An Open Design Privacy-enhancing Platform Supporting Location-based Applications

Tran Khanh Dang, Trong Nhan Phan, Chan Nam Ngo, Minh Tam Vo, Luong Khiem Tran
Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology, Vietnam
{khanh, nhanpt}@cse.hcmut.edu.vn
Nguyen Nhat Minh Ngo
Department of Information Engineering and Computer Science
University of Trento, Italy
ngo@disi.unitn.it

ABSTRACT
The world of location-based services (LBS) has been becoming more diversifying and amazing with its rapid growth in recent years. Moreover, the development has spread to many aspects in all walks of life and got powerful promotion from advanced information and communication technologies. Its pervasive moves, however, leave great concerns behind, which can cause roadblocks in the path of its prosperity. Three of them, identified as heterogeneity, user privacy, and context-awareness, have called for much attention and investigation in both research and industry community world-wide. In response to the call, we propose an elastic and open design platform named OpenLS Privacy-aware Middleware (OPM) for location-based applications as a unified solution to these issues.

Categories and Subject Descriptors
H.2.8 [Information Systems]: Database Applications – Spatial databases and GIS.

General Terms

Keywords
Middleware, user privacy, location-based services, OpenLS, context awareness.

1. INTRODUCTION
LBS turn out to be valuable resources where service providers (i.e., LBS providers) mine for their benefits. For example, “Where is the nearest hospital from my position?”; “How many hotels in the range of two kilometers around me?”, or “Notify me when I and my buddy are close enough to each other.” Such services, for some reason, can be useful to individuals, community, and society as well. Each LBS provider implements his own peculiar way for user demands. This emerges the diversity of not only service entry-points but also multi-platform variants. Moreover, various positioning techniques (e.g., Cell-ID, GPS, and AGPS) add more obstacles for LBS to be built up and utilized. Eventually, LBS providers have to devote lots of energy and efforts to establishing such services. As a result, there is a crucial need to have an elastic LBS middleware hiding the heterogeneity of LBS industry, providing common application programming interfaces (APIs) for quickly developing LBS, and assisting LBS providers in their business. In addition, there will be a shortcoming if context-awareness is missing. The more context LBS are aware of, the more profits they bring to both providers and clients. Frankly to say, context-awareness, itself, is an important factor contributing to the flexibility of LBA and the pleasure of end-users. Last but not least, communication standards are strongly required in order to make the LBS middleware flexible and adaptable to the variety of LBS development.

If the attempt of a LBS middleware aims at generating a common roof and being transparent to LBS development, privacy protection is expected to preserve user sensitive information. In order to get added-values in return, one has to sacrifice his position to LBS. Definitely, he does not feel comfortable when enjoying these services due to the fact that he has no idea whether his location is revealed or misused. Besides, his positions can lead to other privacy violations because places where end-users are mostly reflect their interests. For instance, an adversary can take the advantage of one's position so that he can infer or derive personal matters such as the person may get mental illness because of his frequent visit to the psychiatric hospital or psychology center. Even worse, a malicious attacker can make the best of one's position to know where he either currently is or was and then make something go against him. Consequently, end-users are unwilling or afraid of using LBS. This really makes sense and does harm to the rise of LBS. Thus, there should be a way to prevent adversaries from linking between user position information and user identity.

Hardly can we deny the important role and great influences of LBS to our daily work whereas we are trying our best towards a better life. To meet end-users’ needs, a number of studies have been focusing on this field and coping with LBS ubiquitous development and user privacy protection, ranging from academia [1,3,5,17,21] to commerce [23,25]. Nevertheless, they just put their investigation into either the former or the latter. Some work deals with the problems but at the low level of preserving user privacy [3,5,21]. To the best of our knowledge, there are few LBS middleware platforms supporting heterogeneity and high-level
Others pay attention to privacy protection as well. Depending on LBS middleware, which hides the complexity of positioning techniques as well as LBS infrastructure; and (4) Privacy protection is integrated into the OPM (i.e., Application Middleware and Location Middleware) with more high scalability and context-aware adaptability.

The rest of the paper is organized as follows. Section 2 describes related work that is very close to our direction. In section 3, we briefly review some knowledge about privacy protection as well as context awareness for the conception-related preparation afterwards. Then we introduce our proposed middleware and show how it interacts with other actors in LBS supply chain in section 4. Next, more details about Application Middleware and Location Middleware, constituting the complete middleware, are presented in section 5 and section 6, respectively. Section 7 gives our discussion and future work on the whole. Finally, we make our conclusion in section 8.

2. RELATED WORK

In terms of literature vision, there are several proposals for middleware construction. Some concentrate on position management and transparency [7,28]. Others pay attention to representing and accessing to location data and geographic content [11,13,15]. Having the same interest, Küpper et al. [1] present an open design middleware called TraX, which implements OpenLS and MLP standards and supports a broad range of LBS. It can be deployed and configured in various ways. User privacy-related issues, however, are left as their future work. The same direction as the previous comes from Vo et al. [27], but it has a little difference on their purposes. Their work aims at not only different positioning technologies but also general solutions for the issues known as scalability and interoperability in an LBS middleware. Privacy protection and context-awareness, however, are not taken into consideration. Another approach originates from Jo et al. [3]. Depending on LBS-related standards, they introduce an Open LBS platform to develop LBS. Applying industrial standards from Open Mobile Alliance and 3rd Generation Partnership Project makes the platform reach its high applicability. Besides, user privacy is somewhat assured by the combination of three components including privacy check entity, privacy profile registry, and pseudonym mediation device. Furthermore, the collaboration gives users a good chance to join the operation called personalization. They can make some configurations to specify those who can access their information with a certain degree of accuracy. Yet, privacy solutions are not completely integrated into the platform other than user policy, Different from the others, Ardagna et al. [5] are interested in a privacy-preserving architecture. As a result, a Location Middleware is used to protect users’ positions. Nonetheless, it aims at satisfying privacy preferences so much that other privacy techniques as well as their supports are not discussed and mentioned clearly. On the other hand, it cares about problems involving in multiple location providers more than LBS providers, so the architecture does not help LBS providers conveniently build up their LBA. With the same purpose like the previous, another middleware known as PoSIM is claimed by Bellavista et al. [21]. The middleware concentrates on controlling heterogeneous positioning systems but context awareness via policy management. User privacy and the utility for LBS service providers, therefore, are not investigated so much. Two-level proxy-based middleware is pointed out by Bellavista et al. [17]. It takes the duty of location privacy protection in Wi-Fi network by relying on pre-defined security levels. Even though its mechanism limits frequent position updates, it makes the best use of Wi-Fi network only and reduces other privacy techniques to integration restriction. To sum up, the high level of user privacy, along with context-awareness, is not much concentrated on these platforms.

In reality, the demand of having a middleware for LBS development brings a considerable amount of attraction to industry community. Creativity Software [19] shows an advanced middleware solution named MiddleWise. It ensures smooth communication between location service client applications and the location positioning equipment. The robust, field proven Middleware ensures full connectivity with standards-based location infrastructure in any mobile network [19]. Although there is the component named security and privacy, only security is mentioned according to its description. Having the same concerns, Xypoint Location Platform, from TeleCommunication Systems [22], is recommended. However, its critical privacy functions address mostly profile and position information access control more than making them available to LBS whereas personal privacy is still assured. Preferring to user privacy, Mobilaris [20] notices user position information as a valuable asset needing to be protected. Hence, Pacific Ocean middleware, which complies with all relevant EU standards and uses EthicsEngine™ for advanced protection algorithms under EU directives, is constructed. More details about privacy aspect as well as how it is applied to the middleware are left blank. In short, most commercial products put more efforts on the cost effective and fast launch of LBS while user privacy protection is left behind or supported a little bit.

3. BACKGROUND

3.1 Privacy Protection

Westin defines “Privacy is the claim of individuals, groups or institutions to determine for themselves when, how and to what extent information is communicated to others” [28]. In LBS, typical private information of a user includes identity, location, and path. Thus, privacy protection in LBS can be classified into three categories [42]:

- **Identity privacy** is to protect LBS user’s identity from being revealed directly or indirectly from LBS information. To achieve this, the protection methods limit the quasi identifiers [29] as much as possible (e.g., home or company address). This kind of method is only suitable for anonymous LBS (e.g., navigation, proximity query, advertisement).
- **Location (position) privacy** is to protect LBS user’s location by perturbing or decreasing the precision of user’s location. This kind of method is suitable for LBS that require authentication.
Path privacy protects information associated with LBS user’s movement (i.e., path or trajectory) from which attackers can collect paths to infer personal information of LBS users such as identity, preferences, hobbies, etc. This category of privacy is much more complicated than the above ones.

Lately, the author in [43] also discusses about query privacy in which the methods protect LBS user’s query content. In cases of LBS users in that they are willing or obliged to reveal their location but have the right to keep their query content in private. For example, truck or taxi drivers need to reveal their location to their company, but they can have their query content kept secretly.

Based on that, several research groups have introduced techniques to preserve LBS user privacy against attackers. The most trivial and obvious approach is to remove the information that directly reveals the user identity (e.g., SSID), but other information (i.e., time or location) can also indirectly reveal the user identity. They are the so-called Quasi-Identifier [29]. Thus, more advanced techniques have been developed to prevent that. The techniques can be classified into four categories as followings:

- **Anonymity:** to make user indistinguishable among (k-1) other users for user identity protection [10,30,31]. In anonymous LBS (e.g., navigation), this category of techniques aim for identity privacy.
- **Obfuscation:** to make user location imprecise or inexact [32] or generalize the location into a region [33,34]. In LBS that require user identity, this category of techniques aims at location privacy.
- **Transformation:** to map user location into another location [35] and repeatedly issue queries to process proximity services.
- **Encryption:** like Private Information Retrieval, based on cryptography, the database server cannot know the data in the query and result [36].

Each privacy technique has its own characteristics and application domains. Hence, depending on what kind of objects needing to be protected, privacy techniques should be chosen carefully to give the best protection to users.

### 3.2 Context Awareness

The term “context” is officially defined in Meriam-Webster [44] as “the interrelated conditions in which something exists or occurs”. However, this definition is too general, which cannot clearly explain the term in computer science while this term is of many uses such as contextual help, context menu, and multitasking context switch. To define the term, Schilit et al. [45] list some examples including computing context, user context, and physical context. Later Chen et al. [46] add time context and discuss about context history to collect context over time period to use in some applications and define “the set of environment states and settings that either determines an application’s behavior or in which an application event occurs and is interesting to the user”.

Some other definitions of the term, Schmidt et al. [47] define “knowledge about the user’s and IT device’s state, including surroundings, situation, and to a less extent, location.” Dey and Abowd [48] define “any information that can be used to characterize the situation of an entity. An entity can be a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves”. By combining low-level context, we can have higher-level of context explaining more deeply about users’ situation. While the authors in [46, 47, 48] define the term by synonyms, which lead to infinite loop reference, the authors in [45, 46] do by examples, which is lack the general. Zimmermann et al. [49] define context by extending the work of Dey and Abowd in [48]. They add “Elements for the description of this context information fall into five categories: individuality, activity, location, time and relations”.

Schilit et al. [45] classify context into three categories: (1) computing context includes connection status, bandwidth, communication cost, and IO devices (printer and display for example); (2) user context includes user information, location, social status, etc.; and (3) physical context includes light level, noise, heat, and so on. Chen et al. [46] add time context includes time, year, hour, minute, days in a week, and so on. Zimmermann et al. [49] classify context into five categories: (1) individuality includes information about a single or a group of entities, divided into four sub-categories: (a) natural entity describe natural environment (tree and rock for example); (b) human entity describes human properties (languages, skin color, and age for instance); (c) artificial entity is created upon interaction of human and natural entity (e.g., computer and house); and (d) group entity describes a group of entities that have common properties (Vietnamese, English, Japanese, etc.); (2) time and location; (4) activities describe goal, task, and action of an entity, (5) relations include three sub-categories: (a) social relations between two or more human entities (e.g., friends or colleagues); (b) functional relations between human entity and artificial entity (e.g., a cook with a knife); and (c) compositional relations describe the relationship between a part and a whole (a human entity and a group entity).

Schilit et al. [50] model context information by pairs of key-value, supporting pattern matching on context data. Brown et al. [51] and Pascoe et al. [52] employ SGML to model context into tags and values. SGML later becomes ConteXtML, a simple XML-based protocol to exchange context data between client and server. Chervest et al. [53] and Davies et al. [54] employ Object Oriented Model to model context as object and attributes of it. Bacon et al. [55] use a logical model in which context data are described as rules, and new rules can be added. Henricksen et al. [12] employ ER and UML in that static context data (hardly change, name, date of birth, languages) become attributes and dynamic context data are modeled as relationships between entities. Korpia et al. [13] employ RDF, and context data are categorized into types with values domain such as (Environment:Sound:LowEnergyRatio – (Low, High)), and high level context can be derived from low level context through bayesian network like (Environment:Sound:Type – (Car, Elevator, RockMusic)). Pinheiro et al. [16] employ OWL-S and introduce graph-based context matching method.

### 4. AN OPENLBS PRIVACY-AWARE MIDDLEWARE

As early introduced, the potential support of an elastic LBS middleware is very significant due to the fact that diversity from every aspect is flooding the world of LBS. Thus, what our proposed middleware mainly pursues are solutions for LBS heterogeneity, user privacy, and context awareness. In order to do that, the OPM consists of two parts working in pairs: Application Middleware and Location Middleware. The detail information will be discussed in the two next sections.
In LBS supply chain [25], the OPM plays a central component connecting other parties as illustrated in Figure 1. The activity mechanism is briefly described as follows. A user requests a location-based service (e.g., where is the nearest bus station from my position?) His request is then sent to the LBS provider. The LBS provider sends its request to the Application Middleware and asks about the answer. At this phase, there are three main scenarios related to whether user privacy is considered or not:

- **In case 1**, when there is no need to care about user privacy, the Application Middleware will ask the Location Middleware to get the user’s exact position from the location provider, process the request, contact content provider to get information about Point-of-Interests (POIs) and a suitable map, and then return the result to the LBS provider. The LBS provider, finally, returns the result to the user.

- **In case 2**, if user privacy is taken into account, the Location Middleware will get the exact position of the user from the location provider, make it obfuscated, and then return the cloaked region to the Application Middleware. Later, the Application Middleware has to process the cloaked region instead of a position, contact the content provider to get necessary information about POIs and map and return the result to the LBS provider. The LBS provider returns the result to the user as usual.

- **In case 3**, everything normally happens like the way it does in case 2. However, if expecting to have a better result, a user can get the result set filtered at the Location Middleware, for it knows exactly the precise position of the user.

**5. APPLICATION MIDDLEWARE**

**5.1 General Architecture**

The architecture in Figure 2 is designed for the purpose of supporting service providers as much as possible. It encompasses main components as followings: (1) Core services involve in five OpenLS core services; (2) Co-operator takes the responsibility of handling the interaction of core services; (3) Privacy-aware query processing takes charge of privacy-preserving queries (e.g., point or region processing); (4) Interaction layer makes itself responsible for getting into communication with external and internal actors such as content provider, location provider, service provider, database systems, and so on. Besides, OGC entities are exploited for request parsing; (5) Management services include some like authorization, authentication, validation, session management, catalog, remote services, billing, report, error handler, etc.; and (6) Navigator receives requests and delivers responses for either each core service independently or their convergence in Search Service. Among them, core services and privacy-aware query processing take the major role of the Application Middleware. Furthermore, each has its own specific characteristics and challenges, so they will be discussed more in the remainder of this section.

**Figure 2. The architecture of Application Middleware.**

It is worth mentioning that these core services will follow OpenGIS Location Services (OpenLS) standard, an open standard coming from Open Geospatial Consortium, Inc. ® [24]. Towards reusability and utility for developing LBA, it defines open interfaces and data models for LBA to easily access its core services without caring the heterogeneity at the lower levels. The latest OpenLS specification has been released with five core services as follows: (1) **Directory Service** performs search operation for objects such as places, products, or services; (2) **Gateway Service** is responsible for retrieving the location information of terminal devices; (3) **Geocoder Service**, also known as Location Utility Service, includes Geocode Service and Reverse Geocode Service that determine geographic position (e.g., address and zip code) and transform the position...
information into co-ordinates (i.e., latitude and longitude) and vice versa, respectively; (4) **Presentation Service** displays a map fulfilled with necessary information for visualization; and (5) **Route Service** adds utilities for path finding and navigation.

Each service can be deployed or revoked independently. Nevertheless, an example is given to clarify the interaction among these core services. A user may be interested in search services like “Show me the nearest hospital from my position.” In this situation, the position of the user is retrieved from Location Provider by Gateway Service. Next, Directory Service will execute the query in order to return the result set. Then the result is displayed on the user’s device by Presentation Service. If requested, Route Service can perform navigation or show the way to the hospital. Through the whole process, Geocoder Service plays an important role to support the functionality of geographic information transformation such as converting the position of the hospital into the address displayed on the map for more understanding to the user.

### 5.2 Core Services

To deal with the heterogeneity of LBS, the Application Middleware not only has fundamental functionalities of a middleware but also consists of five OpenLS core services. As introduced, they are gateway service, directory service, geocode and reverse geocode service, route service, and presentation service [24]. In the OPM, these services are also published as web services (e.g., RESTful web service) in an effort to get over multiple platforms. In addition, we adopt GeoNames and OpenStreetMap as sample POI data in our system.

**Gateway Service** is the component communicating with location providers through Mobile Location Protocol (MLP) [26]. The XML-based protocol is popular for telecommunication environment because it defines many ways allowing location information to be retrieved while being independent of underlying network technology. Actually, there are some candidate Location APIs [41] (e.g., WAP location framework, Parlay/OSA) that can be used to communicate with Location Provider. MLP is, however, more popular and recommended in OpenLS Gateway Service specification [24], so it is chosen in the OPM among the others. Our gateway service looks like Figure 3. The brief introduction is given as follows: (1) Network layer indicates the way (either HTTP or TCP connection) to get in touch with location providers (i.e., location servers); (2) MLP processing layer handles MLP-based messages; and (3) OpenLS processing layer treats OpenLS-based messages. We want our middleware to contact location providers by MLP and let the output be OpenLS. The problem, therefore, is the conversion between MLP and OpenLS structures. It is essential because MLP allows the positions of mobile terminals to be retrieved at the same time while OpenLS cannot. As a result, MLP Entity is the object containing all properties of basic MLP services (i.e., SLIS, ELIS, SLRS, ELRS, TLRS) [26] corresponding to results converted between MLP and OpenLS standards and vice versa.

**Directory Service**, whose architecture is showed in Figure 4, allows us to deploy two kind of search: (1) Attribute-based Search looks for POIs with their names or properties (e.g., address, telephone number, description, etc.); and (2) Location-based Search seeks POIs in correlation between them and the pre-specified position (e.g., nearest, within a distance, within a boundary). The former refers to exact match while the latter can lead to similarity search and the collaboration from other services. Directory Service is marked as one of the essential parts to facilitate LBA. Nonetheless, its main obstacles originate from POI data. Because there is no standard for the schema of POI database, the implementation of this service has to cope with the variety of data models and the incompleteness of data sets. These problems are not so critical but are able to affect the quality of results returned by searching operations.

**Route Service**, adds utilities for path finding and navigation. The service can perform navigation or show the way to the hospital. Through the whole process, Geocoder Service plays an important role to support the functionality of geographic information transformation such as converting the position of the hospital into the address displayed on the map for more understanding to the user.
**GeoCode Service** and **Reverse Geocode Service** always stay side-by-side as twins. One changes address-related input into geological co-ordinates (i.e., latitude and longitude) while the other transforms data in the reverse direction. The architecture mentioned in Figure 5 seems to be similar to that of directory service. Real challenges, however, are met when this location utility service is taken into account like multiple reference system, the local way of presenting an address, incomplete data input, abbreviation problems, and the quality and granularity of data sets. Besides, similarity search is employed in case no exact result can be found. Although they are analyzed quite clearly in [23], together with recommended methods, there are still big barriers ahead in terms of implementation.

**Route Service**, illustrated in Figure 7, puts its heavy weight on Route Algorithm (fastest, shortest, and easiest route for example), Route Profile (one way, highway, and under-construction for instance), and User Constraint (like avoid narrow roads if possible). Simply finding a route between two points, however, is not good enough for LBS users and the utility LBS can offer. Similar to the mechanism of protecting user privacy that will be introduced in later sections, route service should be context-aware as much as possible.

### 5.3 Privacy-aware Query Processor

In order to protect user privacy, upon the untrustworthy Application Middleware’s request, the Location Middleware will encrypt, transform, anonymize or obfuscate the location information (e.g., a processed region or obfuscated region) before forwarding it to the Application Middleware. Thus, to process that perturbed location information, the Application Middleware must have privacy awareness (e.g., process the services using the processed region, not the exact location). In addition to reinforcing the high level of user privacy, the privacy-aware query processor must support four types of query [2] as listed below:

- **Public query over public data** indicates queries for exact location information with exact location information. In case user privacy is ignored, a user does not need to hide his location information. For example, a doctor does not need to hide the location where he is in a hospital and is searching for a restaurant for lunch.

- **Private query over public data** shows queries for exact location information with obfuscated location information. For instance, in tourism services, users’ location is obfuscated while POI data’s are not.

- **Public query over private data** specifies queries for obfuscated location information with exact location information. This kind of query is similar to public query over public data with the doctor example, but the query is to search for patients’ location.

- **Private query over private data** involves queries for obfuscated location information with obfuscated location information. In friend finder service, let us say, the location information of both users and their friends is obfuscated.

In these circumstances, public query or data refers objects or their information that everyone knows or are willing to be revealed (e.g., the position of POIs, the position of patrol police car, ambulance or public services) while private query or data mentions those whose access is limited (e.g., user exact position or private places). To process queries with blurred information, we can employ Casper [2] or other query processors [4,6] that can deal with obfuscated location information as input and data or combine many methods to support the heterogeneity of obfuscated regions (both input and data). In OPM, the Application Middleware is easily extensible to plug the query processor into core directory service. Furthermore, the query processors’ configuration can also be adapted to each user or to the whole system like the work in [2] to provide balance between optimality and response time. Last but not least, privacy-aware query processor is designed to be the place where privacy-aware query techniques converge in case new ones other than existing approaches are proposed.
6. LOCATION MIDDLEWARE

6.1 General Architecture

To protect LBS users’ privacy, various privacy techniques locating at the Location Middleware can be employed. In terms of location privacy, user location will be encrypted, transformed, anonymized, or obfuscated via supported privacy techniques before forwarding to the Application Middleware. However, it would be a burden for users to select the privacy techniques suitable for their situations because the privacy techniques are hard to understand and master under the view of users. To hide the complexity and remove the burden of selecting or combining suitable privacy techniques in LBS privacy preservation, our Location Middleware also offers trio-privacy components that work well with each other. They are User Management, Privacy Technique Agent, and Context-aware Privacy Broker. Their further information will be given in the remainder of this section.

The general view in Figure 8 shows main components of which the Location Middleware consists: (1) User Management manages user pseudonyms, profile, and context. All user information will be stored in User Profile; (2) Privacy Technique Agent (PTA) is the place where privacy techniques are integrated. All privacy technique information will be stored in PTA profile; (3) Context-aware Privacy Broker is responsible for which privacy techniques should be applied under consideration of the current context; (4) Filter Agent refines result set if required; (5) Access Control is the place where access control models (e.g., discretionary or mandatory access control) are deployed in order to control data flows of sensitive or personal information exposure; and (6) Positioning Layer is in charge of hiding the complexity of various positioning techniques. It has a part in improving the heterogeneity of LBS.

User Profile stores LBS users’ privacy policy or privacy preferences and dynamic user context. It includes static information that a user declares upon subscribing to the Location Middleware’s location services, and multiple privacy requirements consist of when, where, to whom, why, how, at what extent the user’s location information will be transferred to a requester. It is managed by User Profile Manager, which also responsible for the activation and deactivation of the user profile by specified constraints. The privacy requirements also conform to the constraint categories classified in [8]: (1) When (time constraint) specifies the time when the profile will be activated (e.g., from 8 AM to 10 AM, from 6 PM to 9 PM), and a user can have different requirement for the privacy; (2) Where (location constraint) specifies the location where the profile will be activated (e.g., at company, at bar, at hospital), and a user can have different requirement for the privacy; (3) To whom (action constraint) specifies the requesters corresponding to the profile. For example, colleagues, supervisors, and relatives may be treated with different manner upon requesting a user’s location information; (4) Why (service constraint) specifies the purposes the requester presents so as to acquire user’s location information; For example, company tracking or friend finder; (5) How (notification constraint) specifies if user needs to explicitly accept or deny the location sharing; and (6) At what extent (accuracy constraint) specifies the privacy level a user wants. This can be expressed in a systematic or a more understandable way. For instance, systematic parameters such as k, l, m, n, A_min, A_max, etc. or simply high, medium, low, level 1, level 2, etc. which wrap the systematic parameters for users.

Static information of a user are additional information used to enrich the context of a user like age, nationality, speaking language, etc. while they are hardly change over time [12]. In addition, a user can also add some dynamic information [12] in order to enrich even more the context of that user such as activities, relations, and so on. Those are changeable over time and are collectable and derivable through some other lower level context data [14]. As an example, activities can be derived from smartphone sensors, or a user’s location can be enriched from some context such as he is in a hospital, or he may be a patient, a doctor, or a relative.

6.2 User Management

User Management component, illustrated in Figure 8, includes three sub-components that help manage the User Pseudonym, Profile, and Context as follows: (1) Pseudonym Manager manages user identity-related protection. In LBS, some services like “Find the nearest gas station” do not need to know the true identity of a user (i.e., who the user is). So a pseudonym or a false identity is used instead to protect the true one and prevent the leak of sensitive information form user privacy breaches; (2) User Profile Manager maintains the static context data of a user (e.g., age, nationality, speaking languages) and several privacy requirements that the user declares upon subscribing to the Location Middleware’s services. These data can be changed anytime by the user’s will; and (3) User Context Manager is regularly or on demand updated with latest user dynamic context data. These data are collected though mobile device sensors or derived from those low level context data to form new higher level context data. The User Context Manager can also be integrated directly into the user’s mobile device to have best context data updated to the Location Middleware.

Figure 8. The architecture of Location Middleware.

6.3 Privacy Technique Agent

Privacy Technique Agent, shown in Figure 8, includes two sub-components that help manage the PTA Profile and Context as followings: (1) Privacy Technique Profile Manager maintains the
static context data of a technique agent. The functional and optional input, output, and requirements, similar to a user’s static context data, are hardly changed over time, but the technique agent can also change the information as it wants; and (2) Privacy Technique Context Manager is also regularly, or on demand updated with latest technique agent context data, which are collected or derived and sent to Location Middleware. The same as User Context Manager, Privacy Technique Context Manager can also be integrated into the technique agent to have best context data updated to the Location Middleware.

Privacy Technique Agent Profile (Technique Profile for short) includes functional and optional input, output, and requirements of privacy techniques, which are available to the Location Middleware. Similar to the User Profile, the Technique Profile has static and dynamic information or dynamic context data [12]. It is updated regularly and managed by Privacy Technique Profile Manager, which knows the current load of the PTA’s executor or the availability of privacy techniques for example.

The functional part of Technique Profile shows hard constraints that the current context of a LBS user must match in order to use privacy techniques for privacy preservation. For example, the privacy technique named Interval Cloak [10] supports the input k, delta time, delta distance to work if the current context or the current activated profile of the user lacks the k or either delta time or delta distance doesn’t satisfy the PTA’s requirements. The Technique Filter will skip the Interval Cloak and select another privacy technique for the user. Another example is the situation that the requester of the location information requires the identity of the user to process the service (e.g., a toll bill service), the PTA must support location anonymity instead of identity anonymity to be selected for such situation. The optional part of Technique Profile describes soft constraints or the best circumstances in which privacy techniques are suitably used. Let us say, in case the current location of a user is at a dense area, the technique X, whose characteristic requires the density of user population to work best (e.g., k-anonymity techniques [10]), should be chosen among others, for its type fits well in the situation.

6.4 Context-aware Privacy Broker

The architecture of Location Middleware in Figure 8 also shows a general view of how Context-aware Privacy Broker works with two main sub-components as follows: (1) Technique Filter combines the Profile Manager’s and Context Manager’s information in order to select or combine technique agents and then creates the best execution plan for current user and technique agents’ context; and (2) Execution Engine will execute the execution plan to generate the privacy preserved location information. The Technique Filter takes the input as current user context, current activated user privacy requirement, and current PTA’s context to automatically select or combine privacy techniques for user privacy protection. Later, the execution plan is forwarded to Execution Engine to generate the privacy preserved location information. Similar to MUSIC [16,18], we employ OWL-S but with minor modification to model User Profile and Privacy Technique Profile. The alteration is to change and add the suitable parameters as the six-tuple <when, where, how, to whom, why, at what extent> for privacy requirements because MUSIC context model does not define privacy-related context-awareness. Then we use graph-based methods [18] to select or combine the best algorithms matching the user’s current context. The PTA ideally will be the native agent of the Location Middleware, but also can be other trustworthy technique agents provided by other trustworthy parties. Because the message exchanges between the components, user, and technique agents are OWL-S, technique agents can be web services provided by another trustworthy source. The same thing happens to the Context Managers and Profile Managers (e.g., users’ mobile devices).

7. DISCUSSION AND FUTURE WORK

The OPM is an open design for further LBS integration and LBA development. Its two constituents, Application Middleware and Location Middleware, play an extremely significant role thanks to their mutual support. The former directly assists LBS providers in doing their business. It hides many variants of heterogeneity, reduces overheads in developing LBA, and promotes building up LBS. Besides, complying with LBS standards makes the best use of communication and interaction in pervasive computing. The latter takes care of the discrepancy of positioning techniques. Its characteristic is very important due to the fact that it assures the quality of communication when end-users enjoy LBS. Furthermore, that the Application Middleware and Location Middleware live in harmony with each other has an additional benefit. It can be seen as the pleasure of their clients (i.e., LBS providers) and the satisfaction of their clients’ clients (i.e., LBS users). If the assistance is preserved and consolidated with every passing day, more competitive profits could be found whereas people’s living standards never stop improving.

In terms of user privacy protection, the Application Middleware is flexible to employ multiform privacy-aware query processors processing the input region. The variety of privacy techniques is a part of the heterogeneity problem. Various region shapes, for instance, may appear from the Location Middleware or any privacy components. Thus the Application Middleware must have its privacy-aware processor dealt with many kinds of privacy techniques to strongly support the heterogeneity of privacy protection. The Location Middleware, alternatively, is elastic to be a source of various privacy-preserving algorithms stored in PTA, and it can also adapt various context modeling and matching methods to the User and PTA Profile as well. Besides the conventional privacy-preserving algorithms that operate separately from the database level as shown above, the integration of lately database-centric approaches [37,38,39,40] is also feasible in the OPM. These methods are considered to make great progress over two-stage ones: (1) retrieving the user exact position; and (2) making it blurred by another component afterwards. Hence, the integration of such techniques can greatly reduce the overhead of service processing in Location Middleware, reach the high-level of user privacy protection, and speed up the performance overall, which leads to better scalability of the OPM.

On the other hand, there are also other challenges in developing the Context Aware Service Broker because the User Profile may contain negatively correlative rules or contradictory constraints specified in privacy requirements. For example, the six-tuple <when, where, how, to whom, why, at what extent> has two conflict instances as follows. At first, a user sets up his privacy preference (a) < << 8AM-8PM>, <Mon-Fri>>, Ho Chi Minh City, explicit, work, all, public>. It indicates that in working time on weekdays, the user does not need any privacy related to work. Later, he adds another privacy requirement (b) << 8AM-8PM>, <Mon-Sat>>, Ho Chi Minh City, explicit, all, work, private> because he is taking a vacation. At the same time, the two privacy requirements mentioned above can be activated. Another example
The OPM, however, still suffers drawbacks a normal middleware has. As the role of being a central actor, the middleware easily meets the worrying problem known as bottle-neck. If it wants to serve as more LBS providers as possible, it has to care about multi-transaction processing capability, especially in the real time environment. In addition, quality of services and user privacy protection are mutually exclusive. The tight protection can lead to poor quality of services and vice versa. Therefore, the balance needs necessarily to be found, and design of experiments of the middleware should be given for its performance evaluation. On the contrary, the middleware could be a main point for malicious attacks, from security to privacy perspective. All of these issues, from the OPM in general to components in particular, will be considered as our future work in an effort of meeting the widespread of LBS.

8. CONCLUSION

We propose the OPM, in this paper, as an alternative solution for LBA development. This can result in helping LBS providers resolve the variety of LBS actors and influential factors, user privacy protection, and context adaptability. When compared to other platforms in general and GeoMobility Server [24] in particular, our work is very different. The OPM strengthens privacy protection capability from two vital aspects that support more privacy-aware query processing algorithms at Application Middleware and enable many modern privacy techniques as well as those which enhance context awareness to be integrated into Location Middleware. The flexible association makes itself mostly possible to reinforce privacy protection approaches of all kinds. In addition, we have addressed some obstacles and challenges the middleware may encounter in such ubiquitous environment. As discussed above, LBS-related issues that have been emerging in our middleware do not stop here. There are, actually, still lots of open directions ahead as our future work to perfect the middleware and achieve its accomplishment.

9. ACKNOWLEDGMENTS

The research work is supported by Vietnam National University Ho Chi Minh City under the project number B2010-20-02TD.

10. REFERENCES


