PLBSD: a platform for proactive location-based service discovery
Evangelos Niforatos*, Evangelos Karapanos and Spyros Sioutas

*Madeira Interactive Technologies Institute, Universidade da Madeira, Funchal 9020-105, Portugal; bDepartment of Informatics, Ionian University, Corfu 49100, Greece
(Received 26 April 2011; final version received 15 April 2012; accepted 1 May 2012)

We introduce a platform for the rapid prototyping of proactive location-based service discovery, proactive location-based services are conceptualised along three broad categories: location-triggered services, chain-triggered services, proximity-triggered services, and illustrated through a number of usage scenarios. We report on a workshop with designers and researchers in the area of location-based services that resulted in a set of initial requirements for the platform. We describe how the platform aims at addressing these requirements, and illustrate the implemented features through the development of a proactive location-based application.

Keywords: proactive; location-based services; discovery; application prototyping; platform

1. Introduction
Location Based Services (LBS) rely on the knowledge about user’s geographical location, in order to offer a plethora of options, regarding the provision of nearby services (Barnes 2003, Tilson et al. 2004). Therefore, the fundamental emerging attribute is the information that depicts the user’s geographical location. Such information is utilised by LBS discovery applications, prompting user about nearby available services. An example of LBS discovery application would be a person shopping in a town center, requesting the location of the shop that offers the best discounts on products of a specific brand, within her reach. A database storing data about 3000 shops in a city of 450 Km$^2$, should be able to provide the necessary information, according to proximity and other criteria set (Matapurkar 2006). More specifically, a user of an ordinary LBS discovery application, such as Google Maps, is expected at first to specify the type of the desired service. Thereafter, the application can either demonstrate the closest service, fulfilling user criteria or even display a proximity alert, when user is in a certain range from a service.

The aforementioned paradigm resembles a reactive service discovery process, based on user-specified discovery criteria. Quite a contrary approach is the proactive service discovery process (PSDP), introducing the feature of discovering a service before it is requested. Despite proactive LBS (PLBS) demand much less user attention and interaction, the design and implementation of PLBS prove more difficult, because the services must continuously track their target and evaluate...
location events (Bellavista et al. 2008). A wide range of LBS application development platforms and patents (Honisch 2009, Weinrich et al. 2009) has been introduced lately. Such platforms include PoLos platform, that focuses on the development, deployment and provision of services and consolidates several technologies of LBS world under one framework (Ioannidis et al. 2005). A different approach is presented in Blunck et al. (2010), introducing the PerPos platform for cloud services and pervasive positioning. PerPos platform mainly involves indoor navigation, providing an API for the development of positioning utilities and location-based, special purpose applications. Indooria platform comprises a similar approach for indoor LBS, introducing the term of proactive service discovery (Ruppel et al. 2008). An evaluation of available technologies and approaches for indoor location positioning is presented in (Curran et al. 2011).

What we propose, is a platform (PLBSD) for developing LBSD applications that implements the feature of proactive service discovery, as its main axon, supporting rapid application prototyping. PLBSD offers the potential of differentiating from other LBS discovery applications in the way service discovery is performed. From this perspective, a PLBSD application user does not need to specify a desired kind of service to be discovered, but instead run the application and then leave it to operate in the background. As soon as the application is initiated, it starts informing user for all available services in a close proximity (Küpper and Treu 2006), by displaying an activity chain of events, enabling user to specify the activity he/she is engaged with. Proactive service discovery proves to be especially suitable for the provision of strongly location and time-dependent services (Bellavista et al. 2008), including also information and advertisement services, whereas conceptualisation issues and further concerns are discussed in this article.

In the next section, the requirements for PLBSD platform are identified, during a workshop with LBS application designers. Section 3 includes a formalisation of varying LBS and presents the PLBSD platform, describing the implemented processes, as well as the built-in GUI. Section 4 introduces a customised commercial LBS application, developed on PLBSD platform, as well as a use case scenario. Conclusion and future work are presented in the last section.

2. Requirements for PLBSD

The PLBSD platform concept emerged during informal interviews with designers of LBS that revealed a need for rapid prototyping and the diversity of possible LBS. To elicit an initial set of requirements for the PLBSD platform, we organised a workshop with two interaction designers with expertise in LBS, one computer scientist with expertise in LBS and one Human–Computer Interaction researcher with expertise in the evaluation of LBS. The workshop lasted for 2h and was structured in three phases. First, we introduced our vision on PLBSD discovery to all participants. We purposefully defined proactive LBSD broadly as ‘LBSD’ in order to increase the likelihood of eliciting diverse scenarios of LBS. Secondly, in a brainstorming session, we attempted to derive scenarios of LBS that involve proactive service discovery. In a third step, we used these scenarios to conceptualise the design space of PLBSD and elicit requirements for the development of the
PLBSD platform. We summarise the results of our workshop in the following two sections.

2.1. Scenarios of LBS

In the section below we describe the scenarios of location-based services that emerged during the LBS workshop, grouped under three main clusters: location-triggered services, chain-triggered services and proximity-triggered services. These scenarios guided us in the formalisation and requirements elicitation of the PLBSD platform:

2.1.1. Location-triggered services

Most of the concepts that emerged during our workshop related to the traditional tourism application model (Hawking et al. 2005), involving a mobile user equipped with a mobile device, running the tourism application. Subsequently, the mobile user will be informed about the most important sightseeings, interesting places for visiting and available services. Advertisement LBS also belong to this category, utilising a mobile user’s current geographical position to provide targeted and location-dependent advertisement about nearby services (Aalto et al. 2004), whereas additional information is available via the web.

An application in the area of commerce. PLBSD platform can be used for the development of commercial applications using the proactive service discovery feature for informing about nearby commercial services. Such a common scenario involves a mobile user shopping in a city center, equipped with a mobile device, running the commercial application. Subsequently, the main application objective is the proactive discovery of provided service in the proximity of the roaming user. A service in this case could be, for example, an offer or sale on some specific brand items like shoes and clothes, or the sale of ‘distressed’ items, such as last minute tickets to a cinema movie which is about to start. Proactive feature posits that a user is not required to specify the kind of the service to be discovered and therefore, it is suitable for the provision of strongly location and time-dependent services. As soon as a service is discovered, a mobile user can be informed about its location and obtain additional information. In the case of a service with a digital content, a LBS application will utilise the wireless network technologies to provide the service discovered.

2.1.2. Chain-triggered services

Another aspect of a LBS application is the potential of ordering the service discovery process according to a context-aware framework (Cheverest et al. 2000). Such a LBS application will navigate a mobile user to gradually discover a series of context-alike interconnected services (service chain), providing a new level of location-aware experience. The context-aware LBS could also be used in the area of tourism, navigating, for example, a mobile user in the unveiling of historical consequent facts, offering an entertaining and educational experience. The actual content of such services varies from interactive maps, to multimedia and others.
A context-aware application in the area of tourism. The largest group of LBS applications is in a field known as ‘mobile guides’. A mobile guide can be defined as a portable, location-sensitive and information-rich digital guide to the user’s surroundings (Raper et al. 2007). Particularly, in the area of tourism, a LBS application could monitor the current location of a mobile user and proactively notify her accordingly, when she approaches a sightseeing, while taking into account her prior interaction experience (e.g. Karapanos et al. 2009). Moreover, a tourist guide LBS application could also navigate a mobile user into the exploration of a service chain, providing context-sensitive information, whereas a combination of dynamic information and interactive services support, offer a new user experience (Cheverest et al. 2000). iLand comprises an example of a LBS application that offers a narrative experience, combining storytelling and the information of the surrounding context (Dionisio et al. 2010).

2.1.3. Proximity-triggered services

Proximity-triggered services are available when a mobile user crosses a predefined perimeter and are further divided to synchronous and asynchronous services. Synchronous services involve a service creation and subsequent service discoveries, until that service is no longer available. Asynchronous services, on the other hand, are created once and they remain available through subsequent service discoveries, introducing the service migration concept. Two application scenarios, for synchronous and asynchronous proximity-triggered services, are presented respectively.

Synchronous services. A mobile application that grants mobile users the ability of locally creating a service and acting as service providers. Potential buyers with the application running on their mobile devices, in the proximity (e.g. 200 m) of the service providers, are informed about the provided service by automatic updates, notifying them about the kind of the available service and also indicating the current location of the service provider. In case a potential buyer is interested, the application may provide a chat initialisation with the service provider for obtaining additional information or even bargain the price. At this point the feature of location-based, on the fly created auction emerges, introducing the so-called location aware eBay.

Possible integrations with social networking and auctioning platforms, such as Facebook, Foursquare and eBay, could add the trust feeling (Chow et al. 2010), introducing a rating system and maybe keep track of what is being purchased. This application scenario is currently in development stage.

Asynchronous services. A mobile application intended for research purposes, involving large-scale simulations in real conditions. The simulation application will grant the user the potential of joining a research network, participating in large-scale simulations and experiments, producing valuable feedback on several aspects. Subsequently, this application could be used for simulating disease spread-out in real conditions, studying the way diseases expand in urban areas. More specifically, a mobile device with the simulation application installed, periodically broadcasts flooding messages (e.g. infection messages) under predefined conditions (probability, range, rate etc.). Mobile devices in the proximity receive these messages and become
‘infected’, whereas they continue broadcasting infection messages accordingly. This application scenario is currently in designing stage.

2.2. Requirements

Using the scenarios outline above, we attempted to derive a number of requirements for the PLBSD framework, which are described below.

**Service discovery.** It can be either proactive or reactive. During proactive service discovery, a mobile user is only required to cross a service proximity, whereas during reactive service discovery, a user is first required to set a range of discovery criteria. These criteria can define a specific kind of a service to be discovered, such as the discovery range scope.

**Location.** It can be either fixed or dynamic and describes the geographical position of a service or a person. As fixed location is considered the location whose coordinates remain constant over time, whereas as dynamic, the location whose coordinates alter through time.

**Time.** The term of time may refer to a time period or simply duration. More specifically, a service can be available for a specific time period, e.g. ‘Asian Arts Museum is open everyday from 9:00 to 18:00’, or a predefined amount of time, e.g. ‘Last minute tickets for the Black Swan movie’. Both cases should be supported during the service discovery and provision.

**Service chain.** It is a set of services that are linked through one or more common attribute. This attribute can relate to the common context of the set, e.g. a temporal ordering of historical events along different locations of a city that the tourist is guided through, or a set of services in terms of a total experience gained, e.g. a set of must-see places in a city. In the first case, a specific order of discovering the available services is considered crucial for the overall experience provided, whereas in the second case it is not. A service chain may contain other parameters such as an estimated time for transition across the locations in the set, or the overall estimated duration of the journey. A service chain may be triggered after having visited a variable fraction of the locations in the set.

**Output.** A LBS application’s output strongly depends on the kind and the content of the services discovered. Some services minister only informational aims, such as advertising services, whereas others provide digital content, such as multimedia services.

**Service filtering and prioritisation.** Apparently, service discovery and provision should be somehow filtered and prioritised. This may be performed on parameters such as time, location, context and preferences. Time parameter defines the duration or period that a service will be discoverable, whereas, location parameter defines which service will be discovered and which will not, based upon coordinates and range/proximity criteria. Moreover, service context, combined with user’s preferences (profiling), specify which services will be discovered and presented and which services will be excluded from the discovery scope. Service filtering aim is the reduction of the spamming effect that proactive service discovery feature strongly
incurs, by introducing a context-aware middleware to automatically filter out the LBS content that end user perceives as spam (Bellavista et al. 2008).

3. LBS application design

PLBSD platform offers a number of functionalities, including processes and functions, to the designers and developers of LBS applications. Before proceeding to the technical description, it is necessary to set the PLBSD framework by defining the logic behind it. The PLBSD framework approaches LBS from the three main perspectives of a fixed service, a dynamic service and a service chain.

**Fixed service S(L, t).** It is defined as a circle with a known range ($R$) and centre ($x, y$). Hence, a service’s location ($L$) is comprised of a pair of latitude ($x$) and longitude ($y$) values, combined with the range ($R$), the service is available in. Therefore, a service is defined by its location $L(x, y, R)$ and the time ($t$) that is available for, in terms of duration or period (Figure 1).

**Dynamic service S(P(L), t).** It is similar to a fixed service $S(L, t)$, differentiating in the fact that it is created and provided by a person ($P$). Hence, the location of a dynamic service strongly depends on the location of the person $P(L)$ that provides it. Since a person providing a service can move around, the location of the provided service constantly changes accordingly. Moreover, a dynamic service is provided during a predefined time period ($t$), as shown in Figure 2.

![Figure 1. A fixed service.](image1)

![Figure 2. A dynamic service.](image2)
Service chain $S(ch(L), t)$. It is defined as a sequence of provided services, with a variance in the location and the time, different services are available. The fundamental attribute in this case, is considered to be the service sequence. A service sequence defines the order in which services attached to the same chain will be discovered and presented. Thus, a service chain, $S(ch(L), t)$ is considered as a set of ordered services that are grouped according to a common attribute or feature (Figure 3).

3.1. PLBSD platform overview

PLBSD comprises a platform for the development of LBS discovery applications, that provides a set of available functions and processes, relying on a network architecture that involves satellite positioning, cellular network communication and Internet connection (Dao et al. 2002). Hence, an application based on PLBSD platform utilises a set of wireless technologies (Rao and Minakakis 2003). More specifically, PLBSD retrieves a satellite signal and extracts current geographical location. Also, PLBSD uses cellular network to synchronise the local SQLite database with the remote XML database server (Virrantaus et al. 2006). Finally, PLBSD combines user’s current geographical position and the data stored in the local SQLite database, to proactively discover and inform about, or provide, nearby LBS.

3.1.1. Platform requirements

PLBSD smooth operation depends on specific hardware and software requirements that a mobile device should adhere to.

GPS receiver and (or) aGPS system. A mobile device running the PLBSD platform must be supplied with a fully functional GPS or aGPS system, producing constant geographical location updates. The combination of a standard GPS receiver with an aGPS receiver provides maximum efficiency, enhancing the location information retrieval (Karunayake et al. 2004). In case current geographical location is fully provided by cellular network infrastructure, using an aGPS system, GPS signal is not utilised.
**3G or GPRS connection.** Cellular network provides the main data channel, connecting PLBSD application to the Internet. Internet connection is necessary for updating application’s database with current service information and displaying the location map. Internet connection is also desirable during service discovery, for displaying a relative service image of the service just discovered. Updating can only be triggered by the user, eliminating the risk of overcharging by automatic updates. Therefore, 3G standard is most desirable for its increased data transfer speed, that reduces update and image download time (Ficco et al. 2010). In case 3G is not available by the cellular network in specific remote areas, PLBSD can fall back to GPRS standard. Both 3G and GPRS involve charge for the volume of data transferred. Optionally, a device supplied with Wi-Fi transmitter is capable of connecting to the Internet via known or open Wi-Fi networks and update for free.

**Android 2.2 OS.** PLBSD is developed on Android platform version 2.2 API level 8, hence, it is explicitly intended for LBS application development on this platform or any higher version.

### 3.1.2. PLBSD processes and functions

This subsection describes some major processes, that PLBSD offers to LBS application developers, disengaged from whatever GUI implementations. Although, PLBSD provides a fully functional GUI that it will be described in the forthcoming sections of this article. All processes described below should be initiated by the user.

**Update process.** PLBSD platform uses a local internal SQLite database for storing the available services to be discovered. The local database update relies on a remote back-end software module for downloading all the necessary data, concerning the provided services. During the update, the services XML file is downloaded via the Internet and works as an input to a parsing algorithm, known as AndroidSAX, running locally. AndroidSAX algorithm reads the services XML file, parses it and stores the extracted data into the local SQLite database, in the form of available services (Figure 4). The services XML schema is structured as shown in the example below:

```xml
<?xml version="1.0"?>
<services>
  ...
  <service id="10">
    <name>Anadia Shopping Center</name>
    <s_id>10</s_id>
    <description>Special offer on red wine only for today!</description>
    <category>Shopping</category>
    <url>http://www.madeira-web.com/PagesUK/shop-uk.html</url>
    <img>anadia.jpg</img>
    <radius>100</radius>
    <expiration>—1</expiration>
    <lat>32.64935234514145</lat>
    <lng>—16.90433862953936</lng>
  </service>
  ...
<services>
```
**Proactive service discovery process.** During the time PSDP is active, a service discovery module is initiated and starts searching for available services in the user’s proximity. Service discovery module combines constant geographical location updates and the service data retrieved from the local SQLite database, to create proximity alerts. A proximity alert is only triggered when a user crosses the predefined perimeter of a nearby available service. As soon as a proximity alert is triggered, service discovery module initiates the display process described below. In case update process is executed during the time PSDP is active, then for consistency reasons, PSDP is internally restarted. PSDP is service centralised.

**Spatial query process.** PLBSD platform implements the feature of spatial querying, allowing the discovery of all available services in a predefined range. More specifically, spatial query process combines the current geographical location and a user defined discovery range, in order to extract a spatial criterion. The spatial criterion is afterwards applied on every service in the local SQLite database. Those services, fulfilling the spatial criterion, are selected as an input to the display process described below. Spatial query process is user centralised.
**Display process.** The process of displaying presents the outcome of the PSDP and spatial query processes described above, in terms of an alert dialog providing some information and options, or a map with some relative overlays, respectively. Display process is invoked by PSDP and spatial query processes, therefore, it is indirectly initiated by the user.

### 3.1.3. Built-in GUI

PLBSD platform is equipped with a built-in GUI divided in three tabs, Monitor, Services and Extra tab. Each tab offers a different set of functionalities.

**Monitor tab.** It displays the user’s current geographical location in terms of latitude and longitude values, a list of location providers, the current location is extracted from and the status of the PSDP. In addition, a location changes counter and a corresponding reset button serve experimental and research purposes (Figure 5a).

Additional options are accessed, simply by pressing the Menu keyboard button causing an emerging menu to appear at the bottom of the screen. The emerging menu offers a range of new options, including service discovery start and stop, exit console and exit application. By touching the ‘Start Searching’ button, user can launch the PSDP, that searches for available services in user’s proximity. Accordingly, by touching the ‘Stop Searching’ button, PSDP is halted. The ‘Exit Console’ button allows user to exit PLBSD application GUI and proceed to other operations, whereas service discovery can run in the background providing service discovery for the user.

![Monitor tab](image1)

**Services tab.**

![Services tab](image2)

Figure 5. This figure demonstrates the two first tabs PLBSD platform GUI is comprised of and the corresponding emerging menus, with a different range of available options for each tab. (a) Monitor tab and (b) services tab.
discovery updates. The ‘Exit Application’ button is used for exiting and terminating a PLBSD application completely, including PSDP.

Alternative and faster ways of exiting are provided by keyboard buttons, with Back keyboard button functioning as ‘Exit Console’ button and Home button as a suspending button, preventing PSDP from displaying alert dialogs, as described below. In all cases of exiting, user can relaunch PLBSD application simply by touching on its corresponding icon, or by pressing and holding the Home button, selecting again the application icon.

**Services tab.** It can be accessed simply by touching the corresponding tab at the top of the screen. Services tab offers a set of operations concerning the local PLBSD database management and the services available for discovery. It is worth mentioning, that only services stored locally can be discovered, therefore services tab provides the necessary operation for keeping a PLBSD application up to date. When user accesses the service tab, a list of all available services is presented, featuring the service name and the category it belongs to (Figure 5b). User can be informed about a service in the database simply by touching on it and a new screen appears. The new screen with the ‘Edit Service’ title, displays service information placed in modifying fields, allowing the modification of service attributes, by also providing a corresponding ‘Save and Exit’, at the bottom of the screen. In addition, side touching scrolling bars provide the necessary scrolling function over the service attributes. In case a service is not desired, user can simply delete it by continuously touching on it and subsequently, touch on the delete button appearing. Eventually, deleted services are no longer discoverable.

Lastly, additional options on Services tab can be accessed by pressing the Menu keyboard button and an emerging menu appears at the bottom of the screen, including ‘Add New Service’ and ‘Update Services’ options (Figure 5b). Therefore, the user can easily update the service list contents by simply touching the ‘Update Services’ button and watch the list being populated. The ‘Add New Service’ touch button serve experimental and research purposes, concerning the function of manually adding a service. More specifically, via the option of adding a new service, a user can create and store a new service in the local database by completing the fields of a form appearing in ‘Add New Service’ screen. Hints appear inside the fields of the service form, indicating the current service attribute, enhancing user input. A side scrolling bar, only visible when screen is touched, enables scrolling. Similarly to ‘Service Edit’ screen described above, at the bottom of ‘Add New Service’ screen, a ‘Save and Exit’ button stores and adds the new services to the service list. Alternatively, the Back keyboard button can be pressed, offering the same results while exiting the ‘Add New Service’ screen.

**Extra tab.** It is the third tab of PLBSD platform and can be accessed simply by touching the corresponding tab at the top of the screen (Figure 6a). Extra tab provides the function of detecting all nearby services in a user defined range. Therefore, it includes a field for user input and a hint, allowing user to enter the desired discovery range in meters. At the bottom of the screen, an emerging menu appears, after pressing the Menu keyboard button, for discovering all the available services in the defined range. It is worth mentioning that user input field provides data input control by accepting only positive numeric values, with a default value of 1000 m and limiting the discovery range at a maximum of 10 km.
As soon as the ‘Display all services in range’ button is touched, a map view screen appears, demonstrating the user’s position and all available services inside the user-defined range, in the form of touchable markers (Figure 6b). At the top of the map view, discovery scope is shown, whereas at the bottom, built-in zoom controls appear when user touches the screen, enabling zoom-in and zoom-out functions. Initial zoom level is automatically adjusted, so that the map view screen includes all services discovered in the specified range. When user touches on a marker, an alert dialog appears in the center of the screen, providing information about the service name and a short service description. The user can exit the alert dialog simply by pressing the Back keyboard button. Additional options can be accessed by pressing the Menu keyboard button, via the emerging menu at the bottom of the screen. More specifically, emerging menu includes touch buttons for changing among ‘Normal’, ‘Satellite’ and ‘Street’ views and a ‘Refresh’ button, that renews the map view screen when touched, according to new location updates. Lastly, the user can exit the map view simply by pressing the Back keyboard button.

4. A commercial LBS case study

In this section, a customised commercial LBS application is presented, based on the PLBSD platform and a use case scenario is described. The commercial LBS application utilises the PSDP feature, that runs in the background, being responsible for the discovery of nearby available services and prompting user by triggering the display of service discovery alert dialogs accordingly. A service discovery alert dialog, displays a relative service image and some information about the service.
discovered, regarding its name, description and distance from the user (Figure 7a). Also, three buttons at the bottom of a service discovery alert dialog offer additional options. By touching on ‘Go Online’ button, a user can visit the website of the service just discovered and access what this service provides. ‘Show Map’ button displays a map view screen similar to the one described above, including the same features but displaying only the current user position and the position of the service just discovered. When the ‘Exit’ button is touched or the Back keyboard button is pressed, the alert dialog is terminated.

An additional important feature is the ‘New Service Discovered’ messages appearance. These messages spawn at the top of every screen and inform about new service discoveries when service discovery alert dialogs are suspended. Service discovery alert dialogs are suspended when user browses either a map (Figure 7b), or a service webpage, or when another service discovery alert dialog (Figure 7a) is already displayed.

4.1. Use case scenario

Funchal is the capital city of Madeira and the location of our use case scenario. A mobile user equipped with a mobile device, running the commercial LBS application, performs a database update, as soon as he/she starts shopping in the city center of Funchal. Spontaneously, when the mobile user approaches the ‘Anadia shopping center’, a service discovery alert dialog spawns on the screen of her mobile device (Figure 7a), informing him/her about the presence of a nearby service (Anadia shopping center). Subsequently, our potential customer can learn about the location

Figure 7. This figure demonstrates a service discovery alert dialog and the corresponding map segment, while service marker is touched.
of the service just discovered by touching on ‘Show Map’ button and a map view appears (Figure 7b), demonstrating the user’s location in relation to the service’s location. Figure 8 demonstrates a photo taken during the use case scenario in front of the ‘Anadia shopping center’. Moreover, a service description message provides additional information related to the service just discovered, whereas in this case is a special offer on red wine products only for the current day. The sale example just described, indicates the strong time and location dependency of some particular LBS and the way this feature can be commercially exploited.

5. Conclusion and future work

In this article, we proposed the concept of PLBS, conceptualising it along three main categories: location-triggered, chain-triggered and proximity-triggered services. Through a workshop with designers and researchers in the area of LBS, we elicited a set of initial requirements for the platform and illustrated them using a number of possible usage scenarios. We highlighted how the platform aims at addressing these requirements and illustrated its implemented features through the development of a mobile application that supports proactive advertisement in urban environments. The platform proved to be versatile and scalable, supporting the rapid prototyping
of PLBS. Yet, a number of challenges remain unsolved. Below we discuss two of
them: the need for bidirectional P2P database update as well as P2P service
discovery.

The platform connects to a central remote database for updating its local services
database. A P2P feature may offer charge-free database update by connecting to
nearby application users and share database updates locally. This function would not
only alleviate users from the extra cost of database updates, it could also enable a
bidirectional database update function. For instance, users could be able to create
their own P2P Service Discovery.

Next, rather than relying on geographical positioning services, this P2P feature
could offer the potential for users’ mobile devices to function as a service beacon that
broadcast service discovery alerts (see Heikkinen 2008, Junglas 2008). A usage
scenario of this feature was described in section 2, but it is not currently implemented
in the PLBSD platform.

Notes
1. Even if the initial service instance is not.
2. A service is what a mobile user can provide, such as item purchasing.
3. Also discoverable.
4. A person here acts as a service provider.
5. For experimental and research purposes, service discovery is also triggered when a user
   exits the service perimeter.
6. For experimental and research purposes, service content can be altered locally.

References
Aalto, L., et al., 2004. Bluetooth and WAP push based location-aware mobile advertising
system. In: MobiSys 2004, Boston, Massachusetts, USA, 49–58.
Pervasive Computing, 7 (2), 85–89.
Blunck, H., et al., 2010. PerPos: a platform providing cloud services for pervasive positioning.
In: COM.Geo '10th proceedings of the 1st international conference and exhibition on
computing for geospatial research & application, Bethesda, MD, USA, 11.
services. In: ACM LBSN '10th proceedings of the 2nd ACM SIGSPATIAL international
workshop on location based social networks, San Jose, California, 31–38.
of Location Based Services, 5 (2), 61–78.
GPS Solutions, 6, 169–178.
experience. In: TIEI '11, Funchal, Madeira, Portugal, 1–2.
Ficco, M., Pietrantuono, R., and Russo, S., 2010. Supporting ubiquitous location information
116–123.


