Data Semantics in Location-Based Services

Nectaria Tryfona and Dieter Pfoser

Research Academic Computer Technology Institute,
Akteou 11, 1851 Athens, Greece
`{tryfona, pfoser}@cti.gr, htp://dke.cti.gr`

Abstract. As location-based applications become part of our everyday life, ranging from traffic prediction systems to services over mobile phones providing us with information about our surroundings, the call for more semantics and accurate services is emerging. In this work, we analyze and register the data semantics of Location-based Services (LBS). Initially, we categorize LBS data according to the related concepts and use. We distinguish the (a) Domain Data, including spatial and temporal concepts, namely, position, location, movement and time, (b) Content Data, describing the LBS specific content, and (c) Application Data, consisting of the user profile and the services provided by LBS. Next, we model these three data categories in a way that captures their peculiarities and allows their sharing and exchange among different LBS, when desired. For this, we use semantically rich and expressive models, like UML, as well as the long-praised method of ontologies, realized in the open source, ontology and knowledge-based editor Protégé. To argue about the design choices and show their applicability, we present examples from two characteristic real-world applications, both in the Athens Metropolitan Area: an LBS for tourists carrying mobile devices, and a traffic LBS informing drivers about troublesome situations.

1 Introduction

In the recent years Location-Based Services (LBS) enjoy much attention from both the scientific community and the industry. Work has mostly been concentrated on *delivering* information to the mobile user that is related to his/her location and therefore, presumably, more relevant. Additionally, the technological revolution in this area (e.g., advanced capabilities of handheld devices) as well as commercially oriented solutions to customer's needs (e.g., fast transmission of multimedia data such as, images and video) have driven the focus of LBS away from what they really are: services supported by non-conventional databases, characterized by the spatial and temporal dimension, i.e., spatiotemporal databases. Due to this, data involved in LBS have not been really examined in depth. Consequently, LBS data semantics are not captured properly, LBS data models do not fully accommodate application requirements, and the final system does not always meet user needs.

In this work, we treat LBS as non-conventional applications. In these, it is important to understand and register the related concepts. Initially, we analyze the data

S. Spaccapietra and E. Zimányi (Eds.): Journal on Data Semantics III, LNCS 3534, pp. 168 – 195, 2005.

scenario in LBS and categorize data according to its semantics and use. We distinguish the (a) *Domain Data*, including spatial and temporal concepts, (b) *Content Data*, describing LBS specific content, and (c) *Application Data*, consisting of the user profile and the services provided by the LBS. The goal is to model these three data categories in a way to (i) capture their peculiarities and (ii) allow the sharing and exchange among different LBS. For the second goal, we use semantically rich and expressive models, like UML [4], while for the first one, we adopt the long-praised method of ontologies, realized in the open source, ontology and knowledge-based editor Protégé [38].

To argue about the design choices and show their applicability, we present examples from two characteristic real-world applications, used as case studies, running in the Athens Metropolitan Area: a tourist LBS, in which travelers are carrying mobile devices [8] [37] and a traffic LBS informing the drivers about troublesome situations and alternative routes [24] [5].

Domain Data includes spatial and temporal concepts captured as the object's *position*, *location*, *movement* and *time*. A systematic study reveals that these four spatiotemporal concepts are common and fundamental in all LBS, whether it is, for example, a traffic or a tourist LBS. Thus, it is crucial to share and exchange their semantics. This is achieved by analyzing and model the characteristics and relations of these spatiotemporal concepts. For this, we propose the use of the well-known and long-praised method of ontologies [18] [14] [39], focusing on the comprehension, registration and design of Domain Data. In order to easily realize the ontology, we use the Protégé tool [38]. However, the use of the Protégé tool is just a prototypical one. Any other tool, or standard language such as DAML+OIL [7] would do for this representation.

Special care of location is taken. Until now, all LBS are based on the crude assumption that the location of the mobile object (e.g., a car or a tourist in our case studies) can be simply unambiguously determined; that is, it is *always known* and in *absolute measures*. However this is neither true nor sufficient. In some cases, the position of a moving object is not known, such as when the GPS device is shadowed. In other cases, the user not only cares about her absolute position, but also her surroundings. For example, when tourists visit an archaeological site, the location that matters to them is the actual position in terms of coordinates, as well as a circular 'shape' *around* their current position, which 'includes' items of interest. To capture these semantics, we propose a clear distinction between *location* and *position*. This serves also the need for a better representation, exchange and integration of location from multiple sources, an open problem and a challenge in LBS [26].

Content Data depends on the specific application we are dealing with, e.g., for a tourist LBS this data includes historic facts, restaurant and hotel information. In this work, we model an excerpt of the Content Data existing in a tourist and a traffic LBS by using ontologies for the former LBS and the UML technique for the latter LBS. In the case of the traffic LBS, this leads to the definition and organization of a Moving Object Database (MOD) which includes trajectories, vehicles, routes and their relations and serves as the backbone of the Athens traffic management system. Dealing with these two different LBS scenarios and by using different techniques (i.e., ontologies and UML) shows the diversity of Content Data and consequently the different semantics and design needs. Moreover, it argues that our modeling choices are not tied to specific technology, models and tools.

The third data category comprises *Application Data*, capturing the user profile and service data for the two characteristic application examples. Ontologies are used to denote the data and their semantics. In an LBS scenario, relevant services are discovered, by matching the respective service description with user profiles.

Modeling the semantics of the three data categories leads to the creation of three ontologies: the Domain Ontology, the Content Ontology and the Application Ontology. This structure serves as the backbone architecture to support LBS based on ontologies, with special focus on autonomy and share. To summarize, the contribution of the paper is threefold:

- The categorization of data involved in LBS, based on (data) semantics and use.
- The clear distinction between *location* and *position* in LBS, which solves ambiguities and makes assumptions clear.
- The creation of ontologies (i.e., Domain, Content and Application) for LBS, to represent, share and exchange the concepts of location, position, movement and time among location-based applications.

The rest of the paper is organized as follows: Section 2 gives related, characteristic work focusing on capturing the semantics of LBS. Section 3 presents the types of LBS applications and the Domain, Content and Application Data. Section 4 focuses on the Domain Data and the fundamental related concepts. A clear distinction between *position* and *location* is given; the temporal dimension is treated in a similar way. Section 4 further argues for the use of ontologies in the semantics representation and presents examples in Protégé. Section 5 deals with the Content Data of LBS; it discusses the traffic content data and models their semantics in UML, focusing on the organization of MOD for the traffic management system, while the content data of the tourist LBS are captured in Protégé with ontologies. Section 6 models the Application Data for both the traffic and the tourist LBS with ontologies, and Section 7 concludes this research effort.

2 Related Work

To the best of our knowledge, literature on capturing LBS semantics is quite limited; work has mostly been concentrated on issues related on how to deliver information to the mobile user, rather than what information and semantics are delivered. However, the presence of *location* and *time* play a central role in LBS, and this calls for more rich and complex semantic modeling techniques to capture data involved in the requested services.

In the few existing proposals ([46] [47] [48] [49] [33] [35] and [44]), the use of ontologies has been adopted for this purpose, and quite understandable so, since literature shows many efforts, in other research areas (e.g., biology or business), in which ontologies are used for the analysis and representation of semantics of information.

Ontologies can capture the semantics of information, can be represented in a formal language, and can also be stored to related metadata, thus enabling a semantic approach to information integration. There are several ontology languages as, e.g., compared in [15] and tools to represent ontologies [28] [6] [7]. Protégé [38] is the most popular ontology-editing environment and has been used in many applications, such as medical systems, gene ontology, and business systems.

Furthermore, already, a wide range of applications, such as geographic and biological, call for techniques flexible enough to capture their particularities with respect to space; yet formal ones. [11] proposes a framework for the development of geographic applications by using ontologies. In [11] the reader can also find a systematic review on literature on the use of ontologies in GIS. [3] identifies the role of ontologies in capturing spatial uncertainty. [12] and [13] present methods to bridge the gap between conceptual schemas and ontologies in Geographic Information Systems. Finally, [13] presents the Ontology-Driven Geographical Information Systems framework (ODGIS), which uses ontologies for the comprehensive usage of ontologies for classification purposes, focusing on integrating different kinds of geographic information.

There are some arguments about how useful ontologies are. [20] advises against using ontologies as just a fancy name denoting the result of activities like conceptual analysis and domain modeling [12]. Our position is that ontologies are built to model the semantics of a domain and represent, share and exchange knowledge, while data models and conceptual modeling focus on organizing explicit data and contents resulting in a database. Section 4 elaborates further on this.

Work on ontologies and LBS includes [46] [47] [48] [49] [33] [35] and [44]. They are all based on the assumption that the location is a point with known coordinates.

[46] describes issues involved in supporting an ontology-based information searching process in LBS. It presents an example scenario and gives an architecture based on ontologies that is to be adopted to support share and autonomy in LBS. [49] proposes a collaborative framework for location-based information management consisting of the Query Engine, the Profile Manager, the Data Handler, the TOP Hits Repository, the Data Repository and the Adding Filter. This framework makes it possible to obtain information from heterogeneous sources and improve the request-response efficiency. [33] proposes an ad-hoc model to locate correlative data stores and exchange similar information within a specific community. The model is composed by Data Handlers, Data Stores and proxies and uses ontologies to deal with the spatial relationships between the moving objects. Its continuation [35] proposes the use of ontologies for the management of services in LBS. The proposal exhibits similarities to the newsgroup approach in that both 'systems' are examples of semantic search engines based on user interaction. [44] gives a modular ontology architecture to support different existing ontologies and metadata standards for the web services in Olympia 2008.

The user profile plays an important role in LBS. [47] proposes a profile-based approach to improve the efficiency of the LBS, based on a relational database. As a next step, [48] proposes a way to accommodate user profile needs by using domain and content-depended ontologies. It also suggests the multi-layered abstraction method to organize and present data related to profiles. In this framework, [22] describes a system, which delivers various types of information to mobile devices based on the location, time and profile of the end user. The Event Notification technique has been adopted to trigger actions.

[23] proposes a semantic location model for navigation in mobile environments. It is a hierarchical model and captures connectivity and hierarchical relationships. Again, the assumption here is that location is a point with known coordinates. [42] deals with different types of locations, the way to compute them and to present them. Although the work presented there gives a first taxonomy and general directions about how to handle location in LBS, there is no typical way to categorize them, model them and communicate them with a formal technique, such as a model, ontologies or mathematical representation.

Finally, at the level of services, it is important to point out the effort in achieving an open location services platform (http://www.openls.org/).

3 Types of LBS Applications and Categories of Data

The domain of LBS applications is large and diverse. Here, we present the types of applications supporting location-based services, and analyze the categories of data involved.

3.1 LBS Applications

The GSM Alliance Service Working Group [19] has defined the following types of traditional LBS:

- Emergency Services
- Emergency Alert Services
- Home-zone billing
- Fleet Management
- Asset Management
- Person Tracking
- Pet Tracking
- Traffic Congestion Reporting
- Routing to Nearest Enterprise
- Roadside Assistance
- Navigation
- City Sightseeing
- Localized Advertising
- Mobile Yellow Pages
- Network Planning
- Dynamic Network Control

As LBS, we consider any application involving moving objects and providing services based on positional, temporal and, many times, user profile[1] information. This definition supports the GSM categorization. The position of the object is usually pro-

---

[1] Some works in literature (e.g., [22]) consider the presence of space, time, and user profile *mandatory* to define an LBS; other information, such as ‘history’ may also exist. However this restrictive definition contradicts the LBS categories of the GSM Alliance Service Working Group; for example the Traffic Congestion Reporting service does not require user profile information. Without affecting the validity and applicability of our results, we chose not to consider mandatory the user profile information.

vided by a mobile device, such as a GPS, carried on/by the moving object or, in the rougher cases, by using positioning in cellular networks.

Two real-world, characteristic, and very different LBS applications are used as case studies in this work:

- a tourist information system providing services based on tourist's location, time and profile [8]. The tourist is equipped with a handheld device having GPS capabilities. Consider, for example, the scenario in which he/she is in the archaeological site of Acropolis and asks 'give me the history of Parthenon', or 'what is the closest monument to me?' or 'what artifacts were found here?'. A tourist LBS should provide answers to these queries. Furthermore, the tourist should be able to provide a profile or preferences, and get information relevant to his interests. For example, a user visiting Acropolis might be interested only in information related to Acropolis and the Pericleus era, i.e., [495BC-429BC].
- an LBS system for traffic management, in which vehicles are equipped with GPS devices [24]. The driver can ask questions such as: 'based on my position, where is a traffic jam?', or 'if there is a traffic jam in the next 10 km, give me alternative routes' or 'give me suggestions to go from Athens to Piraeus'. In this example, it is clear that the user profile is not mandatory, since the user might not have any preferences.

For simplicity purposes and without affecting the validity of our results, we assume that, in the two aforementioned applications, we deal with moving-point objects, i.e., the absolute position of the person or the vehicle that moves is a point.

3.2 Categories of Data in LBS

An important task when building a system, is the analysis and comprehension of the categories of data involved in it, i.e., the related concepts, semantics and use. This helps not only in providing, later on, the appropriate techniques to model and communicate these data, but also to accurately understand the requested services and meet the system's requirements.

Analyzing and comprehending the system data is a modular process: first, we realize the dominating types of data and then the more specific ones. Here, we present three data categories, and show their interconnections; Sections 4, 5 and 6 analyze each one of these categories, give examples and elaborate on the interconnections among them.

In LBS applications, we distinguish three categories of data:

Domain Data. It includes the concepts that are present and characterize all LBS applications.

The common factor behind all LBS is the spatial and temporal dimension, and thus, Domain Data includes fundamental spatial, temporal and spatiotemporal concepts. A careful look across different LBS described by the GSM Group, shows that objects position, location, movement, as well as *time* characterize all of them, whether, for example, we talk about a tourist LBS or a traffic LBS.

Position and location are two terms full of assumptions, which are often used interchangeably. However, in LBS, there is the need to express special meanings and semantics with respect to space. In some cases position refers to absolute coordinates, for example, a car on a road network, while in others, what matters is a greater surrounding area, for example, a tourist wants to know the closest restaurants within 200km radius around him, or the spread of a traffic jam. To better capture semantics, we chose to distinguish these two cases: in reality, the spatial dimension introduces *two new concepts in LBS*: objects' *position* and *location*. Similar issues hold for the temporal dimension; thus, we chose to use the concepts *timestamp* and *time horizon*. A systematic analysis of these concepts reveals different semantics and relations with the environment (Section 4).

Content Data. It is the actual data of the specific application we are dealing with. For example, for the tourist LBS, Content Data is the monuments, the parks, restaurants, etc., while for the traffic LBS is the traffic data, route, and others.

Content data can be: (a) descriptive, for example, restaurant names, or description of museums, (b) spatially-referenced, indicating *where* the actual information is located, seen, or recorded, for example, the location of a museum, and (c) temporally-referenced, showing *when* the information is located, seen or recorded in the system, for example, the time of the traffic jam.

Application Data. It includes the subcategories:

- Profile Data, characterizing the user and the device he is carrying. This can be:
    (i) user profile, capturing the user and its preferences. For example, a user visiting Acropolis may be a tourist or a scientist, indicating different interests.
    (ii) device profile, characterizing the mobile device the user is carrying, for example, CPU capability, memory characteristics, screen size.
- Service Data, which corresponds to tasks to be accomplished in the specific LBS application. For example, provide specific services to tourists, or to drivers.

Figure 1 illustrates the data categories and subcategories. The lower left corner gives the concepts (in the case of Domain Data) and examples (in the case of the Application and Content Data). The Application and Content Data depend on the specific application we are dealing with. For example, if the application is tourist services over the mobile phone, then the user profile is the one of the tourist with services like 'information about surrounding restaurants' and the content ontology has organized data about restaurants, museums etc.

The three data categories are interconnected, since, for example, in order to provide the service 'closest restaurant' a reference to the restaurants index is needed (i.e., Content Data) and to the position of the user (i.e., Domain Data). There is no association between Content and Domain Data. The Content refers to specific geographic information (for example, location of restaurants in Athens), but this is general spatial data, outside the Domain, which refers to the *where* and *when* the user is.

4 The Domain Data

Domain data characterize all different LBS described by the GSM Group. In fact, the spatial dimension appears in terms of the object's *position* and *location*, the temporal dimension appears in terms of the *time* a the desired information is located, seen or requested, and both dimensions participate in object's *movement*. Next, a systematic analysis of these concepts is presented.

4.1 Semantics of Domain Data in LBS

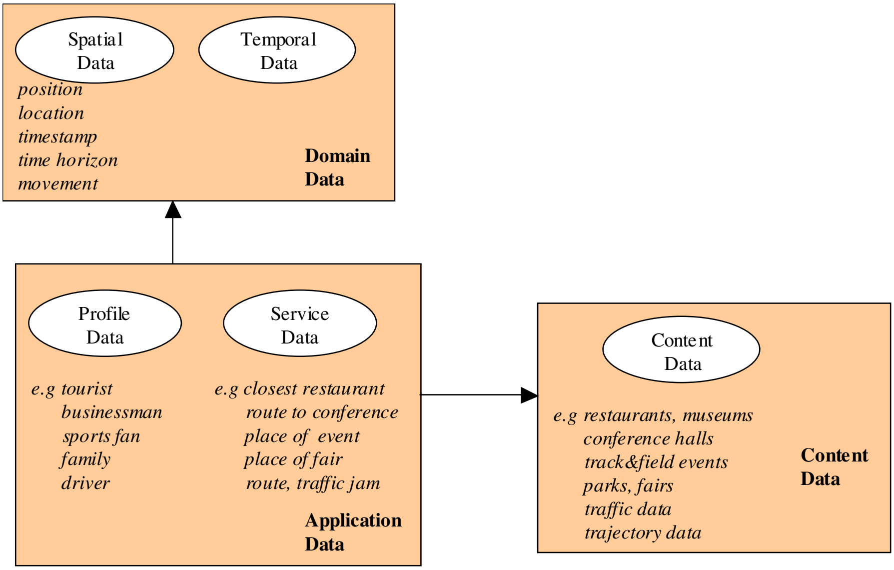


Fig. 1. Categories of data in LBS

A. The spatial dimension introduces the concept of *position* is full of ambiguities and assumptions. Almost in all LBS, there is the crude assumption that the position of a moving object can be simply unambiguously determined [26]; that is, it is *always known* and in *absolute measures*. However this is neither true nor enough. In some cases, for example, the position of a moving object is not known, such as when the GPS device being on it is shadowed.

In other cases, the concept of position has different meanings and values depending on the *application domain* of LBS, and thus cannot be determined by a single notion, it is not unambiguous; thus, it cannot be captured and represented by a unique method or technique. Consider the example of the traffic LBS, in which vehicles are equipped with GPS devices. The absolute, current, *position* of the car is the (x,y) coordinates transmitted by the GPS. However, what really matters to the system to predict and bypass a traffic jam is not just the *position* itself but also the 'shape' or area of 10km *ahead* in the road network, given the fact that the jam is present there.

Analogously, in the tourist LBS providing services to tourists visiting an archaeological site, the *location* that matters to them is the actual (x,y), as well as, the circular 'shape' *around* their current coordinates, which 'includes' items of their interest.

The aforementioned examples are just some of the many we experience everyday indicating that when location *matters*, it is not, only and always, in absolute numbers, but it further depends on the domain of the application.

Moreover, recently, the need to aggregate positions from multiple sources becomes more and more emerging [26]. This is based on the facts that (a) a person may be associated with numerous tracking devices simultaneously, e.g., GPS device on a phone, in a car, etc, and (b) the tracking devices are not always accurate, or may be shadowed as said before, and thus do not deliver the right signal. In this case, the notion of *position* has more than one value and in order to be aggregated, it needs to be analyzed, captured and represented in all possible involved forms.

It is our thesis, that the spatial dimension in LBS is captured by *two new concepts*:

- The absolute (x,y) coordinates of the moving object, which we call ***position***
- the 'surrounding area' of position, which we call ***reach***. The shape of reach can vary: in the first example (i.e., area of interest in the tourist LBS) it is a circle with a given, predefined, radius. In the second one, it is a shape of an oval (spread of a traffic jam in the traffic LBS). The position and reach of the moving object constitute its ***location***. Figure 2 illustrates the two cases.

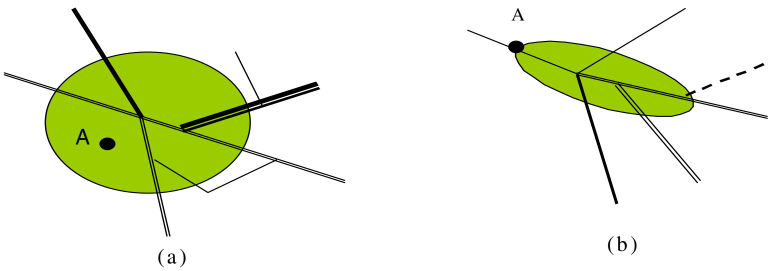


Fig. 2. Different shapes of reach in space: (a) a circular reach surrounding the position A close to a road network, (b) an oval reach ahead of position A on a road network

B. In LBS, it is not only the *where* but also the *when* that matters. Thus, location is related to time. For example, in a tourist LBS it is important to know *when* the tourist is in a location since many facilities depend on that (e.g., when shops are open, etc). Similarly, in traffic LBS for the prediction of a traffic jam, the time a car approaches specific areas matters as rush hours are usually troublesome and matter.

Time, analogously to location, is captured by:

- a ***timestamp*** t indicating when an action or event happens.
- a ***time horizon***, indicating the time period the event or the action has still an effect. For example in the case of a moving vehicle in the traffic LBS trying to avoid a jam, what matters is not only the current time but also the time horizon *ahead* in which the jam will evolve.

Moreover, due to imprecise information, inaccurate measures or device errors, *position*, *reach*, *timestamp* and *time horizon* can have uncertainty, which is usually expressed by the deviation from the accurate value.

C. A fundamental concept in LBS is the ***movement*** of the object. Movement is defined in terms of position of the moving point object and time, and depending on the application needs, it includes some basic concepts, such as:

- heading, which shows the heading of the moving object
- distance, which gives the distance from the previous position
- direction, which shows the angle to the previous position
- duration, which shows the duration traveled from the previous position.

Figure 3 illustrates this design decision.

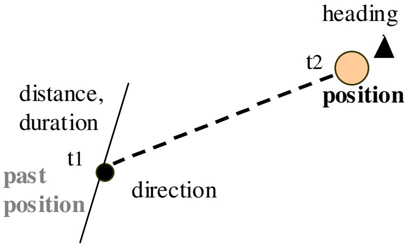


Fig. 3. A diagrammatic description of movement

D. Additionally, the location, i.e., reach and position of the moving object, might be related to others with spatial relationships, which are either topological (e.g, meet, intersect etc) [10], directional (e.g., left, right, etc) or metric, which show distance.

Furthermore, there are applications in which users refer frequently to specific locations for which, neither care nor know their absolute coordinates. For example, the notion of ‘work’ and ‘home’ are obvious to everyone and a reasonable service is to be able to ‘send me all my SMS as approaching *work*’ or to ‘download mail going home’. This calls for the concept of *virtual position* related to the user and not to the coordinates. A known user-centered model is comMotion [25]. The approach we propose here captures also virtual positions as it relates position (and location) to the surrounding environment with spatial relationships (e.g., ‘approaching’ is captured with direction and distance).

4.2 Using Ontologies in LBS

Since Domain Data captures the spatial and temporal dimension and is present in all LBS, the concepts of position, location, time (i.e., time horizon and timestamp) and movement are fundamental and common is all location based applications. This calls for interoperability among LBS, and there is an emerging need to:

- share a common understanding of these spatiotemporal concepts
- make LBS assumptions that exist in literature about these concepts explicit
- exchange and enable reuse of them.

To achieve these goals, we propose the use of ontologies. Here, we focus on the comprehension, registration and design of the aforementioned spatial and temporal concepts. First, we discuss the concept of ontologies and present existing tools and languages supporting them. Then, we show the use of ontologies in LBS, by, initially, using them to represent Domain Data.

About Ontologies. An ontology is an explicit specification of conceptualisation [17]. Ontologies have been long-praised for their efficient use in the comprehension, representation, exchange, share, and integration of domains and concepts [18] [14] [39] [41]. They have been widely used in the past years to describe in an abstract, but accurate way, concepts shared and exchanged among different users, systems, or even people using oral communications. While in the philosophical fields an ontology is the science of being, in the Artificial Intelligence area it is used to describe an engineering, formally defined artifact with specific vocabulary using a set of assumptions regarding the intended meaning of the vocabulary words. Using ontologies to build applications can help avoid problems, such as inconsistency and poor understanding among communicating parties.

The Artificial Intelligence literature contains many definitions of ontology. Many of them contradict each other. Generally speaking, in the engineering world, an ontology is a formal and declarative representation which includes the names for referring to the terms in that subject area and the logical statements that describe what the terms are, how they are related to each other, and how they can or cannot be related to each other. Ontologies therefore provide a vocabulary for representing and communicating knowledge about some topic and a set of relationships that hold among the terms in that vocabulary.

In practical terms, the design of an ontology includes:

- the definition of classes or concepts in the ontology
- the arrangement of the classes in a taxonomic (subclass-superclass) hierarchy, if it exists
- the definition of properties and the description of the allowed values for these properties
- the definition of restrictions on the values of the properties, such as cardinality

An ontology, together with a set of individual instances of classes with specific values of properties, constitutes the knowledge base of the application.

The line between ontologies and conceptual schemas is thin. One could argue that the process of creating ontologies is conceptual modeling. Another approach is that using conceptual models to represent a domain of the application is adequate; proposals like that do exist [16] [41]. However, besides the fact that ontologies offer more flexibility in information representation, there are differences between conceptual schemas and ontologies:

- at the schematic level, an ontology is usually a forest of diagrams, while a conceptual schema –based on the strict literature definitions– is not, and
- ontologies are used to exchange and share common information (for example, the 'location', 'position' and 'time') among applications belonging to the same domain (for example, fleet management, mobile services etc., are all LBS), while conceptual schemas are used to model data in one application.

Some proposals about ontology definition include also the definition of rules to add semantics and to infer knowledge. Rules represent implicit knowledge about classes and their relationships. If one adopts this ontological approach, then this is one more difference between ontologies and conceptual schemas as rules exist only in ontologies.

One way is to see ontologies as an abstraction of conceptual schemas. Overall, ontologies are semantically richer than the conceptual schemas as they are built for different purposes: the former to represent a domain in a knowledge base, and the latter to represent contents of a database.

There are several ontology languages and tools, which are used to build ontologies. The most popular of them are compared in [15]. DAML+OIL [7] is the standard ontology language and close to the standards developed by W3C [45]. Chimaera [6], Ontolingua [28] OntoBuilder [27] and Protégé [38] are some of the most known tools as ontology editing environments.

In literature, there are also proposals for the structure of ontological environments. A representative one is [20] which structures an ontology to sub-ontologies: (a) the upper ontology, which includes abstract and philosophical issues, (b) the domain ontology which includes specific domains, such as tourism, weather, (c) the task ontology which contains knowledge about the usage, and (d) the application ontology which combines and extends the knowledge of all other ontologies. Depending on the application domain, several ontologies can be identified at the levels listed above.

Ontologies in LBS. As the need for capturing more semantics in LBS is growing together with the demand of structured information and services, domain experts started using ontologies in location-based applications. The design of ontologies is a modular task, i.e., it is important to define their structure and their interconnections, starting from the global or more dominant ones and then the more specialized ones, creating in this way, a structure, or an architecture. Moreover, more and more libraries of ontologies *do* exist today, such as the DAML ontology library (available at www.daml.org) or the Ontobuilder [27] (available at http://ie.technion.ac.il/OntoBuilder). This gives the expert the ability to acquire ontologies from different environments; however, it is crucial for integrity reasons to categorize them at the right level of the ontology architecture.

Some works follow specific architectural proposals for LBS. [44] follows the architecture presented in [20] to present an ontology list for semantic GeoServices for

Olympia 2008. [46] presents a different architecture to share ontologies in LBS but also keep their autonomy. The elements in this architecture are: (a) the global ontology, (b) the local ontologies, which correspond to local sources, (c) the shared ontologies, (d) the mediator and (e) the integrated ontology.

In our work, we propose an LBS architecture whose structural components follow this rationale and support the data categories of Figure 1 of Section 3. Thus, we propose the design of the:

- Domain Ontology, thus consisting of the Space Ontology and Time Ontology,
- Content Ontology, and
- Application Ontology, consisting of the Profile Ontology and Service Ontology.

This ontology architecture (a) keeps the Domain Ontology independent of other semantics and characteristics, and thus it can easily be exchanged and shared among LBS, according to Section 3, and (b) respects the, well-documented in literature, separation between applications (i.e., services and profiles) and context (i.e., information). The following sections describe the aforementioned ontologies.

4.3 The Domain Ontology of LBS

After the systematic analysis, clarification and relation of the fundamental concepts of the Domain Data of LBS, we proceed on the design of the Domain Ontology. We use Protégé[2] for the design and, further, the full development of the Domain Ontology of both the tourist and the traffic LBS.

Protégé [38] has (a) a graphical and easy-to-use interface, (b) a flexible knowledge model, and (c) an extensible plug-in architecture. With respect to Section 4.2, it includes:

- classes, which are the modeled concepts
- slots, which represent first-class objects representing properties or attributes of classes. A slot can be of an atomic type (e.g., float, integer, etc) or if an instance type, which means that it is an instance of another class.
- facets, which are constraints on allowed slot values, such as cardinality, defaults, allowed classes and others.
- axioms, which specify additional constraints

The distinction between classes and instances is not an absolute one. Both individuals and classes themselves can be instances of classes [38]. The main advantages of Protégé are that:

- It is easy and understandable enough for the domain expert to use it to develop the ontologies of his interest.
- It is an adaptable tool, which we can tune to support new languages and formalisms quickly. This is important as on the one hand, a number of new semantic-web languages and representation formalisms are emerging, but on the other, there is no agreement made yet.

---

[2] Protégé, as of Feb. 15, 2003, is available in version 2.0.

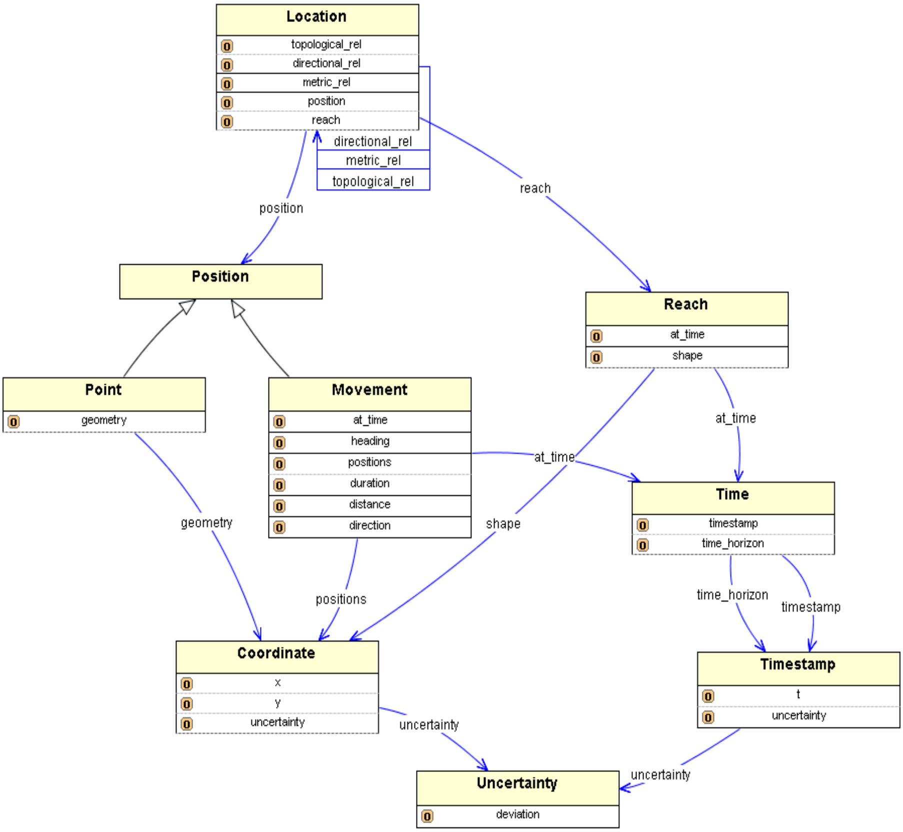


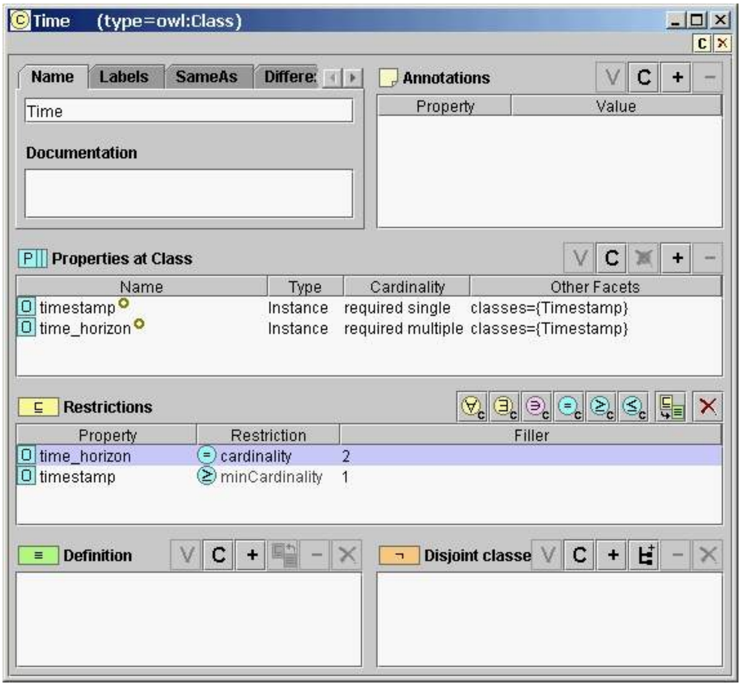
Fig. 4. Classes and relations among them in Protégé, capturing the Domain Ontology of LBS

- It can be used for the development and management of ontologies and applications today without waiting for standards.
- The supported model is an open and extensible one, allowing for plug-ins serving specific purposes.

The output of the design on Protégé can be expressed in widely used semantic web languages, such as RDF (Resource Description Framework), XML, Ontology Inference Layer (OIL), and JDBC which support the share and exchange of the designed data, in our case, the Domain Data.

However, we should make clear, that the use of Protégé is a prototypical one; any other tool or language with equal expressive power would do for this purpose. For this, we do not attempt to present specific implementation details that depend on the particularities of the ontology-editing tool, but rather use it as an illustration for the concepts we discuss.

Figure 4 illustrates the classes, instances, and slots among them, capturing the Domain Ontology of LBS in Protégé. In a class, when a slot is not of atomic type (e.g.,



Name: Time

Property	Value

Name	Type	Cardinality	Other Facets
timestamp	Instance	required single	classes={Timestamp}
time_horizon	Instance	required multiple	classes={Timestamp}

Property	Restriction	Filler
time_horizon	= cardinality 2	
timestamp	≥ minCardinality 1	

Fig. 5. Defining the class Time of LBS, in Protégé

float, integer, etc), but it is an instance of another class then it is depicted with an arrow. In order to explain Figure 4, for reasons of semantic simplicity and comprehension, whenever a slot of a class A is of type instance of class B, we say it that the class A is 'related to' class B.

Location is related to *Position* and *Reach*, while it has *topological_rel*, *metric_rel* and *directional_rel* relations with other *Location*s.

Movement and *Point* are subclasses of *Position*. *Movement* has as slots *heading*, *duration*, *distance* and *direction*, which are of atomic type and *positions* and *at_time* which are instances of classes *Time* and *Coordinate*, and thus depicted as relations.

Point has as slot *geometry,* which is an instance of class *Coordinate* with slots *x* and *y*, capturing the coordinates of the moving point object and *uncertainty*, which is an instance of *Uncertainty* class capturing the *deviation* from the true value.

Reach has as slots *at_time* and *shape*, which are instances of *Time* and *Coordinate,* respectively.

Time has as slots *time-horizon* and *timestamp*. *Timestamp* has as slots *t*, which is of atomic type, and *uncertainty*, which is an instance of *Uncertainty*.

Figure 4 translates to forms, such as the one presented in Figure 5. Figure 5 illustrates the definition of the class Time. The timestamp and time_horizon are instances of Timestamp and their cardinality is defined. time_horizon has cardinality of 2 as it needs two timestamps to be defined. As seen, Figure 5 allows for the definition of details, specifications and restrictions, such as disjoint classes, documentation and others.

The Domain Ontology (cf. Figure 4) of LBS includes:

- the Space Ontology, including the classes of *Location*, *Position*, *Point*, *Reach*, *Coordinate*, *Uncertainty* and *Movement*
- the Time Ontology, including the classes of *Time*, *Timestamp*, *Uncertainty* and *Movement.*

Protégé has a plug-in to import ontologies from other ontology editing environments, as for example, Ontolingua [30] or DAML [7]. The Time Ontology for example is available in the Ontolingua ontology library [29], as Simple-Time, including time-points and time-ranges and following the Allen's time theory [1]. Thus, one would argue, about how useful is to develop new ontology and not importing existing ones from available servers; this approach has been used in other application domains [36]. The Time Ontology we define here includes the movement class and thus is tailored to the needs of LBS. For reasons of integrity the Allen's relation should be included to relate the Time instances (as it happens with the Space instances). However, since this is trivial, we do not present it.

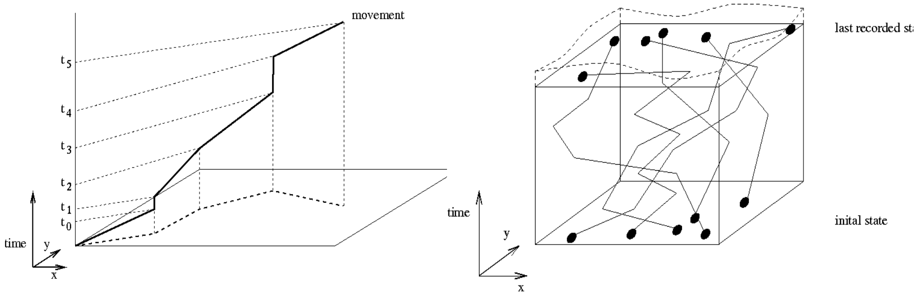
Similarly, the Space Ontology is also available in existing libraries. Again, just its adoption and import to the LBS knowledge base is not enough as the specific slots and instances we design and use are explicit for LBS and absolutely necessary to capture the very specific semantics of LBS.

5 The Content Data and Ontology of LBS

Following Figure 1, the next step is the analysis of the semantics of Content Data and its representation. We present excerpts of the Content Data from both the traffic and the tourist LBS. We chose two different approaches in capturing Content Data for the two LBS:

- for the traffic LBS we chose to use the UML technique in order to organize the huge amount of traffic data. This results to the Moving Object Database (MOD), which includes trajectories, vehicles and routes in UML, which was chosen due it is popularity, high-degree of comprehension and expressiveness. MOD is the core of a traffic management system on which, in many application environments [24], data mining functions are applied to extract information about traffic prediction. The reader can find more details about MOD in [5].
- for the tourist LBS we stayed focused on the use of ontologies, resulting to the Content Ontology.

By using different techniques to capture the semantics of the Content Data, we show the diversity of LBS Content Data and consequently the different semantics and design needs. Furthermore, this builds on the fact that our design choices are not bounded to the use of specific technology and tool.



(a) (b)

Fig. 6. Moving point objects: (a) a trajectory and (b) several trajectories in evolving in a finite region

5.1 Content Data of a Traffic LBS

The organization of traffic Content Data in the Moving Object Database (MOD), calls for a further, more detailed and in-depth understanding of objects, their properties and relations[3] related to the concept of movement.

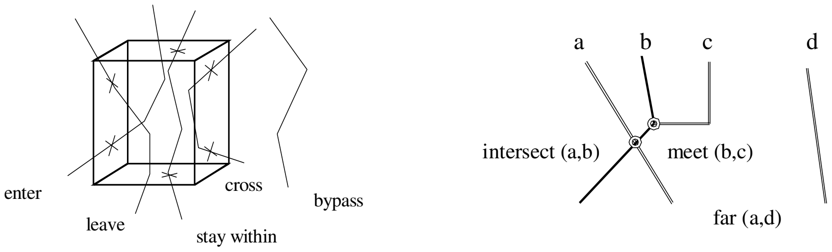
Movement in a Traffic LBS. Consider the following scenario using a traffic management system to monitor the traffic flow in its city area of Athens, Greece. By monitoring the movement of specific vehicles (e.g., delivery trucks, public transport, taxis, etc.) one can ask the following queries: ‘find the vehicles that just entered Athens’, or ‘find the vehicles that left Athens an hour ago,’ or more general ‘find locations with a larger number of vehicles’ (i.e., typical traffic jam pre-condition). Representing such moving objects as point objects their movement can be illustrated as shown in Figure 6. The solid line in Figure 6(a) represents the movement of a point object. Space (x- and y-axes) and time (t-axis) are combined to form a 3D-area. The dashed line shows the projection of the movement in two-dimensional space (x and y coordinates).

In order to record the movement of a vehicle, we need its position at all times, i.e., on a continuous basis. However, GPS and telecommunications technologies only allow us to sample an object's position, i.e., to obtain the position at discrete instances of time such as every few seconds. By, later on, interpolating these samples, we can extract the movement of the object. The simplest approach is to use linear interpolation, as opposed to other methods such as polynomial splines [2]. The sampled positions then become the end points of line segments of polylines, and the movement of an object is represented by an entire polyline in three-dimensional space. In geometrical terms, the movement of an object is termed a *trajectory*; in other words, trajectory is the trace of the vehicle in time.

Figure 6(b) shows a spatiotemporal space (the cube in solid lines) and several trajectories (the solid lines) contained in it. Time moves in the upward direction, and the top of the cube is the time of the most recent position sample. The wavy-dotted lines on top symbolize the growth of the cube with time.

---

[3] In the classical database meaning.



(a) (b)

Fig. 7. Relationships: (a) trajectory/spatial environment and (b) trajectory/trajectory

The trajectory representation is adequate to derive certain properties and relations of the object's movement:

Properties in MOD. Trajectories are characterized by a set of different properties depending on the application requirements. Some of the most common properties are:

- the speed of the movement (indicated by the inclination of the trajectory)
- the heading of the vehicle,
- the covered area, indicating the area the vehicle covered during its trip,
- the traveled distance, and
- the traveled time.

Based on our studies [32] [34], the aforementioned representation is adequate for mobile database modeling, since it gives answers to simple questions, such as 'which area did vehicle A-4592 cover during its trip?' and to more complex ones, like 'which vehicles left Athens after midnight moving East and were found close to each other 2 hours later?'.

Relations in MOD. Through their movement, trajectories relate to their environment in different ways over time. In the following, we discuss to types of relationships, namely how a trajectory can relate to its (spatial) environment and to other trajectories.

Relations between a trajectory and its spatial environment. Trajectories can have relations with the spatial environment. which includes other spatial objects. These can be either infrastructure elements, such as roads, parks, buildings, etc. but also imaginary entities such as city boundaries or query regions. In the temporal context these spatial entities become three-dimensional (i.e., space and time dimensions) represented by e.g., a 3D region. We distinguish five basic relationships (Figure 7(a)), but others can also be included:

- *stay within*, when the trajectory is all the time in the range of interest,
- *bypass*, when the trajectory passes by the range of interest,
- *leave*, when the trajectory leaves the range of interest,
- *enter*, when the trajectory enters the range of interest,
- *cross*, when the trajectory crosses the range of interest.

Relations among trajectories: Additionally, relevant positions among trajectories need to be registered at time points. The most common ones based on topological reasoning [10] are the following (Figure 7(b) depicts four of them):

- *intersect*, indicating that two trajectories intersect,
- *meet*, showing that two trajectories touch at one point
- *equal*, when two trajectories coincide,
- *near*, when two trajectories are close to each other, based on definitions on what ‘close’ means
- *far*, when two trajectories are away from each other.

Note that the concepts of far/near are context sensitive and thus depend on the application domain. For example, what is ‘near’ for two airplanes is rather ‘far’ for two cars and even farther for two pedestrians.

Having defined the above, one can ask for trajectory(-ies) fulfilling one or more conditions; from the simple ‘which area did vehicle X cover during its trip?’ to the more complex ‘which vehicles left this area after midnight moving East and were found close to each other 2 hours later?’

Some of the aforementioned properties and relations have been also presented in Section 4.1 to capture the semantics of movement and location. Here, for the needs of MOD, we present them in more detail. Additionally, for both types of relations there exists a substantial amount of work in literature with respect to the way of how two real world objects are topologically associated. In this work, we just include the fundamental ones.

Organizing MOD for a Traffic LBS. The various concepts relating to trajectories presented in the previous section are organized to define the underlying data model of a MOD. Following the well-known methodology of a database design, including the phases of conceptual modeling, logical modeling and implementation, we initially use conceptual modeling to capture the semantics of the aforementioned concepts in an organized manner. For the conceptual representation, we use the class diagram of UML [4] due its popularity, high-degree of comprehension and expressiveness.

Figure 8 illustrates the conceptual schema of MOD and exhibits five major classes, namely, trajectory, 3D-region, vehicle, road, and road segment and two relations which are modeled as object classes: the relation among trajectories (‘trajectory/trajectory’) and the one between trajectory and 3D-region (‘trajectory/environment’). Due to the fact that movement, changes continuously other properties of the objects involved in the database, such as the speed of the vehicle, the heading of the vehicle, and relations among them, such as far (i.e., the two vehicles are far’), or near (i.e., the two vehicles are near’) it is essential to capture functions or operations on objects. For example, ‘GetSpeed’ shows the speed of the vehicle at a given time point, or ‘Far’ gives a boolean answer about whether or not two vehicles are far from each other. An operation is a service applied on an object. The UML class diagram proved to be expressive enough to capture all the aforementioned elements and semantics.

To capture a ‘trajectory’, we need an identification of the mobile device (indicated by ‘object id’), the actual trajectory (‘trajectory id’) as well as the position of the

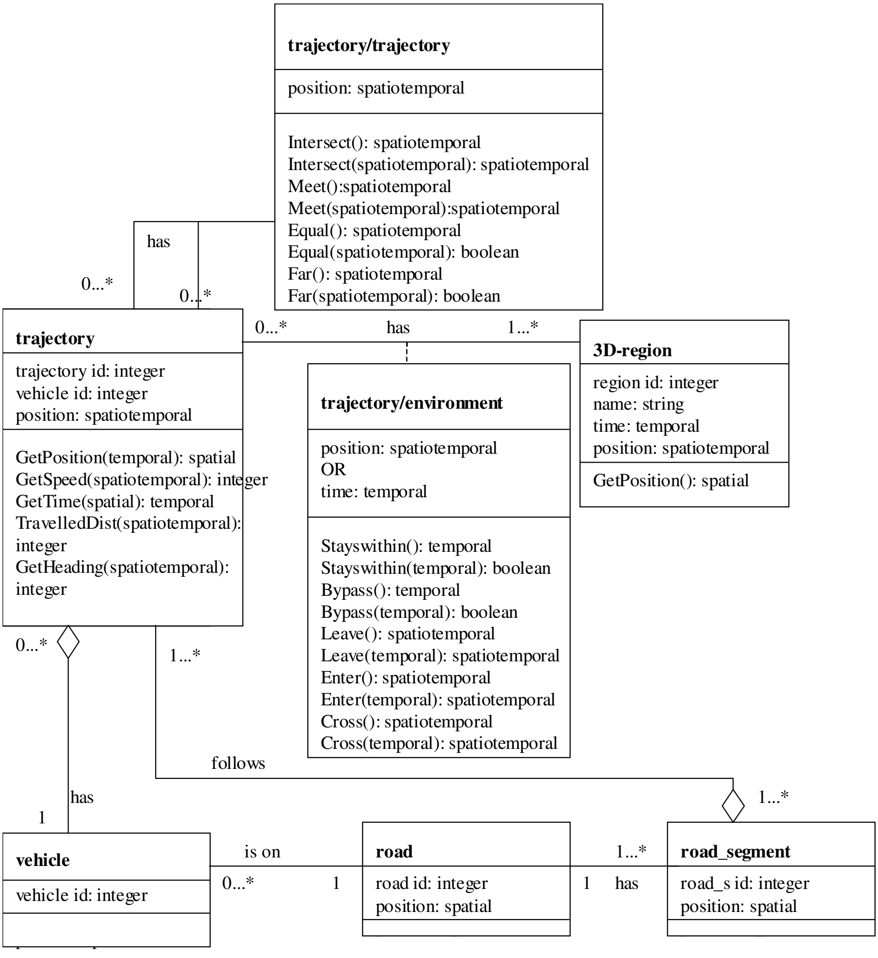


Fig. 8. An excerpt of the database schema of MOD

trajectory itself. In other words, ‘position’ describes the trace of the moving vehicle. The data types used are abstract, since they only should indicate the dimensionality of the parameter. More concrete instances of data types can be found in, e.g., [21]. A set of operations, e.g., GetSpeed(spatiotemporal), GetTime(spatial), and TravelledDistance(spatiotemporal), GetHeading(spatiotemporal) are prototypical and show what type of information can be derived from the trajectory data, e.g., to compute the traveled distance or the heading of a trajectory, we apply an operation that uses a spatiotemporal range as a parameter.

The ‘3D-region’ class is prototypical to denote the spatial environment of the trajectory (as part of a 3D-region representing the 2D-space and the time dimension (cf. Figure 1(b))). It is a fundamental object class of MOD. As stated previously, the 3D-region can be built up as time progresses and the objects move; in this case it shows the total covered area.

Trajectories ‘have’ (one or more) relations either with other trajectories, or their 3D-region class. Figure 8 contains the respective classes functions to compute such relationships. E.g., ‘Leave’ without parameter computes the spatiotemporal positions at which a trajectory left a given instance of a 3D-region class. To restrict the operation, we can use an argument to the function. In the case of Leave it is a temporal argument, i.e., the search for spatiotemporal positions at which the trajectory has left the region is restricted to a given time interval. In the class ‘trajectory/environment’ the parameter ‘position’ or ‘time’ capture the result of the function. Equally, so does ‘position’ in relation ‘trajectory/trajectory’.

Finally, for reasons of integrity, we capture the obvious object classes ‘vehicle’, ‘road’ and ‘road segment’. Note that the relation between vehicle and trajectory is an aggregation, as one vehicle can appear and disappear (due to loss of the GPS signal) and thus its route is a combination of trajectories. The same happens between ‘road’ and ‘road_segment’.

Figure 8 depicts, as a prototypical example, only the basic classes; other classes, for example 3D-lines (e.g., road-networks in time) that exhibit different relations with moving objects (e.g., moving along, etc.) can also be accommodated in this approach.

The rationale and choices presented here have the main advantage of describing two basic concepts: (a) *the trajectory* of the moving object by keeping track of its movement, and (b) the moving object itself, by recording its last known position. The spatiotemporal framework in which the movement takes place can either be built on the fly (i.e., while objects move) or be pre-defined (e.g., Athens in a specific time interval).

5.2 The Content Ontology of a Tourist LBS

Tourist content data includes information relating to entertainment, museums, history, etc. These Content Data can be structured in the form of an ontology/taxonomy grouping the content into a hierarchical set of categories of data.

Figure 9 shows an excerpt of a Content Ontology in Protégé showing related classes in a tourist LBS. The arrow indicates a superclass-class relation; e.g., History is the superclass of Historic_Event and Historic_Site.

Such a taxonomy can be seen as a general means to structure content data. For example, for the tourist LBS, given the fact of the historic battle of Marathon, which happened in the year 490BC, this information can be categorized under the ‘Historic_Event’ class. Figure 9 illustrates this by having the instance ‘Battle_of_Marathon’ connected to the respective class by a dashed arrow. Another example is the historic site ‘Acropolis’ categorized under to the ‘Historic_Site’ class.

The organization of Figure 9 exhibits similarities with other existing taxonomies existing, such as the dmoz.org open directory [9].

Content is related to spatial information in terms of the position of the facilities it describes. For example, the content of a tourist LBS includes the positions of all restaurants in Athens. This spatial data is not related to the Space Ontology (Section 4.3), as the later describes the whereabouts of the moving object.

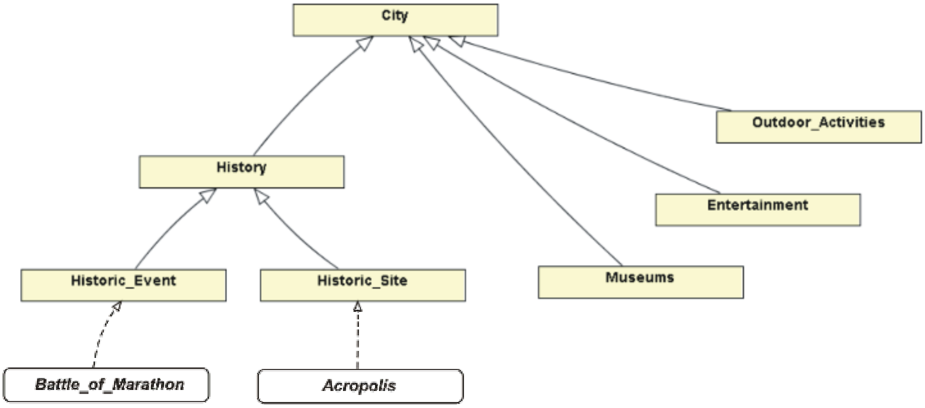


Fig. 9. An excerpt of the Content Ontology for a tourist LBS

6 The Application Ontology of LBS

Following Figure 1, the next step is the analysis and the modeling of the Application Data. Application Data (cf. Figure 1) refers to profiles and services, related to specific applications, in our case, the traffic and tourist LBS. In this section we discuss the types of user profile data and services and possible ways to represent them in the Application Ontology (cf. Section 4.2). We do not deal with the device profile (cf. Section 3.2), as its data is governed by the characteristics of the particular device and this is outside the scope of this work.

6.1 The Profile Ontology

Users do have preferences with respect to what information they usually request, and considering mobility, as to when and to where they do this. Recording these data leads to creating a *user profile*. It represents the choices and the needs of each individual user so that (a) the mobile device behaves in a way desired by the user and (b) information of interest is *forwarded* to the user in both *synchronous* (pull) and *asynchronous* (push) modes. In both cases the *position* of the user and the *time* are essential features and are taken into account. For example, in the tourist LBS, the user profile changes depending on the position of the user (e.g., ‘when I am in Berlin, my profile is *business*, when in Bahamas, my profile is *tourism*) or even on the time (e.g., ‘after 8pm receive only information about entertainment’).

The user profile can be: (a) explicitly defined by the user and (b) implicitly be modified by a data mining module that takes the demographic data of the user and his/her behavior patterns into account, where behavior patterns can be categorized into (i) spatiotemporal behavior (i.e., the user motion patterns in space through time) and (ii) previous choices that the user has made regarding information access.

Figure 10 gives an example of a simple, explicitly defined User Profile Ontology, for the traffic and the tourist LBS, that structures the interested of a ‘User’ based on

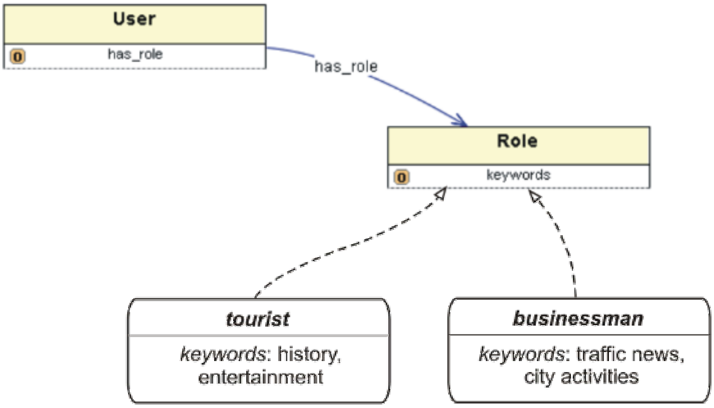


Fig. 10. An excerpt of the user profile of the Application Ontology for the tourist LBS

the concept of a 'Role.' Roles are aggregations of profile specifications with the interests specified as keywords for each role. E.g., if a user "activates" his 'tourist' role, he wants to be notified of services relating to history and entertainment (cf. Figure 10). His 'businessman' role states his interest in traffic news and city activities. As we will see later on in Section 6.3, this explicit specification of interests can be used for automatic service discovery.

6.2 The Service Ontology

Since services rely on data, relating to the content ontology of Figure 9, a similar ontology can be derived to structure services in relation to the data they provide. Services have the spatial dimension, in the sense that they are structured analogously to the *Site*[4] they refer to. For example, for the tourist LBS, the Entertainment, Museums, History, and Outdoor Activities, which are all services, are related to specific sites. The same holds for a traffic LBS, in which the Traffic, i.e., Traffic_Jam and Traffic_Load always refer to specific sites.

Figure 11 illustrates an excerpt of the Service Ontology of the traffic and the tourist LBS.

Site can be a point or an area providing specific services, e.g., facilities in the area.. A service ontology is used to discover services based on a request. A request is specified in terms of the spatial parameter, i.e., *Site*, and additional descriptive information such as keywords. Using the service ontology, all services will be structured according to their respective spatial scope, i.e., their *Site*, and the semantic category they belong to. Matching a request to an actual service is done by matching the *location* (of the user*)* and the keywords characterizing the request to the respective *Site* a service covers and the specific category a service belongs to, respectively. Matching keywords onto categories can be done by, e.g., measuring the word distance between the set of keywords and the matching category descriptions in the taxonomy [43][40].

---

[4] We use the term *Site* to avoid confusion with *location* and *position*, which are reserved words in this work.

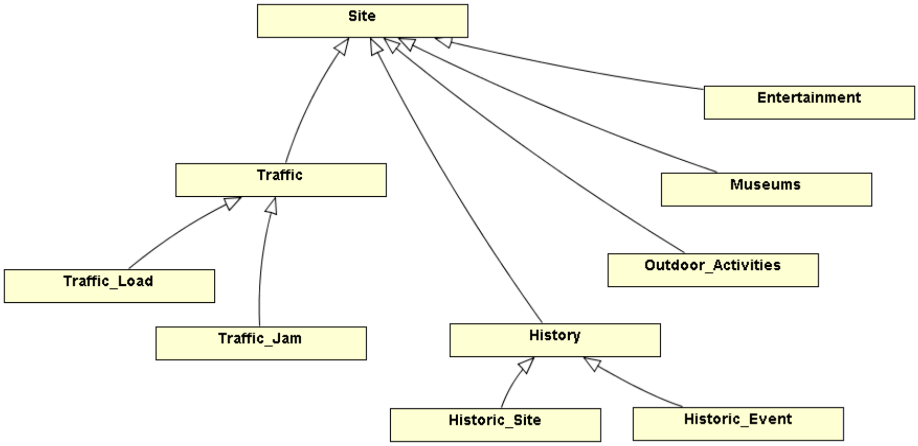


Fig. 11. A simple Service Ontology for a traffic and tourist LBS

Consider the scenario from the tourist LBS in which a service providing historical information relating to Acropolis, by using the classes of Figure 11, is categorized under the 'Historic_Site' class. The typical example 'give me the history of the place I am' (cf. Section 3) relates, spatially, the *location* (in this case *position*) of the tourist with the area he is in.

Analogously, the example from the traffic LBS, 'based on my position, where is a traffic jam in 5km ahead', relates, spatially, the location (cf. Figure 2b) to the area of a traffic jam.

The spatial relation between *Site* and *location* is achieved by using well-known spatial relationships [10].

6.3 Automatic Service Matching

In this section we discuss some ways services are provided by using concepts from the Domain Ontology, and the Application Ontology.

Besides searching for services based on explicit requests (pull), services can be triggered implicitly (push) by matching a user profile onto service descriptions by using agents [31]. Assuming, in a tourist LBS, an extended user profile (cf. Section 6.1) contains information about preferences a person has when she is traveling as a tourist, e.g., history and entertainment (cf. Figure 10). By traveling to a new destination at some point his 'tourist' profile will be matched onto available service description and, e.g., a service related to museums information will be presented to him. On the other hand, when he is on the job, which is that of a traveling salesman by car, his profile specifies that he is interested in traffic-related services. In this case, reaching a new destination, traffic-related services will be presented to him.

Events (other than profile declaration) trigger this service discovery. *Position* and *location* play a central role. It acts as a trigger to send related information to the user. For example, when a tourist interested in history reaches the Athens city centre, a service is activated that presents information about the Athens archaeological museum.

Another factor that can act as a trigger for service discovery and activation is the *history* of the user, which can be registered in his profile. For example, in the tourist LBS, considering a tourist who frequently visits museums. Then, even if he has not defined explicitly 'history' or 'museums' as a preference in his profile, when coming to Athens he still will be presented services that inform him about the Athens archaeological museum. A detailed discussion about the role and use of events in LBS can be found in [22].

7 Conclusions

In this work, we analyze, comprehend and model data semantics of LBS. The analysis of data leads to the *Domain*, *Content* and *Application Data* categories depending on the related concepts and their use. To model these data categories we adopt the semantically rich UML as well as the long-praised method of ontologies, depending on the application needs and the complexity of semantics.

Modeling the semantics of the three data categories leads to the creation of three different ontologies: the Domain Ontology, the Content Ontology and the Application Ontology. This structure, illustrated in Figure 12, serves as the backbone architecture to support LBS based on ontologies, with special focus on autonomy and share.

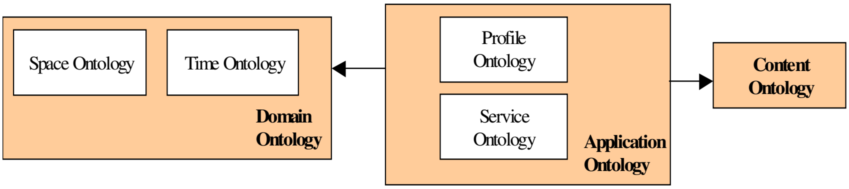


Fig. 12. The Ontology structure of LBS

The three data categories are interconnected; the Application Ontology is connected to both the Domain Ontology and the Content Ontology, since the Service Ontology relates to the Content and to the Space and the Time Ontology, and the Profile Ontology relates to the Space and the Time Ontology. For example, in order to provide the service 'closest restaurant' a reference to the restaurants index is needed and to the position of the user. There is no association between Content and Domain Ontology. The Content refers to specific geographic information (for example, location of restaurants in Athens), but this is general spatial data, outside the Domain Ontology, which refers to the *where* and *when* the user is.

A major contribution of this work, is the creation of ontologies (i.e., Domain, Content and Application) for LBS, to represent, share and exchange the concepts of location, position, movement and time among location-based applications.

Another important strength of this work is the clear distinction between *location* and *position*, which solves ambiguities and makes assumptions about these two concepts clear. Distinguish *location* from *position* further helps on the accurate semantic modeling and the representation, exchange and integration of location from multiple sources.

Finally, the applicability and feasibility of our design choices are shown with examples from two real case studies, the tourist and the traffic LBS for the Athens Metropolitan Area.

Acknowledgements

This work is supported by the *DB-Globe: A Data-centric Approach to Global Computing* project, funded by the European Commission under contract number IST-2001-32645, and the *ΙΧΝΗΛΑΤΗΣ: A Traffic Management System* project, funded by the Hellenic General Secretariat of Research and Technology. The authors wish to thank EMFASIS Telematics, Greece, for providing real data for the case studies.

References

1. Allen, J.: Towards a General Theory of Action and Time. Artificial Intelligence 23 (1984) 123-154.
2. Bartels, R.H., Beatty, J.C., and Barsky, B.A.: An Introduction to Splines for Use in Computer Graphics & Geometric Modeling. Morgan Kaufmann Publishers, Inc. (1987).
3. Bittner, T., and Winter, S.: An Ontology in Image Analysis. In Proc. of Integrated Spatial Databases: Digital Images and GIS (1999) 168-191.
4. Booch, G., Rumbaugh, J., and Cobson, I.: The Unified Modeling Language User Guide. Addison-Wesley (1999).
5. Brakatsoulas, S., Pfoser, D., and Tryfona, N.: Modeling, Storing and Mining Moving Objects Databases. In Proc. of the International Database Engineering and Applications Symposium (IDEAS) (2004) 68-77.
6. Chimaera Ontology Environment, Web page, www.ksl.stanford.edu/software.chimaera (2004).
7. DAML.Org. Reference Description of the DAML+OIL ontology markup language. Web page http://www.daml.org/2001/03/reference (2004).
8. DBGlobe: A Data-centric Approach to Global Computing. IST Project, Project number IST-2001-32645.
9. DMOZ.org: Open Directory Project. Web page http://www.dmoz.org (2004).
10. Egenhofer, M.: Reasoning about Binary Topological Relations. In Proc. of the 2nd Symposium on Spatial Databases (SSD) (1991) 143-160.
11. Fonseca, F.: Ontology-driven Geographic Information Systems. In Proc. of the 7th ACM Symposium on Advances in Geographic Information Systems (1999) 14-19.
12. Fonseca, F., Davis, C., and Camara, G.: Bridging Ontologies and Conceptual Schemas in Geographic Information Systems. GeoInformatica 7(4) (2003) 355-378.
13. Fonseca F.T., Egenhofer, M.J., and Agouris, P.: Using Ontologies for Integrated Geographic Information Systems. Transactions in GIS 6(3) (2002) 231-257.

14. Frank, A.: Spatial Ontology: A Geographical Point of View. Spatial and Temporal Reasoning, Kluwer, Dordrecht (1997) 135-153.
15. Garshol, L.M.: Topic Maps, RDF, DAML, OIL – A comparison. Web page http://www.ontopia.net/topicmaps/materials/tmrdfoildaml.html (2001).
16. Granefield, S., and Purvis, M.: UML as an Ontology Modeling Language. In Proc. of the Workshop on Intelligent Information Integration, 16th International Joint Conference on Artificial Intelligence (IJCAI) (1999) 46-53.
17. Gruber, Th.: A Translation Approach to Portable Ontology Specifications. Knowledge Systems Laboratory – Stanford University, Stanford, CA, Technical Report KSL 92-71 (1992).
18. Gruber, Th.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Int. Journal of Human-Computer Studies, Vol. 43 (1993) 907-928.
19. GSM Alliance Services Working Group, Web page http://www.gsmworld.com/about/structure/serg.shtml (2004).
20. Guarino, N.: Formal Ontology and Information Systems. In Proc. of FOIS'98, Trento, Italy. Amsterdam IOS Press (1998) 3-15.
21. Guetting, R., Böhlen, M., Erwig, M., Jensen, C. S., Lorentzos, N., Schneider, M., and Vazirgiannis, M.: A Foundation for Representing and Querying Moving Objects. ACM Transactions on Database Systems 25(1) (2000) 1-42.
22. Hinze A., and Voisard A.: Location- and Time-based Information Delivery in Tourism. In Proc. 8th International Symposium in Spatial and Temporal Databases (SSTD) (2003) 489-507.
23. Hu, H., and Lee, D.: Semantic Location Modeling for Location Navigation in Mobile Environment. In Proc. Of the IEEE International Conference on Mobile Data Management (MDM) (2004) 52-61.
24. ΙΧΝΗΛΑΤΗΣ: A Traffic Management System. Research Academic Computer Technology Institute, 2002-2004. ENTER Project, Hellenic General Secretariat of Research and Development (2004).
25. Marmasse, N., and Schmandt, C.: A Location-Aware Information delivery with comMotion. In Proc. 2nd International Symposium on Handheld and Ubiquitous Computing (HUC) (2000) 157-171.
26. Myllykaki J. and Edlund S.: Location Aggregation from Multiple Sources. In Proc. of the 3rd International Conference on Mobile Data Management (MDM) (2002) 131-138.
27. OntoBuilder Project. Web page http://ie.technion.ac.il/OntoBuilder (2004).
28. Ontolingua: Ontolingua System Reference Manual. Web page http://www-ksl-svc.stanford.edu:5915/doc/frame-editor/index.html (2004).
29. Ontolingua: Ontolingua ontology library, Web page http://www.ksl.stanford.edu/software/ontolingua/ (2004).
30. Ontolingua: Ontolingua Server. Web page http://www-ksl-svc.stanford.edu:5915/ (2004).
31. Panayiotou, C., and Samaras, G.: Personalized Portals for the Wireless User: An Agent Approach. Journal of ACM/Baltzer Mobile Networking and Applications (MONET), special issue on "Mobile Commerce" (2004).
32. Pfoser, D., Jensen, C.J., and Theodoridis, Y.: Novel Approaches in Query processing for Moving Objects. In Proc. of the conference on Very Large Data Bases (VLDB) (2000) 395-406.
33. Pfoser D., Pitoura, E., and Tryfona N.: Metadata Modeling in a Global Computing Environment. In Proc. of the 10th ACM International Symposium on Advances in Geographic Information Systems (ACMGIS) (2002) 68-73.

34. Pfoser, D., and Theodoridis, Y.: Generating Semantics-Based Trajectories of Moving Objects. International Workshop on Emerging Technologies for Geo-Based Applications, Ascona, Switzerland (2000) 59-76.
35. Pfoser, D., Tryfona, N., and Verykios, V.: Services-based Data Management in a Global Computing Environment. In Proc. of the 3rd International Workshop on Web and Wireless Geographic Information Systems (W2GIS) (2003) 45-53.
36. Pinto, S., and Peralta, D.: Combining Ontology Engineering Subprocesses to Build a Time Ontology. In Proc. of the International Conference on Knowledge Capture (2003) 88-95.
37. Pitoura, E., Abiteboul, A., Pfoser, D., Samaras, G., and Vazirgiannis, M.: DBGlobe: a service-oriented P2P system for global computing. SIGMOD Record 32(3) (2003) 77-82.
38. Protégé Project. Web page http://protégé.stanford.edu (2004).
39. Smith, B., and Mark, D.: Ontology and Geographic Kinds. In Proc. of the International Symposium on Spatial Data Handling (SDH) (1998) 308-320.
40. Sycara, K. P., Widoff, S., Klusch, M., Jianguo Lu: LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. Autonomous Agents and Multi-Agent Systems 5(2) (2002) 173-203.
41. Tryfona N. and Pfoser D.: Designing Ontologies for Moving Objects Applications. In Proc. of the International Workshop on Complex Reasoning on Geographic Data, Paphos, Cyprus (2001).
42. Ubicomp Workshop: Location Modeling for Ubiquitous Computing. Workshop Proceedings, 2001.
43. Valavanis, E., Ververidis, C., Vazirgianis, M., Polyzos, G. and Norvag, K.: MobiShare: Sharing Context-Dependent Data & Services from Mobile Sources. In Proc. of the IEEE/WIC International Conference on Web Intelligence, (2003) 263-271.
44. Weissenberg N., and Gartmann R.: Ontology Architecture for Semantic GeoServices for Olympia 2008. In Proc. Münsteraner GI-Tage, Münster. IfGIprints 18 (2003) 267-283.
45. W3C: OWL Web Ontology Language Reference. Web page http://www.w3.org/TR/owl-ref/ (2004).
46. Yu S., Aufaure M., Cullot N., and Spaccapietra S.: Location-based Spatial Modeling Using Ontology. In Proc. of the 6th AGILE Conference (2003).
47. Yu S., Spaccapietra, S., Cullot, N., and Aufaure M.: User Profiles in Location-based Services: Make Humans More Nomadic and Personalised. In Proc. of the International Conference on Databases and Applications (2004).
48. Yu S., Aufaure M., Cullot N., and Spaccapietra S.: A Collaborative Framework for Location-based Services. In Proc. of the International Conference on Databases and Applications (2004).