CONTEXT MANAGEMENT FOR HETEROGENEOUS ADMINISTRATIVE DOMAINS

Filipe Meneses[1]

Abstract
When accessing user context, context-aware applications often interact directly with sensors or have to deal with specific space representations. This work addresses context representation and management for mobile users. It proposes a generic solution based on a Context Container where the user context can be represented by an unlimited number of dimensions. The proposed solution is based on a Context Manager that integrates the raw data acquired by sensors and enriches user context with new calculated and estimated dimensions.

1. Introduction

Since Weiser [10] published his visionary work in 1991, many location-based and context-aware systems have been developed. As a new research area, many of the results were achieved through simple implementations where location and other contextual data were used directly from sensors to provide adaptation or selection of information accordingly to the user context (in many cases there was a direct link between the location information and the service discover/selection). These were the early days of context-aware computing, and the various experiences made since then have identified many of the research challenges that this area is facing today.

Meanwhile, computing devices, wireless networks and sensors have also evolved, offering increasing and more attractive functionality. As the technology matured and became more accessible, an increasing number of places started to be equipped with wireless communications infrastructures, sensors and other devices, towards a pervasive environment.

However, these infrastructures are very heterogeneous and are deployed and managed by different administrative entities. The exploitation of these new and rich environments for ubiquitous computing requires the development of open, scalable and flexible solutions for multiple problems, including context management. Among these, the necessity to evolve the notion of context is being seen as one of the most crucial. Context-aware applications must be developed on top of generic platforms, using network services with standardized interfaces, and accessing contextual information ready-to-be-used. It is no longer desirable that each application includes its own context engine, which gathers information from multiple sensors, and processes it to select services or information and to adapt its behaviour accordingly. The achievement of a common understanding of context is of great importance and must trigger new developments in this area.

This work addresses the issues of contextual data acquisition, representation and management in multiple and heterogeneous administrative domains. Our goal is to design an open context

[1] Information Systems Department, University of Minho, 4800-058 Guimarães, Portugal (meneses@dsi.uminho.pt)

framework that exploits the emerging pervasive environments to provide advanced contextual information to context-aware applications. This includes the ability to collect data from multiple context sources maintained by different administrative entities, enhance that data with some meaning in the local environment, and provide an interface where applications can access the contextual data.

The proposed approach starts with the definition of a Context Container, an independent entity that represents the context of a mobile user by aggregating contextual information in a multidimensional space. It then proceeds to the notion of calculated and estimated context dimensions as new contextual data derived from raw data acquired by context sensors. Finally, it evolves to the integration of heterogeneous environments where a number of contextual sources, managed by third-party entities, are exploited to enhance the user context. Although defining context as a multidimensional space, the implementation work is being based mainly on the location dimension due to the existence of more sensors, applications and services able to use this dimension.

2. General concepts

The terms "position", "location" and "context" are often used to describe the same, or similar, data. In particular, position and location are often being used to represent "where a user is". Here we define position and location with two different meanings.

A large number of technologies and mechanisms can be used to retrieve the *position* of a mobile user or device. The existence of a space model enables the transformation of position information into *location* information. The position information (raw data) is seen as location information when used on top of a *Space Model*. For example, a pair of geographic coordinates (position) may represent a certain street (location) if represented on top of a streets map (space model).

Space models are a representation of a certain geographic area and are maintained by some administrative entity. Examples of space models are: a) the geographic division of space as a set of polygons defined by a list of geographic points, such as country borders; b) the geographic position of a cell's centre a cellular network, referred by the cellID, and the corresponding cell radius.

We define *context* as all the information that characterizes the user in a specific moment. Although location has been the more exploited dimension of a user context, many other dimensions can be used to support the adaptation of context-aware applications. A context may include dimensions such as user position, expressed in many different referentials, user location, user activity, a list of nearby objects (or people), available resources, etc. The value of some basic dimensions of context may be obtained directly from physical sensors (position, orientation, ambient light level, room temperature, etc.). Other dimensions may be calculated from raw data (speed, acceleration, location, list of nearby objects, etc.). Some other dimensions may even be estimated from the information provided by other dimensions (activity may be estimated from position history and time, more precise location from position and street maps, etc.).

The work described in this paper considers a context-aware system where the *infrastructure* can be very different from one user to another. A single user may use one or more computing devices, mobile or not, to access context-aware applications and the devices can be used alternatively or simultaneously. The contextual information can be collected both from sensors on the mobile device and from sensors on the infrastructure. In the last case, access to the contextual data they provide is made through a network service (e.g. a location service like a GMLC[2] in a mobile cellular network).

3. Context Representation

A major issue in context-aware computing is the design of data formats used to exchange contextual information. Along with the set of protocols required to transfer contextual data, data formats are one of the research challenges for ubiquitous computing [2]. Most of the reported approaches to context-aware systems use proprietary data formats to describe context, in particular in what concerns position and location. This has led to prototypes that demonstrate context-aware systems that work only within the lab or within very restricted and controlled environments.

On the other hand, the context of a user is a point in a multidimensional space with an unbounded number of dimensions. Even for location only, we cannot predict all the sensors that will be available in the future or predict the needs of applications in terms of contextual information. We should then consider a context representation that is, simultaneously, usable by applications, independent from the context sources, and also flexible enough to accommodate future needs.

Taking these issues into account, we envision the context of a user (or of other entity) has an *unbounded and dynamic list of attributes*, represented by standardized and non-standardized data structures. It is unbounded because there is no limit on the number and nature of the used attributes. It is dynamic because the list of attributes can vary with time and because the values of those attributes are also time variant. The context is a *cumulative storage of knowledge* about the user history, being able to remember past experiences and situations. As an example, the context must be aware that a user visiting a certain town has already been there some time ago.

The context is described by a fixed number of *mandatory attributes* and a variable number of *optional attributes*. The mandatory attributes have well defined data formats (e.g. the context must always include the geographic position of the user, described as a pair of geographic coordinates – latitude, longitude pair – in the WGS84 datum). Whenever the value of a mandatory dimension is unavailable, it must be set to "unknown", as also suggested in [2]. The set of mandatory attributes are used to represent the few most relevant and widely used context dimensions. The variable number of optional attributes is described by an XML[3] stream, with an arbitrary data format. Although optional, the representation of these attributes may also be used through normalized data formats. Here, the term optional means that they are not required to be available in a context representation. However, one expects that most of the attributes in a context representation will be optional.

The context attributes are classified into the following categories: primary, calculated and estimated attributes. *Primary attributes* represent raw data acquired directly from sensors (e.g. the cellID code in a cellular network), or data retrieved from space models (e.g. the cellID can be used as a key to retrieve the geographic coordinates of the centre of the cell and its radius from a space model server). *Calculated attributes* represent contextual data that can be calculated from raw data (e.g. the speed of a user can be calculated from successive position readings). These attributes can be calculated by a context manager on its own or with the help of network services. *Estimated attributes* represent contextual data that can be estimated from other attributes. For example, the

[2] GMLC – Gateway for a Mobile Location Centre

[3] We are using XML in our prototype implementation because it simplifies the exchange of contextual data between the system components, but other representations would be possible.

context manager can estimate the name of the street where a user is located by querying a maps server with the geographic coordinates of the user; the result would be: "the user is at the Liberty Avenue, with a probability of 76%".

4. Context Management

The core component of this architecture is the *Context Manager*. Its functionality includes the collection of contextual data, its processing, the provision of an interface for applications to access the contextual data, and security management. The Context Manager holds the *Context Container* described in the previous section.

A single Context Manager is allowed to manage the Context Container of multiple users, providing that each Context data space is maintained completely separated from every other data, and access control is based on authentication on a per user basis.

The *Context Handler*, a process running on user devices, collects contextual data from *Context Sources*, through *Context Sensors*, and updates the Context. Contextual data is obtained from multiple sources such as: i) devices attached to the client device, like a GPS receiver or a bar code reader; ii) "soft sensors" running on the client device and that are able to access data such as the cellID on a cellular network, the identification of the current cell on a WLAN, or beacons broadcast over a local area network. The Context Manager can also obtain position or location data from *Location Services*. These are services usually maintained by network operators or some other administrative entities that provide the position/location of a set of users (subscribers) or that track users within a certain geographic area. An increasing number of cellular network operators are already running this type of service.

Each new contextual data set arriving to the Context Manager, coming from a Context Handler, is stored in the Context Container as a primary attribute without any processing. Additionally, the Context Manager can a) calculate or estimate the value of other attributes based on the newly arrived data set; b) obtain, from *Space Models* servers, extra contextual data related to the newly arrived data set; c) notify one or more applications that new contextual data is available. The extra contextual data retrieved from Space Models services is also added to the Context Container by the Context Manager, and can also be used to calculate or estimate the value of other attributes.

Except for the mandatory attributes, all other contextual data sets, fed by a Context Handler, obtained from Location Services or retrieved from Space Models, can be represented using arbitrary data structures, as long as they are represented in XML format. If normalized data structures are used, the Context Manager can interpret each new arriving data structure and process it accordingly. If not, the Context Manager only stores the new data structure as a new optional attribute (or updates an already existing attribute that matches exactly the same data structure).

Context-aware Applications can access the contextual data maintained by the Context Manager through, and only through, the *Context Query interface*, using a predefined protocol and data formats. Access to the context attributes is provided by a set of generic queries that return one or more attributes in their original (stored) data structure format. Other solutions for the access protocols were also considered, such as those described in [3].

Applications accessing contextual data through the Context Query interface must understand the representation of mandatory attributes and their semantics. On the other hand, it is up to the applications to understand or not some or all optional attributes. As an example, consider a scenario where a tourist arrives at an airport terminal she has never been before. Her context-aware system may discover a local “airport guide application” and suggest its use to her. As she uses this application, her context is updated with contextual data acquired from local Context Sources and through a local Space Model server; her Context Manager enhances her context with attributes that only the airport guide application understands.

Access to contextual data is only permitted through proper authentication. This approach centralizes the authentication and access control on a single point, making the security and privacy problems more treatable. Similarly, updating Context status, through the Context Feeder interface, also requires authentication.

5. Related work

The need to separate the acquisition of sensor data from its use was already demonstrated by many authors in the past. Some of the reported approaches are based on Location Services that provide an abstraction layer that hides the specificities of position technologies from the applications. Harter and Hopper [1] proposed a distributed location system based on the Active Badge location technology. In their proposal location is the elementary element in a symbolic space model, such as a room or the area around a particular office desk.

Leonhardt [6,7] extended the concept of location service by considering that multiple sensors can be used to simultaneously track the position of objects, and that sensors and location-aware applications can reside on different nodes in an open distributed system. In this scenario, the position information from each sensor has to be integrated and to be made available to applications as a single piece of location information. For this, Leonhardt proposed an acquisition function based on a fusion algorithm that integrates location data from multiple sensors. The acquisition function also deals with overlaps and inconsistencies among the location information provided by the various sensors, and provides an abstraction layer that allows applications to be developed independently from the sensor technologies.

The idea of a universal location service was also addressed by Leonhardi [5] that proposed an architecture for a distributed location service based on Internet technology. In this architecture clients can ask for the location of a mobile object (*locationOfObject* query) or for the objects that are at a given location (*objectsAtLocation* query) by issuing a query to a location server. As mobile objects can move from the area managed by one location server to another, there is an object register that maintains a list of the location servers that have location information related to a given object. Similarly, there is a location register responsible for listing the location servers able to respond to an *objectsAtLocation* query. In this system, location information is based on a topological model using the WGS84 datum.

A location service for cellular radio networks is also under specification by the Location Inter-operability Forum - LIF [7]. LIF is an initiative promoted by several companies from the telecommunications industry. The Mobile Location Protocol (MLP) provides applications with the possibility to query a server for the location of mobile stations independent of the underlying network technology. Several types of location services are provided, going from the Standard Location Immediate Service that returns the actual position of a station, to the Triggered Location Reporting Service that notifies a Location Services Client at a specific time interval or on the occurrence of a specific event.

These and other location systems provide a framework that simplifies the development of location-aware applications, as they provide a uniform, and maybe universal, representation of location that is independent of the positioning technologies. This approach also makes it easier to exchange location information between applications (location referencing) and provides applications with a common query language to access location information.

A step further to address the issue of context representation was given by the Context Toolkit approach [9]. Here, a network of heterogeneous sensors that expose their functionality through a normalized network interface – the widgets, replaced the idea of a homogeneous location service located on the infrastructure. This approach also provides support for many different types of context sources. In the same direction, Hong et al [2] proposed an approach where most the acquisition, processing and storage of contextual data is maintained in the infrastructure along with a multitude of services that cooperate to support the applications. The applications can also run on the infrastructure, reducing dramatically the capabilities required from the mobile devices. In this proposal, Hong et al. also describe the major research challenges that have to be addressed before a context-aware infrastructure approach can be deployed. These include the definition of a standardized communication protocol and a definition of the data formats required for applications to access contextual data.

Similarly, Judd et al. [3] proposed a solution based on a limited number of service classes and a Contextual Service Interface to access contextual information using a set of pre-defined query types. This approach also identifies the necessity to define a data format as a key requirement for this solution to be implemented. On the other hand, the discovery of the network services by applications is not addressed, as well as security and privacy.

6. Potential for innovation

In the approach described in this paper, we integrate some of the solutions proposed by the authors referred in the last section. In addition, a generic and flexible approach for the representation and management of contextual information is proposed. Our solution integrates the data collected from the various Context Sources deployed in different administrative domains and, instead of just collecting and storing data, it adds new value to that data through the use of Space Models.

The work will also contribute to the creation of a new Context definition, where the Context will be no longer an aggregation of sensed data or a simple expression of the user location but an open entity in a multidimensional space with an unbounded number of dimensions.

Moreover, using all the data available in the user context, the system is able to estimate new data. With the raw data received from the sensors, with the attributes received from the space models and with the user context history, new contextual attributes can be estimated, enriching the user context. The system also accepts data from multiple administrative domains. The contextual data is accessible to the applications as an XML stream which constitutes an open system, easy to be processed by any application. Other representations may be added, namely when a standard world wide accepted exists to represent the user location context.

7. Methodology and future work

The work will be based on experiments, building and testing the system with data collected by several users. We expect to have a range of sensors to feed the system with different types of data (geographic position acquired by a GPS received, cellular location inside a GSM network, WLAN location, etc.). Several context-dependent applications are being built, including task specific and generic applications. The use of different sensors and applications will allow to test the platform developed to manager the users' context and see how suitable the context representation is.

Later, we will use the developed platform as a simulator for testing different mechanisms to estimate new contextual data. Several users will collect positioning information during several weeks, using different technologies (GSM cellular location and GPS data). These raw data will be used to feed the system and test different methods in the estimation of new context data. Based on the geographic dispersion of collected data, the system will identify areas relevant to the user and then identify referential places (e.g., "at home", "at office").

The practical experiments will allow testing the system (performance, usability, support to different types of dimensions, support to different space models, etc.) in a wide number of different situations and thus demonstrate the validity of the proposed solution. Usability should be understood as the facility of applications to access context data and use context representation.

8. Acknowledgements

This work was supported by the Portuguese Foundation for Science and Technology (FCT) (grant SFRH/BD/8279/2002).

9. References

[1] HARTER, A., Hopper, A., A distributed location system for the active office, in: IEEE Network, Vol 8(1): 62-70, 1994.

[2] HONG, J.I., Landay, J.A., An Infrastructure Approach to Context-Aware Computing, in: Human-Computer Interaction (HCI) Journal, 16(2-3), 2001.

[3] JUDD, G., Steenkiste, P., Providing Contextual Information to Ubiquitous Computing Applications, in: Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom 2003), pp 133-142, Dallas-Fort worth, USA, 2003.

[4] KINDBERG, T., et al., People, Places, Things: Web Presence for the Real World. 3rd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2000), MONET, Vol 7(5), 2002.

[5] LEONHARDI, A., Rothermel, K.:, A Comparison of Protocols for Updating Location Information". Technical Report TR-2000-05. University of Stuttgart, 2000.

[6] LEONHARDT, U., Magee, J., Multi-sensor Location Tracking, in: Proceedings of the 4th ACM/IEEE International Conference on Mobile Computing and Networking – MOBICOM98, pp 203-214, Dallas, USA, 1998.

[7] LEONHARDT, U., Supporting Location-Awareness in Open Distributed Systems. PhD thesis, Department of Computing, Imperial College, London, 1998.

[8] LIF Location Interoperability Forum, available at "http://www.openmobilealliance.org/lif/", 2003.

[9] SALBER, D., Dey, A.K., Abowd, G.D., The Context Toolkit: Aiding the Development of Context-Enabled Applications, in: Proceedings of the 1999 Conference on Human Factors in Computing Systems (CHI '99), pp 434-441, Pittsburgh, PA USA, 1999.

[10] WEISER, M., The computer for the 21st century, in: Scientific American, Vol. 265(3): 94-104, 1991.