A COMMUNITY LOCATION DIRECTORY

BACKGROUND OF THE ART

Location specific information, when it is available, can be very useful for navigation, event monitoring, social interactions, and venue locating. Some existing mobile stations—such as vehicles with navigation computers, mobile phones, personal digital assistants and handheld map devices—have the ability to fix their location using real-time geopositioning. When combined with previously gathered location-specific data, users can obtain maps, directions to and from their current location, and information on nearby venues.

One mechanism for delivering location-specific information periodically delivers a collection of information from a centralized location-specific directory to the mobile station, as a one-time or updatable process like purchasing a new map. This delivery mechanism has been used to provide maps or points-of-interest for navigating the highway system, or locating venues along a travel path.

An ideal situation arises when a mobile station with positioning also contains the ability to communicate data with the Internet. In this case, large databases and stationary computing resources can be accessed on-demand through the Internet, using the mobile station solely for fixing its geographic position and providing a user-interface. For example, nearby movie locations and times can be sent to a mobile phone through the Internet when a subscriber wants to go to a movie, saving the phone from keeping a full database of all movie locations and times.

At present, location-specific data is typically compiled by a ‘monolithic organization,’ i.e., one that makes many controlled changes to the database before releasing it for use. Trained database entry and cartography staff may enter data into the database, or bulk-load external data purchased from a known source. Untrained individuals typically are not allowed to enter or augment the data in location databases.

This leads to five problems: First, location databases created by monolithic organizations (henceforth called monolithic location databases) are expensive to create and maintain. Trained staff, once they know the names and locations of points-of-interest (POIs) in a region, may rapidly enter accurate location information about that region. However, if a trained staff member does not frequent a region, they must take the time to learn about those regions—and the time required to learn about the region costs money.

Second, monolithic location databases go out of date rapidly. Roads have temporary detours, new businesses are established, existing businesses move or close, concerts and conferences exist only briefly in a location. Unless many regionally distributed staff members work diligently to maintain and release the database to subscribers, much of the location information in the database may go out-of-date soon after it is shipped. In some cases, the location-specific information is so ephemeral, such as detours, that monolithic location database suppliers don't bother trying to supply it.

Third, monolithic location databases tend to be inaccurate, because the experts entering the data have no local stake in the accuracy of that data.

Fourth, monolithic location databases tend to provide only limited information. Operating hours, movie times, band line-ups, game schedules and other information are typically not provided by location databases. The cost of acquiring such information typically outweighs the value which the database provider can obtain.

Fifth, monolithic location databases tend to provide information of uniform quality and value regardless of whether the location is heavily or lightly used, because staff members may not understand local behavior, travel patterns, or interests.

As mobile stations increasingly include positioning capabilities, consumers will demand less expensive, more accurate, more up-to-date, more informative and more relevant location information. The monolithic model of location-based data cannot feasibly accommodate their needs.

Summary of the Invention

The invention is a collection of technologies that enable and help untrained users to enter, update, validate, correct and delete data in a community location directory.

One embodiment of the invention is a simple community location directory that could be built using a database server, a database table storing each venue in the directory, an API allowing programmatic access, and an appropriate user interface allowing users to enter, find, correct and delete location information from the database when they are at or near that location, or afterwards. Such a system would provide basic functionality, but would not distinguish between good, malicious and sloppy contributors, so the overall quality of the database may become low.

Another embodiment of the invention is the first embodiment with the addition of a user table, and the addition of a column in the venue table indicating which user contributed the last venue update. A user description could contain a histogram of the number of corrections against the user's contributions ordered by the time of the contribution, and another histogram of the number of contributions the user made ordered by time of the contribution. When the ratio of corrections to contributions in a time period exceeds a maximum error rate, such as shown in Figure 1, the user may be prevented from creating new entries or modifying existing entries in the public database.

Another embodiment of the invention is the first embodiment with the addition of a user table, and the addition of a column in the venue table indicating which user owns the entry if any. When a user owns the entry, it becomes a private placemark available for use by that specific user, may be shared with friends, or may be consulted by emergency personnel to help locate that user in an emergency. The private placemark can be linked to the public directory if desired. A benefit to the person who shares their placemarks is that changes to the information in the public directory are reflected in their placemark. If no one owns the entry, it is publicly shared. A user can edit and remove their private placemarks at will.

Another embodiment of the invention is a system that tracks the user's location periodically, and when the user is in the same location for longer than some time limit, the system automatically prompts the user to enter the location into the community location directory, either as a public venue or a private placemark.

Another embodiment of the invention is the first embodiment, with the ability to differentiate between information relating to stationary entities with regular 'hours' and events occurring at those entities.

Another embodiment of the invention is a system where the user is a device, such as a vehicle, and constructs a directory that consists of routes of those devices and events that can be identified with respect to those devices owing to the nature in which they are used, such as a vehicle stopped in an area already known to be a freeway, or moving in the wrong direction. Such an embodiment would not need the notion of reputation or even a user interface.

Another embodiment of the invention is the first embodiment, with the ability to enter a location region using satellite imagery, a graphical display and a pointing device.

A community location directory would be less expensive to maintain than a monolithic location directory. Since users gain value, and possibly positive recognition, from a locally accurate location directory, they are naturally motivated to enter data into the directory, in some cases without monetary compensation.

Since users have more local expertise, a community location directory could feasibly include more complete and timely information, such as when businesses are open, when events might occur, particular acoustic aspects of a venue, etc. This can make a community location directory more valuable than a monolithic location directory.

A community location directory can augment existing data from an existing monolithic location directory. For example, the locations and names of businesses can come from the monolithic location directory, while the operating hours can be provided by users. A monolithic location directory could provide the initial seed data for a community location directory, and then users can correct that data as it goes out of date. A monolithic location directory could

provide commercial or military information, while users augment it with information relating to community assets, such as recreation facilities or community events.

A community location directory could also include location specific information that is only personally relevant, such as "when mobile user Joe is in the vicinity of 330 Townsend Street, mobile user Joe is likely to be in Suite 209." Such personal "placemarks" would allow users to find personal resources easily. Existing positioning systems, such as GPS, are not accurate enough to locate a person within a large building, or even designate a specific building when the buildings are small. So a person's placemarks could also help emergency personnel, friends and service providers locate that person.

One way to construct a community location directory uses mobile stations as input devices. The Wireless Public Safety Act of 1999, also called E911, mandated that wireless operators be able to locate a mobile station within 100 meters of its actual position 90% of the time. To satisfy this mandate, the mobile station must become a data source that can accurately report its longitude and latitude. In practice, assisted GPS (A-GPS) equipped phones can locate themselves with 10 to 20 meter accuracy. This frees the mobile station user from typing in location landmarks, such as nearby cross-streets or a specific address, making it more likely that a user would be willing to enter information.

Some wireless devices can be monitored periodically to determine their location. One aspect of this invention is a system that prompts a user to enter location data when they have remained stationary for longer than a specified time in an area they've never visited before, or in an area where there is no information. This would free users from having to remember the areas they have previously entered, or having to check to see if the area they are visiting has already been entered into the database.

When any user can contribute entries in a community location directory, malicious or sloppy contributors can introduce errors. Therefore, there is a need for other users to verify and correct public information stored in the community location directory. Furthermore, when a contributor has had numerous corrections applied to his contributions, the system should limit

that contributor's impact on the directory to preserve its overall quality. Finally, when the system has limited a contributor's ability to enter new data into the directory, there should be a mechanism for a contributor to redeem him or herself by making contributions verified by others to be accurate.

Consequently, users need a means to easily and accurately input, verify and change location-related information in a location directory, to share such information publicly, to create and manage their own personal placemark system that may be accessed by others in a controlled way, to help determine the quality of different contributors to the location-directory, and to limit directory modifications that arise from malicious or sloppy contributors. Such a directory can be used to find an individual within a building, to find temporary venues or venues that are open at specific times, to find new businesses, and avoid finding closed businesses, to get temporary routing information related to detours or congestion, and to inexpensively create location directories for general use.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 depicts a submission history and error rate threshold.

Figure 2 depicts a high-level architecture for a mobile location directory.

Figure 3 depicts a possible architecture of a community-contributed location directory.

Figure 4 depicts a possible set of data objects used to implement the system.

Figure 5 depicts a partial hierarchy of venue categories useful in developing a location-specific directory classification system.

Figure 3 depicts a submission histogram used to determine user reputation.

Figure 4 depicts an error histogram used to determine user reputation.

Figure 5 depicts a decay rate used to determine user reputation.

Figure 6 depicts an error rate histogram and threshold curve used to determine user reputation.

Figure 7 depicts a high-level implementation architecture.

Figure 8 depicts a process flow for implementing the functional components of the location directory.

Figure 9 depicts a user interface sequence associated with reputation during entry submission.

Figure 10 depicts a user interface sequence for retrieving entry data from the location directory.

Figure 11 depicts a user interface sequence for deleting entry data from the location directory.

Figure 12 depicts a user interface sequence for inserting entry data into the location directory.

Figure 13 depicts a user interface sequence for verifying entry data is unique in the location directory.

Figure 14 depicts a user interface sequence for modifying entry data in the location directory.

Figure 15 depicts a user interface sequence for correcting entries to the location directory.

I. OVERVIEW AND GENERAL ARCHITECTURE

The invention refers to a community directory of location-specific and temporally appropriate information. The term “community information” refers to information that is provided by end-users rather than service personnel hired to gather, edit, or enter location-specific information. Therefore the invention includes additional facilities to classify new entries provided by end users, generate personal placemarks of individual users, validate new entries, modify, correct and delete private and public entries, and control the ability to contribute to and make corrections to location entries based on the quality of contributors. These components make it more feasible for untrained people to create, as a community, an accurate location directory.

The following entities are relevant to the system:

Directory: Structure and related processing mechanisms used to store and retrieve information using search criteria. A directory can be implemented using a database system.

Mobile Station: A specialized mobile device with the ability to display text or graphics to a user, and the ability to communicate data by wireless means.

User: A specific person authenticated through a login procedure or through use of a personal mobile station.

Point: A longitude, latitude and optionally altitude with accuracy information.

Path: A group of connected points describing a geographic area.

Region: A bounded geographic area.

Point Of Interest: A geographic entity with a fixed position, defined as a point, a group of points, a path, or a bounded region, typically identified by name rather than by address and characterized by type that is a functional description describing a product or service, and can reached by a mobile-device user through any form of transit or locomotion.

Time Period: An extent of time set off by a start and end time or duration.

Availability: A set of time periods indicating when a point of interest is usable (could be "always").

Classification: A description for a geographic entity (such as "French restaurant"), which uses a sequence of descriptive keywords (such as "French") to describe a path through a classification hierarchy. The descriptive keywords are generally qualitative and functional.

Venue: A point-of-interest designated by a name, geographic description, availability and classification.

Event: A temporary point-of-interest that has an event name, an associated venue, a defined duration, availability, and classification.

Reputation: A measure of a user's submission history used to determine whether they can submit new venue or event entries to the directory or make corrections to other user's entries.

Using these definitions, the term "location-specific information" refers to information that describes a particular venue or event. The term "temporally-appropriate information" refers

to information that is viable for a specific time period. Venue classification follows a semantic organization and refers to a hierarchical means of ordering geographic entities using functional information about the venue. It should be noted that the term “hierarchy” relates to logical relationships and should not be construed to mean the data type or physical organization used to represent the collection of venues. Many means of representing the collection are known to those skilled in the art, and the functionality of the collection is not dependent on the data type or physical organization used to implement it.

There are eight functional capabilities used to implement a community contributed location directory of interest in this discussion: (1) insertion of new information, (2) verification of new information, (3) classification of new information, (4) modification of existing information, (5) correction of existing information, (6) deletion of existing information, (7) retrieval of existing information, and (8) control over who can submit or modify information. New information can take the form of a publicly-shared entry or a privately-held placemark. There is a need in the art to associate new location-specific information with a classification so that it can be stored, analyzed and retrieved based on its classification(s), such as finding nearby events under the classification “Dance:Hip Hop”. When two or more users provide contradictory information regarding a location, there is a need in the art to verify both the quality of the information being provided and the credibility of the user providing it. The present invention provides solutions to both of these needs.

A general architecture supporting the development of a community-contributed location directory is depicted in Figure 2.

Figure 2 shows the equipment that could be used in an embodiment of the invention. Suppose a user presses a button on Mobile Station directing an application running on Mobile Station to enter location information in the Location Directory. Mobile Station sends appropriate communication data packets through a Radio Connection to a Base Station, which then routes those packets through Switching Equipment and a Network to the Internet. Those packets can

then be directed to the Location Directory Server, which processes the packets to insert, modify, correct, or delete entries in the community location directory.

Figure 3 depicts an operating environment that is useful in the practice of the present invention. The environment can be characterized generally into three partitions: front end 300, a location data processor 302, and location back end 304. Front end 300 is the environment in which a user of the present embodiment interacts with the location data processor 302 and back end 304 via a device interface 306. Device interface 306 is connected to the location data processor 302 for data communication via connector 308. Connector 308 may be, for example, a combination of the radio connection, base station, switching equipment, and network connection depicted in Figure 2, or, for example, an internet connection.

A user often connects to a site whose interface of first impression is a mobile station interface 306. For example, a customer visiting the location page for a particular venue might first see and interact with a mobile station interface. The advantage for the site developer is that the user, in lieu of being mobile, may have location information 310, or an information request 312 that is easily handled via device interface 306.

In the operating environment that hosts the embodiment of the present invention, device interface 306 is an instantiation of a process that communicates with location data processor 302 via connector 308. Location data processor 302 itself comprises connector 308, location runtime controller process 314, and a set of location-specific programs 316. In the current embodiment, location data processor 302 is implemented with a webserver/application server combination, though many other mechanisms are known to those skilled in the art. The data storage medium as well as the memory of the computer may contain digital representations of the location directory, or parts thereof. The data storage medium may be removable. Such processing components, data storage devices, and data storage medium, are well known in the art and are not explicitly shown in Figure 3 to avoid obscuring the drawing with unnecessary detail.

As a user logs onto a site having location data processor 302, via connector 308, runtime controller 314 executes interaction routines that guide the interaction that occurs between the

user and location data processor 302. A two-way interaction occurs between the user and location data processor 302 wherein the user, in addition to the positional information provided by the mobile station, may request or provide location-specific information pertinent to that position. Location specific data 312 provided in descriptive information provided by the user are then incorporated into the backend 304. Location specific data 314 provided in response to user requests are returned. Both are mediated according to a set of location-specific programs 316. Location-specific programs 316 are created for the purpose of acquiring new information into the directory, verifying new information in the directory, classifying new data into the directory, modifying information in the directory, correcting information in the directory, determining who can insert or modify data in the directory, or retrieving information from the directory.

As the interactions between a user and runtime controller 314 continue, it is generally desirable to engage in quality control of location-specific data 312. This quality control is provided at back end 304 using location data processor 302 via an interaction loop controlled by a reputation program 318. Directory content additions and modifications are maintained over time, and this information forms the basis for accurately diagnosing the validity of submissions or corrections, and the credibility of their users. Such information may be stored electronically in a storage media or could be printed out in human readable form.

Although Figure 3 gives a general description of various operating environments in which a mobile location directory may exist, it will be appreciated that many other operating environments are obvious to those skilled in the art and that the scope of the present invention should not be so limited to the exemplary descriptions as given above.

II. COMMUNITY-CONTRIBUTED LOCATION DIRECTORY DEVELOPMENT

Entries in the community-contributed location directory 324 represent venues or events. The organization of entities in the current embodiment is depicted in Figure 4. The venue and event entries 401 are represented with a name, a location, an availability, and a classification category. Events additionally index the venue they are associated with. Location 404 has a

symbolic name, an address, a geographic description and one or more coordinates. Coordinate values are provided by geopositioning systems.

The presently described embodiment of the current invention implements a community-contributed location directory by designing a framework of venues in terms of the venue category (i.e., a qualitative description) and venue position. When a mobile user is at a location, polling his or her mobile station will provide the position of that location and a degree of accuracy for that position measurement. Requests for information relative to the user's location (e.g., for a particular venue type) are then returned to the user based on their position in relation to the position of the user's mobile station. The ordering of the data returned could be along any of several metrics, such as distance, or travel time relative to the user. Likewise, all of the venues at the requestor's position could be readily accessed. The structural foundation for this framework, and the interacting mechanisms associated with it, are described in the remaining sections.

A. HIERARCHICAL DECOMPOSITION OF THE UNIVERSE OF VENUES

In an embodiment employing the invention, all venues fit into a classification scheme referred to as the "universe of venues" (a similar classification scheme is required for events). Every new venue inserted into the directory using this scheme falls into one or more existing categories, or into an "other" category, so that it can be inserted and later retrieved based on those categories. A mechanism is required to periodically update the universe of venues so as to resolve "other" categories. The classification scheme used to develop a community-contributed location directory can be described with a hierarchical organizational structure. The hierarchy is decomposed into qualitative categories represented with keywords.

The organization of such a classification hierarchy is top-down, or general-to-specific. It can be depicted as a root-like structure with actual venues being organized at the ends of the roots.

Figure 4 depicts the universes of venues and events as a set of ontological objects 403, where each category type has a keyword value and, if the category has a parent category, indexes

that category in the ontology. A connected path through the ontology represents a venue or event category specialization. Categories 402 is a concatenation of the category values along a path in the category ontology 403 and represents the qualitative description of venues or events that have that specialization but different geographic positions.

Given a classification hierarchy that covers the breadth and depth of a universe of venues, the primary mechanism associated with the invention enables information about a particular venue or event to be inserted into the community-contributed location directory. When this hierarchy is applied to venue or event information requests, as a classification mechanism, a community-contributed location directory can locate information relevant to the requests.

Figure 5 depicts a portion of a possible category type hierarchy for a community-contributed location directory. The portion shown is intended only to illustrate the types of relationships that exist between items in the hierarchy, with the understanding that an entire hierarchy would be an exhaustive representation of keywords so that all new additions to the directory can be classified. The type of hierarchy shown in Figure 5 is an inheritance hierarchy in that items further down in the hierarchy inherit the properties of those above them. In the notation of hierarchical decompositions, every keyword represents a venue type, and a terminal in this decomposition represents a venue type that hasn't been subdivided. Venues and events are associated with terminal keywords. For example, sailboarding 571 is a terminal keyword in Figure 5. A community-contributed location directory's content is associated with its terminal categories, so all privately owned/operated sailboarding venues would have a classification with the path originating at the root of the hierarchy and terminating with sailboarding 571, for example "open.private.organized.sports.water.boardsports.sailboarding." An overall hierarchical decomposition, as previously mentioned, is called the universe of venues.

It is entirely possible for a venue or event to be multiply associated with different keyword sequences (i.e., paths through the venue/event hierarchy). For example, a restaurant might serve both Japanese and Korean cuisine, in which case it would be classified both as a Japanese restaurant and a Korean restaurant, each of which would have different categories.

They would, however, refer to the same venue, and so each venue can be described by a collection of venue categories 403. In a similar fashion, a rave event could be an event held at multiple venue types. A rave can also be associated with a music genre and a dance genre. Thus an event can have different event categories 403.

The universes of venues/events are organizational mechanisms only, and have no necessary physical relation to the storage of venues or events in the directory. As such, many classification mechanisms might be used to organize venues and events in a community-contributed location directory.

B. VENUE INSERTION IN THE COMMUNITY-CONTRIBUTED LOCATION DIRECTORY

Venue insertion is the primary mechanism that enables the community-contributed location directory to acquire new location information. The contributions are generally made by users with mobile stations, since venues and events are always associated with position, though any web-enabled device can be used in the present embodiment if the venue's geographic position is known. Contributions may be made for any venue or event. Venues and events are represented with tuples as depicted in 401. Venues are represented with at least a name, a geographic description, a temporal availability, and at least one classification. Temporal availability represents days and times of day when a venue/event is accessible. Events are represented with at least a name, a venue they are associated with, a temporal availability, and at least one classification. Temporal availability for events represents dates, times, and durations of the event.

Venues, being derived from points of interest, can be geographically described with a single point, a group of points, a connected path of points, or a bounded region. In Figure 4, a geographic description is represented by type and by a collection of Coordinates, as part of Location 404. Two representation constructs, a single point, and a polygon, are used to describe all geographic entities. Point representations are used to describe venues with a single position. Venue geography can also be represented with a polygonal representation, where the coordinates

representing the venue geography are identified from the mobile station using a map 404 of the area near the venue. Other types of venues can have their geography represented with groups of points or paths of connected points using a map in a similar manner.

Venues and events can be inserted three ways: (1) for private access, (2) shared with the public directory, or (3) publicly. A private insertion is called a placemark. Sharing a placemark can lead to increased reputation value. Sharing a placemark also provides value to a user because changes made to shared public entries are updated in the user's placemark.

Insertion Process

The position of a location is assumed to be known within some error range, either because the user is using a mobile station that can provide its current position (through GPS, A-GPS, E-OTD or TDOA) or because the user has entered an address or location which can then be converted to longitude/latitude. The process of inserting new information can be initiated one of three ways:

1. The user initiates the process by requesting to insert a new venue or event in the user interface

2. The user initiates the process by requesting information about a venue or event and finding that doesn't exist in the directory.

3. The system initiates the process by noting that the user has been in the same location for a preset amount of time; that the user is in a location not represented in the directory; that information in the current location is sparse, or some other metric.

For cases (1) and (2), the user interface presents existing venues at the user's location with a "new venue" option, as depicted in Figure 6 at 602. Regardless of which approach is used to initiate the insertion, the user is prompted to provide the following information about a venue:

1. Name of venue 604.

2. Whether the venue will be a placemark in the user's personal directory, a publicly available item in the directory at large, or both 606

3. Classifications for the venue 608.

4. Address for the venue 610.

5. Temporal availability for the venue (i.e., open hours) 612.

6. Type of and data for the venue geography (point, path, or region) 614.

7. Other venue specific information.

If the user selects a venue, the venue information is displayed on the user's device along with a "new event" option to, and the user is prompted to provide the following information about the event:

1. Name of the event.

2. Whether the event will be a calendar entry in the user's personal directory, or a publicly available event in the directory at large.

3. Classification for the event.

4. Temporal availability for the event.

5. Other event specific information.

Because an event is associated with a venue, the name of the venue need not be input by the user since the venue must be selected at the time that the event insertion is selected.

The geography input defaults to point input. If the geography type selected is other than point, then the user is prompted to input the geography based on the type and also based on the input capabilities of the user's mobile station. In the case of non-point geographical representations, a map, photograph or satellite image of the area around the user's position may be displayed on the device user interface and the user selects the points, path, or region describing the geography using an input device. Aerial maps 404 may be retrieved for the purposes of determining path and region geographic representations, and may be stored locally on the mobile station while the new venue or event information is being entered, and until verification has been completed.

Preferrably, the insertion process can be interrupted and resumed later as long as steps 1-3. Until these steps have been completed, the new venue or event will not be made public. This process is depicted in Figure 7. The insertion process begins when the user views venues or

events in the vicinity of the mobile station, and elects to add a new venue, 702. Until the user has entered a name and an access value new values can be added or the process can be terminated, 706. If the access value is private", then the process is terminated and the entry is inserted as a placemark. If the access value is "public" or "shared", then the user must provide at least one category 704, and possibly more 708. At this point the user can elect to finish the process or input more venue data. If the user inputs a non-point geographic data type 710, then a map is retrieved and displayed on the mobile device for the purpose of assisting the user in identifying points associated with the venue with the mobile device. It should be noted that any form of pointing device would be applicable in this context, and that many are currently available. When the user has completed the input of venue data 704, a check is performed on their reputation 712. If the check fails, then the entry is inserted as a placemark. If the check succeeds then the entry is inserted into the public directory.

Although Figures 6-7 give a general description of a mobile interaction logic used to perform input operations, various mobile stations and mobile station interfaces exist, such as WAP/WML and J2ME, that could host this functionality. It will be appreciated that many other mobile stations and mobile station interfaces are understood by those skilled in the art and that the scope of the present invention should not be so limited to the exemplary descriptions as given above.

The process of entry insertion that extends beyond the user interface is depicted in Figure 8. The entry point to the insertion, depicted as 602 and 702, is shown as 802. The entire interaction depicted in 604-614, and 704, is depicted as 804. The form data entered by the user at the mobile station is collected and submitted at Connector 308 where it is converted into objects by Runtime Controller 314 and Insertion Program 320. When the user submits the entry for public insertion, the reputation program 316 performs a retrieval of history data 406 from the database 326, at 808, determines the reputation value, and returns a success value to the Runtime Controller. If the reputation value is acceptable, the entry is inserted into the database, at 810, using Classification Program 322.

C. VENUE CLASSIFICATION IN THE COMMUNITY-CONTRIBUTED LOCATION DIRECTORY

Venue classification is the mechanism that enables the input, storage, maintenance, and retrieval of venue and event information in the directory. There are three cases of interest in the classification mechanism:

1. The venue (or event) category input by the user doesn't exist in the universe of venues (or events).

2. The venue (or event) category exists in the universe of venues (or events), but no venue with the current position exists in the directory.

3. One or more venues (or events) with the same category and current position exist in the directory.

These cases are depicted in Figure 9. During the insertion process the user provides the information necessary to submit a venue or event into a form on the mobile station. This information is transmitted to Connector 308 and is converted into objects by Insertion Program 320. When the submission is approved for insertion by the Reputation program 318, Insertion Program 320 works in concert with Classification Program 322. In case (1), 902, the category provided by the user doesn't exist and so the new venue or event is associated with the "other" category 904. In case (2), 906, a new venue 401 is created, from the objects, and inserted into the directory, along with the location 404 and any address or availability 407 information provided by the user. In case (3), 908, if the name can also be matched, then a verification procedure is invoked to compare fields. In all three cases, reputation values 406 are updated, 912.

D. VENUE VERIFICATION IN THE COMMUNITY-CONTRIBUTED LOCATION DIRECTORY

Venue verification is the mechanism that enables the Runtime Controller 314 and Verification Program 324 to identify, when a venue (or event) is submitted, whether it already exists in the directory, and to interact with the user to resolve possible conflicts between the submissions. Verification is required to accommodate a situatioin where a user begins a shared

or public insertion but stops before providing the minimum amount of information for the insertion to take place, in which case a placemark is inserted into the directory. If the user returns and completes the submission data, and submits the entry as depicted in Figure 10, 1002, there is a possibility that someone else has already inserted the same venue/event into the directory. In such a case, the Verfication Program 324, shown as a sequence diagram in Figure 11, interacts with the user (at 1004) to determine which of the conflicting fields is correct, 1102, and gives the user submission credit for the entry. Some conflicting fields can be merged without user input, 1104. For example, if the user's submission has a geographic description with more information, such as a region as opposed to a point, then the Verification Program selects the region description for the final entry.

E. VENUE MODIFICATION IN THE COMMUNITY-CONTRIBUTED LOCATION DIRECTORY

Venue modification is the mechanism that enables an incomplete new venue/event to be fleshed out by the original submitter or, once a venue/event has been made public, to add additional information to the entry that was omitted by previous submitters. The modification mechanism (324 in Figure 3) accommodates two situations:

1. A user hasn't completed the minimal submission information for a new public venue/event.

2. The minimum submission information has been entered for a venue/event but not all of the data fields have been entered.

Venue modification can be performed at any time prior to submission by the original submitter, since the new entry is considered private until all of the information needed to perform an insertion into the directory has been provided. The entry shows up as a personal placemark in the user's directory and can be edited with the same interaction on the mobile station as for public modifications and corrections. Venue/event modification can be performed at any time after a venue/event has been made public by retrieving the venue/event information and selecting the "correct" element on the interface. Figure 12 depicts an interaction sequence that may

implement venue/event modification. The user first interacts with the Modification Program 324 at the venue display menu 1202. The interaction proceeds by the user selecting fields with incomplete entries, such as days/hours shown at 1204 in Figure 12, and entering them. Figure 13 illustrates the logic associated with the interaction sequence. The user elects to correct an entry at 1302. The user selects fields to modify (1304 – 1316) in any order. When done, the user submits the changes to the Modification Program 324, which checks to see if any of the modified fields were non-empty in the venue or event, 1318. If all the modified fields are empty, the entire operation is considered a modification, the user submission reputation is checked, and the update is performed. The internal process associated with venue/event modification is depicted as a sequence diagram in Figure 14. The user fills in information about the modified venue or event at 1402. The modifications are transmitted to the Runtime Controller 308 and the Modification Program 324, where they are converted into objects. The reputation is checked at 1404 and, if acceptable, the update is performed at 1406.

F. VENUE CORRECTION IN THE COMMUNITY-CONTRIBUTED LOCATION DIRECTORY

Venue correction is the mechanism that enables modification of venue/event information when the original submitter misspelled an entry, such as a name, or has submitted incorrect or outdated availability information, or has submitted an incorrect classification. In such cases, modification of an entry is required to secure the quality of the directory. Two issues are important when addressing venue modification:

1. Is the original submitter correct or is the person doing the modification correct?
2. Is the information incorrect or is it simply out of date?

Entry data can be retired in such a way that, after a predefined period of time since its submission, corrections made are assumed to be modifications and do not affect correction counts in user reputations. The amount of time that elapses before an entry is retired can be associated with an individual entry, a classification group, or the entire directory. The correction mechanism (324 in Figure 3) works in concert with the reputation mechanism (318 in Figure 3)

to both correct the entries made by a user and to weight future directory submissions made by the user. There are three sub cases of interest in the correction mechanism: the venue/event category and position exist in the directory, but the (1) address, or (2) name, or (3) availability is incorrect or out of date. Address can be out of date, but then the position would also be incorrect and the entry would have to be deleted and reentered. In all other cases the user can interact with the mobile station to make the correction. A possible interaction sequence supporting venue/event correction is depicted in Figure 15, and the accompanying logic is presented in Figure 13. The entry point for venue/event correction is the same as for venue/event modification, as is the logic. The difference is that correction takes place when a venue/event field already has a value, and the reputation checked is the correction reputation rather than the submission reputation. The Correction Program 324 uses the same sequence diagram as the Modification Program.

G. VENUE DELETION IN THE COMMUNITY-CONTRIBUTED LOCATION DIRECTORY

As information becomes outdated, businesses move, land changes hands, etc., the location directory entries become obsolete and must be removed. There is no automatic mechanism to keep the directory up to date, but a user can notice, when looking at local venues, that a venue that is displayed on the mobile terminal is no longer physically there, or that an event has already occurred. At that point, the user can select the item and delete it from the directory, assuming that the user's reputation supports entry modification. When an item is deleted, it isn't removed from the directory immediately, but it takes a different appearance in the user interface. If a user selects a deleted entry in the interface, the user is asked if the venue is actually still there. If soothe user answer affirmatively, the deleted item is marked for re-inclusion in the directory. After a waiting period in which the entry doesn't get marked for deletion or reinclusion, the entry is marked unavailable in the directory and no longer shows up on the interface. In the case of personal placemarks, a placemark can be deleted by the user at any time, since the placemark is a link to a public entry.

H. USER REPUTATION IN THE COMMUNITY-CONTRIBUTED LOCATION DIRECTORY

User reputation is the mechanism that enables a measure of confidence to be associated with the insertion/classification of new venue/event information into, or correction of existing entries in the community-contributed location directory. A reputation mechanism is desirable due to the contributory nature of the directory, and the fact that the contributors are not be screened beforehand for either their knowledge or their motivations for contributing. As a result, users who contribute erroneous or incorrect information, or maliciously contribute information, will have their contributions censored or even rejected while encouraging users to continue to contribute and so build the scope, depth, and quality of the directory.

There are two types of reputation in this system: submission reputation and correction reputation. They are distinguished from each other because submissions are associated only with a single user, while corrections are associated with multiple users. They differ most significantly by the threshold that must be exceeded to perform submission or correction operations on the directory.

In a preferred embodiment, submission reputation is defined as a ratio of submission corrections to submissions and references made in a particular period of time, as shown in Figures 17-20. Figure 17 illustrates a submission and reference history for a user over a 12-month period, where a submission refers to an entry submission (406 in Figure 4), and a reference (406 in Figure 4) refers to a person linking to the person's placemarks or some other mechanism of determining value of a submission. Included in Figure 17 is a "blackout period" in the fourth month when the user's submission reputation exceeded the submission threshold. Figure 18 illustrates the number of errors (i.e., the number of submissions that were later corrected) over the same 12-month period. It shows that when submissions weren't allowed, errors didn't occur, but that there were also no errors reported in months 1, 7, 9, 10, and 12. In order to avoid punishing heavy contributors who make early mistakes, a decay mechanism is

applied to the error rate. A decay rate applied to the error rate is shown in Figure 19. Using these graphs, Algorithm 1 is used to generate the reputation value at any given time:

$$\text{Reputation}_{\text{month}} = \Sigma\text{error}_i \,/\, \Sigma\left(\text{submissions}_i + \text{references}_i\right) \qquad \text{Algorithm 1}$$

For example, to calculate the value of reputation for month 3 the algorithm expands as follows (assuming a 10% decay rate per month from Figure 19):

$$\text{Reputation}_3 = (er_1 + er_2 + er_3) \,/\, ((sub_1 + ref_1) + (sub_2 + ref_2) + (sub_3 + ref_3))$$
$$= (0 + 3 + 5) \,/\, ((1 + 0) + (4 + 1) + (5 + 2))$$
$$= 8 \,/\, 13$$
$$= 61.5\%$$

Figure 20 illustrates the error rate as a 10-month sliding window, constructed by taking the cumulative number of errors divided by the cumulative number of submissions, on a monthly basis, and shifting the accumulation at the 10-month mark. Overlaid on top of this histogram is a threshold line, above which the user is unable to submit. On this diagram it can be seen that the user exceeds the submission threshold at month 3, when the reputation value reaches 61.5% and exceeds the threshold of 60%. Referring back to the submission histogram, the user is not allowed to make submission for month 4, during which the month 3 error rate decays by 10% according to the graph of Figure 19. This drops the user reputation below the threshold and the user can again make submissions in month 5. Figures 17-20 assume that corrections are made in the same month that submissions are made, which would not generally be the case. Algorithm 1 maintains the highest quality reputation values when corrections are applied as soon after the submission is made as possible.

Figure 21 depicts the process whereby reputation is used and modified during the submission and of entries into the community-contributed location directory. The logic for calculating correction reputation is the same except that there is no consideration of privacy and the correction threshold is different than the submission threshold. Instead of insertions, when performing corrections the result is an update on the venue or event.

Reputation information is associated with a user and so it is persisted information related to the location directory. There is history maintained for submissions, references, and corrections (406 in Figure 4) for each submitter, and these are retrieved (2106, 2108, and 2110) and used to calculate reputation (2112) and determine whether a user can submit entries according to algorithm 1.

III. COMMUNITY-CONTRIBUTED LOCATION DIRECTORY IMPLEMENTATION

The operating environment that hosts the current embodiment of the present invention uses Java® J2EE, which is a programming language commercially available from Sun Microsystems, Inc., Palo Alto, California, to implement location programs (item 608 in Figure 6), Apache, which is a webserver available from the Apache Software Foundation that is used to implement the web-based user interface, WAP, which is a wireless application language used to implement WAP-based user interfaces on wireless devices, Java® J2ME, which is a programming language commercially available from Sun Microsystems, Inc., Palo Alto, California, to implement Java-enabled user interfaces on wireless devices, Weblogic®, which is application server software commercially available from BEA Systems, Inc., San Jose, California, to implement the runtime controller (item 606 in Figure 6), and Oracle9i®, which is database server software commercially available from Oracle Corporation, Redwood Shores, California to implement the backend (item 604 in Figure 6). It will appreciated by those skilled in the art that implementations employing the current invention need not be made using Java, J2EE, Apache, WAP, J2ME, or Oracle, and that other mechanisms could be employed.

CLAIMS

What is claimed:

1. A method for inserting data into a community-contributed location directory, comprising:

identifying a position associated with the information;

identifying whether the information represents a venue or an event;

identifying a name for the information;

identifying whether the information is to be used privately or publicly;

identifying a geographic description of the information;

identifying a period of availability associated with the information;

identifying a classification associated with the information;

establishing a data management structure having a venue storage area for each venue or event of the set of identified information; and

storing one or more entries in each venue storage area.

1. A method for maintaining submission/correction histories for entry submissions, comprising:

Identifying a user submission associated with an entry to the directory;

Identifying a date associated with each attribute of the entry;

Identifying a correction associated with each attribute in the entry;

Identifying a date with each attribute correction of the entry;

establishing a data management structure having a user storage area for each submitter or corrector to entries in the directory; and

storing one or more entries in each user storage area.

3. A method for determining reputation of directory entry contributors; comprising:

identifying the timeframe of submission;

identifying the submission history of the submitter;

identifying the correction history against the submitter;

identifying the submission threshold for the timeframe of submission; and

evaluating the cumulative ratio of corrections to submissions in the timeframe against the submission threshold.

4. The method of claim 3 where the entry is being corrected rather than submitted and additionally; comprising:

identifying the submission history of the corrector;

identifying the correction history of the submitter;

identifying the correction threshold for the timeframe of submission; and

evaluating the cumulative ratio of corrections to submissions for the submitter, the cumulative ratio of corrections to submissions for the corrector against the correction threshold.

5. The method of claim 1 wherein the data management structure reflects the hierarchical relationship among the venues.

6. A machine having a memory which contains data representing one or more location items of a universe of venues generated by the method of any of claims 1 and 2.

A COMMUNITY-CONTRIBUTED LOCATION DIRECTORY

ABSTRACT

Method and apparatus are disclosed related to the development and implementation of a community-contributed location directory for the purpose of acquiring, maintaining, and providing access to location-specific and temporally-viable information. A directory employing the present invention bases its organization on a predefined functional hierarchy of venue descriptions, called a universe of venues. The complete universe of venues is decomposed

hierarchically. A data management structure is established to provide a storage area for each component of the hierarchy resulting from the decomposition. Each such component is called a point of interest. The data management structure may reflect the hierarchical decomposition. Keyword terminals in the hierarchy organize information about venue and event information in the directory. Venue/event information is entered at a mobile terminal or a web-enabled terminal and is used to insert entries into the directory or retrieve information from the directory. Information can be inserted into the directory using point, path and region geographic representations. Information being inserted is validated based on venue/event name, address, position, functional description, temporal availability, and reputation of the contributor.

Docket No. ?

UNITED STATES PROVISIONAL PATENT APPLICATION FOR

A COMMUNITY-CONTRIBUTED LOCATION DIRECTORY

Inventors:

Jack Hodges
Dan Greening

Prepared by:

?
?
San Francisco, CA ?
(415) ?