Modeling Location-based Services with Subject Spaces

Hubert Ka Yau Leung
IBM Toronto Lab
hkyleung@ca.ibm.com

Ioana Burcea
University of Toronto
ioana@eecg.toronto.edu

Hans-Arno Jacobsen
University of Toronto
jacobsen@eecg.toronto.edu

Abstract

The advance in wireless networks and in positioning systems has led to the development of a new generation of mobile applications: location-based services (LBS). LBS offer highly personalized services to users of mobile devices such as telephones, pagers, and PDAs (mobile users) based on their locations, user profiles and context information. The publish/subscribe paradigm is an information dissemination model for loosely-coupled distributed applications, and is appropriate for the implementation of LBS. However, existing publish/subscribe data models and algorithms have limitations when applied to LBS. In this paper, we show how Subject Spaces – a state persistent publish/subscribe model – can be used to model LBS and overcome the limitations of traditional publish/subscribe systems.

1 Introduction

The provisioning of services using location information is known as location-based services. Location-based services (LBS) use the location information of mobile users to provide them with relevant information based on their geographical positions. Information disseminated to mobile users in LBS can potentially be context-sensitive and highly personalized.

Mobile positioning is the enabling technology of location-based services. With the use of network-based position determination equipment (PDE) and handset-based mobile positioning technologies, such as the Global Positioning System (GPS), the location of mobile devices and users can be obtained. The mobile positioning technology is improving over the years. More accurate and precise position information is available. Furthermore, with the technological improvements in pervasive computing, it is expected that a large number of location-aware wireless devices will be available in the future. The network of consumers using PDAs (personal digital assistants), tourists carrying on-line and position-aware cameras and wrist watches, vehicles with computing and navigation equipment will give rise to a wide variety of new wireless applications. This includes the following:

- Tracking the location of mobile 911 callers,
- Tracking and dispatching mobile resources,
- Traffic coordination, and way-finding,
- Location-aware advertising,
- Tourist services, and
- Location-based games.

The first generation of LBS is pull-based. This means that mobile users can query their positions and request services based on their current locations. Examples of pull-based services include requesting directions to the nearest business or service (e.g. the nearest restaurant or ATM).

The next generation of location-based services will not only provide position coordinates and information to mobile users upon request, but will also notify users about information that they would be interested in according to their current position and other relevant context information. For example, a driver driving towards traffic congestion will receive advice to take an alternative route.

This kind of service requires a push-based information system. Information providers push information to end users who come within a certain area. Instead of broadcasting information to everyone who comes close to a certain area, information can be selectively disseminated by using some context information. For example, shops can advertise products to passersby who are in a certain age range. On the other hand, mobile users should be able to filter out unwanted information by specifying their interest in some form of user preferences. For example, some users may not like to receive advertisements with their pagers as they walk down the street.

The research challenges in the provisioning of push-based LBS depend on the support of data management in the system and on the efficient processing of the complex interaction between information providers' data and information consumers' filters.

Most related work on information management in LBS focuses on various implementation aspects of data management, such as mechanisms for storing and retrieving data. However, there have been insufficient studies that devote to the data modeling aspects of LBS. Data models defines the conceptual representation of information and is the foundation for good design of data storage and processing strategies. Existing work on data models in the context of LBS include the data model proposed by Jenson *et al.* [3] and the MOST data model proposed by Wolfson *et al.* [7]. However, as argued in this paper, their approaches have limitations that prevent them from effectively representing information in LBS.

This paper proposes the use of publish/subscribe paradigm to model information in LBS. The publish/subscribe paradigm is an information dissemination model for loosely-coupled distributed applications. Specifically, we describe how the subject space model [6] can be used to represent data in LBS applications. The subject space model is designed for general-purpose state-persistent publish/subscribe systems. Its characteristics allow effective representation of information in LBS applications.

The contributions of the paper include a systematic characterization of the information in LBS applications. This study pioneered in using the publish/subscribe paradigm to model information in LBS applications and laid out a case for the usefulness and expressiveness of the subject space model. The discussions and analysis of this paper will provide insights for future studies of LBS applications.

This paper is organized as follows: Section 2 presents the characterization of information in LBS, and discusses their implications to data modeling. Section 3 argues that the publish/subscribe paradigm can be used to model LBS. Section 4 presents the definitions of the subject space model and uses examples to explain how it can be applied to LBS. Section 5 shows more examples that further illustrate how to use subject space to model data in LBS. Section 6 compares the use of the subject space model with related work and other data models to show the advantages of using the subject spaces to model information in LBS.

2 Information in LBS

This section discusses the types and characteristics of data that can be found in LBS.

2.1 Types of Data

Static geographical features: The static geographical information of the system includes the contour of the landscape, buildings, roads, bridges and other features that describe the geographical environment. This type of information changes infrequently.

Location position information: The location position information refers to the three-dimensional physical coordinates of the user or mobile device in the system. The location information of mobile clients is updated by the infrastructure put in place by the service provider. Depending on the precision of the location detection supported and the mobility pattern of the user, the location may be updated quite frequently.

Dynamic content: Dynamic content in the system is application-specific. It can be weather or traffic conditions, advertisements or any events that may be of interest to some subscribers. Dynamic information changes frequently and it can also be location-dependant.

User profiles: User profiles include user information, which is usually not location-dependent. Profile information includes some or all of the following:

a) Role of the user in the system and service level agreement with the service provider
b) Demographic information about the user,

such as age, gender, etc.

c) User context information representing the current context of the user, such as in a meeting, at sleep, at home etc.

User Preferences and Interests: Subscribers specify detailed conditions under which they would like to be notified. User preferences can also be derived from transaction history or observed usage behaviors. User preferences can be updated over time as the interests of users change. These changes of interests may or may not be location-dependant.

2.2 Data Characterization

This section presents the characterization of information in LBS applications. Related work that addresses each of these issues and their limitations is discussed to further motivate the need for a better data model for LBS applications.

Spatial Aspect: Data in LBS is inherently multidimensional. Both static and mobile location information are represented by points or regions in the 3-dimensional physical space. Dynamic content information and user preferences can be defined by many attributes, and are also multidimensional in nature. Information management in LBS therefore requires support for processing multidimensional data.

There are many studies on multidimensional data structures and spatial database design [2,10]. In fact, the need to represent objects in the 3-dimensional physical space is an important motivation for spatial index structures. LBS also contains static and dynamic content, forming a high-dimensional information space. Since the performance of multidimensional indexes degrades exponentially as the number of dimensions increase in a phenomenon called the curse of dimensionality, it is undesirable to model information in LBS as monolithic high-dimensional objects. Some scoping mechanisms are needed to categorize information and describe their relationships.

Temporal Aspect: The notion of time represents another dimension in which moving objects interact with each other. Therefore, LBS requires support for temporal reasoning. Temporal databases have also been well studied. Temporal databases specialize in managing time-varying data. Work has been done on creating query languages for temporal data, such as TSQL2 [8]. There is also work on indexing and query processing of temporal data [1,11], and data models for temporal database [4].

A good data model for LBS should incorporate both spatial and temporal characteristics.

Inaccuracy, Imprecision, and Uncertainty: Since the reported location of mobile objects is usually not identical to the actual location of the objects, LBS needs to handle inaccurate data. Depending on the available technology, the precision of location information varies. Inaccuracy and imprecision give rise to uncertainty. LBS should always take uncertainty into account during data processing. Unlike most traditional database management systems that process queries based on exact matches, the data model for LBS should allow more general definitions of a "match". The data model should support range queries, nearest neighbor queries and fuzzy matching naturally.

Large Volumes of Data: Many LBS need to deal with a large number of moving objects, and large volumes of data that represent dynamic content and user preferences. The database community has conducted extensive studies in data warehouses that deal with the analysis and processing of large volumes of data. There are also studies that apply data warehouse and On-Line Analytical Processing (OLAP) technologies to spatial databases [9]. However, OLAP was initially designed for static data and postmortem analysis. While the OLAP data modeling aspect may still apply to LBS, the implementation techniques of OLAP, such as data indexing and preprocessing, can only be applied with difficulty in this domain.

Continuous Queries: Mobile users specify their interests and the conditions under which they would like to receive alerts. This is a form of continuous query. Unlike queries in traditional databases that are executed only once to completion and return results based on the current data sets, continuous queries logically run continuously in a database. In terms of data modeling, the challenge here is that queries are also persistent, and are a form of data in the system. The data model for LBS should appropriately represent both data and query, and their interactions.

3 Publish/Subscribe and LBS

In publish/subscribe systems, publishers produce information, while subscribers consume it. Publishers and subscribers are autonomous components that exchange information by publishing events and by subscribing to events of interest. A publisher usually generates a message when it wants the external world to know that a certain event has occurred. All subscribers that have previously expressed their interest in receiving notification of such events will be notified about them. The central component of this architecture is the event broker. This component records all subscriptions in the system. When a certain event is published, the event broker matches it against all subscriptions in the system. When the incoming event satisfies a subscription, the event broker notifies the corresponding subscriber.

Due to the scalability of publish/subscribe systems in terms of number of supported clients and number of processed events per second [5], we believe that this is an extremely promising approach for push-oriented LBS.

Depending on the application to be modeled, each mobile device can act as either a subscriber or a publisher. However, there may be cases when a mobile entity wants to announce certain events to the external world by producing information and, at the same time, it is interested in receiving certain data. In this situation, the same device/client acts as both a subscriber and a publisher. As a subscriber, it receives the desired information, while, as a publisher, it produces events that are forwarded to other interested clients.

In LBS, the location information is needed for expressing both publications and subscriptions. No matter which role the mobile user/device plays in the system, its publications or its subscriptions are associated with its current location. This changing location information is updated by the system, and the clients do not need to update it explicitly. Although each client can have multiple publications or subscriptions, the location information is updated only once for each client instead of once for each publication/subscription. For these reasons, mobile location information should be treated as a special type of data.

There are several mobility scenarios between publishers and subscribers:

- stationary publisher - mobile subscriber
- mobile publisher - stationary subscriber
- mobile publisher - mobile subscriber

where *stationary* means that the entity has a fixed, known location, as in the case of geographical features and locations of building. *Mobile* refers to an entity that changes its location. The mobility of publishers and subscribers adds much complexity to the interaction of data in the system. Note that the case where both the publisher and the subscriber are stationary is not interesting from the LBS point of view.

Generally speaking, the information providers act as publishers in the system, while the information consumers act as subscribers. For examples, user profiles and preferences can be expressed as subscriptions. However, the roles of subscriber and publisher highly depend on the application to be modeled.

4 Modeling LBS with Subject Spaces

We now explain how to use the subject space model to model LBS. Under this model, information is structured by subject spaces, which are metadata of the system. Intuitively, subject spaces are multidimensional spaces and data form regions in these spaces. Publications and subscriptions are declared as a correlation of the regions in the subject spaces.

4.1 Subject space

A subject space is a grouping for related publications and subscriptions that can be described by the same set of properties, and each dimension represents a property. We define a publish/subscribe system to be a set of subject spaces $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_k\}$. The set of subject spaces is used for categorizing information in a publish/subscribe system. Subject spaces are the metadata of a publish/subscribe system and they help describe the values and relationships of publications and subscriptions.

Each subject space is defined as a tuple $\sigma = (D_\sigma, V_\sigma)$, where $D_\sigma = \{d_1, d_2, \ldots, d_n\}$ is the set of dimensions of the subject space and V_σ is the set of values allowed in this space. A subject space is a multidimensional space. Each dimension is defined as a tuple $d = (name, type)$ where *name* is the unique identifier of the dimension, and *type* is a data type. Each dimension d has a domain of values, $dom(d)$, which is the set of all possible values that can be represented by *type*. This model allows each dimension of the multidimensional space to have a different domain. Examples of dimension data types include real numbers, integers, strings, Booleans and enumerated values. User-defined data types that are subsets of these data types are also possible. The domain of a subject space is the Cartesian product of the domains of its dimensions. $V_\sigma = dom(d_1) \times dom(d_2) \times \ldots \times dom(d_n)$.

Example 1 Geographical coordinates can be represented by a subject space *location*, which has 3 dimensions. $D_{location} = \{(x, double), (y, double), (z, double)\}$. ■

4.2 Relationship between Subject Spaces

Subject spaces are related by sharing dimensions. Define $\sqsubset$ as a relation on the set Σ of subject spaces. Specifically, $\sqsubset \subseteq \Sigma \times \Sigma \times \{1, p, 0\}$. In a tuple $(\sigma_1, \sigma_2, \delta) \in \sqsubset$, we refer to the value δ as the degree of containment. Given two subject spaces σ_1 and σ_2,

1. σ_1 fully contains σ_2, $(\sigma_1, \sigma_2, 1) \in \sqsubset_1$, or simply $\sigma_1 \sqsubset_1 \sigma_2$, if $D_{\sigma 2} \subseteq D_{\sigma 1}$;
2. σ_1 partially contains σ_2, $(\sigma_1, \sigma_2, p) \in \sqsubset_p$, or simply $\sigma_1 \sqsubset_p \sigma_2$, if $D_{\sigma 1} \cap D_{\sigma 2} \neq \varnothing$ and $D_{\sigma 1} \cap D_{\sigma 2} \subset D_{\sigma 1}$ and $D_{\sigma 1} \cap D_{\sigma 2} \subset D_{\sigma 2}$;
3. σ_1 and σ_2 are unrelated, $(\sigma_1, \sigma_2, 0) \in \sqsubset_0$, or simply $\sigma_1 \sqsubset_0 \sigma_2$, if $D_{\sigma 1} \cap D_{\sigma 2} = \varnothing$.

Example 2 This example illustrates the use of related subject spaces in LBS. The information about the user profiles can be represented in a user profile subject space $\sigma_{user\_profile}$ with the following structure: $D_{user\_profile} = \{(name, string), (age, integer), (profession, string)\}$.

Suppose that one of the service providers in the system is a coffee shop that sells coffee and cakes. The information about its products is represented using the following three subject spaces: *product*, *coffee*, and *cake*. The product subject space stores general information common to both coffee and cakes. The *coffee* and *cake* subject spaces are supersets of the *product* subject space and store information about each item respectively.

$D_{product} = \{(SKU, string), (price, double), (discount, percentage)\}$

$D_{coffee} = \{(flavor, string)\} \cup D_{product}$

$D_{cake} = \{(type, string), (name, string)\} \cup D_{product}$

With this subject space definition, we may conclude: $\sigma_{\text{coffee}} \sqsubset_1 \sigma_{\text{product}}$; $\sigma_{\text{cake}} \sqsubset_1 \sigma_{\text{product}}$; $\sigma_{\text{user\_profile}} \sqsubset_0 \sigma_{\text{product}}$; $\sigma_{\text{cake}} \sqsubset_p \sigma_{\text{user\_profile}}$. ■

4.3 Region

Data exist in subject spaces in the form of regions. Intuitively, data or regions occupy some volume within a subject space. Formally, a region is defined as a tuple $r = (C_r, V_\sigma^r)$. $C_r = \{c_1, c_2, \ldots, c_j\}$ is the set of constraints of r. A constraint is a subset of the domain of a given dimension. The set of values of constraint c in dimension d is denoted as $dom(c_d)$, i.e. $dom(c_d) \subseteq dom(d)$. V_σ^r is the set of values of region r with respect to subject space σ. V_σ^r can also be interpreted as the spatial extension of region r with respect to σ. Denote the set of dimensions of C_r as D_{C_r}. Let $D_{C_r} \cap D_\sigma = \{d_{i_1}, d_{i_2}, \ldots, d_{i_p}\}$, and $D_\sigma \setminus D_{C_r} = \{d_{i_{p+1}}, d_{i_{p+2}}, \ldots, d_{i_n}\}$. If $D_{C_r} \cap D_\sigma \neq \varnothing$, then $V_\sigma^r = dom(c_{d_{i_1}}) \times \ldots \times dom(c_{d_{i_p}}) \times dom(d_{i_{p+1}}) \times \ldots \times dom(d_{i_n})$. Otherwise, $V_\sigma^r = \varnothing$. A region r is said to be *valid* in σ if $V_\sigma^r \neq \varnothing$. A region can be valid in multiple subject spaces.

Example 3 Location information for mobile users is represented as a point in the location subject space. For example, let the position of a mobile user be ℓ, where $C_\ell = \{x=50, y=20\}$. A point is a special case of a region that has no spatial extension. On the other hand, a building or a shop can be represented as a rectangle on a 2-dimensional map, such as $C_\ell = \{x=[30,100], y=[50, 130]\}$. ■

4.3.1 Object Regions and Interest Regions

There are two types of regions – interest regions and object regions. They have the same definitions as a region but they have different semantics. An interest region represents the set of values within the subject spaces a subscriber is interested in. $\mathcal{I}$ denotes a set of interest regions, and i represents a particular interest region in $\mathcal{I}$. An object region represents values a publisher provides, the state of an entity that may be of interest to one or more subscribers. $\mathcal{O}$ denotes a set of object regions, and o represents a particular object region in $\mathcal{O}$.

Example 4 Using the subject space definitions in example 2, two possible interest regions, i_{coffee} and i_{cake}, in the coffee and cake subject spaces are defined as follows:

$C_{i_{coffee}}$ = {*flavor*="Irish Cream", price<2},

$C_{i_{cake}}$ = {name="black forest", price <20}. ■

4.3.2 Matching Relations between Regions

Define **m** as a relation between regions r_1 and r_2 such that r_1 *matches* r_2. Regions are spatial extensions within the subject spaces. Intuitively, two regions match if they touch each other or they are "close" to each other to some extent. There are many possible matching semantics. The subject space data model does not restrict the meaning of matching and allows multiple matching semantics. Four representative matching semantics and their corresponding matching relations are defined below.

1. **Containment:**

   The containment semantics requires that for r_1 to match r_2, r_2 must be entirely "inside" r_1.

   r_1 $\mathbf{m_c}$ r_2 iff $V_\sigma^{r_1} \cap V_\sigma^{r_2} = V_\sigma^{r_2}$

2. **Enclosure:**

   The enclosure semantics requires that for r_1 to match r_2, r_1 must be entirely "inside" r_2.

   r_1 $\mathbf{m_e}$ r_2 iff $V_\sigma^{r_1} \cap V_\sigma^{r_2} = V_\sigma^{r_1}$

3. **Overlap:**

   Two regions *overlap* each other if they are touching each other in all dimensions.

   r_1 $\mathbf{m_o}$ r_2 iff $V_\sigma^{r_1} \cap V_\sigma^{r_2} \neq \varnothing$

4. **Nearest Neighbor:**

   The nearest neighbor matching semantics is the most general notion of a match. Under this definition, r_1 matches r_2 if r_2 is the closest region to r_1. The degree of closeness is defined by a distance function. If a subject space can be represented as a multidimensional Euclidean space, the distance between two regions can be defined as the Euclidean distance between the closest points between the two regions. In general, the distance function can be defined by using some metrics that indicate the relationships between two values. Let the distance function be *dist*, the definition of the nearest neighbor semantics is:

   r_1 $\mathbf{m_n}$ r_2 iff $\forall r \notin \{r_1, r_2\}$, $dist(V_\sigma^{r_1}, V_\sigma^{r_2}) \le dist(V_\sigma^{r_1}, V_\sigma^{r})$

Example 5: Consider a one-dimensional space with the dimension d in the real number domain. Let constraints be defined as closed intervals of real numbers. Let there be four regions, r_1, r_2, r_3, and r_4.

$C_{r_1} = \{2<d<3\}$, $C_{r_2} = \{1<d<5\}$, $C_{r_3} = \{4<d<6\}$, $C_{r_4} = \{7<d<8\}$

r_2 $\mathbf{m_c}$ r_1; r_1 $\mathbf{m_e}$ r_2; r_2 $\mathbf{m_o}$ r_1 and r_2 $\mathbf{m_o}$ r_3.

Let the distance function be defined as the minimum distance between two regions. Formally, define

$$\mathrm{MINDIST}(p,[lb,ub]) = \begin{cases} lb - p & \text{if } p < lb_i \\ p - ub & \text{if } ub < p \\ 0 & \text{otherwise} \end{cases},$$

where p is a point, lb and ub are the lower and upper bounds of an interval.

$$dist(V_\sigma^{r_1}, V_\sigma^{r_2}) = \min\left(\mathrm{MINDIST}\left(lb_{r_1}, V_\sigma^{r_2}\right), \mathrm{MINDIST}\left(ub_{r_1}, V_\sigma^{r_2}\right)\right)$$

Given the above distance function, r_4 $\mathbf{m_n}$ r_3. ■

4.3.3 Filter

A filter is an integral part of both publications and subscriptions. The definition of a filter applies to both publications and subscriptions. A filter is expressed as

$$\{R \mid \exists\, r_1, r_2, \ldots, r_n \in R : P(r_1, r_2, \ldots, r_n)\}.$$

$R = \{r_1, r_2, \ldots, r_m\}$ is a set of regions and $P(r_1, r_2, \ldots, r_n)$ is a boolean function that takes a number of regions as variables. The expression represents the set of R such that P is true.

4.3.4 Handling Time

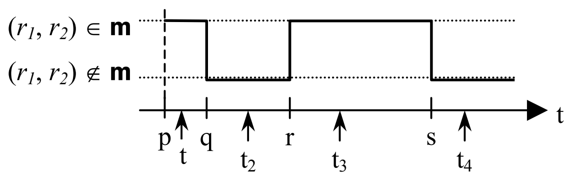
The subject space model can also support the notion of time to express temporal correlation between events. The subject space model, as in most temporal database data models, assumes the time domain to be discrete. The time domain has total order and is isomorphic to a subset of the domain of natural numbers. In other words, the time domain can be represented by the set of natural numbers. The concept of the current time, termed *now*, is an ever-increasing time instant that separates past and future.

We define two functions, the τ-function and the π-function, to represent events in the system. The τ-function returns the most recent time instant when r_1 became a match to r_2; the π-function returns the most recent time instant when r_1 and r_2 turned from a match to a mismatch.

$$\tau(r_1 \,\mathbf{m}\, r_2) = \begin{cases} i & \text{if } r_1 \,\mathbf{m}\, r_2 \text{ at time } i \;\wedge\; (r_1, r_2) \notin \mathbf{m} \text{ at time } i-1 \\ & \quad \wedge\; (\nexists j \text{ where } i < j < now \text{ such that } (r_1, r_2) \notin \mathbf{m} \\ & \quad \text{at time } j-1 \text{ and } r_1 \,\mathbf{m}\, r_2 \text{ at time } j) \\ 0 & \text{otherwise} \end{cases}$$

$$\pi(r_1 \,\mathbf{m}\, r_2) = \begin{cases} i & \text{if } (r_1, r_2) \notin \mathbf{m} \text{ at time } i \wedge r_1 \,\mathbf{m}\, r_2 \text{ at time } i-1 \\ & \quad \wedge\; (\nexists j \text{ where } i < j < now \text{ such that } r_1 \,\mathbf{m}\, r_2 \\ & \quad \text{at time } j-1 \text{ and } (r_1, r_2) \notin \mathbf{m} \text{ at time } j) \\ 0 & \text{otherwise} \end{cases}$$

Example 6 Consider the relationship of r_1 and r_2, where r_1 was added to the system at t_1 and r_2 was in the system when r_1 was inserted. The following diagram depicts the change of relationship between r_1 and r_2.



At t_1, $\tau(r_1 \,\mathbf{m}\, r_2) = p$, $\pi(r_1 \,\mathbf{m}\, r_2) = 0$.
At t_2, $\tau(r_1 \,\mathbf{m}\, r_2) = p$, $\pi(r_1 \,\mathbf{m}\, r_2) = q$.
At t_3, $\tau(r_1 \,\mathbf{m}\, r_2) = r$, $\pi(r_1 \,\mathbf{m}\, r_2) = q$.
At t_4, $\tau(r_1 \,\mathbf{m}\, r_2) = r$, $\pi(r_1 \,\mathbf{m}\, r_2) = s$. ■

4.4 Subscription

A subscription specifies conditions for notifications. A subscription $\mathcal{S}$ is defined as a tuple $\mathcal{S} = (\mathcal{I}_S, f_S)$. $\mathcal{I}_S$ is a set of interest regions that represents the constraints of the subscription. These interest regions can be in different subject spaces. f_S is an expression that represents the set of sets of object regions that satisfy the conditions indicated by the rule of the filter. $f_S = \{\mathcal{O} \mid \exists\; o_1, o_2, \ldots, o_n \in \mathcal{O} : P(o_1, o_2, \ldots, o_n)\}$.

Example 7 Let σ_{car} be a subject space that represent the attributes of a car. The dimensions of σ_{car} can be defined as: D_{car} = {(plate_number, string), (fuel, percentage)}.

The user preference subject space σ_{pref} is for specifying preferred settings for various services. Among others, a dimension in σ_{pref} is *refuel_level*. If the level of fuel in the car falls within the range specified in refuel_level, the driver should receive a reminder to refuel. σ_{gas} is the subject space used by gas stations. σ_{gas} has a price dimension that indicates the current gas price per liter. Like in all LBS, the locations of cars and gas stations are represented as regions in the location subject space $\sigma_{location}$. If a driver would like to receive an alert when his car comes close to a gas station, and the gas price is cheap, he can express the subscription as follows:

Define $\mathbf{m}_{1km}$ as a match relation between objects that are within 1km with each other.

$\mathcal{I}_S = \{\ell_{car}, i_{car}, i_{pref}\}$, where ℓ_{car}, i_{car} and i_{pref} are interest regions in $\sigma_{location}$, σ_{car} and σ_{pref}, respectively.

$C_{i_{car}}$ = {plate_number = *ABC123*, fuel=60%},

$C_{i_{pref}}$ = {*refuel_level* < 20%},

$f_S = \{\mathcal{O} \mid \exists\; \ell_{gas\_station}, o_{car}, o_{pref} \in \mathcal{O} :$

$\ell_{car} \,\mathbf{m}_{1km}\, \ell_{gas\_station} \wedge o_{car} \,\mathbf{m_e}\, i_{car} \wedge o_{pref} \,\mathbf{m_e}\, i_{pref} \}$

■

4.5 Publication

A publication targets content to a subset of the subscribers. A publication $\mathcal{P}$ is defined as a tuple $\mathcal{P} = (\mathcal{O}_P, f_P)$. $\mathcal{O}_P$ is a set of object regions that represents the constraints of the publication. These object regions can be in different subject spaces. f_P is an expression that represents the set of sets of interest regions that satisfy the conditions indicated by the rule of the filter. $f_P = \{\mathcal{I} \mid \exists\; i_1, i_2, \ldots, i_n \in \mathcal{I} : P(i_1, i_2, \ldots, i_n)\}$.

Example 8 A gas station may like to send advertisements to cars nearby whose fuel level is below 70%. This publication can be defined as:

$\mathcal{O}_P = \{\ell_{gas\_station}, o_{gas}, o_{car}\}$, where $\ell_{gas\_station}$ and o_{gas} are object regions in $\sigma_{location}$ and σ_{gas}, respectively.

$C_{o_{car}}$ ={fuel<70%}, $C_{o_{gas}}$ ={price=65¢}

f_P = {$\mathcal{I}$ | ∃ ℓ_{car}, $i_{car}\in\mathcal{I}$: ($\ell_{\text{gas\_station}}$ **m**$_{1\text{km}}$ ℓ_{car}) ∧ i_{car} **m**$_0$ o_{car}} ■

4.6 Matching Relations between Publications and Subscriptions

Define $\mathcal{M}$ as a relation between a publication $\mathcal{P}$ and a subscription $\mathcal{S}$, such that $\mathcal{P}$ matches $\mathcal{S}$.

$(\mathcal{P}, \mathcal{S}) \in \mathcal{M}$ iff $\exists\mathcal{R}\subseteq\mathcal{I}_S : \mathcal{R}\in f_P \wedge \exists\mathcal{R}\subseteq\mathcal{O}_P : \mathcal{R}\in f_S$. In order for a publication to match a subscription, some object regions of the publication must satisfy the subscription filter, *and* some interest regions of the subscription must satisfy the publication filter. If either of these two conditions is not met, this pair of publication and subscription is not a match. This demonstrates the symmetric property of publish/subscribe systems.

Example 9 Reconsider examples 7 and 8. If the car described in the example 7 comes within 1km of the gas station described in the example, the subscription would have satisfied the publication, i.e. $\mathcal{I}_S$ = {ℓ_{car}, i_{car}} $\in f_P$, because the car is within 1km from the gas station and the fuel level of the car is below 70%. However, the publication does not satisfy the subscription because the fuel level is not below the threshold of 20% as indicated in the user preference.

This example illustrates the symmetrical property of the subject space model. Drivers use subscriptions to filter out unwanted information. At the same time, gas stations use publications to target a subset of cars driving by. ■

4.7 Notification Semantics

In a state-persistent publish/subscribe system, a broker only sends notifications upon state transitions. In other words, the broker sends notifications to a subscription $\mathcal{S}$ if a publication-subscription pair ($\mathcal{P}$, $\mathcal{S}$) is added to or removed from $\mathcal{M}$.

State transitions can take place in several situations, include adding a publication or subscription, updating a publication or subscription, and deleting a publication. Note that no notification needs to be sent if a subscription is deleted from the system. Below, we look at each of the operations that may trigger state transitions and specify who should get the notification, if any, and what is the content of the notifications.

After adding a publication $\mathcal{P}$, send notifications to all subscriptions in $Q_P(\mathcal{P})$. The content of the notification is $\mathcal{P}$ – the new publication. Define function Q_P as a mapping from a publication $\mathcal{P}$ to the set of subscriptions that matches $\mathcal{P}$. Formally, $Q_P(\mathcal{P}) = \{\mathbf{S} \mid \forall\mathcal{S}\in\mathbf{S} : (\mathcal{P},\mathcal{S})\in\mathcal{M}\}$.

After adding a subscription S, send notification to the subscription $\mathcal{S}$ if $Q_S(\mathcal{S})$ is non-empty. The content of the notification is $Q_S(\mathcal{S})$ – the set of all existing publications that satisfy the new subscription. Define function Q_S as a mapping from a subscription $\mathcal{S}$ to the set of publications that matches $\mathcal{S}$. Formally, $Q_S(\mathcal{S}) = \{\mathbf{P} \mid \forall\mathcal{P}\in\mathbf{P} : (\mathcal{P},\mathcal{S})\in\mathcal{M}\}$.

After updating a publication $\mathcal{P}$, send notifications to subscriptions in $Q_P^t(\mathcal{P})$. The content of the notification is $\mathcal{P}$ – the updated publication. After updating a publication $\mathcal{P}$, the system should notify the subscriptions that have become matches with $\mathcal{P}$. Let the update take place at time t. Define function Q_P^t as a mapping from a publication $\mathcal{P}$ to the set of subscriptions that were not in $Q_P(\mathcal{P})$ when evaluated at t–1 and are members of $Q_P(\mathcal{P})$ when evaluated at t. Formally, $Q_P^t(\mathcal{P}) = Q_P(\mathcal{P})\big|_t \setminus Q_P(\mathcal{P})\big|_{t-1}$.

After updating a subscription $\mathcal{S}$, send notification to $\mathcal{S}$ if $Q_S^t(\mathcal{S})$ is non-empty. The content of the notification is $Q_S^t(\mathcal{S})$ – the set of publications that just became satisfied by the subscription after the update. After updating a subscription $\mathcal{S}$, the system should notify the updated subscription about publications that have become matches with $\mathcal{S}$. Let the update take place at time t. Define function Q_S^t as a mapping from a subscription $\mathcal{S}$ to the set of publications that were not in $Q_S(\mathcal{S})$ when evaluated at $t–1$ and are member of $Q_S(\mathcal{S})$ when evaluated at t. Formally, $Q_S^t(\mathcal{S}) = Q_S(\mathcal{S})\big|_t \setminus Q_S(\mathcal{S})\big|_{t-1}$.

5 Case Studies

In this section, we show more examples that further illustrate how to use subject space to model data in LBS. We look at the applications of traffic coordination and location-aware advertising.

5.1 Location-Aware Advertising

Retailers can use location-aware advertising to send advertisements to wireless users who have opted to receive information about certain products or services. These advertisements target consumers who are in proximity to the store.

This example demonstrates an application of location-based advertising, and shows how highly customized subscriptions can be expressed under the subject space model.

Like in all LBS, the locations of shops and mobile users are represented as regions in the location subject space σ_{location}. Define σ_{shop} as the subject space that specifies the location and the service of shops. $\sigma_{\text{shop}} \sqsubset_1 \sigma_{\text{location}}$ and D_{shop} = {(*shop_type*, *string*), (*shop_name*, *string*), (*ad*, *string*)} $\cup$ $D_{location}$. The *shop_type* dimension describes the type of business of the store. The *ad* dimension is the text of the advertisement that is sent to shoppers.

A coffee shop can send advertisements to mobile users who come within 1km of the coffee shop. The publication can be expressed as follows:

$\mathcal{O}_{\text{P}} = \{\ell_{\text{coffee\_shop}}, o_{coffee}\}$,

$C_{o_{coffee\_shop}}$ ={*shop_type* = *coffee*, *shop_name* = "ABC Coffee Shop", *ad* = "ABC Coffee Shop brews good coffee!"}

$f_P = \{\mathcal{I} \mid \exists\ \ell_{\text{shopper}}, i_{shop} \in \mathcal{I} : (\ell_{\text{shopper}}\ \mathbf{m}_{\text{1km}}\ \ell_{\text{coffee\_shop}}) \wedge i_{shop}\ \mathbf{m}_{\text{e}}\ o_{coffee\_shop}\}$

Suppose a coffee-lover would like to receive advertisements of coffee shops. However, he does not like to drink two cups of coffee less than 5 hours apart. This subscription can be expressed as follows:

$C_{i_{shop}}$ = {*shop_type* = *coffee*},

$f_S = \{\mathcal{O} \mid \exists\ \ell_{\text{coffee\_shop}}, o_{coffee\_shop} \in \mathcal{O} : \ell_{shopper}\ \mathbf{m}_{\text{1km}}\ \ell_{\text{coffee\_shop}} \wedge (now - \tau(i_{shop}\ \mathbf{m}_{\mathbf{e}}\ o'_{coffee\_shop}) > 5 \text{ hrs})\}$

The condition $\ell_{shopper}\ \mathbf{m}_{\text{1km}}\ \ell_{\text{coffee\_shop}}$ means the shopper is physically *close* to a coffee shop. $\tau(i_{shop}\ \mathbf{m}_{\mathbf{e}}\ o'_{coffee\_shop})$ is the time that the shopper *entered* into a coffee shop, not necessarily the one he is approaching. If the time that he last entered a coffee shop and the current time is more than 5 hours apart, he would receive a notification. According to the definition of $Q_S^t(\mathcal{S})$, this shopper will receive a notification of ABC Coffee Shop publication, with information about the shop location, shop name, shop type and the advertisement.

5.2 Traffic Coordination

A traffic coordination system would like to notify drivers of the traffic condition ahead if they are approaching a traffic jam. Traffic conditions on a highway may be monitored by the following algorithm. Assume sensors are set up on the highway at points A and B to detect the plate number of the cars passing the point. If the moving average of the time for cars to travel from point A to point B is above a certain value, the section of highway from A to B is said to have a traffic jam. If a highway is modeled as a 1-dimensional space, the publication can be formulated as follows:

$$\left(\left(\sum_{j-n}^{j}\Big(\pi(\ell_j\ \mathbf{m}_{\text{e}}[A,B]) - \tau(\ell_j\ \mathbf{m}_{\text{e}}[A,B])\Big)\right) \div n\right) > 2 \text{ min}$$

$\wedge\ (\ell < A) \wedge$ (message = "Traffic jam ahead!").

If the traffic between A and B is slow, cars approaching A, i.e. ($\ell < A$), will be notified.

6 Analysis

In this section, we discuss the properties of the subject space model and argue that it is a better approach to model LBS application than other existing data models.

6.1 Advantages over Related Work

There are two prominent approaches on data models for LBS in the literature: (1) multidimensional data modeling for Location-Based Services by Jensen *et al.* [3] and (2) the MOST data model by Wolfson *et al.* [7]. In this section, we look at these two approaches more closely and compare them with the use of publish/subscribe paradigm to model information in LBS.

The model proposed by Jensen *et al.* is an extension of the OLAP data model. The use of categories in the OLAP data model allows the representation of different levels of granularity of information. For example, the relationships of the concepts of country, province, city and district in the location dimension can be represented by containment relations, and information can be retrieved by the drill-down or roll-up operations. The notion of partial containment is also addressed by extending the OLAP model by defining new containment semantics between categories. The major limitation of this data model lies in the fact that the partial containment relationships between category types are non-commutative. That means if category A partially contains category B, the converse is not true. This mathematical restriction is not suitable in the LBS applications. If a road that extends out of a city partially contains the city, it follows naturally that the city partially contains the road. In the subject space model, the use of set theory can model the notion of containment easily, and the relationships between values are naturally represented by the distance between regions in multidimensional spaces. The notion of matching semantics also adds flexibility in describing the relationship between regions in subject spaces. Also, using OLAP to model LBS gives the illusion that the operations in LBS can be solved by the OLAP technologies. However, OLAP is designed for analyzing historical data, and is inappropriate for processing real-time continuous queries.

The Moving Objects Spatio-Temporal (MOST) data model uses mathematical functions to describe movement patterns of moving objects. Using functions, the object positions can be calculated at any given time. This modeling approach addresses the dynamic nature of data in LBS since frequent data updates are no longer necessary. However, a fundamental drawback of this approach is that *a priori* knowledge about object movement is required. In the LBS case, it is difficult to predict the mobility patterns of users. Therefore, it is difficult, if not impossible, to predefine functions that describe motions of clients. Also, this data model cannot model dynamic content, which is required by many LBS applications.

6.2 Advantages over Other Publish/Subscribe Models

This paper has shown how the subject space model and the publish/subscribe paradigm can be used to model information in LBS. There have been many studies on publish/subscribe systems in the literature. What sets the subject space apart from other publish/subscribe models is the state-persistent characteristic. Conventional publish/subscribe systems are treated as stateless messaging systems. The subject space model presents several advantages over stateless publish/subscribe models. These advantages are discussed next.

1. Avoid redundant notification

The Subject Space model is the first to introduce the notion of state-persistence. State persistence is very natural in publish/subscribe systems and is applicable in many application domains. It can also overcome the redundant notification problem of the stateless models.

In a state-persistent publish/subscribe system, notifications are only sent upon state-transitions. In terms of subject space terminology, a state transition occurs when an interest region *enters* an object region, or vice versa. In LBS, this can happen when a car drives into the 1km radius of a gas station. As the location of the car is continuously updated while it is within the region, however, no further state transitions occur. If the system were to send a notification to the car driving towards a gas station, the notification would be sent only once as the car enters the proximity of the gas station. Traditional stateless publish/subscribe models cannot capture this semantics, and will send redundant notifications to the car as it drives through the region. State-persistence is therefore an advantage over stateless publish/subscribe data models.

2. Symmetrical publish/subscribe

Subject spaces can model symmetrical publish/subscribe. The symmetrical nature of publications and subscriptions was independently identified by Rjaibi *et al.* [1]. In conventional publish/subscribe systems, if a publication matches a subscription, it is also implied that subscription matches the publication. However, there are situations where the publisher would like to send notification to subscribers who do not like to receive the information, or vice versa. A symmetrical publish/subscribe system will only send notification to those subscribers whose subscriptions satisfy the publication, and vice versa. The symmetry allows subscribers to filter out unwanted information, and lets publishers target information to a subset of subscribers.

3. Use of update operations

If LBS are modeled with traditional publish/subscribe systems, when a client that has certain subscriptions changes his position, the system has to unsubscribe in the name of that client for the old position, and subscribe again for the new position. This aspect represents a major disadvantage of a stateless model. In the subject space model, this problem is never met because, when the client is moving, the corresponding location is only *updated* in the system. In other words, only the *location* dimension is updated, all other information contained in the subscription is not modified. Moreover, if the same client has several subscriptions, when he moves, the location is updated only once – for the client, not for each subscription individually.

7 Conclusion

This paper has shown the use of the subject space model, a state-persistent publish/subscribe data model, to represent information in LBS. This data model addresses the characteristics of data in LBS, and provides flexible semantics to model many different scenarios in LBS. We have also shown that the subject space model has advantages over existing data models designed for LBS, and is better than conventional stateless publish/subscribe data models.

About the Authors

Hubert Ka Yau Leung is a software developer at IBM Toronto Lab. He received his M.A.Sc. degree in the Department of Electrical and Computer Engineering at the University of Toronto and a B.A.Sc. degree from the University of Waterloo.

Ioana Burcea is a graduate student in the Department of Electrical and Computer Engineering at the University of Toronto. She received a B.A.Sc. degree from the Politehnica University of Bucharest – Romania. Her main research interests are distributed systems, mobile and pervasive computing and semantic web.

Hans-Arno Jacobsen is a Professor of Electrical and Computer Engineering and of Computer Science at the University of Toronto. His principal areas of research include middleware systems, distributed systems, aspect-oriented programming, and data management.

8 References

[1] B. Salzberg and V. J. Tsotras. A Comparison of Access Methods for Time Evolving Data. *ACM Computing Surveys*, Vol. 31, No. 2, June 1999.

[2] C. Bohm, S. Berchtold, and D. Keim. Searching in High-dimensional Spaces – Index Structures for Improving the Performance of Multimedia Databases. ACM Computing Surveys, 33(3):322 - 373, September 2001.

[3] C. S. Jensen, A. Kligys, T. B. Pedersen, I. Timko. Multidimensional Data Modeling for Location-Based Services. The 10th ACM International Symposium on Advances in Geographic Information Systems, November 2002.

[4] C. S. Jensen, M. D. Soo, and R. T. Snodgrass. Unifying Temporal Data Models. Chapter 6 of Temporal Database Management, dr.techn. thesis by Christian S. Jensen, April 2000.

[5] F. Fabret, H-Arno Jacobsen, F. Llirbat, J. Pereira, K. A. Ross, D. Shasha, Filtering Algorithms and Implementation for Very Fast Publish/subscribe Systems. *SIGMOD* 2001.

[6] H. Leung. Subject Space: State-Persistent Model for Publish/Subscribe Systems, CASCON, 2002. (also available as H. Leung and H.-A. Jacobsen Computer Science Research Group, University of Toronto, CRSG, nb. 459, September, 2002.)

[7] O. Wolfson, B. Xu, S. Chamberlain, L. Jiang. Moving Object Databases: Issues and Solutions. In International Conference of Scientific and Statistical Database Management, (SSDBM'98), pages 111-122, Capri, Italy, July 1998.

[8] R. T. Snodgrass (ed.), I. Ahn, G. Ariav, D. Batory, J. Clifford, et. al. The TSQL2 Temporal Query Language. *Kluwer Academic Publishers* 1995.

[9] T. B. Pedersen and N. Tryfona. Pre-aggregation in Spatial Data Warehouses. In *Proceedings of the Seventh International Symposium on Spatial and Temporal Databases*, 2001.

[10] V. Gaede and O. Günther. Multidimensional access methods. *ACM Computing Surveys*, 20(2):170–232, 1998.

[11] V. J. Tsotras, C. S. Jensen, and R. T. Snodgrass. An Extensible Notation for Spatiotemporal Index Queries. *ACM SIGMOD Record*, 27(1):47–53, March 1998.

[12] W. Rjaibi, K. Dittrich and D. Jaepel. Event Matching in Symmetric Subscription Systems. CASCON, 2002.