Location Information Management

Hani Naguib, George Coulouris
{han21,gfc22}@cam.ac.uk

Laboratory for Communications Engineering, Cambridge University
http://www-lce.eng.cam.ac.uk/qosdream/

Abstract. We describe an approach to the management of location events deriving from a variety of sensors and sensor technologies. The approach uses a region-based spatial algorithm to integrate low-level location information, producing events that approximate closely to the needs of application programs. Preliminary experience indicates that the approach yields a very worthwhile reduction in the event traffic that must be handled by applications.

Introduction

A wide range of location sensing technologies already exist, and more are under development [WR00]. Some, such as the Active Badge system [HH94], provide information about the presence of a user in a region or a room. Other location technologies, such as the Active Bat system [HHSW+99] are much more precise, providing information that is accurate to a few centimetres. All location technologies generate sensor events in response to changes in the locations of locatable objects. The rate at which they are generated is high, especially for high precision location technologies. Only a fraction of the sensor events are of interest to any specific application. The multiplicity of this location information determines the need for a management system. A mechanism is required that will enable applications to see those events that are relevant to its responsibility while filtering out those that are not. Furthermore a uniform way of making this information available in a location-technology independent fashion is needed because of the increasingly varied technologies available. Compounding this requirement will be the usage of various location-technologies together.

In this short paper we outline our approach to meeting these requirements. Section 2 describes our Location Service which tackles these issues. Section 3 details some preliminary results. We conclude in section 4.

Architecture

The research into location information management presented in this paper is part of the QoSDREAM project [QDream01]. QoSDREAM aims to provide a middleware platform for developing distributed multimedia applications. Of relevance to this paper is its approach to handling context information and how this information is made available to the rest of the system.

Figure 1 shows the major constituents of the QoSDREAM architecture. It consists of four major parts. At the lowest level we have the **Location Service** It is in charge of gathering location information from specific location-technologies and presenting it to the rest of the system in a location-technology independent fashion. The Location Service's functionality also include managing and filtering the potentially large amount of location information that can be generated. This information is made available to the rest of the system and applications through an **Event Messaging System**. The Event Messaging System distributes events to interested parties and provides a number of filtering services. **The Distributed Multimedia Service** [MNCK99] (not presented in this paper) allows applications to construct and manage, distributed multimedia components. The Distributed Object Database Management Service provides a means of sharing system wide properties such as information about known physical objects. The information held by the database is largely 'static' (i.e. it is not modified very often), although some location-specific data may also be stored with it.

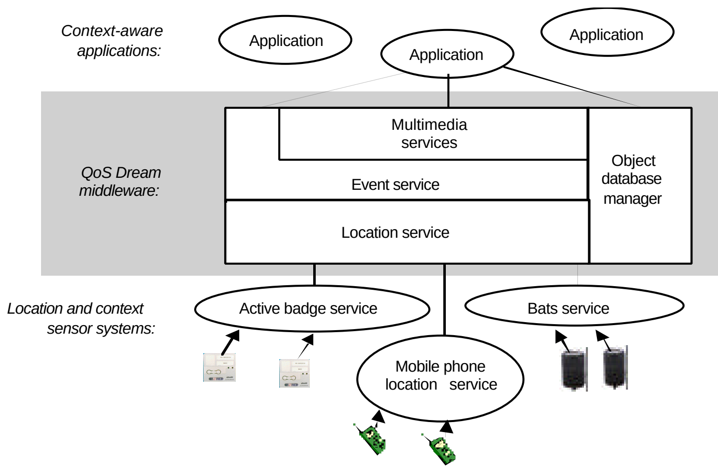

Fig. 1. Overview of QoSDREAM's Architecture

Location Information

Our Location Service provides location-dependent information. The Location Service is a collection of objects that retrieve location information from existing location sensors and make this information available to QoSDREAM applications. The type of location information available is highly dependent on the location-technologies being used. QoSDREAM's Location Service interprets this information and presents it in a location-technology independent format. This is achieved by representing location information in terms of regions and the interactions between regions.

Modelling Location Information

Within QoSDREAM, applications are presented with a simplified model of the real physical world. This model contains the following abstractions:

? **Locators**, which represent objects whose location can be determined by a given location-technology. Active badges and Bats are examples of Locators.

? **Locatables**, these represent objects whose locations need to be tracked. Examples include people and equipment. Locatables must have at least one associated locator, so that their location can be inferred from its locator.

? **Locatable Regions**. Each locatable will in general have one or more regions. These regions define various location specific characteristics of Locatables. For instance in the case of a person, they may include in addition to the person's location, their visual and audible fields.

The main reason for using regions as a way of presenting location information is the varying degree of precision of location technologies. Active Badges will only place a badge within a room, Active Bats provide more fine-grained information. Expressing location information as regions allows the incorporation of a wide range of location technologies. Furthermore the use of regions also aids in the management of location information particularly with regards to filtering this information in order to reduce location-related traffic. This is discussed in the following subsection.

Location related information is presented to applications as interactions between the various regions in the model. In particular the change in overlap status between regions (For example a person walks into a room, his 'location region' overlaps with the room's region).

Management of Location Information

Our Location Service performs management of location information. Figure 2 depicts the various abstractions that form this service.

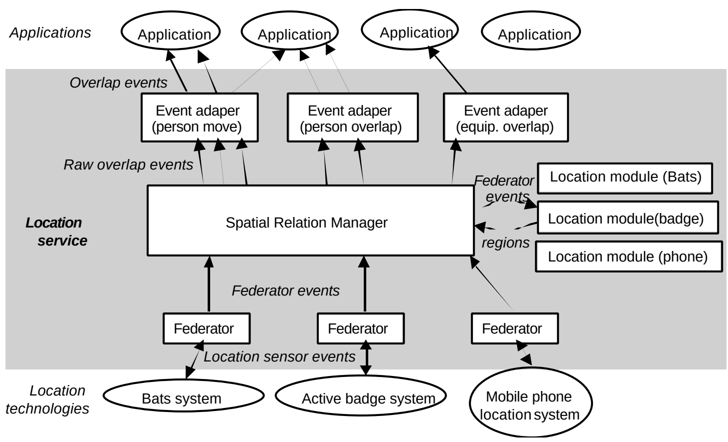

Fig. 2. Location Service Architecture

At the lowest level we have **Federator** objects, these interface with specific location technologies and can detect location information. In our lab we have an 'Active Badge Federator' which is capable of interpreting the data being generated by the Active Badge System. The information gathered by Federators is exported through light-weight events called **Federator Events**. These contain generic information such as the type and id of the federator as well as federator-specific data.

These events can be monitored by applications that require very detailed location information. In general applications are only interested in specific location-related information and so can use other types of events (described below) to receive only events that it is interested in. Federator events are also sent to our **Spatial Relations Manager** (SRM). The SRM's task is to convert information gathered by Federators into overlapping events between the regions in our world model. Before the SRM can reason about the regions and their interactions, **Location Modules** translate Federator events into regions. For example a Location Module for an Active Badge System, would generate a location region for the given locator in the Badge Federator Event. The shape of this region would be the same as that of the room where the locator was detected. This can be derived from the sensorID field in the Federator Event.

Location Modules are Federator-type-specific, and can be loaded from the database by the SRM. The SRM uses the federatorType field of Federator Events to determine the required Location Module.

Once the SRM has obtained updated regions from Location Modules it analyses them looking for changes in overlaps between regions. If those are found then the SRM produces what we call **Raw Overlap Events**. These contain the type of overlap between two regions as well as their previous relationship.

Applications can register to receive these events, but again these events are too general for most applications and cannot be filtered. Instead application register to receive specialised overlap events produced by **EventAdaptors**. These types of events can be filtered and relate to higher level abstractions, such as people and equipment.

EventAdaptors are objects that receive Raw Overlap Events generated by the SRM and can filter and transform these into events that are of interest to applications. The events generated by EventAdaptors are called **OverlapEvents**, and are transmitted by QoSDREAM's Message Service. New EventAdaptors can be easily created and dynamically incorporated into QoSDREAM. By default we provide three types of EventAdaptors:

? **PersonMovement Adaptor**: This generates overlap events when a person's location region overlaps with a geographical location. This event contains a personID and a LocationID fields.

? **PersonOverlap Adapter**: This generates overlap events when a person's location region overlaps with that of another. This event contains a person1ID and person2ID fields.

? **EquipmentOverlap Adapter**: This generates overlap events when a person's location region overlaps with that of an equipment. This event contains a personID as well as a equipmentID fields.

Applications are free to choose what events they wish to receive. They can further filter OverlapEvents on the value of the fields in the event. For example, an application interested in the arrival at work of person George can register to receive PersonMovement events whose personID field equals George's id.

To aid with the scalability the representation of the physical world is divided into zones. These zones can be used by the SRM to cut down on the number of regions it must analyse when looking for overlap events. Zoning also allows for a federation of SRMs to be set-up, each of which is responsible for specific zones (as long as those don't overlap). For instance floors in a building might each be handled by a separate SRM.

Event Messaging Service

The QoSDREAM framework provides an event abstraction that enables applications to register for events by event type. Further selection is performed by event filters which the application can instantiate. Options for event queuing, communication, fault tolerance, event persistence, reliability and security will be offered in a future version of the DREAM framework. The event handling abstraction outlined here is compatible with most general-purpose event handling systems including Corba Events [CDK01], Elvin [ASBB+99, FMK+99] and Herald [BBMS98, Spi00].

QoSDREAM uses events primarily for delivery of contextual information to applications. The messaging service allows application objects to send and receive events and is also used by the Location Service to deliver events generated by EventAdaptors.

The Messaging Service facilitates the management of location information in a number of ways. It allows clients to request the delivery of events without having to know anything about the sources of these events. Similarly event sources do not require any knowledge about their clients. The messaging service also allows applications to specify filters on the events that are sent to it. This coupled with the location service greatly reduces location-related traffic. Without the location service and messaging service, applications would need to monitor all location information generated by the various underlying location technologies.

The Messaging Service is itself independent of the event delivery system used to transfer events from event sources to event clients. Our current implementation of the Messaging Service uses an implementation of the Cambridge Event Architecture called Herald [SPI00] as the event delivery system. Other event delivery systems can be easily added to our Messaging Service by the implementation of a single interface.

Experience

We have developed a pilot application for members of our laboratory to use as a communications aid, allowing them to locate one another, establish and participate in audio/visual conferences. It exploits location information provided by the Active Badge System deployed in the laboratory. Table 1 gives an indication of the typical location-specific traffic found in our lab. This traffic was measured over a five minute interval and at the time there were six members of the lab wearing badges.

The ActiveLab application registers interest only in PersonMovement events (in order to update its display). Thus in this application, the QoSDREAM location architecture reduces the event traffic that the application process is required to handle by a factor of approximately 17.

Table 1. Typical location related traffic

Badge Events (Federator Events)	671 These are the events generated by the ActiveBadge Federator. They are generated whenever a badge is picked up by a sensor. Many of the rooms in the lab have more than one sensor and therefore a badge may be picked up by more than one sensor. Most of these events are as a result of a person's badge being detected, since equipment badges emit their signal less often, than badges worn by people.
Raw Overlap Events	204 These are the events generated by the SRM.
PersonMovement Events	12 These events are generated by the PersonMovement Adaptor in response to people moving in and out of rooms.
PersonOverlap Events	40 These are generated by the PersonOverlap Adaptor. In this example those were caused by people moving into rooms where other people were present.
EquipmentOverlap Events	48 These are generated by the EquipmentOverlap Adaptor. Those were caused by people moving into different rooms and their region overlapping with those of the equipment in the new room.

Conclusion

We have described the approach employed in the QoSDREAM project to the management of location events derived from a variety of sources and technologies. Location events are integrated through the generation of overlap events that are constructed by geometric analysis according to the requirements of applications, reducing the incoming event traffic to proportions that are manageable by applications.

QoSDREAM is a software framework that supports the construction of context-aware multimedia applications. For details on other aspects of the QoSDREAM framwork see [QDream01].

Acknowledgements

The QoSDREAM project is funded by the UK EPSRC under its Multimedia and Network Applications research programme. The following people have contributed to the development of the QoSDREAM framework: Scott Mitchell, John Bates, Mark Spiteri, Martin Harper, Sarah Gunn, Eli Katsiri, Kasim Rehman.

References

[ASBB+99] David Arnold, Bill Segall, Julian Boot, Andy Bond, Melfyn Lloyd, Simon Kaplan, "Discourse with Disposable Computers: How and why you will talk to your tomatoes", Usenix Workshop on Embedded Systems (ES99), Cambrdige Mass, March 1999. http://www.usenix.org/publications/library/proceedings/es99/full_papers/arnold/arnold_html/

[BBMS98] J. Bates, J. Bacon, K. Moody and M.D. Spiteri, "Using Events for the Scalable Federation of Heterogeneous Components", Proc. ACM SIGOPS EW'98, Sintra, Portugal, Sep 1998

[CDK01] G. Coulouris, J. Dollimore and T. Kindberg, Distributed Systems: Concepts and Design, Edition 3, Addison-Wesley 2001.

[FMK+99] G. Fitzpatrick, T. Mansfield, S. Kaplan, D. Arnold, T. Phelps, and B. Segall, "Instrumenting and augmenting the workaday world with a generic notification service called Elvin", Proc. ECSCW'99, Copenhagen, Denmark, Sep 1999

[HH94] A. Harter and A. Hopper, A Distributed Location System for the Active Office. IEEE Network, Vol. 8, No. 1, January 1994.

[HHSW+99] A. Harter, A. Hopper, P. Steggles, A. Ward and P.Webster, The Anatomy of a Context-Aware Application. In Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom '99), Seattle, Washington, USA, August 15 - 20 1999 1999. ftp://ftp.uk.research.att.com/pub/docs/att/tr.1999.7.pdf

[Mit00] R.S. Mitchell, Dynamic Configuration of Distributed Multimedia Components. Ph.D. Thesis, University of London, August 2000. http://www-lce.eng.cam.ac.uk/qosdream/publications/

[MNCK99] Scott Mitchell, Hani Naguib, George Coulouris and Tim Kindberg, A QoS Support Framework for Dynamically Reconfigurable Multimedia Applications. In Lea Kutvonen, Hartmut König and Martti Tienari (eds), Distributed Applications and Interoperable Systems II, pp 17-30. Kluwer Academic Publishers, Boston, 1999. Also in Proc. DAIS 99. http://www-lce.eng.cam.ac.uk/QoSDREAM/publications/

[MSBC00] Scott Mitchell, Mark D. Spiteri, John Bates and George Coulouris, "Context-Aware Multimedia Computing in the Intelligent Hospital", In Proc. SIGOPS EW2000, the Ninth ACM SIGOPS European Workshop, Kolding, Denmark, September 2000. http://www-lce.eng.cam.ac.uk/QoSDREAM/publications/

[QDream01] QoSDREAM's project web page. http://www- lce.eng.cam.ac.uk/qosdream/

[Spi00] M.D. Spiteri, An Architecture for the Notification, Storage and Retrieval of Events, Ph.D. Thesis, University of Cambridge, Jan 2000. http://www-lce.eng.cam.ac.uk/QoSDREAM/publications/

[WR00] R. Want, D. M. Russell. "Ubiquitous Electronic Tagging". Distributed Systems Online, IEEE 2000. http://www.computer.org/dsonline/articles/ds2wan.htm.