The Location Service: A framework for handling multiple location sensing technologies

Gregory D. Abowd, Agathe Battestini & Thomas O'Connell

College of Computing & GVU Center
Georgia Institute of Technology
Atlanta, Georgia 30332-0280, USA
{abowd, abattest, thomas}@cc.gatech.edu

Abstract. Location information remains the single most important piece of context used in ubicomp applications. Many different location sensing technologies are being created and a variety of applications explored. Relatively little attention is paid to the development of software infrastructure to facilitate location-aware programming. There are three goals for such an infrastructure. First, we want to make available to an application designer a reliable location service that does not suffer from the inadequacies of any single location sensing technology. Second, we want to shield the application programmer from the details of collecting and merging the location information from a variety of sources. Third, we want to provide the application programmer reusable and extensible techniques that allow for the interpretation of location information in application-relevant ways. We present a software framework, the Location Service that attempts to meet all of these goals. We describe the structure and rationale that underlies the Location Service and demonstrate its use within the Aware Home Research Initiative at Georgia Tech.

1 Introduction

Nobody questions the value of incorporating context into ubicomp application development, particularly when the context is the location of individuals.. As we examine the literature on location-aware computing, we see three major emphases:

- deployment of specific location sensing **technologies** (see Hightower & Borriello [8] for a recent review)
- demonstration of compelling location-aware **applications**; and
- development of software frameworks to ease application **construction** using location [9].

This paper presents a construction framework, the Location Service, for handling location information about tracked entities. Our goal in creating the Location Service is to provide a uniform, geometric-based way to handle a wide variety of location technologies for tracking interesting entities while simultaneously providing a simple and extensible technique for application developers to access location information in a form most suitable for their needs. The framework we present divides the problem into three specific activities:

- **acquisition** of location data from any of a number of positioning technologies;
- **collection** of location by named entities; and
- **monitoring** of location data through a straightforward and extensible query and translation mechanism.

Our previous work with the Context Toolkit was motivated by this same the separation of concerns [5,6]. However, our experience with the Context Toolkit, and specifically our experience in seeing how others tended to use it, shows that the separation was not complete enough, resulting in too many tedious programming requirements. In this paper, we restrict our attention to location-aware computing so that we can demonstrate the value of this three-layer approach to application construction.

Overview

We begin with a brief overview of other location-aware software construction frameworks in order to clarify the contribution of this work. We then provide an overview of the software framework that separates the activities of acquisition, collection and application-specific monitoring. Each of these activities is then described in detail, emphasizing the specific use of location within the Aware Home Research Initiative at Georgia Tech [2]. We conclude with a description of some applications developed with the aid of the Location Service.

2 Related Work

The initial seminal work on location-aware computing was done at Xerox PARC as part of the PARCTab project [cite ParcTab TR]. The considerable application exploration done with the PARCTab units was a result of up-front effort by the research team to produce a clean API for accessing location information from the IR-based location sensing technology. This API was essentially a query interface to access location information for individuals and rooms. There was no attempt in this work to integrate other sources of location data.

There are two more recent efforts that directly address the question of software infrastructure to support location-aware computing with multiple location sensing technologies. The Nibble system, provided a probabilistic framework for handling positioning data from a wireless LAN [3]. This work is based on the MUSE sensor fusion framework that relies on Bayes network for relating similar pieces of context [4]. While this work presents a more mathematically rigorous approach to the problem of sensor fusion as it applies to location, it is addressing a different problem than the work presented here. Nibble, as presented in that paper, assumes that all location data is coming from the same source, that is, sensor readings of signal strengths for wireless access points. The focus in this paper is on a system that handles a wider variety of location data.

As part of the QOSDream project, Naguib and Coulouris developed a framework for managing location information from a variety of location sensing technologies [11]. The motivation for that work was to create an efficient flow of location update events to applications, so the focus of their infrastructure is on creating mechanisms to determine when significant location updates occur and informing applications of those updates. Though their work suggests that the framework incorporates multiple location sensing technologies, their only reported use of the system only used one sensing technology, Active Badges. Their location management assumes that important entities (tracked objects such as people as well as stationary landmarks such as rooms) are all described in terms of occupied regions within some shared coordinate space. Our approach attaches single points in space to tracked objects and provides a mechanism for translating this geometric representation to other geometric and symbolic forms.

We are obviously deeply influenced by our earlier work on the Context Toolkit [5,6]. One could view the development of a separate infrastructure to support location as an admission that the Context Toolkit is not an appropriate solution. After a couple years' experience using the Context Toolkit, we still contend that the basic separation of concerns and programming abstractions that it espouses are appropriate for many situations of context-aware programming, and this is evidenced by a number of internal and external applications developed using it. However, in practice, we did not see the implementation of the Context Toolkit encouraging programmers to design context-aware applications that respected the abstractions and separation of concerns. Our attempt at defining the Location Service is not meant to dismiss the Context Toolkit but to move toward an implementation of its ideas that goes further toward directing good application programming practices.

In the context of this related work, the contribution of this work can be seen as an explicit demonstration of the integration of multiple different location sensing technologies into a framework that minimizes an application developer's requirement to know about the sensing technology. We also provide a framework in which more complicated fusion algorithms, such as the Bayesian networks used in Nibble, can be used. Finally, we provide an extensible technique for the interpretation and filtering of location information to meet application-specific needs.

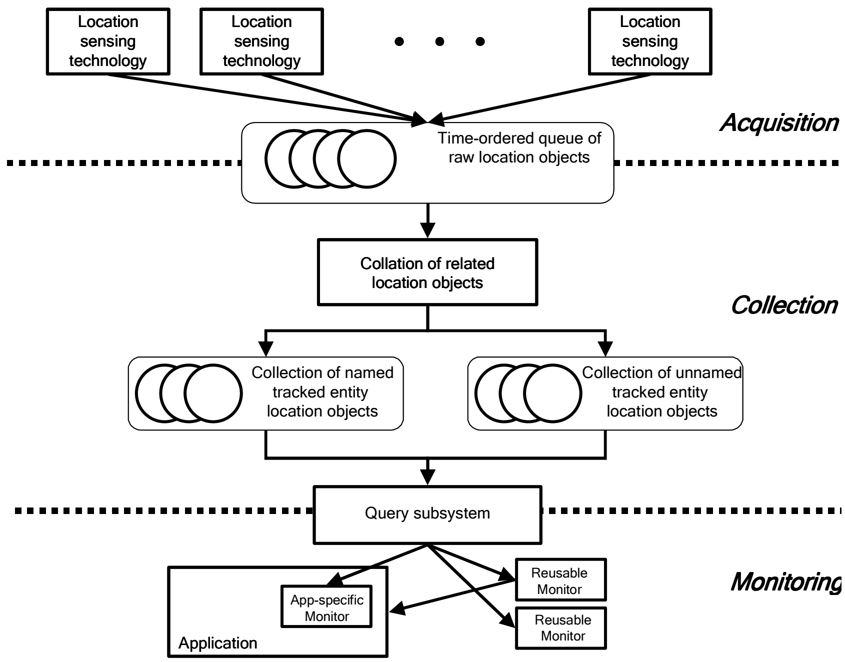

Figure 1: Overall architecture of the Location Service. Arrows indicate data flow

3 The architecture of the location service

Figure 1 shows a high-level view of the architecture of the Location Service. Any number of location technologies acquires location information. These technologies are augmented with a software wrapper to communicate a geometry-based (i.e., three-dimensional coordinates in some defined space) XML location message, similar in spirit to the widget abstraction of the Context Toolkit. The location messages are transformed into Java objects and held in a time-ordered queue. From there, a collation algorithm attempts to fuse separate location objects that refer to the same tracked entity. When a location object relates to a known (i.e., named) entity, then it is stored as the current location for that entity. A query subsystem provides a simple interface for applications to obtain location information for both identified and unidentified entities. Since location information is stored as domain-specific geometric representations, it is necessary to transform location to a form desirable for any given application. This interpretation is done by means of monitor classes, reusable definitions of spatially significant regions (e.g., rooms in a house) that act as filters to signal important location information for any given application.

In the next three sections, we will discuss in more detail the aspects of these three distinct phases of the Location Service. An overall assessment of this separation of concerns reveals several important properties of this service. First, the establishment of a well-defined location message insulates the rest of the Location Service from the details of the specific location sensing technologies. The overall service will work whether or not the constituent sensing technologies are delivering location objects. Second, the collation, or fusion, algorithm within the collection layer can be changed without impacting the sensing technologies or the location-aware applications. Third, the monitor classes are reusable and extensible, meaning simple location requirements don't have to be recreated by the application developer each time and complex location needs can be built up from simpler building blocks.

We describe a complete implementation in Java of the Location Service in the following three sections, followed by an example location-aware application. In our implementation, several critical design decisions were made, and each of these decisions are open to debate. It is important to understand that many of these decisions are open to wider debate. The real contribution of the Location Service is that as the larger community debates these design issues (e.g., the best representation of geometric location information, the best sensor fusion algorithm, etc.), the decisions can easily be integrated into the overall framework.

4 A variety of location sources

Any location service must take into account the use of multiple technologies for providing location information, both for the purpose of providing more accurate and reliable location information and for scaling to cover larger overall spaces, both indoor and outdoor. Different location sensing technologies vary according along a number of dimensions, as outlined by Hightower and Borriello [8]. We present those dimensions here and explain how all location-sensing technologies are viewed with respect to those dimensions.

Form of location data acquired: A location system can provide either geometric data on physical positioning in some well-defined space or it can provide more abstract symbolic representation of location (e.g., Gregory is in the living room). All of the location sensing technologies incorporated into the Location Service produce geometric positioning information.

Frame of reference: Is the location information provided in some absolute frame of reference that all sensing technologies share, or is the information provided relative to one of a set of reference frames? In the Location Service, all geometric location data is given with respect to a set of known reference frames. For example, the Aware Home building is its own reference frame with a known coordinate system; all positioning data within the home is provided with respect to that reference frame.

Local or environmental computation: Some location systems work by having the tracked entities compute their location (e.g., GPS), while others rely on computation in the environmental infrastructure to determine the location of tracked entities (e.g, RFID). From the perspective of the Location Service, an individual location sensing technology could fall into either category (locally computed or environmentally computed location). However, one of the advantages of locally computed location is that the location information can remain private to the tracked entity, and this feature is lost once a location object is communicated to the collector component of the Location Service.

Accuracy and precision: Different technologies produce location information at different levels of resolution and with different repeatability over time. While individual location sensing technologies will differ with respect to accuracy and precision, one of the main reasons for creating a merging service such as the Location Service is to produce an overall accuracy and precision that is better than that of any constituent technology.

Scale: It is important to know the physical range over which a particular sensing technology operates. Again, while individual location sensing technologies will each have limited range, the Location Service extends the effective range of the overall location system to a union of the individual ranges.

Recognition: Some technologies identify entities as well as track them. One of the strengths of the Location Service is that it tracks both identified and unidentified tracked objects. In addition, it attempts to attach identities to unidentified tracked objects through fusion of raw location objects from different sources. This fusion will be described later.

Cost: Most significant here are initial capital outlay costs for a location sensing technology as well as the incremental costs for extending either the number of tracked entities or increasing range of operation. One of the advantages of the Location Service is that a designer can now choose which of a variety of constituent location sensing technologies to invest in for a given space. This decision can now balance overall efficacy of the Location Service with initial capital outlay as well as incremental cost.

Limitations: Every location sensing technology has its advantages and disadvantages. It is precisely because there is no ideal single solution that we pursue integrating solutions such as the Location Service discussed in this paper. For example, in the demonstration system described here, the RFID mats have the advantage of being the most reliable sensing technology and are useful for marking transitions between rooms within a single floor of the Aware Home. However, they are not as reliable for determining entry and exit from a floor. As a balance, fingerprint detectors, which are also used to allow access into and out of a single floor, provide a good way to determine entry and exit from a floor.

4.1 Representing location

As we have already indicated, the Location Service assumes that raw positioning data is delivered as geometric data within one of a set of known reference frames. The raw positioning data object consists of:

- a four-tuple, (x,y,z,d), consisting of a 3-dimensional positional coordinate and a reference frame identifier, d, which is used to interpret the positional coordinate;

- an orientation value as a 3-tuple, if known;
- the identity of the entity at that location, if known;
- a timestamp for when the positioning data was acquired; and
- an indication of the location sensing technology that was the source of the data.

Not every sensing technology can provide all of this information. The Collector attempts to merge multiple raw location objects in order to associate a location value to a collection of named and unnamed entities. This results in a new location object for tracked entities that is stored within the Collector and made available via the query subsystem for applications to access and interpret.

4.2 Details on positioning systems

To exercise the framework of the Location Service, we have instantiated it with a variety of location sensing technologies. We describe them briefly here. The first two location sensing technologies existed prior to the development of the Location Service, and the latter two were developed afterwards to validate the utility of the framework.

4.2.1 The RFID floor mat system

For a while, we have been interested in creating natural ways to track people indoors. While badge technologies have been very popular in location-aware research, they are unappealing in a home environment. Several researchers have suggested the instrumentation a floor for tracking purposes [1,12]. These are very appealing approaches, but require somewhat abnormal instrumentation of the floor and are computationally heavyweight. Prior to this work on the Location Service, we were very much driven by the desire to have a single location sensing technology that would deliver room-level positioning throughout the house. As a compromise between the prior instrumented floors work and badging approaches, we arrived at a solution of floor mats that are act as a network of RFID antennas (see Figure 2). A person wears a passive RFID tag below the knee (usually attached to a shoe or ankle) and the floor mat antenna can then read the unique ID as the person walks over the mat. In our original implementation, strategic placement of the floor mats within the Aware Home (see Figure 2) provided us a way to infer room location reliably as an individual walked throughout the house.

Figure 2: The RFID floor mat positioning system. The top left shows a floor mat placed near an entrance to the Aware Home. Underneath the floor mat is the RFID antenna. Strategic placement of mats around the floor plan of the house provides an effective room-level positioning system.

The RFID positioning system has been operational for the past six months and has been available to application programmers through the Context Toolkit. For integration with the new Location Service, each floor mat can be considered a separate source of positioning information. The floor mats are placed in known locations within the home; when a badged individual walks across a mat, the raw positioning data delivered to the Location Service contains the identity of the individual and a position coordinate within the Aware Home reference frame that represents the centroid of the floor mat. No orientation information is provided.

4.2.2 Overhead visual tracking

Much of the long-term research agenda within the Aware Home involves the use of computer vision to infer automatically the whereabouts and activities of individuals within the home. Although room level location information is useful in many applications, it remains very limiting. More interesting applications and research can be done if better location information can be provided. One method of providing this information is through computer vision. The Aware Home has been instrumented with cameras in the ceiling. These cameras provide an overhead view of the home. Currently, there are four cameras in the kitchen, four cameras in the living room, and six cameras in the hallway. The focus of the visual tracking system has been in the kitchen (see Figure 3).

Figure 3: The visual tracking system. Four overhead cameras track moving "blobs" in the kitchen of the Aware Home. The location and orientation of these unidentified blobs is reported to the Location Service.

The visual tracking system attempts to coordinate the overlapping views of the multiple cameras in a given space, such as the kitchen. The system does not try to identify moving objects; its main objective is to keep track of the location and orientation of a variety of independent moving "blobs" over time and across multiple cameras. A wrapper around the visual tracking system provides the raw positioning data of the unnamed blobs to the Location Service

4.2.3 Fingerprint detection

Commercial optical fingerprint detection technology is now currently available and affordable. With the support of the Phidgets infrastructure developed at the University of Calgary [7], it is now easier to prototype applications that use simple sensors and actuators along a USB connection. Over the span of one week, two undergraduates working in our lab created a fingerprint detection system that operated the electronic locks on two of the doors of the Aware Home. Once this capability was demonstrated, we realized that the recognition activity at the doorway was another source of location information. The fingerprint detection system, written in Visual Basic, was modified to communicate a location object to the Location Service. This location object contains the identity of the person whose finger was scanned as well as a spatial coordinate for the location of the door and an orientation value indicating which way the person is assumed to be facing (toward the door).

4.2.4 Open-air speaker ID

The microphone is another interesting sensor used in the Aware Home to help understand the activities of the household. One of the first applications of audio signal processing is the deployment of an open-air speaker ID system. The core technology of the speaker ID was developed by digital signal processing experts at

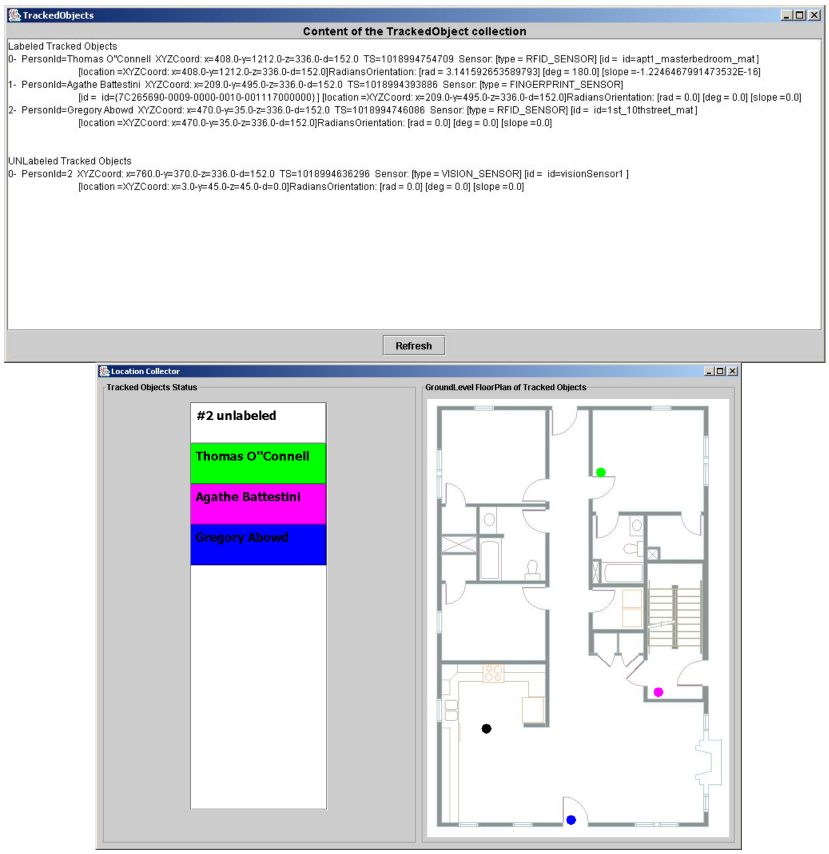

Figure 4: At the top is a listing of the contents of the two collections of tracked entity location objects (named and unnamed). At the bottom a graphical depiction of the same information, showing location relative to the floor plan of the Aware Home.

Georgia Tech and we deployed an "always on" service using this core technology. The microphone is constantly recording 5-second samples and passing that on to the recognizer. Features of the audio sample are compared against features of a known population. If there is a close enough match to a known person, then a location object is communicated to the Location Service. The geometric location provided is the known location of the microphone. In this way, the speaker ID location system functions similarly to the RFID floor mat system. An advantage of the speaker ID system is that it does not require any badges. Another advantage, not currently exploited in our prototype, is that the speaker ID system can identify a speaker as being unknown to the system. In this way, it produces both named and unnamed location objects.

5 Fusion and aggregation of location

Location data from the independent location sensing technologies are sent as XML messages. As long as the individual sensing technology allows for a socket communication, it is straightforward to extract these XML messages and incorporate them into the Location Service. In the Collector subsystem, these XML messages are then converted to Java data objects and stored in a time-sequenced queue. Within the Collector, the main task is to consume the raw location objects and update two collections of tracked objects, one for identified or named entities and one for currently unidentified or unnamed entities.

The Collector has access to the agreed namespace for entities to be tracked. If a label for a location object does not match a name in this namespace, the Collector attempts to translate the label to one of these names using sensor-specific translation tables. So, for example, an RFID location event will contain a label that is a unique integer that is then mapped to the name of an individual who is the declared owner of that tag. Recall that some of the location sensing technologies, such as the visual tracker, produce anonymous location objects that are not assumed to be known entities.

When the Collector reads a location object that does not contain an identity of a known individual, it searches the queue of raw location objects to see if there is one that can be mapped to a known entity and that corresponds to the anonymous one. If so, then the two raw location objects are merged and given the name of the known entity. Each named tracked entity has a special storage area to contain its current location data. Another special storage area contains location data for currently unidentified tracked objects. Figure 4 shows a listing of the contents of the two collections and a simple graphical depiction of location that is derived from the Collector subsystem.

Currently, the algorithm for merging location data uses a fairly straightforward temporal and spatial heuristic. For example, when a location object with no identity is consumed, the merge algorithm tries to find a location object with an identity around the same time and place. While this relatively naïve algorithm has its own problems, it is incorporated into the Collector system in such a way that it can be replaced relatively easily with a more sophisticated routine. For example, with an increased number of location-sensing technologies feeding the queue of raw location objects, scalability of the fusion algorithm may be a concern. Currently, the visual tracking system creates the highest bandwidth of location objects and we are able to handle 15 location object updates per second.

The most important feature of the Collector is that it consumes the raw location data objects and produces what is effectively a database of named and unnamed location objects. Application programmers need only consider this repository of current locations for tracked entities when constructing their applications.

6 Accessing, interpreting and handling location data within an application

Applications need to access location data for relevant entities. Since the Collector provides a model of location data as a repository indexed by identities of known and unknown entities, we need to provide ways for applications to query this repository and trigger events based on application-relevant interpretations of that location information. The Location Service provides a Monitor class in Java that application developers can reuse and extend to perform all three tasks of querying, interpreting and providing application-specific location event triggers.

The kinds of application requests we want to make easy to program are of the following types:

- Where is a particular individual? The answer should be in a form that is relevant to that application (for example, room names for a home).
- Who is in a particular location? Again the location is application-specific.
- Alerting when a person moves into or out of a particular location.
- Alerting when individuals are collocated.

In discussing the functionality of monitors, we will show how these kinds of application questions are supported.

Querying for current location information

The Collector contains current location information for tracked entities. The query subsystem presents a straightforward API to request information from the repositories of named and unnamed tracked entities. Query requests are constructed as objects that represent Boolean searches on all of the attributes for the tracked location objects. A Java RMI server receives query requests and returns all matched records from both named and unnamed repositories.

This query system makes it very easy to request information about any number of entities that are being tracked. In constructing an instance of a Monitor, the relevant parameters of a query are provided and the correct query object is created and issued to the query server. Results of the query are then processed in order of arrival.

Interpreting location for application-specific needs

Up to this point, all of the location information in the Location Service is geometric and relative to a set of pre-defined domains. Application designers want location information in many different forms and it is the role of the Monitoring layer to provide mechanisms for interpreting geometric location data into a variety of alternate geometric and symbolic forms.

We made a conscious decision to represent acquired and collected location information in a geometric form because it is possible to translate that representation into any symbolic representation. The converse is not true. However, to make this translation possible, knowledge of the physical space needs to be encoded, and this knowledge may be very application specific. For example, one application may want to know what floor within a home various occupants are located, whereas another application may want to know the rooms and another one might want to know if someone is facing the television or seated around the dining room table. Each of these spatial interpretations is encoded as a translation table from geometric information to the appropriate symbolic domain. Furthermore, these translation tables are dependent on the coordinate reference frame used for a particular location object, that is, the translation would be different for a home versus an office building.

Most spatial interpretations are based upon carving up a finite coordinate space into a set of subregions. For example, a floor plan for a building is divided into floors that are further divided into named regions, consisting of hallways and rooms. We have provided some facilities for creating spatial interpretations that are region based. We are also working on developing a graphical tool that will make the definition of this class of spatial interpretation even easier.

To facilitate a variety of spatial interpretations, the Monitor class is constructed with a specific instance of a spatial translation table. These translation tables can be reused across different instances of the Monitor class and any of its subclass extensions. Any geometric position data returned from a query is automatically translated with respect to the spatial interpretation and can be delivered to an application.

Filtering and delivering information to applications

Even when it has been translated into a meaningful representation, not all location data is relevant to an application. The last objective of a Monitor is to provide ways to filter the location events that can trigger application-specific behavior. By setting up an application as a Listener for a given instance of a Monitor, the Monitor can then control which location events are delivered to the application.

We have created several examples of monitors that provide the capabilities suggested earlier. All monitors extend the base class Monitor that provides the simple ability to create, send and receive queries and sequentially process the results through a selected spatial interpretation look-up table and send selected events to applications that subscribe as listeners to the monitor.

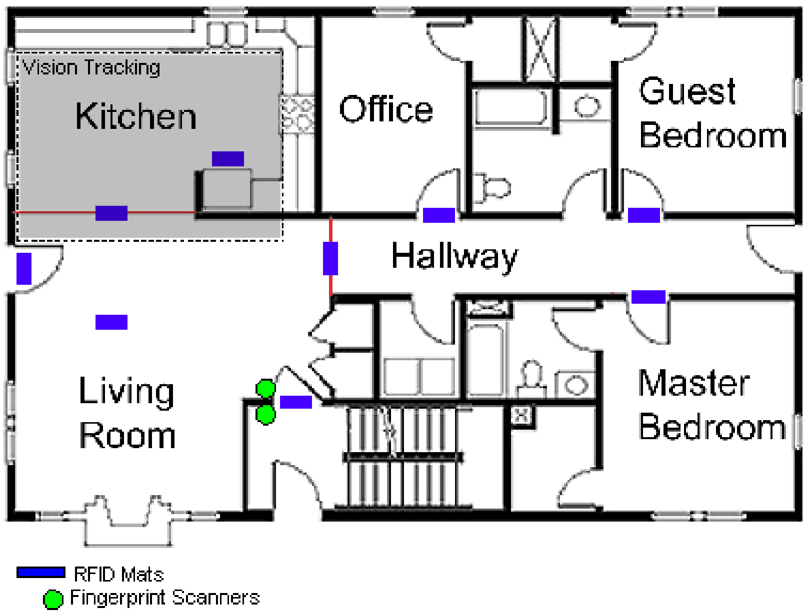

Figure 5: The floor plan of the first floor of the Aware Home, indicating the location of visual tracking, RFID floor mats, and fingerprint detectors that provide location information feeding the Location Service.

7 Sample application development

Figure 5 shows a floor plan of the first floor of the Aware Home, indicating the three location-sensing technologies that have been deployed (the speaker ID system is installed in a basement meeting room). A canonical indoor location application is the In/Out board, which indicates the location of a set of normal occupants of a building. Figure 6 shows a screenshot of a simple In/Out board. This application was originally built within the Context Toolkit to react to location changes from the whole-house RFID system described in Section 4.2.1. It was rewritten to take advantage of the multiple location-sensing technologies.

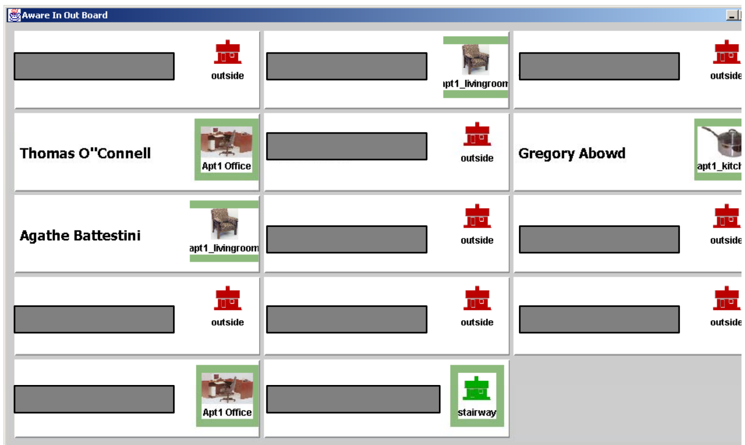

Figure 6: A simple In/Out Board application developed using the Location Service. Names of nonauthors have been suppressed for publication.

One motivating application for us is to use context to facilitate human-human communication [10]. Within an environment like a home or office, we would like to automatically route text, audio and video communications to the most appropriate place based on knowledge of the recipient's context. The Location Service makes it fairly straightforward to determine which room the recipient is located, and in some spaces even the orientation. We have implemented an instant messaging client that will send a text-based message to a display that is nearest to the recipient. A more interesting variation of this messaging can occur if we monitor not only for the location of the recipient but also determine if that person is alone. A final example, not yet implemented would allow us to make inferences about group activities in the home based on collocation in particular rooms, such as the dining room or living room. While location is not always sufficient context to infer human activity, it is often necessary. The Location Service is designed to make it easier inferences on location information to be shared across multiple applications.

8 Conclusions and future work

There is no universal location-sensing technology to support location anywhere-anytime. As a result, for the foreseeable future, operational location systems will have to combine separate location-sensing technologies in order to increase the scale over which location can be delivered. The contribution of the Location Service is as a framework to facilitate the construction of location-aware applications that insulates the application programmer from the details of multiple location-sensing technologies. Motivated by the initial separation of concerns espoused by the Context Toolkit, the Location Service separates problems of sensor acquisition, collection/aggregation, and monitoring of location information. Unlike the Context Toolkit, the implementation of the Location Service presents a cleaner programming interface for the development of location-aware applications; the programmer need only reuse or extend existing monitors to access, interpret and filter location data for entities of interest.

There are several key assumptions that underlie the Location Service, and we reiterate them here as concrete discussion points for future work. First, we assume that geometric representation is sufficient for all location acquisition and collection within the Location Service. When applications need a different representation for location, that interpretation can be made from the geometric representation to any other geometric or symbolic representation. Geometric data is represented along with a frame of reference to facilitate this spatial interpretation. Some may argue that geometric gives up too much information and thus is a privacy concern, but this is an orthogonal issue.

We have decided, somewhat arbitrarily, to assume each location technology can present the raw geometric data as a single point in the reference coordinate space. Upon reflection, relaxing this constraint to allow for raw location representation in the form of regions may be wiser. Geometric reasoning with regions may allow a simple and more efficient way to filter location events, as demonstrated by QOSDream [11]

The Location Service does not store any history of location data. If an application requires historical information, it would either have to be done in the application itself or as part of the state of a monitor. For efficiency purposes, we could easily introduce historical information into the Collector layer of the Location Service.

An important general topic for future research is the extent to which the layered solution presented for the Location Service will actually apply to other context types. such as activity Information about tracked entities is contained within the Collector layer of the Location Service. While this solution is reasonable when we consider only location context, it is a questionable decision once we decide to extend the philosophy of the Location Service to address other forms of context.

Acknowledgments

The authors acknowledge the support of industry sponsors of the Georgia Tech Broadband Institute and the Aware Home Research Initiative. This work is also partially supported by a grant from the National Science Foundation ITR grant (#0121661). The authors would like to thank colleagues at Georgia Tech who helped with the development of a lot of the core technologies presented in this paper. Specifically, we thank Jake Auxier, Aaron Bobick and Irfan Essa for work on the visual tracker, Peter Jensen and Arpit Aggarwal for work on the RFID floor mat system, Sourabh Ravindran, Arpit Aggarwal and David Anderson for work on the speaker ID system, and Christopher Gooley and Scott Saponas for work on the fingerprint detection system.

9 References

1. Addlesee, M., A. Jones, F. Livesey and F. Samaria. The ORL Active Floor. *IEEE Personal Communications*, Vol. 4, No. 5, pp. 35-41, October 1997.
2. Aware Home Research Initiative Web site, http://www.awarehome.gatech.edu
3. Castro, P., P. Chiu, T. Kremeneck and R. Muntz. A probabilistic room location service for wireless networked environments. In *Proceedings of Ubicomp 2001*, pp. 18-34, Springer-Verlag LNCS 2201, Atlanta, GA, 2001.
4. Castro, P. and R. Muntz. Managing context for smart spaces. *IEEE Personal Communications*, Vol. 7, No. 5, October 2000.
5. Dey, A.K. Providing Architectural Support for Building Context-Aware Applications, Georgia Institute of Technology, College of Computing Ph.D. thesis, December 2000.
6. Dey, A.K., D. Salber and G.D. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Anchor article of special triple issue on Context-Aware Computing. *The Human-Computer Interaction (HCI) Journal.* Vol. 16, 2001. See [9] below.
7. Greenberg, S. and C. Fitchett. Phidgets: easy development of physical interfaces through physical widgets. In *Proceedings of UIST 2001*, pp. 209-218, Orlando, FL, November 2001.
8. Hightower, J. & G. Borriello. Location systems for ubiquitous computing. *IEEE Personal Communications*, Vol. 8, No. 4, August 2001.
9. Moran, T. & P. Dourish (eds.) Context-aware computing. Special triple issue of the *Human-Computer Interaction (HCI) Journal*, Vol. 16, 2001.
10. Nagel, K., C. Kidd, T. O'Connell, A.K. Dey and G.D. Abowd. The Family Intercom: Developing a context-aware audio communication system. In *Proceedings of Ubicomp 2001*, Technical Note, pp. 176-183, Springer-Verlag LNCS 2201, Atlanta, GA, 2001.
11. Naguib, H. and G. Coulouris. Location Information Management. In *Proceedings of Ubicomp 2001*, pp. 35-41, Springer-Verlag LNCS 2201, Atlanta, GA, 2001.
12. Orr, R.J. and G. D. Abowd. The Smart Floor: A mechanism for natural user identification and tracking. Short paper in *Companion Proceedings of CHI 2000*, The Hague, Netherlands, April 2000.
13. Want, R., B. Shilit, N. Adams, R. Gold, K. Petersen, J. Ellis, D. Goldberg and M Weiser. The PARCTab ubiquitous computing experiment. Xerox PARC Technical Report CSL-95-1.