METHOD AND APPARATUS FOR FUSING CONTEXT DATA

Inventors: Ying Chen, Beijing (CN); Hui Lei, Scarsdale, NY (US); Jussi P. Myllymaki, San Jose, CA (US)

Correspondence Address:
Scully, Scott, Murphy & Presser, P.C.
400 Garden City Plaza, Suite 300
Garden City, NY 11530 (US)

Assignee: International Business Machines Corporation, Armonk, NY (US)

Filed: May 23, 2008

Related U.S. Application Data
Continuation of application No. 10/314,059, filed on Dec. 6, 2000, now Pat. No. 7,392,247.

Publication Classification
Int. Cl.
G06F 3/048 (2006.01)
G06N 5/02 (2006.01)

U.S. Cl. 706/11; 706/50

ABSTRACT
A system and method for fusing context data for use by context aware applications. The method includes the steps of receiving context data from a plurality of information sources; computing a quality measure for each input context value; organizing context values into one or more clusters, and assigning a single context value and a single quality measure to each cluster; and, selecting one or more clusters according to one or more criteria and aggregating the context values and quality measures of selected clusters to generate a single context value and quality single measure. The single context value and single quality measure are usable by a context aware application to avoid conflict and ambiguity among different information sources providing the context data.

System Architecture
Figure 1
Receive Context Values from Sources

Normalize Context and Belief Values

Insert Context and Belief Values into Database

Cluster Context Values in Database

Select Qualifying Clusters

Aggregate Cluster and Quality Values

System Flowchart

Figure 2
METHOD AND APPARATUS FOR FUSING CONTEXT DATA

RELATED APPLICATION

[0001] This application is a continuation of U.S. Ser. No. 10/314,059, filed Dec. 6, 2002, the entire contents of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

[0002] 1. Field of the Invention

[0003] The present invention relates generally to context-aware applications for computer systems, and more particularly to a novel context fusing system and methodology for supporting context-aware devices and applications.

[0004] 2. Description of the Prior Art

[0005] Pervasive computing applications are increasingly relying on the context of the user whom the application is designed to support. Contextual information is derived from a variety of sources and enables the application to provide the user with qualitatively better service. For instance, context awareness can be seen as key to such pervasive computing features as personalized access, intention inference, resource opportunism, and system proactiveness. The net result of improved context awareness is a reduced demand for human attention by the application.

[0006] Context awareness is made possible by the ubiquity of sensors of different kinds. For instance, several sources of location information are typically available—consider the rapidly expanding use of cellular phones (some of which are already enabled for location tracking in order to help emergency rescue teams), GPS receivers, wireless LAN computers, wireless PDAs such as the Blackberry device, electronic calendars, and active badges. Despite this abundance of contextual information, one key problem remains: resolution of conflicts and ambiguities among different information sources.

[0007] As a motivating example, the following scenario is considered: A user's car is equipped with a GPS tracking device which periodically beacons the car's location to a central server. The user also carries a wireless PDA which beacons its location to the same server. Now, if the car was last reported seen in "San Francisco" 5 minutes ago and the PDA was last reported seen in "San Jose" 10 minutes ago, which location report should the system pick as the most current and accurate? In other words, which context is the user currently in? Surely the user cannot have moved from San Jose to San Francisco in 5 minutes.

[0008] In Paul Castro and Richard Muntz, "Managing Context Data for Smart Spaces," IEEE Personal Communications, Vol. 7, No. 5, (October 2000), pp. 44-46, a context fusion service that uses evidential reasoning techniques, is described. However, this service has a number of disadvantages. First, it works with a pre-determined set of context sources only and does not support dynamically added sources. Second, it requires a separate training phase in order to obtain the probability distribution of input context values. Third, both input and output context values must be quantized into discrete values.

[0009] Asim Smalagic and David Kogan, "Location Sensing and Privacy in a Context Aware Computing Environment," IEEE Wireless Communications, October 2002, teaches a method for fusing location data in a wireless LAN environment based on a combination of triangulation and mapping. The method also requires a pre-determination of context (location) sources and a separate training phase. In addition, it handles homogeneous location data only.

[0010] Another method for fusing location data is described in Jussi Myllymaki and Stefan Edlund, "Location Aggregation from Multiple Sources," Proceedings of International Conference on Mobile Data Management, Singapore, 2002. The method ranks location reports from different sources and selects the most probable one based on factors such as time, user associativity and device mobility. Besides to being location-specific, this method does not consider the relationships between input values and is not able to improve the precision of input values.

[0011] Therefore, a need exists for a general fusion technique that can accurately fuse context data from a dynamic set of sources.

[0012] It would be highly desirable to provide a context fusion system and software architecture for supporting pervasive computing devices.

SUMMARY OF THE INVENTION

[0013] The present invention is directed to a very general framework for context fusion that addresses the problem of resolving conflicting contextual information and removing ambiguities to provide applications and other systems with dependable, aggregated, and vastly improved context information. It reduces the likelihood of errors made by a context-aware application and thereby helps improve user efficiency and satisfaction with the system.

[0014] According to the principles of the invention, there is provided a system and method for fusing context data for use by context-aware applications. The method includes the steps of receiving context data from a plurality of information sources; computing a quality measure for each input context value; organizing context values into one or more clusters, and assigning a single context value and a single quality measure to each cluster; and, selecting one or more clusters according to one or more criteria and aggregating the context values and quality measures of selected clusters to generate a single context value and quality single measure. The single context value and single quality measure are usable by a context aware application to avoid conflict and ambiguity among different information sources providing the context data.

[0015] Advantageously, the system and method for fusing context data according to the invention accommodates the dynamic availability of context sources and is widely applicable for use in a variety of context aware applications. Furthermore, no prior training is required to practice the invention.

BRIEF DESCRIPTION OF THE FIGURES

[0016] The objects, features and advantages of the present invention will become apparent to one skilled in the art, in view of the following detailed description taken in combination with the attached drawings, in which:

[0017] FIG. 1 illustrates the Context Fusion System 10 according to the invention; and,

[0018] FIG. 2 illustrates a flowchart depicting the methodology employed in the system is of FIG. 1.
The Context Fusion System 10 that receives data from one or more context sources 15a, 15b, ..., 15n filters and aggregates the data, and forwards it to an application or user is now described with respect to FIG. 1. In one embodiment, the Context Fusion System 10 follows a “push model” where context sources 15a, 15b, ..., 15n submit context data to the system without an explicit request by the system. In contrast, an alternate embodiment would be a system implementing a “pull model” where context sources 15a, 15b, ..., 15n are explicitly queried, on demand from the application or user, for context data. An embodiment consisting of a mixture of push and pull sources is also possible and can be realized by anyone skilled in the art.

It is understood that context sources 15a, 15b, ..., 15n may differ greatly in terms of their capacity to process and refine context data before submitting it to the Context Fusion System 10. No assumptions are made as to their degree of sophistication and it is only required that the source indicate its quality assessment (belief) on the data it reports. For example, a location context source 16 may simply retrieve the current location of a mobile phone user from a wireless operator’s network and forward the data unaltered to the Context Fusion System 10. Another location context source may record historical location data and apply sophisticated trajectory analysis, pattern discovery, and location prediction algorithms on the raw data and submit only refined data to the Context Fusion System 10.

The input to the Context Fusion System (CFS) 10 comprises a vector of context values $V_1, V_2, ..., V_n$ on a single subject form various sources. Each value represents a context source’s belief on the state of the subject. Let each context value be represented by a tuple $V_i = (C_i, B_i)$ where $C_i$ contains the state of the subject and $B_i$ represents the source’s assessment (belief) of the quality of $C_i$. For instance, in a simple case, $B_i$ may represent the probability between 0 and 1 that $C_i$ is true. In the general case, $C_i$ can be a scalar value, a vector value, a geometric shape, or any other type of data value. Thus, the context fusion system 10 and method has a wide range of applicability as it is able to process context data comprising point values, shape values, discrete values, continuous values, and finite and infinite data ranges.

In the preferred embodiment, the output CFS is a single context value $V_{out} = (C, B)$ where $C$ represents the best knowledge on the state of the subject and $B$ is a quality measure on $C$.

As further shown in FIG. 1, the Context Fusion System (CFS) 10 comprises seven components: a Context Receiver 20, a context Database 30, a Clusterer 40, a Selector 50, an Aggregator 60, a Policy Manager 70, and a Policy Database 80. The CFS 10 interfaces with the external world via one or more Context Sources 15a, 15b, ..., 15n and Context-Aware Applications 90, also shown in FIG. 1.

The Context Receiver component 20 particularly functions to receive context values of the form $V_i = (C_i, B_i)$ from one or more Context Sources 15a, 15b, ..., 15n. The Receiver 20 performs any necessary data format conversion on $C_i$, for instance scaling, normalization, and duplicate removal, as deemed needed for each particular Context Source.

Based on $B_i$ and other external factors, the Receiver component 20 computes a value $Q_i$ that represents a quality measure on $C_i$. Among other things, these external factors may include the following: timestamps associated with $C_i$ (creation, last modification, and expiration time) and the perceived confidence of the Receiver in the Context Source, either specified or derived from historical data.

It should be understood that Context Sources 15a, ..., 15n operate independently of the Receiver and may submit new context values to the Receiver at any time. The normalized context value tuples $V_i = (C_i, Q_i)$ are stored in the Context Database 30. Preferably, the Context Database 30 stores context values as triplets $R = (V_i, C_i, Q_i)$ where $R$ is a global identifier for the user and $R$ is a record in the database. Any database may be used for the Context Database 30, e.g., relational database, flat file, or main memory data structure.

The Clusterer component 40 functions to perform a first pass through the context values and organizing it into one or more clusters. For each cluster, a combined context value $CV$ and quality measure $Q$ are generated. It should be understood that many clustering methods are possible. For example, one clustering method includes making each distinct input value a separate cluster. Another manner for clustering is to make a single cluster out of all input values. According to the invention, several methods may be used to combine context values $CV_i$ into a single $CV$, including, but not limited to: the context value with best quality; the average context value (for continuous range); the average context value weighted by quality (for continuous range); the intersection of context values (for shapes); and, the center of gravity (for shapes and points). Several methods additionally exist for combining quality values. A variety of methods exist for selecting a $Q$, including, but not limited to: the maximum $Q$ value (optimistic quality assessment); the minimum $Q$ value (pessimistic quality assessment); the $Q$ of the $CV_i$ chosen to represent the cluster; and, the average $Q$.

The generated clusters and their associated values $CV$ and quality measures $Q$ are forwarded to the Selector component 50 which functions to determine a subset of the clusters for further aggregation to be implemented. The selection may be performed according to a variety of criteria, including, but not limited to: cluster size (number of context values in the cluster); cluster quality ($Q$ value); and, cluster tightness (how closely the context values fit into the cluster, i.e., how much wasted space the cluster has). A simple measure of tightness is the size of the minimum bounding box of all context values in the cluster. Another simple measure is the sum of the areas of all context values divided by the minimum bounding box of all context values in the cluster.

The selected clusters are forwarded to the Aggregator component 60 which functions to combine the selected clusters into a single context value $CV$ and quality measure $Q$. As described in the context of the Clusterer component 40, several methods exist for combining context values and quality measures.

As shown in FIG. 1, the Policy Manager component 70 guides each of the Clusterer 40, Selector 50, and Aggregator 60 components in their decision-making. A policy may, for instance, dictate what specific clustering algorithm the CFS should use. The policy chosen may depend on the user, application, and type of context (e.g., location vs. calendar context). The policies chosen for the Clusterer 40, Selector 50, and Aggregator 60 components are stored in the Policy Database 80. Particularly, data is stored in the database as tuples $R = (U, A, CT, CP, SP, AP)$ where $U$ corresponds to the user, $A$ is the application, $CT$ is the context type, and $CP$, $SP$, and $AP$ correspond to the Clusterer policy, Selector policy, and Aggregator policy, respectively. The CFS 10 may provide a default set of policies, which may be supplemented or replaced by user-supplied policies. Default or user-supplied policies may be selected by the user or automatically selected by the system 10.
As shown in FIG. 1, the output of the Aggregator component 60 which comprises context data in the form of a single context value CV and a quality measure Q, for example, is input to an Application 90. The Application 90 particularly, is any context aware application that receives context data from CFS 10. The Application may make an explicit request to CFS to get fresh context data (i.e., a pull model) or may receive context data updates from CFS as new data becomes available (i.e., a push model).

The CFS system 10 and system components thereof may be implemented in a server device, e.g., a web-server, or may be distributed among a plurality of interconnected server, computer and database devices. An application program interface (API) 11a, 11b may be provided as part of the system to facilitate the communication of the context source data between the client (context source) and the Context Receiver 20, and/or between the system 10 and the user or application 90, depending upon whether a push-model or pull-model is implemented.

The process 100 implemented by the context fusion system of the invention is now depicted in FIG. 2. As shown in FIG. 2, a continuous process is depicted that includes a first step 105 of receiving the context values from sources, as requested by a user or user application via a pull-model or, as “pushed” to the system from a variety of widely available context data sources. At step 110, an optional step of normalizing the content and the quality assessment (belief) values indicated by the source as to the data it reports is provided. Then, at step 115, the context values and quality assessment (belief) values are inserted into a database where the context values are organized as clusters at step 125 in the manner as described. Then, step 130 includes the step of selecting the qualifying clusters for aggregation which aggregating step is performed at step 135. It is understood that after aggregating the resulting single context value and single quality measure value is made available to the user or user application.

While the invention has been particularly shown and described with illustrative and preferred embodiments thereof, it will be understood by those skilled in the art that the foregoing and other changes in form and details may be made therein without departing from the spirit and scope of the invention that should be limited only by the scope of the appended claims.

What is claimed is:

1. A system for fusing context data obtained from a plurality of sources, said system comprising:
   (a) one or more processors comprising:
      (a) means for receiving said context data including one or more context values from one or more information sources;
      (b) means for computing a quality measure for each context value;
      (c) means for organizing context values into one or more clusters, each cluster having a single context value and a single quality measure associated therewith;
      (d) means for selecting one or more clusters from said clusters having one or more criteria and the assigned single context values and quality measures of said selected one or more clusters to generate a single representative context value and single representative quality measure wherein said single representative context value and single representative quality measure are usable by a context aware application to avoid conflict and ambiguity among different information sources providing said context data,
   (b) further comprising means for guiding clustering, selecting and aggregating operations used in generating said single representative context value, said guiding performed according to a respective clustering policy, selection policy, and aggregation policy, wherein a selection policy implements a step of selecting according to sum of areas of all context values divided by minimum bounding box of all context values in the cluster.
   2. The system of claim 1, implementing a push model whereby updated context data is automatically received from said sources.
   3. The system of claim 1, wherein said context aware application comprises means for requesting updated context data from said sources.
   4. The system of claim 1, further comprising said clustering, selecting and aggregating policies being stored in a policy database.
   5. The system of claim 4, further including means for maintaining default clustering, selection and aggregation policies, said clustering, selecting and aggregating operations being automatically selected.
   6. The system of claim 4, further including means for enabling a user to specify said clustering, selection and aggregation policies used for guiding said clustering, selecting and aggregating operations.
   7. A computer program device readable by a machine, tangibly embodying a program of instructions executable by a machine to perform method steps for fusing context data, said method steps comprising:
      (a) receiving context data including one or more context values from a plurality of information sources;
      (b) computing a quality measure for each context value;
      (c) organizing context values into one or more clusters, at least one of the clusters having a plurality of context values, and assigning a single context value and a single quality measure to each cluster; and
      (d) selecting one or more clusters from said clusters according to one or more criteria and aggregating the assigned single context values and quality measures of said selected one or more clusters to generate a single representative context value and single representative quality measure,
      wherein said single representative context value and single representative quality measure are usable by a context aware application to avoid conflict and ambiguity among different information sources providing said context data,
      further comprising the steps of guiding clustering, selecting and aggregating operations used in generating said single representative context value, said guiding performed according to a respective clustering policy, selection policy, and aggregation policy, wherein a selection policy implements a step of selecting according to sum of areas of all context values divided by minimum bounding box of all context values in the cluster.
   8. The system of claim 1, wherein said context values includes a belief value representing a context source’s belief of the quality of a state of a single subject represented in a context value.

* * * * *