Demo: A Framework for Location Information Processing

Zhengdao Xu and Hans-Arno Jacobsen
Department of Computer Science &
Department of Electrical and Computer Engineering,
University of Toronto
10 King's College Road, Toronto, Ontario, Canada, M5S 3G4
zhengdao@cs.toronto.edu, jacobsen@eecg.toronto.edu

Categories and Subject Descriptors

H. [**Information Systems**]; H.4 [**Information Systems Applications**]: Location-based services; H.3.5 [**Online Information Services**]: Commercial services, Web-based services

General Terms

Algorithms, Design, Management

Keywords

Location Information Processing, Location-based Service, Location-based Toronto Publish/Subscribe System

1. INTRODUCTION

Recently, with the advances in wireless communications and location positioning technology, the potential for tracking, correlating, and filtering information about moving entities (i.e., generally speaking any moving objects, such as airplanes, automobiles, trucks, cyclists, pedestrians, goods, and packages etc.) has greatly increased. The knowledge of spatial, temporal, and causal relationships between moving objects would allow the support of highly personalized and effective location-based services (LBS). Such services could track, correlate, and process object positions, object profiles, and past, present, and future object movement patterns.

A wide range of applications can benefit from this potential, including targeted advertisement, friend finding, buddy tracking, mobile dating, mobile gaming, and goods and packages tracking. For instance, shoppers in the vicinity of a bookstore could be targeted with coupons of books they had been inquiring about at an online store (assuming privacy considerations have been worked out between mobile subscriber, network operator, location position information provider, and bookstore.)

Other, more specialized, application scenarios may require the tracking of a group of children playing in the vicinity of a house. This scenario may extend to the continuous checking that none of the children has left the "virtual playing ground," as defined by a distance around the house. This scenario applies to many other examples, including tour groups, work forces, and delivery personnel.

Mobile gaming scenarios require participants to log object of interest sightings (e.g., static land marks or other, designated, mobile participants) and require the dissemination of notifications should a player have expressed interest in being alerted should a given proximity relation or given spatial constellation with other participants have emerged.

The matchmaking and correlation possibilities, provided, privacy and security considerations have been worked out, are endless, as illustrated by the above scenarios. Many of these applications can be broadly classified as *location-aware information dissemination tasks.* A system design and implementation following the publish/subscribe paradigm can effectively support selective information dissemination tasks [4]; mainly due to highly efficient algorithms for realizing this paradigm [5]. Extensions for enriching matching and filtering with location position information of publishing and subscribing entities have also been addressed [4].

However, we argue that to support full location-awareness for many of the listed selective information dissemination scenarios, additional needs must be addressed. These needs refer to the realization and continuous tracking of a *close-to* or other *proximity relation* among different mobile entities or among different mobile and static entities. The information processing challenges lie in the efficient support of these kinds of operations for large-scale mobile environments, involving thousands of mobile subscribers.

In this demonstration, we capture the close-to relation of moving entities with two types of *location constraints* and focus on demonstrating a *framework for location information processing*, as it would commonly be required to support location-aware selective information dissemination applications. The software demonstration draws from a fully implemented location-based services operating in a development environment of a mobile network operator in Canada.

2. LOCATION CONSTRAINT PROCESSING

The location constraints for modeling proximity relations are as follows:

1. n-body constraint: constraints of the form $|{p_1}^t, {p_2}^t, \ldots, {p_n}^t| \leq d$ designate whether the n moving points $p_1, p_2, \ldots, p_n$ are in a circle with a given diameter d. p_i $(1 \leq i \leq n)$ is the identifier of the object i; p_i refers to the coordinate of object i at time t. d is referred to as the alerting distance.

2. n-body static constraint: constraints of the form $|A, {p_1}^t, {p_2}^t, ..., {p_n}^t| \leq d_A$, (where A is the coordinate of some static point) designate whether the n moving points $p_1, p_2, ..., p_n$ are within a given range from the static point A. Here d_A is the alerting distance.

The popular buddy tracking application is an example of a 2-body constraint problem (i.e., two buddies asking to be notified should their location position be less than a specified value.) The mobile gaming scenario presented above can be modeled as an instance of the n-body (static) problem. The child-care scenario presented in the introduction is an instance of the n-body static constraint matching problem with the house as the static body and the children as mobile bodies that need to be tracked.

More formally, the problem we are solving can be stated as follows: *Given a set of constraints $C = \{c_1, c_2, ..., c_m\}$, which designates the desired location constraint relationships among a set of n possibly moving points $P_t = \{p_1, p_2, ..., p_n\}$ in the space at time t, find all constraints c_i in C that are satisfied.*

Our algorithms for efficient location constraint evaluation are based on various object indexing schemes (i.e., grid index and K-d-tree). To index moving objects in space, the whole space is divided into small partitions. In order to track the movement of the mobile objects, each object (mobile or static) is associated with a certain partition according to its current position. This information is continuously updated as objects move. The distance between the partitions in space serves as a rough estimate for the bounds of the distance between objects inside those partitions. This information can be exploited to selectively evaluate potentially matching constraints. This avoids the re-evaluation of all constraints and reduces processing cost. The algorithm details can be found in our companion paper [8].

Related approaches include the buddy tracking system [6] and our previous work [7, 8, 4]. However, the buddy tracking system only solves the 2-body constraints and assumes the same alerting distance for all registered constraints, which we consider a limitation for most of the scenarios outlined above.

3. THE L-TOPSS PROTOTYPE

L-ToPSS (Location-based Toronto Publish/Subscribe System) is our research prototype supporting location-aware information dissemination applications.

Publish/subscribe systems enable clients to exchange information by publishing and subscribing to information of interest. Publish/subscribe systems assist in the dissemination and filtering of large amounts of data and decouple the communication among all clients. However, traditional publish/subscribe approaches do not directly support the correlation of position data of subscribing and publishing clients.

In L-ToPSS, we have experimented with different location-aware publish/subscribe approaches. In Burcea and Jacobsen [4] we correlate subscriber and publisher location to support location-aware matching. In this paper and its companion paper [8] we embed location constraint matching in L-ToPSS and interpret it as an instance of the publish/subscribe matching problem. In this interpretation location constraints constitute subscriptions, location updates constitute publications, and constraint evaluation refers to the identification of all matched subscriptions (i.e., constraints) for a given location update (i.e., publication).

To support this operation, we have developed a general framework for location information processing. The high-level architecture of this framework is depicted in Fig. 1. This framework has been designed to support a large variety of location-based services.

The framework is based on the services offered by a mobile wireless network operator in Canada. These services include location position identification, host-to-mobile subscriber notification dissemination, and mobile subscriber-to-host messaging.

Location position of mobile subscribers is extracted from the mobile wireless network in a pull-based fashion via a web service accessed through an XML query, specifying mobile identification numbers of all mobile subscribers whose location is to be identified. The request must also specify access credentials and will only return location information of subscribers who have agreed to have this location position service enabled for their mobile device. Some carriers such as Bell Mobility [1] provide Internet accessible proxy servers which return the location position data of subscribers who have opted to enable this service and who are billed for location requests. Our framework is designed to interoperate with proxy servers following the Openwave Location Studio SDK APIs [2] for location position identification. In our prototype implementation location position information is determined by the mobile carrier based on cell sector technology, triangulation, and GPS technology. Each location request specifies a desired accuracy, which is matched by the carrier against its possible accuracy. The carrier will try in a best effort fashion to achieve the highest possible accuracy to match the request.

The mobile subscriber can submit location constraints or publish information through a network data link. In our current implementation this is based on the http protocol and browser technology running on the subscriber's mobile device. Servlets on our server handle the submission of data (e.g., location constraints). Notifications signaling successful location constraint evaluations are emitted from our location processing framework via a messaging service to the mobile subscriber. This service is also accessible as a proxy service operated by the network operator. For constraint processing, the L-ToPSS server matches the requested position data against the location constraints, which triggers the notification.

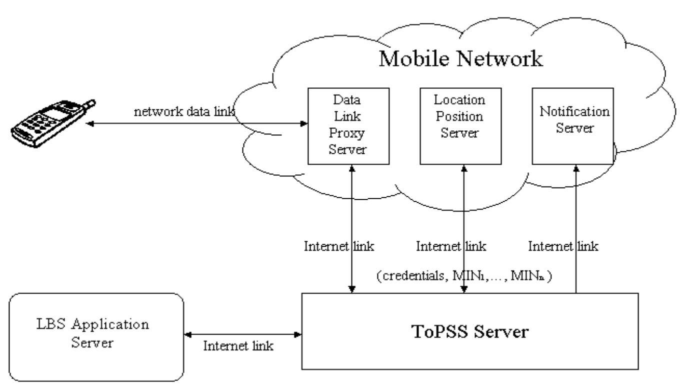

Figure 1: LBS System Architecture

Fig. 2 provides a more detailed architecture of the L-

ToPSS Server. The *publish/subscribe staging* component processes incoming messages from the mobile subscriber via the network data link. Messages are, for instance, location constraint submissions (insert, delete, activate, and deactivate.) In our implementation, the staging component is based on Tomcat. Location constraint submissions are parsed, verified, logged, and inserted in the location constraint matching engine through servlets associated with web forms and pages accessible to the mobile subscriber over the network data link.

The *location update manager* schedules location requests of all active mobile subscribers. A mobile subscriber is active, if it has activated one or more location constraints. Mobile subscribers are billed for location requests by the network operator (e.g., the operator may offer a bulk rate billed as number of location requests per month, which is the case in our deployment.) Our location processing framework design allows different subscribers to choose different location request frequencies. As this frequency determines part of the cost of using any service based on our framework, it is an important control parameter for a subscriber. Note, in the mobile wireless network our framework interoperates with, location position information can only be retrieved in a pull-based fashion. An alternative maybe to determine the location position of mobile subscribers on the mobile device itself and submit it as a message to our host. This maybe possible through self-determination or a GPS receiver on the mobile device. Few of the standard mobile devices available today include a GPS receiver. The *notification engine* is responsible for alerting the mobile subscribers should a constraint have been evaluated.

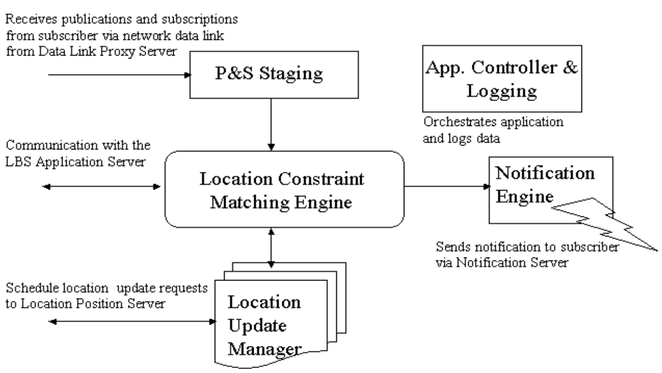

Figure 2: L-ToPSS Architecture

4. DEMONSTRATION

In this demonstration, we will show the functionality of our location processing framework and demonstrate location constraint processing based on a wildlife tracking application. We will also summarize our experience in building the framework and supporting a location-based service with it.

In this application, participants (e.g., park rangers, hikers etc.) patrolling the parks log wildlife sightings. A recorded sighting stores animal type, time, and location position of the sighting. Other participants (e.g., hikers, children etc.) can subscribe to be notified should their location be in the vicinity of a sighting of a specified type. The L-ToPSS server implements the evaluation of the registered location constraints.

In our demo we will use the WAP 2.0 phone simulator [3] to show the interaction between our framework and mobile users. The L-ToPSS server will run on our demonstration machine and interact with a data link proxy server, a location position server, and a notification server.[1]

5. REFERENCES

[1] Bell Developers Website, http://www.developer.bellmobility.ca.

[2] Openwave location studio sdk 1.0 , http://www.openwave.com/us/products/mobile/developer_products/location_sdk/index.htm.

[3] Openwave phone simulator , http://www.openwave.com/us/products/mobile/developer_products/phone_simulator/index.htm.

[4] Ioana Burcea and Hans-Arno Jacobsen. L-ToPSS - Push-oriented location-based services. In *4th VLDB Workshop on Technologies for E-Services (TES'03)*, 2003.

[5] Françoise Fabret, H. Arno Jacobsen, François Llirbat, João Pereira, Kenneth A. Ross, and Dennis Shasha. Filtering algorithms and implementation for very fast publish/subscribe systems. *SIGMOD*, 30(2):115–126, 2001.

[6] Arnon Amir. Alon Efrat. Jussi Myllymaki. Lingeshwaran Palaniappan. Kevin Wampler. Buddy tracking - efficient proximity detection among mobile friends. In *INFOCOM04*.

[7] Zhengdao Xu and H. Arno Jacobsen. Efficient constraint processing for highly personalized location based services. In *VLDB04*, 2004.

[8] Zhengdao Xu and H. Arno Jacobsen. Efficient constraint processing for location-aware computing. In *6th International Conference on Mobile Data Management (MDM'05), Ayia Napa, Cyprus*, 2005.

[1]Should Internet access be enabled for the demonstration the operation of the real system can be demonstrated, otherwise mock-ups of the three servers will run on the demonstration machine.