Submitted for publication to the IEEE

Personal Service Areas for Location-Based Wireless Web Applications

Ariel Pashtan[1], Andi Heusser[2], Peter Scheuermann[2]

[1]Aware Networks
Buffalo Grove
IL 60089, USA

[2]Dept. of ECE,
Northwestern University,
Evanston, IL 60208, USA

ariel@awarenetworks.com, aheusser@gmx.ch, peters@ece.nwu.edu

Abstract

Location-based wireless services enable mobile users to access Web-based information about resources in their immediate vicinity. We describe in this paper how mobility elements of context such as direction and speed of travel are used in the formation of a personal service area. A tourism scenario is presented to provide the motivation for using dynamic context, and we describe the architecture of CATIS, an experimental context-aware tourist information system that leverages XML technologies and Web services. The CATIS context and profile manager adapts Web-based information that is delivered to tourists based on their personal service area and user preferences. We review mobile location area concepts and describe a context-sensitive algorithm for delineating a mobile user's personal service area. Web services performance is dependent on the underlying databases, and we show how response time is improved through a layered caching scheme. We end by addressing future network scalability approaches.

Keywords: location-based services, Web services, mobile Web, wireless, context-aware, location area, caching.

1. Introduction

With the advent of mobile users' location tracking capabilities, location-based wireless services are positioned as a major motive for the development of future wireless systems. These types of services enable mobile users to access information relevant to their context. What makes context-aware applications much more complex in the realm of mobile users are the spatio-temporal variables that make up the dynamic context of the user. While location is a key element of the user context, other applicable elements that affect a location-based service include direction of travel, speed, time of day, personal preferences, and terminal type. One of the main considerations in the design of location-based services is their ability to determine a geographical area of relevance to the mobile user. We describe a novel algorithm that tracks user location, derives user direction and speed of travel, and consequently determines a service area's shape in a computationally efficient manner. A related layered caching scheme for storing environment information was developed to improve user response times. Such a personalized definition of a service area can enable, for example, tourist information services to wirelessly deliver city information to tourists' mobile terminals that is better attuned to the users' mobility context.

2. Personalized Tourist Services

An important aspect of context-awareness is the system's ability to deliver information that is personalized, i.e., adapted to a user's specific needs. Tourism is an area that will benefit from context-awareness as demonstrated by research projects in mobile context-awareness ([1],[3] & [7]). The FIPA (Foundation for Intelligent Physical Agents) standards organization [6] and m-ToGuide [5], a project sponsored by the European Commission's 5th Framework Program, also targeted tourism applications as a new service for mobile users where context plays a significant role in the delivery of pertinent information to tourists on the go. In general, context refers to information about the terminal platform, the user, and the surrounding environment ([2], [9], [10]). User context typically refers to the user's personal preferences for information content, e.g., type of cuisine if searching for a restaurant. Environment context includes elements such as the user's location, current time of day, location of landmarks, and the prevailing weather.

Mobile users traveling in a city can leverage context-awareness to enhance their touring experience by receiving tailored information that familiarizes them with landmarks and events in their vicinity. Wireless terminals that serve mobile users are not very conducive to interactivity (e.g., small screens, space constrained keyboards). Context-awareness can improve user interaction by knowing a-priori the user's mobility situation (e.g., direction and speed of travel), personal preferences, information interests, and environment conditions, so that the user doesn't have to specify these constraints, and information delivery is automatically adapted to his circumstances.

In the scenarios we considered, the tourist can specify categories of information interests, and the tourist's wireless terminal will display related location-based information. Mobility context, in particular, can be a key element in adapting the delivered information to a relevant geographic area as described in the following scenario.

It is evening and Sue is in the New-York metropolitan area. Sue is traveling in a rented car. Based upon the time context, her wireless terminal automatically initiates a request to get restaurant information for eating dinner. The network-based tourist information service is queried for restaurants in the neighborhood of Sue. In addition to considering the user preferences for restaurants and the type of terminal that she is using, the tourist information service tracks her direction and speed of travel. Restaurants that are within a predetermined range of her location, for example up to 30 minutes of driving, and in her detected general direction of travel, will be suitable candidates as they will not divert her from her overall trajectory. The derived restaurant list is transformed into Web pages in a format appropriate for display on her wireless terminal.

This information adaptation to the user's detected trajectory and speed of movement provides for a differentiated service not typically found in location-based services. Typically, mobile tourist information systems, e.g., [3], generate tour suggestions and display information that is relevant to a stop on the tour when the tourist is detected at the stop. Other systems, e.g. [7], display information that is relevant to the tourist's current location irrespective of any tour. Still others, e.g., [8], allow the tourist to select an area's name or zip code, and corresponding points of interest are then displayed on the user's terminal.

3. Context-Aware Services Architecture

CATIS (Context-Aware Tourist Information System) is an experimental context-aware wireless Web information system implemented by the authors. The adaptation of tourism information performed in CATIS is done based on context information, including user preferences, retrieved from a context and profile manager network element. The CATIS architecture, shown in Figure 1, is Web service-based and was implemented using both Java Web services and Microsoft's .NET Framework. The major architecture components include:

- A thin client terminal that hosts a Web browser. The client terminals send location updates to the network on a periodic basis.
- An application server that delivers web content prioritized according to the user's pre-specified preferences. The application server formats the generated information content and adapts it for display on the user's client terminal.
- A context and profile manager (CPM) that keeps track of the user's dynamic context (e.g., location, direction of travel, speed, and wireless terminal features) as well as static preferences (e.g., preferred cuisine). User context and preferences are tracked in user profiles stored in an XML database. The CPM is also responsible for querying Web services and for filtering Web content data according to user profiles context.
- A UDDI (Universal Description, Discovery and Integration) services directory that provides users with a centralized registry of tourist information services (e.g., a restaurant finder service). The UDDI specifications describe service-related information and provide a query and update API to access service information in the registry.
- A collection of Web services that deliver tourist content. Three different Web services were implemented so far; these allow for finding restaurants, hotels, and entertainment parks. Each Web service is a wrapper around a database and provides functions to query and retrieve service information. The functions are described in a Web services description language (WSDL) documents that are published in the UDDI server.

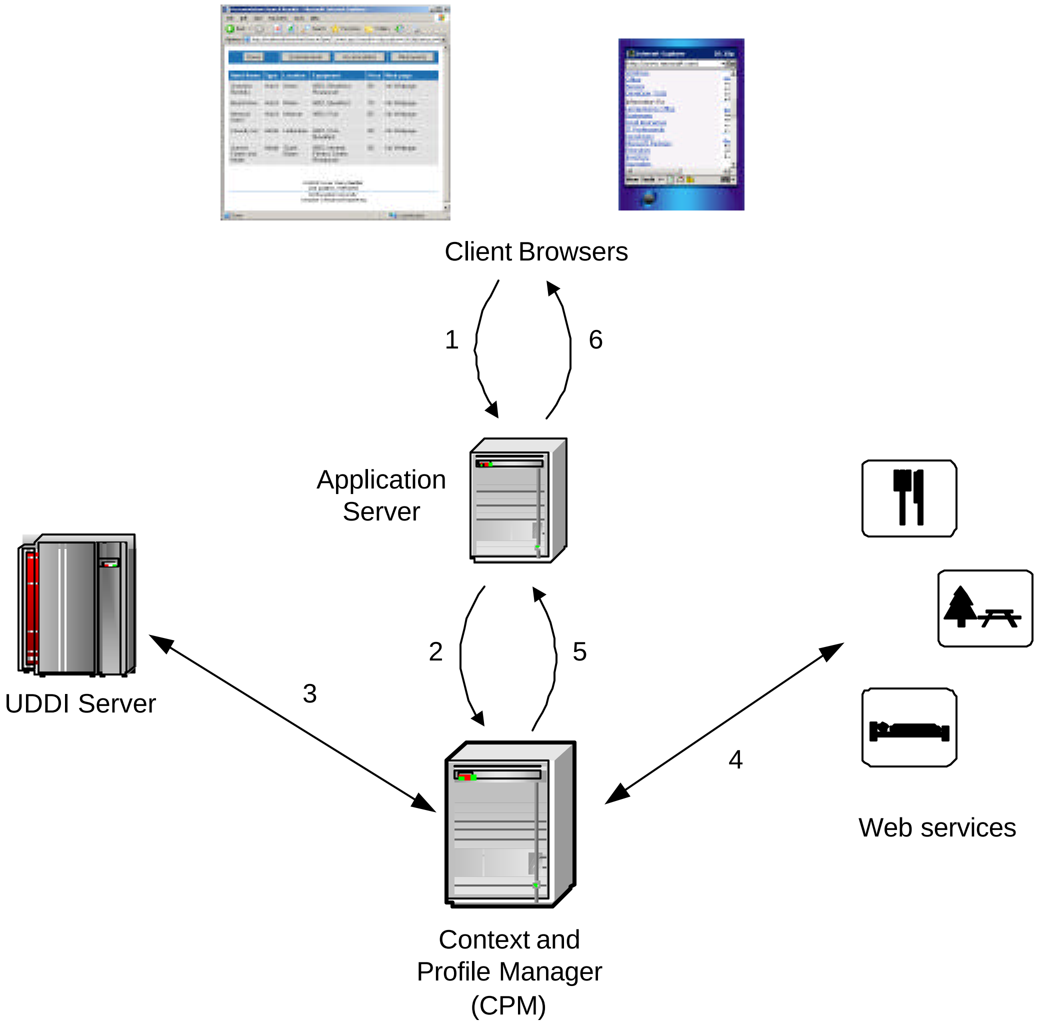


Figure 1 CATIS Architecture

Given the Web's intrinsically distributed and heterogeneous nature, communication mechanisms must be platform-independent and as lightweight as possible. These mechanisms were realized in the simple object access protocol (SOAP) [4], an XML-based protocol for messaging and remote procedure calls, which is used here on top of HTTP for communication between all network elements. While the client terminals use HTTP for browser-based access to the application server, they also use SOAP for relaying context information to the CPM.

To register with the system, a user logs on to the application server and enters his preferences (e.g., restaurant likings) by selecting appropriate fields in profile forms. These preferences are then forwarded for storage in the CPM's database where they are aggregated in user profiles. If the client terminal possesses a GPS receiver then the client sends location updates to the CPM at regular intervals. We focus here on GPS-based methods where user location is available at the mobile terminal, although other methods for location tracking could be used such as network-based triangulation that relies on multiple cell sites for determining a terminal's position. User direction and speed are computed from the user's tracked location history. Latest values of user dynamic context are stored in the same user profiles where user preferences are kept.

When a tourist seeks local information, for example, about restaurants in her vicinity, she connects to the application server to request the information (step 1 in Figure 1). The application server forwards the query to the CPM (step 2). The CPM sends an inquiry to the UDDI Server to get the addresses of the available restaurant Web services (step 3). The CPM then sends a request to the identified Web services with a specification of a service area that surrounds the user (step 4). The Web services search their databases for the appropriate resources and filter out those that do not fall in the specified area. The Web services return an XML list to the CPM. The CPM filters the XML list according to the user's context and augments the restaurant information with user and environment context such as the user's cuisine preference level, distance and direction to the restaurant. The following code shows the restaurant XML information augmented with context data in the blue elements:

```
</Restaurants>
  <Restaurant>
    <Name>Oceanique</Name>
    <Street>505 Maine St</Street>
    <Cuisine>Caribbean</Cuisine>
    <PriceLow>10</PriceLow>
    <PriceHigh>15</PriceHigh>
    <Latitude>42.034052</Latitude>
    <Longitude>-87.67793</Longitude>
    <Open>11:00</Open>
    <Close>22:00</Close>
    …
    <CuisinePreferenceLevel>3</CuisinePreferenceLevel>
    <Distance>0.33</Distance>
    <Direction>-76.75</Direction>
  <Restaurant>
</Restaurants>
```

This context-enriched information is then sent to the application server (step 5). The application server uses an XML style sheet to generate a prioritized restaurant list in the appropriate markup (e.g., HTML) for browser display and forwards it to the client terminal (step 6).

4. Mobile Location Areas

As described above, the CPM is required to determine a relevant service area for finding appropriate resources in the user's vicinity. A related concept of a wireless location area was defined in cellular networks for the purpose of tracking mobile users so that calls can be efficiently established when mobile users travel within a network or roam to other networks. Various methodologies (e.g., two-tier and multi-tier schemas) have been proposed and a detailed survey is presented in [11]. Generally, the main trade-off is between the frequency of updates when mobile users cross the cell boundaries and the search time latency when a particular user needs to be located. Cellular networks decrease the cost of user tracking by forming groups of cells that are called *paging location areas*. A mobile user updates his location in a location registry database only when he crosses boundaries between location areas. To find a user, all cells in a given location area are queried in what is referred to as *paging*.

A complementary direction of research in recent years has focused on designing databases which manage spatial-temporal data of large numbers of moving objects. This field is known as *moving objects databases* (MOD) [13] & [15]) and it addresses the unique modeling requirements of location data, access methods of moving objects, as well as the processing of novel types of spatial-temporal queries. MOD databases have resorted to using dynamic attributes, i.e., attributes that change dynamically over time without any update operations, and provide for queries that can simultaneously handle space and time constraints, including projections of the future. An example such query would be "retrieve the fleet's vehicles that will be in the town of Northbrook between 3PM and 5PM". In this query, the *specified location area* is the town of Northbrook. An answer to such a query requires the database to keep a model of user trajectories, and of events, e.g., accidents or traffic congestion, that may cause a reevaluation of the query so that the presented data remains accurate. In order to determine the boundaries of the areas of interest for preferred services one may use a MOD-based tracking system. However, MOD databases do not have the capability of incorporating explicitly the user's preferences and all the mobility parameters in their query processing framework. The *personal service area* elaborated in the following is a context-sensitive dynamic area that is automatically determined from the user's current mobility circumstances.

5. Personal Service Area

For a traveling tourist, a key issue is how to find out her relevant surrounding region, as this will affect the choice of local resources displayed on her terminal. This local service area is determined by several factors that are either obtained directly or calculated from the user's context. Besides location, additional mobility elements of context that affect the displayed information include direction and speed of travel since, for a given travel time, fast moving users can reach places that are further away.

CATIS, tracks a user's previous locations and from these the system can compute the user's direction of travel as well as average speed. We devised a sliding-grids window methodology in which location, direction and speed of travel are taken into account to deliver a list of recommended points of interest (POIs). For example, POIs in the opposite course of travel, and beyond a certain distance threshold, will not be listed.

Our sliding-grids window algorithm uses a two-step approach whereby first the tourist's pertinent service area is determined and then a list is generated based on a preferred maximum distance of travel to a POI. This approach relies on detecting the user's direction of travel and leveraging a pre-partition of the metropolitan area into a grid of square-shaped cells where the user is traveling as shown in Figure 2. The size of the cells is pre-defined so that the time to traverse a cell is about the time the user is willing to travel to a listed POI at a given speed (e.g., half an hour). The cell's size will determine the number of cells that need to be retrieved from a Web service following a user query. In Figure 2, two successive queries are issued by the user, first at *Time 1* and then at *Time 2*, and the shaded cells are the corresponding retrieved cells. The *personal service area* is derived by the intersection of the retrieved cells with a circle that represents the maximum distance that the user is willing to travel to a listed POI. This algorithm proves to be more efficient compared to an exhaustive computation of distance over all POIs, and also adapts the list of relevant POIs to the user's direction of travel.

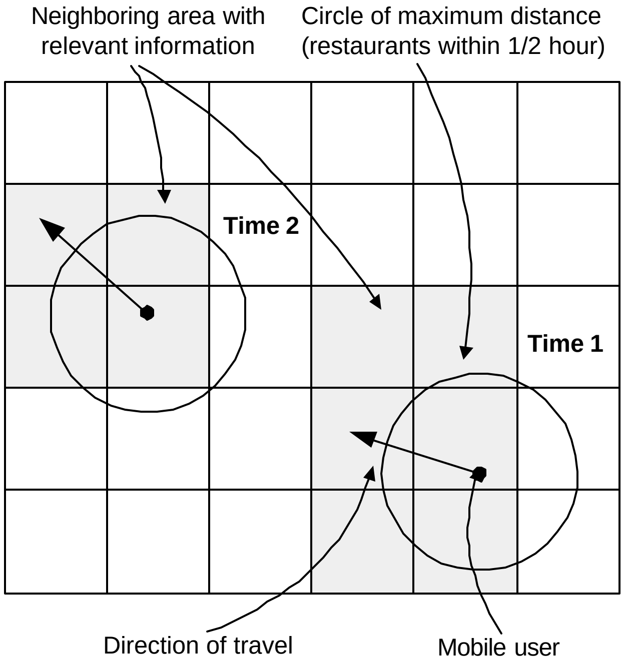


Figure 2 Personal service area based on direction, speed, and distance to resource

To enable a service area selection that is attuned to the user's speed, we refined the metropolitan grid of POIs into three levels of overlapping grids, one grid per level (Figure 3). This grid layering is a practically motivated structure which provides a good balance between the Quadtree (a tree-based data structure) and the grid-files [12], for the purpose of efficient cache utilization.

Each POI, as well as the user's location, falls into exactly three cells, one at each level. At level 1 the grid cells have a side length of W miles, at level 2 they have a side length of S miles, and at level 3 the cells have a side length of F miles, where $W<S<F$. Smaller grids are used for slower moving users, and the selection of the sizes depends on typical speed limits in the metropolitan area. Our current implementation has a grid partition as shown in Figure 3 where grids of size W are for walking users, grids of size S are for slow moving vehicles, e.g., 30 miles per hour, and grids of size F are for fast moving vehicles, e.g., 60 miles per hour. This scheme could be extended to include more grid levels, when higher accuracy is desired in the service area computation.

From the user's speed, the CPM determines the appropriate grid level, and from the user's direction, the CPM determines the list of cells that should be used in the personal service area computation. Once the cells' POI information has been retrieved from the Web service, a filtering step takes place to select only those POIs that are within a pre-determined distance, e.g., within half an hour of travel from the user, at the current user speed.

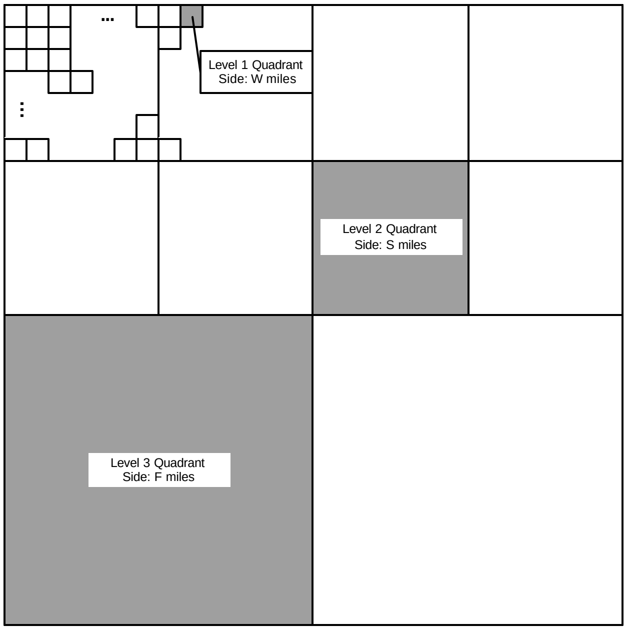


Figure 3 Overlapping grid layers used in determining personal service areas

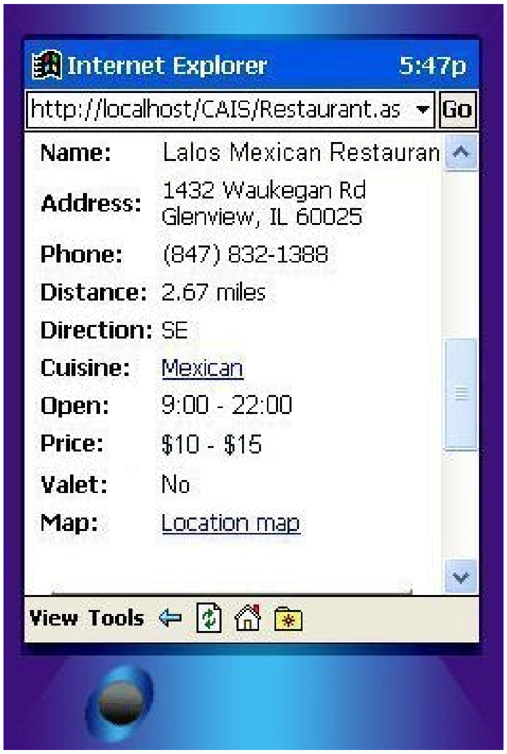
The list of restaurants that is displayed on the user's terminal is ordered by cuisine preferences first, and then by distance within the user's personal service area as shown in the PDA-like screen in Figure 4. The user can select any of the restaurants in the list to get more detailed information shown in Figure 5.

Name: Lalos Mexican Restaurant
Cuisine: Mexican
Distance: 2.67 miles
Direction: SE

Name: P.F. Chang's China Bistro
Cuisine: Chinese
Distance: 5.46 miles
Direction: S

Name: Lupita's Restaurante
Cuisine: Mexican

Figure 4 Matching restaurant list



http://localhost/CAIS/Restaurant.as

Name:	Lalos Mexican Restauran
Address:	1432 Waukegan Rd Glenview, IL 60025
Phone:	(847) 832-1388
Distance:	2.67 miles
Direction:	SE
Cuisine:	Mexican
Open:	9:00 - 22:00
Price:	$10 - $15
Valet:	No
Map:	Location map

Figure 5 Detailed restaurant information

6. System Performance

The CATIS restaurant Web service includes about 2200 restaurants from the greater Chicago land area. We applied performance tests, described in the following, to evaluate system response to single user queries. The tested system consisted of three Microsoft Windows-based Pentium 4 PCs, one for the application server, the second for the CPM and UDDI directory, and the third for the Web service. The PCs communicated over the Internet.

Web Services Performance

The CATIS restaurant Web service prototype uses the Xindice native XML database. Our tests revealed that each call from the CPM to the Web service takes between 5 and 15 seconds. When we used a Microsoft Access database, response time improved to between 1 and 5 seconds.

The same tests were repeated when the Web service did not query a database. This allowed us to measure the SOAP messaging overhead across the network. In this latter case, the Web service response time was typically below 1 second. While network delays can impact response times, we concluded that the Web services performance is affected mainly by the type of database used to store Web service data.

Caching Solutions

As previously described, the personal service area is specified in terms of cells that cover a geographical area in the vicinity of the mobile user. In an area visited by many mobile tourists, e.g., a city, many personal service areas may overlap. As described above, each Web service request can take up to 15 seconds. In this case, it would be beneficial to cache the restaurant information as many users may have similar requests.

The CPM caches the Web services information in the three-layered grid structure shown in Figure 3. When different mobile users make successive requests to the CPM in the same geographical area, rather than issuing successive Web service calls for different requests in the same geographic area, the CPM retrieves the requested POI information directly from its cache. Without caching, it can take on average 1 minute to display a restaurant list (Xindice was the Web service database). With caching, the response time is reduced to between 2 to 10 seconds, depending on the number of restaurants in the area.

Since restaurant data does not change frequently, we did not have to deal with cache consistency issues. Nevertheless, the CPM does refresh its cache on a periodic basis to allow for restaurant information changes when they do occur. Unlike restaurant data, dynamic user context such as location is stored in the respective user profiles and reflects therefore the most recent measured value.

7. Future Considerations

Currently, the CATIS CPM uses geographic distance as one of the sorting criteria for displaying the list of restaurants. However, two objects that are a certain distance *d* apart may not be reachable via road segments that are of distance *d* since there may be a railway crossing between them or another physical barrier. This information was not available in the restaurant Web services databases. We intend to explore if this information could be obtained from a commercially available GIS software package linked to our system.

We plan to study in the future different cache replacement policies that will take into account the type and size of Web services data. In addition, we intend to investigate various cache refreshment policies for dynamic information such as the number of rooms available in hotels, or the wait time in restaurants, provided this information is readily available.

As a metropolitan area can have many thousands of mobile users, one needs to consider additional scalability features beyond caching. In particular, it appears that a distributed architecture would fit very well into this scenario. All CATIS components including the application server, the CPM and the Web services could be replicated and distributed. One particular scheme that has proven very successful with scalable distributed file systems could potentially also be used in our framework [14]. Namely, when an application server or CPM becomes overloaded its contents could be split by adding a new server or CPM. Furthermore, new servers and CPMs will be acquired only if the overall system utilization does not drop below a specific threshold.

Finally, other context variables such as the mobile user's current transport mode could be considered too for information adaptation purposes. For example, if a user is driving a car, it makes sense to search for Web services that can deliver information in audio form. Automated approaches could be investigated to detect user situation changes and decide when to activate corresponding Web information services.

Acknowledgments

We would like to thank Agnes Voisard and Goce Trajcevski for their earlier comments.

References

[1] G. Abowd et al., "Cyberguide: a Mobile Context-Aware Tour Guide", Wireless Networks 3, pp. 421-433, 1997.

[2] G. Chen and D. Kotz, "A Survey of Context-Aware Mobile Computing Research", *Technical Report TR2000-381*, Dept. of Computer Science, Dartmouth College, 2000.

[3] K. Cheverst et al., "Experiences of Developing and Deploying a Context-Aware Tourist Guide: The GUIDE Project", ACM MOBICOM 2000.

[4] F. Curbera et al., "Unraveling the Web Services Web," IEEE Internet Computing, March/April 2002.

[5] EC 5th Framework Program, "Mobile Tourist Guide", IST-2001-36004, 2003, http://dbs.cordis.lu/.

[6] FIPA, "FIPA Personal Travel Assistance Specification", 2001/08/10, www.fipa.org.

[7] A. Hinze and A.Voisard, "Location and Time-Based Information Delivery in Tourism", Proc. of the Intl. Conf. on Spatio-temporal Databases (SSTD3), Lecture Notes in Computer Science N. 2750, Springer Verlag, 2003.

[8] Metromix.com, 2004, http://metromix.chicagotribune.com/ .

[9] A. Pashtan and J. Yanosy, "An Adaptable Services Framework", 5th WWRF Proc., Mar. 2002.

[10] A. Pashtan et al., "Adapting Content for Wireless Web Services", IEEE Internet Computing, Sept. 2003.

[11] E. Pitoura and G. Samaras, "Locating Objects in Mobile Computing", IEEE Transactions on Knowledge and Data Engineering, 13(4), pp. 571-592, July 2001.

[12] H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, 1990.

[13] G. Trajcevski, P. Scheuermann, "Triggers and Continuous Queries in Moving Objects Databases", 6th International Workshop on Mobility in Databases and Distributed Systems, (DEXA'2003), Sept. 2003.

[14] R. Vingralek et al., "Distributed file organization with scalable cost/performance", Proc. of the ACM-SIGMOD Int. Conf. on Management of Data, May 1994.

[15] O. Wolfson, "Moving Objects Information Management: The Database Challenge", Proc. 5th Workshop on Next Generation Information Technologies and Systems (NGITS'2002), June 25-26, 2002.