e. to provision the data client

[0013] The present application therefore provides a system for integrating a client with a host device where a client application has access to a user interface of the host device, said system comprising: a host independent engine adapted to provide an execution environment for the client application; a host native application adapted to provide access to a user interface on the host device; and a platform abstraction layer adapted to isolate said host independent engine and said host native application, said platform abstraction layer being configured to adapt host independent engine device calls to the host native application and adapt host native application calls to the host independent engine..

[0014] The present application further provides a system for inputting symbols to a client from a host, the host having a native system for inputting symbols from a host symbol table by navigating a host cursor to move between adjacent symbols displayed within a host grid, and to change host symbol pages, the host further having a keyboard, said system comprising: a client symbol table conforming to the host symbol table; a client grid displaying symbols from said client symbol table; a cursor to move between adjacent symbols displayed within the client grid; wherein the client grid further displays the indicia of at least one keyboard key such that when a user actuates one of said at least one keyboard key, the cursor jumps to the corresponding symbol.

[0015] The present application still further provides a method for integrating a client with a host device where a client application has access to a user interface of the host device, said method comprising the steps of: executing the client applications on a host independent engine; providing a user interface on the host device through a host native application; and isolating said host independent engine from said host native application using a platform abstraction layer by adapting in said platform abstraction layer host independent engine device calls to the host native application and adapting in the platform abstraction layer host native application calls to the host independent engine.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] The present system and method will be better understood with reference to the drawings in which:

FIGURE 1 is a schematic diagram of the components and dataflow according to the present system and method;

FIGURE 2 is a screen-capture of a host application showing various applications that can be selected in the host environment;

FIGURE 3 is a screen-capture of a client application started from a host environment;

FIGURE 4 is a screen-capture of a client application started from the host environment in a host application;

FIGURE 5 is a schematic diagram of the components and dataflow for a user interface according to one aspect of the present system and method;

FIGURE 6 is a view of a host symbol table;

FIGURE 7 is a view of various input options on a host device;

FIGURE 8 shows a 9*6 grid in a generally QWERTY keyboard layout with symbols mapped to certain letters;

FIGURE 9 shows a 9*6 grid in a generally AZERTY keyboard layout with symbols mapped to certain letters;

FIGURE 10 shows a 9*6 grid in a generally QWERTZ keyboard layout with symbols mapped to certain letters;

FIGURE 11 is a block diagram illustrating a host mobile station;

FIGURE 12 is a flowchart showing the upgrade in the provisioning of a client; and

FIGURE 13 is a flowchart showing the downgrade in the provisioning of a client.

DETAILED DESCRIPTION OF THE DRAWINGS

[0017] The present system and method is directed to a divided architecture for a client on a host device. One example of such an arrangement would be a data-enabled cellular telephone with a data device client running on top of the host telephone environment. Other examples of clients running on host environments would, however, be known to those skilled the art and the above is not meant to limit the scope of the present method and system. The examples below will use a host that is a cellular telephone and a client that is a data device client merely for illustration purposes.

[0018] A host device will require certification prior to being released for sale and use in a given market. Examples of certification include GSF- and PCS-type certification review board (PCTRB) certifications. These certifications are for the hardware and tightly-tied applications to this hardware.

[0019] In order to include a client that has communications capabilities without having to certify the client, the integration of the client requires a divided architecture in which the phone and the tightly-tied applications to the phone remain unaltered.

[0020] One example of an architecture to accomplish this is illustrated in **Figure 1**. **Figure 1** shows a method and system for a divided architecture **10** which includes client applications **20** running on top of a virtual machine **22.**

[0021] Client applications **20** can be any application that is designed to run on a virtual machine **22**. In the example of **Figure 1**, these could include a messages application **31** for viewing messages that have been received, a contacts application **32**, which presents an address book including phone numbers, e-mail addresses or other contact information for individuals or companies, calendar application **33** for scheduling appointments and managing time, a browser application **34** for browsing the internet or other network, a compose-message application **35** to compose messages for SMS or e-mail, a save-messages application **36** to view messages that have been saved, a search-messages application **37** to search for a particular message, a lock application **38** to lock the keyboard and screen of the mobile device, and a set-up application **39** to change the set-up configuration for client **15**. Other applications **30** could also exist as part of client applications **20** and the above-listed applications are not meant to be limiting. Further, other clients besides client **15** could exist on the host device and these other clients could have applications **29** which could be invoked from application **30**.

[0022] Virtual machine **22** is preferably started at power-up of the host device and stays running no matter what. In one preferred embodiment, the virtual machine is a JAVA virtual machine and client applications **20** are JAVA applications.

[0023] All client applications **20** use virtual machine **22** to invoke instances of objects created by client applications **20**.

[0024] A feature call such as a hardware call on system **10** from client applications **20** would normally go through client OS **24**. Client OS **24** includes a number of primitives for interacting with hardware. However, in the case that client applications **20** are built onto a host device and because the host device has acquired certification for its host dependent features such as hardware, software and firmware, it is preferable that instead of interacting with the features directly, client OS **24** interacts with a host abstraction layer **26**. Host abstraction layer **26** converts calls from client applications **20** to host calls through a native interface **28**. Native interface **28** invokes host applications **40** in order to use the host dependent features on the host device.

[0025] Because host applications invoke the features of the host device rather than client applications directly utilizing the features, the above architecture provides that client applications **20** can run on a host environment and use the features of the host

device without having to re-certify. This enables the client to be added to the host device after certification, including an after-market addition to the host device.

[0026] One example of a client application using the above includes the making of a telephone call when the host device is a cellular telephone. When in the host environment this simply involves using host applications to create the telephone call where these host applications use certified hardware, firmware and software to connect through the wireless system. However, when in a client application **20**, the above architecture requires the invoking of a host application in order to make the phone call. A client application could be an address book or contact application **32** that includes phone numbers for individuals. A user may wish to select a phone number from the address book and have the wireless device phone that person. In order to accomplish this, a user may select the phone number and select an option to phone that phone number. In this case, contact application **32** indicates through virtual machine **22** to OS **24** that it needs to make a phone call. Instead of using the host dependent featrue directly from OS **24**, a notification is sent to host abstraction layer **26** which, through native interface **28** invokes the correct host application **40** to make the phone call. In the example of **Figure 1**, this would be phone application **42.** Phone application **42** then starts the phone call and the user proceeds as if the phone call was started from client application **20.**

[0027] Similarly, client application **20** could give a user the option (instead of phoning the phone number) to use a short-message service or a multi-media message service to contact the individual. In each of these cases, a different host application **40** is invoked, but this is done similarly through the host abstraction layer **26** and native interface **28.** In the example of **Figure 1**, these host applications include SMS application **44** or MMS application **46.**

[0028] An alternative example of a client application **20** could be an e-mail message that includes a phone number within it, for example, in messages application **31.** Messages application **31** could give a user the option to contact the phone number with the message. A phone-related application **42**, short-message service (SMS) application **44**, or multi-media message service (MMS) application **46** is started within host applications **40.** This is done through the client OS **24** to the host abstraction layer **26** where the request is converted with a native interface **28** for a host application **40.**

[0029] As one skilled in the art will realize, data is supplied between the applications **20** and host applications **40**. In the example above, the phone number would be supplied to host application **40** including phone application **42**, SMS application **44**, and MMS application **46**.

[0030] It is further desirable that a client application can be activated from a host application **40**. Reference is now made to **Figure 2**. **Figure 2** shows a screen capture of a host application. The host application lists a series of client applications that can be activated. As used herein, activated can mean to both start a client application **20** or to bring an already started client application **20** to the foreground. In order to activate a client application **20**, the user scrolls to the client application that s/he desires and selects the client application. Reference is made again to **Figure 1**. When an application is selected in the host environment, a client application selection application **48** uses a set of application programming interfaces (APIs) by which the host operating system can request a client application **20** to activate.

[0031] Client application selection application **48** uses a client abstraction layer **50** to activate an application within client applications **20**. Client application selection application **48** calls a function that is translated in client abstraction layer **50**. Client abstraction layer **50** then uses virtual machine **22** to activate a client application **20**.

[0032] Client abstraction layer **50** in alternative embodiments can either inject the client OS **24** event into virtual machine **22** which causes the selected client application **20** to become active or, alternatively, performs a "reverse native call", either through client OS **24** or via client connect **52** to manipulate the native representation of some client object which causes the selected client application **20** to become active.

[0033] Client connect **52** can be used for network features for client applications. This enables, for example, client **15** to communicate using a specific protocol that was not originally supported on the host device. Client connect **52** involves a protocol stack to perform this messaging, and thereby increase and improve client functionality.

[0034] An example of the above is a client calendar application as illustrated in the screen-capture **Figure 3**, or a client e-mail application as illustrated in the screen-

capture **Figure 4**. Requests for a client application to be activated are converted into function calls through native interface **28,** which, in turn, makes calls on client applications **20.** These applications **20** are then brought into the display foreground.

[0035] **Figure 3** represents calendar application **33** and **Figure 4** represents a screen-capture of the display of messages application **31**. As will be appreciated by one skilled in the art, a screen bar or other marker on the screen capture could be used to indicate that the client is in the host environment.

[0036] Alternatively, client application selection application **48** may communicate directly with virtual machine **22** in order to activate a client application **20**. This may occur, for example, in the case where client application selection application **48** knows the code or a hook to start client application **20**.

[0037] Once virtual machine **22** receives a message to activate an application, either from the client application selection application **48** directly or through client abstraction layer **50**, a client application is activated and needs to assume control of the host user interface. In order to do this, client application **20** makes a call back to client application selection application **48** indicating that client application **20** needs the user interface. Client application selection application **48** then uses host code to take over the UI and thus becomes a portal between client application **20** and the host. Client application selection application **48** adapts all of the host inputs to events for the client and takes over control of the user interface. Client application selection application **48** is an uncertified embodiment of a host native application **60** described in more detail below. As will be appreciated by one skilled in the art, other embodiments could by certified.

[0038] If the host requires control of the user interface back, client **15** is notified through client application selection application **48** of this.

[0039] It is further desirable when using a host and a client that the device settings be synchronized in certain situations. Device settings could include locale settings, time zone settings or theme settings, Locale, as described herein, includes various settings such as the language of the interface, for example, English or French or Spanish. It could also include the keyboard configuration, e.g., QWERTY, AZERTY, QWERTZ or DVORAK. Theme settings could color patterns and background images.

[0040] In setup application **39** referred in **Figure 1**, a user can choose between a mode that allows the user to use host settings for the device settings or custom settings. As one skilled in the art will appreciate, a different mode setting could be used for theme, locale and time zone, or these could be all included in one mode setting.

[0041] If the mode is set to the host settings, the device settings for the client are synchronized with the host's device settings. Any changes in the host's device settings are propagated to the client and the client's device settings are, therefore, also changed. For example, if a user changes the language from French to English in a host application **40**, this is propagated to client **15** and client applications **20** will use English.

[0042] In the case where the client display is set to mimic the host display, propagation of changes in the host display is accomplished by having a listener application monitoring the host device settings. Upon a change in the host device settings, the host listener will notify setup application **39** that a change has been made to the host device settings and this change will be reflected in the client device settings.

[0043] Alternatively, propagation of a change in the device settings could include polling every time a graphical user interface from the client takes over. This polling involves comparing the host device settings with the client device settings and thereby determining if a change has been made. If a change has been made, the client device settings are updated.

[0044] Thus if the mode is 'automatic' or 'host settings', changes in the host device settings are pushed to the client, either through a listener or by polling, as described above.

[0045] If, conversely, the mode is set to 'manual' or 'client settings', the user can update the device settings in the client and client applications will use these display setting instead of the host device settings. If the mode is set to 'client settings', changes to the host's device settings will be ignored by client applications **20**.

[0046] Reference is now made to **Figure 5**. In order to interact with a user of a host device, client applications **20** need to provide a user interface **90**. On a host device having one or more input devices such as a keypad, keyboard, roller wheel, scrollstrip, touch-pad, d-pad or other navigation device, and a screen, a user must be able to input actions for client applications **20**, and the results of the input and client applications **20** operations need to be displayed on the screen. In order to accomplish this, three components are provided within the I/O architecture of the present system. These are a host-native application (HNA) **60**, platform abstraction layer (PAL) **80**, and a host-independent engine (HIE) **70**.

[0047] HIE **70** is a platform independent component. Since PAL **80** contains the host abstraction layer (HAL) **26** and the client abstraction layer (CAL) **50**, translation between the client and the particular host is performed in it. Client data **27**, as best seen in **Figure 1**, is further included in PAL **80**, and client data **27** includes abstractions such as mapping files to map user interface inputs on the host to client I/O software. In a preferred embodiment PAL **80** is a C function interface, along with mapping files.

[0048] HIE **70** includes both virtual machine **22** and client OS **24**. These are used to activate, start, or call instances of objects in client applications **20** when client **15** is object oriented, or call functions in client **15** when it is not.

[0049] HNA **60** resides beyond PAL **80,** and thus can adapt user interface **90** to conform and adapt to a particular host on behalf of client applications **20**. HNA **60** can take radically different forms depending on the design of the host application infrastructure and the user interface requirements of the host operating system. For example, it is envisaged that in alternate embodiments, a keyboard and display user interface are required, or alternatively a radically different voice-only interface can be provided. HNA **60** is responsible for creating the framework necessary to receive input and update output when a user brings a client application **20** to the foreground. In a keyboard/display embodiment, this may involve creating windows, buttons or graphics widgets of any kind. In a voice-only host embodiment, this may involve speech recognition and voice synthesis.

[0050] For input **92**, HNA **60** is responsible for passing user actions to HIE **70** through platform abstraction layer **80**. Inputs **92** can include button presses, keystrokes, stylus inputs, roller wheel motions, scrollstrip motions, touch-pad

motions, d-pad motions, voice commands, accelerometer motions or other inputs that would be known to those skilled in the art. These inputs are translated and/or mapped as received from the host operating system in PAL **80** and fed through the input function of the platform abstraction layer **80**.

[0051] For output **94**, HNA **60** may receive screen updates from HIE **70** at any time, including when client application **20** is not in the foreground. These updates must be stored and memory is used by HNA **60** to maintain a complete frame buffer copy separate from the application display area. If HNA **60** is in the foreground when receiving an update from HIE **70**, then the application display area must be updated as well as the frame buffer so that the display on the host device reflects the screen change immediately. Whenever HNA **60** transitions into the foreground, it must update the application display area with the complete contents of the frame buffer.

[0052] Other output **94** types envisaged include audio tones, voice, and signals to actuate host-specific features, such as an offset motor or led for discrete notification or indication.

[0053] In a preferred embodiment, HNA **60** uses a framework that updates the user by simply displaying a graphic image provided by PAL **80**, and processes user actions by adapting them to be sent down as events to PAL **80**. This greatly reduces the complexity of HNA **60** thus enhancing the portability of the client to other host devices.

[0054] Reference is now made to **Figures 6-10** which illustrate a specific example of how HNA **60** adapts user actions and provides a user interface on behalf of client applications adapted to the semantics of a particular host.

[0055] First, a system for symbolic input on a particular host is shown in **Figure 6**. The system employs a particular semantic for symbolic input that the users of traditional application on the host will expect to be valid on all applications utilized on the host.

[0056] Operationally, host graphical user interface element **100** offers a 9 x 6 symbol table to a user. Cursor **102** is moved along a 9 x 4 grid in order to select a symbol. Since the number of rows of the grid is smaller than that of the table, the symbols are offered on two pages,

[0057] Referring now to **Figure 7**, the user may manipulate any number of input devices on a particular host, including a keypad **104**, a 4-directional D-pad **106**, rocker switches **108**, as well as a QWERTY keyboard provided in two portions, a left keyboard portion **110L** and a right keyboard portion **110R**. Most notably, the host semantics for symbolic input on this particular host require that cursor **102** be moved on the grid by manipulating D-pad **106** to select a particular symbol.

[0058] In this particular example, HNA **60** preserves the semantics of symbolic input on the host while adapting actions a client application user is likely to desire for symbolic input given the data-centric features of client applications **20**.

[0059] Referring now to **Figure 8**, operationally, client graphical user interface element **114** offers a slightly different 9 x 6 symbol table on two pages, wherein symbols can still be selected by operation of d-pad **106**, thus preserving the host symbolic input semantics. Cursor **116** is now moved however on a 9 x 3 grid instead of a 9 x 6 grid. This grid height is preferable in order to be able to map one row of each of the letter rows of the keyboard of **Figure 7** to one of the symbol rows of the grid. The width of the grid is maintained the same as in the host graphical user element **114** to allow a traditional host application user to quickly learn the layout of the symbol table while utilizing client applications, and to continue to contemporaneously support the use of the d-pad **106** for selecting a symbol.

[0060] Thus, a user is enabled to directly input one of **27** symbols using one keystroke instead of having to resort to using the d-pad **106-D**, while still accepting input using d-pad **106-D**.

[0061] Note that the embodiments of **Figures 8-10** deliberately do not use the right-most key of the top row of a standard keyboard, i.e. in the case of the embodiment of Figure 8, the key marked by the 'P' indicia in right keyboard portion **110R**, since the topmost rows of the three standard keyboards shown herein contains 10 keys. This has been shown above to provide advantages and thus should not be considered a limitation. Nonetheless, it is envisaged that the techniques taught herein could be adapted to other keyboard layouts and grid sizes on a per host basis by those of skill in the art, and thus those adaptations are also within the scope of this application.

[0062] In alternate embodiments, the unused keys can remain unutilized, or they can be assigned a function to further enhance symbolic input, such as toggling between the various symbol pages. It is also envisaged that toggling between symbol pages can be accomplished by use of any one of the many other keys available on the particular keyboard available on the host keyboard.

[0063] **Figures 9 and 10** illustrate how the HNA **60** can further adapt and conform to the semantics of variants of a host. In particular, **Figure 9** shows adaptation and conformity to a host variant having an AZERTY keyboard, and **Figure 10** shows the same in the case of a QWERTZ keyboard. Note that in each case, the indicia used in user interface element **114** conforms to the layout of the particular host keyboard, and that the input actions taken by the user are adapted to select the corresponding symbol.

[0064] To summarize the example, for input, PAL **80** adapts keystrokes by mapping each grid location on the 9 x 3 grid onto a key on the keyboard graphical user interface element, shown as QWERTY, AZERTY and QWERTZ variants in **Figures 8,9 and 10** respectively. For output, PAL **80** enhances the display by showing the indicia of a corresponding alphabetic key directly below each symbol.

[0065] Referring to the drawings, **Figure 11** is a block diagram illustrating a host mobile station including preferred embodiments of the techniques of the present application. Mobile station **1100** is preferably a two-way wireless communication device having at least voice and data communication capabilities. Mobile station **1100** preferably has the capability to communicate with other computer systems on the Internet. Depending on the exact functionality provided, the wireless device may be referred to as a data messaging device, a two-way pager, a wireless e-mail device, a cellular telephone with data messaging capabilities, a wireless Internet appliance, or a data communication device, as examples.

[0066] Where mobile station **1100** is enabled for two-way communication, it will incorporate a communication subsystem **1111**, including both a receiver **1112** and a transmitter **1114**, as well as associated components such as one or more, preferably embedded or internal, antenna elements **1116** and **1118**, local oscillators (LOs) **1113**, and a processing module such as a digital signal processor (DSP) 1120. As will be apparent to those skilled in the field of communications, the particular design of the communication subsystem **1111** will be dependent upon the communication

network in which the device is intended to operate. For example, mobile station **1100** may include a communication subsystem **1111** designed to operate within the Mobitex™ mobile communication system, the DataTAC™ mobile communication system, GPRS network, UMTS network, EDGE network or CDMA network.

[0067] Network access requirements will also vary depending upon the type of network **1119**. For example, in the Mobitex and DataTAC networks, mobile station **1100** is registered on the network using a unique identification number associated with each mobile station. In UMTS and GPRS networks, and in some CDMA networks, however, network access is associated with a subscriber or user of mobile station **1100**. A GPRS mobile station therefore requires a subscriber identity module (SIM) card in order to operate on a GPRS network, and a RUIM in order to operate on some CDMA networks. Without a valid SIM/RUIM card, a GPRS/UMTS/CDMA mobile station may not be fully functional. Local or non-network communication functions, as well as legally required functions (if any) such as "911" emergency calling, may be available, but mobile station **1100** will be unable to carry out any other functions involving communications over the network **1100**. The SIM/RUIM interface **1144** is normally similar to a card-slot into which a SIM/RUIM card can be inserted and ejected like a diskette or PCMCIA card. The SIM/RUIM card can have approximately 64K of memory and hold many key configuration **1151**, and other information **1153** such as identification, and subscriber related information.

[0068] When required network registration or activation procedures have been completed, mobile station **1100** may send and receive communication signals over the network **1119**. Signals received by antenna **1116** through communication network **1119** are input to receiver **1112**, which may perform such common receiver functions as signal amplification, frequency down conversion, filtering, channel selection and the like, and in the example system shown in **Figure 11**, analog to digital (A/D) conversion. A/D conversion of a received signal allows more complex communication functions such as demodulation and decoding to be performed in the DSP **1120**. In a similar manner, signals to be transmitted are processed, including modulation and encoding for example, by DSP **1120** and input to transmitter **1114** for digital to analog conversion, frequency up conversion, filtering, amplification and transmission over the communication network **1119** via antenna **1118**. DSP **1120** not only processes communication signals, but also provides for receiver and transmitter control. For example, the gains applied to communication signals in

receiver **1112** and transmitter **1114** may be adaptively controlled through automatic gain control algorithms implemented in DSP **1120**.

[0069] Network **1119** may further communicate with multiple systems (not shown). For example, network **1119** may communicate with both an enterprise system and a web client system in order to accommodate various clients with various service levels.

[0070] Mobile station **1100** preferably includes a microprocessor **1138** which controls the overall operation of the device. Communication functions, including at least data and voice communications, are performed through communication subsystem **1111**. Microprocessor **1138** also interacts with further device subsystems such as the display **1122**, flash memory **1124**, random access memory (RAM) **1126**, auxiliary input/output (I/O) subsystems **1128**, serial port **1130**, keyboard **1132**, speaker **1134**, microphone **1136**, a short-range communications subsystem **1140** and any other device subsystems generally designated as **1142**.

[0071] Some of the subsystems shown in **Figure 11** perform communication-related functions, whereas other subsystems may provide "resident" or on-device functions. Notably, some subsystems, such as keyboard **1132** and display **1122**, for example, may be used for both communication-related functions, such as entering a text message for transmission over a communication network, and device-resident functions such as a calculator or task list.

[0072] Operating system software used by the microprocessor **1138** is preferably stored in a persistent store such as flash memory **1124**, which may instead be a read-only memory (ROM) or similar storage element (not shown). Those skilled in the art will appreciate that the operating system, specific device applications, or parts thereof, may be temporarily loaded into a volatile memory such as RAM **1126**. Received communication signals may also be stored in RAM **1126**.

[0073] As shown, flash memory **1124** can be segregated into different areas for both computer programs **1158** and program data storage **1150**, **1152**, **1154** and **1156**. These different storage types indicate that each program can allocate a portion of flash memory **1124** for their own data storage requirements. Microprocessor **1138**, in addition to its operating system functions, preferably enables execution of software

applications on the mobile station. A predetermined set of applications that control basic operations, including at least data and voice communication applications for example, will normally be installed on mobile station **1100** during manufacturing. A preferred software application may be a personal information manager (PIM) application having the ability to organize and manage data items relating to the user of the mobile station such as, but not limited to, e-mail, calendar events, voice mails, appointments, and task items. Naturally, one or more memory stores would be available on the mobile station to facilitate storage of PIM data items. Such PIM application would preferably have the ability to send and receive data items, via the wireless network **1119**. In a preferred embodiment, the PIM data items are seamlessly integrated, synchronized and updated, via the wireless network **1119**, with the mobile station user's corresponding data items stored or associated with a host computer system. Further applications may also be loaded onto the mobile station **1100** through the network **1119**, an auxiliary I/O subsystem **1128**, serial port **1130**, short-range communications subsystem **1140** or any other suitable subsystem **1142**, and installed by a user in the RAM **1126** or preferably a non-volatile store (not shown) for execution by the microprocessor **1138**. Such flexibility in application installation increases the functionality of the device and may provide enhanced on-device functions, communication-related functions, or both. For example, secure communication applications may enable electronic commerce functions and other such financial transactions to be performed using the mobile station **1100**.

[0074] In a data communication mode, a received signal such as a text message or web page download will be processed by the communication subsystem **1111** and input to the microprocessor **1138**, which preferably further processes the received signal for output to the display **1122**, or alternatively to an auxiliary I/O device **1128**. A user of mobile station **1100** may also compose data items such as email messages for example, using the keyboard **1132**, which is preferably a complete alphanumeric keyboard or telephone-type keypad, in conjunction with the display **1122** and possibly an auxiliary I/O device **1128**. Such composed items may then be transmitted over a communication network through the communication subsystem **1111**.

[0075] For voice communications, overall operation of mobile station **1100** is similar, except that received signals would preferably be output to a speaker **1134** and signals for transmission would be generated by a microphone **1136**. Alternative voice or audio I/O subsystems, such as a voice message recording subsystem, may

also be implemented on mobile station **1100**. Although voice or audio signal output is preferably accomplished primarily through the speaker **1134**, display **1122** may also be used to provide an indication of the identity of a calling party, the duration of a voice call, or other voice call related information for example.

[0076] Serial port **1130** in **Figure 11** would normally be implemented in a personal digital assistant (PDA)-type mobile station for which synchronization with a user's desktop computer (not shown) may be desirable, but is an optional device component. Such a port **1130** would enable a user to set preferences through an external device or software application and would extend the capabilities of mobile station **1100** by providing for information or software downloads to mobile station **1100** other than through a wireless communication network. The alternate download path may for example be used to load an encryption key onto the device through a direct and thus reliable and trusted connection to thereby enable secure device communication.

[0077] Other communications subsystems **1140**, such as a short-range communications subsystem, is a further optional component which may provide for communication between mobile station **1100** and different systems or devices, which need not necessarily be similar devices. For example, the subsystem **1140** may include an infrared device and associated circuits and components or a Bluetooth™ communication module to provide for communication with similarly enabled systems and devices.

[0078] A mobile communications device, such as a phone, is typically formed of software, firmware, and hardware adapted to provide communications services over a wireless communications network. This process of forming the relationship between the mobile communications device and the service is known in the art as provisioning. Typically a network operator provisions the mobile via a subscription to a service contract. Thus, once the mobile has been provisioned, the user of the mobile is often referred to as a subscriber.

[0079] In a voice and data network such as GSM (Global System for Mobile Communication) and GPRS (General Packet Radio System), CDMA (Code Division Multiple Access), or various other third generation networks such as EDGE (Enhanced Data rates for GSM Evolution) or UMTS (Universal Mobile Telecommunications Systems), both voice and data services may be available to

mobile communications devices. Example voice services include voice calling and Short Messaging Service (SMS). Example data services include Internet browsing, email, and Multimedia Messaging Service (MMS).

[0080] Although many services may be available on a given network, only those subscribers that use mobile communications devices that have been provisioned for those services will be able to benefit from them. This may present problems for the subscriber and the network operator alike. On one hand, the subscriber may desire an existing service he does not have, i.e. an upgrade, or desire disabling a service, i.e. a downgrade. On the other hand the operator may want to offer a new service, but may hesitate if subscribers cannot benefit from them.

[0081] One known solution is to provide an out of band communications link, such as a Universal Serial Bus, on the mobile communications device, and enable the subscriber to load new software onto the mobile via the out of band communication link using a personal computer, thus re-provisioning the device. This may be an unacceptable solution to both the subscriber and the operator as there is a significant risk that the mobile, by error, receives a wrong or incomplete load, and may require servicing. Furthermore, this solution may be unacceptable to the subscriber who does not have access to a personal computer.

[0082] However, since mobile station **1100** is a host communications device that hosts client **15**, client **15** may be provisioned directly by a user of mobile station **1110**.

[0083] Referring now to **Figure 12**, an exemplary upgrade of client **15** is illustrated.

Method of:	User upgrade to UPGRADED client service
Pre-Condition(s):	Device is configured to allow service change Current Service Type = BASE Service Lock = NO_LOCK Service Change = NO_CHANGE
Acts and steps:	1. USER activates service configuration menu item. 2. CLIENT examines factory settings for service type 3. CLIENT displays "Contact Operator" informational dialog to indicate to the user that they must do this in order to get service. (Display **1203**) 4. USER selects to proceed. 5. CLIENT displays "Verification" informational dialog to indicate user desires to upgrade and indicates the need to reinstall desktop. (Display **1204**) 6. USER selects to proceed. 7. CLIENT displays "Loss of Data" question to verify the user has done a

backup prior performing upgrade. (Display **1205**)
8. USER selects to proceed.
9. CLIENT saves new service type and sets 'service change' flag.
10. CLIENT informs Host applications of required change to point to Client applications
11. CLIENT triggers clear of Host calendar data. (Display **1206**)
12. CLIENT informs user the device will be powered off. (Display **1207**)
13. CLIENT requests Host to power off the device.
14. USER starts device.
15. HOST processes the change and clears 'service change' flag.
16. USER de-installs current Client desktop to remove base configured desktop
17. User re-installs Client desktop and selects Upgrade configuration.
18. USER connects device to Client desktop
19. Client Desktop synchronizes with wireless data server
20. Client Desktop associates identifier with user's corporate email account
21. Client Desktop downloads keys & service books
22. Client Desktop performs bulk download of calendar
23. CLIENT registers on network
24. USER starts receiving wireless email and calendar events.

Post Condition(s): Provisioning recorded as complete.
Current Service Type = UPGRADED
Service Lock = NO_LOCK
Service Change = NO_CHANGE

[0084] Referring now to **Figure 13**, an exemplary downgrade of client **15** is illustrated.

User wants to downgrade to base service

Method of: user downgrade to base service

Pre-Condition(s): Device is configured to allow service change
Current Service Type = BASE
Service Lock = NO_LOCK
Service Change = NO_CHANGE

Acts and steps:
1. USER activates service configuration menu item.
2. CLIENT examines factory settings for service type.
3. CLIENT displays "Verification" informational dialog to indicate user desires to downgrade. (Display **1303**)
4. USER selects to proceed.
5. CLIENT displays "Loss of Data" question to verify the user has done a backup prior performing upgrade. (Display **1304**)
6. USER selects to proceed.
7. CLIENT saves current service type
8. CLIENT informs Host applications of required change to point to Host applications
9. CLIENT performs delete of all data contained in the CLIENT file system
10. CLIENT informs Host applications to perform "KillDevice" IT policy request to clear all corporate data in Host file systems. (Display **1305**)
11. CLIENT sets the 'service change' flag.
12. CLIENT informs user the device will be powered off. (Display **1306**)
13. CLIENT requests Host to power off the device.
14. USER starts device.
15. HOST processes the change and clears 'service change' flag.
16. USER de-installs current Client desktop to remove upgrade configured desktop
17. User re-installs Client desktop and selects base configuration.

Post Condition(s):	Provisioning recorded as complete. Current Service Type = BASE Service Lock = NO_LOCK Service Change = NO_CHANGE

[0085] Thus, the host device user is enabled to upgrade or downgrade client service, i.e. to provision the data client.

[0086] The embodiments described herein are examples of structures, systems or methods having elements corresponding to elements of the techniques of this application. This written description may enable those skilled in the art to make and use embodiments having alternative elements that likewise correspond to the elements of the techniques of this application. The intended scope of the techniques of this application thus includes other structures, systems or methods that do not differ from the techniques of this application as described herein, and further includes other structures, systems or methods with insubstantial differences from the techniques of this application as described herein.

CLAIMS

We claim:

1. A system for integrating a client with a host device where a client application has access to a user interface of the host device, said system comprising:
   a. a host independent engine adapted to provide an execution environment for the client application;
   b. a host native application adapted to provide access to a user interface on the host device; and
   c. a platform abstraction layer adapted to isolate said host independent engine and said host native application, said platform abstraction layer being configured to adapt host independent engine device calls to the host native application and adapt host native application calls to the host independent engine.

2. The system of claim 1, wherein said platform abstraction layer is configured to translate host independent engine device calls to the host native application and translate host native application calls to the host independent engine.

3. The system of claim 1 or 2, wherein said host independent engine device calls include user interface update events.

4. The system of any of claims 1 to 3, wherein said host native application calls include user interface actions.

5. The system of claim 2, wherein said host includes a native system for inputting symbols to a client from a host symbol table, the system further comprising:
   a. a client symbol table in said platform abstraction layer conforming to the host symbol table.

6. The system of claim 5, wherein said platform abstraction layer is adapted to map said host symbol table to said client symbol table.

7. The system of claim 6, wherein said system further comprises a client grid displaying symbols from said client symbol table.

8. The system of claim 7, wherein said platform abstraction layer is further adapted to add indicia of at least one keyboard key to said client grid, said platform abstraction layer further being adapted to provide said client grid with said added indicia to said host native application for display on the user interface.

9. The system of claim 8, wherein said system is adapted to move a cursor to a corresponding symbol when a user actuates said at least one keyboard key.

10. The system of claim 9, wherein said indicia are configured in accordance with a keyboard on said host device.

11. The system of claim 10, wherein said indicia configuration is a keyboard layout selected from the group consisting of a QWERTY keyboard layout, a DVORAK keyboard layout and a QWERTZ keyboard layout.

12. The system of any of claims 1 to 11, wherein the host device is a wireless data device.

13. A system for inputting symbols to a client from a host, the host having a native system for inputting symbols from a host symbol table by navigating a host cursor to move between adjacent symbols displayed within a host grid, and to change host symbol pages, the host further having a keyboard, said system comprising:

a. a client symbol table conforming to the host symbol table;
b. a client grid displaying symbols from said client symbol table;
c. a cursor to move between adjacent symbols displayed within the client grid;

wherein the client grid further displays the indicia of at least one keyboard key such that when a user actuates one of said at least one keyboard key, the cursor jumps to the corresponding symbol.

14. The system of claim 13, wherein said indicia are configured in accordance with a keyboard on said host device.

15. The system of claim 14, wherein said indicia configuration is a keyboard layout selected from the group consisting of a QWERTY keyboard layout, a DVORAK keyboard layout and a QWERTZ keyboard layout.

16. A method for integrating a client with a host device where a client application has access to a user interface of the host device, said method comprising the steps of:
    a. executing the client applications on a host independent engine;
    b. providing a user interface on the host device through a host native application; and
    c. isolating said host independent engine from said host native application using a platform abstraction layer by adapting in said platform abstraction layer host independent engine device calls to the host native application and adapting in the platform abstraction layer host native application calls to the host independent engine.

17. The method of claim 16, wherein the adapting host independent engine device calls to the host native application step and the adapting host native application calls to the host independent engine step include translating host independent engine device calls to the host native application and host native application calls to the host independent engine.

18. The method of claim 16 or 17, wherein said host independent engine device calls include user interface update events.

19. The method of any of claims 16 to 18, wherein said host native application calls include user interface actions.

20. The method of claim 17, further comprising the steps of:
    mapping a host symbol table to a client symbol table in said platform abstraction layer.

21. The method of claim 20, wherein said further comprises the step of displaying symbols from said client symbol table on a client grid.

22. The method of claim 21, further comprising the steps of adding indicia of at least one keyboard key to said client grid, and providing said client grid with said added indicia to said host native application for display on the user interface.

23. The method of claim 22, further comprising the step of moving a cursor to a corresponding symbol when a user actuates said at least one keyboard key.

24. The method of claim 23, further comprising the step of configuring said indicia according to a keyboard layout, wherein keyboard layout is selected from the group consisting of a QWERTY keyboard layout, a DVORAK keyboard layout and a QWERTZ keyboard layout.

25. The method of any of claims 16 to 23, wherein the host device is a wireless data device.

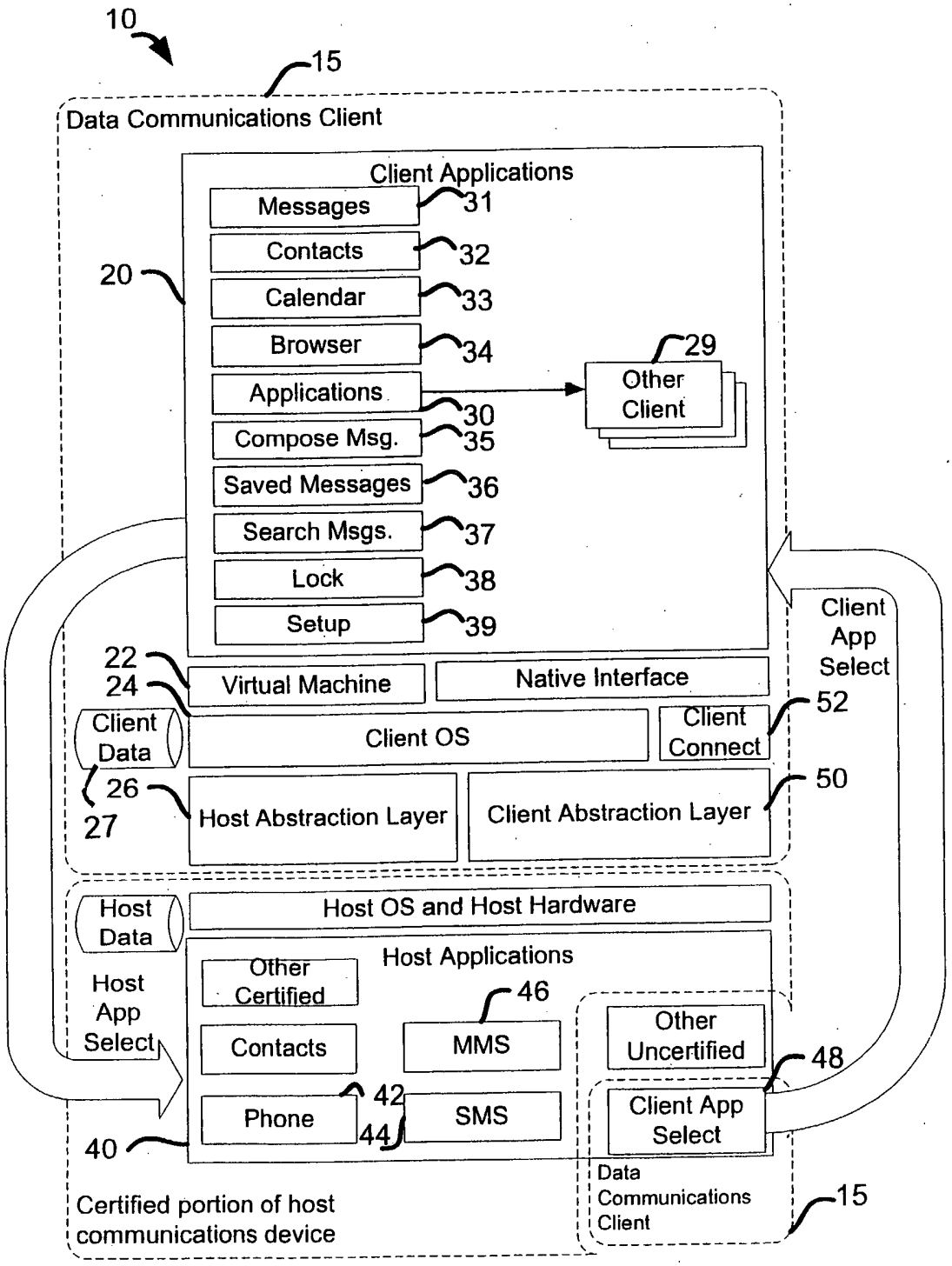

10
15
Data Communications Client
Client Applications
Messages
31
Contacts
32
20
Calendar
33
Browser
34
29
Other Client
Applications
30
Compose Msg.
35
Saved Messages
36
Search Msgs.
37
Lock
38
Setup
39
Client App Select
22
Virtual Machine
Native Interface
24
Client Data
Client OS
Client Connect
52
26
27
Host Abstraction Layer
Client Abstraction Layer
50
Host Data
Host OS and Host Hardware
Host Applications
Other Certified
Host App Select
46
Contacts
MMS
Other Uncertified
48
Phone
42
SMS
44
Client App Select
40
Data Communications Client
15
Certified portion of host communications device

FIG. 1

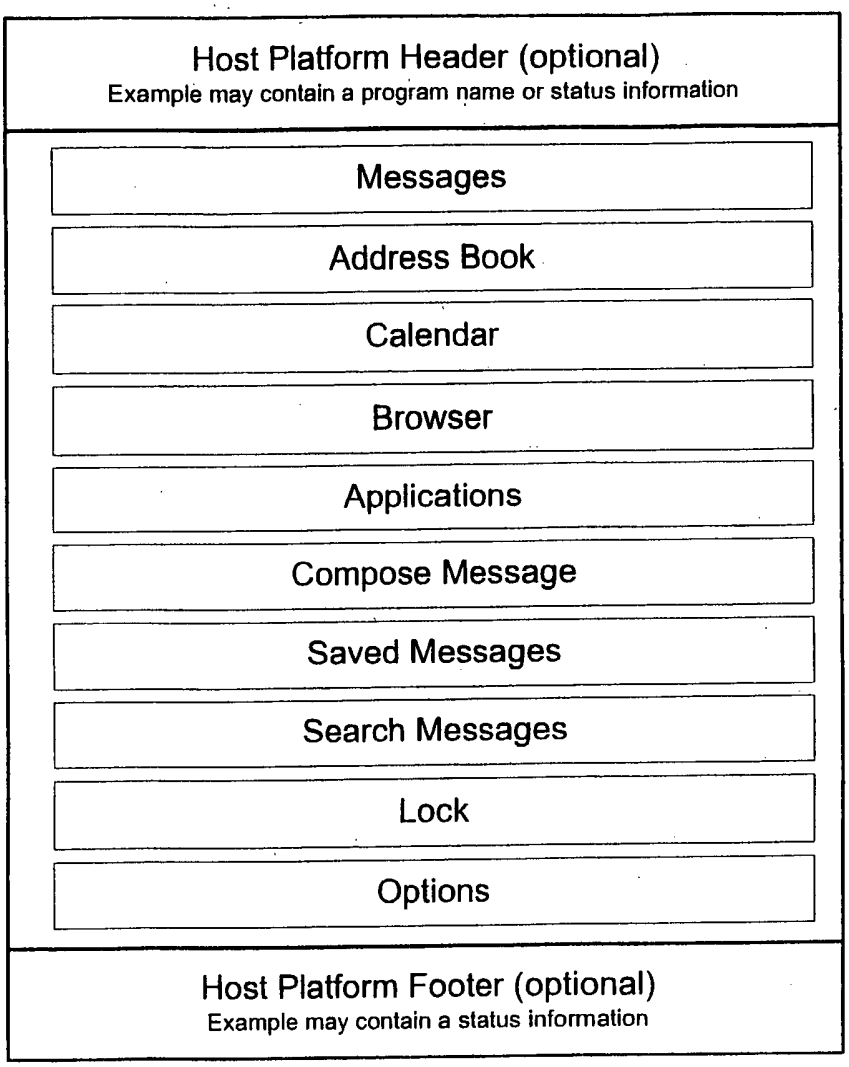

Host Platform Header (optional)
Example may contain a program name or status information
Messages
Address Book
Calendar
Browser
Applications
Compose Message
Saved Messages
Search Messages
Lock
Options
Host Platform Footer (optional)
Example may contain a status information

FIG. 2

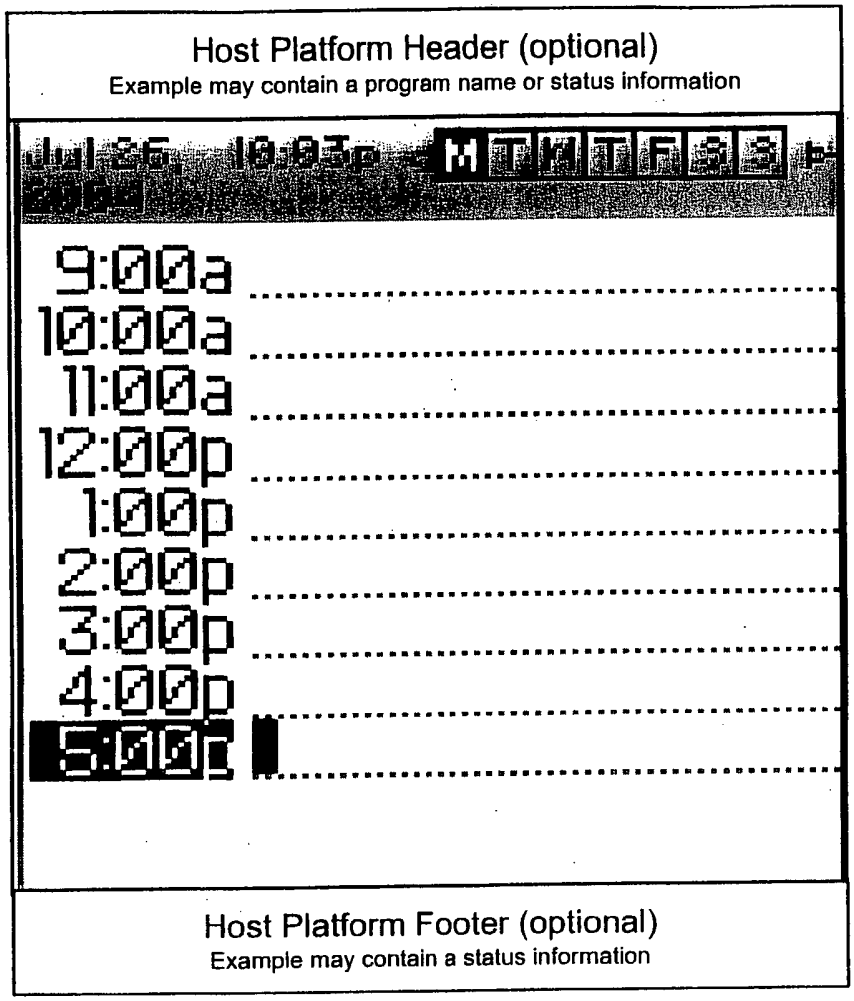

Host Platform Header (optional)
Example may contain a program name or status information
M T W T F S S
9:00a
10:00a
11:00a
12:00p
1:00p
2:00p
3:00p
4:00p
5:00p
Host Platform Footer (optional)
Example may contain a status information

FIG. 3

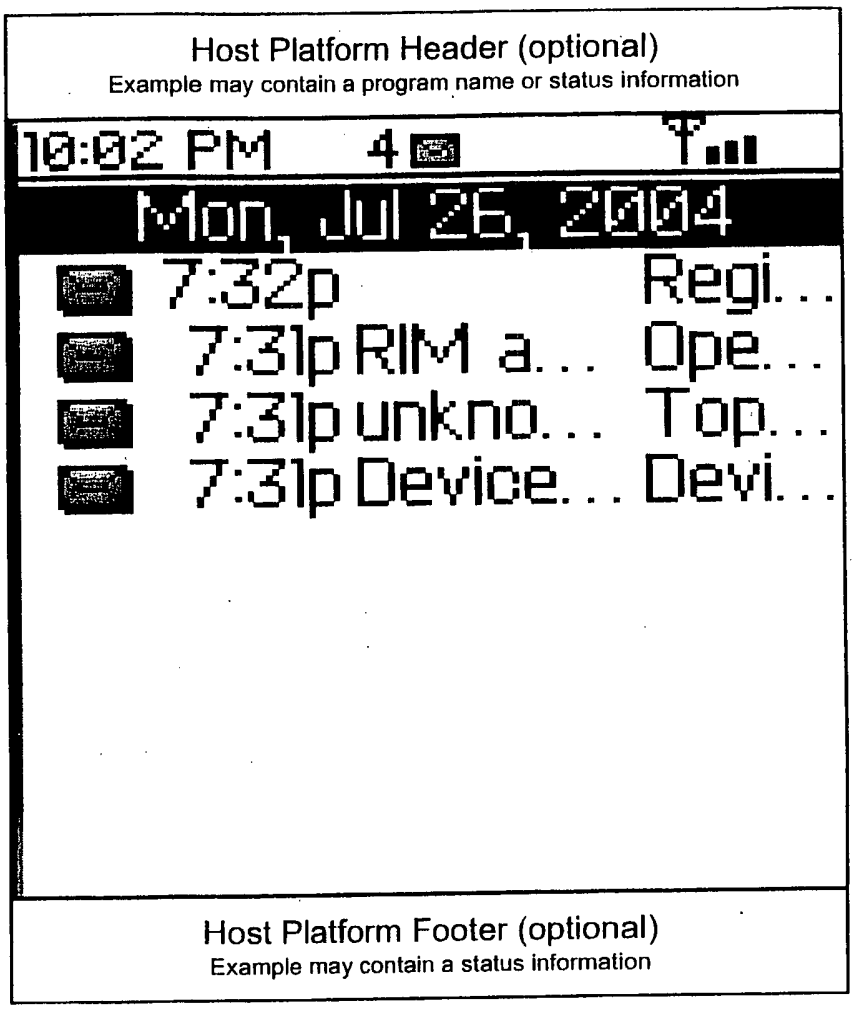

Host Platform Header (optional)
Example may contain a program name or status information
10:02 PM
Mon, Jul 26, 2004
7:32p Regi...
7:31p RIM a... Ope...
7:31p unkno... Top...
7:31p Device... Devi...
Host Platform Footer (optional)
Example may contain a status information

FIG. 4

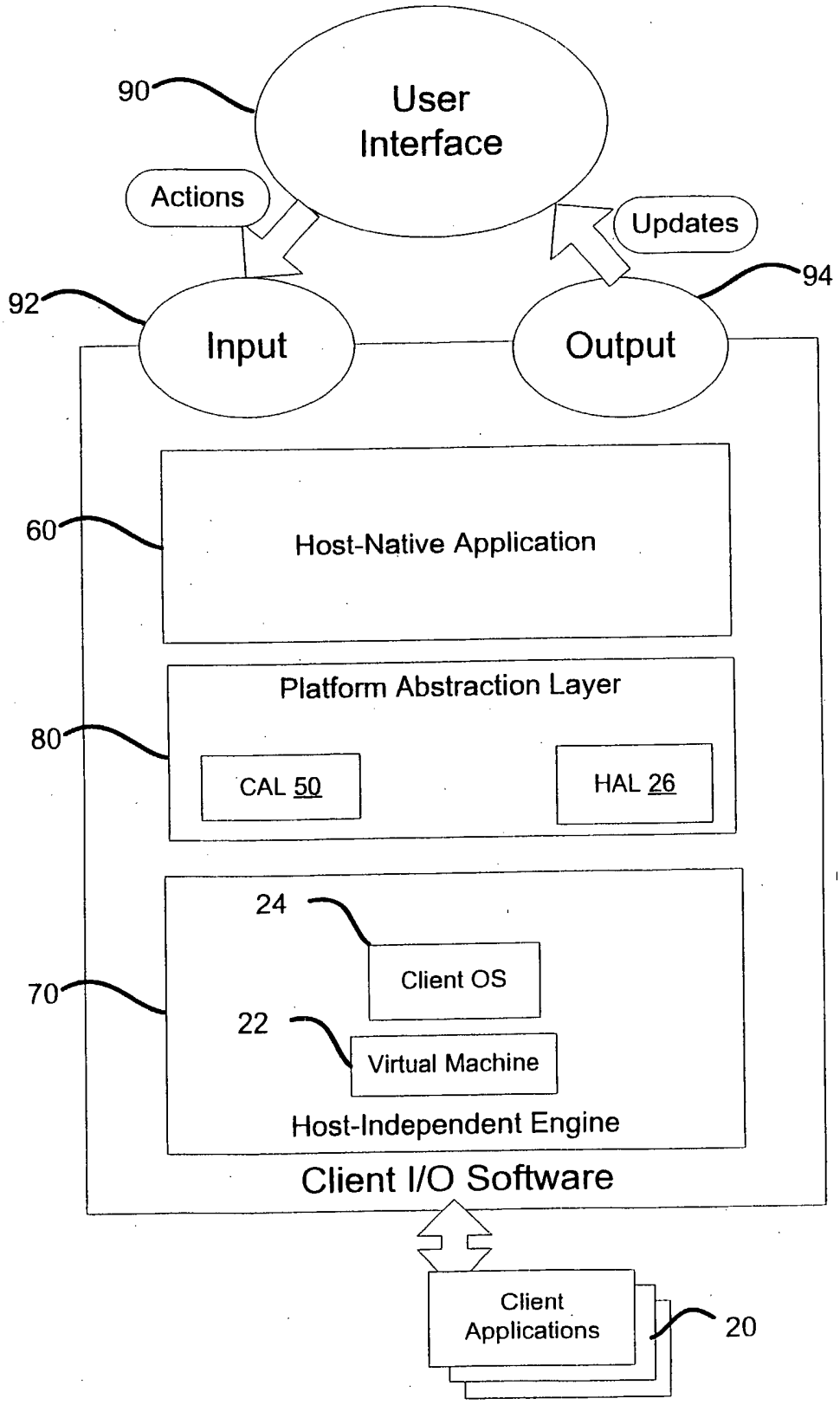
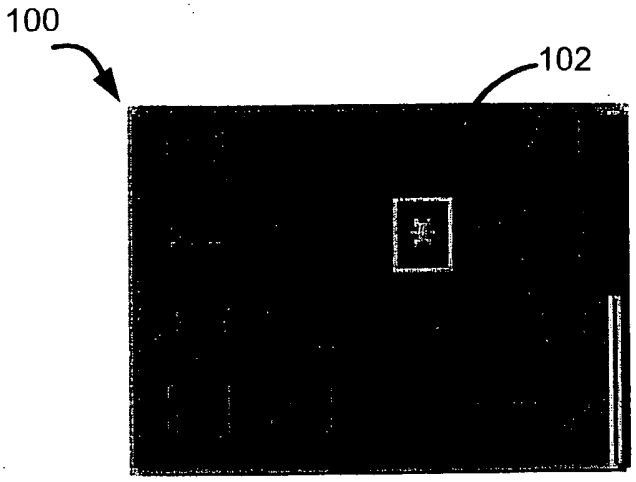
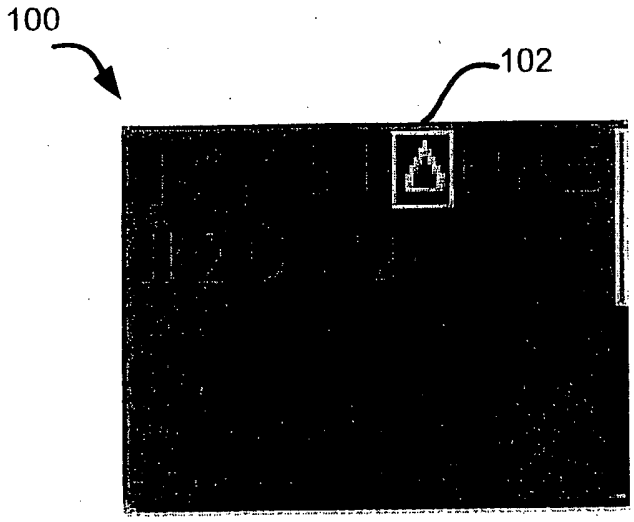

90
User Interface
Actions
Updates
92
94
Input
Output
60
Host-Native Application
Platform Abstraction Layer
80
CAL 50
HAL 26
24
Client OS
70
22
Virtual Machine
Host-Independent Engine
Client I/O Software
Client Applications
20

FIG. 5



HOST SYMBOL 1

FIG. 6A



HOST SYMBOL 2

FIG. 6B

FIG. 6

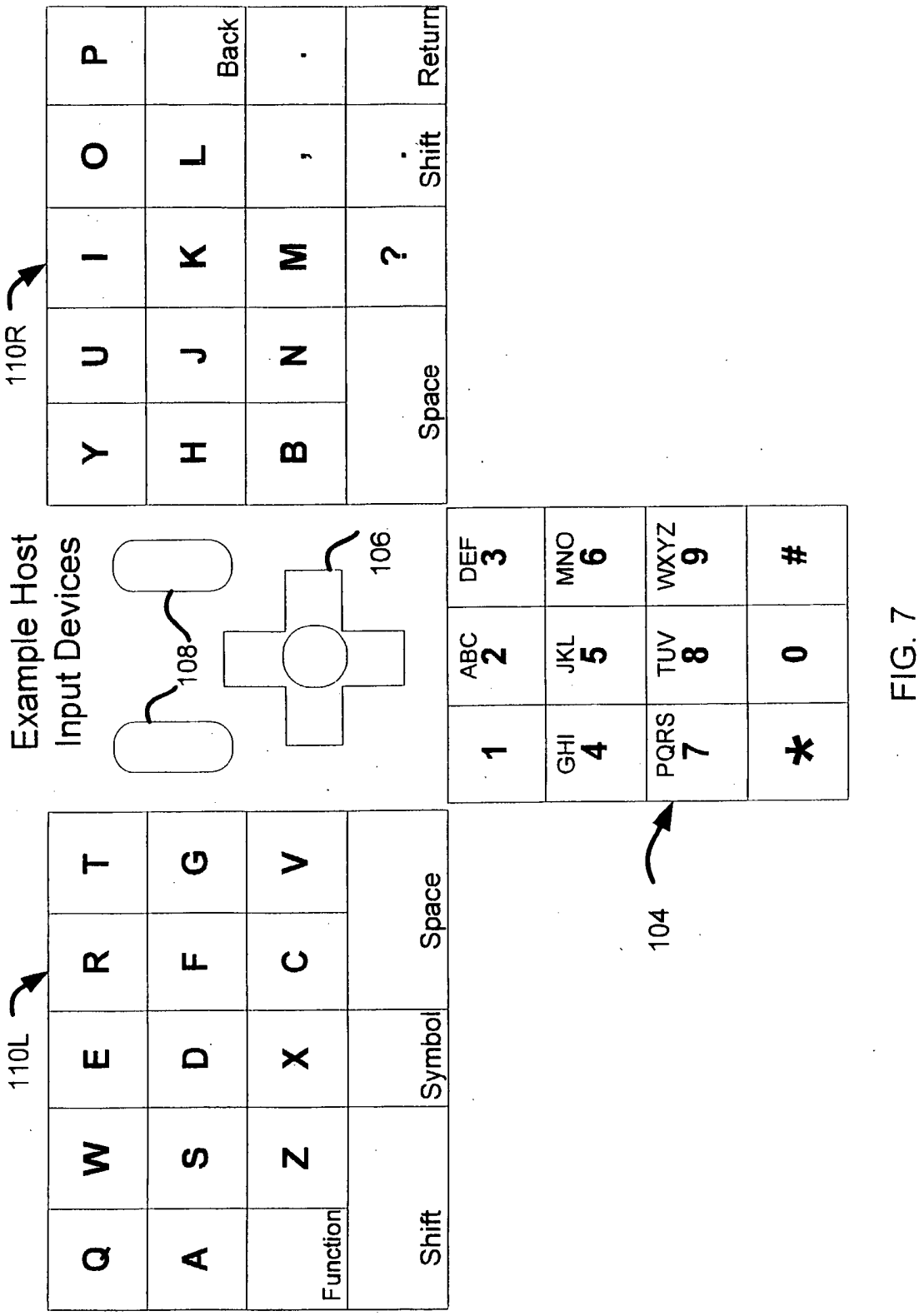
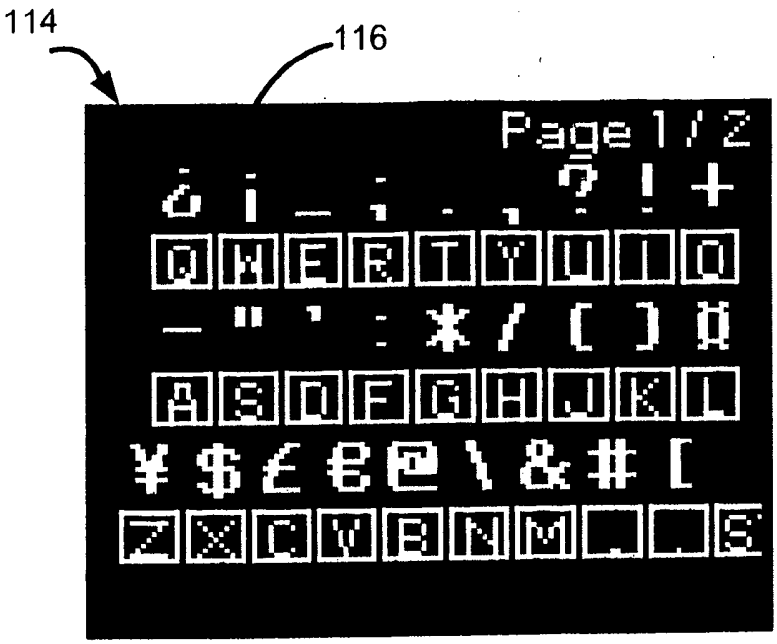
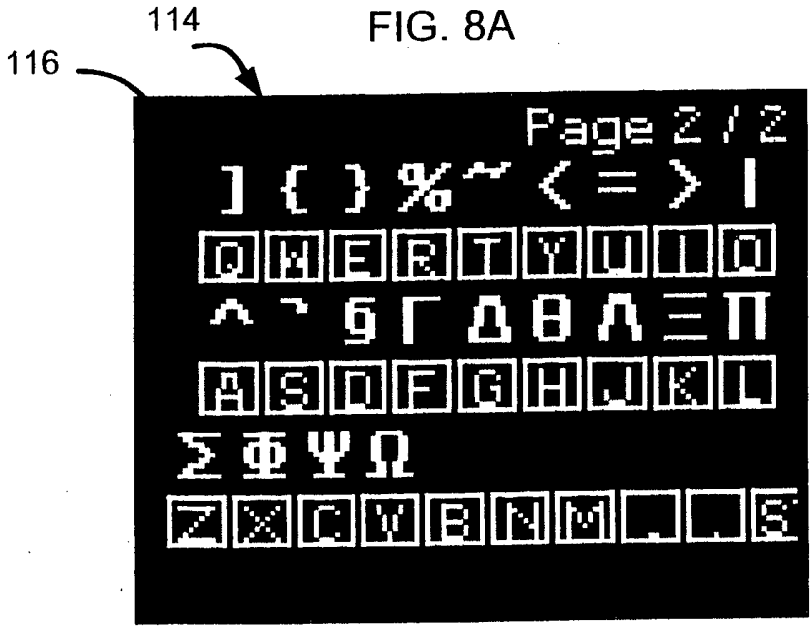

Example Host Input Devices
110L
110R
104
108
106
Q W E R T
A S D F G
Z X C V
Function
Shift
Symbol
Space
Y U I O P
H J K L
B N M , .
Space
? .
Shift
Back
Return
1
ABC 2
DEF 3
GHI 4
JKL 5
MNO 6
PQRS 7
TUV 8
WXYZ 9
* 0 #

FIG. 7

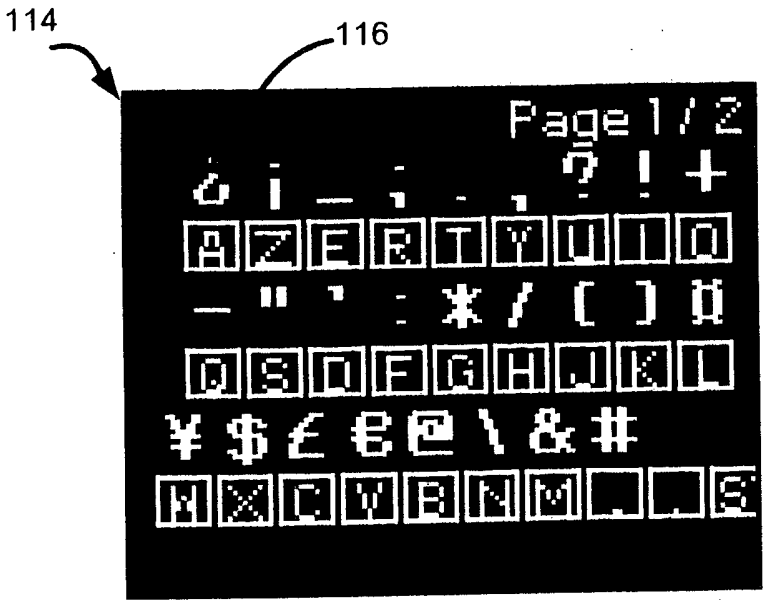


QWERTY
FIG. 8A

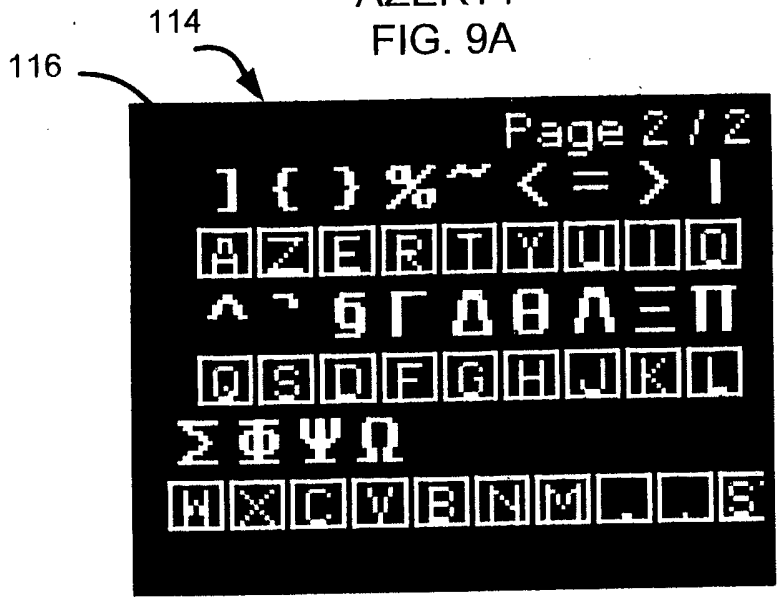


QWERTY
FIG. 8B

FIG. 8



AZERTY
FIG. 9A



AZERTY
FIG. 9B

FIG. 9

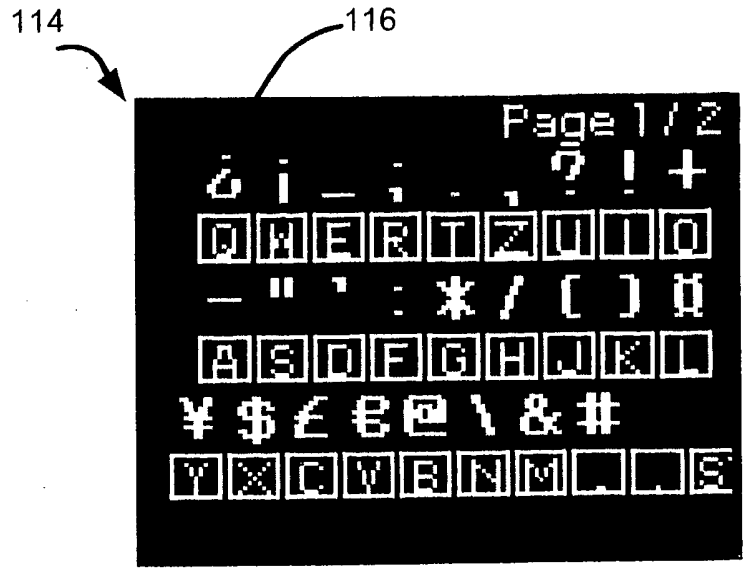


FIG. 10A

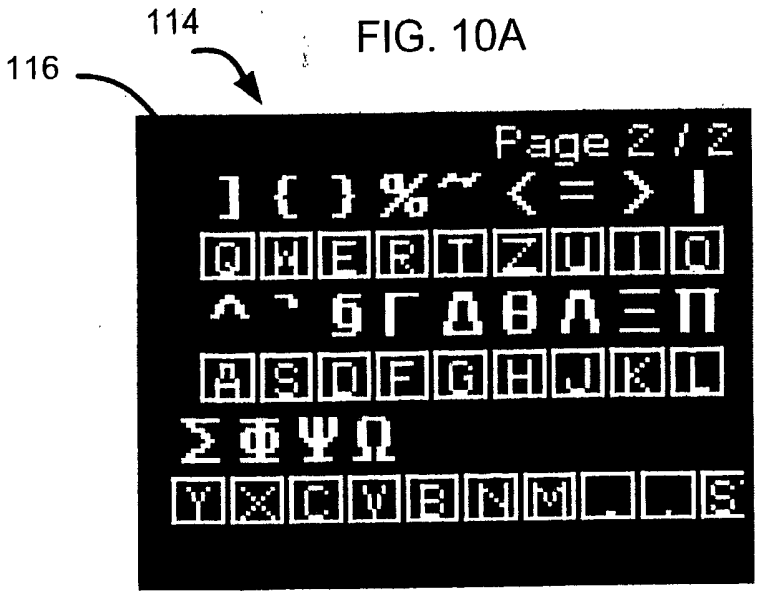


FIG. 10B

FIG. 10

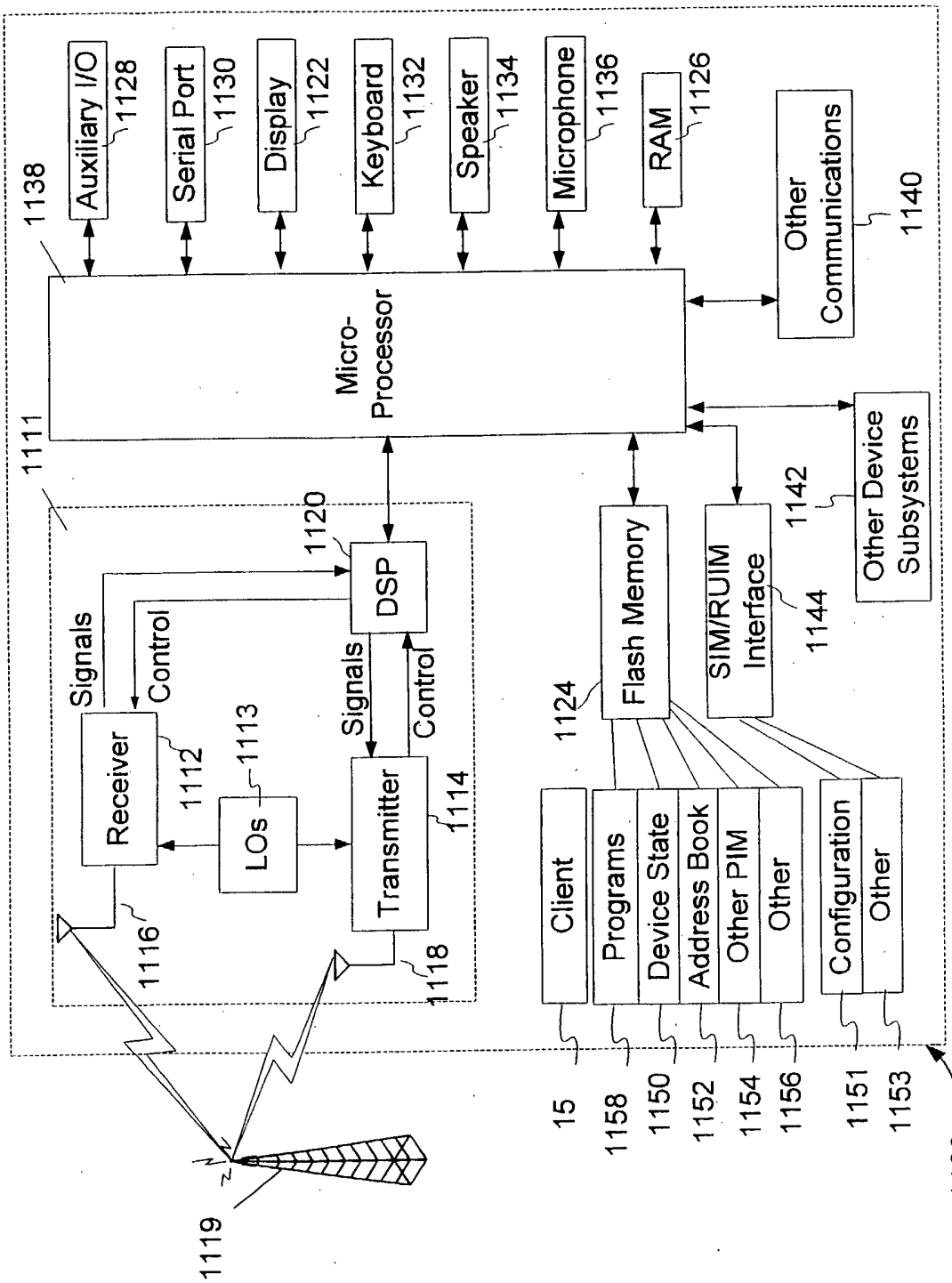

Auxiliary I/O
1128
Serial Port
1130
Display
1122
Keyboard
1132
Speaker
1134
Microphone
1136
RAM
1126
Other Communications
1140
1138
Micro-Processor
1111
Other Device Subsystems
1142
SIM/RUIM Interface
1144
Flash Memory
1124
1120
DSP
Signals
Control
Signals
Control
Receiver
1112
1113
LOs
Transmitter
1114
1116
1118
1119
Client
Programs
Device State
Address Book
Other PIM
Other
Configuration
Other
15
1158
1150
1152
1154
1156
1151
1153
1100

FIG. 11

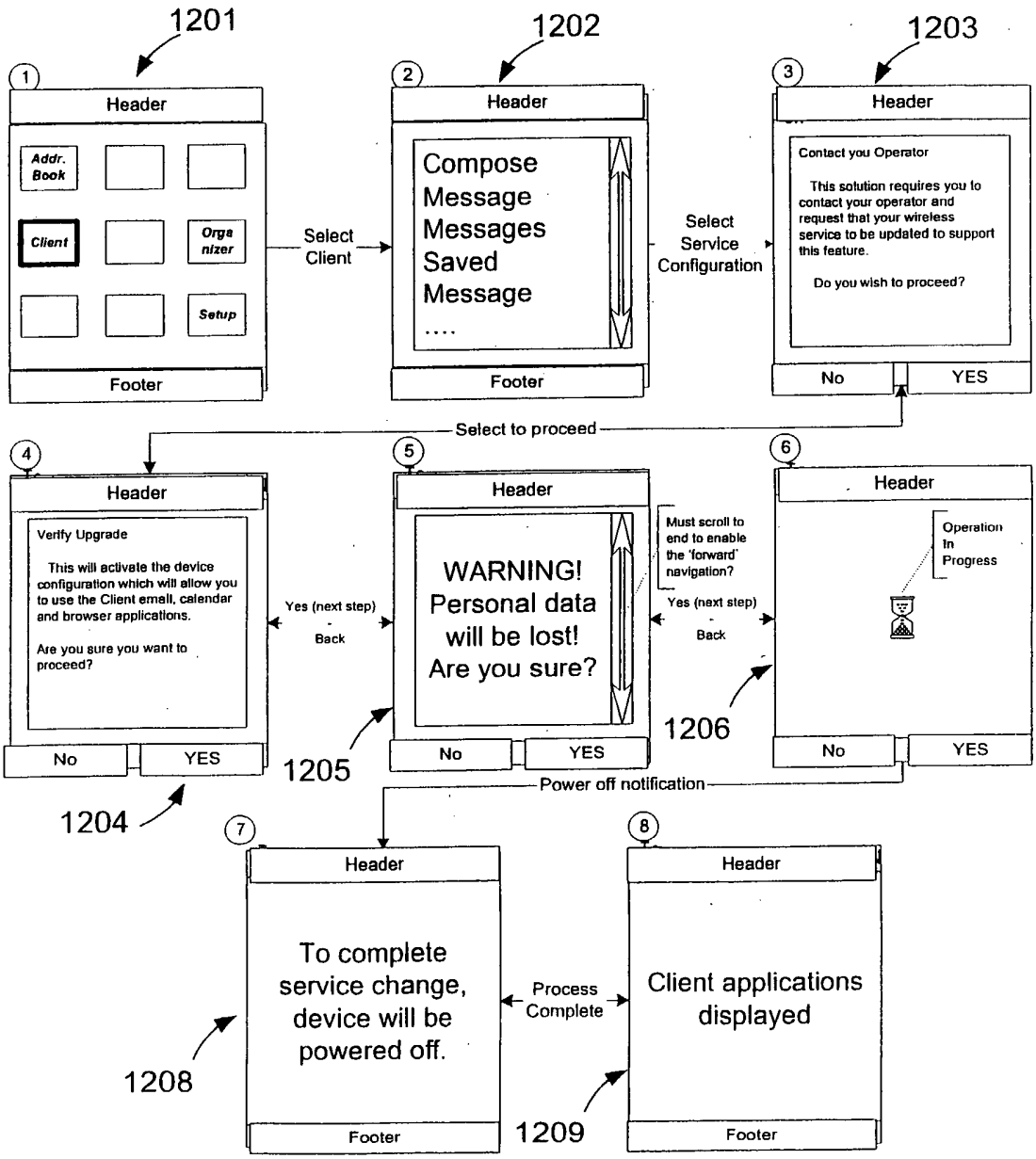

1201
1202
1203
1
Header
Addr. Book
Client
Orga nizer
Setup
Footer
Select Client
2
Header
Compose Message
Messages Saved Message
....
Footer
Select Service Configuration
3
Header
Contact you Operator
This solution requires you to contact your operator and request that your wireless service to be updated to support this feature.
Do you wish to proceed?
No
YES
Select to proceed
4
Header
Verify Upgrade
This will activate the device configuration which will allow you to use the Client email, calendar and browser applications.
Are you sure you want to proceed?
No
YES
1204
Yes (next step)
Back
5
Header
WARNING! Personal data will be lost! Are you sure?
No
YES
1205
Must scroll to end to enable the 'forward' navigation?
Yes (next step)
Back
6
Header
Operation In Progress
No
YES
1206
Power off notification
7
Header
To complete service change, device will be powered off.
Footer
1208
Process Complete
8
Header
Client applications displayed
Footer
1209

FIG. 12

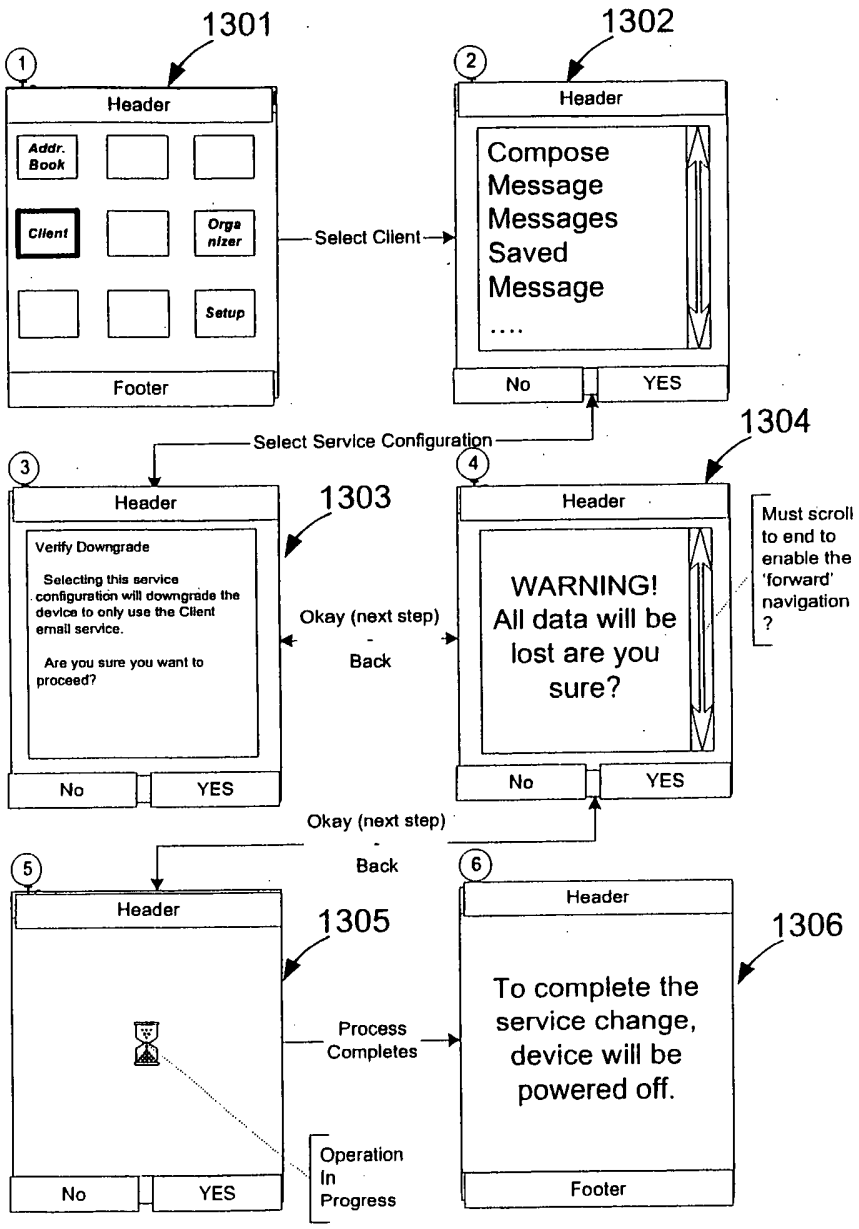

1301
1302
1
2
Header
Addr. Book
Client
Orga nizer
Setup
Footer
Select Client
Header
Compose Message
Messages Saved
Message
....
No
YES
Select Service Configuration
1304
3
4
Header
1303
Verify Downgrade
Selecting this service configuration will downgrade the device to only use the Client email service.
Are you sure you want to proceed?
No
YES
Okay (next step)
Back
Header
WARNING! All data will be lost are you sure?
No
YES
Must scroll to end to enable the 'forward' navigation ?
Okay (next step)
Back
5
6
Header
1305
Process Completes
Operation In Progress
No
YES
Header
1306
To complete the service change, device will be powered off.
Footer

FIG. 13

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2005/001176

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7): G06F 9/44, G06F 3/02

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(7): G06F

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

Electronic database(s) consulted during the international search (name of database(s) and, where practicable, search terms used)
Delphion, IEEE Xplore, Canadian Patent Database & keywords: integrate client/host; divided architecture; abstraction; host independent (engine); host application; host device; host native application; client application; user interface; data enabled cellular telephone; translate host call; translate client call; isolate/decouple host/client; translate call; host/client function call; thin-client hosting; client/host symbol table; grid; cursor; inputting symbols; screen image sharing; synchronize/coordinate/share, device settings; coordinating input, output; keyboard; qwerty.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	US 20040044728 A1 (*Gargi*) 4 March 2004 (04-03-2004) abstract, [0003], [0004], [0022], [0026], [0038], [0039], [0044], claim 2, Fig. 4	1-7, 12, 16-21, 25 8-11, 22-24
Y	US 20030191799 A1 (*Araujo et al.*) 9 October 2003 (09-10-2003) [0017], [0062], [0064], [0066]	1-7, 12, 16-21, 25
Y A	US 5241625 (*Epard et al.*) 31 August 1993 (31-08-1993) col. 3 lines 5-26, col. 24 lines 32-38; col.46 lines 54, 55; col. 50 line 33; col. 61 line 64 to col. 62 line 9	5-7, 12, 20, 21, 25 13-15

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

[X] See patent family annex.

* Special categories of cited documents :

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
4 October 2005 (04-10-2005)	8 November 2005 (08-11-2005)

Name and mailing address of the ISA/CA	Authorized officer
Canadian Intellectual Property Office Place du Portage I, C114 - 1st Floor, Box PCT 50 Victoria Street Gatineau, Quebec K1A 0C9 Facsimile No.: 001(819)953-2476	Cristian S. Popa (819) 997-2299

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2005/001176

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of the first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons :

1. [ ] Claim Nos. :
because they relate to subject matter not required to be searched by this Authority, namely :

2. [ ] Claim Nos. :
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically :

3. [ ] Claim Nos. :
because they are dependant claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows :

1. Claims 1-12, 16-25: *System and method for integrating a client with a host device by using an abstraction layer configured to translate between the client and the host*

2. Claims 13-15: *System for inputting symbols to a client from a host by using a symbol table*

1. [X] As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. [ ] As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. [ ] As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claim Nos. :

4. [ ] No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claim Nos. :

Remark on Protest

[ ] The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

[ ] The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

[ ] No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/CA2005/001176

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US2004044728	04-03-2004	US2004044728 A1	04-03-2004
US2003191799	09-10-2003	US2003191799 A1	09-10-2003
US5241625	31-08-1993	US5241625 A	31-08-1993