Location History in a Low-cost Context Awareness Environment

Teddy Mantoro and Chris Johnson

Department of Computer Science, Australian National University
and
Smart Internet Technology Cooperative Research Centre
Building 108, North Road,
Australian National University ACT – 0200, Australia

{teddy,cwj}@cs.anu.edu.au

Abstract

Location awareness is a crucial part of the context-awareness mechanism for ubicomputing. This paper explores how usefull is the location awareness history for an office based low-cost context-awareness environment. Capturing location awareness data into a relational database is simple and feasible in office environment. We use extended SQL to access the location awareness history database to provide direct support for speech commands. The mechanism improve flexibility for developing context awareness application in the Intelligent Environment.
.

Keyword: *Location awareness, Location history, Context aware computing, Ubicomputing, Intelligent Environment.*

1. Introduction

The idea of "computer everywhere" became familiar as Ubiquitous Computing since Mark Weiser presented his vision in Scientific American (Weiser 1991). Weiser also first articulated the idea of Ubicomputing – a shorthand of Ubiquitous Computing – as invisible computation (Weiser 1998a; Weiser 1998b): making many computers available throughout the physical environment, while making them effectively invisible to the user.

The future of Intelligent Environments (IE) is based on the premise that computing and sensing devices will become ubiquitous. Context-awareness mechanisms are seen as the best way for computer applications to operate in such a computing rich environment. Until recently, the availability and cost of gadgets – sensors – and networks have been practical barriers to the ubiquitous deployment and widespread acceptance of context-aware computing. Context-aware computing research has been driven by the development of gadgets, mainly in the form of various sensors of location, proximity and presence – and the lure of the gadget appears as important as enhanced functionality in continuing to drive development of higher resolution, robust, easily deployed, numerous sensors. But in seeking the goal of making our everyday interactions with computers and communications devices unobtrusive and effectively smarter, through context awareness, we already have sufficient hardware devices as the means. The significant challenges lie in making use of already existing devices through the design of enabling systems and software for ease of deployment, network and sensor scalability; through sensor fusion rather than in precision; and enabling wider acceptance through design for user needs, by human factored interfaces and increased human trust in the systems' care for our privacy and security.

Location-awareness is the most important part of context-awareness for mobile computing systems. The off-the-shelf availability and everyday use of a number of moderate-cost mobile devices (handheld and laptop computers), installed wireless and wired networking, and associated location information, lead us to focus our attention on context-aware computing that rests lightly on our everyday environment, and to ask in particular: "what effective location-aware computing can be achieved with minimal, unobtrusive, commodity hardware and software?"

The example of the indoor office environment is a promising target for location-awareness which still presents research challenges. Many everyday tasks performed by people in the office are mediated by quite low precision knowledge of other people's location, such as whether they are in the building, in an area of rooms, in a specific room, and so on. For example, deciding how to contact someone (by email, by phone, face to face); timing the issuing of a reminder for a scheduled event depending on the expected travel time of the event, given locations of person and event and some knowledge or model to the time needed; determining how to deliver an incoming message notification, depending on physical and social location of the addresses. The human decisions associated with these traditional tasks can be assisted by the evolutionary step of providing people with information from location-aware systems, without taking the revolutionary and less acceptable step of their total replacement by computer-mediated communication. Our university office environment with its existing installed network of numerous desktop and mobile computers can itself become our low-cost laboratory, in the usual research paradigm of context-aware computing. With only a small amount of additional sensor and network hardware, this laboratory environment can demonstrate that software incorporating only low levels of reasoning supported by lightweight knowledge is able to provide such helpful location-awareness services.

The system we describe here is the first stage of a system combining location sensing and deduction, camera and voice input and sound output, releasing users from keyboard, stylus and mouse and allowing them to interact with their computational environment more in the way they do with other people.

2. Context Aware Computing

Context-aware computing as a new class of computing software engineering exploits the rapid change in access to relevant information and the availability of communication and computing resources at a given environment of a mobile computer user.

Context is about relationships, focused on the relationship between an interactive device and surrounding elements. Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. Context-awareness is the use of context to provide task-relevant information and/or services to a user wherever they might be. A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task (Dey and Abowd 1999).

The taxonomy of context-aware features (Pascoe 1998) is:

- *Contextual sensing* (the ability to detect contextual information and present it to the user, augmenting the user's sensory system).
- *Contextual adaptation* (the ability to execute or modify a service automatically based on current context)
- *Contextual resources discovery* (to allow context aware applications to locate and exploit resources and services that are relevant to the user's context).
- *Contextual augmentation* (the ability to associate digital data with the user's context – a user can view the data when he is in that associated context).

According to Hull and Neaves et. al. (Hull et al. 1997) *Context-Aware Computing* is the ability of computing devices to detect and sense, interpret and respond to aspect of a user's local environment and the computing devices themselves.

3. Intelligent Environment

The IE is an environment of ubiquitous computing and sensing devices with significant processing done by the sensor (e.g. robust feature extraction from a camera image), knowledge of physical detection relationships between separate sensors and devices, adaptation of IE and its applications to accommodate changes in numbers and type of sensors and devices, resource detection, and modeling.

A very important aspect of IE is providing users with security and privacy: modelling access control with richer contextual information.

We are developing a variant of role-based access control (RBAC) for this. Given the sensitivity of information that is generated and stored in such environment, as well as the many complex interactions that will take place in the intelligent environment, security policies can potentially be quite complex and access control most take into account the richer contextual information.

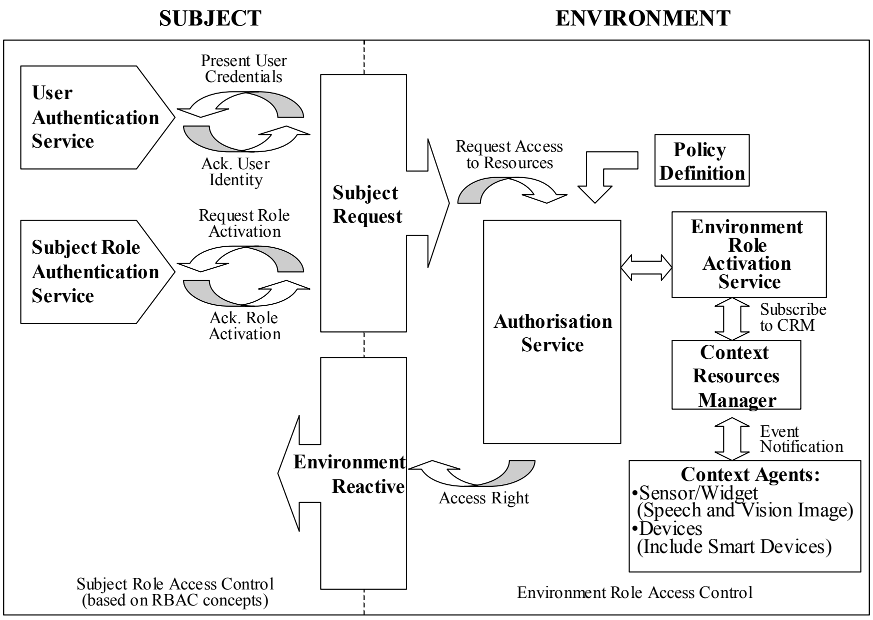

Figure 1
Block Diagram of Distributed Support Architecture for Intelligent Environment

Unfortunately the existing paradigm of traditional software engineering cannot be used as a basis of constructing on intelligent environment. It need to be conceived as an information processing system where a largely invisible and ubiquitous computing infrastructure is integrated and different types of knowledge are properly represented to assist people with a variety of activities in the mobile computing user. To do this, distributed support for IE needs to be developed.

Traditional RBAC offers an elegant solution to the problem of managing complex access control rule sets. The basis of RBAC is the concept of a role, where a role is a grouping mechanism that is used to categorize subject based on various properties. Such properties include job title, user functions or responsibilities.

Although RBAC is very useful for modeling access control in a variety of applications, its roles are inherently subject centric (Covington et al. 2001; Sandhu et al. 1996). It cannot be used to capture security-relevant context from the environment, which could have an impact on access decisions. We extend the RBAC model to allow policy designers to specify such environmental context through a new type of role, namely the *environment role*. In this paper we focus on the subject role and environment role and explore how it can be used and implemented to enable context aware applications (Figure 1).

4. Location Context Awareness

Location is the most important aspect of context for mobile users, such as finding the nearest resources, navigation, locating objects and people. Location in the context-awareness application needs the modeling of the physical environment and the representation of the location (Harter and Hopper 1993; Schmidt et al. 1999)

Numerous location models have been proposed in different domains, and can be categorized into two classes (Jiang and Steenkiste 2002):

- Hierarchical (topological, descriptive or symbolic, such as a room).
- Cartesian (coordinate, metric or geometric, such as GPS).

The other issue is location representation. Most context aware applications adopt a a distributed collaborative service framework that stores location modeling data in a centralized data repository. Location related queries issued by end-users and other services are handled by a dedicated location service. This distributed service paradigm is attractive because of its scalability and modularity. However we need an effective and efficient location representation method to make this work (Jiang and Steenkiste 2002).

We use both the hierarchical and Cartesian location models for representing locations and enabling the rapid changes of location information between distributed context services within a space.

The hierarchical location model has a self-descriptive location representation. It decomposes the physical environment to different levels of precision. We use a tree structure to handle location structure and we store it as an object/entity in a relational database model.

An example for Hierarchical location is the postal address (see Table 1: RoomDB entity):

address=IE://Australia/Canberra/AustralianNationalUni-versity/FEIT/DCS/2ndFloor/Room235.

RoomNo	LocationID	Level	Building	…
235	125	2	DCS	…
103	323	2	DE	…
…	…	…	…	…
211	121	2	DCS	…
223	123	2	DCS	…
237	124	2	DCS	…
…	…	…	…	…

Table 1: RoomDB entity

Cartesian location uses a grid on a physical environment and provides a coordinate system to represent locations. The GPS coordinate system that is defined by (*longitude, latitude, altitude*) or relative location on a space using Cartesian system can be used as Cartesian location (var x, var y, var z), e.g. the user X is located in (18, 5, 15). In this paper, we are not going to explore the shape and other extension location of the object.

Any simple mobile device has a physical location in space. At the spatial dimension, there are some devices (e.g., GPS-based map systems) where the exact Cartesian position in 2D or 3D space is important in defining a sense of absolute physical location. If location information is sufficient in understanding position, the location is considered in relation to other existing objects or sensors.

In the indoor office environment GPS information is neither commonly available nor very accurate.

In an IEEE 802.11b wireless LAN environment the location of mobile devices can be determined at the spatial precision of a group of 3 or 4 individual offices, by measuring the signal strengths of a few most visible access points (Small et al. 2000). This accuracy is sufficient to support the everyday tasks identified earlier.

4.1 Location Context Aware History

Since IE is also a ubiquitous computing environment, we assume that sensor and actuators, simple push button and sliders, and computer access will be available in every area. People can be identified by mobile computing devices (PDA/handheld), vision image recognition, active/passive badge, or by the activity of accessing available resources at static locations.

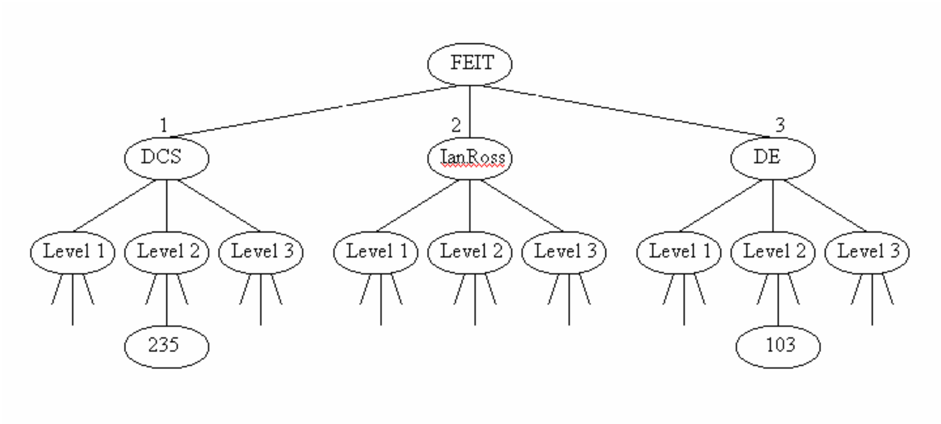

Figure 2
Hierarchical Location Structure.

We can identify the user's location by creating a history database of events, whenever user accesses to identify himself (such as when using iButton or login to the network) or whenever the receptor/sensor/actuator (such as webcam, handheld, active/passive badge) captures the user's identity in a certain location. The following is an example of the location history (table 2):

UserID	LocationID	Date	Time	Device
Teddy	125	13/8/02	04.02	IButton
Teddy	323	20/8/02	05.01	PDA
…	…	…	…	…
Teddy	125	26/8/02	03.02	Sun15
Teddy	125	26/8/02	03.58	Ibutton
Teddy	125	27/8/02	04.05	PC5
…	…	…	…	…

Table 2: HistoryDB entity

5. Context Agents using Speech Recognition

In the IE research area, gesture and speech are becoming a new form of human interaction with the computer. Adding speech recognition and speech synthesis to the architecture of this ubiquitous computing environment can be crucial in improving the distribution of context information. Speech interaction also allows hands-free access to the IE, even when the user is away from the computer (Nieuwoudt and Botha 2002; Rebman et al. 2002).

We develop a context agent prototype using ASR called Speech Context Agent (SpeechCA) in the distributed support architecture for IE. We use the cross platform Java Speech API (JSAPI) that implements speech synthesis and voice recognition.

The SpeechCA needs a grammar to control the recognition process by telling the speaker what words they are expected to say and the patterns in which these words may occur. The advantage of a grammar file is that it makes the recognition faster and more accurate. A sample grammar file is below:

```
grammar javax.speech.SpeechCA;
public <sentence> =   Teddy Mantoro | Chris Johnson |
                      locate | closest | where |
                      printer| phone | extension|
                      occupation |
                      and | or |
                      room |building;
```

6. Exploring Location Awareness History using Speech Commands

A user can give a speech command to SpeechCA and SpeechCA can interpret the instruction using its "dictate" capability. The instruction will be interpreted as a SQL query to a location awareness history database. The result will be in text data type and it can be sent to a speech synthesizer as the IE react ion from this speech command.

Below is the scenario of locating people using a speech context agent:

> At 04.10 pm on 27 August 2002, Chris tried to locate where Teddy is. He asked the SpeechCA.
>
> **Chris**: Locate Teddy, please …
> **SpeechCA:** Teddy is located at room 235 in DCS building, 5 minutes ago. The most probable location of Teddy is in Room 235 in DCS or Room 103 in DE.

When a user gives the instruction *locate /name/*, SpeechCA will respond by generating a query concerning Teddy's location to the location history and room database. It uses SQL to query his whereabouts, at a given time, and extends the query's to try to find a allocation logged with the minimum time difference, in the set for the same day.

If the minimum time difference is close to zero, it means that SpeechCA found Teddy's location and will respond with */name/ is located at room /r.Room/ in /r.Building/ building.* If it is not close to zero, SpeechCA's response will be */name/ is located at room /r.Room/ in /r.Building/ building, /difftime/ minutes ago.* We assume that if the different time is close to zero, the user is still in the same place. In office environment, close to zero can be maximum 10 second for example.

```
# 5 minutes ago ….
Select r.Room, r.Building,
        difftime as min(time() -  h.time) ;
From RoomDB r, HistoryDB h  ;
Where   h.userid = 'teddy' and  h.date = date() and
          r.LocationID = h.LocationID ;
Order  by difftime
```

When SpeechCA find that the minimum time difference is not close to zero, SpeechCA will try to find where the most probable location is in the log history database based on certain policy. For example, the policies for location checkpoints are:

1. The same day of the week
   (assume regular work schedules, to find Teddy's location based on the history data of his location in almost the same time and same day of the week).
2. All the days in a one week range
   (to find Teddy's location based on the history data of his location in almost the same time during a week).

It can be implemented by requesting the following query:

```
Select r.RoomNo, r.Building ;
From RoomDB r, HistoryDB h ;
Where r.userid = 'teddy' and ;
    r.LocationID = h.LocationID and ;
    (r.LocationID in {  ;
        Select h1.LocationID ;
        From HistoryDB h1 ;
        Where dayofweek(h1.Date) = day(Date());
            and (h1.Time <= Time() + 15 and ;
            h1.Time >= Time() – 15) } or ;
            h1.LocationID in {  ;
                Select LocationID  ;
                From HistoryDB h2  ;
                Where (daynumber (Date())- ;
                  daynumber(h2.Date) <= 8) and ;
                  (h2.Time <= Time() + 15 and ;
                  h2.Time >= Time() – 15) })
Order By h.Date, h.Time
```

If the result delivers many hits, taking the most common location or the most recent can be considered. Based on the results of the query above, IE will then respond with: *The most probable location of /name/ is in Room / r.Room / in /r.Building/ {or Room in / r.Room / in /r.Building/}.*

By simple extension, further conditions can be discovered, e.g.:

Find people if they have a meeting (two or more people in the same location, at the same current time).

User: locate Teddy and Chris today.
SpeechCA: They are located at Room 235 in DCS building 5 minutes ago.

7. Discussion of a Low-cost Experimental Environment for Context Awareness

The experimental environment for context awareness should be low-cost and easily deployed.

The aims are for:

- simple deployment and
- wide acceptance through low-cost, by using:
  - easily deployed simple sensors
  - existing widely available software tools
  - existing protocols relying only on wireless and wired TCP networks – making no attempt to fuse telephone (wired or wireless with the TCP network).

It is mainly motivated by the need to demonstrate effectiveness realism, by giving assistance with existing everyday tasks location and time elements of context in a multiple room, multiple linked building office environment available as tool across currently used office-wide devices desktops, semi-mobile laptops, and highly mobile PDAs.

7.1 Known Issues.

Below is the issues in related to the use of location history context awareness:

a. The precision and amount of identification and location info can be obtained and made available by:
   - Desktop computer login and activity can be captured for user identity and location info.
   - Office environment is an indoor environment, so that at the moment we have no off the shelf GPS location but wire LAN location is sufficient.
   - We use very few special purpose sensors, in low-cost and adaptable setup experimental environment.
b. The results from the speech context agent are used to achieve what users would assess as effective, useful location aware computing for their purposes.
c. The design parameter for our distributed software architecture and system architecture location: cheap and indoor means low precision information use existing info sources.
d. Unobtrusive for users by minimized privacy concerns: use own computing and information appliances (desktop, PDAs) little additional wiring, no badges, few cameras, no bats/snoops.
e. Common database tools are used for fusion of current sensor data, historical sensor data.
f. Compression/inference of history data (to reduce the growth of history data, we can use deductive or extract database technique). The purpose of this technique is to make sure that the data will not grow without control.

7.2 Issues ignored

This work ignored several issues such as:

a. Resource discovery, resource description, and zero configuration deployment of new resources such as printers.
b. Scalability. We are not considering of decentralizing the database to thousand of sensors at the moment.
c. Extending to campus and further geographical scope.
d. Security: privacy and sensitivity of location info.

8. Conclusion

In this paper we showed that low-cost context-awareness can be implemented using existing devices and by providing an improved interface between user and devices. We gave an overview of available technology that can be used for acquiring information from the user and the environment. For instance, we use speech context agent to accept speech command. To gain location context information the usage of sensors that create location awareness history data is proposed.

Furthermore we introduced the mechanism for accessing location awareness history using SQL in the subject and environment's role architecture. This mechanism improves flexibility for developing context awareness application in the Intelligence Environment.

9. Future Works

Further research topics that can be considered from this work are:

a. Integration with desirable features of context aware system speech input and output
b. Webcam or CCTV for movement, face counting ('social' context), and face recognition, followed by vision command context agents for context awareness application.
c. Cheap and cheerful making of building 'maps'.

Our goal is to develop interfaces that implicitly can understand what the user wants to do in the office low-cost context environment, which is only possible if a context aware application exists for the devices. It should be based on the user's current context and relevant history of location, activity, identity and time. The context aware application could be designed to use the incoming context information services to use the incoming context services to encode some action.

10. References:

COVINGTON MJ, LONG W, SRINIVASAN S, DEY AK, AHAMAD M, and ABOWD GD. (2001): Securing Context-Aware Applications Using Environment Roles. In: *SACMAT'01*, Chantilly, Virginia, USA

DEY AK, and ABOWD GD. (1999): Towards a Better Understanding of Context and Context-Awareness, Atlanta, Georgia: Georgia Institute of Technology

HARTER A, and HOPPER A. (1993): A Distributed Location System for the Active Office: University of Cambridge

HULL R, NEAVES P, and BEDFORD-ROBERTS J. (1997): Towards Situated Computing. In: *1st International Symposium on Wearable Computer*, pp 146-153

JIANG C, and STEENKISTE P. (2002): A Hybrid Location Model with a Computable Location Identifier for Ubiquitous Computing. In: *The Fourth International Conference on Ubiquitous Computing (UBICOMP 2002)*, Goteborg, Sweden, to appear

NIEUWOUDT C, BOTHA EC. (2002): Cross-language use of acoustic information for automatic speech recognition. *Speech Communication* 38:101-113

PASCOE J. (1998): Adding Generic Contextual capabilities to Wearable Computer. In: *2nd International Symposium on Wearable Computers*, pp 92-99

REBMAN CMJ, AIKEN MW, and CEGIELSKI CG. (2002): Speech Recognition in the Human-computer Interface. *Information and Management* 2011:1-11

SANDHU RS, COYNE EJ, FEINSTEIN HL, and YOUMAN CE. (1996): Role-Based Access Control Models. *IEEE Computer* 29:38-47

SCHMIDT A, BEIGL M, and GELLERSEN HW. (1999): There is more to context that location. *Computer & Graphics* 23:893-901

SMALL J, SMAILAGIC A, and SIEWIOREK DP. (2000): Determining User Location For Context Aware Computing Through the Use of a Wireless LAN Infrastructure, Pittsburgh, USA: Institute for Complex Engineered Systems. Carnegie Mellon University, Pittsburgh, PA 15213

WEISER M. (1991): The Computer for the 21th Century. *Scientific American* vol. 265:94-104

WEISER M. (1998a): The Future of Ubiquitous Computing on Campus. *Communications of the ACM* Vol. **41**:41-42

WEISER M. (1998b): The invisible Interface: Increasing power of the Environment through Calm Technology, Opening Keynote Speech. In: *Cooperative Buildings - Integrating Information, Organization, and Architecture. Proceedings of CoBuild'98* (In: Streitz N, Konomi, S. & Bucrkhardt, H.-J., ed, Darmstadt: Lecture Note in Computer Science 1370. Springer: Heidelberg). Available from http://www.Darmstadt. gmd.de/CoBuild98/abstract/0weiser.html