# User Mobility Model in an Active Office

Teddy Mantoro and Chris Johnson

Department of Computer Science, Australian National University,
North Road, ACT 0200, Australia
{teddy,cwj}@cs.anu.edu.au

**Abstract.** User mobility in an Active Office represents human activity in a context awareness and ambient intelligent environment. This paper describes user mobility by detecting their changing locations. We have explored precise, proximate and predicted user location using a variety of sensors (e.g. WiFi and Bluetooth) and investigated how the sensors fit in an Active Office to provide interoperability to detect them. We developed a model to predict and proximate user location using wireless sensors in the Merino layering architecture, i.e. the architecture for scalable context processing in an Intelligent Environment.

## 1 Introduction

The Intelligent Environment (IE) is an environment with rapid and rich computing processing. Ubiquitous and ambient intelligent computing by embedded sensing devices within the environment will be the key point in IE because of the significant processing done by sensors. As a consequence, the environment provides information on anywhere-anytime devices so that a user may access it on an anywhere-anytime basis too.

Unfortunately, computers and communications systems today are underused because the range of control mechanisms and application interfaces are too diverse. Context awareness mechanisms could be the best way to implement computer applications in IE. It is necessary to consider the mechanism that might allow users to manipulate systems in simple and ubiquitous ways and to make computers more aware of the facilities in their surroundings [4],[15].

This paper discusses an important problem in ubiquitous and ambient intelligent computing, namely how to handle users' mobilities in an Active Office. This work contributes to a model is given to predict and proximate users' locations to understand user mobility in an Active Office using wireless sensor and scalable context processing.

There are significant challenges. These are making use of already existing devices which enable the design of systems and software, ease of deployment, network and sensor scalability, through sensor fusion rather than in precision, and enabling wider acceptance through better design for user needs, by human factored interfaces and increased human trust in the system's care for user privacy and security.

E. Aarts et al. (Eds.): EUSAI 2003, LNCS 2875, pp. 42–55, 2003.
© Springer-Verlag Berlin Heidelberg 2003

In a context awareness mechanism, the question of Who, What, Where and When in IE is responded to by Identity, Activity, Location and Timestamp. In this system the scope of 'who' will be user identity, persona, profile, personalisation/ internationalisation and user model. 'What' is responded to by users' activity that can be represented by user mobility, i.e. the changing of a user's location to another location. Thus, a context awareness mechanism, based on user's locations and time stamps, could deduce a user's activity.

These awareness-mechanisms bring the computer into the user's daily activity. We explore computer capabilities to recognise user location, activity and social context as defined by the presence of other people and to assist people with the variety of activities at work.

This paper describes an Active Office as an implementation model of an IE. An Active office is defined as a normal office, which consists of several normal rooms with minimal additional decorations (minimal intrusive detectors and sensors), without badging people. An Active Office uses wireless communication i.e. Bluetooth and WiFi to enable user mobility. In this paper we look at the use of Wifi rather than the use of Bluetooth.

The Active Office uses a scalable distribution context processing architecture (Merino service layers architecture) to manage and respond to rapidly changing aggregation of sensor data. The Merino architecture has four tightly coupled layers to manage the interaction between users and the environment in the Active Office by detecting the movements and transformations between the raw physical devices/sensors within the environment and the application programs. The infrastructure supports interoperability of context sensors/widgets and applications on heterogeneous platforms [2], [5]. In order for an Active Office to provide services to users, an Active Office must be able to detect its current state/context and determine what actions to take based on the context [2], [5].

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

In an Active Office, the terms of changing user location need to be clearly defined, since we are not using coordinate mapping for the Active Office. The user is considered to be not significantly moving when he is typing on his computer, or opening the drawer of his desk. The user needs to changed location significantly to be considered as his having change location. "Significant" means moving from room to room or moving from one side of the room to the centre or to the other side. We explore changing user location to find the user moving from one location to another location to explain user mobility. In an Active Office we also consider two important variables: speed and location resolution. We will discuss this problem in detail in section Location Characteristics in an Active Office.

This paper describes user mobility in an Active Office as a representation of human activity in the context awareness environment. First, we will present an overview of Merino service layers architecture, followed by a description of a distributed system architecture in an Active Office, mobility in an Active Office, location user model, and precise, proximate and predicted user location. Then, discussion of user mobility will be followed by a conclusion and further research topics arising from this work will be suggested.

## 2 Merino Service Layers Architecture

Merino Architecture has two important parts. The first part is the managing of the interoperability between physical devices/sensors within the IE environment and the program application which manages the interoperability between user and environment. This part contains its abstraction layers i.e. the core Sensors and Device Layers, Context and Device Abstraction Layers, and the highest abstraction level i.e. the Smart Environment Agent Layers [5].

The Sensor Layer is the innermost service layer that represents the range of sensors, which detect the physical environment. The next layer is the Context Layer, which performs core tasks of filtering and aggregating the raw data from the Sensor Layer and ensures maintenance of the interoperability between sensors.

The second part of Merino Architecture consists of a Context Repository and a User Model. A Context Repository is a key element that unifies and manages the whole object in the environment. The Context Layer interacts with the Context Repository. This handles an object's global name structure and it manages the naming/subnaming authority.

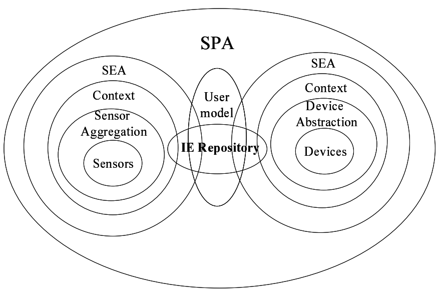


**Fig. 1.** Merino service layers architecture for an Intelligent Environment

A User Model overlaps with a Context Repository. The Merino Architecture treats them almost similarly, the main difference is that the User Model is about people only. The User Model contains personal identity, security and privacy, to confirm the emerging environment in a local and a global space, whereas the Context Repository holds context information from the Context Layer. Data from the Context Repository, which is associated with the user, is stored in the User Model. For example, data from

a movement detector, a WiFi or a Bluetooth detector, and a keyboard activity monitor are held in the Context Repository. However, once it is associated with an individual user, it will be kept in the User Model.
The devices and Device Abstraction Layer are motivated by the need to send data to low level devices in the IE including attributes of the device e.g. a phone number, or a Bluetooth MAC address in a mobile phone. A Smart Environment Agent may communicate with the phone to change its state, e.g. to request switching to a silent mode during a meeting.

## 3 Distributed System Architecture in an Active Office

The key role of a distributed context processing in IE is an IE domain. An IE domain is an administrative domain, which at least contains an IE repository, a resources manager, knowledge based and various sensors (above the level of dumb sensor that communicates using standard protocol).

A resources manager uses sensors/widget agents to have direct communication between sensors/devices and the IE repository. A resources manager also distributes sensor data or aggregate sensor data to another IE repository in the other IE domain.

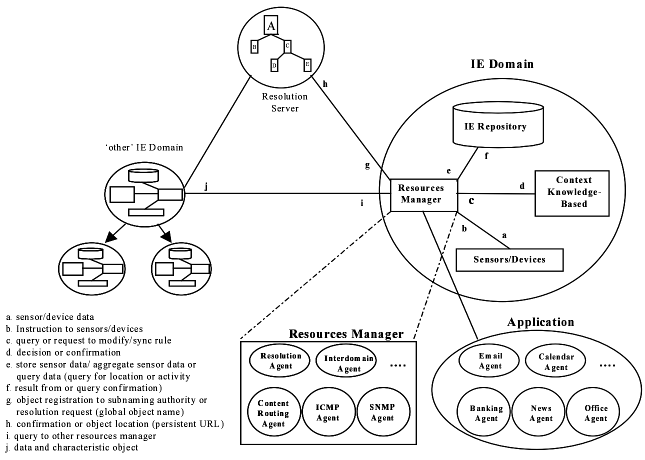


**Fig. 2.** Distributed Architecture for Active Office

The structure of an IE domain can be hierarchical (it might have a single higher level IE domain or have several lower level IE domains) or scattered without level (Figure 2). The communication between IE domains is based on a request from each resources manager (peer-to-peer basis). The resources manager will send a request to the resolution server to require information about the object. The resolution server accepts the object's global name and uses persistent mapping from the object's global name to send the persistent location (persistent URL) of the object to the resources manager. Finally, the resources manager sends a request to a relevant resources manager in another IE domain to obtain the object including the characteristics of the object.

Every object in the IE domain is defined to be self-describing object data-structures. This means that the structure, as well as the value, is always included in the object's content. This approach has two advantages: firstly, it avoids the complexity of separate schemes and maintains an object across multiple distributed servers. Secondly, it permits meta-tools e.g. browsers to manipulate the object without knowledge of the specific contents [11].

Communication within the IE domain uses logical multicast. This means that communication between the IE repository and the sensors could be implemented using IP multicast on a well-known group 'content filtering' in the receivers or message server distribute content routing/filtering (such as Elvin or Spread) for an automatic update of the content object purpose [13]. A particular object has a unique identifier and is located using a search that starts in the local IE domain, and expands to adjacent and higher lever IE domains such as a resolution server (using persistent URL or Handle system).

For example, Anna works for the CS department. When she goes to the department of Engineering the sensor at the CS department would aggregate that Anna is not present and another sensor at the department of Engineering would detect an unknown object and ask its local resources manager for the address of the server for this object. The local resources manager multicasts the request to the resolution server and discovers the correct home server for the object. The object's location information in the home server is then updated appropriately.

An IE repository keeps all object information including the characteristic of the object. It holds data about any rapid changing of the object. It contains all relevant data within the IE domain such as the sensor data, the aggregate sensor data, the context data and the rich context data. The IE repository uses a distributed object database for Context Repository and User Model purposes.

A resources manager is a network management map application that provides information about the status of objects (devices/sensors) in the IE network. The resources manager detects, controls, manages and concludes the functionality of all objects in the IE domain.

The resources manager will show any failures in low level devices i.e. sensors, access points, hubs, bridges and routers. Furthermore, the resources manager will detect any traffic problems and will identify the what, the where, the cause and the time-length of the problems. The resources manager has the capability to control and manage the functionality of an IP network using the Internet Control Message Protocol (ICMP) and 'trap' mechanism using the Simple Network Management Protocol (SNMP), so that the fail-safe mechanism can be implemented.

To set up the resources manager a set policy (knowledge-based) needs to be established that determines the acceptable levels of traffic, broadcasts and errors on any devices/sensors at any segment.

The knowledge-based context contains context rules of the interaction between the user and the Active Office environment. Rules are used to represent heuristic, or "rules of thumb" which specify a set of actions that need to be performed in a given situation. A generic rule is composed of an *if* portion and a *then* portion. The *if* portion of a rule is a series of patterns which specify the facts where the rule would be applicable in the IE domain. The process of matching the facts to the patterns (pattern-matching) is done by a specific agent in the resources manager called an inference engine agent. The inference engine agent will automatically match the facts against the patterns and determine which rules are applicable to the context. The *then* portion is a set of actions that needs to be executed when the rule is applicable to the situation (context). The actions of applicable rules are executed when the inference engine agent is being instructed to begin the execution. The inference engine agent selects the rules, and then the actions of the selected rule are executed. The process continues until there are no more applicable rules.

The knowledge-based context also has a dynamic knowledge cooperation aspect [3] to allow the rules and the sensor/devices to dynamically affect the actions of the service selection process within the IE domain or the inter-IE domain.

In the IE domain, all objects can be distributed to other physical locations or logical locations. Every object has a global resolution name and the global resolution name needs to be registered in the resolution name server. IE resolution is a global resolution server, which can be implemented using URN, Persistent URL (PURL), or Handle Server for persistent global mapping/resolution purposes. The resolution mechanism is identical to the DNS for the Naming Authority Pointer (NAPTR) Resources Record (RR) to delegate the lookup's name [1],[10]. We use the Unique Resolution Name (URN) as a global unique name and the Unique Resolution Locator (URL) as a locator to locate any object, anywhere, anytime.

## 4 User Mobility in an Active Office

A location is the most important aspect to provide a context for mobile users, e.g. finding the nearest resources, navigation, locating objects and people. A location in the context awareness application needs a model of the physical environment and a representation of the location [4], [7], [12].

This paper explores user mobility in an Active Office. We began from understanding user location, then changing location from a current location to another location. By analysing the history data we can get the pattern of the user mobility. We strongly believe that by understanding user mobility we can better understand user activity.

In the following part we will explore the user location model and location characteristics in an Active Office to match the location model in an Active Office.

### 4.1 User Location Model

Numerous location models have been proposed in different domains, and can be categorized into two classes:

- Hierarchical (topological, descriptive or symbolic).
- Cartesian (coordinate, metric or geometric).

The other issue is the location of representation. Most context awareness applications adopt a distributed collaborative service framework that stores location modelling 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 [6].

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

Cartesian location imposes 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 in space using the Cartesian system can be used as a Cartesian location (var x, var y, var z), e.g. the user X is located at (18, 5, 15). In this paper, we are not going to explore the shape and other extension locations of the object.

Any simple mobile device has a physical location in space. In 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 the location information is sufficient in understanding the position, the location is considered in relation to other existing objects or sensors.

GPS does not work in an indoor Active Office environment because it has approximately 5-15 meters accuracy and the signal is too weak. It works only for an outdoor environment.

In the Wireless LAN environment, the location of mobile devices can be determined with spatial precision for a group of three or four individual offices, by measuring the signal strengths of a few of the most visible access points [14]. This accuracy is sufficient to support everyday tasks in the Active Office. Both IEEE 802.11b and Bluetooth can be used to find the proximity of physical user locations.

### 4.2 Location Characteristics in an Active Office

Since an Active Office 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. The users can be identified by mobile computing devices (PDA/handheld), vision image recognition, or by active/passive badge. Users also can be identified by activity when accessing available registered resources at static locations.

In an Active Office we assume that the user has a regular work schedule, has some routine activity that can be used to predict his location in a specific timestamp.

User Location in an Active Office implies the ability of IE to understand a user changing location on a ‘significant’ scale. The user is considered to be moving if his location is changing. The term of “change location on a ‘significant’ scale” means that the change in user location affects the user’s access to available resources. For example, picking up the phone or picking up a glass from the table while the user is typing at the computer is not a significant location change. By ‘significant’ we mean when the user moves from room to room or moves from one side of the room to the centre, or to the other side, or he moves from one building to another.

The approximate significant location scale could be affected by two important aspects: speed and location resolution. A sensor continuously senses on a real time base, i.e. the speed of a user changing location in the office between 0-20 Km per hour and a location resolution of about 1-3 meters in the office. The slower the user changes location, the more the sensor detects and delivers service information about the location the user passes through.

The Active Office can be designed to understand ‘significant’ change in user location by using sensors that can measure proximate location. When the user moves, it means that the user’s access to the resources also changes. The availability of resources depends on the user location. For example, when the user moves from his room to a meeting room, the proximity or the availability of resources, e.g. a printer, may also change.

We believe that accessible resources for users in an Active Office can be detected based on their proximate location. We also believe that a hierarchical location model will be more relevant than a Cartesian location model because a hierarchical location model could scale room and building, while the technology for gridding/mapping the office using a Cartesian model is not available yet at the time of writing.

## 5 Sensor Aggregation for User Location

User locations in an Active Office can be categorised as follows:

1. Precise Location
2. Proximate Location
3. Predicted Location

The above category is based on the sensor’s capability in covering an area. The problem is to combine these known location data to determine the user’s actual location for office activity purposes which is a different precision matter and sometimes user location is not available at all at arbitrary times. How to characterise the pattern of user location based on aggregate current sensor data and history sensor data is also a problem.

### 5.1 Precise User Location

Precise location is based on sensors that cover less than a meter range, e.g. swipe card, keyboard activity, biometric sensor/finger-print, iButton, etc.

By registering the location of desktop computers, swipe cards, iButtons, fingerprints, as sensors, an Active Office will have more precise information about user locations than from proximate sensors, e.g. WiFi or Bluetooth.

### 5.2 Proximate User Location

Proximate location is based on a sensor that covers more than a meter range, e.g. WiFi, Bluetooth, WiMedia, ZigBee, active/passive badge (depending on the range), voice recognition (microphone), face recognition (digicam), smart floor, etc.

Proximate location is detected by Wireless LAN is an interesting proximate sensor in an Active Office because it can be used to access the network and also it can be used to sense a user location within the scale of a room or an office. Bluetooth, as a wireless personal area network, favours low cost and low power consumption, over a range and peak speed. On the other hand, WiFi as a wireless local area network, favours higher speed and greater range but has higher cost and power consumption. The range of the Bluetooth to sense another Bluetooth in a closed space, such as an Active Office, is about 3 meters for class 2 and 25 meters for class 1. The range of WiFi to sense a user with a WiFi device is about 25 meters.

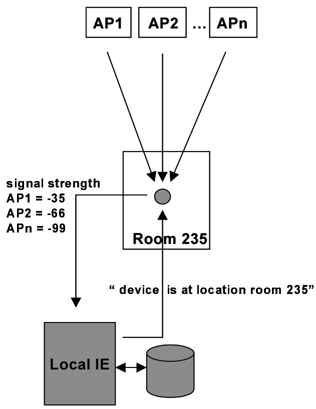


**Fig. 3.** Measure by devices of WiFi APs' signal strengths

Bluetooth permits scanning between devices: when Bluetooth capable devices come within range of one another, the location of one Bluetooth will be in the range of the other Bluetooth. We can use Bluetooth capable devices as sensors or an access point to sense a user with Bluetooth capable devices. From our experiment, unfortunately a Bluetooth signal strength is not useful enough to sense a user's location. We used the Bluetooth access point as a sensor for several rooms within the range without measuring any signal strength. For example, when a user is close to a certain access point, the user's location will be proximately close to the access point and it could represent user location from several rooms.

WiFi does not only have a higher speed and longer range than Bluetooth but the signal strength of Wi-Fi also can be used to detect user location. We have two scenarios to determine user location using WiFi. Firstly, by determining the signal strength from the WiFi capable device which stores data in a local IE repository with the server sending the current user location. Secondly by determining the signal strength from the WiFi access points and storing the signal strength data in the local IE repository with the server sending the user location. The difference between these scenarios is that in the first scenario the process of sensing is in the devices, so we need a good user's mobile device, whereas in the second scenario the process of sensing is in the access point, so we do not require a user's mobile device with a high capability.

We use a self organizing map (Kohonen map) approach of artificial neural network to cluster the signal strength data by giving random weight, doing normalisation, calculating Euclidian distance, finding the winner for clustering. The training is continued until the learning rate is zero and the final weight is obtained, which leads to output activation and cluster winner allocation [8]. A self organizing map using a Kohonen map is suitable to cluster locations based on signal strength data measure in the local IE. Using this method we can directly determine current user location.

In our experiment using WiFi, we used 11 access points to measure signal strength on two buildings, 5 in the Department of Engineering and 6 in the Department of Computer Science. The result was good enough to predict current user location. On the 2nd level of the Department of Computer Science, we found that most places had a good signal from more than two access points and we could predict accurately (96%) in rooms of 3 meters width. On the 3rd level, where not all rooms/locations were covered by more than one access point, we had only a reasonable degree of accuracy (75%) in predicting a user's current location.

### 5.3 Predicted User Location

Since IE is also a ubiquitous and ambient computing environment, we assume that sensors and actuators, simple push button and sliders, and computer access will be embedded and available in every area. People can be identified by the activity of accessing available resources at static locations or by sensing the user's mobile computing devices (PDA/handheld).

**Table 1.** HistoryDB entity

| UserID | LocationID | Date | Time | Device |
|---|---|---|---|---|
| TM | 125 | 13/8/02 | 04.02 | Ibutton1 |
| TM | 323 | 20/8/02 | 05.01 | VR10 |
| … | … | … | … | … |
| TM | 125 | 26/8/02 | 03.02 | Sun15 |
| TM | 125 | 26/8/02 | 03.58 | FR4 |
| TM | 125 | 27/8/02 | 04.05 | PC5 |
| … | … | … | … | … |

We can identify the user's location by recording a history database of events, whenever a user accesses to identify himself (such as when using iButton, typing at a desktop computer or logging into the network) or whenever the receptor/sensor /actuator (such as webcam, handheld, active/passive badge) captures the user's identity in a particular location.

We develop a history data from precise users' locations (see a section on Precise User Location). Table 1 is an example of a locations history data in an IE Repository. The history data above can be used to predict user location. It is also possible to develop a probabilistic model to find the most probable location of a user in the log of history data based on a particular policy. We have implemented the policies below for a user's location checkpoints i.e. [7]:

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

We use simple extended SQL query to implement the above policies to find user location. In addition, we also develop SpeechCA (speech context agent) using speech synthesis and speech recognition by cross platform Java Speech API (JSAPI). It makes the Active Office recognise and understand instructions from the user when he questions the Active Office, then at different levels, he queries the history data and feeds the answer to the speech synthesizer [7] as the Active Office responds to the user's query.

## 6 Discussion

In this part we discuss how the Active Office determines user location based on three location categories: precise location, proximate location, predicted location.

In Figure 4, we show how an Active Office processes the information to determine a user's location by aggregating the relationship between user data and location data.

Aggregate precise location has first priority and is followed by proximate and predicted locations respectively. This means that when the Active Office receives information from aggregate precise location, then the current user's location is determined. If not, then we check using aggregate proximate location data.

In the case of there being no user location at all in aggregate, precise, or proximate location then predicted location will be used, as in the case, for example, when user X works in room Y which is not covered by WiFi or Bluetooth access points, and there is only a workstation present. If at the time of query user X is not accessing the workstation, history data is used to find the most probable user location based on data for the same day at the certain time of the week or using all the days at the certain time in a one week range.

Our experiment using Wireless LAN i.e. WiFi and Bluetooth as proximate location, gave a significant result to sensing a user's proximate location. The result can be improved by getting interoperability between sensors to aggregate sensor data. From our experiment, compared with Bluetooth, WiFi seems the better way to obtain precise location data using proximate sensors.

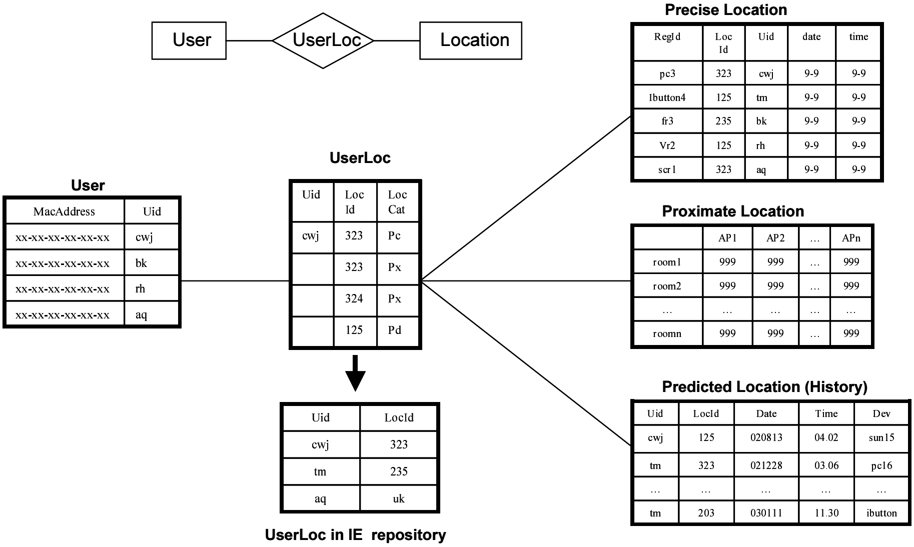


**Fig. 4.** Aggregate user's locations in an Active Office

## 7 Conclusion

This paper proposed an implementation model of an Active Office environment which is simple, efficient, scalable, fault tolerant and applicable to a range of heterogeneity computing platforms.

This paper has described the requirement of recognizing user locations and activities in the Merino service layer architecture, i.e. the architecture for scalable context processing in a ubiquitous computing environment and ambient intelligent (embedded sensing devices) environment.

In an Active Office, a user has a regular work schedule. A user has a routine activity that can be used to predict his location in a specific timestamp. A user's activity can be represented by user mobility, and user mobility can be seen from the user's changing location on a significant scale. So, in an Active Office once we can capture a user's location then we can map a pattern of user mobility.

The users' locations in an Active Office can be categorised into three, i.e. precise location, proximate location, predicted location. The category is based on the sensor capability to sense the area and the use of history data. The priority order in deciding a current user location is first precise location followed by proximate location, and then predicted location.

Our experiment using WiFi and Bluetooth in determining proximate location in an Active Office has shown good results in sensing user location. At present, WiFi rather than Bluetooth seems to be the better way to fix precise location using proximate sensors . The result can be improved by developing interoperability between sensors to yield aggregate sensor data.

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

- Aggregation of smart sensors (more interoperability between sensors) using Elvin notification system to notify the difference between current location and previous notification.

- Managing location information in Merino service layer architecture:
  - format representation,
  - conflict resolution,
  - privacy of location information.

## References


1. Daniel, R. and Mealling. M. Resolution of Uniform Resource Identifiers using the Domain Name System. RFC 2168 (1997)
2. Dey, A. K., Abowd G. D., et al.: A Context-Based Infrastructure for Smart Environments. 1st International Workshop on Managing Interactions in Smart Environments. MANSE'99. (1999)
3. Gajos, K.: A Knowledge-Based Resource Management System For The Intelligent Room. Master Thesis. Massachusetts Institute of Technology. Massachusetts. (2000): 58
4. Harter, A. and Hopper A.: A Distributed Location System for the Active Office. IEEE Network Vol 8, No 1, (1994)
5. Kummerfeld B. and Quigley A. et al.: Merino: Towards an intelligent environment architecture for multi-granularity context description. Workshop on User Modelling for Ubiquitous Computing. Pittsburgh, PA, USA. June 2003
6. Jiang, C. Steenkiste P.: A Hybrid Location Model with a Computable Location Identifier for Ubiquitous Computing. The Fourth International Conference on Ubiquitous Computing (UBICOMP 2002). Goteborg, Sweden. (2002)
7. Mantoro, T. Johnson C. W.: Location History in a Low-cost Context awareness Environment. Workshop on 'Wearable, Invisible, Context awareness, Ambient, Pervasive and Ubiquitous Computing'. Australian Computer Science Communications. Volume 25, Number 6, Adelaide, Australia. February 2003
8. Mantoro, T.: Self-Organizing Map and Digraph Analysis of Electronic Mail Traffic to Support Interpersonal Resources Discovery. Master Thesis. Department of Computer Science. Asian Institute of Technology. Bangkok, Thailand. December 1994: 96 (CS-94-44)
9. Manzoni, P. and Cano J.: Providing interoperability between IEEE 802.11 and Bluetooth protocols for Home Area Networks. Computer Networks 42 (1).2003: 23–37
10. Mealling, M. and Daniel R.: The Naming Authority Pointer (NAPTR) DNS Record. Updated RFC 2168. (2000)
11. Schilit, W. N.: A System Architecture for Context awareness Mobile Computing. PhD Thesis. The Graduate School of Arts and Sciences. Colombia University. Colombia. (1995) 144

12. Schmidt, A. Beigl M. et al.: There is more to context that location. Computer & Graphics 23 (1999): 893–901
13. Segall, B. Arnold, D. et al.: Content Based Routing with Elvin4. In Proc. AUUG2K (June 2000)
14. Small, J. Smailagic A. et al.: Determining User Location For Context awareness Computing Through the Use of a Wireless LAN Infrastructure. Institute for Complex Engineered Systems. Pittsburgh, USA. (2000)
15. Weiser, M.: Some Computer Science Issues in Ubiquitous Computing. Communications of the ACM. 6(7) (1993).: 75–84