Evaluating Location Dependent Queries Using ISLANDS

Marie Thilliez, Yohan Colmant, Thierry Delot

LAMIH Laboratory - CNRS UMR 8530
University of Valenciennes
Le Mont Houy
59313 Valenciennes Cedex 9 France
{firstname.lastname}@univ-valenciennes.fr

Abstract.

The recent emergence of handheld devices and wireless networks has implied an exponential increase of terminals users. So, today, service providers have to propose new applications adapted to mobile environments. In this article, our goal is to present, through a Proximity Electronic Commerce (PEC) application, the query management in ISLANDS (Information and Services LocalizAtioN and Discovery Service), a localization service adapted to such environments. ISLANDS supports location dependent queries such as "retrieve the closest cinema". Therefore, its query evaluator has to determine the location of the query issuer. The originality of our approach relies on the possibility to estimate the issuer localization using the information of nearby participants when the underlying handheld device is not equipped with geo-localization techniques such as GPS. So, ISLANDS may return approximate solutions for location dependent queries.

1. Introduction

The emergence of both handheld devices and wireless networks [VAR00] has implied an exponential increase of terminals users. Today, service providers have to offer new services adapted to mobile environments [BEC00, CHE03]. Due to the mobility of users, the information available in the communication area formed by the juxtaposition of the users networks rapidly evolves and localization services are needed to provide a correct and up-to-date information to users. As existing localization solutions (naming services, trading services, discovery services, …) do not support the constraints in term of distribution, dynamicity and heterogeneity of both terminals and networks imposed by such distributed applications, new solutions have to be defined. In this article, we describe the query management in ISLANDS (Information and Services LocalizAtioN and Discovery Service) [THI03b], an extended directory service proposed to locate data and services in mobile and dynamic environments.

In ubiquitous environments, it is very important for users to be able to query information according to their physical location, such as "where are the music stores close to me ?". Such queries are called Location Dependent Queries (LDQ) [SEY01]. Thus, ISLANDS supports LDQ, the query evaluator has to determine that location and the originality of our approach relies on the possibility to estimate it using the information of nearby participants. Indeed, only few handheld devices are equipped with geo-localization techniques such as GPS today.

The Peer-To-Peer (P2P) architecture highly facilitates the management of the dynamicity inherent to mobile environments [THI03a]. With such an architecture, the information is distributed on different nodes, called peers, corresponding to different devices. As the devices are very heterogeneous (for example, PCs vs. mobile phones), ISLANDS relies on a hybrid P2P architecture [YAN01] where central and light peers are distinguished as presented in Fig. 1. Central peers centralize information and share it with the other peers. So, central peers generally correspond to robust servers whereas light peers correspond to handheld devices. ISLANDS (or at least its query evaluator for light peers) is deployed on every peers which participate to the application. In addition, information managed on a remote peer may also be referenced not to store it on the local peer. This aspect is interesting for light peers which resources are strongly limited. Then, the distribution between the different services is completely transparent for users and queries are automatically forwarded from one peer to another when it is necessary to compute the query result.

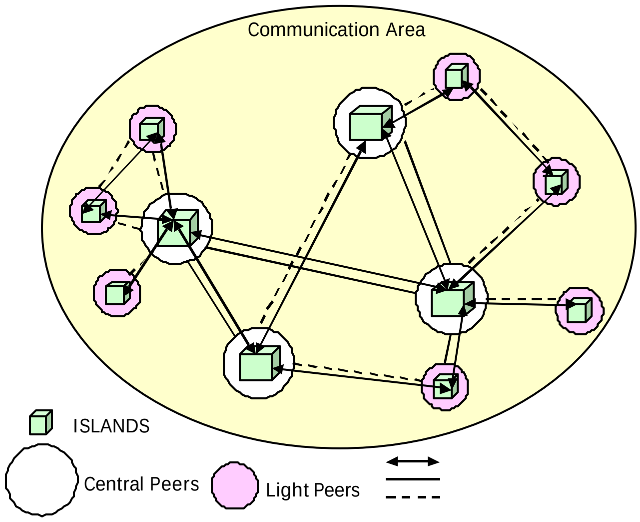

Fig. 1. Hybrid Peer-To-Peer Architecture

2. Location Query Management

2.1. Location Dependent Queries

In mobile applications, a participant may need to retrieve the closest doctor to him/her or the set of fast foods close to him/her. In ISLANDS, in order to select an information according to its location or its proximity, we have introduced three simple and user-friendly operators to verify proximity constraints.

First, the *inside* operator may be used to retrieve several elements located in a same area, for example the doctors located at a particular stage of a building. Secondly, the *closest* operator may be used to retrieve one particular element at the shortest distance from the user who issued the query or from the specified location parameter, for example a user can query the closest fast food to Virgin. Finally, the *close* operator is an evolution of the *closest* operator. Indeed, this operator can be used to retrieve several elements close to the issuer of the query or close to a specified location parameter.

To illustrate the use of the location operators, we present in Fig. 2 an example of query expressed in DSML[1]. DSML (Directory Services Markup Language) is a markup language for representing directory services information in XML. DSML can be used either to represent the information contained in a directory in XML or to query directories using the DSML request format. To support the location operators, it has been necessary to extend the standard DSML request format. The following example represents in DSML the query retrieving "the Fast Foods in the fifty meters around me" using the *close* operator.

---

[1] http://www.oasis-open.org/committees/dsml/

```
<?xml version="1.0" encoding="UTF-8"?>
<dsml:searchRequest dn="o=PeerDirectoryService" xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core">
  <dsml:filter>
    <dsml:and>
      <dsml:equalityMatch name="objectClass">
        <dsml:value>Store</dsml:value>
      </dsml:equalityMatch>
      <dsml:equalityMatch name="type">
        <dsml:value>fast-food</dsml:value>
      </dsml:equalityMatch>
    </dsml:and>
  </dsml:filter>
  <dsml:localization>
    <dsml:close>
      <dsml:location>me</dsml:location>
      <dsml:distance>50</dsml:distance>
    </dsml:close>
  </dsml:localization>
</dsml:searchRequest>
```

Fig. 2. Example of an extended DSML query

2.2. Query Evaluation

One of the main difficulties when evaluating location dependent queries, that is queries containing one of the location operators presented above, is to determine the localization of the user who issued the query. If the query is submitted using a fixed computer, its location is well known and the evaluation of the localization is trivial. On the contrary, it may be very difficult to determine the localization of mobile peers. Indeed, their localization can change at any moment and so potentially has to be computed for each query evaluated. In the case when the localization of the query issuer can not be computed using geo-localization solutions such as GPS, we propose another solution which consists in determining an approximate location in order to evaluate location dependent queries and propose approximate query results. Since ISLANDS is deployed in a P2P environment, the evaluation of the localization is determined using the localization information, usually stored in directory services such as LDAP, of nearby peers. Therefore, our evaluation algorithm computes an approximate localization thanks to the addresses of some referenced remote peers, described in XML as presented in Fig. 3, according to different criteria such as the network connection type (Bluetooth, Adhoc WLAN, …), the date of the last update of the localization information and so on. Then, the set of computed *<localization description, estimated proximity>* couples are used to perform locations matching.

```
<locationDescription>
   <building name = « Building 1 »>
      <stage name = « first »>
         <section name = « North »>
            <room name = « Virgin »>
            </room>
         </section>
      </stage>
   </building>
   <gps>
      <lat> 27.7 </lat>
      <long> -15.1 </long>
      <alt> 197161.4 </alt>
   </gps>
</locationDescription>
```

Fig. 3. An example of the XML representation of the peer location

The architecture of the prototype of the ISLANDS query engine is presented in Fig. 4. Standard DSML Requests are evaluated by the standard DSML Request Evaluator. For location dependent queries, such as the one presented in Fig. 2, a standard query (without location features) is computed. In our example, this query retrieves the set of fast foods. When the query is a location dependent one, the location of the user who issued the query is computed by the issuer location evaluator while the standard part of the query is evaluated. Then, the location query evaluator applies the location operators on the DSML Response. The locations are compared more easily thanks to the XML representation. As concerns the query forwarder, it is used to continue the evaluation of the query on other peers when necessary. Finally, the query result is build by the result composer using the different partial results.

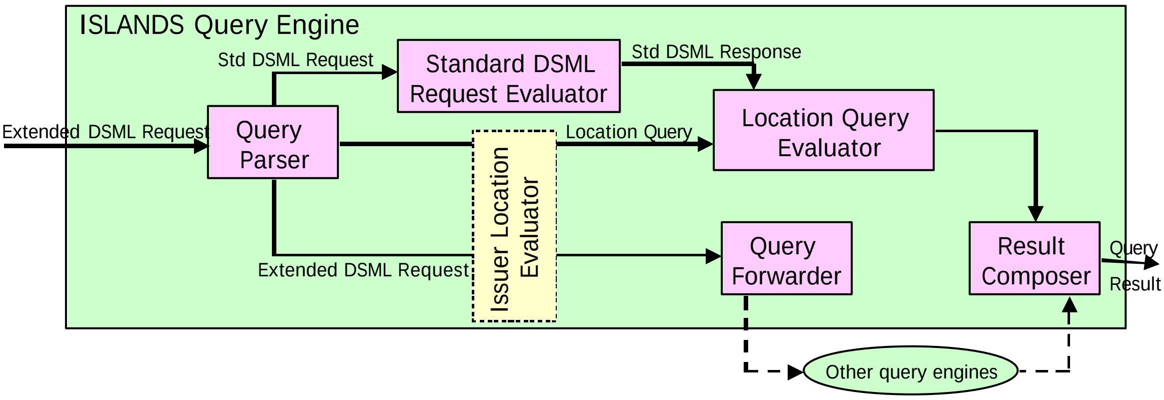

Fig. 4. ISLANDS Query Engine Architecture

3. Demonstration & Prototype

Our demonstration relies on a Proximity Electronic Commerce (PEC) application. In such an application, a user may choose and buy goods depending on his/her preferences and on his/her physical location. First, a potential client, fitted with an handheld device such as cell phone, enters in the commerce zone and then, he/she can send queries in the wireless communication area dynamically formed by the juxtaposition of the different personal networks. These queries are evaluated and the client can retrieve several results such as merchants offers. Then, if the client is interested in one or more specific offers, he/she goes to the merchant and pay his/her goods thanks to his/her handheld device.

Our prototype is developed in Java. Central peers are represented by laptops and PCs whereas pocket PCs Compaq iPAQ H3950 are used as light peers. Those different peers are connected with each other using either Bluetooth or Wifi technologies. The query engine deployed on central peers contains all the modules described in Fig. 4. On the contrary, on light peers, the query engine only provides a query forwarder and the issuer location evaluator. Besides, data are stored either in directory services for central peers and in DSML files for light peers. Distribution transparency is assured using the Java Naming and Directory Interface (JNDI) API. Distribution indexes are also generated to guarantee good performances when evaluating distributed queries. The evaluation of the localization relies on the use of a simplex algorithm. Since the majority of software pieces used to compute this algorithm are designed to solve very general problems, they can not be used on handheld terminals due to consumed resources. So we have implemented a resolution algorithm dedicated to our problem in order to minimize the overhead required to evaluate location dependent queries on light peers.

Finally, in our prototype, users do not express directly their queries using DSML. In fact, those queries are automatically generated thanks to the choices performed by users using the ISLANDS query interface.

The choices proposed to users in the query interface are dynamically parameterised thanks to an XML file broadcasted in the commerce zone. This file contains the available search criteria as well as parameters used to establish the correspondences between the parameters of the query and the directory information. Fig. 5 illustrates the choices performed through the query interface to generate the DSML request presented in Fig. 2.

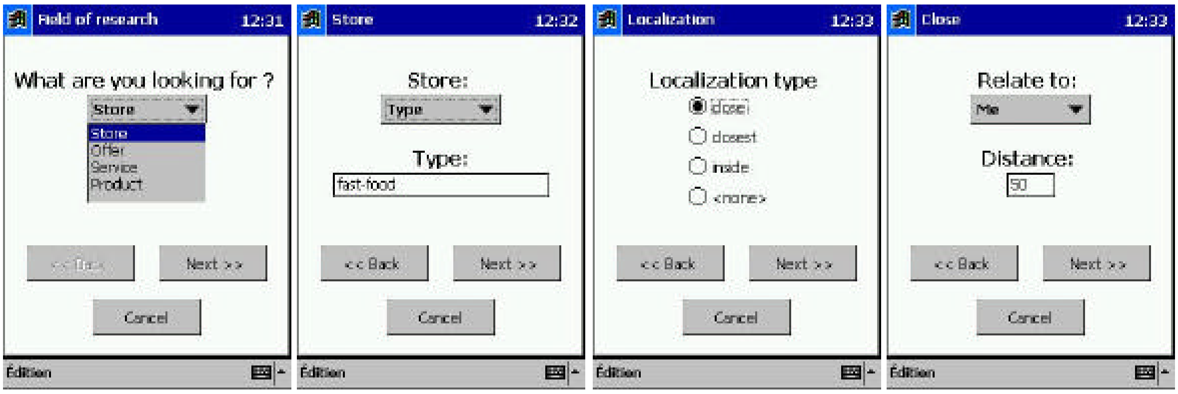

Fig. 5. Query interface

4. Conclusion & Perspectives

In this article, we have presented the query evaluator of the ISLANDS service. We have notably focused on the management of location dependent queries. The originality of our approach resides in the location process of the user which is required to evaluate such queries. Indeed, using ISLANDS, the user location can be estimated since today only few handheld devices provide GPS localization techniques.

In the near future, we have to propose mechanisms to determine precisely when the location algorithm has to be executed. For the moment, the issuer location evaluation is computed for each location dependent query. However, this may be very penalizing for light peers (query execution time, resource consumption) and mechanisms to detect whether the location information has become obsolete are needed.

5. References

[BEC00] M. Bechler, H. Ritter, J. H. Schiller, *Quality of Service in Mobile and Wireless Networks: The Need for Proactive and Adaptive Applications*, Int. Conference on System Sciences (HICSS), 2000.

[CHE03] Y.-F. R. Chen, C. Petrie, *Ubiquitous Mobile Computing*, IEEE Internet Computing, Vol. 7, N° 2, 2003.

[SEY01] A. Y. Seydim, M. H. Dunham, V. Kumar, *Location Dependent Query Processing*, Proceeding of MobiDE, 2001.

[THI03a] M. Thilliez, T. Delot, S. Lecomte, N. Bennani, *Hybrid Peer-to-Peer Model in Proximity Applications*, Int. Conf. on Advanced Information Networking and Applications (AINA), 2003.

[THI03b] M. Thilliez, T. Delot, *ISLANDS : an Information and Services LocalizAtioN and Discovery Service*, Submitted to the BDA, 2003.

[VAR00] U. Varshney, R. Vetter, *Emerging Mobile and Wireless Networks*, Communications of the ACM, Vol. 43, Issue 6, 2000.

[YAN01] B. Yang, H. Garcia-Molina, *Comparing Hybrid Peer-To-Peer System*, Int. Conf. DataBases on Very Large DataBases (VLDB), 2001.