Location Handling System for Mobile Computing

Hari Balakrishnan, Tetsuya Nakamura, and Masato Matsuo†

Abstract

To provide adaptive networking services to users in the future ubiquitous network, it is important to handle the user's context including his or her location. This paper describes the design and implementation of Cricket, a location-support system for in-building, mobile, location-aware applications, which was developed at MIT LCS (Massachusetts Institute of Technology Laboratory for Computer Science) as a topic within the framework of the NTT-MIT collaboration. We also describe CHANSE, developed at NTT, which supplies location information to applications by flexibly assigning data sources while hiding the details of the monitoring devices such as an RFID (radio frequency identification) tag system and GPS (global positioning system). This paper gives an overview of these systems.

1. Introduction

The emergence of network-enabled devices and the promise of ubiquitous network connectivity have made the development of pervasive computing environments an attractive research goal. A compelling set of applications enabled by these technology trends is context-aware, location-dependent ones, which adapt their behavior and user interface to the current location in space. To do this, they need to know their physical location with some degree of accuracy, but existing systems and network architectures have the following problems.

① Existing systems do not have a building-wide deployment where spatial regions can be determined to within 0.1–0.2 m^2.

② System development is expensive because an application developer must worry about the details of sensing methods for handling the user's location.

③ Existing systems lack robustness because they cannot substitute one source for another if a source fails.

† NTT Network Innovation Laboratories
Musashino-shi, 180-8585 Japan
E-mail: matsuo@ma.onlab.ntt.co.jp

In the NTT-MIT collaboration, as solutions to the first two items, MIT has developed the Cricket system [1], which allows applications running on user devices and service nodes to determine their physical location. It aims to be a location-support system, rather than a conventional location-tracking system that tracks and stores location information for services and users in a centrally maintained database. By separating the processes of tracking services and obtaining location information, it can accommodate multiple resource discovery systems. And by not tracking users and services, it handles user-privacy concerns adequately. NTT, on the other hand, targeted the last two items and developed CHANSE [2], which supplies location information to applications by flexibly assigning data sources while hiding the details of the monitoring devices such as an RFID (radio frequency identification) tag system and GPS (global positioning system). Applying the Cricket system to CHANSE is the future issue.

2. Cricket

2.1 System architecture

Cricket uses beacons to disseminate information about a geographic space to listeners. A beacon is a small device attached to some location within that space. Typically, it is obtained by the "owner" of the location (e.g., the occupant of a room in an office or home or a building administrator) and placed somewhere unobtrusive like on a ceiling or wall. Cricket does not attach any semantics to the space information advertised by the beacon: any short string can be disseminated, such as the name of a server to contact to learn more about the space or a name resolver for the space to discover resources. Cricket beacons are inexpensive and more than one of them can be used in any space for fault-tolerance and better coverage.

To obtain information about a space, every mobile and static node has a listener attached to it. A listener is a small device that listens to messages from beacons, and uses these messages to infer which space it is currently in. The listener provides an application programming interface (API) to programs running on the node that allows them to learn where they are, so that they can use this information to appropriately advertise themselves and their location to a resource discovery service. The listener can be attached to both static and mobile nodes. For example, when a user attaches a new static service to the network (e.g., a printer), she does not need to configure it with a location or other any attribute; all she does is attach a listener to it. Within a few seconds, the listener infers its current location from the set of beacons it hears, and informs the device software about this via the API. This information can then be used in its own service advertisements.

When a mobile computer has a listener attached to it, the listener constantly listens to beacons to infer its location. As the computer (e.g., a hand-held computer carried by a person) moves within a building, the navigation software running on it uses the listener API to update its current location. Then, by sending this information securely to a map server (for example), it can obtain updates to the map displayed to the user. Furthermore, services that appear as icons on the map are related to the user's current location. The services themselves learn their location information using their own listener devices, avoiding the need for any configuration of individual nodes. The only configuration required in Cricket is setting the space's string to be disseminated by a beacon. The specific string is a function of the resource discovery protocol being used, and Cricket allows any one of several possibilities. Cricket also provides a way for the owner of a room to securely set and change the space identifier that is sent in the advertisements. This is done by sending a special message over the same RF channel that is used for the advertisements, after authenticating the user by a password.

At this stage, we have chosen to allow this change only from within the physical proximity of the room or location where the beacon is located. This makes the system somewhat more secure than if we allowed this to be done from afar. The boundaries between adjacent spaces can either be real (e.g., a wall separating two rooms) or virtual (e.g., a non-physical partition used to logically divide up a room). The precision of the system is determined by how well the listener can detect the boundary between two spaces, while the granularity of the system is the smallest possible size of a geographic space whose boundaries can be detected with a high degree of precision. A third metric, accuracy, is used to calibrate individual beacons and listeners: it indicates how closely the distance from a beacon estimated by a listener matches the true distance. While our experiments show that the distance accuracy of our hardware is better than a few centimeters, the precision and granularity of the system are more important factors. These depend on the algorithms and the placement of beacons across boundaries. Our goal is a system with almost 100% precision with a granularity of a few meters (a region within a room).

The rest of this section describes the design of Cricket, focusing on two fundamental issues: (i) the mechanism for determining the location (the beacon-listener protocol) and (ii) beacon configuration and positioning.

(1) Location determination

Initially, we were hopeful that a purely RF-based system could be engineered and made to work well, providing location information at the granularity of a room, or ideally fraction of a room. We attempted to limit the coverage of an RF transmitter to define the granularity of a geographic-space and used the received signal strength to infer the most likely location. Despite many weeks of experimentation and significant tuning, this approach did not yield satisfactory results [3]. This was mainly because RF propagation within a building deviates greatly from empirical mathematical models (e.g., see also [4]), and in our environment, the signal behavior detected by our inexpensive, off-the-shelf radios was not reproducible across time. We therefore decided to use a combination of RF and ultrasound hardware to enable a listener to determine the distance to beacons, from which the closest beacon can be inferred more unambiguously.

We achieved this by measuring the one-way propagation time of the ultrasonic signals emitted by a beacon, taking advantage of the fact that the speed of sound in air (about 0.34 m/ms at room temperature) is much less than the speed of light (and radio waves) in air. In each transmission, a beacon concurrently sends information about the space by RF transmission, together with an ultrasonic pulse. When the listener hears the RF signal, it uses the first few bits as training information and then turns on its ultrasonic receiver. It then listens for the ultrasonic pulse, which usually arrives a short time later. The listener uses the time difference between the receipt of the first bit of RF information and the ultrasonic signal to determine the distance to the beacon. Of course, the value of the estimated distance is not as important as the decision of which beacon is the closest. Using the time-of-flight of signals to measure distance is not a new concept. GPS uses the one-way delay of radio waves from satellites to estimate distance, while radio-altimeters in aircraft use the time for an electromagnetic signal to reflect off the ground to determine altitude. Collision avoidance mechanisms used in robotics [5] determine the distance to obstacles by measuring the time-of-flight of an ultrasonic signal being bounced off them. It is also possible to measure the distance using the relative velocity of two signals: it is common practice to use the time elapsed between observing lightning (electromagnetic waves) and the accompanying thunder (sound) to estimate the distance to the lightning.

(2) Beacon positioning and configuration

The positioning of a beacon within a room or space plays a nontrivial role in enabling listeners to make the correct choice of their location. For example, consider the positioning shown in Fig. 1. Although the receiver is in Room A, the listener finds the beacon in Room B to be closer and will end up using the space identifier advertised by the latter. One way of overcoming this is to maintain a centralized repository of the physical locations of each beacon and provide this data to listeners. Systems like the Bat [6] essentially use this type of approach, where the central controller knows where each wall- or ceiling-mounted device is located, but it suffers from two problems that make it unsuitable for us.

First, user-privacy is compromised because a listener now needs to make active contact to learn where it is (one feature of Cricket is that listeners are completely passive). Second, it requires a centrally managed service, which does not suit our autonomously managed environment particularly well. Fortunately, there is a simple engineering solution to this problem that preserves privacy and is decentralized. Whenever a beacon is placed to demarcate a physical or virtual boundary corresponding to a different space, it must be placed at a fixed distance away from the boundary demarcating the two spaces. Figure 2 shows an example of this in a setting with both physical and imaginary boundaries. Such placement ensures that a listener rarely makes a wrong choice, unless caught within a small distance from the boundary between two beacons advertising different spaces. In this case, it is often equally valid to pick either beacon as the closest.

2.2 Implementation

In this section, we describe the implementation of Cricket. We describe the system parameters and hardware configuration and discuss some deployment

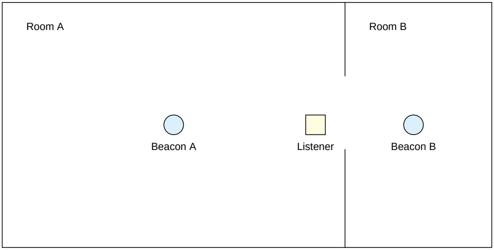

Fig. 1. The nearest beacon to a listener may not be in the same geographic space.

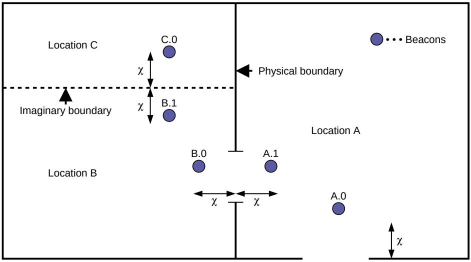

Fig. 2. Correct positioning of beacons.

issues concerning the ultrasonic hardware.

(1) System parameters and hardware

The message size of a beacon RF transmission is 7 bytes in our implementation, and the RF transmission rate of our radios is 1,200 bit/s. It therefore takes about 47 ms for the message to completely reach a listener, during which time an ultrasonic pulse can travel at most about 14 m. The typical range of our RF radios is about 9 m in the building. Therefore, no listener can be farther away than this if it is to detect which space it is in. Cricket is implemented using inexpensive, off-the-shelf, simple hardware parts that cost less than US$10 per beacon and listener.

(2) Ultrasound deployment issues

Ultrasonic interference at the receiver can lead to incorrect distance estimates. It is therefore important to reduce ultrasonic leakage to other locations while trying to provide full coverage to the location served by a Cricket beacon. We achieve this by proper alignment of the ultrasonic transmitters. Figure 3 shows the radiation pattern of the ultrasonic transmitter used in the Cricket beacons. This is shown in (r, θ) polar coordinates, where r corresponds to the signal strength in dB and θ corresponds to the offset in degrees from the front of the ultrasonic transmitter. From the radiation pattern, it can be seen that the direction in which the ultrasound transmitter faces (0°) has the maximum signal strength, while the signal strength drops to 1% (–20 dB) of the maximum value at ±50° away from the 0° direction. We align the ultrasonic transmitter such that the direction of its peak signal strength is at 45° to the horizontal. The beacon is mounted such that the ultrasonic transmitter faces the location intended to be covered by the

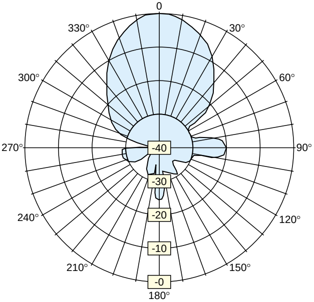

Fig. 3. Radiation pattern of an ultrasonic transmitter.

beacon. This causes the amount of ultrasonic energy transmitted toward distant locations to be small compared with that toward the intended ones. This alignment is easily accomplished by positioning the transmitter at an angle of 45° to the beacon's circuit board and mounting the board flat on the ceiling or wall of the room, as shown in Fig. 4.

3. CHANSE

NTT recently developed context-handling middleware called CHANSE (Context Handling Architecture for Networking Service Environment) [2], which

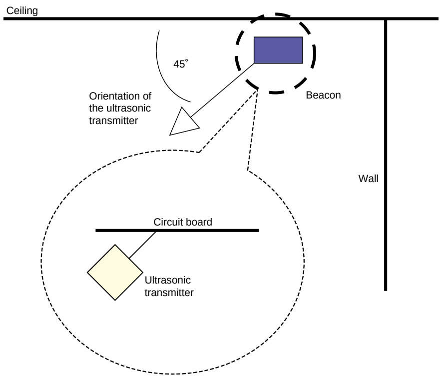

Fig. 4. Correct alignment of Cricket ultrasonic transmitter.

flexibly obtains applicable context data from available data sources and supplies generalized context information to applications. The "context data" (CD) is the raw original data from a data source, such as physical data supplied by a sensor or processed data extracted from a database. The "context information" (CI) is converted from CD or generated from more CD to match the data representation or format designated by each application. CHANSE has a robust architecture that can handle various kinds of data sources and share context information over multiple applications:

① It retrieves necessary context data from alternative sources if a primary source fails to provide the required context data. For example, when you leave a building, CHANSE switches from context data provided by a wearable infrared sensor in an indoor environment to that obtained from a GPS device in an outdoor environment, to get your location.

② It conceals from applications the method of sensing as well as the method of context data conversion. This enables independent development for applications and sources.

③ It also allows applications to prescribe the abstract level of context information that is the attribute value such as "notation" or "precision". For location, there are various kinds of notation such as "latitude and longitude", "address", or "landmark". And for "address", for example, one application might need "city name" while another might need "building name" or "room number". To provide various levels of context information as well as other information (like cost, sampling rate, etc.), CHANSE specifies the API for the application.

A conceptual view of CHANSE in terms of content data and information is given in Fig. 5.

3.1 CD-to-CI translation

The translation of context data into context information is the most important part of CHANSE. CHANSE specifies multiple mechanisms for translating CD into CI, so it can map the CD obtained from data sources to a wide variety of CI. Of course, there is no need to translate CD into CI if the "notation" and "precision" of CI requested by an application equal those of the CD. If only "precision" is different, CHANSE just reduces the precision of the CD. If only "notation" is different, the obtained CD is translated to CI using the translation database. It is important to note that the translation rules and the database consulted for the translation are applied from outside CHANSE, and these are designated by indicating the

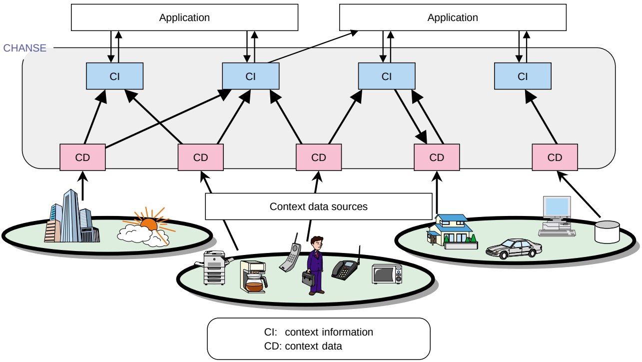

Fig. 5. Conceptual model of CHANSE.

notation and precision of CI and CD for translation. This approach is what makes it possible to handle many different contexts besides location.

3.2 Implementation

The prototype system provides location information while responding to various user situations. It makes use of three completely different kinds of location detection systems: a radio frequency tag (RFT) system, GPS, and a commercially available schedule management tool (scheduler). As a result, even indoor locations that GPS cannot detect are determined by the RFT; and if any equipment failure or the like makes it impossible to observe the user location, it resorts to information retrieved from the scheduler. This switching of CD sources is handled by CHANSE at appropriate timing, hiding the complexity from the application. In practice, when a CHANSE-based location handling system is built, it will be necessary to provide CD representations of the data acquired from each device and CD-to-CI translation rules. These will be specified by CD source providers or other kinds of providers such as a translation rule provider. The prototype was developed using the following environments. The three data sources and the applications run on Windows 2000. CHANSE itself was implemented in Java (JDK 1.3.0-02) running on Solaris 7 with a Sun UltraSparc IIi processor (440 MHz) and 512 MB of RAM. Cybozu [6] is used as the scheduler.

(1) CD sources

• RFT

The RFT system was developed in NTT Network Innovation Laboratories. It consists of several base stations, a server for consolidating the data, and tags. The tags, at 30 mm by 60 mm by 5 mm, are small enough to be carried readily. Each tag sends out a self-identifier (a seven-letter ID) at 7-s intervals on a frequency of 303.8 MHz. When base stations receive these signals, they report the amplitude and ID to the server. The server collects data from all the base stations, calculates the relative distance from each, and determines the location with a precision of 1 to 3 m by triangulation using the base stations as reference points based on the amplitude of the tag signals arriving at each base station. By using the signals from at least three base stations, it is possible to pinpoint the single location where all three circles intersect, which is the location of the tag. Naturally the more base stations there are, the smaller the error in location detection. For registration in CHANSE, the entire area encompassing the central server and base stations was treated as one CD source. The attribute value of the registered CD source is the ratio of (“notation”, “precision”) = (relative coordinates, 1 to 3 m).

• GPS

The GPS receiver used with this prototype emits latitude and longitude signals at 1-s intervals and has a precision of approximately 30 m. Having dimensions of 76 mm by 48 mm by 30 mm, it was connected to a notebook PC by an RS232C interface for the tests. For CD source registration in CHANSE, the entire GPS system was treated as a CD source. The attribute value of the registered CD source is (“notation”, “precision”) = (latitude and longitude, 10 to 50 m).

• Scheduler

Location determination using a scheduler depends on the accuracy of the schedule and location information entered into it in advance by the user, so in that sense, it is not as precise or reliable as the RFT and GPS methods. It should, however, at least provide a means of estimating location when the other sources are unavailable. CD source registration in CHANSE is made treating the server on which the schedule management tool is installed as one CD source. The attribute value is (“notation”, “precision”) = (landmark, site).

(2) Rules of CD-to-CI translation

In general, the representations of CD acquired from each device and the rules of CD-to-CI translation are specified by CD source providers. For example, GPS provides “latitude and longitude” coordinates, and the obtained coordinates can be translated into an “address” by using a map database. Also the coordinates can be translated into a “landmark” by a car-navigation system, for example. In these cases, CD source providers and/or translation rule providers register in CHANSE the attribute value of CD, the methods of translating CD into CI, and the databases (e.g., a map database and car-navigation system database).

(3) Application

Only a simple application was created, since the purpose of the prototype was to confirm the CHANSE functions. The application, shown in Fig. 6, issues requests to CHANSE for user location data, and plots on a suitable map the location data returned by CHANSE in response to the request. The application issues a CI request to CHANSE that indicates the CI notation and precision. It can plot on a map, provided in advance, the resulting location data received from CHANSE. In this way, CHANSE can reduce the time and cost of application development because application developers need not to worry about the details of sensing methods for handling the user’s location.

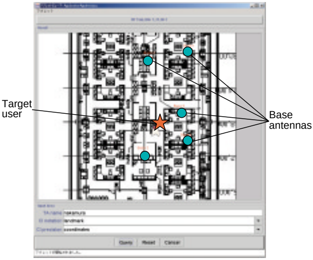

Fig. 6. GUI snapshot.

(4) Operation

In the prototype system, CHANSE responds to requests for location data from the application by selecting from the three CD sources as appropriate, and translating the obtained CD as needed to provide CI in the format designated by the application. Specifically this involves the following processing. Since the CD provided by the three CD sources is expressed in different “notations”, translation databases are provided for each and are used to translate the obtained CD into the requested CI. Based on “precision”, CHANSE ranks the candidates and attempts to get CD in the priority order of RFT > GPS > scheduler. A function for switching CD sources while continuously providing CI was also implemented. If no CD is obtained from a CD source within a certain time, CHANSE judges it to be unavailable and switches to the next candidate and attempts to obtain CD from it. As a result, CHANSE is a robust system that can handle various kinds of data sources.

4. Conclusion

In this paper, we described the design and implementation of two kinds of location handling systems for mobile computing: Cricket, which has been developed at MIT LCS within the framework of the NTT-MIT collaboration, and CHANSE, which has been developed at NTT Network Innovation Labs. The Cricket system is not only a widespread building-wide deployment, but can also determine location within 0.1–0.2 m^2, so it can distinguish regions of rooms. On the other hand, CHANSE can handle the location of a user going outdoors. In addition, CHANSE allows application providers to develop various location-aware services easily, because they need not worry about the details of the sensing methods. It is also a robust system because it can substitute one source for another if a source fails. By registering the Cricket system in CHANSE as one of the CD sources, we expect to make a much more flexible location handling system.

References

[1] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support system," Proc. 6th ACM MOBICOM, Boston, MA, U.S.A., pp. 32-43, Aug. 2000.

[2] T. Nakamura, M. Matsuo, M. Kubota, and K. Koyanagi, "Concept, Design, and Implementation of Context-handling Architecture for Adaptive Networking Services," Communications, Internet and Information Technology (CIIT) 2002, US Virgin Islands, pp. 75-81, Nov. 2002.

[3] A. Chakraborty, "A Distributed Architecture for Mobile, Location-Dependent Applications," Master's thesis, Massachusetts Institute of Technology, U.S.A., May 2000.

[4] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization For Very Small Devices," Tech. Rep. 00-729, Computer Science Department, University of Southern California, U.S.A., Apr. 2000.

[5] Ultrasonics and robotics, http://www.seattlerobotics.org/encoder/may97/sonar2.html, May 1997.

[6] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The Active Badge Location System," ACM Transactions on Information Systems, Vol. 10, pp. 91-102, Jan. 1992.

[7] Cybozu (Share 360) site, http://www.share360.com/

Hari Balakrishnan

Associate Professor, EECS Department and member of the Laboratory for Computer Science (LCS) at Massachusetts Institute of Technology (MIT).

He received a Ph.D. in computer science from the University of California at Berkeley in 1998. His current research interests are in wireless networks, sensor computing, and Internet-based overlay networks. He received an Alfred P. Sloan Research Fellowship (2002), a National Science Foundation CAREER Award (2000), the ACM doctoral dissertation award for his work on reliable data transport over wireless networks (1998), and several best paper awards. He received the Spira Award (2001) and the Junior Bose Award (2002) for teaching excellence at MIT.

Tetsuya Nakamura

Software Engineer, NTT MCL (Multimedia Communications Laboratories), Inc.

He received the B.S. degree in physics from Kyoto University, Kyoto in 1996 and the M.E. degree in quantum engineering and systems science from the University of Tokyo, Tokyo in 1998. Since joining NTT in 1998, he has been working on R&D of a context handling architecture for adaptive networking service systems and context-aware services using various sensor devices such as RFID tags. He is currently working on a framework for information cooperation to support communities in a ubiquitous network. He is a member of the Information Processing Society of Japan (IPSJ).

Masato Matsuo

Senior Research Engineer, Supervisor, NTT Network Innovation Laboratories.

He received the B.E. and M.E. degrees in mechanical engineering from Kyoto University, Kyoto in 1986 and 1988, respectively. After joining NTT in 1988, he researched advanced intelligent network systems, a distributed telecommunication service software architecture, and adaptive networking service systems in ubiquitous computing environments. He is a member of IPSJ and the Institute of Electronics, Information and Communication Engineers of Japan.