Combining Even Notification Services and Location-based Services in Tourism

Annika Hinze⋆, Agnès Voisard⋆⋄
{hinze,voisard}@inf.fu-berlin.de

⋆ Institute of Computer Science
Freie Universität Berlin, Germany

⋄ Fraunhofer ISST,
Berlin Germany

Abstract

Today's mobile devices allow end users to get information related to particular domain based on their current location, such as the fastest route to the nearest drugstore. However, in such Location-Based Services (LBS), richer and more targeted information is desirable. In many applications, end users would like to be notified about relevant events or places to visit in the near future according to their profile. Last but not least, they do not wish to get the same information many times unless they explicitly asked for it.

In this paper, we describe our system, TIP (Tourism Information Provider), which delivers various types of information to mobile devices based on location, time, profile of end users, and their "history", i.e., their accumulated knowledge. The system hinges on a hierarchical semantic geospatial model as well as on an Event Notification System (ENS).

1 Introduction

Today's mobile devices allow end users to get information that belong to a particular domain based on their current location. Examples include the address of the nearest drugstore or even the fastest route to it. However, in such Location-Based Services (LBS), more complex and targeted information is desirable, such as information based on the present time, or on the near-future, or on a particular user profile. Moreover, in most applications, end users would like to be notified about relevant events or places to visit in their domain of interest, i.e., according to their profile. Last but not least, end users do not wish to get the same information many times unless they explicitly asked for it.

Delivering different types of information based on location, time, profile of end users, and their ”history” - which leads to accumulated knowledge - is concerned with many issues that range from high level profile definition languages to wireless transmission of information. A whole range of applications concerns scheduling assistance, i.e., the best way(s) to combine time and location. Take for example a set of ”mobile” events such as a tennis tournament, a film festival, a trade fair, or a conference. An end user would like to optimize his or her path through events/places (i.e., company booths or tennis courts or conference rooms) he or she is interested in. An intelligent system would deliver a list of options in terms of paths that combine the time and the location of the visitors as well as their interest and their possible priorities (some events, e.g., an invited talk, should not be missed). Such applications fall under the general topic of combinatorial optimization which is not the focus of the present paper.

Instead we are concerned here with applications involving mobile end users who already acquired knowledge on a particular area and who would like to get more information at a certain location, at a certain time, on particular topics and to relate pieces of information (current and past) with each other. A typical application is that of a tourist walking around in a region. If this person already visited a place he or she does not want to be reminded of basic historical facts related to that place. He or she would rather like to get further information related to the place in question. This is similar to Internet usage, where successive clicks provide more and more information. However, to our knowledge, there is no available systems making connection with pages visited before (history) and offering the notion of personal profile. Besides, a mobile device such as a cellular phone or a Personal Digital Assistant (PDA) has particular requirements. First, the amount of information that can be presented is limited. This means that the information should be conveyed in a simple form. Web page like layout is unacceptable. In addition, the computing power of such a device is restricted and many operations as well as rendering need to be performed elsewhere.

The information delivered to the users is extracted, and then combined, from various databases, such as a spatial databases storing maps and other information, an event database (e.g., conference or film time table), referred to as *scheduled event database* in the following. However, we are concerned with applications that go beyond pure database querying. In the applications considered here, end users get information they did not explicitly asked for. This push behavior comes from one of the following two situations: (1) Based on a certain profile, time, or location, targeted information is automatically extracted from the various databases, combined, and delivered to the user; or (2) External events not stored in a database (e.g., cancellation of a presentation) but of relevance for the user are sent.

A solution that comes to mind is to extend an Event Notification System (ENS) or an active database. Event notification services, or alerting services, are increasingly used in various applications such as digital libraries, stock tickers, traffic control, or facility management. The idea is to react to a particular event by taking a certain action. Combining ENS and LBS in this context means considering the following features:

- Personal profiles.
- History-dependent information delivery.
- Spatial/spatio-temporal information organized hierarchically.
- Spatial/spatio-temporal[1] information organized in a semantic network such that facts can be related to each other.

In this paper, we offer an environment that combines the structured information, the various profiles, and all possible events. We use operators as means to navigate though the semantic network of information. A typical example of operators is proximity in several dimensions, as in ”*near*” this position, ”*near*” in time, or ”*near*” in a semantical context. This is referred to as *proximity search* (in various dimensions) in the following.

Several system exist that process context-aware and location-based information. Two major approaches can be distinguished: services for outdoor experiences and services focussing on indoor activities. Examples of outdoor services include tourist guides, such as Nexus [32], Guide [6], Crumpet [25], and the Oldenburg guide [3]. Examples of indoor systems include museum guides, e.g., Hippie [23], Electronic Guidebook [15], and Rememberer [9]. Substantial consideration has been given to database modeling and querying of moving objects (e.g., in [28, 31, 30, 10, 12], to cite only a few). [28] distinguishes three query types: instantaneous (evaluated once at definition), continue (evaluated regularly after definition), and persistent (sequence of instantaneous queries evaluated at every database update). However, special mechanisms for profile definition and event filtering are not discussed in this approach. In the context of location based services, continuous queries are of particular interest. Most systems only consider only queries regarding the changes in user location (e.g., in the Oldenburg guide). Extensive research on location-based services focuses on the technical challenges of interaction with the moving users or the spatiotemporal observation of moving objects in general.

[1] Note that considering pure temporal information does not make sense here.

In most systems, the context is merely the user location measured either at certain access points (e.g., in Electronic Guidebook) or at a given location (e.g., in Nexus). That is, continuous queries, or profiles, only consider the user location. Additional contexts to consider would be, e.g., user interest, local time, and technical communication parameters. However, only few systems encompass the notion of events or profiles. In the Guide system, tourists are actively informed about general events, such as the opening hours of a museum. This information is broadcasted to all user of the system, whether they are interested or not. Similarly, users are informed about each available sight. In the Nexus predecessor, VIT, keyword profiles are used to select the information sources by topic (similar to advertising). In Crumpet, the information delivered on request is sorted according to user interests. Additionally, pro-active tips are delivered to interested users. The user's interest is defined in terms of profiles that are gradually adapted to the user based on specific user feedback. The Oldenburg guide uses continuous queries regarding the the spatiotemporal location of the users as profiles. Information about (moving) objects is delivered depending on its importance to the user, where importance depends on the spatial distance between object and user.

A large variety of event notification systems has been implemented, as for example SIFT [35], Salamander [21], Siena [4], OpenCQ [20], Elvin [26], Gryphon [29], NiagaraCQ [5], LeSubscribe [24], Hermes [8], and A-TOPSS [18]. The few ENS used in the context of tourist information have been focussing on traveller support in travel planning and route guidance, e.g., in the Genesis system [27]. For event notification systems, we distinguish document-centered approaches from event centered approaches. In document centered systems, the events are the publication of a new document. These systems are called publish/subscribe systems or systems for Selective Dissemination of Information (SDI). Examples are SIFT, NiagaraCQ, Hermes and A-TOPSS (see above). In most event centered systems, the message reporting an event is formatted as attribute-value pairs.

As far as profile definition languages are concerned, it is common to distinguish subject-based from content-based filtering. In the former, the profiles are subjects or topics in which the events are ordered. Content-based filtering allows for a more fine grained filtering, better adapted to the system users. The profile definition languages used in content-based ENS mirror the characteristics of the systems: document-centered systems use concepts from Information Retrieval, such as keyword-based search (e.g., SIFT and Hermes) or XML query languages (e.g., NiagaraCQ and A-TOPSS). Event-centered systems use Boolean combinations of predicates on attribute-values. Additionally, several of these systems carry the notion of composite events that are temporal combinations of events, such as a sequence. Composite events have been extensively studied in active databases [11, 16, 36]. Only few sophisticated profile definition languages using composite events have been implemented [19, 14]. A combination of concepts from active databases and event notification systems are Event Action Systems (EAS). In such systems (e.g., in Yeast [17]) the user profile defines an action that is triggered by a certain event. In several aspects our work is very close to EAS, for example, in the detailed action definition.

A few remarks are noteworthy. First, in this paper we are not concerned with technical details such as the transmission of information from the mobile device to the server and vice versa. The fact, for example, that the global positioning system (GPS) is not operational inside a building and that the system may need to be coupled with a wireless local area network (LAN) - as it was done for instance recently in a Tokyo museum - is not our concern here. Positioning systems for indoor activities are infrared Active Badges [33], the ultrasonic-based Bat system [34], or Radar based on wireless LAN [1]. Location systems for outdoor activities may use a wireless phone location system such as the US E911 [7] or GPS [22]; for an overview of location systems see [13]. [2] studied the impact of technical parameters such as computing power of the client, speed and throughput of the network connection on the on the client interaction. In this paper, we are not concerned with privacy issues either even though we these issues are definitely crucial. We assume, however, that end users can send their location to the server as well as selected information from their history. This means that users' history is stored on the device. Finally, the way the information is eventually conveyed to the user - through drawings, texts, voice, short messages services (SMS), in general unified messages - is not our focus here.

This paper is organized as follows. Section 2 presents our reference application, which is a description of the information delivered to a tourist in the city of Berlin. Section 3 is concerned with the combination of LBS and ENS. It presents the major concepts as well as the architecture of our system, TIP, and a description of the information organization. Section 4 concerns the matching of profiles and events. Finally, Section 5 presents our conclusion and our ongoing and future work.

2 Example Application

This section describes a typical application in the tourism domain, which will serve as a reference in this paper. We first briefly describe our application scenario and then detail the information that the mobile user gets on the mobile device.

2.1 Application: A tourist in Berlin downtown

Let us consider a tourist in Berlin on Saturday, February 22, 2003, with the following profile. This person has been visiting major European cities for over a week. She is for the first time in Berlin and is interested in (i) basic architectural facts in the city; (ii) wall-related facts; and (iii) Jewish history. This person has a few items to visit on her agenda but would like to be notified whenever something of interest w.r.t. both her location and profile occurs. Besides, she gets hungry after a while and has quite particular requirements in terms of restaurants.

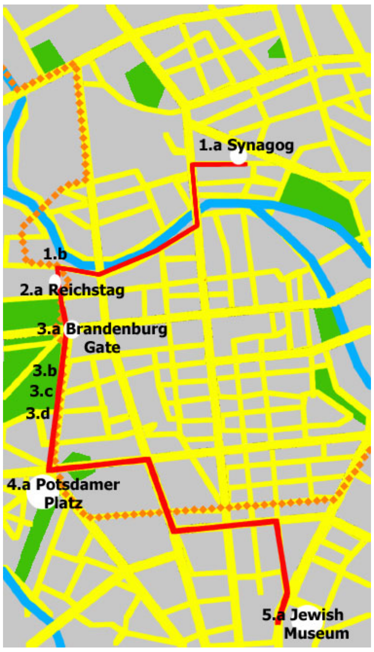

Figure 1: Trajectory of the tourist

Following is the planned trajectory of the tourist (see map in Figure 1) described in terms of major points of interest and addresses.
1.a Starting point: **Synagog** (Oranienburger Strassee 38)
2.a **Reichstag** (Platz der Republik)
3.a Facing the **Brandenburg gate** (Strasse des 17. Juni)
4.a **Potsdamer platz**.
5.a End point: **Jewish Museum** (Lindenstrasse 9).
Note that due to unexpected events and changes of plans, the trajectory will eventually not be the shortest path through all these locations, as illustrated further.

2.2 Delivered information and interaction with the user

We suppose that the tourist walks around with a PDA. The information that she gets on the device as well as the actions taken by the system are represented through the successive PDA screens displayed in Figure 2.

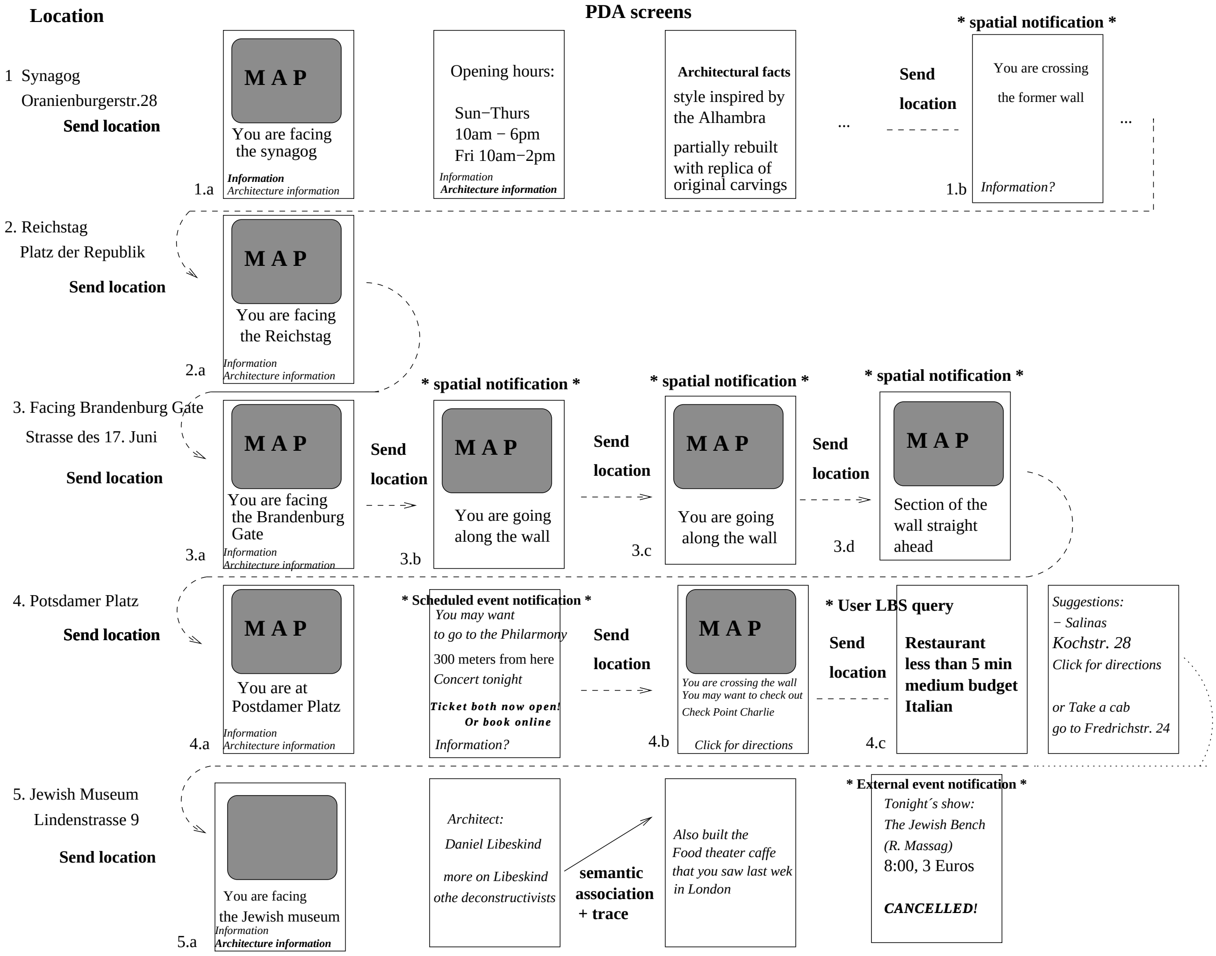

Figure 2: Successive PDA screens

This simple figure illustrates many aspects of the system and in particular many kinds of events. In the figure, basic information from the system is displayed in normal font on the screen while recommendations and further information are displayed in italics. The fact that the user is walking is materialized by arrows linking screens. The various events considered by the system are indicated between star symbols. The fact that the location of the tourist is sent to the system is represented by ”**Send location**”. When it is the case, two possible actions are displayed, ”Information” and ”Architecture Information”, as indicated in italics on the screens. The former is equivalent to asking more general information at a particular location (i.e., major monuments at this location) while the latter takes the profile of the end user (i.e., interested in architecture) into account. The option chosen by the user is highlighted on the screen.

The five lines of PDA screens represented on the figure correspond to the five major locations (addresses) of the tourist's trajectory, as given in Section 2.1. The tourist starts at the Synagog, walks, gets the notification that she is crossing the former wall, goes to the Reichstag, faces the Brandenburg gate, walks, gets the notification that she is walking along the wall, goes to Potsdamerplatz, walks, poses a LBS query, and finally ends up at the Jewish Museum where she gets more information. Each time a ”Send location” event occurs, the number of the step is followed by a letter in order to reference a new location.

Tourist side/mobile device actions. As we can see from Figure 2, the end user ”sends” her location to the system. This is done automatically, periodically. She also sends queries to the system. These are of two kinds:

- Place-related query, such as in Location 3.a, when facing the Brandenburg Gate and pressing the ”Information” button , which is equivalent to asking *What is this monument?*

- Service, such as *Find an Italian restaurant, medium budget, less than 5 minutes walk* (4.c).

System actions. As far as the system is concerned, it uses the current location and time of the tourist to either answer LBS queries or to provide her with more information. Further information includes: more basic information (e.g., at the Reichstag, 2.a), more profile-related information (e.g., architectural facts at the Synagog, at location 1.a), location-related information (e.g., *You may want to check out Check Point Charlie* at location 4.b), location/time related information (e.g., *Concert at the Philharmonie tonight* at location 4.a), ”unexpected” information (e.g., at the Jewish museum, location 5.a, *The performance is cancelled today*), semantically-related information (e.g., *Danial Libeskind also built the Food Theater cafe in London*) and history/trace-related information (e.g., *You saw that cafe last week in London*).

3 Event Management and System Description

This section is concerned with the overall notification system. We first give the major concepts to consider when combining event notification systems and location-based services. We then describe the architecture of our system, TIP. The data model used in our tourist application is given at the end of this section.

3.1 Major concepts in ENS-LBS combinations

The main terms used in the context of event notification systems are that of *events* and *profiles*.

- An *event* is the occurrence of a state transition at a certain point in time. We distinguish primitive and composite events: A primitive event describes a single occurrence, while a composite event describes the temporal combination of events (e.g., a sequence). Composite events can be described using an event algebra [14]. Composite event operators are, e.g., sequence $(E_1; E_2)_t$, conjunction $(E_1, E_2)_t$, disjunction ($E_1|E_2$), and negation $(\overline{E_1})_t$. For example, a *sequence* $(E_1; E_2)_t$ occurs when first $e_1 \in E_1$ and then $e_2 \in E_2$ occurs. The parameter t defines in the profile the maximal temporal distance between the events. The occurrence time $t(e_3)$ of the sequence event instance $e_3 := (e_1; e_2)$ is equal to the time of e_2, i.e., $t(e_3) := t(e_2)$.

  We distinguish *events instances* from *event classes*. An event class is defined by a query. Concrete event instances are denoted by lower Latin e with indices, i.e., $e_1, e_2, \ldots$, while event classes are denoted by upper Latin E with indices, i.e., $E_1, E_2, \ldots$. The fact that an event e_i is an instance of an event class E_j is denoted *membership*, i.e., $e_i \in E_j$.

- A *profile* is a (standing) query executed on the incoming events. In ENS, the result of an event successfully evaluated against a certain profile is a notification about the event. Unmatched events are rejected.

In the LBS context, we need to specify the terms events, actions (notification), and profiles that define the actions to be carried out after certain events occur.

Events in LBS. We distinguish location events that are user-related and external events that are independent of the users:

- **Location events** are events that are connected to a specific user, time, and location. A location event occurs when the user presses the *Information*-button.

- **External Event** are caused by external sources that send event messages to the system. External events are also connected to a certain time and location, however, they are user-independent. An example of external event is *Concert tonight at the Philharmonie*.

Location events trigger a system reaction that results in the dissemination of information to the respective users. External events are, depending on the users' profile and location, forwarded to selected users.

Actions/Information Dissemination. In an ENS, the action that is triggered by an event is the possible forwarding of the event information. Similarly, SDI services forward filtered documents or parts of documents. In our system, the following three forms of actions are distinguished:

- **Information Delivery.** In this case, the action defined in the profile specifies the information data to be selected from the database and to be sent to the user. The selected information data depends on the location event, its time and location, on the event history, on the user/generic profiles, and on the semantical network of the information data. Depending on personal profiles, only selected information about a certain sight are delivered. Depending on generic profiles, additional information may be delivered about the interconnection of sights already seen, such as at location 5.a in the reference example. The scenario also illustrates two examples of *spatial notification*. The first one concerns the fact that the tourist crosses the former wall (in locations 1.b and 4.b), the second one that the tourist is walking around the wall (location 3.b, 3.c, and 3.d). The first notification requires an intersection query between a trajectory and the geometry of an object of interest (here, the wall), the second one a proximity query between the trajectory and an object of interest.

- **Recommendations.** Here, additional information about semantically-related items is given. The selected information depends on the information event, its time and location, the history of events, the user profile, and the semantical network of information data. A representative example is the one occurring at location 4.b (*You may want to check out Check Point Charlie*).

- **Scheduled/external Message Delivery.** In this form of action, the delivery depends on the external/scheduled event, the time and location it refers to, and the user profile. An example of scheduled event occurs in 4.a, while an example of external event is given in 5.a.

Profiles/Conditions. In our system, the profiles are similar to triggers in active databases or profiles in an event action system. In contrast to profiles in ENS, the profile structure is not defined as event-condition(-notification) but as event-condition-action. The action is defined as the selection of information from the various databases. This information is not extracted from the event message as in typical SDI systems but from the system databases described further (more precisely, it is a semi join). We distinguish the following three kinds of profile:

- **Personal profiles** are defined by end users. The personal profile influences the information selected for the user. An example of a personal profile is ”Send only information about architectural facts”. Simple personal profiles consist of keywords selecting topics of information. More advanced personal profiles may consist of attribute value-pairs or database queries that specify certain information. For example, the recommendation of restaurants may be based on user profiles defining the food style (e.g., Italian), the price level (e.g., moderate), and additional restrictions (e.g., vegetarian).

- **Generic profiles** are defined in the service. They are based on a general structural relation between the information data. An example of a generic profile is ”Advise the visit of sights that are in the same semantical group and have not been visited yet”. Simple generic profiles may use only the most recent location event, while sophisticated generic profiles are based on the users event history.

- **Application Profiles** are application-specific profiles defined by an application expert, e.g., the provider of the tourist information. Application profiles mirror semantical relationships between objects of interest. For example, a tourist information guide provides specific information to be delivered if the tourist visits in Berlin the German Dome at Gendamenmarkt *after* the French Dome. This action cannot be described in generic profiles, because it requires application knowledge.

3.2 Architecture of TIP (Tourism Information Provider)

Figure 3 depicts the architecture of the whole system.

The notification system is composed of mobile devices, and of a time, location, and profile based dissemination system.

Mobile devices. The application scenario described in the previous section illustrates the need to send a location at any time and to ask basic queries. A critical issue is the visibility of the history. For privacy reasons, the history should be

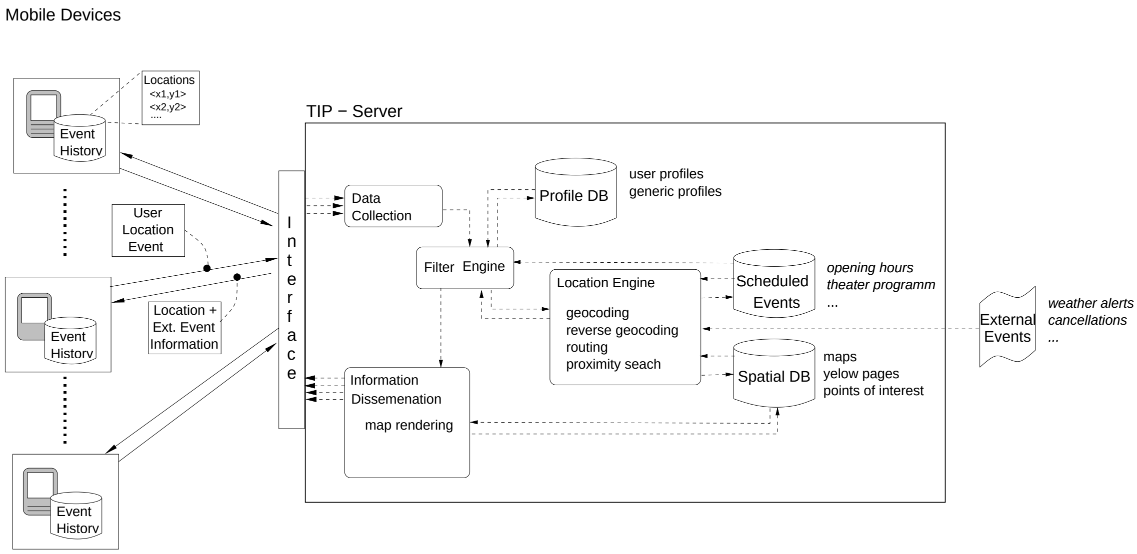

Figure 3: Architecture of TIP

stored on the device. This means that each time end users pose a query their history should be shipped to the system. It is up to the user to make parts of the history visible. In other words, location/time events can be removed from the history (e.g., the end user did not want to mention that she was in that cafe in London last week).

TIP server. The system hinges on three thematic databases, which are:

- Profile database. This database contains generic profiles as well as personal profiles.
- Scheduled event database. This database contains any unexpected event (e.g, weather alert) as well as events that have a limited validity in time such as concert schedules and various programs.
- Spatial database. This databases encompasses more than (vector) maps of the area. It contains points of interests (POI) such as museums, restaurants - all classified through categories - or teller machines. They are organized in a semantical network. POIs are gathered in semantic clusters.

Note that external events are not stored in a database but are coming from an external source.

In addition, the location engine maps locations to maps themselves and assists users in geospatial-related operations. The basic operations are:

- **Geocoding.** Through this operation, the enduser gives an address and the system returns a (longitude,latitude) pair, which may be used to find places of interest in a certain area.

- **Reverse geocoding**. In this operation, which is mostly used here, the user send a (longitude, latitude) pair and the systems returns an address.

- **Routing**. As seen from the typical LBS query in the example (i.e., *Where can I find an Italian restaurant nearby*, at location 4.c) we need navigation tools. The algorithms usually use a two-sided[2] (memory-bounded) A* algorithm to route someone from one coordinate to another (note that geocoding/reverse geocoding are often couples with this algorithm).

- **Proximity search**. This is a broad issue as it concerns many dimensions: time, location, and semantics. The buffer allowed in each dimension can be set by default depending on the profile of the user (e.g., when walking, ”nearby” means about 5 minutes, when looking at the map, 100 meters) and may be changed. With the spatial database, it is a typical point query or region query with fuzziness on the location (e.g., 100 meters). The implementation of these operators is not discussed in this paper.

The role of the notification system is then the following:

- Compare the profile of the user to deliver relevant information (architectural facts). This information can be of various granularity (e.g., at location 5.a., *Further information on Libeskind, i.e., more architectural work by him* after the click).

- Compare the situation with the profile of the user and the relevant part of his/her history to deliver information (e.g., *You saw the Food Theather cafe in London last week*, at location 5.a).

- Look for relevant information in the scheduled events and spatial databases by applying spatio-temporal operators (e.g., *You may want to go to the Philharmonie where there is a concert tonight*, at location 4.a, or *You are crossing the former wall* at locations 1.b and 4.b).

- Look for external events (e.g., *Performance cancelled*, at location 5.a).

- Process typical LBS queries (e.g., *Italian restaurant nearby* at location 4.c).

[2]Unless the arrival time is an issue in which case we need a 2-pass algorithm.

3.3 Data Model

The data model used in our application is depicted in Figure 4 as an entity relationship diagram. the main entities we consider are the following. On the information delivery side: Location, Object of interest (which belongs to a Semantic Class), Item of interest, Information (e.g., information delivered to a user *at a certain level in the hierarchy of information*. On the user side: User/user profile. Finally, on the event side: History and Event (e.g., location event, scheduled event, or external event).

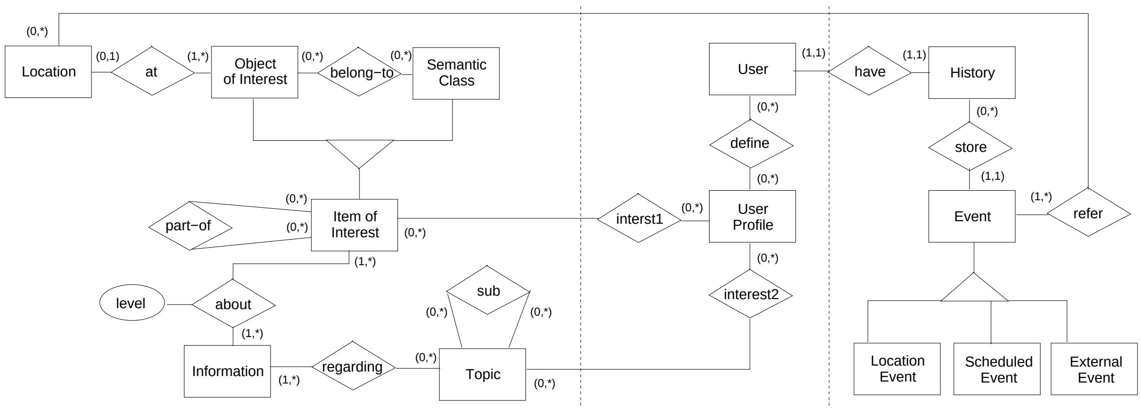

Figure 4: Data model used in TIP

4 Matching Profiles and Events

This section describes the various profiles considered by TIP as well as the way to deliver relevant information to the user based on a traversal of the hierarchical semantic structures according to the data model given in Figure 4.

4.1 Defining User Profiles

A user profile describes information of interest for an end user. Here, we assume the simple form of subject-based user profiles. In the TIP system there exists a hierarchy of subjects that a user can subscribe to. Subscribing to a subject higher in the hierarchy encloses a subjects in the partial tree with the selected subject as root-node. A user profile is used within generic profiles and application profiles. It has no specific actions defined.

Variable	Description
L	set of all locations in the database
I	set of all information data in the database
O	set of all objects of interest in the database
T	hierarchy of topics
G	set of semantical groups in the database
$H(u)$	visible history of user u
$P(u)$	user profile of user u containing a set of topics $\in H(u)$

Table 1: Notations used in this paper

Method	Description
$covers(object, location)$	true if the object is near the location, false otherwise
$member(object, group)$	true if the object is referred to in that semantical group , false otherwise
$above(t)$	set of all topics between t and the root topic *general* of T plus t(transitive closure required)
$below(t)$	set of all child topics of topic t in T plus t (transitive closure required)

Table 2: Methods used for Profile definition

4.2 Defining Generic Profiles

For the formal definition of profiles, we use a notation inspired by Event-Condition-Action (ECA) rules in active databases enhanced by logical operators: ON event IF condition DO action.

We use the notations as given in Table 1 to refer to the various data sources. The methods used for the profile descriptions are shown in Table 2.

1. **General information, without user profile.** A simple location event le_1 occurs ($le_1 \in E_1$) when a user u_1 is at location l_1 at a certain point in time. If it is the k^{th} occurrence of this event in the user history ($le_1^{[k]} \in E_1^{[k]}$), and if general information exists at the k^{th} level for an object that is near this

location, this information is sent.

ON location event $u1.le1$
IF $u1.le1.location \in L \wedge \exists o \in O : covers(o.location, u1.le1.location) \wedge$
$\exists k \in \mathbb{N} : \forall j \in \mathbb{N} : 0 \leq j < k \rightarrow \exists l(j) \in L \wedge near(l(j), u1.le1.location)$
$\wedge\, l(j) \in H(u1) \wedge \exists i(o, k, general) \in I$
DO $\forall o$ (as above) : $send(i(o, k, general))$

2. **Information that matches the user profile.** A simple location event le_1 occurs ($le_1 \in E_1$) when a user u_1 is at location l_1 at a certain point in time. If it is the k^{th} occurrence of this event in the user history ($le_1^{[k]} \in E_1^{[k]}$), and if some information exists for a topic that is a subtopic of the topics defined by this user, and an object exists near this location, this information is sent. The default profile is *general*. The information sent depends on the level (k^{th}) and on the history of topics.

   ON location event $u1.le1$
   IF $u1.le1.location \in L \wedge \exists o \in O : covers(o.location, u1.le1.location) \wedge$
   $\exists k \in \mathbb{N} : \forall j \in \mathbb{N} : 0 \leq j < k \rightarrow l(j) \in L \wedge near(l(j), u1.le1.location)$
   $\wedge\, l(j) \in H(u1) \wedge \exists t \in P(u1) : \exists i(k, below(t)) \in I$
   DO $\forall o$ (as above)$, \forall t$ as above : $send(i(o, k, t))$

3. **Recommendation of objects that are near by (semantically, spatially).** All objects that are near object o are recommended, if they have not been visited yet.

   ON location event $u1.le1$
   IF $u1.le1.location \in L \wedge \exists o \in O : covers(o.location, u1.le1.location) \wedge$
   $\exists g \in G : member(o, g) \wedge \exists q \in O \wedge member(q, g) \wedge q \neq o \wedge q \notin H(u1)$
   DO $\forall o$ (as above)$, \forall q$ as above : $recommend(q)$

4. **Information related to a semantic group.** Information about a group this

object o belongs to, if the user has already visited two objects in the group.

ON location event $u1.le1$
IF $u1.le1.location \in L \wedge \exists o \in O : covers(o.location, u1.le1.location) \wedge$
$\exists g \in G : member(o, g) \wedge \exists q \in O \wedge member(q, g) \wedge q \neq o \wedge q \in H(u1)$
$\exists k \in \mathbb{N} : \forall j \in \mathbb{N} : 0 \leq j < k \rightarrow \exists l(j) \in L \wedge$
$(near(l(j), u1.le1.location) \vee near(l(j), q.location)) \wedge l(j) \in H(u1)$
$\wedge \exists i(g, k, general) \in I$
DO $\forall o$ as above, $\forall g$ as above : $send(i(g, k, general))$

5. **External events that are near by (temporally, spatially) and do match the user profile.**

4.3 Defining Application Profiles

A number of combinations of events are conceivable for application profiles. In general, they are formed by a number of events that happened in the user history combined with a current event. We can define this composite event using our algebra as briefly introduced in Section 3.1. We now show two examples.

- Several events $E_1, E_2, \ldots E_{n-1}$ without a specified order happened before the current event E_n: The profile is defined as a sequence of a composite and a primitive event, where the composite event is conjunction of the previous events $(E_1, E_2, \ldots E_{n-1}); E_n$.

- Several events $E_1, E_2, \ldots E_{n-1}$ happened at a certain order before the current event E_n: The profile is defined as a sequence of events $E_1; E_2; \ldots; E_{n-1}; E_n$.

Combinations of the two forms for composite events can also be considered. Additionally, time intervals may be defined to ensure that the history events occurred within a certain time span, e.g., $((E_1, E_2, \ldots, E_{n-1})_\infty; E_n)_t$ Here, the history events are in no special order, and the last history event occurred no longer than time span t before the current event E_n.

5 Conclusion

In this paper we presented our TIP (Tourism Information Provider) environment to allow end users to get relevant information based on four criteria: (i) their location, (ii) the current time, (iii) their profile, and (iv) their accumulated knowledge.

At present, relevant information based on location is offered by many phone companies as location-based services (LBS). Time-related information is provided by event notification systems (ENS). However, to the best of our knowledge, no system at this point allows one to obtain information based simultaneously on the four criteria cited above. Our approach, the TIP system, gathers all these concepts in a single framework based on mobile devices, various databases, and event notification services.

Delivering information based on profiles means matching a profile definition with various predefined profiles stored in a database. Delivering the right information according to the four criteria cited above is an innovative aspects which presents many challenges, from a high level of abstraction down to a technical level. At a high level, the information of possible relevance needs to be organized in an efficient semantical way. At a more technical level, because of privacy issues, some information - the history - needs to be periodically sent from the mobile device to the system. This encompasses the physical means of sending information. Because sending the whole history of the user each time would not be sensible, we also have to find the most appropriate amount of information to be sent. At present, we send the whole visible history, which is the history that the user is willing to make public (private information is hidden by simply removing it from the history). We are currently working on a heuristic approach to make the best usage of the history w.r.t. the information stored in the various databases.

At a first sight, many aspects of TIP may seem similar to successive mouse clicks from a static user in an Internet browsing session. However, we are not aware of any system, even on the World Wide Web, that would deliver customized information based on a user profile and on his or her history. In addition, in our approach, we also had to consider the small amount of information that can be delivered on mobile devices, hence to convey the information in a simple manner.

TIP is currently under implementation at the Freie Universität Berlin. The implementation is done in Java^{TM} using Oracle 9i^{TM}. The spatial database is stored in Oracle Spatial^{TM}. We are also in the process of specifying memorization levels for end users. The application described in this paper assumes a perfect memory. This prevents the system from repeating information which was already delivered. For instance, a tourist who is in Berlin for the second time and who is interested in wall-related history may not want to be reminded many times that the wall was built in August 1961. However, a tourist who does not have a good memory would like to be reminded of certain facts, even basic. Hence an end user with this profile should be able to specify his or her memorization level.

Eventually, our goal is to implement a platform that will be able to combine location, time, profile, and history in scheduling tasks, for instance in trade fairs where people want to visit booths at a certain interval in time, based on their profile but also on the information they accumulated and their priorities (e.g., visited booths in the past may have a low priority during a further visit, or a high one if the person wants to see it again or to show it to someone). One of the major issues here concerns the definition of various profiles. As far as the platform is concerned, it hinges on combinatorial optimization algorithms, which we plan to introduce in our system by modifying the location engine.

References

[1] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM (2)*, pages 775–784, 2000.

[2] T. Brinkhoff. The impact of filtering on spatial continuous queries. In *10th Intl. Symposium on Spatial Data Handling (SDH 2002)*, 2002.

[3] T. Brinkhoff and J. Weitkämper. Continuous queries within an architecture for querying XML-represented moving objects. In *Proceedings of the 7th Intl. Symposium on Spatial and Temporal Databases (SSTD)*, volume 2121 of *Lecture Notes in Computer Science*, pages 136–154, Heidelberg/Berlin/New York, 2001. Springer Verlag.

[4] A. Carzaniga. *Architectures for an Event Notification Service Scalable to Wide-area Networks*. PhD thesis, Politecnico di Milano, Milano, Italy, 1998.

[5] J. Chen, D. DeWitt, F. Tian, and Y. Wang. NiagaraCQ: A scalable continuous query system for internet databases. In *Proc. of the ACM SIGMOD Conf. on Management of Data*, 2000.

[6] K. Cheverst, K. Mitchell, and N. Davies. The role of adaptive hypermedia in a context-aware tourist GUIDE. *Communications of the ACM*, 45(5):47–51, 2002.

[7] Federal Communications Commission. Fcc wireless 911 requirements fact sheet, 2001. available at `http://www.fcc.gov/e911/`.

[8] D. Faensen, L. Faulstich, H. Schweppe, A. Hinze, and A. Steidinger. Hermes – a notification service for digital libraries. In *ACM/IEEE Joint Conference on Digital Libaries*, 2001.

[9] M. Fleck, M. Frid, T. Kindberg, E. O'Brien-Strain, R. Rajani, and M. Spasojevic. From informing to remembering: Ubiquitous systems in interactive museums. *Pervasive Computing*, 1(2):13–21, 2002.

[10] L. Forlizzi, R. H. Güting, E. Nardelli, and M. Schneider. A data model and data structures for moving objects databases. In *Proc. of the ACM SIGMOD Conf. on Management of Data*, 2000.

[11] N. H. Gehani, H. V. Jagadish, and O. Shmueli. Composite event specification in active databases: Model & implementation. In *Proceedings of the Intl. Conference on Very Large Data Bases (VLDB)*, 1992.

[12] R. H. Güting, M. H. Böhlen, M. Erwig, C. S. Jensen, N. A. Lorentzos, M. Schneider, and M. Vazirgiannis. A foundation for representing and quering moving objects. *Transations of Database Systems (ACM TODS)*, 25(1):1–42, 2000.

[13] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[14] A. Hinze and A. Voisard. A parameterized algebra for event notification services. In *Proceedings of the 9th International Symposium on Temporal Representation and Reasoning (TIME 2002), IEEE Computer Society*, 2002.

[15] S. Hsi. The electronic guidebook: A study of user experiences using mobile web content in a museum setting. In *Proceedings of the IEEE Intl. Workshop on Wireless and Mobile Technologies in Education (WMTE)*, 2002.

[16] H. V. Jagadish and O. Shmueli. Composite events in a distributed object-oriented database. In *Proceedings of the International Workshop on Distributed Object Management*, 1992.

[17] B. Krishnamurthy and D. S. Rosenblum. Yeast: A general purpose event-action system. *ACM Transactions on Software Engineering*, 21(10), 1995.

[18] H. Liu and H.-A. Jacobsen. A-topss - a publish/subscribe system supporting approximate matching. In *Proceedings of the Intl. Conference on Very Large Data Bases (VLDB)*, pages 1107–1110, 2002.

[19] L. Liu and C. Pu. Complex event specification and event detection for continual queries. Technical report, OGI/CSE, Portland, 1998. available at `ftp://cse.ogi.edu/pub/tech-reports/`.

[20] L. Liu, C. Pu, and W. Tang. Continual queries for internet scale event-driven information delivery. *IEEE Transactions on Knowledge and Data Engineering*, Special issue on Web Technologies, 1999.

[21] G. R. Malan, F. Jahanian, and S. Subramanian. Salamander: A push-based distribution substrate for internet applications. In *USENIX Symposium on Internet Technologies and Systems, Monterey, California, December 8-11, 1997*, volume 32, 1997. available at `http://www.eecs.umich.edu/~rmalan/publications/mjsUsits97.ps.gz`.

[22] J. C. Navas and T. Imielinski. Geocast - geographic addressing and routing. In *Mobile Computing and Networking*, pages 66–76, 1997.

[23] R. Oppermann, M. Specht, and I. Jaceniak. Hippie: A nomadic information system. In *Proceedings of the First International Symposium Handheld and Ubiquitous Computing (HUC)*, 1999.

[24] J. Pereira, F. Fabret, H. Jacobsen, F. Llirbat, R. Preotiuc-Prieto, K. Ross, and D. Shasha. LeSubscribe: Publish and subscribe on the web at extreme speed. In *Proc. of the ACM SIGMOD Conf. on Management of Data*, 2001.

[25] S. Poslad, H. Laamanen, R. Malaka, A. Nick, P. Buckle, and A. Zipf. Crumpet: Creation of user- friendly mobile services personalised for tourism. In *Proceedings of 3G 2001 - Second International Conference on 3G Mobile Communication Technologies*, 2001.

[26] B. Segall, D. Arnold, J. Boot, M. Henderson, and T. Phelps. Content Based Routing with Elvin4. In *Proceedings of the AUUG2K Conference*, 2000.

[27] S. Shekhar and A. Fetterer. Genesis: An approach to data dissemination in advanced traveler information systems. *IEEE Bulletin of the Technical Committee on Data Engineering*, 19(3):40–47, 1996.

[28] A. P. Sistla, O. Wolfson, S.Chamberlain, and S. Dao. Modeling and querying moving objects. In *ICDE*, pages 422–432, 1997.

[29] R. Strom, G. Banavar, T. Chandra, M. Kaplan, K. Miller, B. Mukherjee, D. Sturman, and M. Ward. Gryphon: An information flow based approach to message brokering. In *Proceedings of the International Symposium on Software Reliability Engineering*, 1998.

[30] J. Su, H. Xu, and O. H. Ibarra. Moving objects: Logical relationships and queries. In *Proceedings of the 7th Intl. Symposium on Spatial and Temporal Databases (SSTD)*, volume 2121 of *LNCS*, heidelberg/Berlin/New York, 2001. Springer Verlag.

[31] M. Vazirgiannis and O. Wolfson. A spatiotemporal model and language for moving objects on road networks. In *Proceedings of the 7th Intl. Symposium on Spatial and Temporal Databases (SSTD)*, heidelberg/Berlin/New York, 2001. Springer Verlag.

[32] S. Volz and D. Klinec. Nexus: The development of a platform for location aware application. In *Proceedings of the Third Turkish-German Joint Geodetic Days Towards A Digital Age, Istanbul, Turkey*, page 599608, 1999.

[33] R. Want, A. Hopper, V. Falcao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1992.

[34] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, 1997.

[35] T. W. Yan and H. Garcia-Molina. The SIFT information dissemination system. *ACM Transactions on Database Systems*, 24(4):529–565, 1999.

[36] S. Yang and S. Chakravarthy. Formal semantics of composite events for distributed environments. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 400–407, 1999.