TOWARDS LOCATION-BASED SERVICES STANDARDIZATION
An Application based on Mobility and Geo-Location

El Kindi Rezig
University of Science and Technology, USTHB, BP 32 EL ALIA, 16111, Bab Ezzouar, Algiers, Algeria

Valérie Monfort
Université De Sfax, MIRACL, Sfax, Tunisie
Université Paris 1 Panthéon Sorbonne, 90 rue de Tolbiac, 75634 Paris cedex 13, Paris, France

Keywords: LBS, Mobility, Ambient context adaptability, Web services.

Abstract: Location based services (LBSs) have improved users' life drastically because of their useful applications such as location-based advertising, vehicle tracking, mobile commerce… etc. However most LBS applications are provider-specific and can't (generally) cohabit with other LBS applications from other providers, consequently, consumers have to use different platforms and applications in order to discover and interact with different LBS applications. In our paper, we present an infrastructure based on geo-location and Web services to promote the uniformity of the way LBS providers publish their services on one hand, and the way consumers discover and interact with the LBSs they look for on the other hand. Moreover the proposed infrastructure is context-aware and promotes dynamical services accessibility by offering consumers, as they are moving, the nearest services they are interested in.

1 INTRODUCTION

Location based services (LBSs) (Sen, Sengupta, 2007) are offered through devices with location capabilities such as mobile phones, GPS devices…etc. These services include a wide range of applications such as looking for the closest restaurant, finding someone,…etc. LBSs have become a lucrative business for both wireless carriers and developers of location-aware applications. Today, LBSs have become part of users' everyday life, using several media as: mobile phones, PDAs…etc to find the closest clinic, to do bank transactions…etc.

Geo-location is the cornerstone of this research work. Geo-location aims to locate a user, an object, a data … on a map with geographical coordinates. This operation is possible using a device able to be located (GPS, …) and to publish (on real time for instance) the geographical coordinates. The location can be seen on online maps using laptops, PDAs, Mobile phones... .

Moreover, we noticed that Service Oriented Architecture (SOA) (Curbera, Khalaf, 2008) offers a great flexibility to Information Systems (IS) because each application owns interfaces masking implementation details. So, applications own interfaces including services and are seen as black boxes independently connected to a middleware as Enterprise Application Integration bus (EAI) with its connectors and adaptors. Moreover, Web services are based on standards and are till now the cheapest and simplest solution to support interoperability between platforms. Based on Web services, Enterprise Services Bus (ESB) (Chappell, 2006) (Mulesoft, n.d) is a kind of services Web based EAI and allows loosely coupling with low costs. However, even if Web services offer technical connection means for interoperability, they address neither adaptability nor context aware adaptability.

We aim:

- To define a platform that uniforms the way LBSs providers publish their services and the

Kindi Rezig E. and Monfort V. (2010).
TOWARDS LOCATION-BASED SERVICES STANDARDIZATION - An Application based on Mobility and Geo-location.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Databases and Information Systems Integration*, pages 310-315
DOI: 10.5220/0002975503100315

way consumers discover and interact with LBSs.

- To associate the location of a user to a set of services proposed by different entities belonging to this zone as restaurants, banks, hospitals, airports,… The user can define the distance between him and the entities.
- To provide required services from anywhere for mobile users via any media.
- To adapt the panel of offered services according to the current location of the user.

This paper presents a concrete design and implementation of context awareness for mobile users and an improved platform to standardize the discovery, interaction and publication of LBSs, We mainly used Web services and Microsoft .NET 3.5 to achieve this. We shall process as follows, The second section introduces Web services. The third section presents LBS. The fourth section proposes a case study. Section five shows different implementations of the presented platform.

2 WEB SERVICES

Web services (Web Services Architecture, 2004), like any other middleware technologies, aim to provide mechanisms to bridge heterogeneous platforms, allowing data to flow across various programs (Web Services and Service-Oriented Architectures, n.d), (Windows Communication Foundation, n.d). The Web service technology looks very similar to what most middleware technologies looks like. Consequently, each Web service has an Interface Definition Language, namely WSDL (Web Service Description Language) (Web Services Description Language, 2007), that is responsible for the message payload, itself described with the equally famous protocol SOAP (Object Access Protocol) (Simple Object Access Protocol 1.2, 2007), while data structures are expressed by XML (Extensible Markup Language) (Extensible Markup Language, n.d). Very often, Web services interfaces are stored in UDDI (Universal Description Discovery and Integration) registry (The UDDI Version 3.0.2 Specification, 2004) as shown Fig. 1.

In fact, the winning card of this technology is not its mechanism but rather the standards upon which it is built. Indeed, each of these standards is not only open to everyone but, since all of them are based on XML, but it is also pretty easy to implement them for most platforms and languages. For this reason, WS are highly interoperable and do not rely on the underlying platform they are built on, unlike many ORPC (Object Request Procedure Call). Web services standards are gathered in WSA (Web Service Architecture) (Web Services Architecture, 2004).

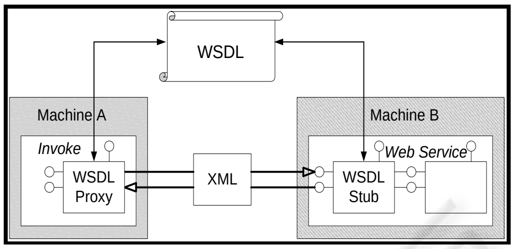

Figure 1: WSDL as a starting point for client and service generation.

According to a vast majority of industrial leaders, Web service is the best fitted technology for implementing Service Oriented Architectures. Web services provide a minimalist mechanism to interconnect different applications. But one fundamental point is the importance of the WSDL (Web Services Description Language, 2007) being the exact interface of the system. As we said earlier, most of ORPC solutions take a great care of hiding the message layer details from the developer. The WSDL contract constitutes the design view upon which developers can generate both client and server sides (proxy and stub). WS-BPEL (Business Process Execution Language) is a Web services orchestration language. An orchestration specifies an executable process that involves message exchanges with other systems, such that the message exchange sequences are controlled by the orchestration designer. WS-BPEL (Web Services Business Process Execution Language, 2007) provides a language for the specification of Executable and Abstract business processes. By doing so, it extends the WS interaction model and enables it to support business transactions. WS-BPEL defines an interoperable integration model that facilitates the expansion of automated process integration (Ferrara, 2004) in both the intra-corporate and the business-to-business spaces.

Geographical discovery of services has improved users' everyday life dramatically, and it has made people' life easier, services include road navigation, geographical social networks, ….etc, such solutions already exist today and they are being increasingly popular for their big usefulness, however most of the existing geographical solutions are vertical, meaning that they are vendor-specific and can't live in the same environment because of their heterogeneity. Having Web services that can express their offers geographically improves drastically the capabilities of today's Web services, in fact, customers will be able to search for them geographically and according to their location constraints. Moreover, Web services resolve the problem of the geographical services heterogeneity.

Our goal is to design and implement a location-aware system that takes advantage of Web services with geographical capabilities, our system is able to deliver the services that the client looks for according to his current location, we mean the nearest services to the client.

3 TOWARDS A HOMOGENOUS LBSS ENVIRONMENT

Location based services application can be described as applications that are dependent on the user's location in order to work. In other words LBS applications give information based on the device's geographical position (D'Roza, Bilchev, 2003). LBSs can be divided into two categories as: Triggered and user-requested. In the triggered category, LBS relies on a certain condition to be met in order to retrieve the user's position and use it in the location-aware application, for example, when a vehicle has an accident, an automatic emergency call with the vehicle's position would be triggered. In user-requested LBS, users transmit their current location to have relevant information like driving directions, finding the nearest shopping malls…etc

Many efforts are in progress to standardize LBSs, the most popular LBS standard is OpenLS (OpenLS Implementation Standards, 2010) defined by the Open Geospatial Consortium (OGC), it offers implementation specifications to the most critical parts of LBSs. Unfortunately, the current methods of designing LBS applications are not enough to get the most of LBS applications using open standards. This limitation is addressed in our paper using SOA and GML.

Location based services providers increasingly worry about the visibility of their services to potential consumers, unfortunately because LBSs are provider-specific each editor has a discovery policy through which consumers can find it. Moreover, providers of LBSs may use different ways to access and interact with their services. Eventually, LBS providers have different ways to describe their services in terms of their locations and how to interact with them. To express all the aforementioned features we defined a three-layered stack. Layers are aspects that LBSs are concerned about: Discovery, Description, Access.

When every provider offers a different stack, users wouldn't be able to discover it as long as they aren't sticking to the provider's discovery policy; the same issue is applicable to the access and description policies. The following Fig.2 shows the scenario where every LBS provider defines its own concerns stack.

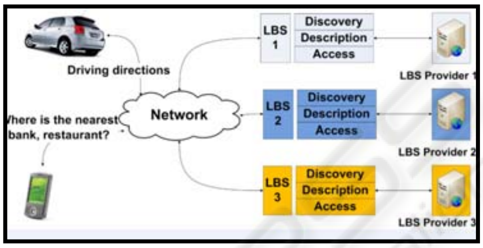

Figure 2: LBS providers in a heterogeneous environment.

In order to improve the aforementioned situation, each LBS provider should stick by the same concerns stack, this following way. All LBSs use the same discovery, services description and access policies. By having this scenario, LBS providers share the same platform between each other, this helps maximizing their visibility, and consumers would have more choices to pick the most convenient LBS. In order to achieve an homogenous LBS environment, we have used Web services as they are the best way to make heterogeneous environments interoperable. Each LBS is a Web service, and with Web services we can associate standardized items to the three layers of the concerns stack as follows:

- **Discovery**: We use the UDDI registry to publish and discover all LBSs.
- **Description**: Descriptions of LBSs is done through the Web service's WSDL, geographical description is done through the Web service's GML profile (explained in section 5).
- **Access**: All interactions with the Web services are done with SOAP.

The following Fig.3 shows the LBS providers in a homogenous environment:

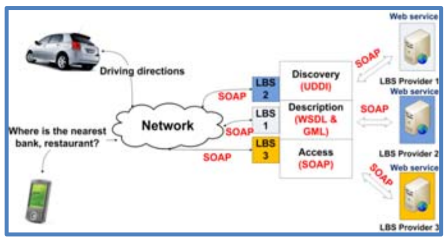

Figure 3: LBS providers in a homogenous environment.

Web services are the fitted solution for flexibility to changes and interoperability but as mentioned in previous related works (Hmida, Tomaz, Monfort, 2006) (Haddad, Moreaux, Rampacek, 2006) (Tomaz, Hmida, Monfort, 2006) they do not offer context aware adaptability.

4 CASE STUDY

4.1 Context

Users can ask for geographical services (LBSs wrapped in web services) via a UDDI registry. They can also use their cars, and, while driving, ask for the closest services as banks or hospitals, … Since all he LBSs are Web services, users can interact with these services using the WS's WSDL and the WS's GML profile.

4.2 Technical Architecture

4.2.1 First scenario: User accesses geographical services (LBS application wrapped in web services)

Let us focus on the LBSs accessibility. Fig. 4 shows two parts as client and server sides. On the client side, the user is detected by different kinds of devices to determine his location and the area corresponding to his location. Then, on the server side, a specific interface (as a gateway) provides a mapping between the area and the services corresponding to this area.

4.2.2 Second scenario: An actor publishes geographical services

This second scenario explains how to fulfil the UDDI registry and the correspondence between an area and a service. Then, for instance, a bank manager wants to inform potential users that his bank offers a set of services as ATM. He can ask for a service provider to subscribe an account. A contract is signed between the bank and the service provider with financial engagements. The service provider proposes a GUI pattern as a form, to be fulfilled by the bank manager to promote bank services. So, the publisher gives the Web service WSDL as well as its GML (Geography Markup Language) profile that geographically expresses the Web service. GML is an XML grammar that expresses geographical information, ranging from complex graphical representations to simple points representations, GML has been defined by the Open Geospatial Consortium (OCG). GML can also serve as an interchange format for geographical data. The accuracy of this language has led us to choose it in order to describe geographical information.

The bank manager uses a Publication module and the service provider defines the link between the services and the area including bank via UDDI Publish Interface and then via UDDI registry. Then, this module is in charge of the communication between the publisher and the UDDI registry, it publishes the information the publisher gives but most importantly, it associates the GML profile to the publisher's Web services via tModels.

A tModel is a data structure representing a service type (a generic representation of a registered service) in the UDDI (Universal Description, Discovery, and Integration) registry. Each business registered with UDDI categorizes all of its Web services according to a defined list of service types. Businesses can search the registry's listed service types to find service providers. The tModel is an abstraction for a technical specification of a service type; it organizes the service type's information and makes it accessible in the registry database. For every published Web service, the publish module publishes a tModel that holds the GML profile information. This tModel will be used in the geographical discovery. So, every published Web service has two descriptors: i)WSDL that explicitly describes the Web service interface and ii)GML profile that expresses the Web service geographical information.

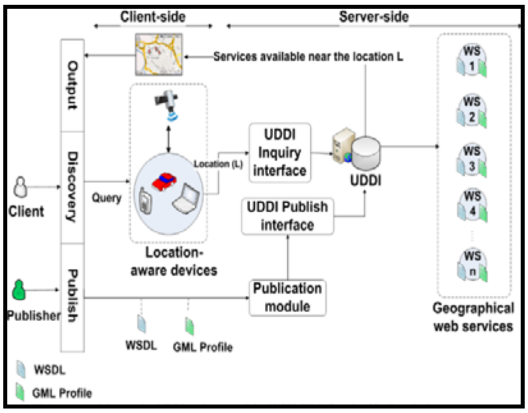

Figure 4: Technical architecture.

5 IMPLEMENTATION

The Windows 7 sensors and location platform has brought a tremendous facility in developing location-aware applications in the Microsoft .NET platform, while most existing location platforms today are vendor-specific, the Windows 7 sensors and location platform defines a uniform model for location devices, each sensor respecting this model will enable developers to handle it without worrying about the implementation details of the sensor's driver. In order to simulate a GPS receiver we use a virtual location sensor: Laptop Lojack (Laptop LoJack project, 2009), an open source virtual GPS ready to work with the Windows7 sensors and location platform. We implemented the system as a WPF application since it offers rich and intuitive GUIs that is convenient to any user. The Windows Presentation Foundation (or WPF) is a graphical subsystem for rendering user interfaces in Windows-based applications. WPF, previously known as "Avalon", was initially released as part of .NET Framework 3.0. WPF is built on DirectX, that provides hardware acceleration and enables modern UI features like transparency, gradients and transforms. WPF provides a consistent programming model for building applications and provides a clear separation between the user interface and the business logic. We chose Microsoft .NET 3.5 for the development environment, for location awareness we used the Windows 7 sensors and location platform to make the application location-aware. To interact with UDDI registry we used Microsoft UDDI SDK. As UDDI registry, we used jUDDI. This project is under the Apache foundation (jUDDI project, 2009). The core layer shows back end applications. This section presents the code of some relevant parts of our application written in C#. In order to return the most relevant services according the user's location, the following code in Fig5 has been proposed (as a part of the discovery module).

```
//Services_list holds the service nodes in the web service's GML
profile
// For each service we check whether it's near the user's location
foreach (XmlNode node in services_list)
{
String[] latlong =
node.SelectSingleNode("position").SelectSingleNode(@"gml:Point",
xnsm).SelectSingleNode(@"gml:pos", xnsm).InnerText.Split(' ');

// ser_pos holds the service's geographical coordinates
      ser_pos = new VELatLong(Convert.ToDouble(latlong[0]),
Convert.ToDouble(latlong[1]));
//user_defined_distance represents the distance within which the
user wants to find services
// If the user puts 60 in the user_defined_distance, services
more than 60 meters away won't be considered
if (distance(ser_pos, user_pos) < user_defined_distance) // The
service is within the user defined distance
{
  ...
```

Figure 5: The distance checking code.

The application's GUI has been made with WPF, the following picture 5, shows the search interface with the results shown on the map.

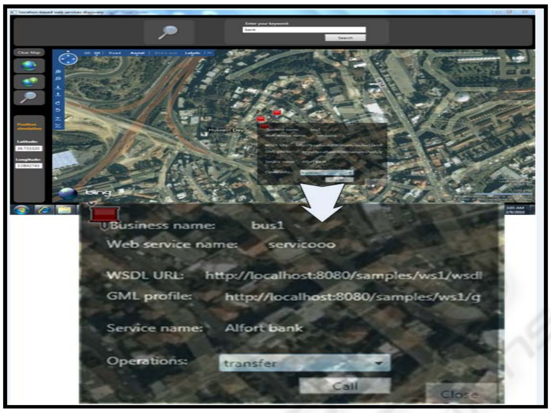

Figure 5: The search results shown on the map.

6 RELATED WORKS

Several research works aim to use meta modelling such as (Farias, Pires, Sinderen, 2007), (Frankel, 2003) (Gottschalk, der Aalst, Wil, Vullers, 2008), (Klein, Hélouet, Jézéquel, 2006), (Lundesgaard, Solberg, Oldevik, France, Aagedal, Eliassen, 2007) (Matthias, Dustdar, Rosenberg, 2007), context awareness platform as WComp (TIGLI, LAVIROTTE, REY, HOURDIN, CHEUNG- FOO-WO, CALLEGARI, RIVEILL, 2009) or other platforms to promote adaptability. (Sen, Sengupta, 2007) presents the need to standardize LBSs using open standards and the key benefits of this move, however it just focuses on the standardization of the geographical part of the LBS applications using GML, thus, no means to standardize the LBS applications globally. We did not find any research work using context aware platform in this domain. Moreover, we did not find any research work using the same approach to locate a user and to associate Web services to geographical location with GML extensions.

7 CONCLUSIONS

This research paper presents a new means to standardize LBS environments, on one hand this means allows LBS providers to be more visible to consumers, on the other hand consumers don't have to use proprietary mechanisms to access and interact with LBS applications. Adaptability is given by Geo-location that allows to select services according to the location context. We used web services as a technical means to standardize LBS applications, we used Microsoft .NET 3.5 for the development environment and for location awareness we used the Windows 7 sensors and location platform. We described a genuine mechanism to associate the position of a user to an area and to services (defined and subscribed in this area) via an extended GML and UDDI. We implemented the whole case study and we showed examples of code. We have now to improve adaptability to auto adaptability with reflection mechanism and to also improve Geo location data bases.

REFERENCES

Web Services Architecture, retrieved from the website http://www.w3.org/TR/ws-arch/

Simple Object Access Protocol 1.2, 2007, retrieved from the website http://www.w3.org/TR/SOAP

Web Services and Service-Oriented Architectures, retrieved from the website http://www.service-architecture.com/

Windows Communication Foundation retrieved from the website http://msdn.microsoft.com/Webservices/build-ing/wse/

The UDDI Version 3.0.2 Specification, 2004, retrieved from the website http://www.uddi.org/pubs/uddi_v3.htm

Web Services Description Language, 2007, retrieved from the website http://www.w3.org/TR/wsdl20/

Extensible Markup Language, retrieved from the website http://www.w3.org/XML/

Web Services Business Process Execution Language, 2007, retrieved from the website http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel

F. Curbera, R. Khalaf, N. Mukhi, Quality of Service in SOA Environments. An Overview and Research Agenda (Quality of Service in SOA-Umgebungen). it - Information Technology 50 (2): 99-107, 2008.

D. Chappell, Enterprise Service Bus. Publisher: O'Reilly, 2006 Alexander Ryan

De Farias, C. R. G., Pires, L. F., and van Sinderen, M. A MOF Metamodel for the Development of Context-Aware Mobile Applications. In Proceeding of the 22nd ACM Symposium on Applied Computing (SAC'07) (2007).

Ferrara, A. (2004). Web services: a process algebra approach. In ICSOC '04: Proceedings of the 2nd international conference on Service oriented computing, pages 242–251, New York, NY, USA. ACM Press.

Frankel S. David., Model Driven Architecture: Applying MDA to Enterprise Computing, Wiley Publishing, Inc(2003)

Gottschalk, F. and van der Aalst, Wil M. P. and Jansen-Vullers, M. H. and La Rosa, M. (2008) *Configurable Workflow Models.* International Journal of Cooperative Information Systems (IJCIS).

Haddad, S., Moreaux, P., and Rampacek, S. (2006). Client synthesis for Web services by way of a timed semantics. In Proceedings of the 8th Int. Conf. on Enterprise Information Systems (ICEIS06), pages 19–26.

Mulesoft, retrieved from the website http://www.mulesoft.org/display/MULE/Home

Hmida, M. B., Tomaz, R. F., and Monfort, V. (2006). Applying aop concepts to increase Web services flexibility. Journal of Digital Information Management (JDIM), 4(1):37–43.

J. Klein, L. Hélouet, and J. M. Jézéquel. -- Semantic-based weaving of scenarios. -- In *Proceedings of the 5th International Conference on Aspect-Oriented Software Development (AOSD'06)*, Bonn, Germany, March 2006. ACM

S. Lundesgaard, A. Solberg, J. Oldevik, R. France, J. Oyvind Aagedal, F. Eliassen, *Construction and Execution of Adaptable Applications Using an Aspect-Oriented and Model Driven Approach*, IFIP DAIS 2007, LNCS 4531, 76-89, 2007

Matthias, B., Dustdar, S., and Rosenberg, F. A survey on context-aware systems. *International Journal of ad Hoc and ubiquitous Computing 2* (2007).

J.Y. Tigli, S. Lavirotte, G. Rey, V. Hourdin, D. Cheung-Foo-Wo, E. Callegari, M. Riveill. «WComp Middleware for Ubiquitous Computing: Aspects and Composite Event-based Web Services ». *Annals of Telecommunications*, volume 64, n° 3-4, pages 197, april 2009. ISSN 0003-4347

Tomaz, R. F., Hmida, M. B., and Monfort, V. (2006). Concrete solutions for Web services adaptability using policies and aspects. The International Journal of Cooperative Information Systems (IJCIS), 15(3):415–438.

Winograd, T. Architectures for context. *Human-Computer Interaction (HCI) Journal 16* (2001)

GML Encoding Standard, 2010, retrieved from the website http://www.opengeospatial.org/standards/gml

Laptop LoJack project, 2009, retrieved from the website http://laptoplojack.codeplex.com

jUDDI project, 2009, retrieved from the website http://ws.apache.org/juddi/

T. D'Roza, G. Bilchev, 2003, An Overview of Location-Based Services, BT Technology Journal, Springer Netherlands, volume 21, number 1,

OpenLS Implementation Standards, 2010, retrieved from the website http://www.opengeospatial.org/standards/ols

Sumit Sen, Smita Sengupta, Open standards in location based services, 2007, retrieved from the website: www.gisdevelopment.net/technology/lbs/techlbs002.htm