Location-aware mobile applications based on directory services

Henning Maass

Philips Research Laboratories, 52066 Aachen, Germany

Location-aware applications are becoming increasingly attractive due to the widespread dissemination of wireless networks and the emergence of small and cheap locating technologies. We developed a location information server that simplifies and speeds up the development of these applications by offering a set of generic location retrieval and notification services to the application. The data model and the access protocols of these services are based on the X.500 directory service and the lightweight directory access protocol LDAP since these are becoming the standard attribute-value-pair retrieval mechanisms for Internet and Intranet environments. This approach establishes a smooth migration path from conventional to location-aware applications. The paper presents the location information server concepts, defines its directory data model and access services, and discusses the implementation options of the location information server.

1. Introduction

With the increasing popularity of mobile communications and mobile computing, the demand for location-aware and adaptive applications grows. Location-aware applications exploit knowledge about the physical location of real-world objects such as mobile persons and devices, to adapt their functional behaviour and their appearance towards the user. The economic deployment of location-aware applications will very soon become possible due to the progress in miniaturisation and the resulting cost reduction of locating technologies such as GPS [5]. Furthermore, the next generation wireless multimedia networks will utilise such high radio frequencies or even infrared links that the radio cells will be limited to the size of a room. This will allow to retrieve location information from the wireless network mobility management functions without additional costs.

To enable the fast and efficient development of location-aware and adaptive applications, we are developing a sophisticated location information server (LIS) based on directory data models and services that is presented in this paper[1]. In addition to earlier work on location-aware applications ([11,20,23]) our approach treats location-awareness not as an isolated class of applications but conceptually integrates them into a generalised framework for mobile multimedia communication services. Consequently, the location information server is integrated part of a software platform for mobile multimedia applications[2] and interacts with the other platform support functions and mobility management services. The platform shields the applications from the distribution and heterogeneity of the underlying communication networks and locating infrastructures and offers a set of sophisticated support functions and high-level APIs to the application programmers. In multisite corporate networks the platform can be distributed over several mobility management domains, allowing the APIs and support functions to co-operate via directory service based signalling protocols as depicted in figure 1.

The support functions of the mobile application platform offer three different location abstraction levels:

- Location-transparent:
  This abstraction level completely hides the effects of mobility to applications and users. Network services and resources can be transparently accessed by means of a *resource and service broker* function that maps the application's service type requests on adequate service provider instances. Additionally an application-defined quality-of-service (QoS) for the underlying network connections is sustained through a *QoS manager* function. Applications operating on this abstraction level are thereby given higher priorities than others in case of conflicting resource requirements or wireless network congestion.
- Location-tolerant:
  This abstraction-level allows applications and users to tolerate those effects of mobility that can not be hidden by the platform. Reasons can be congestion of radio cells, degradation of radio link qualities or change of terminals in case of user mobility. The trader function allows the application to perform a service and service type re-negotiation to achieve a graceful service degradation instead of dumb service termination. A *profile handler* function allows to retrieve user and terminal characteristics to perform application adaptation according to the type of terminal currently being used.
- Location-aware:
  This abstraction-level allows applications and users to be aware of their mobility and the absolute and relative physical positions of real-world objects. Applications can exploit this information for customising their functionality and users can benefit from this information for

[1] This paper is a revised and extended version of a conference paper presented at MobiCom '97 [14].

[2] The work this paper is based on is being supported by the German Federal Minister of Education, Science, Research and Technology (BMBF) as part of the project line ATMmobil.

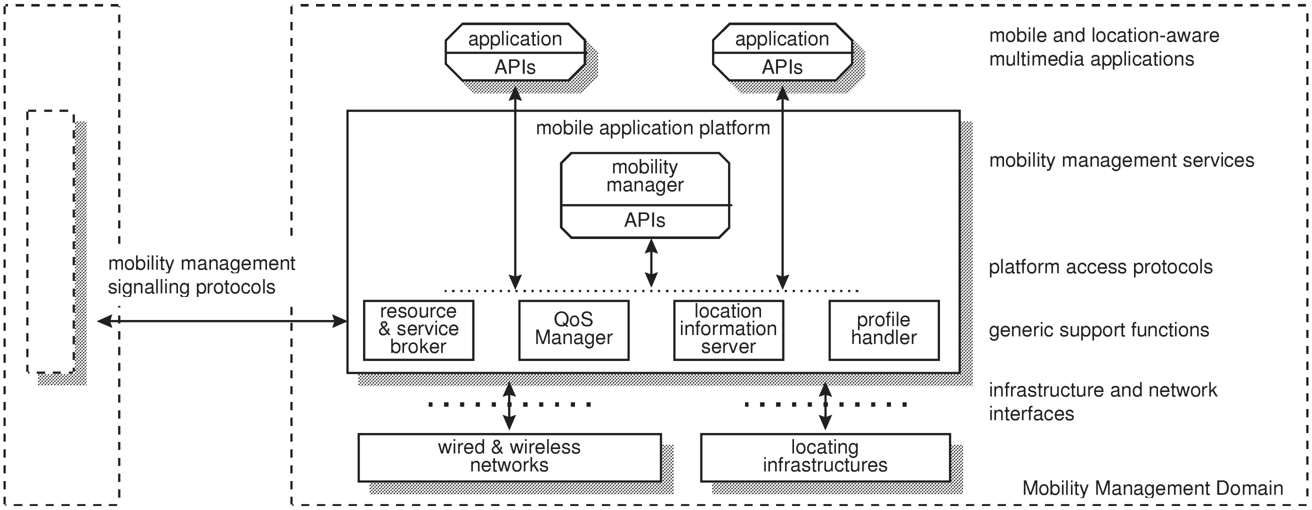

Figure 1. Platform for mobile and location-aware multimedia applications.

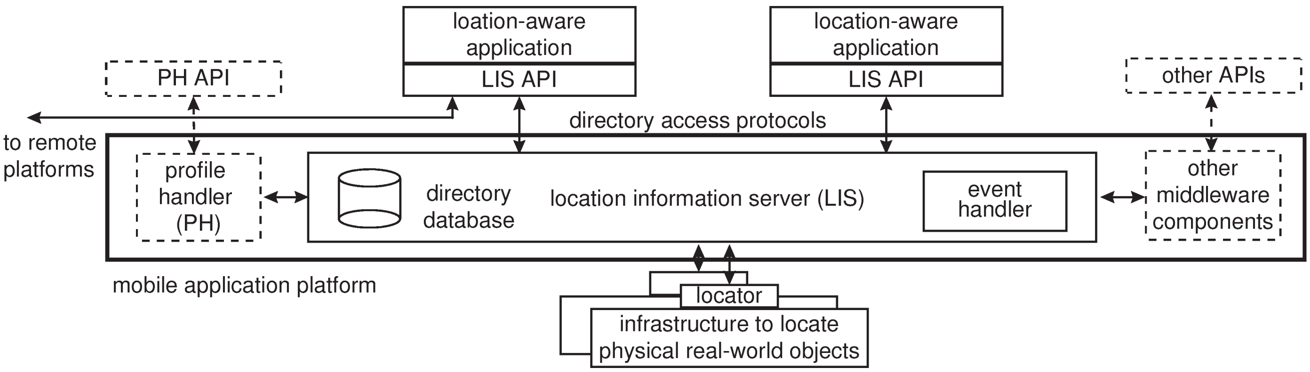

Figure 2. Location information server in mobile application platform.

navigation purposes. This abstraction level is realised by the *location information server* function that allows applications to query location information and to be notified about the occurrence of predefined location-related events.

For remote invocations of the location information server – used when application and platform run on different machines – the APIs communicate with the platform through a suite of mobile application platform access protocols based on the widely accepted directory access protocol DAP [10] and lightweight directory access protocol LDAP [17]. This has the key advantage that the application programmers can employ off-the-shelf DAP and LDAP APIs that are widely available on numerous platforms [7,18,24]. Therefore no proprietary communication protocol stacks have to be developed, neither for the application, nor for the location information server. Additionally, since many networked applications need the directory access protocols anyway, the additional effort for the location-aware features of the application is minimised and a smooth migration path from conventional to location-aware applications is established.

The location information server acquires information about the – absolute or relative – physical location of real-world objects in which an application is interested. The LIS hides from the application which locating technology is actually used by presenting it a generic locating model. The LIS has map and relationship knowledge to translate the low-level position information from the locating infrastructures into location information having a meaningful abstraction level for the application. The application can query the LIS about current locations of objects (LIS directory database) or can request to be notified when certain location-related conditions between objects and locations are fulfilled (LIS event handler). Other support functions of the mobile application platform can internally access the LIS functionality too and vice versa, see figure 2.

2. Motivation for X.500-based approach

Usually, the location information server and the location-aware applications will run on different machines. The applications may, e.g., run on mobile laptops, cordless communication terminals, and stationary PCs, whereas the LIS will usually run on a network server being interfaced to the physical locating infrastructure. Therefore the communication protocol between server and application must be able to bridge the gap between computing platforms of many different types which makes the choice of standardised protocols preferable due to their widespread availability.

The X.500 directory access protocols have recently been accepted as a standard means for accessing attribute-value-pair information from several types of servers, especially in the area of the Inter- and Intranets, the Intelligent Network, and modern mobile communication networks:

- The lightweight directory access protocol LDAP has been accepted as de-facto standard Internet directory access protocol [12]. LDAP has helped to promote X.500 directory services in the Internet since it is based on TCP/IP and can be implemented much easier than the original OSI transport service-based DAP protocol.
- Manufacturers of networking technology have begun to integrate LDAP into their products. Netscape's Internet browser uses LDAP to access directories, in the near future all their client and server products will use LDAP for any attribute-value-pair information access [16]. Even the clients will eventually contain low-end LDAP servers to replicate information. Novell announced that they would integrate LDAP access into their Novell Directory Services NDS.
- New features are currently being added to the directory standards to better cope with dynamic information, e.g., by controlling the lifetime of directory entries and supporting automatic deletion of entries [25]. This will turn directories from the providers of static information they are today into information brokers between all kinds of servers and applications.
- Microsoft's Internet conferencing tool NetMeeting [15] uses LDAP to exchange user profile information about users wishing to join conferences. It can be expected that other Internet applications will follow this approach too. Networked applications in the Inter/Intranet environment will therefore most probably have LDAP already built-in soon.
- The directory access protocol DAP has been agreed as interface between service control functions (SCF) and service data(-base) functions (SDF) in the Intelligent Network (IN) standards [2]. These standards are relevant for applications that want to perform service control over telecommunication networks, e.g., PBXs.
- X.500-compatible services are used to store and access mobility management data for public and private mobile communication networks, e.g., in the third generation mobile networks UMTS [1], in the Japanese Personal Handyphone System PHS [21], and in private telecommunication networks [13]. These standards are relevant for applications that want to extend the mobility management functionality of mobile networks with location-aware features.

For these reasons it seemed wise to base the location information server protocols on the widely accepted LDAP protocol with the option to alternatively use the DAP. The location information server therefore offers the following service model to location-aware applications:

- All LIS data that shall be made available to the application is represented by the location information server in an X.500-conformant directory information tree (DIT).
- Applications access the location information server through the standardised directory access protocols LDAP or DAP for local and for remote operations.
- Application programmers employ standardised and widely available APIs for DAP [24] and LDAP [18] to communicate with the LIS.
- Other support functions of the mobile application platform and remote LIS servers in other parts of the network can access the LIS functionality through the Directory System Protocol DSP or through LDAP.
- To make use of the LIS, the application and other support functions must have knowledge about the directory schema of the LIS, and has to know which directory services implement the desired locating functionality, in which sequence they have to be invoked, and which parameters have to be conveyed to the LIS.
- The LIS has to implement functionality normally offered by an X.500 directory system agent (DSA) to implement the required parts of the X.500 service model and protocols. Additionally it has to interface to the physical locating infrastructures and has to process the data received from them to represent it in the DIT.

Whether this service model should be implemented from scratch into the location information server or whether existing directory servers should be extended with the LIS functionality will be discussed in section 8.

3. LIS concepts and requirements

The location information server shall be able to hide the actually used locating technology from the application, therefore a generic locating model has been defined (see also figure 3): The location information server locates **objects** representing either **persons** or **resources** inside **areas**. To make objects automatically locatable, they have a **tag** attached to them that identifies and localises its wearer. Resources serving as a tag can be for example badges, cordless phones, PDAs or laptops. The **relations** between objects and tags are maintained in the LIS. A **locator** forms the interface between the locating infrastructure and the LIS. The following locating principles can be used for tags:

- The tag is located relative to areas by means of a sensor installed in that area. This information is then collected by the locator and published to the LIS. The LIS then uses a map to translate sensor identifiers into area identifiers.
- The tag can determine its absolute geographical position and publishes this information through the locator

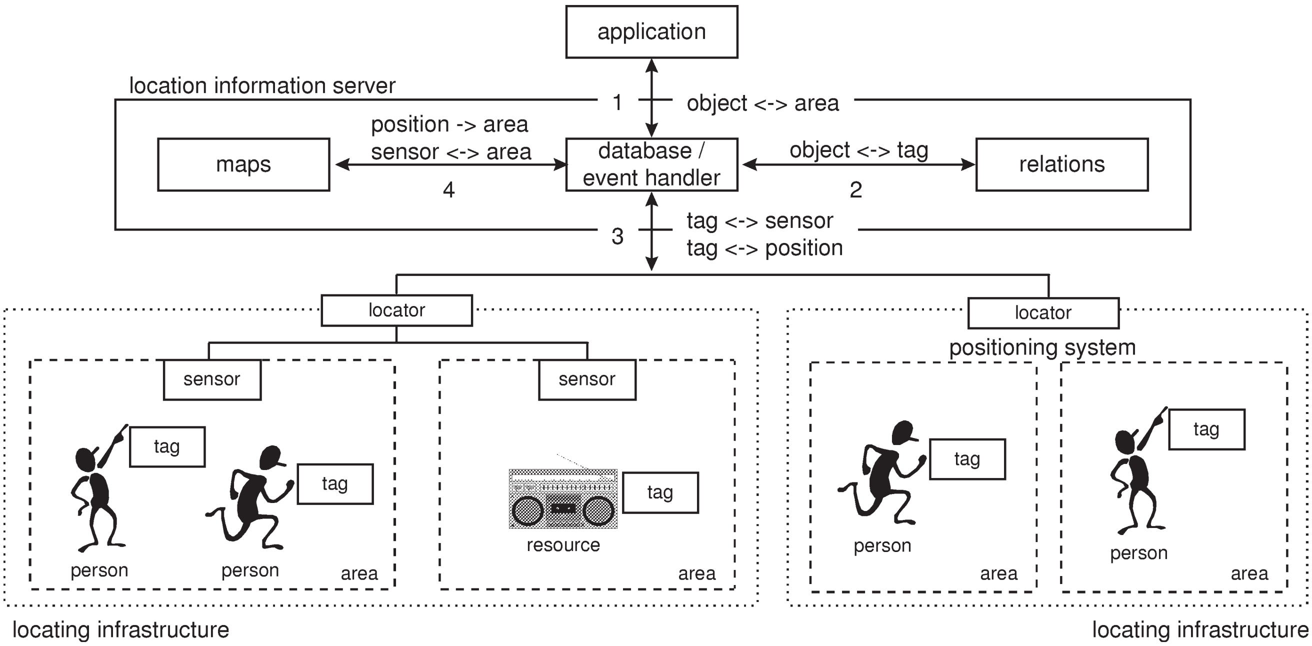

Figure 3. Generic locating model.

to the LIS. By means of a map the LIS translates the geographical position into an area-identifier.

For navigation applications the LIS may additionally provide **graphical map** information in human-readable format and may calculate **shortest paths** between areas. Additionally the LIS may find objects that are **nearest** to a specified area and fulfil certain conditions. All entities of the locating system have a unique identifier, i.e., **tag-ID**, **sensor-ID**, **object-ID**, or **area-ID**. The network site in which a mobile object is located is identified by a **mobility management domain identifier mmd-ID**.

3.1. Locating technologies

This section briefly describes locating technologies that can be used to implement the locating infrastructures for real-world objects:

- Dedicated locating infrastructure:
  The tag is a special locatable device. The areas of the locating infrastructure are equipped with sensors that can detect the presence of the device. This approach can for example be realised based on an Active Badge System [4,22]. The tags are credit card-sized badges that communicate with the sensors via infrared light. This system is limited to indoor usage and the locating area normally equals a room. Another example for infrared-based tags are the palm-sized ParcTabs [23]. Radio-based tag detection techniques are also available, e.g., in contactless smart cards operating at distances of up to one meter.
- Wireless network infrastucture:
  The tag is the terminal of a wireless communication network. In this approach, the radio cells of the wireless communication network serve as sensors and produce information where the mobile terminal is located. The information can be retrieved from the network's mobility management function. This approach is applicable in indoor and outdoor environments and is appealing since no additional infrastructure has to be deployed in environments where mobile devices are using wireless communications anyway.
- Absolute positioning techniques:
  The tag contains a GPS receiver [5] which is used to calculate its absolute position. It then uses a geographical map stored either in the tag itself or in the locating server to determine the area the object is in. This approach is currently limited to outdoor environments but GPS relay senders for indoor usage are under development.

3.2. Application requirements on the location information server

This section discusses the functionality that an application requires from the locating infrastructure and the location information server. In these examples, an area usually corresponds to a locating granularity being obvious to humans, e.g., a room or a floor and the locatable objects are persons or pieces of equipment.

3.2.1. Location retrieval services

An application may be interested in the whereabouts of objects that carry a tag to automatically determine their physical location but it also may be interested in the location of static objects or mobile objects having no tag. Therefore, the location information server must also be able to handle locatable objects without tags by storing their location internally and offering retrieval and location update services. Typical location retrieval requirements of location-aware applications are for example:

- In which area is person A (and when was the last sighting)?
- How many or which persons are present in area B?
- Where is equipment C (e.g., an apparatus in a hospital)?

3.2.2. Location-dependent object and area selection services

Location-aware applications sometimes do not want to know the location of a certain object instance but want to know which persons or resources of a certain type (e.g., printer, doctor) are present in or nearest to a given area:

- Which object of type D is present in area B?
- Which one is the nearest object of type D (relative to my own location) and where it is located?
- Which (nearest) area fulfils certain conditions (e.g., which one is the nearest unoccupied meeting room)?

3.2.3. Event services

To avoid repeatedly polling the LIS, some location-aware applications want to be informed by the location information server when an application-defined event occurs. This functionality is, e.g., needed to start a function on a mobile device as soon as it's wearer enters a certain area or to inform a user that a colleague he wants to visit is now present in his office. Examples of events are:

- Inform me when person A enters (or leaves) area B!
- Inform me when area B is empty!
- Inform me next time person A meets person B anywhere!

The event services of the LIS shall consist of two parts: They shall consist of a registration part in which an application expresses its interest in a certain event. When the specified event finally occurs, the application shall receive a notification denoting the occurred event. Important for applications on mobile terminals is that notifications do not become lost in situations where the application is temporarily disconnected from the LIS.

3.2.4. Map retrieval services

Location-aware systems often offer a navigation application to help human users in finding other persons, resources or areas. For this purpose, the LIS shall offer map information and navigation instructions to its applications:

- Provide me with a human-readable map of area A!
- Provide me with the shortest path between area A and B!

3.2.5. Identifier mapping services

In the above mentioned examples, applications usually require information about objects and their actual areas and not about tags and their sensors respectively positions. For the internal operation of the LIS, for other support functions of the mobile application platform, and for location-aware applications operating on the tag-level, bi-directional mapping services between the identifiers of areas, sensors, and positions are required. These mappings are determined upon installation of the locating infrastructure and are therefore not only required by the LIS and the applications, but also by the system administrator.

3.2.6. Relationship services

Location-aware applications operating at tag-level instead of object-level or applications that want to communicate with an object through its tag are often interested in the relation between objects and tags:

- Which object is associated with tag A?
- Which (if any) tag does object B possess?

Some applications want to change these relations, e.g., when persons change their terminal serving as a tag:

- Associate tag A with object B!

How all these application requirements are realised by means of directory services offered by the LIS to the application is defined in the following chapters. First the data model and the required directory schema are defined, then the necessary LIS directory service invocations the application has to perform are described.

4. Introduction to X.500 and LDAP

The X.500 set of recommendations [9] standardise a distributed directory service as OSI layer seven service and protocol. The directory service is offered by the Directory System Agent DSA to the Directory User Agent DUA, representing an application or human directory user. The DSA offers the services listed in table 1 to access the directory information via the Directory Access Protocol DAP.

The content of the directory is not held by a single DSA but may be distributed over a set of co-operating DSAs to enable the establishment of a world-wide global directory service. For this purpose, a DSA interacts with other DSAs through the Directory System Protocol DSP to hide the physical data distribution from the directory user.

Table 1
X.500 DAP directory services.

Read	read a single entry from the directory
Compare	compare an attribute value with a given value
Search	search for one or more entries in the directory
List	list the subordinate entries of a directory entry
Add	add an entry to the directory
Delete	delete an entry from the directory
Modify	modify the content of an entry
ModifyRDN	modify the last component of the entry's name (RDN)
Abandon	cancel an outstanding directory access operation

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.