

(12) PATENT
(19) AUSTRALIAN PATENT OFFICE

(11) Application No. **AU 199896840 B2**
(10) Patent No. **753377**

(54) Title
Method and apparatus for updating databases

(51)⁶ International Patent Classification(s)
G06F 017/30

(21) Application No: **199896840** (22) Application Date: **1998 .10 .14**

(87) WIPO No: **WO99/23580**

(30) Priority Data

(31) Number	(32) Date	(33) Country
08/965002	1997 .11 .05	US

(43) Publication Date : **1999 .05 .24**

(43) Publication Journal Date : **1999 .07 .22**

(44) Accepted Journal Date : **2002 .10 .17**

(71) Applicant(s)
Ericsson Inc.

(72) Inventor(s)
Billy Gayle Moon; Tripurasundari Ramesh; vikas Jain; Brian Bankler

(74) Agent/Attorney
WATERMARK PATENT and TRADEMARK ATTORNEYS, Locked Bag 5, HAWTHORN VIC 3122

(56) Related Art
EP 550368
WO 97/15017
WO 96/37853

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

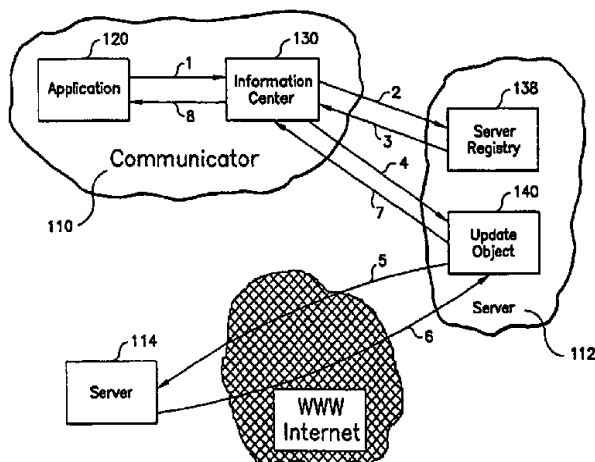


INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/30		(11) International Publication Number: WO 99/23580
A1		(43) International Publication Date: 14 May 1999 (14.05.99)
(21) International Application Number: PCT/US98/20942		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
(22) International Filing Date: 14 October 1998 (14.10.98)		
(30) Priority Data: 08/965,002 5 November 1997 (05.11.97) US		
(71) Applicant: ERICSSON INC. [US/US]; 7001 Development Drive, P.O. Box 13969, Research Triangle Park, NC 27709 (US).		
(72) Inventors: MOON, Billy, Gayle; 305 Parkbranch Lane, Apex, NC 27502 (US). RAMESH, Tripurasundari; 403 Danton Drive, Cary, NC 27511 (US). JAIN, Vikas; 5801-13 Tattersall Drive, Durham, NC 27713 (US). BANKLER, Brian; 910 Balmoral Drive, Cary, NC 27511 (US).		
(74) Agents: GRUDZIECKI, Ronald, L. et al.; Burns, Doane, Swecker & Mathis, L.L.P., P.O. Box 1404, Alexandria, VA 22313-1404 (US).		Published With international search report.

IP AUSTRALIA
24 MAY 1999
RECEIVED

(54) Title: METHOD AND APPARATUS FOR UPDATING DATABASES



(57) Abstract

A method and apparatus for updating databases is disclosed. A database is updated transparently using an IDL (Interface Descriptive Language) to interface with websites on the WWW (World Wide Web). Multi-media information is parsed such that the relevant information can be integrated into an application or application framework for a user's use. The update can be performed when the device moves to a new location or a particular event occurs and can be used to re-configure an entire system or just a particular application. In some embodiments, the device checks to see if a particular database is already available before obtaining a new database. The device can purge unused databases to provide space for new or additional databases.

METHOD AND APPARATUS FOR UPDATING DATABASES

BACKGROUND

The present invention relates to updating databases, and more particularly, to
5 transparently updating a database with information available on a computer network or server.

Computers (or computer-related devices) generally use hardware, a power source, stored information, programmable memory, and user input to assist computer users in a countless number of tasks. Unfortunately a computer can only store a limited amount of
10 information. For this reason, computer users often connect to a network or server to obtain additional information. A modem or a similar telecommunication device allows computers to transmit and receive information from other computers. By connecting to the Internet, a computer has access to networks and servers throughout the world.

Computers have become increasingly useful as they have become smaller in size.
15 The traditional desktop computer has been replaced by smaller computers, which are sometimes referred to as laptops, palmtops, or Personal Digital Assistant (PDAs). With the development of wireless technologies, these portable computers can send and receive information from anywhere in the world. These portable computers are particularly useful for the traveling business person (TBP) and are often used by the TBP in a number
20 of different locations.

The information used by a single computer can be referred to generally as a collection of databases. Some of these databases are permanently set in the hardware and are used for the basic operation of the device. Other databases are a permanent part of the software and are used for configuring that particular software. Still other databases are a
25 temporary part of the software. These databases can be stored separate from the software and are usually used for a particular purpose, such as storing numbers used in numerical functions. These temporary databases can be downloaded from a network or server, or generally speaking, from the Internet. While most computers are able to download information, Applicants are unaware of any existing products or methods for updating a
30 database transparently, *i.e.*, without significant user interaction. For example, Applicants

COPY

-2-

are unaware of any products or methods for interfacing with the Internet and obtaining information to update a local database without user interaction.

Conventional devices require the user to manually download information or stay on-line to use a selected Internet website. This usually means that the user has to close or
5 discontinue using a particular application while downloading the information. In most cases, the user has to generate a query and/or find a particular web site. If the user stays on-line, the user usually has to pay additional online, cellular, or long distance charges. Downloaded information is usually stored in a file on the user's computer that has to be opened and manipulated by the user. If the user wants to use different information or
10 ensure that particular information is up to date, he has to repeat this process. In addition, the user has no way of knowing when a particular database stored on his or her computer was last updated, and as a result, the user might make calculations based on out-of-date information. Moreover, if the product is shipped to a different country and the correct language is not loaded, the user might not even have the necessary instructions to operate
15 the device.

The Kerrigan et al. patent, U.S. Patent No. 5,404,488 discloses an interface for coupling data from a realtime data feed into an application. The interface includes a means for caching the most recent data values received from data feeds, a means for retrieving cached data in response to a request for updates and a means for sending at
20 least some of the cached data values in response to a request for updates. The Kerrigan et al. patent does not disclose a system for updating a database comprising a request generator that generates an indirect request based upon an occurrence of an event other than a direct user request. The Kerrigan et al. patent discloses an interface which permits a realtime data feed from a direct request for realtime data.

25 The IBM Corp. patent, EP 0550 368 A, discloses a method and system for accessing a database using an interface format including a database device driver interface. The system transmits a dynamic access request from the user application to the database device driver. The database driver produces a dynamic access command for transmission of the dynamic access request and receives a response from the



SUBSTITUTE SHEET

AMENDED SHEET

database system. The IBM Corp. patent does not disclose a system for updating a database including a request generator that generates an indirect request based upon an occurrence of an event other than a direct user request. The IBM Corp. patent discloses a user application sending a request to a database driver

5 without any discussion of insignificant user interaction to request data.

The Kangaroo Inc. patent, WO 96 37853 A, discloses an on-line database updating network system and method using object-oriented programming of discrete blocks, with each block being separately accessible, modifiable and replaceable. The Kangaroo Inc. patent discloses a system in which such

10 updating originates with a user inputting identification data to the user terminal using a user input device. The Kangaroo Inc. patent does not disclose a system for updating a database including a request generator that generates an indirect request based upon an occurrence of an event other than a direct user request.

Any discussion of documents, devices, acts or knowledge in this

15 specification is included to explain the context of the invention. It should not be taken as an admission that any of the material formed part of the prior art base or the common general knowledge in the relevant art in Australia on or before the priority date of the claims herein.

SUMMARY

20 In accordance with a first aspect of the invention there is provided a system for updating a database including:

a request generator that generates an indirect request based upon an occurrence of an event other than a direct user request;

a query generator that generates a query responsive to the indirect request

25 and provides the query to a server interface upon receipt of the indirect request, wherein the server interfaces with a server to receive information responsive to the query and the server interface parses extraneous information if received from the server which is responsive to the query but not responsive to the indirect request;

30 a receiver for receiving information responsive to the indirect request obtained by the server interface; and

a database up-dater that updates the database with the information received by the receiver.



In accordance with a second aspect of the invention there is provided a method for updating a database including the steps of:

generating an indirect request for an update based upon an occurrence of an event other than a direct request;

5 providing a query to a server interface responsive to the indirect request, wherein the service interface is a script file describing an Interface Descriptive Language;

obtaining information from a server responsive to the query;

receiving information from the server at the server interface;

10 parsing information which is non-responsive to the indirect request if received; and

updating the database with the information which is responsive to the indirect request.

In accordance with a third aspect of the invention there is provided a
15 device including:

a request generator, the request generator operable to generate an indirect request to update the database based upon an occurrence of an event other than a direct request;

20 a query generator, the query generator operable to generate a query responsive to the indirect request to update the database;

a query transmitter, the query transmitter operable to transmit the query to a network of interconnected computers;

25 an information receiver, the information receiver operable to receive information responsive to the query that is extracted from one of the interconnected computers, wherein the information which is non-responsive to the request is parsed; and

30 a database up-dater, the database up-dater configured to use the information to update the database.

Exemplary embodiments of the present invention allow a user of a particular application to readily benefit from information available on a network or the Internet. According to the present invention, a database is updated transparently, i.e., it is not necessary for the user to exit the application, monitor the updating, or perform any related steps. In an exemplary embodiment, an IDL



(Interface Descriptive Language) interfaces with websites on the World Wide Web (WWW) to find and obtain relevant information for updating a currency database stored on a remote computer. The retrieved information is then integrated into an application or application framework for the user's use.

5 In some embodiments, the application indicates the time the database was last updated to assist the user in determining when to update a database. In other embodiments, the database updates itself automatically or periodically (e.g., daily, weekly, or hourly). In still other embodiments, the application updates itself when the device moves to a new location or a particular event occurs.

10 In further embodiments, the device checks to see if a particular database is already available before downloading a new database. In addition, the device can purge unused or less used databases to provide space for new databases.

In further embodiments, an application can use a downloaded database to dynamically build and update its graphic user interface (GUI), which can include
15 the menus, icons, controls, dialog boxes, or error messages.

In further embodiments, the device can discern its location and choose a particular database accordingly.

In a preferred embodiment, the device uses a hierarchical database structure to maximize the amount of information available to the user. Databases
20 are designated as global or local databases. A database can be used to re-configure an entire system or just a particular application. Furthermore, the device can recognize individual users and store preferences of each. If a particular event occurs, the device can use a local database or obtain a new database to provide a particular default setting.

25 "Comprises/comprising" when used in this specification is taken to specify the presence of stated features, integers, steps or components but does not preclude the presence or addition of one or more other features, integers, steps, components or groups thereof."

BRIEF DESCRIPTION OF THE DRAWINGS

30 The foregoing, and other objects, features, and advantages of the present invention will be more readily understood upon reading the following detailed description in conjunction with the drawings in which:

FIG. 1 illustrates a computer-related device and two servers;



FIG. 2 shows schematically how a database is updated in an exemplary embodiment of the present invention;

FIG. 3 shows schematically how a database is updated in another exemplary embodiment of the present invention;

5 FIG. 4 illustrates a graphic user interface of an exemplary embodiment of the present invention;

FIG. 5 is a diagram for interfacing with a server on the World Wide Web (WWW) to update a database in accordance with the present invention;

10 FIG. 6 illustrates the interrelationship of databases in an exemplary embodiment of the present invention; and

FIG. 7 illustrates a hierarchical architecture of databases in an exemplary embodiment of the present invention.

DETAILED DESCRIPTION

Referring now to the drawings, FIG. 1 illustrates a computer-related device
15 and two servers. The computer 10 can be a traditional desktop computer or a smaller handheld device. The computer 10 may or may not have a keyboard for entering information. The computer 10 may instead have a touch pad, a touch screen, or a stylus for entering information. In some models, the computer 10 provides both data and voice communications and can communicate freely with
20 modern cellular and satellite communication systems. An exemplary personal digital assistant which can be used as computer 10 is described in commonly assigned, U.S. Patent Application Serial no. 08/796,119, filed February 6, 1997, entitled "Switching of Analog Signals in Mobile Computing Devices," the disclosure of which is incorporated herein by reference.

25 The computer 10 can connect to a network or a server 12 either directly or indirectly. A server is a computer that provides services to other computers on a network. The computer 10 may already be connected to a network, or the computer 10 might use a modem to transmit and receive information via commercial phone lines or to other channels. This connection might be a wired
30 connection such as twisted pair, coaxial cable, or optical fiber; or this connection might be a wireless connection as a microwave, infrared, or radiowave signal. The computer 10 might be connected to a single network, multiple networks, or



4b

the Internet. When the computer 10 connects to the Internet, the computer 10 has access to networks and servers throughout the world.

If the computer 10 is not already part of a network, the computer 10 might access the Internet via an Internet Service Provider (ISP). The ISP may or may not interface with a regional network or a regional network operator. ISPs and regional networks are typically connected to highspeed backbones. Different ISPs, Regional Networks, and Backbones interconnect at National Access Points (NAPs) or other points of

10



-5-

interconnection such as Points of Presence (POPs) to form what is commonly referred to as the World Wide Web or the Internet 14.

The server 12 may or may not be provided by the same ISP or regional network accessed by the computer 10. If the server 12 is provided by a different ISP or regional
5 network then the computer 10 may have to route information through a point of interconnection such as a NAP or a POP to access the server 12. According to the current protocol, information is routed in the form of packets using a HTTP (hypertext transfer protocol) address such that the information sent and received by the computer 10 can take multiple paths to a particular server 12. According to other protocols, the information can
10 be sent and received via reserved channels.

If the server 12 receives a request from computer 10, the server 12 can access a server 16 somewhere on the WWW 14. The server 12 should be programmed to contact the correct server 16 and to interact with it correctly. The server 12 should be able to receive information from the server 16. The information received from the server 16 is
15 usually multi-media rich such that the server 12 parses out unwanted information and returns only relevant information to the computer 10.

FIG. 2 shows schematically how a computer 10 (shown in FIG. 1) updates a database in an exemplary embodiment of the present invention. The computer 10 has a request generator 20 that generates a request 21. The request 21 can be generated as the
20 result of user input, an instruction, a change in the location of the computer, or a particular event, such as the receiving or sending of an e-mail or a phone call.

The request 21 is received by an information center or a query generator 30. The query generator can be constructed wholly separate from or as part of the computer 10 (shown in Fig. 1) and/or the server 12 (shown in Fig. 1). In an exemplary embodiment,
25 the query generator 30 checks previous updates, location readings, and event readings to generate the most appropriate query 31.

The query 31 is received by a server or web interface 40. The server interface 40 interfaces with the server 50 to obtain information 51. The server interface 40 can use an IDL 41 to interface with the server 50. An exemplary IDL is described in more detail
30 below. The information 51 received from the server 50 is usually multi-media rich. The

-6-

server interface 40 can parse unwanted information and return only the relevant information 61 to an information receiver 70. The information receiver 70 uses a database up-dater 80 to update a database. The updated database can be used to configure the applications used by the computer 10 or it can be used by a particular application for a particular purpose.

In a preferred embodiment, the web interface 40 uses a script file to describe an IDL that interfaces with a particular website to extract information. An IDL can be written to do a specific task, and in this case, the IDL is written to follow an abstract map of the website and extract information. The map can describe the website in terms of a table such that the IDL knows exactly or approximately where the information is to be found. The extracted information can be turned into data objects that are manipulated by additional code. The IDL should be written to anticipate minor changes in the layout of the website. However, if the structure of the website changes drastically, the IDL should be changed accordingly.

FIG. 3 shows schematically how a database is updated according to another exemplary embodiment of the present invention. The computer 10 has a request generator 20 that generates a request 21. However, the request generator 20 passes the request to a database checker 25 that works in conjunction with the information center or query generator 30. The database checker 25 checks local databases to determine what information is already available locally. If the requested information is available locally, then the database checker 25 informs the device that it is not necessary to send a query 31 to the web or server interface 40. If the request information is not available locally, the database checker 25 informs the query generator 30 of what information is needed.

If necessary, the database checker 25 can check the programmable memory to determine if there is enough available memory for the anticipated new information 61. If there is not enough memory, then the database checker 25 can tell the computer 10 to purge non-essential databases or other files to create additional space.

-7-

As in FIG. 2, the query generator 30 provides a query 31 to a server or web interface 40. The server interface 40 interfaces with a server 50 and can parse information 51 to provide information 61. The server interface passes information 61 to a information receiver 70 and a database up-dater 80.

5 It will be appreciated by those skilled in the art that for some applications it is more important that the computer 10 (shown in Fig. 1) check to see if a database is already available on computer to before upgrading it or obtaining a new one. If, for example, the present invention is used in a particular embodiment to obtain currency conversion rates, it is not as important to check the database for local databases before
10 upgrading it. If, however, the present invention is used, for example, to provide language translation then it is more important to check the memory to see if the language database is already stored locally before obtaining a new database. Language databases do not fluctuate or change as frequently as currency conversion rate databases do, and therefore, do not need to be upgraded as often.

15 Using the example of currency conversion, techniques for updating databases according to the present invention can be implemented by a computer application for currency conversion. For example, the request 21 can be a request to update the currency conversion rates stored in a local database at computer 10. The user can push a button or click on a icon displayed on a graphical user interface (GUI) of computer 10 to generate
20 the request 21. Alternatively, the computer application could be programmed or set to update the local currency conversion database periodically or when, for example, the computer 10 enters a new country.

The query generator 30 can look to see what default conversions are currently displayed on a graphic user interface (GUI) and generate a query 31 for these conversion
25 rates. Additionally the query generator 30 can check to see if the user has recently entered a new country or made a phone call to a different country and obtain the currency rate for these countries as well. The query generator 30 sets, for example, the Uniform Resource Locator (URL) and filter parameters and sends the query 31 to the web interface 40. The web interface 40 uses the information provided by the query generator 30 to
30 connect to the correct server 50. The web interface 40 parses the information received

-8-

from the server 50 and provides the computer 10 (shown in FIG. 1) with information with which it can update the appropriate database. This updated database is then used for the currency calculation.

Using the example of language translation, the request generator 20 in FIG. 3 can be implemented in an application for language translation. An American user of the computer 10 could use the application, for example, to translate an e-mail message to Dutch before sending it to a friend in Holland; and similarly, translate the reply message back to English. Alternatively, the request generator 20 can be part of the operating memory of the computer 10 and used to configure all or some of the applications that run on it. For example, a computer 10 transported to Holland could automatically configure itself such that all instructions are in Dutch.

As shown in FIG. 3, the database checker 25 would receive the request 21 and check to see if the necessary language database is available. If the database is available, the database checker informs the application or operating system that the database is available to perform the requested translation. If the language database is unavailable then the database checker informs the query generator 30 to generate a query 31, which the server interface 40 uses to interface with a server 50. The web interface 40 then sends the necessary information 61 to information receiver 70; and, the database up-dater 80 adds the database to the local cache of language databases. If necessary, the computer 10 can delete unused databases to provide additional space. The computer then uses the new database to perform the desired translations or configurations.

The ability to download language databases on demand allows the computer to be smaller in size while remaining as useful as a larger computer. The memory is maximized because the databases are stored according to need and the amount of usage. The databases can be purged depending on their usage statistics and the last time they were used.

Moreover, language databases can be used to configure the software and build controls dynamically. The computer can display control names using a language database corresponding to the current location of the computer. For example, a computer that is made in Sweden and shipped to China, can automatically load a Chinese database

-9-

when it is first powered up so that the computer can display application controls, GUI panel messages, and/or error messages in Chinese. No longer is it necessary to individually configure manufactured computers because the computers can automatically configure themselves upon arrival at their initial destination. A computer can change its display whenever it is used in a new or different country or the user can override this feature.

FIG. 4 illustrates a graphic user interface of an exemplary embodiment of the present invention. The graphic user interface (GUI) 90 can be used in conjunction with a mouse-like device and an on-screen pointer; or the GUI 90 can be used with a stylus or a touch-sensitive screen. Depressing a button 91 can select a particular database. Other databases (not shown) are used to configure the GUI and to create its overall look and feel (e.g., colors, fonts, icons, sounds). Some databases, such as the conversions for length or weight, are static and do not need to be updated. Other conversions such as currency rates need to be updated periodically, e.g., weekly, daily, or at the start of a new trip.

As shown in FIG. 4, a GUI 90 can assist the user in converting from a group of default currencies 92 to default currencies 94. In a preferred embodiment, the application attempts to predict what conversions the user wants to make based on previous conversions, the location of the device, and/or recent phone calls made by the device. If the desired currency is not shown, then the user can select from an additional menu of currencies, and if necessary, the application will obtain the necessary currency rate and add this currency to the display panel. In addition, the GUI 90 can display a ticker tape history 96 and should have a display 98 which indicates when the currency rates were last updated.

If the user wants to update the currency rates, the user can select the update control 100. When the user selects the update control 100 or the application is updating itself automatically, a thread is launched which runs as a concurrent process in the background of the application. The user can continue to use other features of the application even while the update is taking place. In some embodiments, the update control button 100 is rendered on the GUI 90 as if it was depressed and retains this

-10-

depressed appearance until the database is updated. When the update is finished, the update control button 100 is returned to its normal appearance. The user can select the convert button 102 to perform the currency conversion calculation.

FIG. 5 is a diagram for interfacing with a server on the World Wide Web (WWW) to update a database in accordance with the present invention. In step 1, an application 120 running on a communicator 110 sends a request for an update to an information center 130. The information center 130 is a program on the communicator 110 responsible for retrieving information from a server 112.

In step 2, the information center 130 sends a query to server 112. The server 112 has a server registry 138. The server registry 138 is a program that stores the address of all the different services available. The information center 130 queries the server registry 138 to find out the address or the contact information for an update object.

In step 3, the server registry 138 returns the address of the update object to the information center 130. The update object is responsible for accessing update information from the internet. It provides the information necessary to access a particular server on the WWW, including the HTTP (hypertext transfer protocol) address, and interfaces with the server/website in order to obtain the update information. It will be appreciated by those skilled in the art that the server 112 can run an IDL to create the update object 140.

In step 4, the information center 130 requests the update object 140 for an update.

In step 5, the update object 140 interfaces with a remote server 114 to obtain the update information.

In step 6, update information flows across the update object 140.

In step 7, the update object 140 returns to the information center 130 the update information.

In step 8, the information center 130 provides an updated database to the application 120.

FIG. 6 illustrates the interrelationship of databases in a preferred embodiment of the present invention. Each database can be specified as local or global. As discussed above, a computer can obtain a database from a network or a service provider 250. The

-11-

database can be used to configure the entire system 280 or just a particular application 284. As the user uses the computer, he can override the default settings to create his own settings. He can also choose from databases located on the computer locally 290 to configure the entire system 280, his user settings 282, or a particular application 284. An application can achieve its own custom look and feel by defining its own set of properties or overriding existing properties. If the user desires, he can specify that a local change 283 be designated as a global change 281 such that the local change 283 is made in other applications. That is, the user can convert a local change into a global change.

As mentioned above, databases can be downloaded based on location changes of the device. For example, when a device owner travels to Sweden, the editor or another application can obtain a Swedish language database or the latest Swedish Geography database and travel information. This information may already be cached locally, but if it is not, it can be obtained from a service provider 250. Cached databases can be purged to create room for newer databases. A new database might be used for only a particular application 284 or it might be used to change the entire system 280.

FIG. 7 illustrates a hierarchical architecture of databases in a preferred embodiment of the present invention. A device 210 can obtain databases from a local operator 212. The local operator 212 might offer databases that are specific to a particular location, *e.g.*, maps or phone books. The device can also obtain databases from a remote network 214. The network 214 may provide databases not available from operator 212. Databases that are available in the device 210 can be specified as global databases 81, user-specific databases 82, or application-specific databases 83.

Numerous variations or modifications of the disclosed invention will be evident to those skilled in the art. While the foregoing description makes reference to particular illustrative embodiments, this patent is intended to cover all variations or modifications that do not depart from the spirit and scope of the disclosed invention.

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1. A system for updating a database including:
 - a request generator that generates an indirect request based upon an occurrence of an event other than a direct user request;
 - 5 a query generator that generates a query responsive to the indirect request and provides the query to a server interface upon receipt of the indirect request, wherein the server interface interfaces with a server to receive information responsive to the query and the server interface parses extraneous information if received from the server which is responsive to the query but not responsive to the
10 indirect request;
 - a receiver for receiving information responsive to the indirect request obtained by the server interface; and
 - a database up-dater that updates the database with the information received by the receiver.
- 15 2. The system in accordance with claim 1, wherein the request generator generates the indirect request automatically at pre-set intervals.
3. The system in accordance with claim 1, wherein information obtained from the server is converted to data objects.
4. The system in accordance with claim 1, wherein the system includes an
20 application that indicates the time the database was last updated.
5. The system in accordance with claim 1, wherein the request generator is located in a wireless, portable device.



-12A-

6. The system in accordance with claim 1, further including a server registry that returns the address of an update object to the query generator.

5 7. The system in accordance with claim 6, wherein the server registry provides instructions to interface with the server.

8. The system in accordance with claim 1, wherein the database is designated as a local database or a global database.



9. The system in accordance with claim 8, wherein a local database can be converted to a global database.

10. The system in accordance with claim 1, wherein the indirect request is automatically generated when the request generator moves to a new physical
5 location.

11. The system in accordance with claim 10, wherein the request can be manually overridden.

12. A method for updating a database including the steps of:
generating an indirect request for an update based upon an occurrence of an
10 event other than a direct request;
providing a query to a server interface responsive to the indirect request,
wherein the service interface runs a script file describing an Interface Descriptive
Language;
obtaining information from a server responsive to the query;
15 receiving information from the server at the server interface;
parsing information which is non-responsive to the indirect request if
received; and
updating the database with the information which is responsive to the
indirect request.

20 13. The method in accordance with claim 12, wherein request generator generates the indirect request automatically at pre-set intervals.



14. The method in accordance with claim 12, further including the step
of:
converting information into data objects.

15. The method in accordance with claim 12, further including the step
5 of:
designating the database as a local database or a global database.

16. The method in accordance with claim 12, wherein the request is
generated from a wireless, portable device.

17. The method in accordance with claim 12, wherein the request is
10 automatically generated when the request generator moves to a new physical
location.

18. A device including:

a request generator, the request generator operable to generate an indirect
request to update the database based upon an occurrence of an event other than a
15 direct request;

a query generator, the query generator operable to generate a query
responsive to the indirect request to update the database;

a query transmitter, the query transmitter operable to transmit the query to
a network of interconnected computers;

20 an information receiver, the information receiver operable to receive
information responsive to the query that is extracted from one of the
interconnected computers, wherein the information which is non-responsive to the
request is parsed; and



a database up-dater, the database up-dater configured to use the information to update the database.

19. The device in accordance with claim 18, wherein the information is extracted using an Interface Descriptive Language.

5 20. The device in accordance with claim 19, wherein the information is converted to data objects.

21. The device in accordance with claim 18, wherein the request generator is operable to generate a request automatically when the request generator moves to a new physical location.

10 22. The device in accordance with claim 18, wherein the device is a wireless, portable device.

23. A system substantially as hereinbefore described with reference to at least one of the accompanying drawings.

24. A method substantially as hereinbefore described with reference to at least one of the accompanying drawings.

25. A device substantially as hereinbefore described with reference to at least one of the accompanying drawings.

DATED this 20th day of August 2002

ERICSSON INC

WATERMARK PATENT & TRADE MARK ATTORNEYS
290 BURWOOD ROAD
HAWTHORN VICTORIA 3122
AUSTRALIA



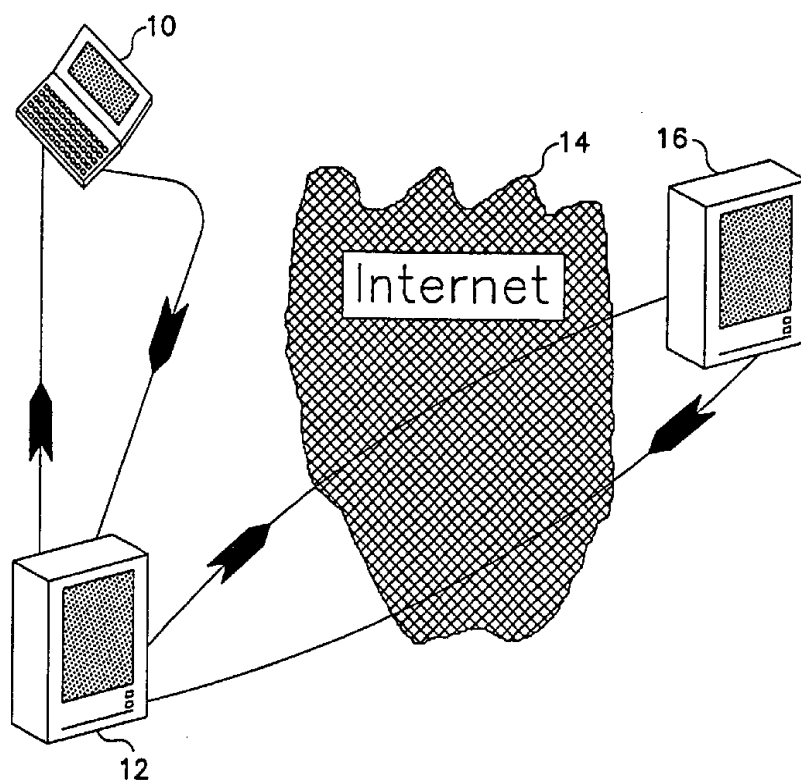


FIG. 1

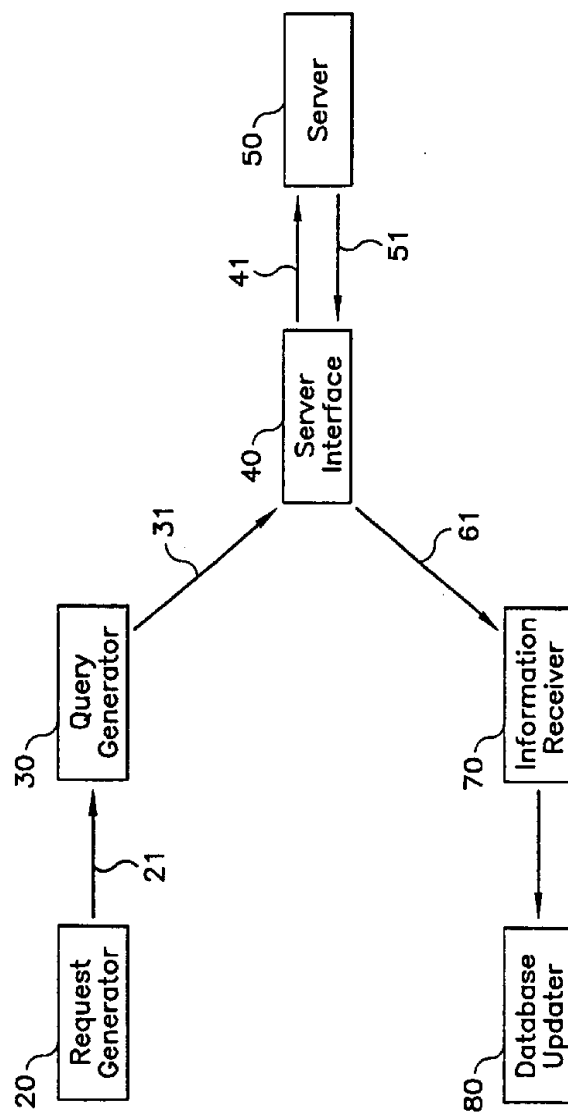


FIG. 2

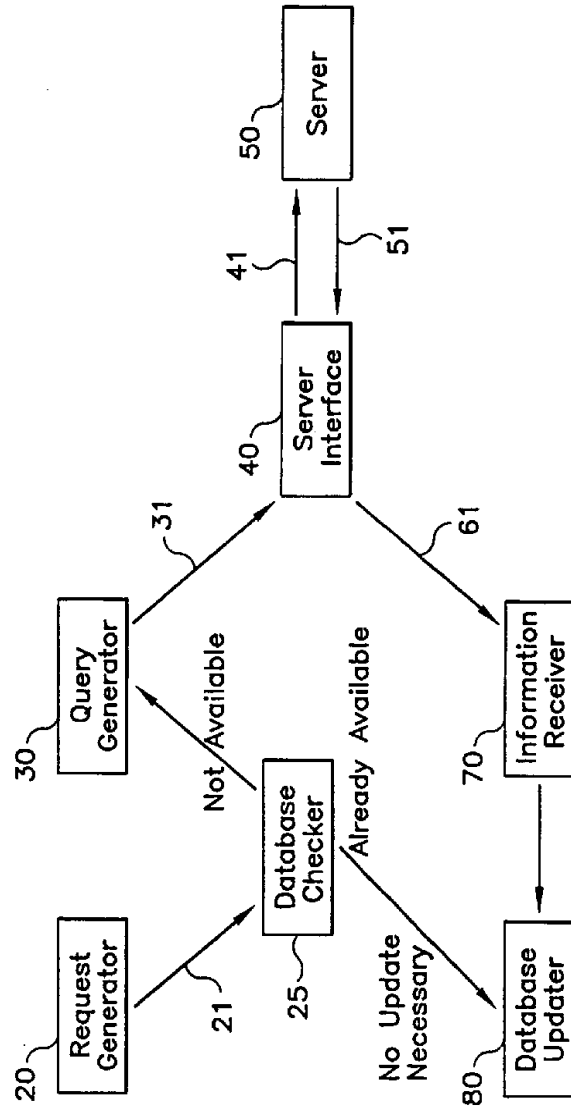
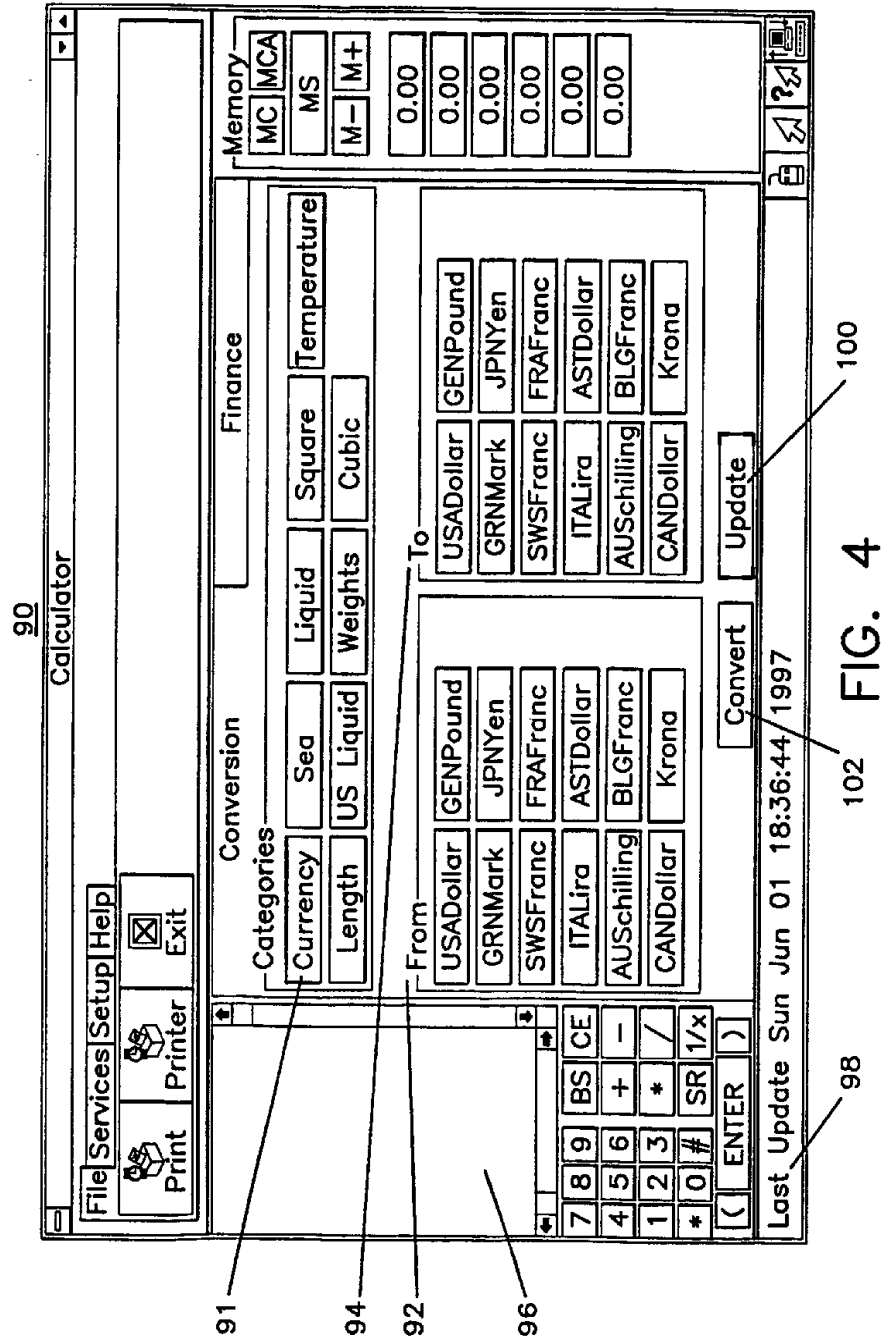


FIG. 3



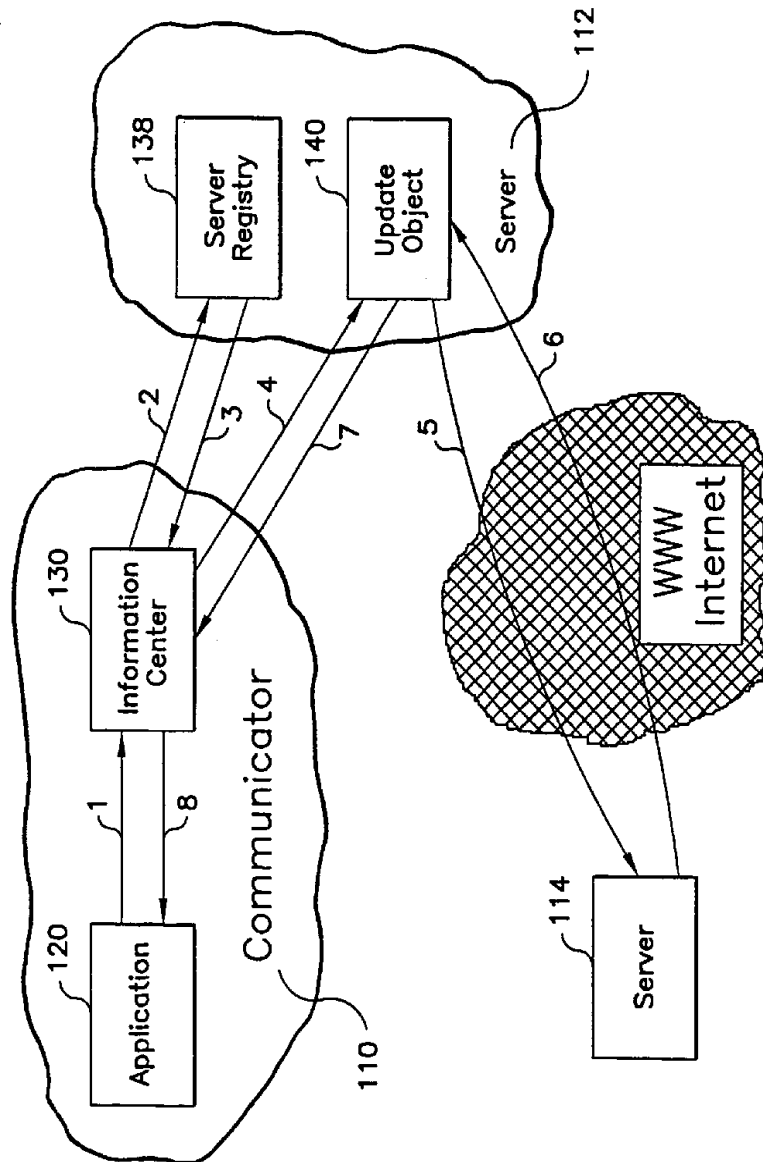


FIG. 5

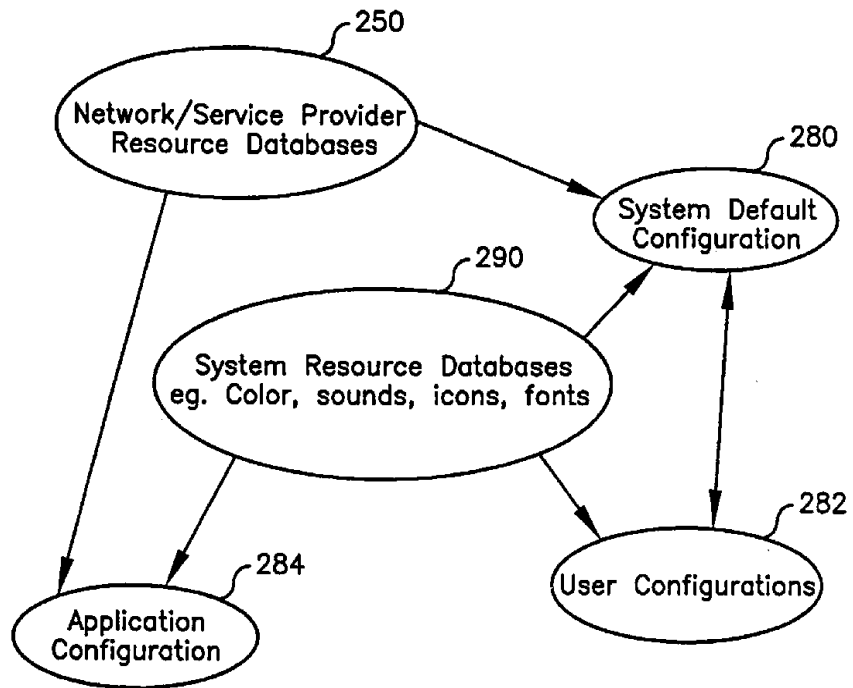


FIG. 6

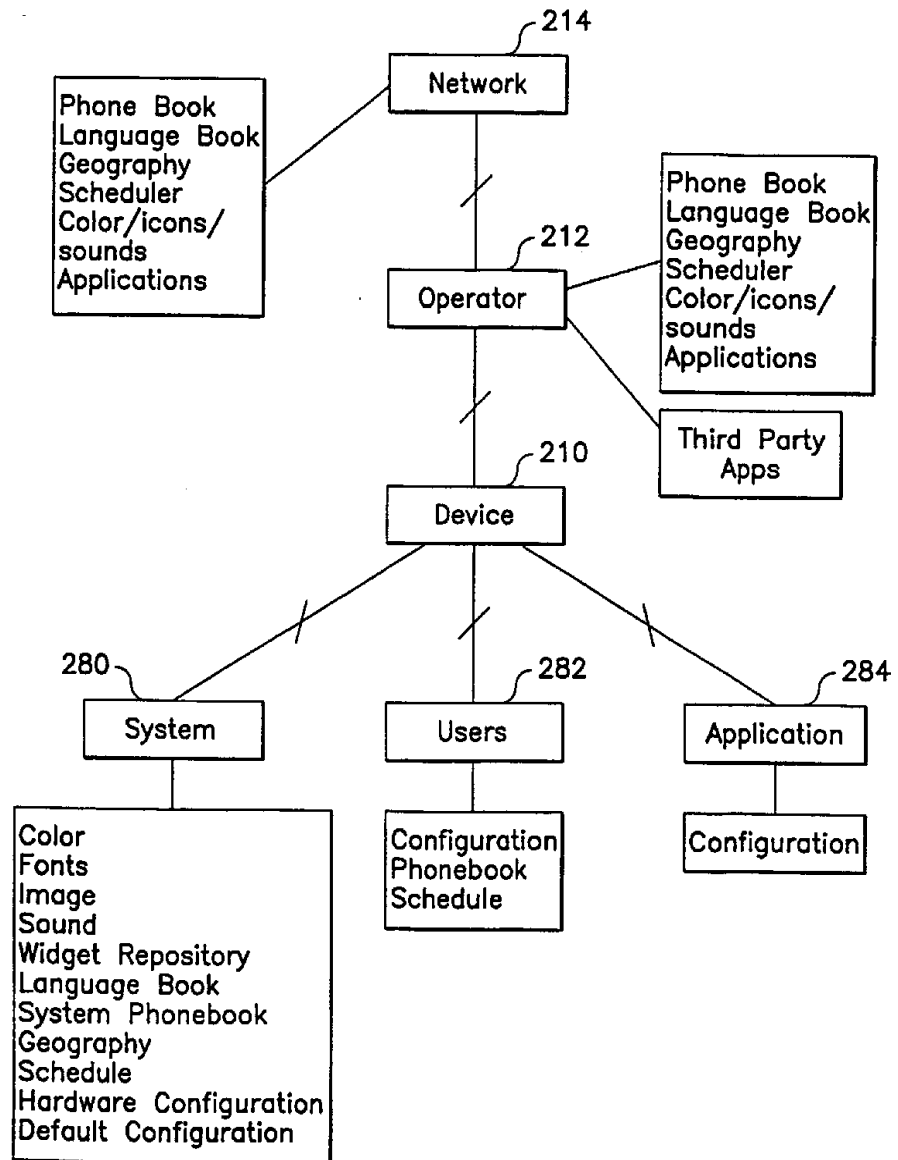


FIG. 7