service module can use the information to determine a current device context.

In the case where the context of the device is the device's location, sources **606** provide various information to the location providers **604** that pertains to the device's current location. As an example, the sources of the information can include various information transmitters such as a GPS system, cell phone or cell ID, wireless transmitters that transmit location information, location beacons, 802.11 transmitters and various other sources of information. The sources of information can also include other computing devices that might, for example, provide location information through Bluetooth technology. In addition, a source of information **606** might include a person who, for example, physically enters location information into the device **600** so that the device can process the information to determine its location.

When the device **600** receives the location information from the sources **606**, it processes the information with the location providers **604** and provides the information to the location service module **602**. The location service module **602** processes the location information and determines a particular node on one or more of the hierarchical tree structures to which it has access which corresponds to its current location. The location service module **602** can then traverse the tree structures to determine a complete location for the device. Once the complete location is determined, the device **600** can begin to interact with one or more applications **608** that can query the device about its particular location so that one or more location-dependent services can be rendered to the device. In this example, the applications **608** are illustrated as being separate from the device. It is to be understood, however, that the applications could be executing on the device, e.g. a browser application.

As shown, the applications **608** can make calls to the device to ask the device where it is located. The device is configured to receive the calls and respond in an appropriate manner to the application. Once the application has the device's location information, it can then render location specific services to the device.

Consider the following example: You are a traveler and have a hand-held mobile computing device that contains a Master World tree and a Secondary World tree for SeaTac International Airport. You are scheduled to depart on a plane for China from Concourse C. SeaTac International Airport has designed its Secondary World to have the following nodes: "Arrivals", "Departures", "Concourses", "Airlines", "Gates assigned to Airlines", and "Gate Location". When you arrive at the airport, as you enter the airport your computing device receives location information from different sources and with that information your device determines that your location is in the Arrivals node. SeaTac International has bank of servers that are executing applications to assist you while you are in the airport. There are applications that can help you find services, locate facilities (e.g. coffee shops, restaurants), give directions (e.g. how to get to your departure gate), update you on the status of your flight, and even check you in automatically for your flight. Consider also that as you walk through the airport your location changes. Your computing device, however, can receive continuous location information updates so that it can continue to determine its location as you move through the airport. At one point, as you pass a Starbucks coffee shop, your hand held device notifies you that if you purchase a latte at Starbucks and present your hand held device, you will receive a 50 cent discount on your latte. In this example, the utility of the Secondary World is demonstrated. By knowing where its particular customers are in its facility, SeaTac International is able to provide a host of services that were not possible before.

Assume further that you are in the airport and your flight is canceled. You must find a place to stay for the night. Accordingly, you wish to determine the closest Double Tree hotel because you really like the warm chocolate chip cookies they give you when you check in. You execute a search engine on your computing device to find the nearest Double Tree hotel. The search engine application first determines your current location in the Master World as indicated by the EID of the Master World node that corresponds to your location. Executing a search, the search engine application looks for a Double Tree hotel that has an attribute that includes an EID that matches your EID. If it finds one, it simply indicates for you the result. If it does not find one with the corresponding EID, it can use an adjacent geozone to search for a Double Tree hotel. It may also provide driving directions to the hotel. The search engine application was able to do this because it was able to ascertain your location in the Master World. It did this quickly and automatically with little or no effort from you.

Consider further that as you are driving from the airport to the hotel you decide that you want to find the nearest Kinko's so that you can print 100 copies of a presentation that you are to give in the morning. Consider that your hand-held computing device is a cellular phone and that Sprint is the carrier. Sprint has defined its own Secondary World that might, for example, be designated in terms of cell nets. By virtue of having Sprint's Secondary World on your computing device, you are able to ascertain your location in Sprint's Secondary World and, accordingly, your location in the Master World. Consider that Kinko's also has a Secondary World that links with the Master World. By executing a search application on your device, you are able to ascertain the location of the nearest Kinko's as well as driving directions thereto. All of this is possible because your device has access to the Master World and one or more Secondary Worlds. In this example, the Master World provides a mechanism to daisy chain two or more Secondary Worlds together. This is possible because the Secondary Worlds have at least one reference or link into the Master World.

Exemplary Device Architecture

FIG. **7** shows computing device **600** in somewhat more detail. In this particular embodiment, device **600** comprises an architecture that includes the following components: a location service module **602**, a location provider interface **700**, an application program interface (API)/Events module **702**, a privacy manager **704** a location conversion module **706**, one or more applications **608** and one or more location providers **606**. Also included in the architecture is an active directory **708**, Web service **710**, location database **712**, and personal places **714**. The architecture can be implemented in any suitable hardware, software, firmware or combination thereof. The architecture mentioned above is advantageous in that it enables each computing device to determine its own context or location.

Common Location Provider Interface

One particularly advantageous aspect of the described embodiment is that it employs a common interface **700** that provides a standard interface through which the location providers **606** communicate. By having a common interface, the location providers are extensible (to support future providers) in that they can be dynamically added or removed from the collection of location providers. All that is required of a particular location provider **606** is that it be written to support the common interface.

In this example, there are several location providers **606**. These location providers provide location information in different forms. For example, a GPS location provider might provide location information that is GPS specific. Similarly, an IP/Subnet location provider might provide information that is specific to an Internet Protocol. A mobile phone location provider might provide location information in the form of a cell ID. In addition, a location User Interface (UI) might provide location information in the form of a user entry that specifies a city, street or building. All of the location information that is provided by the various location providers is processed by the location service module **602** so that a current device location can be determined. To determine the current device location, the location service module **602** may have to consult with an active directory **708**, a Web service **710**, or a location database **712**. In the illustrated example, the active directory **708** might, for example, maintain a secondary world and other networking metadata such as subnet and "site" information that can help determine location based on networking connectivity. Web service **710** can hold the master or secondary worlds, the attributes of which can be used to find location. For example, if a cell phone knows its cell tower ID, then the location provider can query the secondary world to ascertain the nodes that match that cell tower ID. Location database **712** is basically a version of the web service that is hosted or cached locally.

Location Providers

As indicate above, the architecture contemplates multiple different location providers that can provide location information to the location service module **602**. This information can come in many different forms and quality levels. The information is then processed by the location service module **602** to determine a current device location. To do this, the service module **602** ascertains from the location information a particular node ID (EID and/or LUID) and a URL that is associated with the tree structure with which the node is associated. Once the location service module ascertains a node ID, it can then query the tree structure (or more accurately a server that manages the tree structure) using the URL to ascertain more information about the tree structure. For example, if the location service module **602** ascertains a LUID from a particular Secondary World, it might then query an active directory **708** (or an Intranet server—which is another location database) to discover the parents and the children of the node. This would then enable the location service module to build the Secondary World.

The location providers **606** can provide the location information to the location service module **602** in many different ways. For example, some location providers **606** may continuously provide information (e.g. the GPS provider may continuously provide GPS coordinates). Alternately, the location providers can periodically provide location information such as at specific times or on the occurrence of definable events. For example, a user may define specific times when the location information should be updated. Alternately, the location information might be updated only when a device's location changes (i.e. a location change event). Additionally, the location providers might provide location information when polled by the location service module **602**. For example, the location service module **602** can call the location provider interface **700** and request location information from one or more of the location providers.

One specific location provider **606** is shown as a cache. The cache provider essentially maintains a current device context or location. That is, once the location service module **602** has ascertained its current location, it writes this location to a cache. This enables the device **600** to ascertain its location with a degree of confidence in the event all of the other location providers are not able to provide location information (e.g. the GPS provider may not receive GPS information because the GPS transmitter that supplies it with the information is unable to contact a requisite number of satellites).

Confidence and Accuracy Parameters

One important and useful feature of the described embodiment is that one or more of the location providers are configured to assign confidence parameters and/or accuracy parameters to the information that they provide to the location service module **602**. Confidence parameters provide a measure of a provider's confidence in the information that it provides to the location service module **602**. For example, assume that a GPS transmitter must receive information from five or more satellites in order to provide highly confident information. Assume that only three satellites are available at the time. The GPS transmitter would then transmit its information based only on the three satellites. The GPS provider would then know that the information it receives from the GPS transmitter was based only on three satellites rather than the desired five or more. In this case, the GPS provider can set a confidence parameter on the location information that indicates that it has a lower confidence level than if the information were based on the desired five or more satellites. In this case, the location service module **602** can take the confidence parameters for all of the location providers into account when determining the location of the device. This is discussed in more detail below.

With respect to the accuracy parameters, consider that the location information that is received from the location providers is accurate to varying degrees. Some information may be accurate to within one mile, while other information may be accurate to within **100** feet. The location providers are desirably configured to assign accuracy parameters to the location information that they provide to the location service module **602**. The accuracy parameters give the location service module an indication of the accuracy of the information.

When the confidence and accuracy parameters are used by the location service module **602**, the module can make decisions on how to use the location information it receives from each provider. For example, the location service module **602** might disregard completely any information that has a low confidence parameter. It might, on the other hand, strike a balance between the accuracy of the information and its confidence. For example, the module **602** might be programmed to use information with lower levels of accuracy only when there is a high level of confidence in the information. The module **602** might utilize the parameters to assign weights to the information so that the location is calculated as a weighted function of the confidence and accuracy of the information.

Another use of the confidence parameters is as follows: Assume that the location service module has determined a device location and has written that location to a cache. At the time when the location is written to a cache, it is assigned perhaps a high confidence level. Assume further that all of the other location providers are unavailable to provide location information. For a period of time, the location service module **602** can use the cache location as a current location and be fairly confident that its information is generally accurate. In this case, the location service module might assign a linearly decreasing confidence level to the information over time so that at some point, it ceases to use the information or informs the user that the information cannot be guaranteed.

Location, Trust, and Timestamp

When the location providers provide their information to the location service module **602**, the information can include, in addition to the confidence and accuracy parameters, the actual location information in a known format, a trust parameter and a timestamp. The trust parameter is a metric that is assigned by the location service module **602** to one or more of the location providers and defines the trust that the location service module has for the particular location provider. The timestamp is a metric that defines the time when the location information was provided by the location provider. This assists the location service module **602** in ascertaining whether information is stale and might need refreshed.

Once the location service module **602** has all of the location information, it can then set about determining the location of the device.

FIG. **8** is a flow diagram that describes steps in a method of determining a device context which, in this example, is the device location. These steps are implemented by the location service module **602**.

Step **800** gets the current device context. The current context can be the last calculated device context that is stored in the cache. Step **802** determines whether any of a number of context providers are available to provide context information. The location service module might do this by polling the context providers to ascertain which of the providers are active and valid. Step **804** determines whether all of the providers are inactive. If all of the providers are inactive, step **806** decreases the confidence in the current context over time and uses the current context as the device context. Step **802** then continues to monitor for current active and valid providers. If step **804** determines that one or more of the context providers are active, then step **808** orders the active and valid context providers. When the location service module **602** orders or sorts the context providers, it does so as a function of the confidence of the provider's information and/or the trust that the location service module has in the location provider. This provides a ranked list of the location providers. Step **810** checks to ascertain whether the context information appears to be correct. For example, where the context is the location of the device, the location service module **602** might know that five seconds ago the current location was Redmond, Wash. Accordingly, location information that indicates that the current location is Beijing, China would be incorrect. Step **812** then determines whether any of the context information conflicts with either the device's current context or the context information from other providers. For example, the location service module **602** can compare the context information from each of the context providers with the information in the cache. If any of the information conflicts with the cached information, then the information from that context provider can be discarded. Similarly, if context information varies inordinately as between the context providers, then step **814** can select the context providers having a predefined level of trust and perhaps use just their information (Step **816**). If there are no conflicts, then step **816** determines the current context based upon the information that is provided by all of the context providers. In the described embodiment, this step is implemented by using the information to map to a particular node in one or more of the hierarchical tree structures mentioned above. For example, the location of the device can be ascertained by mapping the information to a particular node, and then completely traversing the tree structure until the root node is reached. Step **818** then updates the current context by perhaps writing it to the cache and returns to step **802** to determine the active and valid context providers.

The method described above provides a way for the location service module to receive location information and use only the location information that appears mostly likely to represent a current location. Conflicting information can be discounted or disregarded thereby assuring that only the most trusted, accurate and confident information is utilized to determine the device's current location.

Self Monitoring

In addition to the confidence and accuracy parameters, one or more of the location providers are advantageously programmed to self monitor their own operation for various irregularities that can occur. On the occurrence of an irregularity, the location providers are configured to notify the location service module **602**. For example, the source from which the location provider receives its information may go off line for a period of time so that the location provider is unable to receive any additional information. In this case, the location provider might generate a "provider out" message and send it to the location service module **602**. When the location service module **602** receives the "provider out" message, it can then take steps to exclude the location information from that provider from any location calculations that it performs. When the location provider's source comes back on line, it can generate a "provider on" message that informs the location service module **602** that it is able to transmit location information to the module. Of course, the location service module can be notified by the location providers on the occurrence of other operational irregularities, with the above example constituting but one specific case.

Applications

Once the location service module **602** has determined the device's location, it can receive queries from one or more applications **608**. In the FIG. **7** example, the applications include a web site application, an Outlook application, and a service discovery application. In the present example, the web site application can be any web site application that is capable of rendering location-specific services. For example, the user of the device **602** might access Amazon.com's web site to buy a favorite book. When the user purchases their book, Amazon.com must now compute the taxes that the user must pay. In this example, a script executing on Amazon.com's web site might query device **602** to learn of the user's location. In this particular example, the device might respond to the query by returning the state in which the user is making the purchase. Amazon.com can then assess the tax automatically. Amazon.com might also desire to know where the individual is located so that they can select an optimal shipping method (UPS or Express Mail). Depending on where the individual is located, one method may be preferred over the other. The Outlook application might query the location service module to ascertain the location because it (or the operating system, e.g. Windows) may change device settings based on the location of the computing device. For example, the user may print on one particular printer while at work, and another particular printer when at home. When the Outlook application determines that the user has gone home for the day, it can automatically change the device settings for the printer at the user's home. It might acquire the print settings from a personal places data store **714**. Thus, the device is automatically configured for use depending on the user's location. The service discovery application might query the device to determine its location so that it can render a particular service depending on where the device is located. For example, if the user asks the application to locate the nearest color printer, the service discovery application might query the location service module to ascertain the device's current location so that it can use this information and find the nearest color printer. Consider also that the Outlook application could configure itself email to a work location (when an individual is at work) or to a home location (when an individual is at home). In addition, the Outlook calendar can become location aware, e.g. when you change time zones, your appointments would show up in the proper time slots.

As one can imagine, the possibilities are seemingly endless. This functionality is made possible through the use of the Master World and one or more Secondary Worlds.

Application Program Interface/Events

In the described embodiment, the applications **608** communicate with the location service module **602** through one or more application program interfaces (APIs) and/or events. The applications can make function calls on the API to query the location service module as to its current location. Similarly, the applications can register for location notifications by using an events registration process. For example, an application may register for a notification when the user changes their location. Consider the case where an application requests to be notified when the user arrives at work or at home so that the application can change the device's configuration (such as printer configuration).

FIG. **9** is a flow diagram that describes steps in a method in accordance with the described embodiment. The steps that are described are implemented by device **600**. Step **900** receives information that pertains to the current context of the device. In this particular example, a portion of the information is received from one or more context providers which, in this case, are location providers. Step **902** processes the information on and with the device to ascertain the current context of the device. In the illustrated example, the device maintains (or has access to) one or more of the Master World and one or more Secondary Worlds. When the device receives all of the location information, it maps the information to a particular node in the hierarchical tree structure that defines the Worlds. It then traverses the tree structures to ascertain the complete context (i.e. location) of the device. Step **904** receives calls from one or more applications that request information that pertains to the device's current context or location. In the illustrated example, the applications can call one or more APIs to request the information or the applications can register for event notifications. Step **906** then supplies the applications with at least some information that pertains to the current device location. As will be discussed below, a security policy or privacy policy can be applied to the information before it is returned to the applications.

Privacy Manager

In one embodiment, a privacy manager **704** (FIG. **7**) is provided. Although the privacy manager is illustrated as being incorporated on the device, it could be implemented by a trusted entity such as a trusted server that is not part of the mobile computing device. The privacy manager can be implemented in any suitable hardware, software, firmware or combination thereof. In the illustrated example, the privacy manager comprises a software module that is incorporated in the mobile computing device.

The privacy manager **704** addresses privacy concerns that are associated with the information that is collected by the computing device. Specifically, the location service module can calculate detailed information regarding the location of the computing device. It may be desirable, in some instances, to filter the information that is provided to various applications. That is, it is entirely likely that a user may not want their specific location information provided to untrusted applications. In these instances a user might just desire for location service module **602** to inform such applications that the user is in the State of Washington.

FIG. **10** shows a flow diagram that describes steps in a privacy protection method in accordance with the described embodiment. These steps can be implemented by the privacy manager **704**.

Step **1000** defines a plurality of privacy levels. Exemplary privacy levels are set forth in the table immediately below:

Privacy Level	Approximate Scale	Level of Revelation
0	—	No location information is returned
10	100,000 Km	Planet/Continent
20	1,000 Km	Country
30	100 Km	State
40	10–100 Km	City & County or Region
50	10 Km	Postal Code & Phone Area Code
60	1 Km	Full Postal Code (Zip + 4) & Area Code and Exchange
70	100 m	Phone Number & Building/Floor
80	10 m	Room #
90	1 m	Exact Coordinates

In the illustrated table, 10 different privacy levels are defined and each has an associated approximate scale. For example, a privacy level of 0 means that no location information is returned. A privacy level of 90 means that very detailed location information is returned.

Step **1002** assigns various privacy levels to the individual nodes in one or more hierarchical tree structures. For example, each node of the Master World and the Secondary Worlds can have a privacy level associated with it. The root node of the Master World tree structure might have a privacy level of 10, while the node that represents a current location in a Secondary World might have a privacy level of 90. Step **1004** determines the context of the computing device. In the present example, the context is the device location and examples of how this is done are given above. Individual applications that call the location service module can have privacy levels associated with them. These privacy levels can be assigned by individual users. For example, a trusted application might have a privacy level of 90, while an untrusted application might have a privacy level of 30. Step **1006** receives context queries from one or more applications. Here, an application calls the location service module **602** (FIG. **7**) to ascertain the location of the device. Step **1008** determines the privacy level associated with the application or applications. For example, if a untrusted application calls to request location information, the privacy manager **704** would determine that the application has a privacy level of 30. The privacy manager then traverses (step **1010**) one or more hierarchical tree structures to find a node with a corresponding privacy level so that it can select the information that is associated with that node. In this example, the traversal might involve jumping from the

Secondary World to the Master World to find the node that corresponds to the state in which the user is located. Once the corresponding node is found, step **1012** returns the context information (e.g. location information) associated with the node. In this case, the location service module would inform the application that the user's location is the State of Washington.

As an example, consider the following: There is a web site that gives up to the minute weather of various locations. Accordingly, you might assign this web site a privacy level of 60 so that you can receive weather information for the geographical area that corresponds to your present full postal code. Another web site might be a corporation intranet web site that is a trusted web site. Thus, any applications associated with this web site can be assigned a privacy level of 90 so that you can give them precise location information as to your whereabouts.

Thus, in the present example, the computing device is able to determine the source (i.e. application) of its queries and modulate the information that is returned to the application as a function of the application's identity. The computing device is able to do this because it has access to the Master World and one or more Secondary Worlds. The above description constitutes but one exemplary way of accomplishing this feat.

Location Beacons as a Location Provider

In one embodiment, one of the location providers comprises a location beacon that beacons or transmits information to enable a computing device to actively participate in its current context. Location beacons can comprise standalone devices that can be retrofitted onto existing infrastructures, e.g. a smoke detector or wall outlet in order for the device to have a power source.

FIG. **11** shows an exemplary beacon **1100** that is mounted on a structure **1102**. Structure **1102** can be any suitable structure such as a wall in a conference room or public place, a smoke detector, an electrical socket and the like. In the described embodiment, the location beacons are small inexpensive devices that can be permanently mounted in special locations such as conference rooms, building lobbies, airport gates, public places and the like. The beacons announce the physical location in the form of an EID and/or LUID to all mobile devices that are within range, such as laptops, tablet PCs, hand held computers, mobile phones, wearable computers and the like.

In the described embodiment, the location beacon can identify the particular locations by beaconing standard information that will be understood by the mobile computing devices. In the present example, the beacons can transmit one or two location identifier pairs comprising an EID/URL pair and a LUID/URL pair. The beacon might also transmit multiple LUIDs. The EID and LUID give the present node location in the Master World and Secondary World respectively. The URLs provide a reachable location for the Master and Secondary Worlds. For example, the URL associated with the Secondary World can give a service location that the device can use to query information about the Secondary World so that it can derive its context and take advantage of resources or services that are associated with the nodes in the Secondary World.

The beacons can also transmit a digital signature that can be used by the device to ascertain that the beacon is valid and legitimate. Any suitable signature or verification method could be used. In addition, and of particular use in the context-aware environment, the beacon can be programmed to transmit code download pointers to devices within range. The code download pointers can g enable the computing device to access software code that permits them to interact with their environment. Consider the following example: You walk into a conference room with your cell phone computing device and immediately a ii beacon in the conference room transmits your location in the form of an EID/URL pair and a LUID/URL pair. Your device uses the information pairs to ascertain its location in the Master and Secondary Worlds as described above. The beacon also transmits a code download pointer that points to software code that enables you to operate the video projector in the conference room using your hand-held cellular phone. In this manner, the beacon serves as more than just a location beacon—it permits you, through your computing device, to actively participate in your surroundings.

The beacons can transmit the information in any suitable way, e.g. wireless methods including infrared and radio frequencies. In one embodiment, Bluetooth short range radio frequency communication can be used to provide a low cost, low power alternative.

Conclusion

The embodiments described above provide a uniform, standardized way to enhance the world of context aware computing. The embodiments provide a way for individuals to uniquely experience the world around them by ascertaining their location in the world in a standard way. The embodiments also provide a way for service providers to uniquely position their goods and services in a manner that is sensitive to and appreciates the contexts, e.g. locations, of various consumers of the goods and services. Unique and useful architectures and data structures are employed to facilitate the user's computing experience and provide for an individual-centric experience.

Although the invention has been described in language specific to structural features and/or methodological steps, it is to be understood that the invention defined in the appended claims is not necessarily limited to the specific features or steps described. Rather, the specific features and steps are disclosed as preferred forms of implementing the claimed invention.

We claim:

1. A system for determining the location of a computing device comprising:

a beacon that is mountable in a location;

the location being associated with one or more nodes on one or more hierarchical tree structures that are configured to define a complete context of the location; and

the beacon being configured to provide specific information pertaining to the one or more nodes that are associated with the location, the specific information being useable by a computing device to derive the complete context by mapping the specific information to one or more specific nodes and traversing tree structures that are associated with the nodes to derive the location.

2. The system of claim **1**, wherein the computing device is a mobile computing device.

3. The system of claim **1**, wherein the computing device is a desktop computing device.

4. The system of claim **1**, wherein the complete context of the device comprises the device's location.

5. The system of claim **1**, wherein one of the hierarchical tree structures comprises multiple nodes each of which represent a geographical division of the Earth.

6. The system of claim **1**, wherein one of the hierarchical tree structures comprises multiple nodes each of which represent a physical or logical entity.

7. The system of claim **1**, wherein:

one of the hierarchical tree structures comprises multiple nodes each of which represent a geographical division of the Earth; and

one of the hierarchical tree structures comprises multiple nodes each of which represent a physical or logical entity; and

a link that connects at least one node from each tree structure.

8. The system of claim **1**, wherein the specific information comprise a unique ID that pertains to the one or more nodes.

9. The system of claim **8**, wherein the unique ID pertains to a node of a hierarchical tree structure that comprises multiple nodes each of which represent a geographical division of the Earth.

10. The system of claim **8**, wherein the unique ID pertains to a node of a hierarchical tree structure that comprises multiple nodes each of which represent a physical or logical entity.

11. The system of claim **1**, wherein the specific information includes information that permits the computing device to ascertain more information about it location.

12. The system of claim **11**, wherein said specific information includes at least one URL that is associated with a tree structure.

13. The system of claim **11**, wherein said specific information includes a pair of URLs that are associated with different tree structures.

14. The system of claim **1**, wherein the specific information includes a unique ID for each of a pair of nodes for different tree structures, and a URL associated with each ID that provides a protocol for ascertaining additional information about an associated tree structure.

15. The system of claim **1**, wherein the specific information includes one or more code download pointers.

16. The system of claim **1**, wherein the specific information includes a signature.

17. The system of claim **1**, wherein the beacon is a program-once beacon.

18. The system of claim **1**, wherein the beacon is a re-programmable beacon.

19. The system of claim **1**, wherein the beacon is configured to use wireless transmissions to provide the specific information.

20. The system of claim **1**, wherein the beacon is configured to use radio frequency transmissions to provide the specific information.

21. The system of claim **1**, wherein the beacon is configured to use short range radio frequency transmissions to provide the specific information.

22. A system for determining the location of a computing device comprising:

one or more locations each of which being associated with a node on a hierarchical tree structure that comprises multiple nodes that are connected together to define a complete location;

a beacon mounted at each location and configured to transmit node-specific information that can be used by a computing device to derive the device's complete location.

23. The system of claim **22**, wherein the multiple nodes of the hierarchical tree structure represent geographic divisions of the Earth.

24. The system of claim **22**, wherein the multiple nodes of the hierarchical tree structure represent physical or logical entities.

25. The system of claim **22**, wherein the node-specific information comprises a node ID and a URL that identifies the tree structure with which the node is associated.

26. The system of claim **22**, wherein the node-specific information comprises a code download pointer.

27. The system of claim **22**, wherein the node-specific information comprises a signature.

28. A system for determining the location of multiple computing devices comprising:

one or more computing devices each of which being configured to access one or more hierarchical tree structures and traverse the tree structures to ascertain the complete location of the device;

one or more beacons mounted in various locations that are each associated with a node on a hierarchical tree structure to which the one or more computing devices have access, the one or more beacons being configured to transmit node-specific information concerning the node with which it is associated, the one or more computing devices being configured to receive the transmissions and ascertain a device location by mapping the node-specific information to a node and traversing the associated hierarchical tree structure to ascertain the complete location of the device within the hierarchical tree structure.

29. The system of claim **28**, wherein at least one beacon is incorporated into existing infrastructure.

30. The system of claim **28**, wherein the computing devices comprise mobile computing devices.

31. The system of claim **28**, wherein the computing devices comprise desktop computing devices.

32. The system of claim **28**, wherein the computing devices comprise hand-held computing devices.

33. The system of claim **28**, wherein the one or more hierarchical tree structures comprises a tree structure having multiple nodes each of which represent a geographical division of the Earth.

34. The system of claim **28**, wherein the one or more hierarchical tree structures comprises a tree structure having multiple nodes each of which represent a physical or logical entity.

35. The system of claim **28**, wherein the node specific information comprises a unique ID and a URL that is associated with the tree structure with which the node is associated.

36. The system of claim **28**, wherein the node-specific information comprises a code download pointer.

37. The system of claim **28**, wherein the node-specific information comprises a signature that identifies the beacon.

38. A method for determining the location of a computing device comprising:

transmitting information from a beacon, the information uniquely identifying a node of a hierarchical tree structure having multiple nodes, each node representing a location;

receiving the information with a computing device;

using the computing device, mapping the information to a node of the hierarchical tree structure; and

ascertaining the location of the computing device using the hierarchical tree structure.

39. The method of claim **38**, wherein the computing device comprises a mobile computing device.

40. The method of claim **38**, wherein the transmitting comprises transmitting radio frequency information.

41. The method of claim **38**, wherein the transmitting comprises transmitting infrared information.

42. The method of claim **38**, wherein the transmitting comprises wirelessly transmitting information.

43. The method of claim **38**, wherein the information comprises a URL that is associated with the tree structure and which can be used to ascertain additional information about the tree structure.

44. The method of claim **38**, wherein the information includes a code download pointer.

45. The method of claim **38**, wherein the information uniquely identifies two nodes, each node being associated with a different hierarchical tree structure.

46. The method of claim **45**, wherein one of the hierarchical tree structures includes multiple nodes each of which represent a geographical division of the Earth.

47. The method of claim **45**, wherein the information comprises a URL that is associated with each node and which can be used to ascertain additional information about the tree structure with which it is associated.

48. One or more computer-readable media having computer readable instructions thereon which, when executed by one or more computers, implements the method of claim **38**.

49. A system for determining the location of a computing device comprising:

one or more beacons that are mountable in a location;

the location being associated with one or more nodes on a hierarchical tree structure that is configured to define a context of the location; and

the beacon being configured to provide specific information pertaining to the one or more nodes that are associated with the location, the specific information being useable by a computing device to derive the context of the location by mapping the specific information to one or more specific nodes and traversing the hierarchical tree structure that is associated with the nodes to derive the context.

50. The system of claim **49**, wherein the nodes on the hierarchical tree structure can represent a logical entity.

51. A method for determining the location of a computing device comprising:

receiving information with a computing device, the information having been transmitted from a beacon, the information uniquely identifying a node of a hierarchical tree structure having multiple nodes, each node representing a location;

using the computing device, mapping the information to a node of the hierarchical tree structure; and

ascertaining the location of the computing device using the hierarchical tree structure.

52. The method of claim **51**, wherein said ascertaining comprises traversing two or more nodes on the hierarchical tree structure.

53. One or more computer-readable media having computer readable instructions thereon which, when executed by one or more computers, implements the method of claim **51**.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 6,327,535 B1
DATED : December 4, 2001
INVENTOR(S) : Evans et al.

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 7,
Line 28, add -- LUID -- before "indicates".

Column 12,
Line 51, add -- the -- before "Master".

Column 16,
Line 51, delete "g" after the second occurrence of "is".

Column 26,
Line 1, delete "g" after "can".
Line 5, delete "ii" before "beacon".

Signed and Sealed this

Seventeenth Day of September, 2002

Attest:

JAMES E. ROGAN

Attesting Officer

Director of the United States Patent and Trademark Office