Tip: Personalizing Information Delivery in a Tourist Information System

Article *in* Information Technology & Tourism · August 2009

DOI: 10.3727/109830509X12596187864071 · Source: DBLP

3 authors:

Annika Hinze
The University of Waikato

Agnes Voisard
Freie Universität Berlin/Frauhofer FOKUS

George Buchanan
City, University of London

Information Technology & Tourism, Vol. 11 pp. 247–264
Printed in the USA. All rights reserved.

1098-3058/09 $60.00 + .00
DOI: 10.3727/109830509X12596187864071
www.cognizantcommunication.com

TIP: PERSONALIZING INFORMATION DELIVERY IN A TOURIST INFORMATION SYSTEM

ANNIKA HINZE,* AGNÈS VOISARD,† and GEORGE BUCHANAN‡

*University of Waikato, Hamilton, New Zealand and Humboldt University, Berlin, Germany
†Fraunhofer Institute for Software and Systems Engineering (ISST), Berlin, Germany
and Freie Universität, Berlin, Germany
‡City University, London, UK

Advanced tourist information systems should offer more than relatively static information about sights and places. Instead, semantically rich information about sights should be delivered to the mobile users. Furthermore, tourists should not be overwhelmed by a stream of superfluous data that are unrelated to their interest, location, and knowledge of a place. Personalization of the information delivery to each traveler, together with their travel history, is therefore crucial. This article presents the major design issues of the personalized Tourist Information Provider (TIP). TIP is a combination of an event-based system (EBS) and a location-based service (LBS) applied to a mobile environment. We discuss the lessons learned from developing its kernel using a semantic network of sight-related information and considering the travelers' interest and travel route, with emphasis on modeling decisions and their impact on the final system.

Key words: Location-based services; User profiles; Semantic modeling

Introduction

The development of advanced mobile information systems such as a tourist information system requires the application and transfer of semantic web technologies and modeling techniques into the area of mobile applications. This article describes some of the problems that emerge when Web technologies are applied in this context. We considered three major design factors in developing the tourist information system:

1. Timelines: Instead of providing only a large number of discrete pieces of static information about sights (as a travel book would do), information should be dynamically tailored to the personal needs and interests of the tourists as they travel.
2. Rich information: Semantically rich information about sights (e.g., groups of sights and their relation to each other) should be easily accessible to the tourist at any time. The modeling techniques developed for the semantic

Address correspondence to Annika Hinze, Department of Computer Science, University of Waikato, Hamilton, New Zealand. E-mail: Annika.Hinze@cs.waikato.ac.nz

web should be employed for two reasons. Firstly, they have been developed to capture semantically rich information. Secondly, tourist-oriented information should not be modeled independently of and unconnected to publicly available sources such as the (semantically enriched) Web.
3. Personalization: Tourists should not be overwhelmed by a stream of superfluous data unrelated to their interest or their location. Given the potentially vast information available, personalization of delivered information is crucial to a good user experience. Additionally taking advantage of the tourist's acquired knowledge based their travel history brings a new dimension to the quality of the delivered information.

In this article, we describe the design of a personalized mobile Tourist Information Provider (TIP). We discuss the lessons learned from developing the system kernel using a semantic network of sight-related information and considering the travelers' interest and travel route. We identify the specific modeling requirements of the application. For modeling all relevant data, two approaches are compared: semantic web-related (i.e., RDF and RDFS) and traditional database-related (object-relational) techniques. In a second step we revisit the model to address requirements of a growing system. Note that our focus is on modeling and user adaptivity and not on other issues that arise in such systems, such as availability, privacy, and performance.

The article is organized as follows. The next section discusses functionality. From a reference application scenario we derive a list of functional requirements, and reflect on key related work. The third section focuses on the first version of TIP, presenting two alternatives for the underlying data model: an RDF-based approach and an object-relational one, before progressing to the main implementation considerations. The last section presents the next generation of TIP, which addresses issues such as (model) scalability.

Designing a Personalized Tourist Information System

This section presents the design considerations for our personalized tourist information system. The next section introduces a sample application scenario that serves as illustration throughout the article. We then identify specific requirements, which are used as the basis for analyzing related work.

Application Scenario

Our example scenario is divided into a sequence of steps for easier further reference. Consider two tourists doing sight-seeing in Berlin: Anne visits the city for the first time; Berit has been there before. They are currently at the *Gendarmenmarkt* (a square in central Berlin). Both Anne and Berit carry a telephone equipped with GPS-enabled functions and a TIP client. As they plan to spend their time sightseeing together, their different traveling background and interests will be reflected in the information presented to them on their mobile devices.

Step 1: Before starting their sightseeing tour around the *Gendarmenmarkt*, Anne and Berit connect to TIP and define the types of sights and the information topics they are interested in: both are interested in cathedrals and statues; Anne is particularly interested in architecture and Berit in history.

Step 2: When Anne and Berit reach the *Gendarmenmarkt*, they both select "Show information" on their individual TIP clients. The TIP server uses their location's GPS coordinates to access tourism information. As a result, Anne is presented with information about the architecture of cathedrals and statues near to her current location. Berit receives information about the same sights, but with details regarding historical facts. Anne's and Berit's way over the *Gendarmenmarkt* is shown in Figure 1. The TIP client offers both tourists to visit the German Cathedral (*Deutscher Dom*) and the French Cathedral (*Französischer Dom*) as well as the statue of Schiller (*Schillerdenkmal*). Note that information about the concert house (*Konzerthaus*) is not offered to them as it does not match their interests.

Step 3: They decide to first visit the French Cathedral. When they are about to enter the cathedral, each of them pushes the Information Button again. Anne is now provided with detailed general information about the cathedral plus

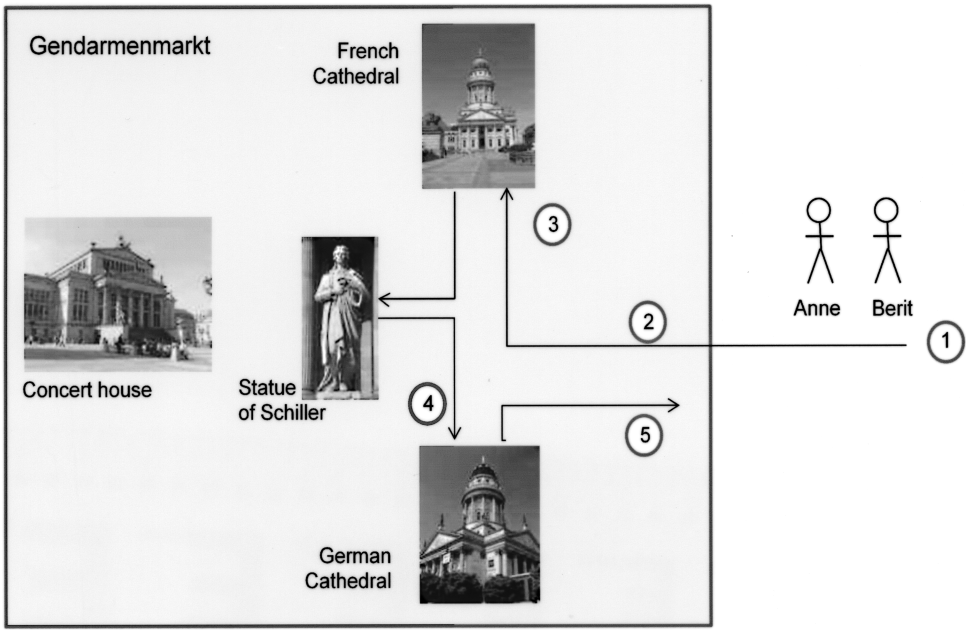

Figure 1. Anne's and Berit's way over the *Gendarmenmarkt* square. Numbers in circles refer to the steps in the scenario.

specific information regarding the architecture of the cathedral. Berit is presented with brief general information about the cathedral plus detailed elaborated information regarding historical facts. She has already received more general information during her last visits—she may still access these if she likes.

Step 4: After having visited the cathedral, Anne and Berit continue to the Schiller statue and then to the German Cathedral. Just as at the French Cathedral, they are provided with general/specific information and facts about architecture and history, respectively. Additionally, Anne is given a comparison between the architectures of the German and the French Cathedral, because she has visited both cathedrals.

Step 5: Finally, the TIP clients suggest to both tourists to visit further cathedrals in central Berlin: the Berlin Cathedral (*Berliner Dom*) and the Nikolai Church (*Nikolaikirche*). They continue towards the Berliner Dom.

Requirements Analysis

Analyzing our scenario, we define various functional requirements for the envisioned system (FR1–FR6). The requirements presented here set up a framework that is used to contrast the different design approaches for the application.

FR1. User profiles and information filtering: As illustrated in our scenario, the system needs to support personalized information delivery. The user's preferences have to be defined and stored, typically in a personal profile (cf. Step 1). The information provided to the users is to be filtered according to their profile.

FR2. Location-based information delivery: To support the delivery of timely information, the system has to support the registration of the users' positions. Additionally, a filtering of the delivered information is necessary so that only information relevant to the current position is delivered. (cf. Step 2 of scenario).

FR3. Information delivery based on hierarchical information structure: The system should support information delivery that provides different detail levels of information. The presented information depends on the level of user interests (e.g., from general information to detailed expert knowledge). More information should be available to the user on request (cf.

Step 3). Thus, the stored information regarding the sights has to reflect a hierarchical structure. We refer to these (possibly interconnected) information hierarchies as the vertical information network.

FR4. Clusters of sights: In addition to providing information for each stored sight, the system also has to offer the possibility to group the sights in semantic clusters. Semantic clusters are groups or classes of sights that share certain semantic features. The cluster emphasizes this feature. Examples of semantic features are

- Sight type or category (e.g., churches—as in our scenario—or theaters)
- Geography (e.g., sights near to a given point—*Gendarmenmarkt*)
- Predefined attributes (e.g., regarding the same architect or the same period)
- Visitor interest (e.g., sights preferred by Swedish arts students)
- Sight or event conditions (e.g., indoor vs. outdoor events)
- Visitor access (e.g., wheelchair access, parking, public transport)

A semantic relation can be geographic or based on a common attribute such as a category of place or a common architect. Information about semantic clusters might be explicitly stored in the database (e.g., for sights at the *Gendarmenmarkt*) or the information can be derived from given context or data (e.g., sights near by a given point can be determined dynamically based on GPS data). In Step 4 of our scenario, Anne is provided with joint information about the German and French Cathedral. The clustering of sights and the storage of information about clusters is referred to as *horizontal information network*.

FR5. Information delivery based on user history: Revisiting users are not presented with information that has already been delivered to them unless they explicitly access it. Instead, a connection to their previous visit is made and more detailed information is offered. For example, in Step 4, Anne is given detailed information because she has visited the cathedral before. The system stores information about the user's travel history. The delivery of information regarding visited clusters also requires the user history to be stored.

FR6. Collaboration with other services: The information service needs to collaborate with other services on the same device to offer a seamless information access for the user. For example, a map service could show the user as they are traveling and also touristic points of interest as provided by the information system. A recommendation service may recommend further points of interest based on the user's profile (cf. Step 5). Further services may include online ticketing.

These functional requirements determine the features to be implemented for the system and provide constraints to the underlying data model.

Related Work

This project is related to many areas such as location-based services, event-based systems, and semantic modeling. We first describe existing work in these areas and then focus on the state of the art in tourist information systems.

Location- and Context-Based Services. Location-based systems emerged as a combination of general information and positioning (Schiller & Voisard, 2004). Typical applications include navigation systems and "yellow maps," which are a combination of maps and information from yellow pages. A typical query would be "Show me drugstores in my area," "my area" being either entered by the user or inferred from the user's location.

The more general capacity to obtain information about the environment in which a system is working is called "context awareness" (Dey & Abowd, 2000). Most tourist information systems use only the current user location as context. We propose to incorporate and combine more complex information: the user-stated preferences, their travel history, and the similarity between users. The user's history and preferences are used as guides within the semantic network of sight-related information.

Event-Based Systems. Event-based systems (EBS) react to external or internal events with predefined actions (Hinze, Sachs, & Buchmann, 2009). EBS are used for a number of applications such as gaming (Kabus & Buchmann, 2007), traffic management (Joseph et al., 2006), and disaster management (Wu, Liu, & Mani Chandy, 2008). The few EBS used in the context of tourist information focused on support in travel planning and route guidance (e.g., in the Genesis system) (Shekhar & Fetterer, 1996).

The profiles define filters on events. It is common to distinguish subject-based from content-based filtering. In the former, the profiles identify topics under which the events are ordered. Content-based filtering allows for a more fine-grained filtering, better adapted to the system users. Context-dependent information delivery as in TIP requires a new form of content-based filtering: The content is the context of the user and their environment (e.g., their location, interest, and travel history).

A combination of concepts from active databases and event notification systems is offered by Event Action Systems (EAS) such as Yeast (Krishnamurthy & Rosenblum, 1995). In such systems, the user profile defines an action that is triggered by a certain event. In several aspects our work is very close to EAS (e.g., in the detailed action definition for the information delivery). For an extensive analysis of related event-based approaches see Hinze and Voisard (2003).

Semantic Modeling/Semantic Clustering. Modeling the semantic relationships between objects is one of the main aims of the semantic web (Barners-Lee, 1998). It has been noted that semantic metadata are crucial for building the semantic web to describe the meaning of items and their relationships (Lassilia, 1998). This is in line with the development in other disciplines such as database and software engineering, which use semantic data models and schemas to define and give meaning to data and parts of programs (see, e.g., Guarino, 1998). The standard for this semantic modeling is RDF. Using RDF, WWW resources are annotated with semantic information, which uses underlying conceptual models (schemas) to define the classes and properties used for these semantic annotations (Nejdl, Wolpers, & Capelle, 2000). For TIP, we considered RDF and object-relational design as two contrasting modeling approaches.

The concept of semantic clustering has been used, for example, in content-based image retrieval (Sheikholeslami, Chang, & Zhang, 2002): content-based image retrieval is enhanced by using the heterogeneous features embedded in the images. To support various features, like texture, color, and shape, which may require different similarity measurements, a semantics-based clustering and indexing can be used.

The combination of location-based services, event notification, and semantic web modeling for a mobile tourist information system is a novel approach. Several system exist that process location-based information; extensive study has been dedicated to the mobile/ubiquitous aspects of location-based services (see, e.g., Saltenis, 2000; Zhao, 2000). In contrast, a large variety of event notification systems has been implemented; very few, however, are used in the context of tourist information.

Tourist Information Systems. Only few tourist information systems encompass the notion of events or personal profiles. Here, we discuss selected systems for mobile tourist information delivery using personalized approaches: Guide, Crumpet, Tourist Guide, Catis, Cyber-guide, Flame, and AccesSights.

The Guide system (Cheverst, Davies, & Mitchell, 2002) is an earlier system and does not support GPS locations but broadcasts information near sights (in Lancaster). User profiles are used for the information filtering. The system maintains a user history and welcomes the user back to a sight already visited. The delivery of sight-related information is not influenced by the user history but it is used when listing "sights near-by."

In Crumpet (Poslad, Laamanen, Malaka, Nick, & Zipf, 2001), the user profile is developed by tracking the users travel movement (i.e., user history). Several mobile devices are supported for user information. No recommendations are supported. The Tourist Guide system (Simcock, Hillenbrand, & Thomas, 2003) uses GPS and a simple user context of the location and buildings in sight. The system provides three modes of operation: map mode, guide mode, and attraction mode, thus supporting two services (information and map) but no external ones. In the guide mode, a trail is marked on the map with an interesting related set of attractions. These attractions are shown up on the map in red. The attraction mode acts as a digital tourist guide, supplying users with sound, images, and textual tourism information.

Catis (Pashtan, Blattler, Heusser, & Scheuermann, 2003) is a flexible system that uses user profile, location, event history, and other context data to filter the delivered information. It provides flat unstructured information. Restaurant recommendations are given based on the user's preference and travel history. Cyberguide (Abowd, Atkeson, & Hong, 1997) is an indoor guide system; no user profiles are supported. Location and orientation of the user within a building are used to filter the information. Recommendations are not supported. The system has a wide range of additional features (e.g., group communication).

The FLAME 2008 platform (Holtkamp, Gartman, & Han, 2003) aimed at providing visitors of the Olympic Games—being tourists, journalists, or locals—with personalized information. At any time and location, a user is in a particular *situation*, such as in a stadium, in a taxi, in a hotel, on the phone (Meissen, Pfennigschmidt, Voisard, & Wahnfried, 2004). Situation dimensions are defined according to ontologies. The demand of a user is evaluated dynamically and, at any time, a particular situation induces a relevant set of services—for in instance, in a stadium, the visitor will be offered services such as places to eat or information on the players. The history of the users is not taken into account.

AccesSights (Klante, Krösche, & Boll, 2004) is a multimodal system providing information to both normally sighted users and visually impaired people. AccesSights supports three modes: orientation phase, movement phase, and information perception phase. Both visual display and auditory information is given to users. Normally sighted users perceive sight information via audio and visual on a map, whereas blind people listen to information. AccesSights uses loudness to indicate the distance between the users' current location and sights of interest.

We summarize our evaluation for the selected tourist systems, contrasting their features with the identified requirements in Table 1. This panel of extant systems demonstrates that while certain functional requirements are well supported (e.g., FR2: Location), other needs are poorly met (e.g., FR3 to FR6). We will next turn to the system design required to deliver a comprehensive answer to these requirements.

Data Modeling and Core Architecture

This section is devoted to the data model and architecture of the TIP core system. We discuss two possible models (using different techniques) for their suitability for TIP. Initially, we translate the general system requirements into specific requirements regarding the data model. Then, we compare the two modeling approaches, one from the semantic web community (RDF model) and one from the database community (object-relational model).

Data Model Requirements

Based on the functional requirements introduced previously, we identify the following requirements regarding the data modeling (MR1–MR5). We also define four additional requirements on system support (MTR1–MTR3).

- **MR1. User profiles:** For each user, a profile must be stored. User profiles refer to (predefined) topics of information.
- **MR2. Location:** The location of users and sights must be represented in the system.
- **MR3. Hierarchy of information:** Information regarding sights (data objects) is ordered into topics (e.g., architecture, history). For every sight and for each topic available for that sight, a hierarchy of information regarding the sight has to be stored. An example for two sights is shown in Figure 2 (cf. Step 4 in our scenario).
- **MR4. Clusters of data objects and assigned information:** Data modeling must support clustering (grouping) of sights and a hierarchy of information. Collections consist of one or more elements as shown in Figure 3.
- **MR5. Personal history data:** The system must store the travel histories of the users, that is, the visited locations and the provided information (levels). This information has to be available for information selection and recommendation.

Table 1
Analysis of (Selected) Related Work According to Requirements

	FR1: Personalization	FR2: Location	FR3: Information Hierarchy	FR4: Sight Clusters	FR5: User History	FR6: Other Services
Guide	+	(–)	–	–	(+)	+
Touris Guide	–	+	–	–	–	(+)
Crumpet	+	+	–	–	+	–
Catis	+	+	–	–	+	+
cyberguide	–	+	–	–	–	–
Flame	+	+	–	–	–	–
AccessSights	–	+	–	–	–	–

+: system addresses requirement; –: system does not fulfill the requirement.

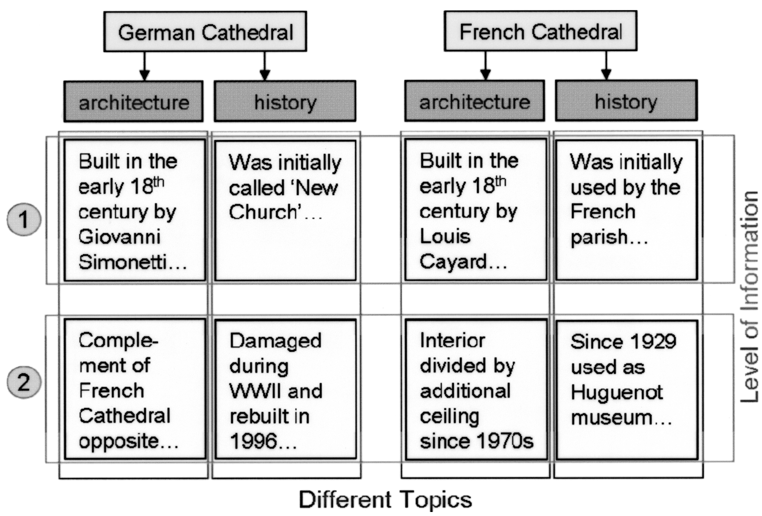

Figure 2. Information structure for two sights. For each sight, the information is ordered according to topics. For each topic, information with increasing level of detail is available.

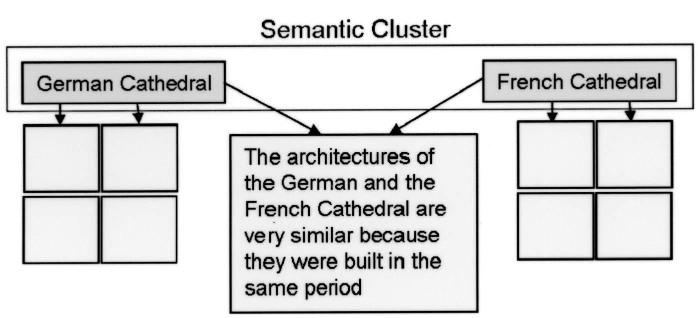

Figure 3. Information structure for a semantic cluster.

System Support for Modeling Technique

Because we compare alternatives for modeling techniques, we also need to specify requirements on the system support for a chosen technique. We, therefore, define the following four requirements on the support of modeling techniques:

MTR1. Support for all modeled features: The storage system should preferably support the direct implementation of all modeled features.

MTR2. Support for effective querying: Query languages available for the data model have to support a wide range of filter queries (for details regarding the language requirements see Hinze & Voisard, 2003).

MTR3. Efficient storage and retrieval of data: Restrictions of mobile access to the data while travelling requires efficient retrieval. The amount of user-related data (e.g., user histories) will grow over time—demanding efficient storage strategies.

MTR4. Extensibility of data set: The data structure has to support new information within the existing framework and also extensions to the framework itself.

Semantic Web Approach: RDF

We now introduce and analyze a semantic web approach to modeling the application data. In the next section, we describe the contrasting database-related approach. We initially analyzed two semantic web techniques: Topic Maps (Biezunski, Bryan, & Newcomb, 2002) and RDF (Lassila & Swick, 2003) (for details of the analysis see Löffler, 2004). Here, we focus on the RDF-based approach (using the graph notation). We assume that the reader is familiar with the basic concepts of RDF and RDFS (Brickley & Guha, 2003).

Figure 4 shows a section of the RDF graph of the example data regarding sights (users and their histories are not shown here). We distinguish four classes of locations (sights): statue, square, building, and cathedral, where cathedral is a subclass of building. The hierarchical information in various topics for single sights (e.g., as introduced in Fig. 3) is assigned to the classes by the property topic having the subproperties general, architecture, history. The property is of range rdf:Seq allowing an ordered list of property values for each resource.

For the semantic grouping of sights we use a class group with properties contains, currentSight, and hasInfo. The property contains is of range rdf:Bag and therefore allows the creation of loose collections of sights. currentSight refers to the sight at which the tourist is located when this group is accessed. The clusters for recommending sights are described by cluster, which also has the property contains.

Analysis of the RDF Model. The user profile details (MR1), location of the user (MR2), and the user travel history (MR5) are temporal data. For efficiency reasons, we did not model them in RDF but in a relational database. Functions for efficient handling of location data (as known from location-based services) have to be provided by the system implementation.

The subPropertyOf relationship allows for simple implementation of the hierarchy of topic information (MR3). Clusters of data objects can be built using the rdf:Bag construct, which allows to form loose collections of objects. Additional information regarding a cluster (MR4) can be assigned using a property–value pair.

The effective and efficient access/storage of the information (MTR1–MTR3) heavily depends on the query language and storage system used. In theory, the model and database are easy to extend (MTR4) due to the simplicity of extending RDF and RDFS data. However, for a given implementation, the extensibility depends on the used storage system. A disadvantage of current RDF storage systems is the necessary translation into other data models. We used the Sesame system (Broekstra, Kampman, & van Harmelen, 2002), which allows the storage and manipulation of RDF data in a database. Sesame supports the RDF languages RQL (Broekstra & Kampman, 2002), SeRQL (Aidministrator, 2000), or RDQL (Seaborne, 2002). All queries are translated from an RDF query language into the native database query language (e.g., SQL). Relationships between the data might be lost in the translation process. Additionally, the result set is translated back into RDF. For several RDF query languages, the result of a query cannot

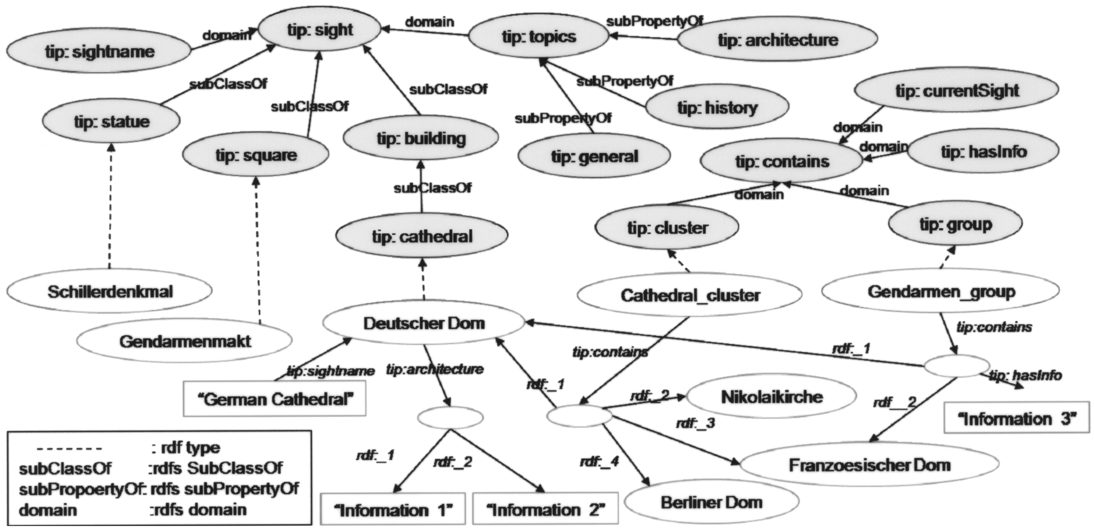

Figure 4. Selection of the RDF/S data model (tourism ontology in dark shading) and example data (TIP dataset in light shading).

be queried further. This leads to serious performance problems (cf. Bönström, Hinze, & Schweppe, 2003). Another disadvantage is the mixed data modeling. An integrated and seamless access might not always be ensured and may even lead to inconsistency in the data.

Object-Relational Approach

The second model uses an object-relational approach. An object-relational database combines the advantages of object-orientation with the functionality of a relational database system. Due to the hierarchical structure of the sight-related data, we modeled these in an object-oriented way. Figure 5 shows a refined model (compared to the RDF model). User-related data, like profiles and user event history, are modeled in a strictly relational model. The object-oriented features were implemented as table hierarchies. The object hierarchy is modeled in "is a" relationships; the lowest level objects (e.g., cathedral) inherit all attributes from the more general ones. Semantic groups are built in a many-to-many relationship, which allows the assignment of one sight to various semantic groups; semantic groups contain one or more sights. The current sight at which the user is located is referred to by an attribute. The clusters for sight recommendation are modeled similar to groups.

Analysis of the Object-Relational Model. The user profile details (MR1), location of the user (MR2), and the user travel history (MR5) are modeled using the same technique as for the sight-related data (not shown in Fig. 5). The locations of sights are modeled as attributes of sights. Functions for handling location-related data can be reused from a spatial extension for the DBMS (e.g., from PostGIS for PostgreSQL, or Oracle Spatial for Oracle; see Rigaux, Scholl, & Voisard, 2001). The hierarchy of topics (MR3) is directly supported by the object-oriented features of the database model. The grouping of sights is captured by a relationship. Information about a group (MR4) is stored as an attribute of the relation group.

As in the case of RDF, the effective and efficient access and storage of the information (MTR1–MTR3) depend heavily on the storage system used for the implementation. An extension of the data set results in the simple task of inserting additional data in the database. The extension of the modeling level (e.g., topics and types of sights) is a more complex task. The advantages

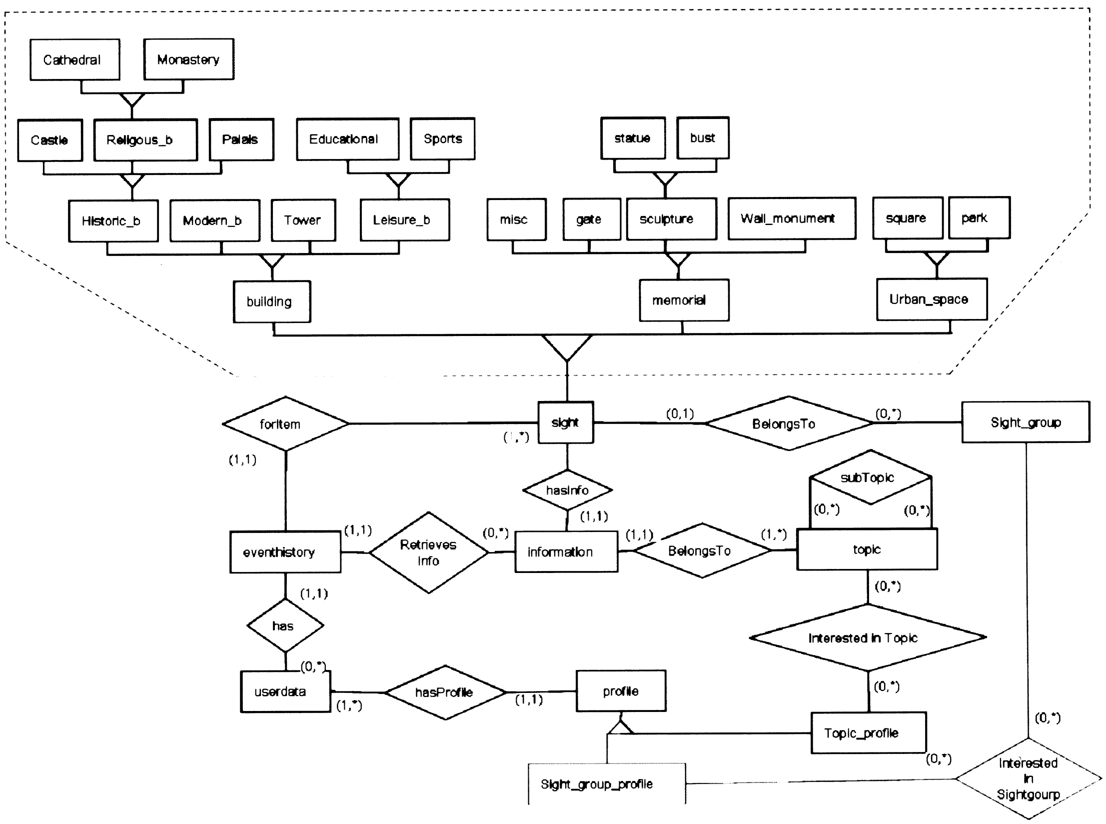

Figure 5. Extended object-relational data model.

of this approach lie in the use of well-developed concepts that are supported by sophisticated implementations (i.e., object-oriented database systems, GIS systems). For example, attributes of objects can easily be used by data mining algorithms (e.g., for clustering the sights for recommendation according to certain characteristics). Moreover, tuning and indexing of database systems has a long tradition that can be drawn upon for this application.

On the other hand, the resulting model is more rigid and less flexible (MTR4).

Implementation

As our project progressed, the implementation of the TIP system underwent several evolutionary steps. We here first discuss the implementation of the TIP core—that is, the initial TIP implementation that formed the nucleus of a much larger TIP system. [In our publications, this initial implementation is referred to as "TIP 1" (i.e., the first generation of the system) or as "TIP core" (referring to the central conceptual component that this implementation represents).] Then we discuss the implications of the two contrasting data models for the implementation.

TIP Core Architecture

We implemented the TIP core as envisioned in the scenario. Figure 6 shows the architecture of the TIP core system. The TIP core consists of several components: a spatial component for the location information of sights and users, a profile component handling generic profiles and user profiles, an event component, the event history, and the database of sight-related information. Tourists send their location and their ID to TIP, which, in turn, uses this information for filtering according to the user profiles and to update the event history. The spatial component and the user location component identify the sights near a user. The events component recognizes repeat visitors. Taking the results of these filter steps, the information relevant to the user is identified and provided to the user.

Positioning in TIP is done via user-based positioning system, such as GPS in combination with information of a cellular network. We assume the existence of a TCP/IP communication system with HTTP connections; the presentation at client side uses a Web browser. Figure 7 shows a screenshot of the TIP system (version 1) as it presents information to Anne at the French Cathedral (cf. Step 3 in the scenario).

Implications of Modeling Technique

The implementation of the RDF model used Sesame 0.96 with a PostgreSQL 7.3 database and Sesame RDF as query language. Even though Sesame is one of the most advanced RDF systems, Sesame RQL does not offer the functionality needed for our application (see http://sourceforge.net/mail archive/forum.php?thread_id=3460493&forum_id =8435). For example, aggregate functions, container queries, and mathematical functions are not

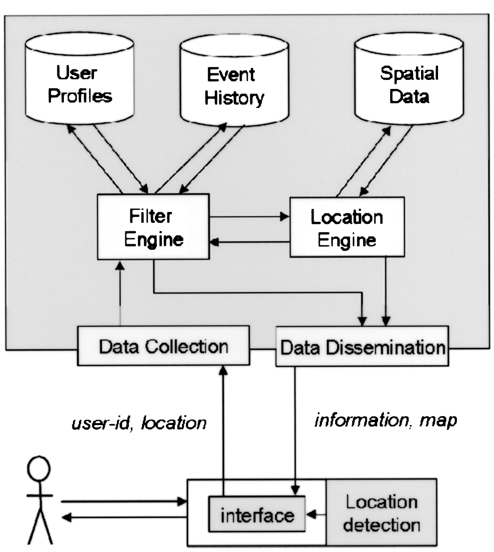

Figure 6. TIP 1 architecture.

supported. We implemented replacements for these features in Java.

Selecting an appropriate RDF query language prove to be a difficult task at that early stage of the semantic web. We found that most languages did not support the features needed: containers, bags and sequences, inference, reification, schema-related queries, transitive closure, predefined functions, and open path queries (Haase, Broekstra, Eberhart, & Volz, 2004). Even orthogonality of query and result set is not always given. The Sesame version of RQL did not implement all RQL features. We decided to use Sesame's RDF because of its wide acceptance in the semantic web community. For the object-relational implementation, we used PostgreSQL 7.3. To achieve all required object-relational features (see http://archive.postgresql.org/pgsql-general/2003-5/msg00495.php), we used views on tables to create data hierarchies. Using the object-relational model, only relationships that have been explicitly modeled can be implemented. The relationships between sights and topics are more complex and implicit. The object-relational model does not support inheritance for relationships; instead one is obliged to use objects to capture relationships and their inheritance. Further details of the data modeling aspects of the implementation can be found in (Löffler, 2004) and (Hinze, Loeffler, & Voisard, 2004).

As a result of the implementation phase, we found that neither approach completely fulfills all of our four requirements on modeling technique. In both cases, some of the features need to be implemented using workarounds to address shortcomings of the storage systems. Effective querying is limited in the RDF implementation. On the other hand, the object-relation system is more rigid than that RDF system. Thus, the TIP core approach works successfully upon a fixed architecture of semantic clusters. This aspect had to be readdressed in version 2 of the TIP system.

Remodeling for a Growing System

We argued that RDF was the most appropriate data model for TIP but, nevertheless, for efficiency and effectiveness reasons used an implementation based on an object-relational database structure. Over time, a number of new services

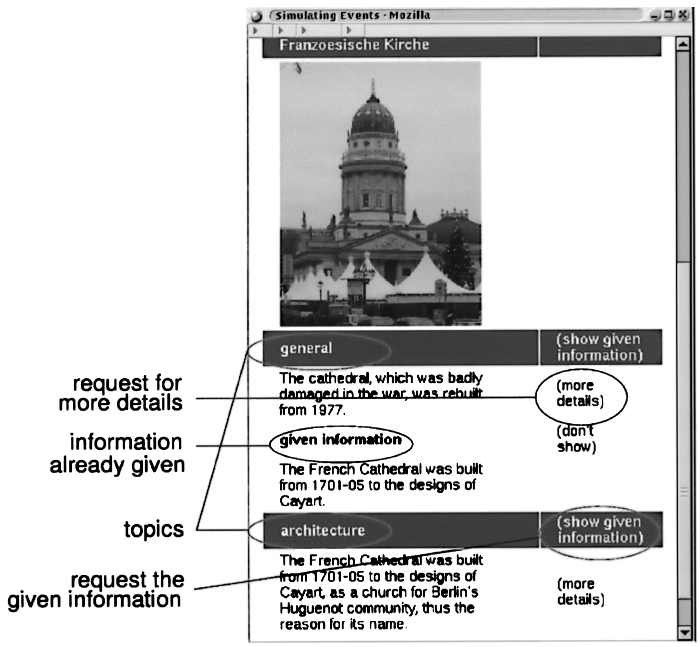

Figure 7. Information provided to Anne at the French Cathedral (TIP screenshot).

were included in the TIP system. This had implications on the data model, as now the initial database was extended, for example, to store users' feedback about the sights they visited and a friend list of users (Hinze & Junmanee, 2006; Qiu & Hinze, 2008). In addition, with increasingly complex applications, the need arose to include new tourism information into the database in a semiautomated way. For a new TIP system generation, we analyzed the existing database structure and the usage of the data.

Requirements for System Growth

We found that the initial TIP database did not sufficiently support system growth. The issue was already discussed (MTR4) but became more pronounced as the system and its applications grew. We briefly summarize issues identified in the data structure.

P1. Flexibility in schema design: The model for sights in the database structure is not flexible enough for a growing TIP system. The subclasses (e.g., "building") of the "sight" table are hard-coded as tables. If a new subclass (e.g., landscape) needs to be added into the database, the programmer has to create a new table and to adapt the program code as the necessary information may not be available at design time. In practice, the sight-group structure and the relationship between sight and sight-group may vary from case to case. A new model needs to be able to reflect complex and comprehensive situations at run time.

P2. Usage: Analyzing the code, we realized that the data stored in subclasses of the sight table were not accessed by the new services as the structure is not sufficiently general enough. The programmer would have needed to create additional variations for each possible type of sight. In addition, it was not possible for users to subscribe to aspects stored in these tables (e.g., to be shown material and dimensions of statues). As a result, the information stored in those tables is virtually lost.

P3. Richer semantics: The semantic clusters in the original model were not rich enough to sufficiently describe relationships between sights.

We were able to model that the German Cathedral and the *Gendarmenmarkt* are two sights that share a location (cathedral on the place). However, we also want to express richer semantics of belonging: a statue in the German Cathedral is a separate sight but also *part of* the cathedral. Note that shared location alone may not indicate such a subsight status. It was not possible to express this distinction in the TIP model introduced above.

Enhanced Data Model

The three issues identified above were addressed in several steps.

Step 1. The rigid structure, surrounded by the dot frame in Figure 5, was refined to a more generic structure to represent complex relationships between sight and sight-group. In the refinement, the hierarchical structure between sight-groups is now defined in "subsightgroup" (see Fig. 8 top right). The refinement allows a sight to belong to more than one sight-group. A sight-group can inherit from more than one upper sight-groups and from different ancestors. These new features can be applied to the situation like the following example. "Eiffel tower" is a sight, and it belongs to sight-group "Tower." The ancestor of sight-group "Tower" is building. On the other hand, "Eiffel tower" can also belong to sight-group "19th century's building." This case could not be handled by the original TIP model. As a consequence, the hierarchical relationship between the sight table and its sight-group (subtables surrounded by dot frame in Fig. 5) does not need to be specified at design time but at run time. This (1) makes the system more flexible and (2) the structure allows to better model complex situations in the real world.

Step 2. In a similar extension, we now allow hierarchies of sights (see Fig. 8 top). It was not possible to express this relationship with the old model. The hierarchical relationship between sights can now be defined at run time and the hierarchies can vary depending on the actual sit-

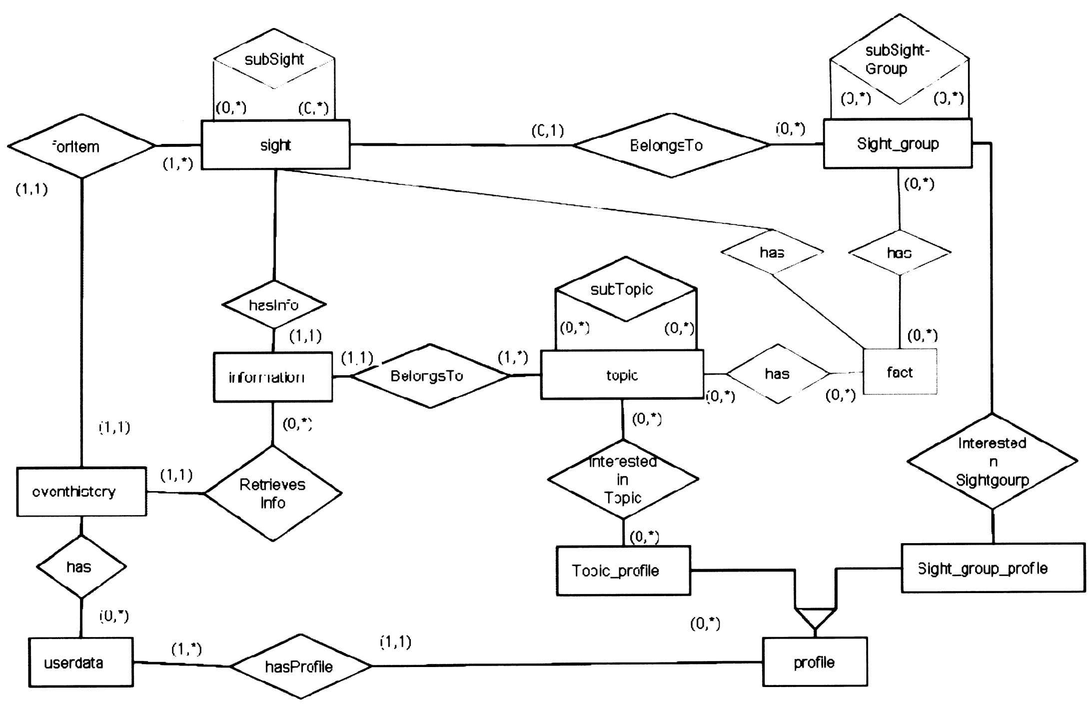

Figure 8. Refined structure for next TIP version.

uation. The refinement replaces the previously hard-coded complex structure (dotted line in Fig. 5).

Step 3. The data stored in the tables surrounded by the dotted frame in Figure 5 were not used in the TIP service extensions. Each of those tables defines which attributes a particular sight-group has; the data describe the sight according to the sight-group's attributes. The refinement shown in Figure 8 opens a new way to store those sight-groups' attributes and their values: each sight and sightgroup has facts assigned. The fact entity is used to represent all attributes that sight-groups could be have. The relationship between sight-group and fact is used to specify the attributes of each sight-group. This is a generic way of representing all tables' structures inside the dot frame in Figure 5. Furthermore, the data in those tables will be stored in a "sight has fact" relationship. We observe that there is a circle in this new model. Therefore, the order in which to fill the tables should be born in mind: firstly, we determine which sight-groups the sight belongs to. Secondly, we identify which attributes those sight-groups have according to the relationship "sight-group has fact." Lastly, the value of a fact that the sight has can be stored in "sight has fact" relationship. For example: sight "Eiffel Tower" belongs to sight-group "tower." "Tower" has these facts: material, height and usage. The values of the facts, which "Eiffel Tower" has such as "made of steel," "300 meters height," "radio and TV transmission," are stored in "sight-has-fact" relationship with sight id, fact id, and value.

The relationship "topic has fact" is used to build connections between topics and sight-group's attributes. This is a new feature, which allows storing data only once and combine it as needed. For example, "material," "height," and "usage" can be linked to topic "architecture." In this way, users can retrieve more detail information about the fact from different accesses such as from sight, topic or sight-group.

Implementation of an Enhanced TIP

A design goal for the new version was enhanced flexibility and modularity of the system—exchanging or adding new components and or data to the TIP server should not lead to complete restructuring of the implementation and models. Another design aspect is the application of open standards, frameworks, and software packages. As a result, TIP uses a pure Web browser interface and is designed to be easily adaptable regarding the various client devices and communication protocols. We refrained from using XML-based communication since the high data load is adversarial in a mobile environment and application of compression is restricted by the limited battery power of the targeted client devices (converting and additional latency).

The second version of TIP has a multilayered client/server architecture (see Fig. 9). The business logic of the application uses the Jakarta Tomcat application server. The application uses Java Servlets and Java Server Pages, which dynamically interact with a Web server for displaying dynamically created HTML sites. The Tomcat server provides the runtime environment, and an Apache web server provides the HTTP access for client communication. The control of the multilayered application software is in the Struts framework. The framework implements the model-view-controller (MVC) design pattern, which ensures the encapsulation of the control data. Another advantage

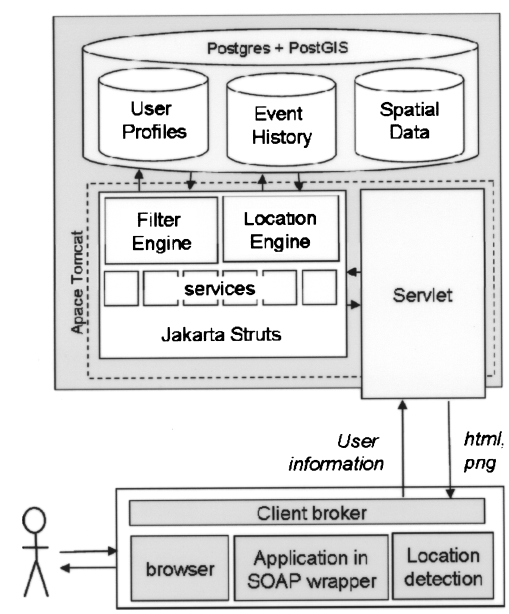

Figure 9. TIP 2 implementation.

is the independence of application logic and application data. This independence is a necessary condition for the reusability and modularity of the components of the system. The control layer manages the user interaction and translates these data into the application model. The application data regarding users and sights are stored in a Postgres database that uses the spatial extension postGIS.

An alternative approach using P2P for client communication was designed but not yet fully implemented. The implementation used J2ME, the Java environment developed for mobile devices, and the Wireless Toolkit (WTK). A restriction is currently the lacking support of mobile hardware for J2ME and the resulting limited functional range. We hope to extend this approach for personal ad hoc recommendations when the necessary functional and technical support for P2P systems is available. The implementation is described in more detail in Ottlinger (2004). Figure 10 shows screenshots of the TIP 2 system presenting recommendations to the user.

Exploiting the TIP Architecture

In the previous section, we discussed the development of the TIP architecture from the first prototype to a more advanced and flexible system. The current TIP infrastructure is a clear advance from early tourist context-aware architectures such as Guide (Cheverst et al, 2002). However, questions may now be asked of the TIP system from the developer and user side, respectively: firstly, to what degree can TIP as advanced software architecture be repurposed; and secondly, to what degrees can a real-life user exploit what TIP offers?

TIP has a number of features that allow for its adaption to other tasks. For example, its focus on monumental buildings in a cityscape could readily be modified to supporting the interpretation of artifacts in a museum. The order in which a visitor sees different exhibits would naturally need to be considered in comparing between items, and if a visitor is consistently interested in objects from a particular place or culture, this could be usefully used in tailoring information. Similar concerns could also be discovered when familiarizing university students with their campus, or a visitor to a public library. A variant of TIP has been used in interpreting a high street's history over the 20th century. The movement to a more model-independent scheme with generic behaviors reported in Section 4 naturally enhances the readiness with which TIP could be modified for these different scenarios. We also found that the TIP data model could be used for modeling people's memories (Schweer & Hinze, 2007).

Especially in context-aware systems as TIP,

Figure 10. TIP 2 screenshots of recommendations based on proximity and profile.

care must be taken to match the system design with the specific user needs of the environment: adoption is closely connected with a good fit with the user's task. It has often been argued that a technology-based approach to context cannot fully capture the intricacies of user preferences. However, it has also been observed that users prefer simple rule-based systems over complex ones (e.g., using neural networks and machine learning) as users better understand, and therefore trust, the system. For increased user acceptance, we incorporated social networking aspects for feedback and recommendations, and studied user trust in tourism information (Qiu & Hinze, 2008). The question of whether people have a limited travel experience when guided by an electronic system instead of freely exploring a city is a philosophical one. Clearly, different tourists will prefer different travel approaches. However, compared to a traditional guide book, systems like TIP allow in-time personalized information access (via search or by triggered information delivery) tailored to their current situation. Moreover, TIP's diverse services offer a wide variety of further options to explore the tourist's surroundings, such as armchair traveling, being guided by audio books or by friends' preferences, and access to historic maps.

Summary and Conclusion

In this article, we focused on the modeling aspects of an advanced tourist information system. Beside dimensions such as time and location, common tourist information systems consider the profile of the mobile user (e.g., their preferences and interests). However, our system goes beyond such a classical approach and aims at offering two novel features: (i) taking the history of the user into account—in other words, the system "knows what the user knows" and (ii) links between sight-related information (i.e., a semantic network of tourist information).

We presented a sample scenario to derive the functional requirements of an advanced mobile tourist application. These requirements were then used to examine two different data modeling approaches in detail: a semantic web approach using RDF and an object-relational approach. Finally, the implementation of the kernel of the system was briefly described and the main implications of the two contrasting data models were identified. Our conclusions regarding the two different modeling approaches are as follows: the object-relational model has the clear advantage of being a well-understood, highly developed, and technically well-supported model. In contrast, the RDF model and storage systems are still, in comparison, embryonic. For the implementation of TIP, the object-relational modeling was preferred over the RDF-based modeling. However, we are convinced that more mature storage systems for RDF and related formats will be the appropriate means of choice.

As the system grew, we observed that the data to be modeled often had to be tailored for the model, whereas the final goal is to have a means of modeling that suits the inherent characteristics of the data. We thus refined our model further to reflect a more general structure of the types, groups, and cluster of sights. In doing so, we moved further away from the initial RDF/S model towards a more abstract relational database model. However, as the model developed, limitations of typical relational databases when used for semantic web applications emerged. For example, relational databases allow their application users to query only for data instances (table entries) but not for the schema—schema queries are typically reserved to database administrators. The semantic RDF/S model directly supports such schema queries. The same functionality could be achieved in the relational database only through workarounds: complex "meta" tables capture the semantic schema information within relational table entries. This structure makes it much harder to see the semantic schema information.

In conclusion, we found the RDF/S model to be conceptually superior for semistructured data and concepts. However, current RDF/S query language implementations still not fully support these advantages. In practice, the traditional database approach allowed us to directly implement the desired queries, but the model became indirect, concealing the actual data structure. Our future work relates to the TIP application development as well as further architectural developments. We believe that the semantic network of sight-related information may be further enhanced based on connections derived from user history, user preferences, and user comments. This would involve data mining on the semantic web of user histories and subsequently changing the web. We addressed the performance issue created by reanalyzing the histories of all the users by introducing a trust measure (Qiu & Hinze, 2008). This may be further improved by an incremental algorithm that would work on both semantic classes and instances. In our work on the third version of the TIP system, we explore in depth the mobility aspect of the system while considering collaboration between a number of location-based services (Hinze, Michel, & Eschner, 2009). We address user acceptance issues and interaction design for the location-aware service collaborations in TIP 3.

Acknowledgments

We have been fortunate to have worked with many gifted students in this project. We particularly wish to thank the following students for their contribution to the TIP system implementation: Katja Loeffler, Phillip Oettlinger, Sven Bittner, Steven Koenig, Doris Jung, Saijai Junmanee, Quan Qiu, Jenny Chen, Andy Wang, Xiaotie Huang, Xin Gao, Wenyi Lin, Ping-Ju Hsieh, Cindy Thunack, Yann Michel, Lisa Eschner, and Andrea Schweer.

Biographical Notes

Annika Hinze is currently a guest Professor at the Humboldt University Berlin for the area of context-aware systems. She is also a Senior Lecturer (Associate Professor) at the University of Waikato, New Zealand, where she leads the research group on Information Systems and Databases (ISDB). Her research focuses on context-aware systems, event-based systems, interaction design, and digital libraries. She received her Ph.D. degree in Computer Science from the Freie Universität Berlin in 2003. Together with Alejandro Buchmann, she coedited the book *Principles and Applications of Distributed Event-Based Systems* (2009).

Agnès Voisard received her Ph.D. degree in computer science from the University of Paris at Orsay (Paris XI) and INRIA in 1992. In 1993, she was appointed Assistant Professor of Computer Science at the Freie Universität Berlin. She is now head of the department Location-based Services at the Fraunhofer Institute for Systems and Software Engineering (ISST) in Berlin as well as adjunct faculty at the Freie Universität Berlin, Institute of Computer Science. Her areas of expertise include geographic information systems, location-based services, and event-based systems. With Philippe Rigaux and Michel Scholl, she coauthored the book *Spatial Databases—with Application to GIS* (2001). With Jochen Schiller she coedited the book *Location-based Services* (2004).

George Buchanan is currently a Senior Lecturer (Associate Professor) in the Centre for Human-Computer Interaction Design at City University, London. His research interests center on the usability of information systems, particularly context-aware systems, digital libraries, and small-screen information access. In addition to his work at City, he is also a member of the New Zealand Digital Library Group, contributing to the popular Greenstone digital library system. Previous to his academic career, George ran a successful desktop publishing software development business after taking his first degree in Computer Science at the University of York, UK.

References

Abowd, G., & Atkeson, C., & Hong, J. (1997). Cyberguide: A mobile context-aware tour guide. In *ACM wireless networks*(Vol. 3, pp. 421–433).

Aidministrator Nederland b.v. (2000). The SeRQL query language (Chap. 5). In *User guide for Sesame 0.96*. Retrieved from http://sesame.aidministrator.nl/publications/users/ch05.html

Biezunski, M., Bryan, M., & Newcomb, S. (2002). *ISO/IEC 13250, topic maps*. Retrieved from http://www.ornl.gov/sgml/sc34/document/0058.htm

Bönström, V., Hinze, A., & Schweppe, H. (2003). Storing RDF as a graph. In *Proceedings of the First Conference on Latin American Web Congress, LA-WEB*, November 10–12, 2003, LA-WEB. IEEE Computer Society, Washington, DC.

Brickley, D., & Guha, R. V. (2003). *RDF vocabulary description language 1.0: RDF schema*. Status: Working Draft. Retrieved from http://www.w3.org/TR/rdf-schema/

Broekstra, J., & Kampman, A. (2002). *Query language definition*. Technical Report On-To-Knowledge EU-IST-1999-10132 deliverable 9.

Broekstra, J., & Kampman, A., & van Harmelen, F. (2002). Sesame: An architecture for storing and querying RDF data and schema information. In *Proceedings of the First International Semantic Web Conference (ISWC 2002)*. Volume 2342 of LNCS, Springer Verlag.

Cheverst, K., Davies, N., & Mitchell, K. (2002). The role of adaptive hypermedia in a context-aware tourist guide. *Communications of the ACM, 45*(5), 47–51.

Dey, A., & Abowd, G. (2000): Towards a better understanding of context and context-awareness. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*.

Guarino, N. (1998). Formal ontology and information systems. In N. Guarino (Ed.), *Formal ontology in information systems*. IOS Press.

Haase, P., Broekstra, J., Eberhart, A., & Volz, R. (2004).

A comparison of RDF query languages. In *Proceedings of the Third International Semantic Web Conference*.

Hinze, A., & Junmanee, S. (2006). Advanced recommendation models for mobile tourist information. In *OTM Conferences* (1, pp. 643–660).

Hinze, A., Loeffler, K., & Voisard, A. (2004). Contrasting object-relational and RDF modeling in a tourist information system. In *Proceedings of the tenth Australian World Wide Web Conference (AusWeb)*, Gold Coast, Queensland, Australia.

Hinze, A., Michel, Y., & Eschner, L. (2009). Event-based communication for location-based service collaboration. In *Twentieth Australasian Database Conference, ADC 2009*, Wellington.

Hinze, A., Sachs, K., & Buchmann, A. (2009). Event-based applications and enabling technologies. In *Proceedings of the ACM Conference on Distributed Event-based Systems (DEBS)*, Nashville, TN.

Hinze, A., & Voisard, A. (2003). Location- and time-based information delivery in tourism. In *Advances in Spatial and Temporal Databases (SSTD 2003)*, LNCS 2750.

Holtkamp, B., Gartmann, R., & Han, Y. (2003). FLAME2008: Personalized web services for the Olympic Games 2008 in Beijing. In *Proceedings of the e-Challenges Workshop*.

Joseph, A. D., Beresford, A. R., Bacon, J., Cottingham, D. N., Davies, J. J., Jones, B. D., Guo, H., Guan, W., Lin, Y., Song, H., Iftode, L., Fuchs, S., Lamprecht, B., Kyamakya, K., Fernandez, J. G., Garcia, J. C., Garcia, Y. S., Santos, J. d., Nimesh, M., Pan, G., Wu, Z., Wu, Q., Shan, Z., Sun, J., Lu, J., Yang, G., Khan, M. K., & Zhang, J. (2006). Intelligent transportation systems. *IEEE Pervasive Computing 5*(4), 63–67.

Kabus, P., & Buchmann, A. (2007): A framework for network-agnostic multiplayer games. In *GAME-ON International Conference on Intelligent Games and Simulation*.

Klante, P., Krösche, J., & Boll, S. (2004). AccesSights—a multimodal location-aware mobile tourist information system. In *International Conference on Computers Helping People with Special Needs, ICCHP*.

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

Lassila, O. (1998). Web metadata: A matter of semantics. *IEEE Internet Computing, 4*, 30–37.

Lassila, O., & Swick, R. R. (2003). *Resource description framework (RDF) model and syntax specification. Status: W3C recommendation*. Retrieved from http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

Löffler, K (2004). *User-adapted information delivery in context-aware systems*. Master's thesis, Freie Universität Berlin, Germany, Department of Computer Science.

Meissen, U., Pfennigschmidt, S., Voisard, A., & Wahnfried, T. (2004). Context- and situation-awareness in information logistics. *EDBT Workshop on Pervasive Information Management*.

Nejdl, W., Wolpers, M., & Capelle, C. (2000). The RDF schema specification revisited. In *Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik*, Modellierung 2000.

Ottlinger, P. (2004). *Design and implementation of an extensible software architecture for distributing context-sensitive information* (in German). Master's thesis, Freie Universität Berlin, Department of Computer Science.

Pashtan, A., Blattler, R., Heusser, A., & Scheuermann, P. (2003). CATIS: A context-aware tourist information system. In *Proceedings of the 4th International Workshop of Mobile Computing*.

Poslad, S., Laamanen, H., Malaka, R., Nick, A., & Zipf, A. (2001). Crumpet: Creation of user-friendly mobile services personalized for tourism. In *Proceedings of the International Conference on 3G Mobile Communication Technologies*.

Qiu, Q., & Hinze, A. (2008). *Trust-based recommendations for mobile tourists in TIP*. Technical report, University of Waikato.

Rigaux, P., Scholl, M., & Voisard, A. (2001). *Spatial databases—with applications to GIS*. San Francisco: Morgan Kaufmann.

Seaborne, A. (2002). *A programmer's introduction to RDQL. HP Labs*. Retrieved from http://www.hpl.hp.com/semweb/doc/tutorial/RDQL/

Schiller, J., & Voisard, A. (Eds.). (2004). *Location-based services*. San Francisco: Morgan Kaufmann.

Schweer, A., & Hinze, A. (2007). The digital parrot: Combining context-awareness and semantics to augment memory. In *Memos Workshop at the British HCI International Conference*, September.

Sheikholeslami, G., Chang, W., & Zhang, A. (2002). SemQuery: Semantic clustering and querying on heterogeneous features for visual data. *IEEE Transactions of Knowledge and Data Engineering, 14*(5), 988–1002.

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

Simcock, T., Hillenbrand, S. P., & Thomas, B. H. (2003). Developing a location based tourist guide application. In *Proceedings of the Australasian information Security Workshop Conference on ACSW Frontiers 2003*, Adelaide, Australia, at Conferences in Research and Practice in Information Technology Series (Vol. 34, pp. 177–183). Darlinghurst, Australia: Australian Computer Society.

Wu, M., Liu, A., & Mani Chandy, K. (2008). *Virtual environments for developing strategies for interdicting terrorists carrying Dirty bombs*, ISCRAM.