SYSTEM AND METHOD FOR INTEGRATING CONTINUOUS SYNCHRONIZATION ON A HOST HANDHELD DEVICE

TECHNICAL FIELD OF THE APPLICATION

[0001] The present application relates, in general, to the integration of a client into a host handheld device and, in particular, to a system and method for integrating a continuous synchronization client on a host handheld device which identifies changes in a host database (58) within the host handheld device and updates a client database in response to the identified changes.

BACKGROUND

[0002] Many handheld devices may be synchronized with a desktop computer, that is, information is transferred between the desktop computer and the handheld device to update each. This process typically occurs by coupling the handheld device to the desktop computer via a wired connection and executing synchronization software. Some of these handheld devices may also be capable of wireless synchronization with the desktop computer by establishing wireless communication therebetween and executing the synchronization software. It has been found, however, that these handheld devices are not capable of continuous synchronization to update the information stored on the desktop computer when changes are made to the information stored on the handheld device. Therefore, a need has arisen for a system and method for implementing continuous synchronization from the handheld device to the desktop computer.

SUMMARY

[0003] The present application discloses a handheld held device for use in a wireless communications network enabled for continuous synchronization with the user's desktop computer. The handheld device including a host database having first and second API associated therewith, a client database and a notification handler module. The notification handler module sends a request for notification of a change in the host database to the first API. After a change has occurred in the host database, the notification handler module receives notification of the change in the host database from the first API. The notification handler module then requests the contents of the host database from a second API and requests the contents from a client database to compare these contents such that the change in the host database may be identified. The notification handler module then sends a command to update the client database based upon the change in the host database. Once the client database has been updated, the handheld device may communicate with the desktop computer via the wireless communication network to update the desktop computer database based upon the updated client database.

[0004] In one embodiment, the record in which the change in the host database has been made is identified by the

first API. In this embodiment, the notification handler module requests the contents of this record in the host database from the second API and requests the contents related to the record in which the change was made in the host database from the client database such that the contents of the records can be compared and the record in the client database can be updated.

[0005] In one embodiment, the change in the host database may be a change in a calender record. In another embodiment, the change in the host database may be a change in a contacts record. In a further embodiment, the change in the host database may be a change in a task record.

[0006] In one embodiment, the change in the host database may be deleted information in the host database. In another embodiment, the change in the host database may be created information in the host database. In a further embodiment, the change in the host database may be updated information in the host database. In yet another embodiment, the change in the host database may be a new record in the host database. In this embodiment, additional fields may be added to the new record and stored in the client database in association with the information contained in the record.

[0007] In another aspect, the present application is directed to a method for updating a client database within a

handheld device to enable continuous synchronization. The method includes sending a request for notification of a change in a host database to a first API, receiving the notification of the change in the host database after the change has occurred in the host database, requesting contents of the host database from a second API, requesting contents from a client database, comparing the contents of the host database and the client database, identifying the change in the host database and updating the client database based upon the change in the host database.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] For a more complete understanding of the features and advantages of the present systems and methods, reference is now made to the detailed description along with the accompanying figures in which corresponding numerals in the different figures refer to corresponding parts and in which:

[0009] Figure 1 depicts an exemplary network environment including a host handheld device wherein an embodiment of the present application may be practiced;

[0010] Figure 2 depicts an exemplary software architecture of a host handheld device within a wireless communication environment wherein an embodiment of the present patent application may be practiced;

[0011] Figure 3 is a flowchart of an embodiment of integrating a continuous synchronization client on a host handheld device; and

[0012] Figure 4 is a flowchart of another embodiment of integrating a continuous synchronization client on a host handheld device.

DETAILED DESCRIPTION

[0013] While the operation of various embodiments of a handheld device is discussed in detail below, it should be appreciated that the present application provides many applicable inventive concepts which can be embodied in a wide variety of specific contexts. The specific embodiments discussed herein are merely illustrative of specific ways to use the handheld device, and do not delimit the scope of the present application.

[0014] Referring initially to Figure 1, depicted therein is an exemplary network environment 10 including a wireless packet data service network 12 wherein an embodiment of the present patent application may be practiced. An enterprise network 14, which may be a packet-switched network, can include one or more geographic sites and be organized as a local area network (LAN), wide area network (WAN), metropolitan area network (MAN) or the like for serving a plurality of corporate users. A number of application servers 16-1 through 16-N disposed as part of the enterprise network 14 are operable to provide or effectuate a host of internal and external services such as email, video mail, Internet access, corporate data access, messaging, calendaring and scheduling, information management and the like. Accordingly, a diverse array of personal information

appliances such as desktop computers, laptop computers, palmtop computers and the like, which are generically and collectively referred to herein and depicted as desktop computer 18, may be operably networked to one or more of the application servers 16-i, where i = 1, 2,...,N, with respect to the services supported in enterprise network 14.

[0015] Additionally, a remote services server 20 may be interfaced with enterprise network 14 for enabling a corporate user to access or effectuate any of the services from a remote location using a suitable handheld device 22. A secure communication link with end-to-end encryption may be established that is mediated through an external IP network, i.e., a public packet-switched network such as the Internet 24, as well as the wireless packet data service network 12 operable with handheld device 22 via suitable wireless network infrastructure that includes a base station 26. In one embodiment, a trusted relay network 28 may be disposed between Internet 24 and the infrastructure of wireless packet data service network 12. By way of example, handheld device 22 may be a data-enabled handheld device capable of receiving and sending voice, text messages, web browsing, interfacing with corporate application servers and the like.

[0016] For purposes of the present patent application, the wireless packet data service network 12 may be implemented in any known or heretofore unknown mobile communications technologies and network protocols, as long as a packet-switched data service is available therein for transmitting packetized information. For instance, the wireless packet data service network 12 may be comprised of a General Packet Radio Service (GPRS) network that provides a packet radio access for mobile devices using the cellular infrastructure of a Global System for Mobile Communications (GSM)-based carrier network. In other implementations, the wireless packet data service network 12 may comprise an Enhanced Data Rates for GSM Evolution (EDGE) network, an Integrated Digital Enhanced Network (IDEN), a Code Division Multiple Access (CDMA) network, any 3rd Generation (3G) network or the like.

[0017] Referring next to figure 2, an exemplary software architecture for enabling continuous synchronization of a handheld device in a wireless communication network is illustrated and generally designated 50. Handheld device 52 is in wireless communication with desktop computer 45 via networks 56 which generically and collectively represent the end-to-end connection created, for example, by base station 26, wireless packet data service network 12, trusted relay

network 28, Internet 24, remote services server 20, enterprise network 14 and application server 16 as described with reference to figure 1. Handheld device 52 is preferably a two-way wireless communication device having at least voice and data communication capabilities. Depending on the exact functionality provided, the handheld device 52 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.

[0018] Handheld device 52 preferably includes a microprocessor which controls the overall operation of handheld device 52. The microprocessor interacts with various subsystems of handheld device 52 including communication subsystems, user interface subsystems, memory subsystems and the like as well as other subsystems known to those skilled in the art. The microprocessor preferably enables execution of software applications on handheld device 52. A predetermined set of applications that control basic operations, including at least data and voice communication applications for example, will normally be installed on handheld device 52 during manufacturing. For example, handheld device 52 preferably includes a native personal information manager (PIM) application having the

ability to organize and manage information relating to, for example, calender information, task information, contacts information and the like.

[0019] Typically, the native PIM application would have the ability to send and receive data items via a wired or wireless connection to synchronized and updated with the user's corresponding data items stored or associated with desktop computer 54 when prompted by the user to synchronize. The illustrated native PIM application includes a host database 58 for storing the information, a high level API 60 for accessing the information stored in host database 58 and a low level API 62 for managing the information stored in host database 58. This configuration alone, however, does not allow for continuous synchronization with desktop computer 54 but instead, would require the user to initiate the synchronization process.

[0020] In the illustrated embodiment, handheld device 52 serves as a host device such that a client application may be added onto the host to supplement, supplant or otherwise perform functionality that the host normally would not be capable of performing. In the present case, the native configuration of handheld device 52 does not allow for continuous synchronization with desktop computer 54 such that changes made in host database 58 are not updated in

desktop computer 54 without user intervention. By adding a client synchronization application, however, this functionality is achieved. Specifically, the client synchronization application includes a client database 64 for storing a mirror image or enhanced mirror image of the information stored in the host database. The client synchronization application also includes a client API 66 for accessing and managing client database 64 and a notification handler module 68 that controls the acquisition, processing and duplication of information between host database 58 and client database 64 as explained in greater detail below.

[0021] To achieve continuous wireless synchronization, the client synchronization application interfaces with a multilayer transport stack 70 that is operable to provide a generic data transport protocol for any type of data via a reliable, secure and seamless continuous connection to a wireless packet data service network. In one implantation, transport stack 70 includes a bottom layer (Layer 1) that is operable as an interface to the wireless network's packet layer. Layer 1 handles basic service coordination within the exemplary network environment 56. For example, when handheld device 52 roams from one carrier network to another, Layer 1 verifies that the packets are relayed to

the appropriate wireless network and that any packets that are pending from the previous network are rerouted to the current network. The top layer (Layer 4) of transport stack 70 exposes various application interfaces such as the client synchronization application to the services supported on handheld device 52. The remaining two layers, Layer 2 and Layer 3, are responsible for datagram segmentation/reassembly and security, compression and routing, respectively.

[0022] Notification handler module 68 provides an interface between the native PIM application and the client synchronization application and in some embodiments partially or completely disables the native synchronization functionality. Specifically, and with reference to figure 3, the operations controlled by notification handler module 68 are described in process 100. The process begins with notification handler module 68 sending an interrupt type request to low level API 62 for notification of a change in host database 58 (block 102). The request is sent to low level API as high level API 60 is unable to receive and respond to such interrupt type requests. Once the request has been sent, if no change is made to the information stored in host database 58, no action is taken by either notification handler module 68 or low level API 62 with

respect to the request (block 106). If a change, such as a deletion, an update or an insertion to an existing record, the creation of a new record or the deletion of an entire record, is made to the information in host database 58, low level API 62 sends a notification of the change in host database 58 which is received by notification handler module 68 (block 108).

[0023] Notification handler module 68 now send a request to high level API 60 for the contents of host database 58 (block 110) and a request to client API 66 for the contents of client database 64 (block 112). It is noted that notification handler module 68 must request the contents of host database 58 from high level API 60 as low level API is unable to receive and respond to such a request. The contents of these two databases 58, 64 are then compared to one another by notification handler module 68 (block 114) such that notification handler module 68 may identify what change has been made in host database 58 (block 116). Once notification handler module 68 determines what change has occurred in host database 58, notification handler module 68 provides this information to client database 64 such that the contents of client database 64 will again mirror the contents of host database 58 (block 118). Once client database 64 has been updated, the continuous connection

between handheld device 52 and desktop computer 54 via networks 56 allows desktop database 72 to be updated using desktop API 74 (block 120).

[0024] Importantly, in addition to simply updating client database 64 with the changes made to the information in host database 58, when the change made in host database 58 is a new record, in one embodiment, notification handler module 68 adds additions fields to the new record which are stored in association with the new record in client database 64. For example, the additional fields may be useful or required to achieve the continuous synchronization capability over networks 56.

[0025] Another implementation of the client synchronization application on a host handheld device will now be described with reference to figure 4 and process 200. The process begins with notification handler module 68 sending an interrupt type request to low level API 62 for notification of a change in host database 58 (block 202). As before, the request is sent to low level API as high level API 60 is unable to receive and respond to such interrupt type requests. Once the request has been sent, if no change is made to the information stored in host database 58, no action is taken by either notification handler module 68 or low level API 62 with respect to the request (block

206). If a change, such as a deletion, an update or an insertion to an existing record, the creation of a new record or the deletion of an entire record, is made to the information in host database 58, low level API 62 sends a notification of the change in host database 58 which includes the record ID of the record in which the change in made. The notification is received by notification handler module 68 (block 208).

[0026] Notification handler module 68 now send a request to high level API 60 for the contents of the changed record of host database 58 (block 210) and a request to client API 66 for the contents of any corresponding record in client database 64 (block 212). As before, notification handler module 68 must request the contents of the changed record from high level API 60 as low level API is unable to receive and respond to such a request. The contents of these two records are then compared to one another by notification handler module 68 (block 214) such that notification handler module 68 may identify what change has been made in the record (block 216). Once notification handler module 68 determines what change has occurred, notification handler module 68 provides this information to client database 64 such that the contents of the effected record in client database 64 are updated and client database 64 again mirrors

the contents of host database 58 (block 218). Once client database 64 has been updated, the continuous connection between handheld device 52 and desktop computer 54 via networks 56 allows desktop database 72 to be updated using desktop API 74 (block 220).

[0027] While this application has described a mobile device with reference to illustrative embodiments, this description is not intended to be construed in a limiting sense. Various modifications and combinations of the illustrative embodiments as well as other embodiments of the mobile device, will be apparent to persons skilled in the art upon reference to the description. It is, therefore, intended that the appended claims encompass any such modifications or embodiments.

What is claimed is:

1. (Currently Amended) A method for updating a client database (64) within a host handheld device (52) to enable continuous synchronization, the host handheld device (52) having the client database (64), a host database (58), a first API (62) and a second API (60) resident thereon, the method comprising:

sending a request for notification of a change in the host database (58) to the first API (62);

receiving the notification of the change in the host database (58) after the change has occurred in the host database (58);

requesting contents of the host database (58) from ~~a~~ the second API (60);

requesting contents from the client database (64);

comparing the contents of the host database (58) and the contents of the client database (64);

identifying the change in the host database (58); and

updating the client database (64) based upon the change in the host database (58).

2. The method as recited in claim 1 wherein receiving the notification of the change in the host database (58) after the change has occurred in the host database (58)

further comprises identifying a record in which the change occurred.

3. The method as recited in claim 2 wherein requesting contents of the host database (58) from a second API (60) further comprises requesting contents of the record in which the change occurred.

4. The method as recited in claim 3 wherein requesting contents from a client database (64) further comprises requesting contents related to the record in which the change occurred in the host database (58).

5. The method as recited in claim 1 further comprising wireless updating a desktop computer database (72) based upon the updated client database (64).

6. The method as recited in claim 1 wherein sending a request for notification of a change in a host database (58) to a first API (62) further comprises sending a request for notification of a change in a host database (58) to a low level API (62) and wherein requesting contents of the host database (58) from a second API (60) further comprises

requesting contents of the host database (58) from a high level API (60).

7. The method as recited in claim 1 wherein receiving the notification of the change in the host database (58) after the change has occurred in the host database (58) further comprises receiving the notification of a new record in the host database (58).

8. The method as recited in claim 7 wherein updating the client database (64) based upon the change in the host database (58) further comprises adding the new record to the client database (64) and adding additional fields in association with the new record in the client database (64).

9. A network system (50) comprising:

a desktop computer (54) having a database (72);

a wireless communication network (56) communicably linked to the desktop computer (54); and

a handheld device (52) communicably linked to the wireless network (56), the handheld device (52) including a host database (58) having first and second API (62, 60) associated therewith, a client database (64) and a notification handler (68), wherein the notification handler (68) sends a request for notification of a change in the host database (58) to the first API (62), receives the notification of the change in the host database (58) after the change has occurred in the host database (58), requests contents of the host database (58) from the second API (60), requests contents from the client database (64), compares the contents of the host database (58) and the contents of the client database (64), identifies the change in the host database (58) and sends a command to update the client database (64) based upon the change in the host database (58), the handheld device (52) communicates with the desktop computer (54) via the wireless communication network (56) to update the desktop computer database (72) based upon the updated client database (64).

10. The network system (50) as recited in claim 9 wherein the notification of the change in the host database (58) after the change has occurred in the host database (58) further comprises identification of a record in which the change occurred.

11. The network system (50) as recited in claim 10 wherein the contents of the host database (58) further comprise contents of the record in which the change occurred.

12. The network system (50) as recited in claim 11 wherein the contents from a client database (64) further comprises contents related to the record in which the change occurred in the host database (58).

13. The network system (50) as recited in claim 9 wherein the notification of the change in the host database (58) after the change has occurred in the host database (58) further comprises a notification of a new record in the host database (58).

14. The network system (50) as recited in claim 9 wherein the command to update the client database (64) based upon the change in the host database (58) further comprises a command to add the new record to the client database (64) and add additional fields in association with the new record in the client database (64).

15. A handheld device (52) comprising:

a host database (58) having first and second API (62, 60) associated therewith;

a client database (64); and

a notification handler (68), wherein the notification handler (68) sends a request for notification of a change in the host database (58) to the first API (62), receives the notification of the change in the host database (58) after the change has occurred in the host database (58), requests contents of the host database (58) from the second API (60), requests contents from the client database (64), compares the contents of the host database (58) and the contents of the client database (64), identifies the change in the host database (58) and sends a command to update the client database (64) based upon the change in the host database (58).

16. The handheld device (52) as recited in claim 15 wherein the notification of the change in the host database (58) after the change has occurred in the host database (58) further comprises identification of a record in which the change occurred.

17. The handheld device (52) as recited in claim 16 wherein the contents of the host database (58) further comprise contents of the record in which the change occurred.

18. The handheld device (52) as recited in claim 17 wherein the contents from a client database (64) further comprises contents related to the record in which the change occurred in the host database (58).

19. The handheld device (52) as recited in claim 15 wherein the notification of the change in the host database (58) after the change has occurred in the host database (58) further comprises a notification of a new record in the host database (58).

20. The handheld device (52) as recited in claim 19 wherein the command to update the client database (64) based upon the change in the host database (58) further comprises a command to add the new record to the client database (64) and add additional fields in association with the new record in the client database (64).