# Location Service in Mobile Computing Environments

**Ulf Leonhardt, Jeff Magee, and Paul Dias**

Dept. of Computing, Imperial College, 180 Queen's Gate, London SW7 2BZ, UK
{ul, jnm, pbd}@doc.ic.ac.uk

*With the advent of mobile computing devices and cheap location sensing systems, location information has become an important resource both for mobile and 'desktop users'. In this paper, we describe some key concepts a scaleable ubiquitous location service should be based on. Firstly, we show how such a service can accommodate multiple location sensing systems. Secondly, we discuss hierarchy-based access control policies as a flexible and powerful mechanisms to protect users' privacy. Thirdly, we address some issues concerning the visualisation of location information.*

## 1 Introduction

With users and computing equipment becoming increasingly mobile, location information is vital for many purposes, such as communication channel setup, navigation, and systems management. We envisage that environments that have a high proportion of mobile computing and communication devices will employ a general ubiquitous location service to cater for a wide spectrum of applications. The very basic services provided by such a system are:

- to find the location of a specified located-object.
- to find all located-objects at a given location.

Both types of queries should be answered with a well-defined level of confidence.

The location of objects can be measured in different ways for different types of objects. Some methods will only be available in certain geographic areas. Hence, the input of various sources of location must combined in order to achieve ubiquitous service coverage. Moreover, a higher (spatial and temporal) positioning accuracy can be achieved by exploiting overlaps among sightings from different sources. Statistical and other models (cf. Liu's work [2]) can be also used to interpolate and extrapolate locations.

In this paper, we argue in favour of combining all the location information that is available from the various primary sensing systems, along with mobility patterns and other heuristics, under the roof of an abstract location service that provides transparency from the hardware and software that actually gathers the sightings. Such a service would support context-aware applications (as described by Schilit in [18]), as well as location-aware services (such as Advanced Traveller Information Systems [19]). We describe an abstract framework that will be capable of accommodating arbitrary location sensing systems.

The automated gathering of location information potentially infringes people's privacy if too much data is disclosed to the wrong party. Therefore, our work investigates mechanisms to exactly specify and supervise the level of access to location data that is wanted. [1]

Finally, we wish to support non-local roaming, and non-local location lookup. Essentially, we aim at the notion of global coverage for the location service. We investigate an architecture of federated location servers to build a global distributed location service.

In section 2 we will describe our conceptual framework for the location service, followed by a discussion of the issues of security and server federation in sections 3 and 4, respectively. Finally, we address the visualisation of location information in section 5.

## 2 Conceptual Framework

**Definition:** *A **located-object** is an object whose location is of interest to the location service.*

[1] However, we cannot solve the underlying problems of society with our technology.

The term located-object (first used by Schilit [10]) refers to all the real-world entities (people and hardware) that can be tracked by the location service.

In the remainder of this section, we define a model for location data at different levels of abstractions. A cell serves to represent raw location data, zones are used for location tracking, and hierarchical domains are the interface for client interaction and management operations.

### 2.1 Cells: Source of primary sightings

We assume that location sensing systems represent an object's location in terms of a well-defined geographical area (e.g. a room, a square on the map, an IR or RF cell, etc.). In the case of GPS, this area is given by the coordinates and the accuracy margins.

**Definition:** *A **cell** is a well-defined geographical area. A **point of sighting** (POS) is an abstraction for the physical location of a monitoring device that reports located-objects in its vicinity. (Examples: badge sensor, workstation, cellular network base station, GPS receiver). A point of sighting is associated with a well-defined geographical area, a cell.*

A point of sighting can also be an abstract location generated by an underlying location tracking system (e.g. a GPS location, the cell being defined by the accuracy interval). An object can be in different cells at the same time if, and only if, the cells do overlap geographically.

### 2.2 Zones: A flat integrated location space

**Definition:** *A **location zone** identifies the geographical area defined by the intersection of one or more cells. A **zone space** $Z_{C,X}$ is defined by the set C of permitted point of sightings (i.e. cells), and the set $X \subseteq P(C)^2$ of permitted overlaps between cells.*

A location zone space provides a set of mutually exclusive locations for located-objects. By treating all the different sources of sightings essentially as cells, we achieve their integration. User-specific location tracking and prediction is done best in such an integrated setting.

By defining the zones as the overlaps between different cells, we achieve potentially a higher positioning accuracy. The zone space can also be partitioned into independent geographical coverage areas. Hence zone space computations can be distributed.

We wish to track different kinds of located-objects, thus not all the objects will be visible to all the location sensing system. Also, the location of some objects will be measurable with higher accuracy than that of others. Consequently, more than one zone space needs to be supported by the location service. We believe, that there should be a single *complete* zone space, and several derived *restricted* zone spaces.

Restricted zone spaces can be computed automatically, given a complete zone space and the set of cells we want it to restrict to. The converse procedure requires additional geographical information to determine the overlaps of the new cells.

A potential problem are moving cells. Here, the permitted overlaps must be computed and updated dynamically. If the location of the moving sensor or base station is being tracked as well, no external intervention would be necessary. Interpreters of the location information would have to be aware of its relative character.

Instead of using moving cells, we can use a fixed grid of cells to capture the location of freely moving real cells or non-cellular location systems (such as GPS). In this case, we can achieve an arbitrary degree of accuracy by using finer-grained grids. The benefit of this approach is that the zone space remains fixed, which should reduce its conceptual and implementation complexity.

### 2.3 Hierarchical location domains

We use location domain hierarchies to present location information to clients and for management purposes. A *hierarchical* scheme was chosen in order to support multiple-resolution presentation of locations in a scaleable and flexible way. Location domains can be defined independently of the underlying location sensing systems.

**Definition:** *A location domain is a named (abstract) location for a group of location zones, and/or other location domains (no cycles permitted). The set of zones and domains contained in a location domain may change over time.*

There is a single root location domain of which all other location domains must be direct or indirect members. Basically, the location domain hierarchy is a fixed structure. Mobile object's move within the hierarchy by joining and leaving domains. Membership implicitly propagates up the hierarchy.

However, domains themselves can be mobile. Hence, we envisage to have dynamic links in the domain hierarchy that change as the domains move around. A distinction between static and dynamic links in the domain hierarchy may be necessary (e.g. for naming purposes).

Each domain has a set of associated location zones in order to map location from the zone space into the domain space. If an object is located in one of the zones belonging to a location domain, it automatically becomes a member of that domain.

Non-mobile objects can also be location domain members. Domain membership queries can thus accomplish basic location-dependent mapping functions (Example: find the nearest telephone).

The location domain hierarchy provides a structure both for client interaction and management operations. It can be defined independently of the characteristics of the underlying location sensing systems. It is also a suitable framework for access control and server federation.

## 3 Security

Access to location information should be subject to access control. On the other hand, the usefulness of the location service depends on location information being accessible. Hence we need flexible yet powerful access control mechanisms.

Traditional authorisation methods can be employed to completely block off queries from unauthorised parties. However, often the outcome of a query itself determines whether the disclosure of its result is desirable. (Example: A policy "Do not disclose the people present in the meeting room").

Following Sharma [8], an alternative concept of access control can be derived from the hierarchical structure of location information in general and the location domain space in particular. We use *visible domain sets* (VDS) in access rules to specify authorisation to 'see' a located-objects in certain domains. In this way, we restrict the available positioning accuracy in a flexible way.[3] If the object's location is then queried, the disclosed information can be automatically restricted to the VDS.

Location access policies apply to a pair of querying and target object. Additionally, constraints on the time of the query, the location and the policy set of the querying objects may be used to modify the scope of the policy. Access policies are specified by access rules. Typically, access rules are defined using domains (cf. domain service [12]) as scopes for source and target objects. In this setting, a user can also specify his or her private location access rules in term of his or her personal user representation domain.

Access control can be enforced by user agents or a centralised service. Since the user agents approach has scalability problems, we favour a hybrid solution where a home area server acts on behalf of a its local users.

## 4 Federation

Our location service should support wide-area roaming as well as remote location lookup. We aim to achieve both by peer-to-peer interaction between location servers.

### 4.1 Roaming

Located-objects roaming in a foreign location service area should (if authorised and authenticated) be able to access the local location service, and the local location service should be able to trace the foreign object. Roaming support generally (cf. Mobile IP [4], GSM [7]) requires a temporary residence for the object to be set up by following certain registration and hand-over protocols. Registration sets up a temporary residence of the located-object, whereas hand-over transfers state and communication channels from the previous to the new site.

Registration with the location service needs to establish a representing name for the object, and its name in all the location spaces it is visible in (e.g. Badge ID, UNIX login, IP home address). Further, a security profile governing the access to the foreign object's location has to be constructed. Additionally, mobility profiles may

---

[2] Read: power set of *C*.

[3] Typically, such a VDS contains the root location domain.

be installed. The necessary information is either retrieved from the object's home domain (or a centralised service), obtained from the located-object itself, or from the object's previous location (hand-over).

Often, the location subsystems (Badge service, GPS service, cellular phone network) will perform independent registration procedures. The location service needs to exploit the information gained by the subsystems about a new located-object.

Hand-over of located-objects between sites accomplishes the transfer of state information from the old to the new site. Here, state means the location trace of the object (if it is maintained at all). Again, subsystem perform hand-over of their communication channels independently. In any case, the location service must coordinate the registration and hand-over in its primary subsystems. The location service can trigger the registration/de-registration and hand-over of located-objects in its subsystems (e.g. Active Badge service).

### 4.2 Remote location lookup

The different sites can be integrated into a logically centralised service (analogous to the approach of AFS to file system distribution [6]). Conceptually, there is one uniform location service covering all participating sites. All clients see the same location domain space, which is distributed over several location servers. The distribution of the domain space is transparent to the client. Client mobility is supported by the uniformity of the location domain space.

If querying a (location or object) domain, the client query would more or less directly be forwarded to the location server it sits on. Availability could be increased by replication of location domain, requiring update propagation with a well-defined consistency model.

It is unclear whether this approach addresses the need for privacy to a sufficient extent. A global trusted service may not appear very trustworthy to some people.

Following the analogy to distributed file systems, we can also adapt an approach similar to NFS. Here, the individual sites would choose to mount parts of the location domain space from other servers. The domain space seen by clients at different sites will not be identical, but the implementation may be easier. Foreign servers would appear just behave like clients issuing standing or on-off queries. Roaming object support would require identification of the object's home domain by the host domain to set up a temporary residence.

## 5 Visualisation

Often, location information will directly be presented to users via client programs for navigation, informational and management purposes. There are different representation schemes at various levels of abstraction (domains, maps, virtual reality), and they can be rendered with different levels of graphical sophistication (text, 2D-graphics, 3D-graphics). The choice very much depends on the requirements and the available resources (bandwidth, storage capacity, performance).

We are currently investigating the idea of using 'real' location information to position virtual objects in virtual worlds, such as multi-user dungeons. Thus the virtual world can simulate essential aspects of the 'real' world in real-time. Such technology could be applied to build virtual reality interfaces to monitoring and management applications, as well as for training and entertainment purposes.

A less sophisticated virtual-reality-like interface is the Active Map, that is, a map that shows the current locations of located-objects (Figure 1). Active Map implementations have been described by several researchers [10, 1, 14]. Mark Weiser [15] believes that a map display alone is of use by giving users a feeling of community. Additionally, further useful features, such as user search, alarms, and paging can be included.

The location domain model presented in this paper seems to lends itself quite naturally to presentation both by a virtual reality interface and by an Active Map. Clients wishing to implement such interfaces will typically submit standing queries to the location service, such that an object's movement can be followed on the map or in the dungeon with minimal latency.

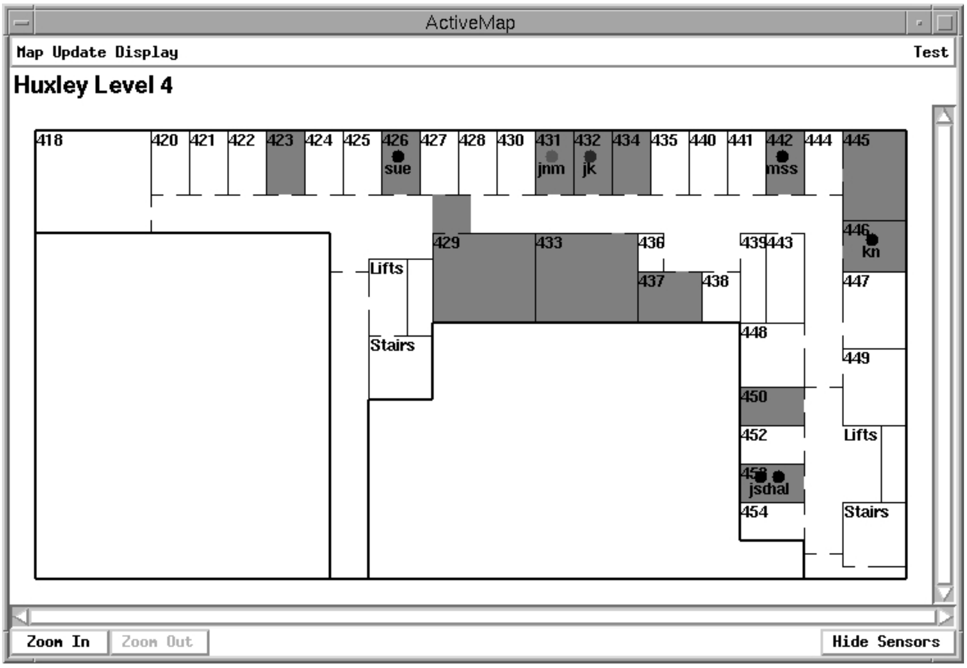


Figure 1: Active Map with highlighted areas indicating coverage of Active Badge sensors

## 6 Implementation Status

Currently, we are actively using a location service based on Active Badges and `rusers` demons. The sightings from both sensing systems are gathered by independent subsystems, and then combined by the location server. At the moment, we use one server per service area. Further, we have implemented and tested roaming support for located-objects that are visible to the Active Badge service.

The server software uses the `Regis` platform [3] which provides distribution, multi-threading, and message-passing communication. CORBA-interfaces have been implemented for the Active Badge subsystem. On the client side, we use Java applets [17] to access the service (see Figure 2), thus catering for heterogeneous client platforms (currently SunOS, Solaris, Windows95, Linux). We have also implemented a collection of Tcl/Tk based client applications, including an Active Map of our department (Figure 1).

In our experience, both usefulness and acceptance of the location service are crucially depending on its temporal and spatial coverage. We have also found that the location service itself is, for most users, not incentive enough to wear their Active Badges on a regular basis (other useful badge applications can achieve this: X-teleporting, badge-triggered door locks, etc.) At the moment, the bulk of the sightings in our system is generated by the UNIX `rusers` service. Further, Active Badge and `rusers` demons tend to be complementary in their temporal coverage. Often, Active Badges are obstructed if worn while working on the computer, and `rusers` demons are not very useful if somebody is standing or walking around.

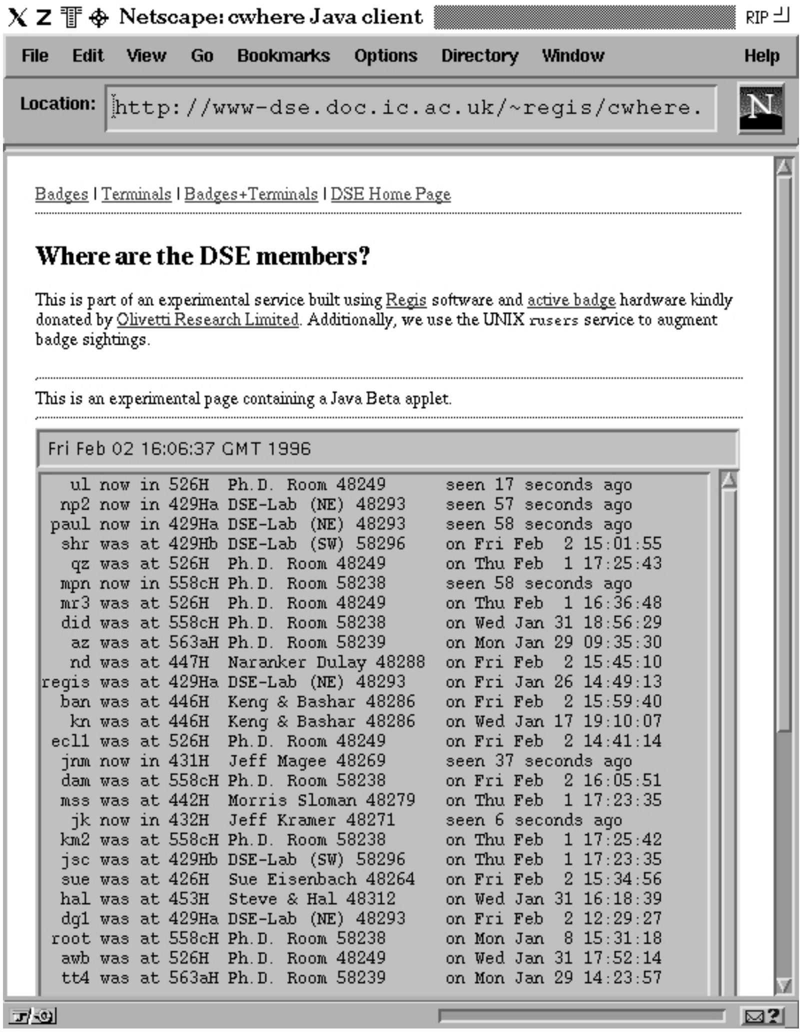


Figure 2: Java client to the location service

## 7 Related Work

Schilit, Spreitzer, and Theimer [10, 11, 9] have pioneered the idea of a location service. While they use various sources of location information, they apparently lack the notion of hierarchical locations. They have also explored the user-agent approach which we have decided not to follow in our work.

Harter and Hopper [1] have presented a location service based on the Olivetti Active Badges. We use an architecture based on servers local to administrative domain and server federation similar to theirs.

Rizzo et al [5] discusses the problem of using multiple location sensing systems. They unify the sightings by expanding them with information from a location directory. This expansion is in fact equivalent to the mapping from the cell space into the zone space. Therefore, we can also use similar update algorithms. However, they lack the notion of hierarchical presentation of location information and hierarchy-based access control.

## 8 Conclusions and Future Work

We conclude that provision of a universal location service is a vital platform support component for mobile computing. In order to achieve ubiquitous coverage, multiple location systems must be incorporated, and server must support wide-area roaming as well as remote location lookup. Fine-grained access control to location information is necessary to prevent violations of people's privacy.

We have described a conceptual framework for handling location information at different process stages, and to present locations in a well-structured hierarchical domain model. The domain model lends itself to hierarchy-based access control policies which allow for fine-grained protection of users' privacy. We also believe that location domains are a suitable abstraction to build a global location service.

In the future, we will develop further our prototype implementation. We will aim to integrate GPS, and to explore our ideas of server federation.

Being itself part of a mobile environment, the location service will often be used by mobile computers (e.g. by an Active Map application on a personal organiser). Also, some mobile devices can sense their own location (those equipped with GPS receivers, for example). Some mobile computers will also require a certain level of access to the location services during states of disconnection. Hence, our design and architecture needs to address these issues.

*Acknowledgements*

This work was supported by a Research Training Fellowship of the European Union's Human Capital and Mobility programme.

## References

[1] A. Harter and A. Hopper. A distributed location service for the active office. *IEEE Network*, 36(1):62-70, January 1994.

[2] G. Liu. Efficient mobility management for wireless mobile computing and communications. Licentiate's thesis, Telecommunications System Laboratory, Dept. of Teleinformatics, Royal Institute of Technology Stockholm, March 1995.

[3] J. Magee, N. Dulay, and J. Kramer. Regis: A constructive development environment for distributed programs. *Distributed Systems Engineering*, 5(1), September 1994.

[4] Mobile-IP Working Group of the Internet Engineering Task Force. IP Mobility support . IETF internet draft, July 1995.

[5] M. Rizzo, P. Linington, and I. Utting. Integration of location services in the open distributed office. Technical report, Computing Laboratory, University of Kent, Canterbury, 1994.

[6] M. Satyanarayanan. Distributed File Systems. In S. Mullender, editor, *Distributed Systems*. ACM Press Frontier Series. Addison-Wesley, second edition, 1993.

[7] J. Scourias. Overview of the global system for mobile communications. May 1995. Available from `http://ccnga.uwaterloo.ca/ jscouria`

[8] S. Sharma. Security for Active Badges. M. Sc. Thesis, Dept. of Computing, Imperial College, London, UK, September 1994.

[9] M. Spreitzer and M. Theimer. Scaleable, secure, mobile computing with location information. *Communications of the ACM*, 36(7), 1993.

[10] B. Schilit and M. Theimer. Disseminating active map information to mobile hosts. *IEEE Network*, pages 22-32, September 1994.

[11] M. Spreitzer and M. Theimer. Architectural considerations for scaleable, secure, mobile computing with location information. In *The 14$^{th}$ International Conference on Distributed Computing Systems*, pages 29—38, Poznan, Poland, June 1994. IEEE CS Press.

[12] K. Twidle. *Domain Services for Distributed Systems Management*. Ph.D. thesis, Imperial College, London, May 1993.

[13] N. Yialelis and M. Sloman. An authentication service supporting domain based access control policies. Technical Report DoC 95/13, Dept. of Computing, Imperial College, September 1995.

[14] P. Dias. Location information in computer systems: Applications and implications. M.Eng. thesis, Dept. of Computing, Imperial College, London, June 1995.

[15] M. Weiser. Some computer science issues in ubiquitous computing. *Communications of the ACM*, 36(7), 1993.

[16] D. Wells et al. *Guide to GPS Positioning*. Canadian GPS Associates, 1986.

[17] T. Ritchey. *Java!* New Riders Publishing, Indianapolis, Indiana, 1995.

[18] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *IEEE Workshop on Mobile Computing Systems and Applications*, Santa Cruz, December 1994.

[19] S. Shekkar and D. Liu. Genesis and Advanced Traveller Information Systems (ATIS): Killer applications for mobile computing? *MOBIDATA*, 1(1), November 1994. Available from `http://rags.rutgers.edu/journal/cover.html`