Dynamic Registration and Discovery of Location-Based Services Using Spatial Registries

Nathanaël Cottin
University of Technology of Belfort-Montbéliard
SeT laboratory
90010 Belfort Cedex, France
nathanael.cottin@yahoo.fr

Maxime Wack
University of Technology of Belfort-Montbéliard
SeT laboratory
90010 Belfort Cedex, France
maxime.wack@utbm.fr

ABSTRACT

Service-Oriented Architectures rely on services cooperation to answer requesters' needs. Services invocation requires access points, such as Internet URLs, provided by specific services called registries. Traditional registries allow to register (advertise) services and provide requesters with suitable services' binding information. However, these registries are not designed to take a particular concern about Location-Based Services (LBSs). LBSs' discovery must be strongly related to requesters' provided geographical locations as well as LBSs' coverage areas which define LBSs' geographical limits of action. This work introduces Spatial Registries, a new type of registries specialized in LBSs' registration and discovery. Spatial Registries' operations on LBSs registration and discovery are described. Different architectural designs and security issues related to Spatial Registries are also discussed.

Categories and Subject Descriptors

H.1.0 [**Information Systems**]: MODELS AND PRINCIPLES—*General*; H.3.3 [**Information Systems**]: INFORMATION STORAGE AND RETRIEVAL—*Information Search and Retrieval*

General Terms

Design, Reliability

Keywords

Spatial Registry, Location-Based Service, load-balancing

1. INTRODUCTION, MOTIVATION

Wireless e-commerce, also known as mobile-commerce, relies on emerging technologies to allow users get access to services over networks based on their locations. These services which take positioning information into account, called Location-Based Services (LBSs), require registries to handle publication (i.e. advertisement) and discovery procedures. The latter can integrate complex (though powerful) mechanisms such as load-balancing strategies and services' descriptions matching using ontologies [1].

Focusing on Service-Oriented Architectures (SOAs), traditional registries are able to register and provide discovery mechanisms for any kind of service. However, these registries do not take any particular care of LBSs' specificities, such as the ability to answer requests based on the requesters' locations. These particular characteristics are described as any other property within services' descriptions. Such registries may potentially lead to registration incoherences and inappropriate discovery results due to their lack of specific support for LBSs' position and geographical coverage information.

Thus, architectures, communication protocols, registration and discovery procedures adapted to efficient LBSs' registration and discovery must be defined. This article proposes this novel type of registry, called Spatial Registry, specialized in LBSs' registration and discovery. LBSs' components and Spatial Registries' global registration and retrieval procedures are explained at specification (i.e. mathematical) level along with minor implementation directives.

The remainder of this paper is structured as follows: next section describes some related work; section 3 exposes LBSs and Spatial Registries' definitions and main concepts; section 4 studies relashionships between LBSs and Spatial Registries based on previous section's results; centralized and distributed architectural approaches are then criticized in section 5; the second last section considers security issues related to Spatial Registries and LBSs' registration, discovery and invocation; finally, conclusions summarize the contributions and results of this work and expose many future research directions and implementation perspectives.

2. RELATED WORK

Registries are a key component of distributed systems. Among all architectures and implementation technologies, the W3C's SOA [2] is the current set of standards to design web services which communicate over the Internet. This SOA benefits from the experience of the Object Management Group (OMG) on CORBA [3]. The OMG designed this architecture based on a middleware, called Object Request Broker (ORB), to allow interoperable and transparent communications between heterogeneous systems. Another key technology for the sole Java programming language is Sun's Entreprise JavaBeans (EJBs) [4]. All these representative systems make use of registries for components' (i.e.

AUPC'09, July 13–17, 2009, London, United Kingdom.
Copyright 2009 ACM 978-1-60558-647-2/09/07 ...$10.00.

services) registration and discovery. These registries, respectively CORBA Naming and Trading Object Services, JNDI and OASIS UDDI, are presented hereafter.

2.1 JNDI

The Java Naming and Directory Interface [5] is the core component for discovering EJBs across networks. It basically represents a hierarchical set of server components references (packages and classes names) used during the discovery process. JNDI also supports searching within directories by means of attributes' description.

2.2 CORBA Naming and Trading Services

The Common Object Request Broker Architecture is the first set of standards created by the Object Management Group (OMG) consortium to address distant objects invocations in an heterogeneous environment. CORBA relies on the OMG's Model Driven Architecture (MDA) [6], which extends the Object Management Architecture (OMA). The latter defines innovative concepts, such as a standardized communication medium (the ORB) and a specific Interface Description Language (namely OMG IDL) used to publish servers' publicly-available interfaces independently from programming languages. Servers are made accessible by means of their standardized Interoperable Object References (IORs). The OMG suggested a set of defaut servers called services as part of the OMA. Services aim at providing common horizontal facilities used across businesses.

The OMG defined two complementary registration and discovery services known as Naming (white pages) and Trading Object (yellow pages) Services. The Naming Service (NS) [7] is probably the most widely used service compared with other CORBA common services. Its IDL specification defines three basic operations related to servers: registration (providing server name, reference and description), discovery (based on server name) and destruction. The Trading Object Service (TOS) [8] allows to advertise servers depending on their capabilities. Requesters ask TOSs to obtain references for services having certain characteristics by means of the OMG IDL's query operation from the "Lookup" interface.

2.3 OASIS UDDI

The W3C's SOA makes use of many concepts implemented and proven by CORBA, such as platforms independence and operating systems heterogeneity handling (as this architecture relies on XML encapsulated within HTTP), services proxying (stubs and skeletons), common horizontal facilities (such as registries) specification and interfaces publication. The W3C's SOA also benefits from both HTTP and XML advantages in terms of firewalls crossing and information automatic manipulation (encoding, decoding and validation).

Among the extended list of SOA's facilities, the OASIS maintains the Universal Description Discovery and Integration (UDDI) [9], a set of specifications for a standard registry. The UDDI combines six complementary APIs similar to white, yellow and green (containing binding information) pages to register and discover web services. The UDDI's "Inquiry" discovery interface provides requesters with references (URLs) to registered services which meet a particular technical or business need. UDDI registries natively support the SOAP protocol and remains compatible with CORBA's Internet Inter-ORB Protocol (IIOP) and Java RMI. The OMG also published mapping rules between the OMG IDL and the Web Services Description Language (WSDL) to ensure interworking between CORBA and W3C's SOA.

3. DEFINITIONS AND CONCEPTS

This section tackles the main mathematical concepts necessary to specify LBSs and Spatial Registries' components. These elements are necessary to define relationships between LBSs and Spatial Registries.

3.1 Location-Based Service

According to the OpenGeospatial Consortium, a LBS is "a wireless-IP service that uses geographic information to serve a mobile user. Any application service that exploits the position of a mobile terminal". Indeed, a LBS can be considered as a particular type of service having the ability to answer users' requests depending on its geographic coverage areas.

A selected LBS is likely to fulfill requesters' needs when the following criteria are met:

- The service description matches requesters' requirements, as performed by traditional registries;
- The service geographically covers the position (or area) given by the requester. This constraint is specific to Spatial Registries.

Each LBS s, part of the set of available LBSs E (which includes services which may not be registered by Spatial Registries), refers to a couple expressed by equation 1. It is composed of the following elements:

- A coverage area CA_s expressed by equation 2. Usually this area is a well-delimited set of m disjoined sub-areas $CA_{s,x}$ described with geographic coordinates $coord_{s,x}$ along with a decay gradient $g_{s,x}$. Each sub-area coverage zone may be a simple (rectangle, circle) or complex (polygon, free form) shape;
- A set of descriptions $desc_s$ which declare the operational behaviours handled by this LBS.

$$s = \{CA_s, desc_s\} \tag{1}$$

$$\begin{cases} CA_s = \bigcup\limits_{x\in[1,..,m]} CA_{s,x}, \quad \bigcap\limits_{x\in[1,..,m]} CA_{s,x} = \emptyset \\ CA_{s,x} = \left\{g_{s,x}, \bigcup coord_{s,x}\right\} \end{cases} \tag{2}$$

Extrapolating from figure 1, each sub-area's decay gradient $g_{s,x}$ indicates the softness degree from the barycenter to the border of $CA_{s,x}$. This gradient, expressed by equation 3, is a ratio which reflects an average loss of Quality of Service (QoS), such as a result relevancy. For simplification purpose, this coverage area softness gradient is not discussed in this article (it is assumed that $g_{s,x} = 1$, which corresponds to a neutral gradient) and thus does not interfere with discovery procedures accuracy. Sub-areas' barycenters $b_{s,x}$ could be added to each LBS components (from equation 1) to improve implementations performances.

$$g_{s,x} = \frac{QoS_{center_{s,x}}}{QoS_{border_{s,x}}} \tag{3}$$

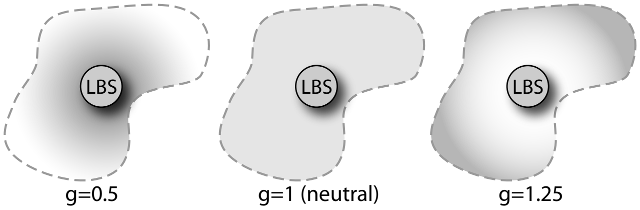

Figure 1: Gradient impact on area softness

3.2 Spatial Registry

Traditional registries allow to register services and dynamically obtain registered services' contact information (i.e. references). Even if, as any other services, LBSs may be registered within traditional registries (inhibiting the notion of geographical coverage), using LBSs specific features require specialized registries we call Spatial Registries. The latter incorporate LBSs' coverage areas and requesters' locations in providing requesters with relevant services.

The most important functionality of registries is to allow services discovery. Common registries make use of registered services descriptions (possibly via ontologies) to select suitable services. Spatial Registries restrict possible results to services which not only match descriptions but also geographically cover requesters' provided locations.

Figure 2 points out the inclusion of both services' description and location matching processes to complete LBSs' discovery ("lookup"). This location matching task is specific to Spatial Registries. Indeed, Spatial Registries may return less suitable services than traditional registries as an additional geographical constraint is performed.

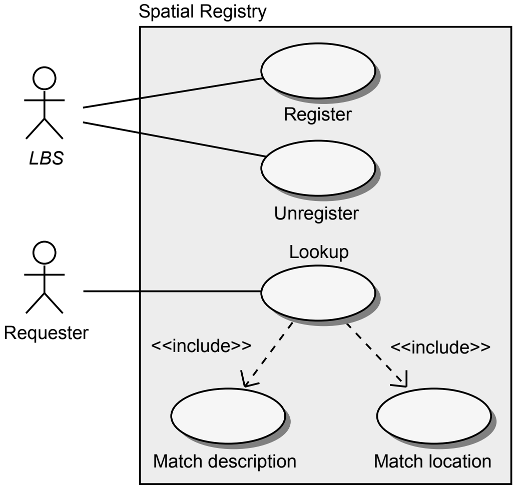

Figure 2: Spatial Registry functionalities

Given a set of Spatial Registries R, each Spatial Registry $r \in R$ is virtually positioned with an initial position value l_r which may evolve. This position places the registry in its geographical context and is used to register LBSs. One could also add a preferred geographic registration area to decide LBSs' registration acceptance but this information would limit Spatial Registries' extensibility.

A given Spatial Registry r also owns a global coverage area GCA_r which directly derives from its set of registered LBSs $E_r \subseteq E$. This coverage area is obtained by equation 4 using the registered services' coverage areas, given that CA_s.

$$GCA_r = \begin{cases} \bigcup_{s \in E_r} CA_s, E_r \neq \emptyset \\ \emptyset, E_r = \emptyset \end{cases} \quad (4)$$

A Spatial Registry r is indeed composed of its current location l_r and a set of registered services information E_r. These elements are mathematically expressed by equation 5.

$$r = \{l_r, E_r\}, r \in R, E_r \subseteq E \quad (5)$$

4. REGISTRIES' COMMON OPERATIONS

Independently from Spatial Registries' internal structure, the main issues lie in selecting the most appropriate registry from a set R of available Spatial Registries for LBSs' registration and discovery procedures.

4.1 LBS registration

Registering a LBS $s \in E$ aims at selecting the most relevant Spatial Registry $r \in R$ to perform registration. This registry minimizes the distance between its current location and the barycenter of CA_s, as depicted by figure 3. A complementary election protocol which minimizes registries' GCAs may be required when the LBS is equidistant from multiple Spatial Registries.

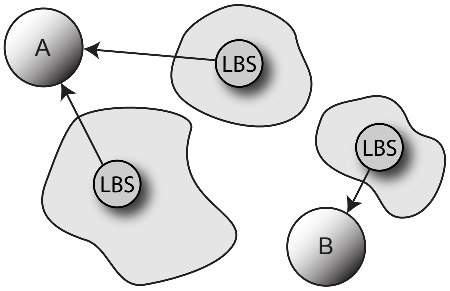

Figure 3: LBSs' registration on registries A and B

The registration process may ensure that a LBS only registers to a single Spatial Registry. This can be mathematically expressed by the assertion of equation 6, which indicates that all sets E_x of services registered within registries x realize a partition of the set $E' \subseteq E$ of all registered LBSs. Given this registration principle and allowing registries replication, Spatial Registries lead to introduce a novel load-balancing approach primarily based on services location information rather than hosts capabilities, jointly with common balancing techniques.

$$\begin{cases} \bigcup_{r \in R} E_r = E' \\ \bigcap_{r \in R} E_r = \emptyset \end{cases} \quad (6)$$

Registering a LBS s within a Spatial Registry r has the following effects on r:

- The discovery information of s is added to the set of r's registered services E_r;

- GCA_r is updated to contain the set of coverage areas provided by s. Maintaining this GCA simplifies services discovery.

4.2 LBS unregistration

Unregistering a LBS s from a Spatial Registry r consists in removing this service from the set of registered services, as shown by equation 7, assuming that equation 6 is verified and given that $\bar{E}_r$ stands for E_r before unregistration.

$$E_r \leftarrow \bar{E}_r - \{s\} \tag{7}$$

As a result, this LBS is no more discoverable by requesters using Spatial Registries, although this service may still be running (it remains part of the set of services E).

4.3 LBS discovery

As far as more than one Spatial Registry is available, a requester needs to ask for the appropriate registry, based on its provided location and the service description. A straightforward idea depicted by figure 4 is to match the requester's location against each registered LBS coverage area to determine which services geographically enclose this location.

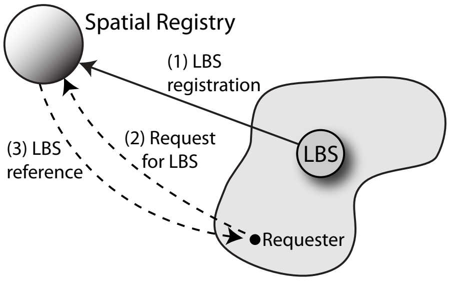

Figure 4: Registered LBS discovery

Given a Spatial Registry r having registered n LBSs, each registered LBS coverage area must be checked against the requester's location l_{req} to create the set S_r of suitable services references. This set S_r is defined by equation 8.

$$S_r = \bigcup_{s \in E_r} s, GCA_s \cup l_{req} = GCA_s \tag{8}$$

S_r is then refined, taking other constraints into account, such as description matching (by means of ontologies matching for example). Let $S'_r \subseteq S_r$, the result returned by the Spatial Registry r. Rejected registries must return an empty set ($S'_r = \emptyset$) as no service matches. Supposing that non-empty intersections between GCAs' coverage areas may occur, the main set S' of all suitable services from all available registries is given by equation 9.

$$S' = \bigcup_{r \in R} S'_r \tag{9}$$

This basic discovery process happends to be time consuming when at least one of the following conditions is encountered:

- Many Spatial Registries are invocable;
- Many LBSs are registered;
- Services matching against requesters' locations and required descriptions is computationally difficult to perform.

4.4 Implementation perspective

Storing GCAs within registries and updating each time a LBS registers and unregisters allows to directly reject Spatial Registries which do not contain the requester's location (and return an empty set) without checking each registered LBS. Remaining candidate registries must then select the appropriate services to build S' using each LBS coverage area. It may thus be useful to add the GCA to the set of components of a Spatial Registry r (see equation 10), although this GCA can be obtained with equation 4.

$$r = \{l_r, E_r, GCA_r\}, E_r \subseteq E \tag{10}$$

Such redundancy may advantageously replace the definition given by equation 5 in an implementation perspective.

5. ARCHITECTURAL CONSIDERATIONS

Registries can be organized in federations and collaborate to complete their tasks (i.e. LBSs registration, unregistration and discovery). This section is dedicated to architectural designs benefits. Based on the paradigms defined by [10], two concurrent architectures may improve Spatial Registries' scalability and performances:

- A centralized (i.e. federated) architecture, using a Global Spatial Registry (GSR) as the sole entry point to get access to federated registries. The services discovery process using the GSR has two variants explained hereafter, depending on whether the GSR returns Spatial Registries access points or directly performs services selection;
- A fully distributed (peer-to-peer) system in which Spatial Registries keep references to other "friendly" Spatial Registries' references to eventually redirect registering LBSs and discovery requests to.

Each alternative design is explained hereafter.

5.1 Centralized architecture

Deploying a centralized architecture implies a Global Spatial Registry (GSR) as the single access point for both LBSs and requesters. Spatial Registries may only be visible to requesters asking for LBSs references after contacting the GSR. The GSR has knowledge about all available Spatial Registries but Spatial Registries do not know each other. This centralized model is depicted by figure 5.

Services registration and discovery need to ask the GSR to get access to registries. Spatial Registries may only accept services registration and discovery requests issued by the GSR. The GSR owns a copy of each registry's location and may hold the corresponding GCA to redirect both LBSs and requesters to the appropriate registries. Registered Spatial Registries may not hold any GCA information as far as they notify the GSR each time a LBS successfully unregisters (or when a service reference is detected to be outdated).

As for any centralized system the main drawbacks lie in the bottleneck of the GSR and the inactivation of all registries when the GSR shuts down or crashes. On the contrary, communications security and global stability are

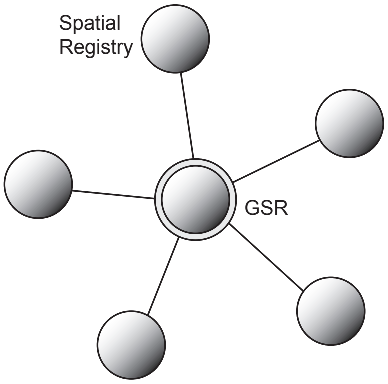

Figure 5: Centralized architecture example

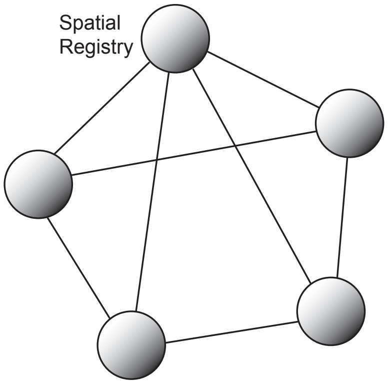

Figure 6: Distributed architecture example

simplified due to the central component of the architecture which drives common actions on Spatial Registries.

5.1.1 LBSs registration

Any LBS wishing to register must ask the GSR. The latter consults the set of registries' locations and selects the most appropriate registry. It then forwards the LBS information to the registry. The corresponding GCA is updated, either by the GSR or by the selected registry.

5.1.2 LBSs unregistration

Unregistering a LBS simply consists in removing its registered information from the GSR's set of GCAs. In order to keep consistency with Spatial Registries, the corresponding Spatial Registry must also remove any reference to this LBS from its own GCA.

5.1.3 LBSs discovery

Discovering a LBS requires to ask the GSR for a registry. The GSR selects a set of suitable registries (i.e. registries able to answer the request) either using its local set of GCAs of concurrently asking its known registries.

The final purpose of the requester is to obtain a set S' of services matching its request. This set can be directly generated by the GSR (thus hiding Spatial Registries to the requester) or the requester can construct S' based on the set of suitable registries' references provided by the GSR.

5.2 Distributed architecture

An alternative distributed architecture illustrated by figure 6 may overcome the centralized approach drawbacks, although it has its own constraints due to registries' distribution.

This design approach is similar to peer-to-peer interactions between Spatial Registries, thus avoiding the use of a GSR.

5.2.1 LBSs registration

A LBS requiring registration must own a reference to a Spatial Registry to ask for registration. The contacted registry either directly performs the service registration or redirects to a more appropriate registry, based on its local knowledge of registries' GCAs. The latter may not be the optimal registry, though. Reaching the optimal registry depends on each registry knowledge of the global topology. Indeed, registration accuracy is strongly related with the connectivity degree of the Spatial Registries' graph. At most one exchange between registries (i.e. *hop*) is necessary to select the most appropriate registry in case (1) the graph of Spatial Registries is fully connected (complete) and (2) each Spatial Registry holds information about all other registries' positions and references. A weakly connected graph is likely to result in non-optimized registrations as the effectively most appropriate registry may not be contacted. Indeed, graph search techniques must be used to ensure a best-effort (minimizing registries' *hops*) and valuable registration of LBSs.

The redirected registration request may enclose the set of already contacted registries to avoid an infinite registration process (in the particular case an infinite redirection between registries occurs). This set is also useful to make registries aware of other registries' activation and thus extend their local knowledge of the graph of registries.

5.2.2 LBSs unregistration

A LBS unregistration process requires the appropriate Spatial Registry (i.e. which performed registration) to remove any information related to this service and update its GCA. In case the LBS is not registered by the contacted Spatial Registry, the latter forwards the service unregistration request until either the whole graph of registries is covered (each registry is contacted) or the appropriate registry unregisters this LBS.

Infinite looping is prevented by means of adding to the redirected request the set of already contacted registries and a time-to-live value.

5.2.3 LBSs discovery

A Spatial Registry receiving a LBS discovery request first compares the requester's location with its GCA. In the case this location is outside the registry's GCA, the registry redirects the requester to another Spatial Registry possibly able to answer the request, based on its local knowledge of other registries GCAs.

Discovery requests additional fields similar to LBSs registration prevent from infinite loops.

6. SECURITY CONCERNS

Security is of great importance in SOAs as requesters may not be granted access to restricted services. Common security issues related with distributed systems and SOAs have been deeply addressed and standardized in the literature

([11] and [12]). As distributed components, LBSs and Spatial Registries do conform to these security aspects. Indeed, security is involved at the following stages:

- At GSR level: the GSR, in the sole case of a centralized architecture, may reject unauthorized LBSs' registration and discovery requests;

- At Spatial Registry level: Spatial Registries must grant or deny access to LBSs registration and discovery depending on services or requesters credentials and services coverage areas. In the case of a centralized architecture, Spatial Registries may only accept requests for services registration issued by the GSR. They may then add complementary security checks the GSR may have not performed. Services coverage areas may also lead to reject registrations in case they do not comply with Spatial Registries policies. Obtaining network references to services is also subject to security rules which may involve requesters' localizations;

- At service level: finally, LBSs' internal security is also required to decide whether to accept or decline invocations. A LBS must primarily reject requests which mention requesters' locations outside the scope of its coverage area. Both coarse-grained and fine-grained security policies, respectively global security policies at service level and specific security policies per operation, may be defined for each LBS.

As suggested by [13], information required to grant or deny access to a given service or a particular operation can be included in its IDL declaration to avoid known unsuccessful requests (prior to requests submissions). This also applies to the GSR, which is primarily a service itself.

7. CONCLUSIONS, FUTURE WORK

This article suggested a novel type of registry specialized in Location-Based Services registration and discovery called Spatial Registry. Spatial Registries' main concepts, mathematical notations, as well as architectural and security considerations have been exposed to demonstrate the benefits of this new concept. The centralized approach has already been implemented using both rectangular and circular geographic areas.

Next step consists in adding support for other area shapes (mainly polygons) and softness gradient as well as implement the suggested distributed architecture. The latter requires to investigate graphs protocols for best Spatial Registry selection during the registration process and fast services matching based on ontologies.

Beyond implementing the peer-to-peer vision, tremendous perspectives and challenges are offered. Further work will focus on LBSs balancing between Spatial Registries to avoid LBSs absorption by a small number of registries. We will also evaluate both architectures and compare benchmarks to apply adaptative strategies. Considering coverage areas' bounding rectangles may contribute to improve discovery performances.

Selection of Spatial Registries during LBSs registration and discovery processes may then be subject to suitability and correlation indices. These indices may be expressed by the probability to fit the requester's requirements in terms of geographic coverage and service description. One might also think about creating Spatial Registries dedicated to certain types of business LBSs, thus improving descriptions matching performances.

Moreover, current LBSs' coverage areas remain identical for each published operation. A possible extension is to specify a coverage area per operation. The global coverage area of a given LBS will be defines by the sum (union) of its coverage areas per operation. The impact of this extension on registration and discovery procedures will be evaluated.

Interoperability between common registries and Spatial Registries is also part of our concerns to ensure that LBSs and other services can still be discovered.

Finally, we will consider allowing Spatial Registries' dynamic positioning. Evolution of their initial locations may be directed by LBSs' coverage areas dynamicity. Therefore, Spatial Registries will integrate LBSs' registered information dynamic replication and migration using load-balancing techniques based on registries' GCAs and LBSs' information.

ACKNOWLEDGMENT

This work is partially supported by the 7th EU's Framework Programme for Research and Technological Development (http://ec.europa.eu/research/fp7) as part of the ASSET project (http://www.project-asset.com), 2008–2011.

8. REFERENCES

[1] J. Euzenat and P. Shvaiko. *Ontology Matching.* Springer-Verlag, 2007.

[2] T. Erl. *SOA Principles of Service Design.* Prentice Hall, July 2007.

[3] H. Balen. *Distributed Object Architectures with CORBA.* Cambridge University Press, 2000.

[4] R. Monson-Haefel and B. Burke. *Enterprise JavaBeans 3.0, Fifth Edition: Developing Enterprise Java Components.* O'Reilly, 2006.

[5] R. Lee and S. Seligman. *JNDI API Tutorial and Reference: Building Directory-Enabled Java Applications.* Addison-Wesley Longman Publishing, Boston, MA, USA, 2000.

[6] OMG. MDA Guide. Technical Report Version 1.0.1, June 2003.

[7] OMG. Naming Service Specification. Formal Version 1.3, October 2004.

[8] OMG. Trading Object Service Specification. Standard Version 1.0, May 2000.

[9] OASIS. UDDI Spec. Draft Version 3.0.2, October 2004.

[10] M. Bakhouya and J. Gaber. Approaches for Ubiquitous Computing. *Wireless Ad hoc and Sensor Networks*, pages 111–142, 2008.

[11] OMG. Security Service Specification. Formal Version 1.8, March 2002.

[12] OASIS. Web Services Security: SOAP Message Security. Standard Specification Version 1.1, February 2006.

[13] N. Cottin. *Contribution à la sécurisation des échanges électroniques en environnement réparti objet.* PhD thesis, University of Technology of Belfort-Montbéliard, December 15 2003.