US 20060116817A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2006/0116817 A1

(43) Pub. Date: Jun. 1, 2006

Salmre et al.

(54) LOCATION AWARE MOBILE-DEVICE SOFTWARE DEVELOPMENT

(75) Inventors: **Ivo W. Salmre**, Aachen (DE); **Frank Mantek**, Wermelskirchen (DE)

Correspondence Address:
MERCHANT & GOULD (MICROSOFT)
P.O. BOX 2903
MINNEAPOLIS, MN 55402-0903 (US)

(73) Assignee: **Microsoft Corporation**, Redmond, WA

(21) Appl. No.: **10/998,081**

(22) Filed: **Nov. 26, 2004**

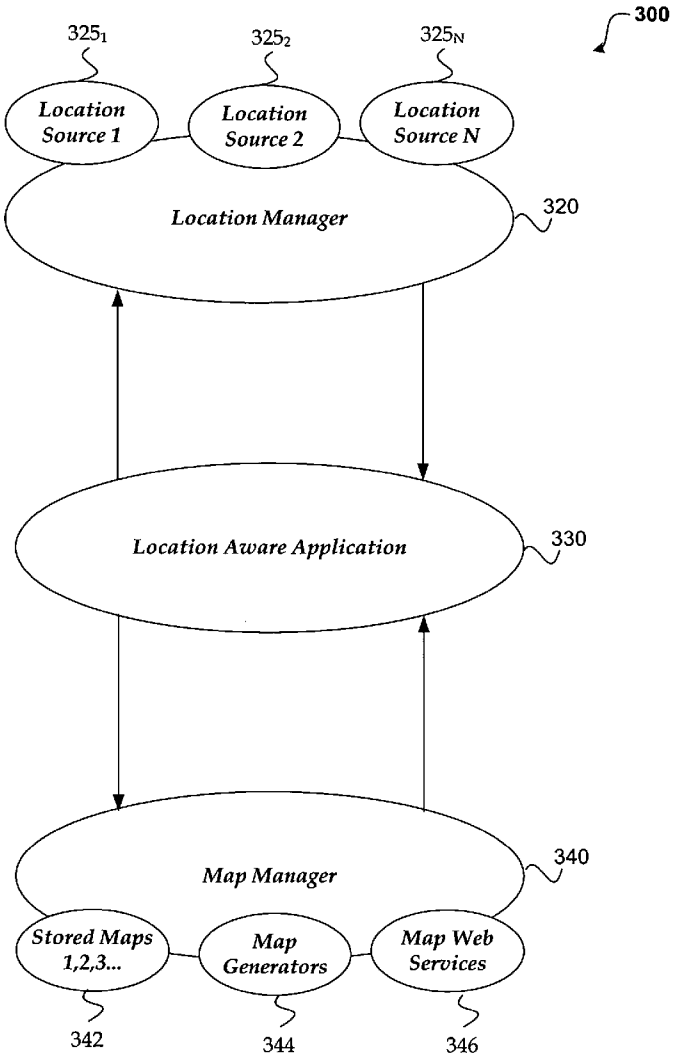
Publication Classification

(51) **Int. Cl.**
G01C 21/30 (2006.01)

(52) **U.S. Cl.** **701/211**; 340/995.1; 340/995.14; 701/208

(57) **ABSTRACT**

A development environment makes it easier to develop and use location aware software applications. Combining the currently separate and complex tasks of location information gathering, analysis, and display and making them integrated such that non-experts can build location services into their applications increases a developer's capability to write rich location-aware software applications. A higher level programming model allows a programmer to use location information as a logical entity rather than just raw location data, such as latitude and longitude. A location data store and map data store may be accessed by different location aware applications on the device. Instead of each location application relying on its own data store, applications may share their location related information.



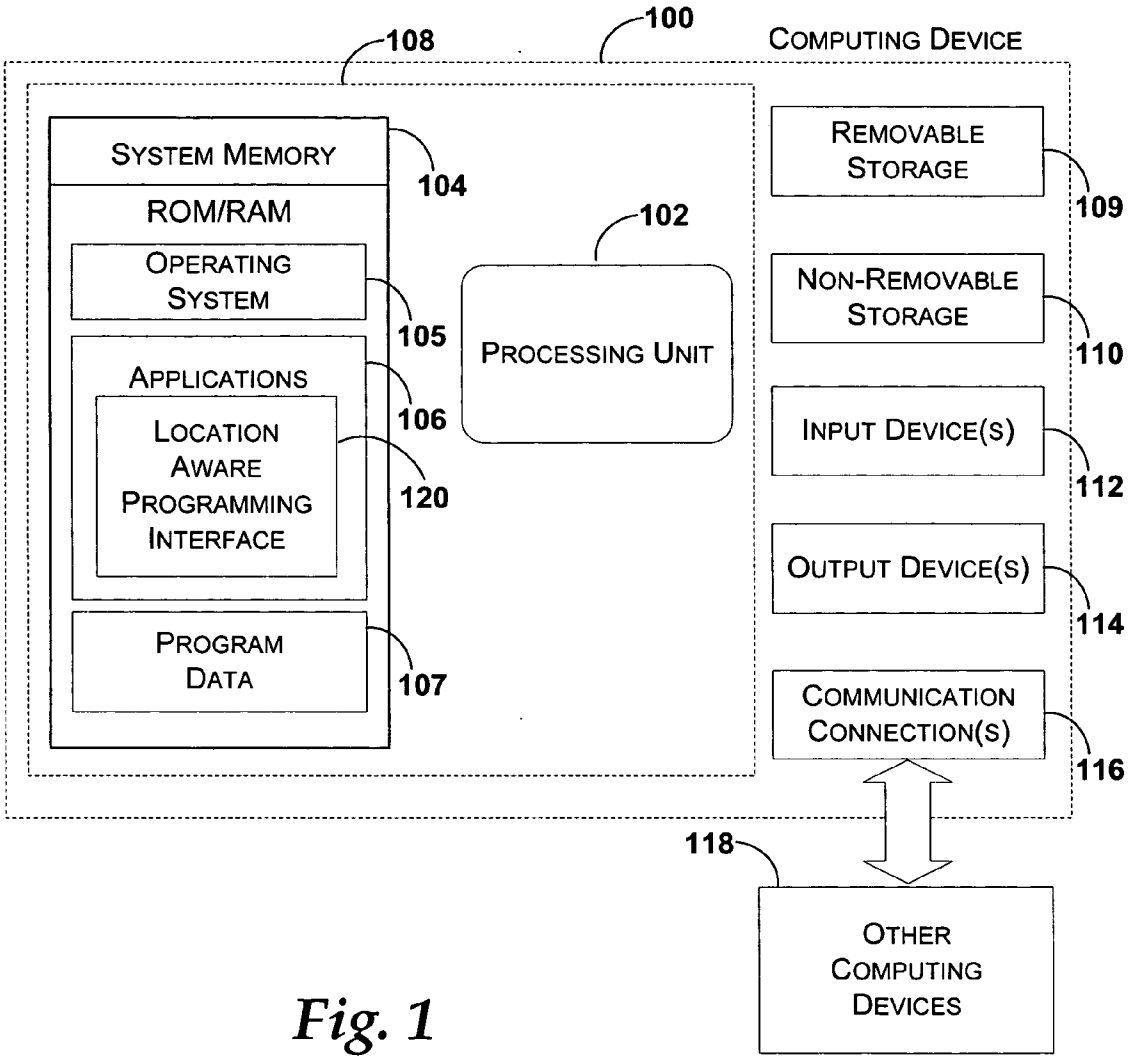

100
COMPUTING DEVICE
108
SYSTEM MEMORY
104
ROM/RAM
OPERATING SYSTEM
105
APPLICATIONS
106
LOCATION AWARE PROGRAMMING INTERFACE
120
PROGRAM DATA
107
102
PROCESSING UNIT
REMOVABLE STORAGE
109
NON-REMOVABLE STORAGE
110
INPUT DEVICE(S)
112
OUTPUT DEVICE(S)
114
COMMUNICATION CONNECTION(S)
116
118
OTHER COMPUTING DEVICES

Fig. 1

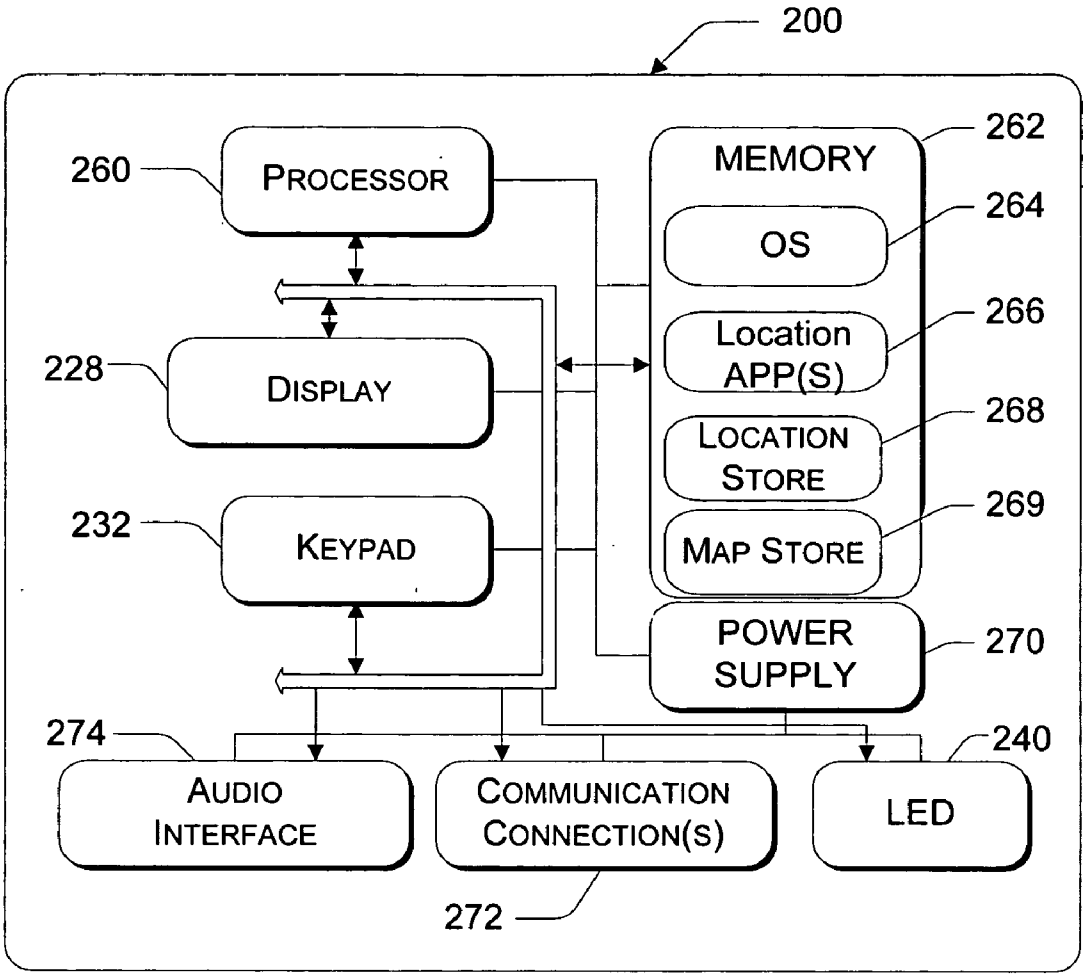

200
260
PROCESSOR
MEMORY
262
264
OS
266
Location APP(S)
228
DISPLAY
268
LOCATION STORE
269
232
KEYPAD
MAP STORE
POWER SUPPLY
270
274
240
AUDIO INTERFACE
COMMUNICATION CONNECTION(S)
LED
272

Fig.2

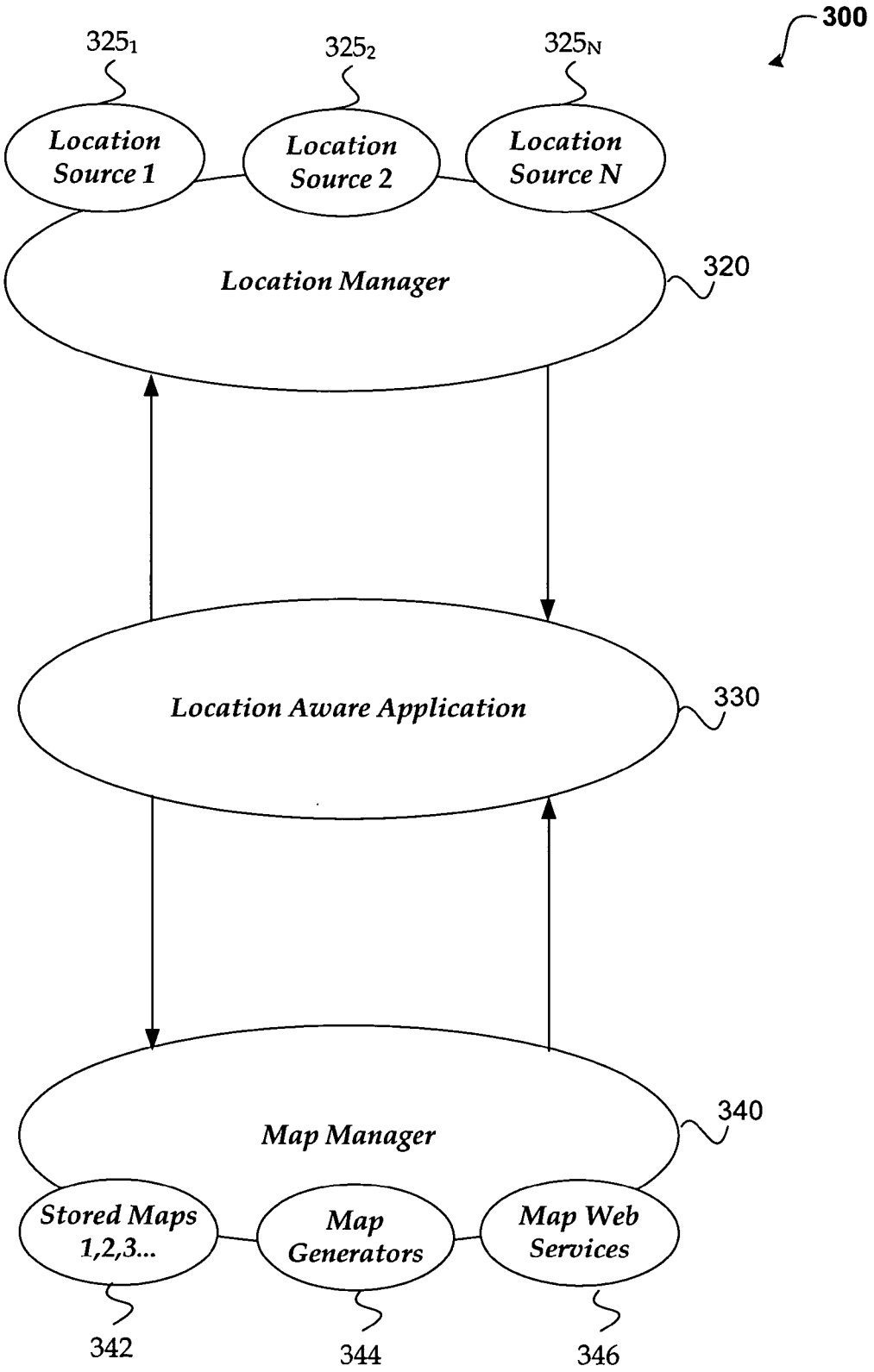

300
325₁
325₂
325N
Location Source 1
Location Source 2
Location Source N
Location Manager
320
Location Aware Application
330
Map Manager
340
Stored Maps 1,2,3...
Map Generators
Map Web Services
342
344
346

Fig. 3

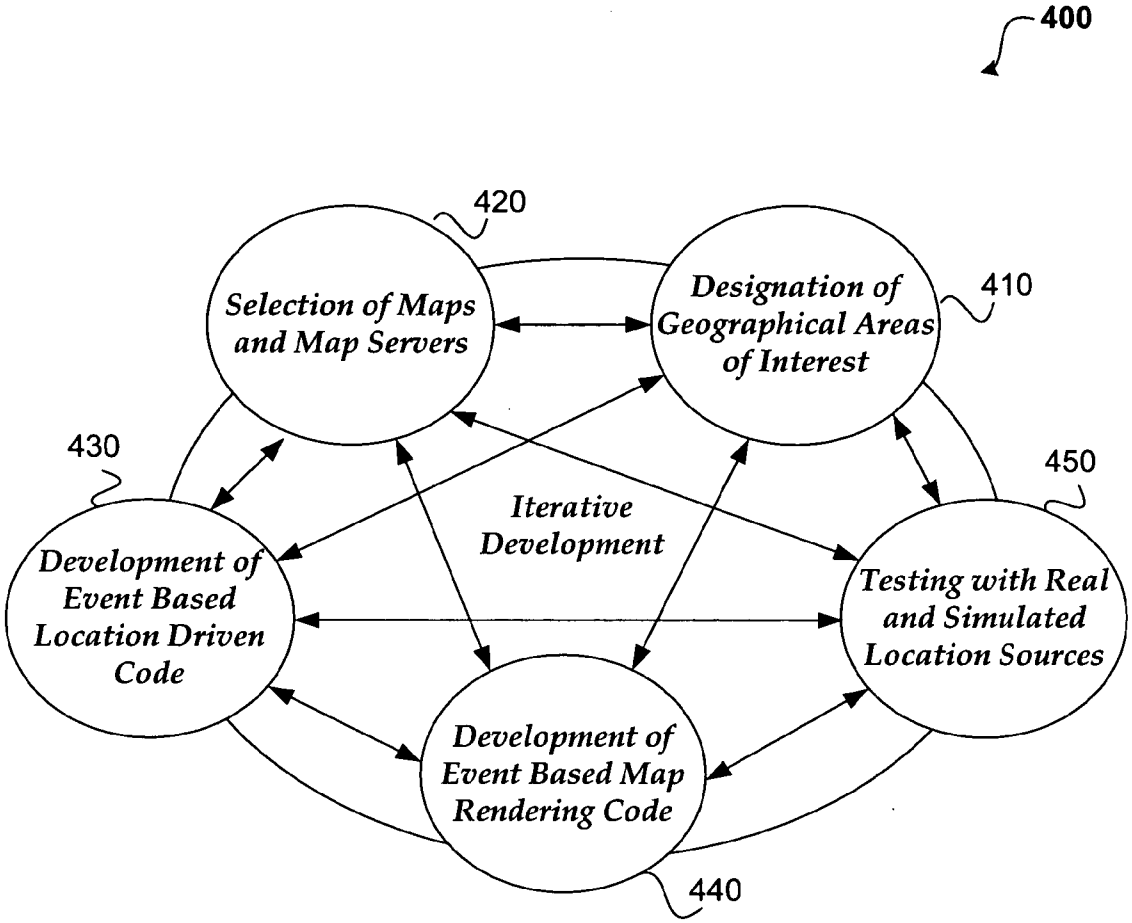

400
420
Selection of Maps and Map Servers
Designation of Geographical Areas of Interest
410
430
450
Iterative Development
Development of Event Based Location Driven Code
Testing with Real and Simulated Location Sources
Development of Event Based Map Rendering Code
440

Fig. 4

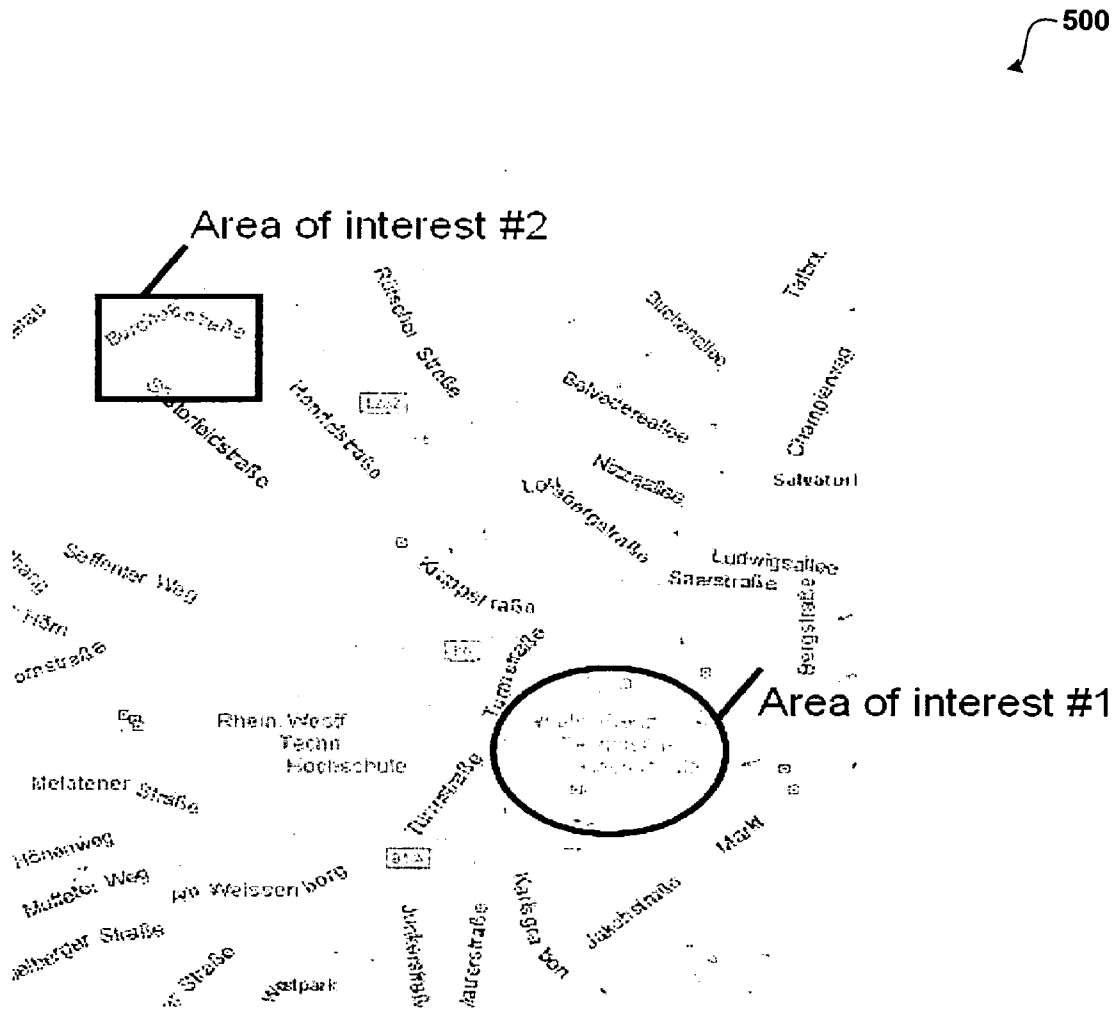

500
Area of interest #2
Area of interest #1
Rhein. Westf
Techn
Hochschule
Melatener Straße
Seffenter Weg
Ludwigsallee
Saarstraße
Turmstraße
Kruppstraße
Nizzaallee
Belvederealle
Championweg
Salvatori
Mark
Jakobstraße
Karlsgraben
Westpark

Fig. 5

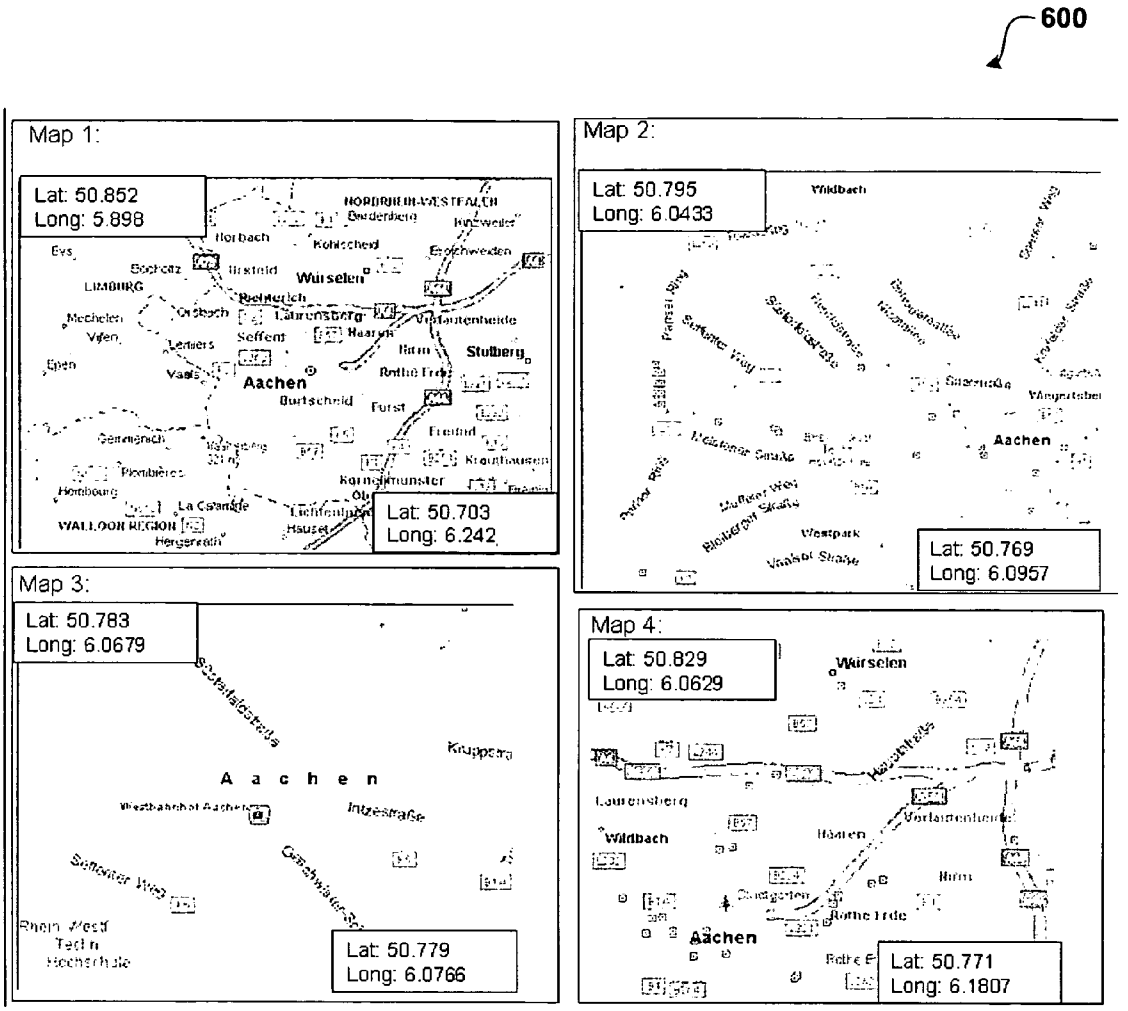

600
Map 1:
Lat: 50.852
Long: 5.898
Lat: 50.703
Long: 6.242
Map 2:
Lat: 50.795
Long: 6.0433
Lat: 50.769
Long: 6.0957
Map 3:
Lat: 50.783
Long: 6.0679
Lat: 50.779
Long: 6.0766
Map 4:
Lat: 50.829
Long: 6.0629
Lat: 50.771
Long: 6.1807

Fig. 6

700

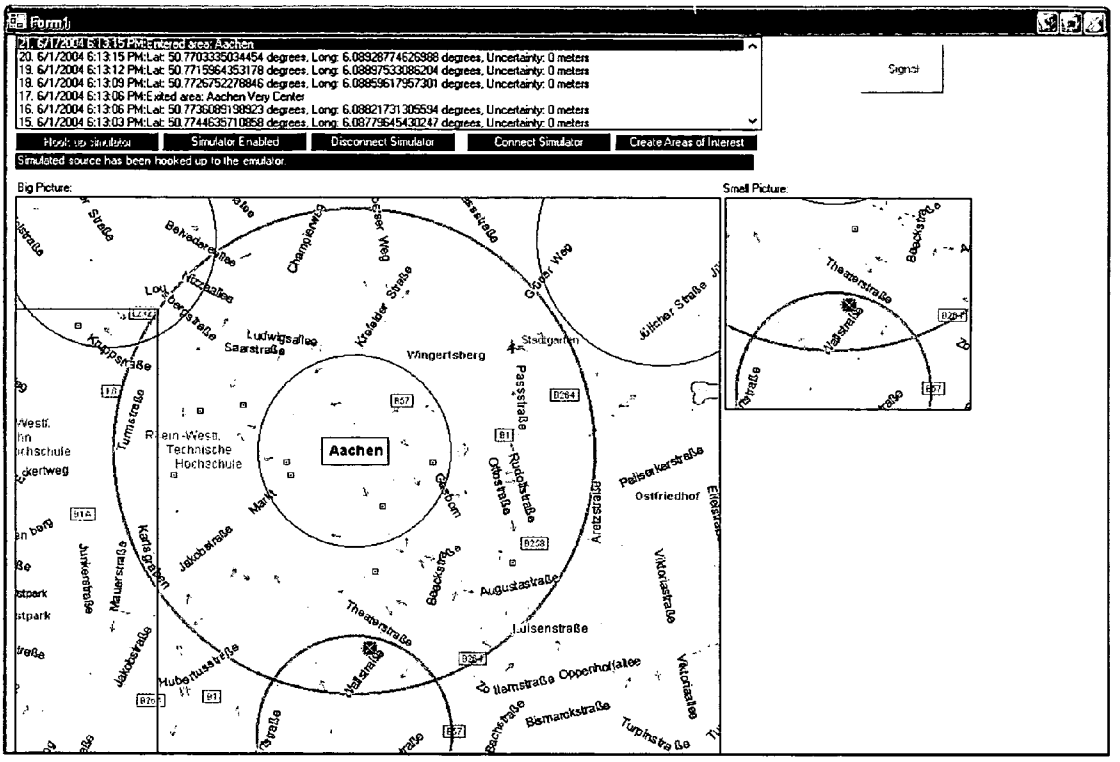

Form1
21. 6/1/2004 6:13:15 PM:Entered area: Aachen
20. 6/1/2004 6:13:15 PM:Lat: 50.7703335034454 degrees, Long: 6.08928774626988 degrees, Uncertainty: 0 meters
19. 6/1/2004 6:13:12 PM:Lat: 50.7715964353178 degrees, Long: 6.08897533086204 degrees, Uncertainty: 0 meters
18. 6/1/2004 6:13:09 PM:Lat: 50.7726752278846 degrees, Long: 6.08859617957301 degrees, Uncertainty: 0 meters
17. 6/1/2004 6:13:06 PM:Exited area: Aachen Very Center
16. 6/1/2004 6:13:06 PM:Lat: 50.7736089198923 degrees, Long: 6.08821731305594 degrees, Uncertainty: 0 meters
15. 6/1/2004 6:13:03 PM:Lat: 50.7744635710858 degrees, Long: 6.08779645430247 degrees, Uncertainty: 0 meters
Hook up simulator
Simulator Enabled
Disconnect Simulator
Connect Simulator
Create Areas of Interest
Simulated source has been hooked up to the emulator.
Signal
Big Picture:
Small Picture:
Aachen
Wingertsberg
Ludwigsallee
Saarstraße
Krefelder Straße
Championweg
Grüner Weg
Jülicher Straße
Passstraße
Rudolfstraße
Ottostraße
Pelzerstraße
Ostfriedhof
Aretzstraße
Augustastraße
Luisenstraße
Theaterstraße
Hallstraße
Beckstraße
Viktoriastraße
Viktoriaallee
Bismarckstraße
Turmstraße
Jakobstraße
Karlsgraben
Mauerstraße
Junkerstraße
Hubertusstraße
Rhein-West. Technische Hochschule
Markt
Gasborn
Turpinstraße
Belvederestraße
Nizzaallee
Krippstraße
Stadtgarten

Fig. 7

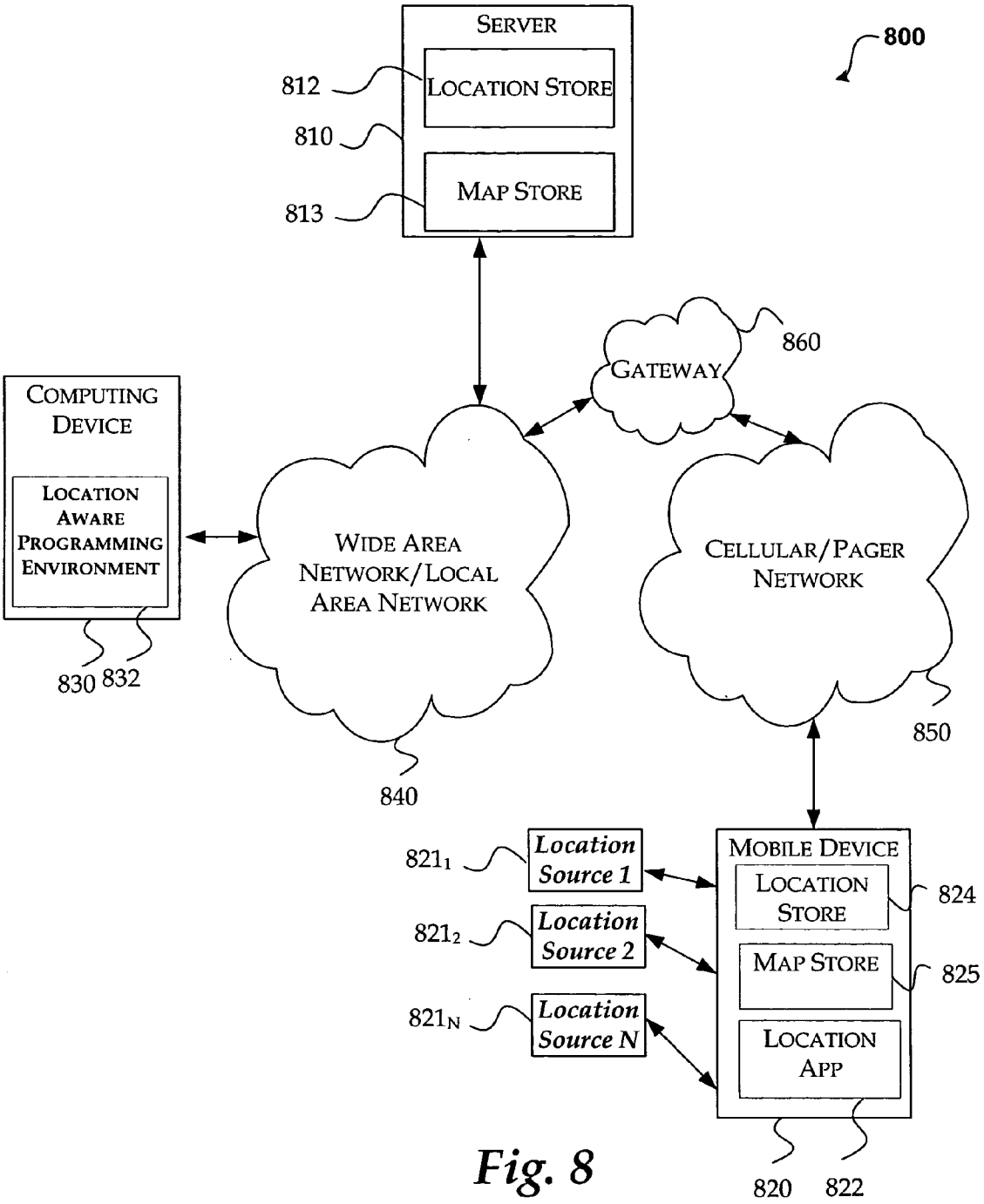

SERVER
LOCATION STORE
MAP STORE
812
810
813
800
GATEWAY
860
COMPUTING DEVICE
LOCATION AWARE PROGRAMMING ENVIRONMENT
830
832
WIDE AREA NETWORK/LOCAL AREA NETWORK
840
CELLULAR/PAGER NETWORK
850
821₁
Location Source 1
821₂
Location Source 2
821N
Location Source N
MOBILE DEVICE
LOCATION STORE
824
MAP STORE
825
LOCATION APP
820
822

Fig. 8

LOCATION AWARE MOBILE-DEVICE SOFTWARE DEVELOPMENT

BACKGROUND OF THE INVENTION

[0001] In the past several years more and more applications utilizing location information have been developed. Some of these location applications include map programs, flight-tracker programs, and navigation programs. Many of these applications are being developed for mobile computing devices. For example, in-car navigation systems have been developed to help users navigate through a city. Some cellular phones also include GPS based navigation programs. It is very difficult, however, to develop these applications. Positioning systems typically only provide location data, such as latitude and longitude, and require the application to process and interpret the information.

SUMMARY OF THE INVENTION

[0002] Embodiments of the present invention are related to a system and method for making it easier to develop and use location aware software applications. Combining the currently separate and complex tasks of location information gathering, analysis, and display and making them integrated such that non-experts can build location and mapping services into their applications increases a developer's capability to write rich location-aware software applications.

[0003] According to one aspect, an application programming interface for building location aware applications is used. This higher level programming model is directed towards allowing a programmer to use location information as a logical entity rather than just raw location data, such as latitude and longitude. The location information is abstracted into areas of interest. The developer also does not need to concern themselves with building their own map rendering software and may use rendering engines developed by third parties.

[0004] According to another aspect of the invention, an event driven programming model is used in building the location aware applications. Events may be generated in response to the location of an item relative to an area of interest. For example, a programmer designates one routine to be executed when a device enters a first area of interest, and another routine to be executed when a device leaves a second area of interest.

[0005] According to yet another aspect of the invention, a location data store may be accessed by different location aware applications. Instead of each location application relying on its own data store, applications may share their location related information.

[0006] According to yet another aspect of the invention, shared maps may be accessed by different location aware applications. Instead of each location application relying on its own map store and map rending technology, applications may work with a shared repository of maps data and map rendering engines.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] **FIGS. 1 and 2** illustrate an exemplary computing devices that may be used in exemplary embodiments of the present invention;

[0008] **FIG. 3** illustrates a location aware system;

[0009] **FIG. 4** shows a state diagram of interaction for development of location aware applications;

[0010] **FIG. 5** shows defining exemplary areas of interest;

[0011] **FIG. 6** illustrates exemplary maps that may be used by a location aware application;

[0012] **FIG. 7** illustrates an exemplary simulation screenshot for a location aware application; and

[0013] **FIG. 8** is a functional block diagram generally illustrating an location aware development system in accordance with aspects of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0014] Generally, the present invention is directed at providing a method and system for making it easier to develop and use location aware software applications. Combining the currently separate and complex tasks of location information gathering, analysis, and display and making them integrated such that non-experts can build location services into their applications increases a developer's capability to write rich location-aware software applications

Illustrative Location Aware Development System

[0015] **FIG. 3** illustrates a location aware system, in accordance with aspects of the invention. Location system **300** includes location manager **320**, location aware application **330**, and map manager **340**. Location manager is coupled to location sources **325**(1-N). Map manager is coupled to map store(s) **342**, map generator(s) **344**, and map web service(s) **346**.

[0016] Location manager **320** is configured to receive location information from on or off device location sources (**325**(1-N)) and abstract the location information into location events that are provided to location aware application **330**. The location sources may be anything that provides location information about an item. For example, one location source may be a GPS receiver. Another location source may be a user input that indicates the position of an item. For example, a user may indicate through a user input on a device that they are at a certain location. The user may enter a location into their device using a keypad, touch screen, voice recognition, and the like. Other location sources may be simulated location sources. Instead of a developer having to be in the field to test application **330**, they may use simulated location data. Another location source may be an RFID tag. Each location source may provide different resolutions of location data. For example, some location sources may be very accurate while other location sources may only be accurate to within a neighborhood, on a road, and the like.

[0017] A location source can correspond to a proximate or distant location source. A proximate source corresponds to location generating hardware, including, but not limited to a GPS receiver or to the mobile device itself which is getting its location from network information. A distant location source corresponds to location information either pushed or pulled from a remote networked source (e.g. such as a pushed SMS message arriving with the location of my friend's car, or location information returned (pulled) via a web-request). Both the proximate and distant location

sources are treated as abstract logical entities. This makes the programming model similar for many different kinds of location information.

[0018] Once the location data is received, location manager **320** determines if any location event is triggered that relates to application **330**. Instead of only being able to provide raw location data from the location sources to application **330**, location manager **320** provides location events upon the occurrence of an item having some logical relationship with an area of interest, such as entering or leaving an area of interest.

[0019] Generally, an area of interest provides an abstraction of the location data (See **FIG. 5** and related discussion). An area if interest can relate to any area and can be indoors or outdoors. For example, an indoor location could be used to show the location of a fixed asset (e.g. conference room) and/or moveable asset (e.g. people, office equipment, inventory).

[0020] An outdoor area of interest could be a city block, an area around a building or some other landmark, or some other predefined area. An outdoor area of interest may also be a movable asset (e.g. a car or a person).

[0021] Instead of an application having to interpret the raw location data, application **330** registers for location events based on the relation of the item to areas of interest. Upon the occurrence of the predefined location event, location manager **320** notifies application **330**. At that point, application **330** executes an event driven code routine that was developed to handle the location event.

[0022] Location aware application **340** responds to the generated location events. Upon the occurrence of a location event, location aware application **330** supplies a set of weighted criteria to MapManager **340**. For example, the weighted criteria may include a suggested map resolution along with a set of objects that the applications would like displayed on the map they draw. The application may also supply objects that they want rendered on top of the existing map. In addition to the application responding to event driven rendering of maps, the application can also be self driven. For example, an application may have a timer that periodically "pulls" information and renders a new map. So in addition to "push" event-driven responses, the application can also "pull" data and take action.

[0023] Map manager **340** abstracts the map information from the application such that it does not need to know details regarding how to handle maps or location information. Instead of the application having to include complex map rendering code, the application provides map manager **340** with the name or type of map it would like displayed. Map generators **344** are not tied to any single application, or rendering engine. For example, one rendering engine may be vector based, whereas another rendering engine may be bitmap based. Location aware application **330** is provided with an interface to interact with map generator **344**.

[0024] Map Manager **340** provides a rich set of features to location aware application without application **330** having to know the details of how to implement the map code. For example, suppose application **330** wants to allow a user to point to an area on a map and then have that data translated to a latitude and longitude, or in the case of an indoor map into indoor coordinate systems. Map manager **340** provides that information to application **330**. MapManager **340** acts as the broker between application **330** and the map-rendering technologies and supplies functionality to the application. Some of the functionality includes mapping X/Y screen coordinates on the device to Lat/Long/Altitude map coordinates on the map. For example, a user may click on the screen that is currently displaying a map to input their location. In response to the user input, map manager maps the X/Y screen location to the Lat/Long/Altitude map location. Similarly, map manager **340** maps Lat/Long/Altitude map data to X/Y screen locations allowing the application to dynamically render on top of the currently displayed map. The altitude on some map renderings, such as a ski-area map, may significantly affect the X/Y location. There are many ways of converting an X/Y coordinate into a Lat/Long/Altitude coordinate. Generally, map manager receives the X/Y screen coordinate, matches the coordinate to the map, and then, based on the scale of the map, its projection technology and map boundaries provides the Lat/Long/Altitude coordinate. The application does not need to understand how the maps themselves are being rendered or what "projection technology" is being used. Many different map projection technologies are possible, including, but not limited to, classical map-projection techniques such as Mercator projections, Cylindrical projections, Polar projections, Grid projections and custom projection technologies (e.g. a ski-area map, a stadium seating map, etc). The application may seamlessly move between different projection techniques.

[0025] MapManager **340** also supplies high level events about the set of available maps to application **330**. These events provide the application important information such as when there are several maps available that meet the criteria requested by the application. Application **330** is not confined to using a single map for display. For example, one application may use a street map for one period of time and then use a metro map (e.g. showing bus routes, subway tunnels, etc.) during another time. Map manager **340** ranks the maps according to their order of suitability. When there are no maps available that meet the criteria provided by application **330**, but map manager **340** may locate maps that meet some of the criteria, then those maps may be provided.

[0026] MapManager **340** also allows for the dynamic provisioning of new maps onto the device. The maps may be downloaded to the device at any time. For example, a new map may be downloaded for a public transportation application when a new transit route is added. According to one embodiment, these maps are represented using two files, one binary and opaque to MapManager and one that it understands. According to one embodiment, the map is represented using an XML file. The map manager is able to read the XML file and obtain the information necessary to add the map to the on-device local map data store. Maps may also be added to the map store by the developer or through other sources.

[0027] The following is an exemplary XML representation.

```
-<MapData Version="1.01"
MapProjection="SimpleGrid"
```

-continued

```
MetersPerPixel="2.80253449893166"
MapInfoClass="none" CustomMapRenderer="none">
    -<SimpleLatLongRectangle
    MapFileName="device__map__1__0.PNG">
        <MapCoordinate Position="NorthWestCorner"
        Latitude="50.7867" Longitude="6.07814549019608" />
        <MapCoordinate Position="NortEastCorner"
        Latitude="50.7867" Longitude="6.09407106666667" />
        <MapCoordinate Position="SouthWestCorner"
        Latitude="50.7766295454545"
        Longitude="6.07814726827094" />
        <MapCoordinate Position="SouthEastCorner"
        Latitude="50.7766295454545"
        Longitude="6.09406985757576" />
    </SimpleLatLongRectangle>
</MapData>
```

[0028] MapProjection refers to the technology that is used to display the map. In this particular example, the map projection type is a simple grid. MetersPerPixel refers to the average resolution of the map. MapInfoClass refers to whether there is a custom class that is needed to answer questions about this map. This field allows a developer to include a custom class within the programming environment that may be called when this map is displayed. Similarly, CustomMapRenderer refers to whether a custom rendering engine is needed to draw the map. The MapFileName field refers to the binary file name that contains information that the map-renderering engine that will be used to draw the map. The SimpleLatLongRectangle field identifies the map as a rectangle defined by latitude and longitude coordinates.

[0029] The following example illustrates some of the custom fields being utilized.

```
-<MapData Version="1.01" MapProjection="SimpleGrid"
MetersPerPixel="2.80253449893166"
MapInfoClass="CustomMapInfo1"
CustomMapRender="CustomMapRender1">
    <CustomMapInfo1 MapFileName="device__map__1__0.PNG"
    classId="CustomMapEngine.MapInfoParser" />
    <CustmMapRender1 MapFileName="device__map__1__0.PNG"
    classId="CustomMapEngine.MapRenderEngine" />
</MapData>
```

[0030] MapManager **340** may have multiple maps available on the device in a map store (**342**) that may be used by application **330**. Map manager **340** searches though the set of maps available on the device in map store **342** and looks for the best map to be used. After being selected, the best fit map is selected and rendered.

[0031] **FIG. 4** shows a state diagram of interaction for development of location aware applications, in accordance with aspects of the invention.

[0032] Starting at state **410**, geographical areas of interest are defined. The selection of the areas of interest may occur before or after the location aware application is deployed. Additionally, this information may be sent out to the devices at any time. An area of interest is a predefined region that relates to a map. The area of interest may be any size that is definable. For example, an area of interest may be the size a square meter or an entire country. The areas of interest are defined based on the needs of the location aware application. For example, suppose the location aware application being developed determines when a child leaves their school and returns home. In this particular example, areas of interest may be an area including the school and another area of interest including the child's home. Some areas of interest may be selected by the developer, while other areas of interest may be selected by a user. For example, many applications, such as the example described above, are configured after the application is provided to the user. In the above example, a parent may select the areas of interest.

[0033] State **420** shows the selection of maps and map servers. A developer may select between the maps available in the system or may create their own maps. According to one embodiment of the invention, the available maps are shared across location applications such that different location aware applications may use the same map. In this way, each location aware application does not need to develop their own maps. In most instances, the specifics of the map rendering is hidden from the developer of the location aware application. Should the developer desire to include their own rendering software they may incorporate their rendering software into the environment at state **440**.

[0034] At state **440**, the developer may also set up their preference for the rendering. For example, the developer may want to use a vector based rendering engine for one part of the application and a bitmap based system for another part of the application. The map engines may be provided by many different sources. Any developer who wants to build an engine may supply a map engine that conforms to the programming interface. Each map engine is plugged into the development system.

[0035] At state **430**, the developer programs the event based location code that is executed in response to events associated with the areas of interest within their application. This event based code is dependent upon the type of location aware application being developed and may include many different types of data. For example, a program may perform an action when a user reaches a predetermined location. In the tracking the child example above, an example routine may be programmed to highlight the home or school based on the location of the child at that particular time. The developer may also place items onto the map in response to the events.

[0036] At state **450**, the location aware application may be tested using real and/or simulated location sources. The system provides a simulator of a location source such that an application may be tested before it is put to us in the field. Using a simulated data source allows a developers event based location code to be tested upon the occurrence of a location event. Another use of a simulated location-source data is to "play back" previously recorded real-world location data. This is useful for application development, testing and debugging.

[0037] **FIG. 5** shows defining exemplary areas of interest, in accordance with aspects of the invention. As illustrated in **FIG. 5**, two areas of interest have been defined. Area of interest #1 is defined by an oval boundary area and area of interest #2 is defined by a rectangular border. Although the areas of interest have been defined using predetermined geometric shapes, the areas of interest may be more defined in any way that defines the area. For example, the area of

interest could be defined as a neighborhood, a county, an area between certain roads, and the like. According to one embodiment of the invention, the developer uses a set of predefined shapes to designate the areas of interest. The user may also select a point on the map and then provide a radius to extend from that point to be the area of interest. The area of interest may also be set using latitude and longitude coordinates.

[0038] While the map illustrated in **FIG. 5** shows a city map including streets, the map may be any type of map. For example, the map could be a ski map, a map of a forest, a park, a golf course, building floor plans, stadium seating maps and the like. As discussed above, when and area of interest is designated an event may be triggered based on something entering or leaving the area.

[0039] While not illustrated, an area of interest may be a certain type of location. For example, a location aware application may be interested in knowing when it is close to a coffee shop or a movie theater. In this particular example, each coffee shop or movie theater known by the system could be an area of interest. As more coffee shops become available, their location could be dynamically added to the application such that the application is aware of them. Other location aware applications may define an area of interest as a particular store, sporting arena, bus stop, office building, and the like.

[0040] **FIG. 6** illustrates exemplary maps that may be used by a location aware application, in accordance with aspects of the invention. As illustrated, these maps are defined as a simple rectangle defined by latitude and longitude. Each of the four maps includes an upper left latitude and longitude and a lower right latitude and longitude. As discussed above with regard to **FIG. 3**, the maps may be defined in many different ways. Any location aware application could use these maps for display when appropriate.

[0041] **FIG. 7** illustrates an exemplary simulation screenshot for a location aware application, in accordance with aspects of the invention. Within the simulator, two map views are displayed including a smaller more detailed view and a global, overall view. Illustrated on the maps are predefined areas of interest which are represented by circular regions and square regions. When the simulated movement enters or leaves an area, the developer may determine if the appropriate code they developed is working properly.

Illustrative Operating Environment

[0042] With reference to **FIG. 1**, one exemplary system for implementing the invention includes a computing device, such as computing device **100**. In a very basic configuration, computing device **100** typically includes at least one processing unit **102** and system memory **104**. Depending on the exact configuration and type of computing device, system memory **104** may be volatile (such as RAM), non-volatile (such as ROM, flash memory, etc.) or some combination of the two. System memory **104** typically includes an operating system **105**, one or more applications **106**, and may include program data **107**. In one embodiment, application **106** may include location aware programming interface **120** that is used in building location aware applications for mobile devices. This location aware programming interface can be integrated with map selection, management and rendering interfaces. This basic configuration is illustrated in **FIG. 1** by those components within dashed line **108**.

[0043] Computing device **100** may have additional features or functionality. For example, computing device **100** may also include additional data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Such additional storage is illustrated in **FIG. 1** by removable storage **109** and non-removable storage **110**. Computer storage media may include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, program modules, or other data. System memory **104**, removable storage **109** and non-removable storage **110** are all examples of computer storage media. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computing device **100**. Any such computer storage media may be part of device **100**. Computing device **100** may also have input device(s) **112** such as keyboard, mouse, pen, voice input device, touch input device, etc. Output device(s) **114** such as a display, speakers, printer, etc. may also be included.

[0044] Computing device **100** may also contain communication connections **116** that allow the device to communicate with other computing devices **118**, such as over a network. Communication connection **116** is one example of communication media. Communication media may typically be embodied by computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave or other transport mechanism, and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. The term computer readable media as used herein includes both storage media and communication media.

[0045] **FIG. 2** illustrates a mobile computing device that may be used in one exemplary embodiment of the present invention. With reference to **FIG. 2**, one exemplary system for implementing the invention includes a mobile computing device, such as mobile computing device **200**. Mobile computing device **200** includes processor **260**, memory **262**, display **228**, and keypad **232**. Memory **262** generally includes both volatile memory (e.g., RAM) and non-volatile memory (e.g., ROM, Flash Memory, or the like). Mobile computing device **200** includes operating system **264**, such as the Windows CE operating system from Microsoft Corporation, or another operating system, which is resident in memory **262** and executes on processor **260**. Keypad **232** may be a push button numeric dialing pad (such as on a typical telephone), a multi-key keyboard (such as a conventional keyboard). Display **228** may be a liquid crystal display, or any other type of display commonly used in mobile computing devices. Display **228** may be touch-sensitive, and would then also act as an input device.

[0046] One or more application programs, such as location app(s) **266** are loaded into memory **262** and run on the operating system **264**. One or more location aware applications (**266**) reside on mobile computing device **200** and are programmed to perform instructions relating to location based applications. The location aware application may reside in the hardware or software of the device. Mobile computing device **200** may also include volatile and non-volatile storage within memory **262**. Memory **262** also includes location data store **268** that is used to store location based information and map store **269** that is used to map based information. Location store **268** is a global facility for storing location that may be used by all location aware applications (**266**) on the device. Map store **269** is also a global facility for storing map information that may be used by all location aware applications (**266**) on the device. According to one embodiment, the repository (**268**) is on-device cache of location and map information. There may be more than one location store on mobile device **200**. For example, there may be a private and public location data store.

[0047] Mobile computing device **200** includes power supply **270**, which may be implemented as one or more batteries. Power supply **270** might further include an external power source, such as an AC adapter or a powered docking cradle that supplements or recharges the batteries.

[0048] Mobile computing device **200** is shown with two types of optional external notification mechanisms: LED **240** and audio interface **274**. These devices may be directly coupled to power supply **270** so that when activated, they remain on for a duration dictated by the notification mechanism even though processor **260** and other components might shut down to conserve battery power. Audio interface **274** is used to provide audible signals to and receive audible signals from the user. For example, audio interface **274** may be coupled to a speaker for providing audible output and to a microphone for receiving audible input, such as to facilitate a telephone conversation.

[0049] Mobile computing device **200** also includes communications connection(s), such as a wireless interface layer, that performs the function of transmitting and receiving communications. Communications connection **272** facilitates wireless connectivity between the mobile computing device **200** and the outside world. According to one embodiment, transmissions to and from communications connection **272** are conducted under control of the operating system **264**.

[0050] **FIG. 8** is a functional block diagram generally illustrating an location aware development system **800**, in accordance with aspects of the invention. Server **810** and computing device **830** are computing devices such as the one described above in conjunction with **FIG. 1** and mobile device **820** is a mobile computing device such as the one described above in conjunction with **FIG. 2**.

[0051] Typically a user develops a location aware applications using a computing device, such as computing device **830**. The location aware application, however, may be developed using a mobile device, such as mobile device **820**. Location aware programming environment **832** is configured to provide a developer with a set of programming interfaces to develop a location aware application. Application **832** is configured to communicate with server **810**. According to one embodiment, server **810** is configured to provide computing device **830** with location information from location store **812** and map information from map store **813**. Server **810** or a mobile device may also be configured to simulate a location source while developing a location aware application. For example, the simulated location source can run on the mobile device and simulate the input of a GPS source and the like. The computing devices may communicate using any one of several client-server protocols.

[0052] Mobile device **820** includes location aware application **822** that has been developed using the location aware programming environment (**832**). More than one location aware application may be included on device **820**. Attached to mobile device **822** are N location sources (**821**(1-N)). These location sources, both on and off-device, as described above, provide location data to location application **822**. Location store **824** includes location information that may be accessed by all of the location aware applications on device **820**. Map store **825** includes map information that may be accessed by all of the location aware applications on device **820**. For example, location store **824** and map store **825** may include maps, rendering engines, and any other information that may be shared across the applications.

[0053] Cellular/pager network **850** is a network responsible for delivering messages to and receiving messages from wireless devices. The cellular/pager network **850** may include both wireless and wired components. For example, cellular/pager network may include a cellular tower that is linked to a wired telephone network. Typically, the cellular tower carries communication to and from cell phones, long-distance communication links, and the like. The wireless devices can also connect directly to WAN's, LANs, etc, using hardware such as Wi-Fi cards that are becoming increasingly available for mobile devices.

[0054] Gateway **860** routes messages between cellular/pager network **850** and WAN/LAN **840**. For example, a computer user may send a message that is addressed to a cellular phone. Gateway **860** provides a means for transporting the message from the WAN/LAN **840** to cellular/pager network **850**. Conversely, a user with a device connected to a cellular network may be browsing the Web. Gateway **860** allows hyperlink text protocol (HTTP) messages to be transferred between WAN/LAN **840** and cellular/pager network **850**.

[0055] The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.

What is claimed is:

1. A system for location aware applications, comprising:

a data store that includes map information and location information, wherein the map information and location information may be shared by different location aware applications on a device;

a location manager configured to receive location information from at least one location source that includes

a location of an item, and in response to the location of the item relative to an area of interest, generate a location event;

a map manager that is coupled to the data store and that is configured to abstract map information and manage map operations including displaying maps, wherein the map manager may utilize more than one map rendering engine; and

a location aware application that is coupled to the location manager and the map manager and that is configured to execute event driven code in response to the location event, and also configured to interact with the abstracted map information.

2. The system of claim 1, wherein the data store comprises a location store and a map store, wherein the location store includes information about maps and the location store includes information about areas of interest.

3. The system of claim 1, wherein the at least one location source comprises at least one of a proximate location source and a distant location source.

4. The system of claim 3, wherein the at least one location source is selected from a group comprising: a GPS receiver, a user input; a phone system; an RFID tag and location data pushed to or pulled by a device on a network.

5. The system of claim 3, wherein the location manager is further configured to manage location information for 1 to N items.

6. The system of claim 1, wherein generating the location event comprises generating a location event in response to at least one of the following events: a determination that the location of the item has entered the area of interest, a determination that the location of the item has exited the area of interest, and a determination of the movement of a location source inside an area of interest.

7. The system of claim 1, wherein the location aware application is further configured to provide the map manager with a suggested map resolution.

8. The system of claim 1, wherein the map manager is further configured to receive a user input and convert the user input into a location on a map.

9. The system of claim 7, wherein the location aware application is further configured to supply the map manager with at least one object to display on a map.

10. The system of claim 9, wherein the map manager supplies a set of maps to the location aware application, and wherein the location aware application determines the map to display from the supplied set of maps based on weighted criteria.

11. The system of claim 1, wherein the data store is further configured to be dynamically provisioned.

12. The system of claim 11, wherein the map information includes a description of each map using an XML representation.

13. The system of claim 11, wherein the description comprises a map projection; a resolution; an indication of the map rendering engine used to display the map and an area the map includes.

14. A computer implemented method for location aware applications, comprising:

determining when an item enters, exits and moves inside an area of interest;

generating a location event in response to the determination;

executing event driven code supplied by a location aware application in response to the event;

selecting a map from a map store, wherein the map may be shared by other location aware applications; and

displaying the map using a map rendering engine, wherein the map rendering engine is selected from at least two map rendering engines.

15. The method of claim 14, further comprising defining areas of interest, wherein the area of interest is a predefined area on a map.

16. The method of claim 15, wherein determining when the item enters and exits an area of interest comprises obtaining location information from at least one location source and determining whether the item is within the area of interest.

17. The method of claim 15, wherein generating the location event comprises generating a location event in response to at least one of the following events: a determination that the location of the item has entered the area of interest, a determination that the location of the item has exited the area of interest and a determination that the location of the item as moved a specified distance inside an area of interest.

18. The method of claim 15, further comprising supplying at least one object to the map rendering engine to overlay on the map.

19. The method of claim 15, further comprising dynamically provisioning the map store.

20. The method of claim 15, wherein the map includes a description of the map using an XML representation.

21. The method of claim 20, wherein the description comprises a map projection; a resolution; an indication of the map rendering engine used to display the map and an area the map includes.

22. A computer-readable medium having computer-executable instructions for location aware applications, the instructions comprising:

defining areas of interest;

obtaining a location of an item from a location source and storing the location in a location store, wherein the location store may be shared by other location aware applications;

determining when the item is within one of the areas of interest;

generating a location event when it is determined that the item is within one of the areas of interest;

executing event driven code associated with a location aware application in response to the location event;

selecting a map from a map store, wherein the map may be shared by other location aware applications; and

displaying the map using a map rendering engine, wherein the map rendering engine is selected from at least two map rendering engines.

23. The computer-readable medium of claim 22, further comprising defining the areas of interest after the location aware application is deployed.

24. The computer-readable medium of claim 23, further comprising determining when the item is outside one of the areas of interest, and in response to the item being outside, generating the location event.

25. The computer-readable medium of claim 22, further comprising supplying at least one object to the map rendering engine to overlay on the map.

26. The computer-readable medium of claim 22, further comprising dynamically provisioning the map store.

27. The computer-readable medium of claim 22, wherein the map includes a description of the map using an XML representation.

28. The computer-readable medium of claim 27, wherein the description comprises a map projection; a resolution; an indication of the map rendering engine used to display the map and an area the map includes.

29. The computer-readable medium of claim 22, wherein selecting the map from the map store comprises determining a best fit map in response to weighted criteria received from the location aware application.

30. The computer-readable medium of claim 22, wherein displaying the map using the map rendering engine comprises switching between map rendering engines depending on a type of map being rendered.

31. The computer-readable medium of claim 30, wherein displaying the map using the map rendering engine comprises selecting a projection technique depending on the type of map being rendered, wherein the projection technique is selected from at least two projection techniques of: Mercator, Polar, Grid, and Custom.

* * * * *