Designing Ontologies for Moving Objects Applications[1]

Nectaria Tryfona and Dieter Pfoser

Department of Computer Science
Aalborg University, DK-9220 Aalborg, Denmark
{tryfona,pfoser}@cs.auc.dk

Abstract. The focus of this work is on the design of ontologies to support the exchange and sharing of information among applications dealing with moving objects. This application area ranges from fleet management systems to the monitoring of the users of mobile phones to provide location-based services. The use of ontologies for interoperability purposes has been long-praised; here, we focus on the comprehension and registration of the fundamental mobile ontologies to exchange information. In particular, to better understand and register the ontologies and their relations involved in this domain, we first analyze the important concept of *movement*, its properties and relationships within the spatiotemporal environment. Based on this, we further describe fundamental mobile ontologies; this description can act as a guideline for recording additional mobile ontologies depending on the needs of the application. We then show how mobile ontologies and moving objects interact. We present an architectural approach that serves as a means for application systems to share information based on mobile ontologies. Two different application scenarios–one with traffic management systems and another coming from the mobile communications area–using and exchanging ontologies illustrate the applicability of this approach.

1. Introduction

The need for the efficient use of applications dealing with *moving objects*, nowadays, becomes emerging. A traffic management system, which monitors the current traffic, records troublesome situations, such as traffic jams, and provides options, such as alternative routes in case of an accident, is a characteristic application in this area. Other examples range from fleet management systems (e.g., taxis) to the monitoring of the users of mobile phones in order to provide location-based services. In these application domains, a *moving object*, i.e., an object changing location in time, can be a moving vehicle, the user of a mobile, an army unit, or even a group of migrating birds.

An important issue in this area is *interoperability*, i.e., the need for exchanging data, information, and knowledge among different systems, to efficiently handle scenarios dealing with moving objects. Consider the scenario, in which the city of Athens uses a traffic management system to monitor the traffic flow in its city area. As some of the monitoring results are crucial traffic parameters for its neighboring areas (e.g., the number of vehicles entering city borders, the type of vehicles, speed, and direction) it would be desirable to communicate them to systems in the, topologically, adjacent areas. For this communication to take place successfully, an agreement on the fundamental concepts related to objects movements, their relations

---

[1] This work was done during the authors' sabbatical-leave at the Computer Technology Institute, Aktaiou 11 & Poulopoulou Str., Athens, Hellas.

and their properties needs to be made. It is our position that to exchange the semantics and behavior of moving (or mobile) objects it is essential to understand the basic mobile ontologies and their interrelations in the domain of a moving object application and register them. Sharing or understanding the same mobile ontology is a pre-condition to data and content sharing and integration. The concept of ontologies has been widely used in Artificial Intelligence in order to describe the semantic content, the participating objects, and their relationships [11].

This work is part of a larger effort [3,4] focusing on the definition of common characteristics in moving object applications and the definition and implementation of an ontology-based architecture that supports interoperability. To better understand the different domains of moving objects applications, work has been done so far separately on moving vehicles [15] and mobile communications [13]. Our experience from this theoretical research as well as real system needs [3] indicates that all applications concerning moving objects have some basic characteristics such as trajectories and their relations to space, which are fundamental in the exchange process.

Our scope here is to sketch the *requirements* and *semantics of mobile ontologies* and to propose an architecture based on which applications of moving objects can exchange information. More specifically, in this work:

(a) We present a systematic study, based on experiments and theoretical research, of the concepts involved in the movement of objects. We analyze the concept of movement, which is fundamental for these applications, show its characteristics and its relation with space.

(b) Based on (a), we describe the concepts and relations that constitute a mobile ontology. In order to easily realize the ontologies, we use the Unified Modeling Language (UML) [18]. However, the use of UML notation is just a prototypical one. Any other modeling or formal language (e.g., second order logic) can serve for representational purposes.

(c) We discuss implementation issues and the architecture of systems working with mobile ontologies and the way these systems can exchange information. We give different communication scenarios in order to illustrate the usability of this approach.

We are not aware of any other work towards this direction. However, the use of ontologies in spatial systems design and development has been long-praised [7,19]. Similar approaches deal with geographic ontologies for urban systems [9].

We should point out that it is not our intention to register all the ontologies of the moving world or to present a new language for registering them. We rather want to help comprehend this domain by realizing the concepts involved in this world and the way essential ontologies are combined and interact with each other and the underlying space. Putting aside the interoperability issue, the use of generic mobile ontologies instead of explicit moving objects for specialized domains (e.g., traffic systems or mobile network management) makes the representation generic enough to be adopted for the understanding of different scenarios (e.g., moving vehicles or users of mobile phones).

The rest of the paper is organized as follows. Sect. 2 presents movement and its properties as the fundamental concepts of an application domain dealing with moving objects. Based on these concepts, Sect. 3 discusses the essential mobile ontologies;

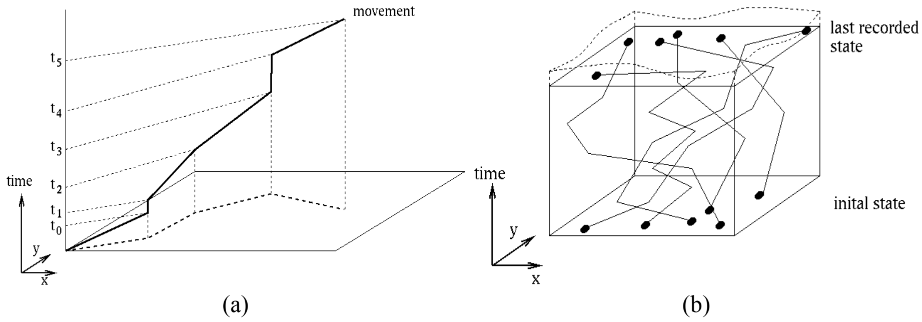

Fig. 1. Moving point objects: (a) a trajectory and (b) several trajectories in 3D (space and time) framework.

shows their relationships to moving objects, gives their description in UML, and gives different communication scenarios to illustrate the usability of this approach. Sect. 4 presents the architecture of a system for the management of moving objects based on mobile ontologies. Sect. 5 summarizes and points to future work.

2. Representation of Movement

To realize the ontologies (as well, as their properties and relations) involved in a mobile scenario, it is essential to understand and to study the fundamental concept of applications dealing with moving objects; that is the *movement of an object*, whether it is a moving vehicle or the user of a mobile phone.

Consider a database scenario in Athens, using a traffic management system to monitor the traffic flow in its city area. The movement of vehicles is monitored. Typical queries in this database would be: ‘find the vehicles that just entered Athens’, or ‘find the vehicles that left Athens an hour ago’, or more general ‘find the vehicles that come close to each other now’ (i.e., typical traffic jam pre-condition). In a scenario like this, moving objects are represented as point objects, since their volume or size does not play a critical role. The solid line in Fig. 1(a) represents the movement of such a point object. Space (x- and y-axes) and time (t-axis) are combined to form a 3D-framework. The dashed line shows the projection of the movement in two-dimensional space (x and y coordinates).

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 [1]. 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*[2]. Fig. 1(b) shows the spatiotemporal space (the cube in solid lines) and several trajectories (the solid lines). 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.

[2] For the rest of the paper, we use “movement” and “trajectory” interchangeably.

The approach presented in Fig. 1(b) can be used to depict two situations: (i) the covered area of all the current trajectories in time, or (ii) a pre-defined area in time, for example, the region of Athens in the time interval [10-12] in the morning.

Trajectories are characterized by a set of different properties depending on the application requirements. The most common properties are:

- the *speed* of the movement (or trajectory), showing how fast the object moves,
- the *heading*, showing the direction of the object,
- the *covered area*, indicating the area the object traveled,
- the *traveled distance*, and
- the *traveled time*.

Based on our studies [15,16], 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 ?'

3. Mobile Ontologies and Moving Objects

After describing a way to understand, represent and interpret the common parameter in all applications dealing with moving objects (that is the movement of objects) we study the mobile ontologies and their relation to moving objects, as well as, a way to capture and represent the semantics of mobile ontologies.

3.1 Mobile Ontologies

An ontology is an explicit specification of conceptualization [10]. Ontologies 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 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 misunderstanding among communicating parties.

In the engineering world, ontologies are built to serve multiple purposes (e.g., car behavior on a traffic system and mobile phone tracking) in a specific domain (e.g., domain of moving object applications), in contrast to object classes, which are built for concrete application domains. Defining fundamental mobile ontologies allows us to realize the 'mobile world' and its characteristics, where some tend to be common, and for this, easy to be exchanged and shared between different mobile applications. This approach is, on the one hand, generic enough to cover different types of moving objects, such as mobile phones, moving taxis, or even migrating groups of species, and on the other, specific enough to show the kind of data involved in mobile applications, and to ensure the consistency of communication among agreed partners.

The mobile ontological domain comprises the following elements.

Ontologies. Since trajectory has a central role in this domain, it is viewed as the fundamental ontology. Additionally, the space and time in which a mobile ontology is present are also essential. For this, we capture the 3D-spatiotemporal framework as ontology. The explicit representation of these ontologies is given in Sect. 3.2, after having fully defined the rest of the ontological domain.

Object classes. Object classes are connected, or affected by, or create ontologies. Thus, having the trajectory as ontology, typical object classes (creating a trajectory) are moving vehicles, mobile devices such as PDA or phones, people in motion, or groups of people in motion, such as an army unit. Analogously, having the 3D-framework as ontology, typical instances of object classes can be spatiotemporal regions, or cities in time periods, or even road networks in time. Fig. 2 associates ontologies, object classes and objects of a database.

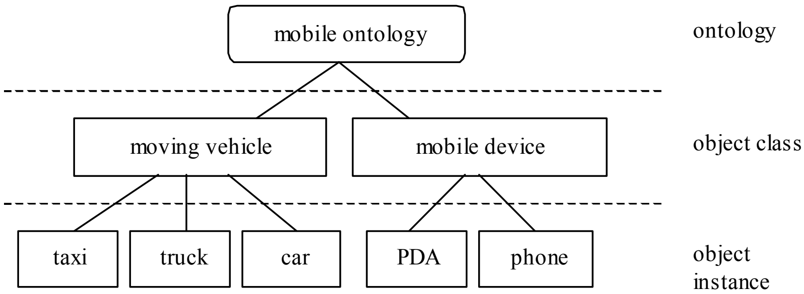

Fig. 2. Mobile ontologies, moving object classes and moving objects.

One could argue that Fig. 2 illustrates a hierarchy of object classes. This does not contradict the existing theories [10] commenting on the fact that ontologies are often equated to taxonomic hierarchies of classes, *but are not limited to this form*. For example, while in our case it is rather difficult to realize an application dealing with all types of mobile ontologies, it is obvious that moving objects or mobile devices 'create' trajectories, which are captured as ontology and have some common characteristics independent of the application domain they belong to. In order to materialize or specify a mobile ontology, definitions and axioms need to be taken into account at the object level. For example, what is a moving vehicle object, or a mobile device object based on very specific, non-common characteristics and behavior.

Relations among ontologies. It is important to describe the relations in which mobile ontologies participate. In our example, there are two types of relations:

- *Relations between a trajectory and the 3D framework*: Trajectories have relations with the 3D framework (i.e., space and time dimensions) as represented in Fig. 1(b). We distinguish five basic relationships (Fig. 3), 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 area of interest,
    - *leave*, when the trajectory leaves the area of interest,
    - *enter*, when the trajectory enters the area of interest,
    - *cross*, when the trajectory crosses the area of interest.

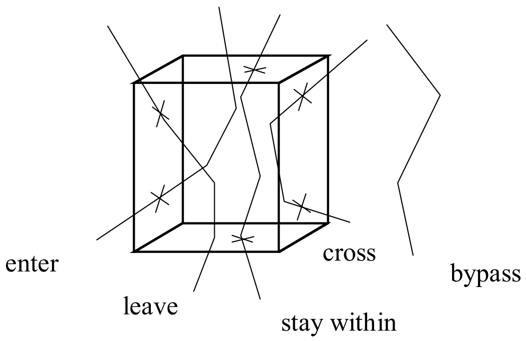

Fig. 3. Relations between trajectories and a 3D-spatiotemporal framework.

- *Relations among trajectories*: Additionally, relevant positions among trajectories need to be registered at time points. The most common ones based on topological reasoning [5] are the following (Fig. 4 depicts three 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.

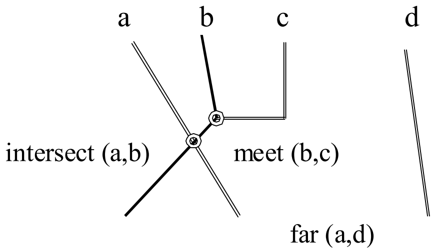

Fig. 4. Relations among trajectories.

For both types of relations, there exists a substantial amount of work in literature with respect to the way two real world objects are topologically associated. Here, we present the fundamental relations.

(a) *Events*. A spatial event is an action that takes place at a specific time point and space and it can be either scheduled or not. A mobile ontology can participate in events, such as traffic jams, accidents, or high traffic a rate of mobile phones.

(b) *Processes*. A process is an action that takes place in time, and goes on for an interval of time, i.e., it has a duration. It is used to describe on-going actions in which mobile ontologies participate, such as traffic flow, number of connections (i.e., phone calls), etc.

One could argue that the trajectory is an on-going process. We choose to view trajectory as ontology, and not as a process on another ontology, since its role in this application domain is central. Events and processes participating in mobile scenarios are triggered not only by ontologies and their relations, which are studied here, but also by special conditions (i.e., what causes a traffic jam, or, how a traffic jam is

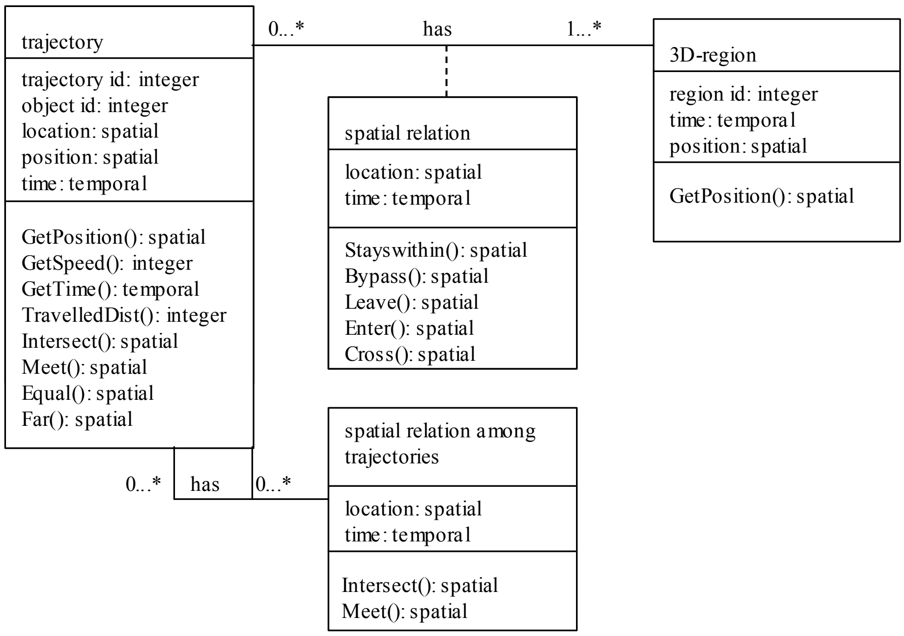

Fig. 5. Mobile ontologies and their relations in UML.

measured) and parameters. For this, their further exploration is outside the scope of this work.

3.2 Representing and Sharing Mobile Ontologies

In this section, we define and present mobile ontologies based on the characteristics and relations described above. The goal is to give a representation without ambiguities that can be shared and exchanged among users. To create this representation it was essential to understand the objects, relations, events, and processes mobile ontologies comprise, as discussed in Sect. 3.1.

In this work, we use the UML notation for this purpose. Many, different, formal languages (ONTOLIGUA [10]) and approaches (Ontology Inference Layer [8]) exist for this purpose. Our scope is not to show another definition language, but to illustrate the way fundamental mobile ontologies and object classes are combined under one framework based on their characteristics, behavior and interconnections in the mobile world. Thus, this representational approach is just a prototypical one; any other language would do for this purpose. Later on, in Sect. 4, we discuss how different languages can be employed.

Fig. 5 captures the mobile ontologies and their relations by following the rational analyzed in Sect. 2 and described in Fig. 2. This is by no means an exhaustive description of the ontologies of the mobile world.

To describe the ontology *trajectory,* we need an identification of the moving vehicle or the mobile device (indicated by 'object id'), the actual trajectory ('trajectory id'), the current position ('position') in space, as well as the location of the trajectory itself. In other words, 'location' describes the trace of the moving vehicle. Location and position are of type spatial [12]. A set of functions (i.e., GetPosition(), GetSpeed(), GetTime(), and TravelledDistance()) are prototypical to show what type of information needs to be retrieved and calculated in order for the ontology to be up-to-date. Finally, as stated in Sect. 3.1, a trajectory has relations with other ones; this is indicated by the Intersect(), Meet(), Equal() and Far() functions.

The *3D-region* (e.g., a 3D-framework representing the 2D-space and the time dimension (Fig. 2)) is a fundamental ontology of a 'mobile world'. As stated in Sect. 2, the 3D-region can be built up as time progresses and the objects move; in this case it shows the total covered area. Another option is that the 3D-region is pre-defined; for example Athens in the spatial dimension. The 'position' property indicated the actual position of the region, while the function GetPosition() indicates the currently applied region (which is meaningful only when the region is built up while the object moves).

Each trajectory 'has' (one or more) *spatial relation* with space and time. The 'location' indicates the spatial position where the relation is valid. Similarly, the trajectories have at points in time and space relations among each other.

Fig. 5 depicts, as a prototypical example, only trajectories and 3D-regions as ontologies; other spatial ontologies, for example 3D-lines (e.g., road-networks in time) that hold different relations with moving objects (e.g., moving along, etc.) can also be accommodated with this ontological approach.

The rationale and choices presented in Fig. 5 have the main advantage of describing two basic concepts: (a) the trajectory of the mobile ontology by keeping track of its trace, 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 the objects move) or be pre-defined (e.g., Athens in a specific time interval). In this way, we bridge the gap between the ontological world and the modeling requirements, i.e., the database modeling phase dealing with object classes.

To achieve sharing and exchange of information, especially, when the participant DBMSs are heterogeneous, it is important to express the ontologies in a highly abstract, semantically rich modeling language. Using the UML object class diagram allows us to better comprehend and agree on their characteristics and relations among them. Moreover, the exchange and sharing of the *current*, *updated* values of ontologies and objects (e.g., position, location, relations, etc), and their relations is supported.

Finally, we believe that using a schematic language to describe ontologies allows for a better realization of the choices and detection of mistakes, especially since considerable work has been done in successfully extending UML towards the accommodation of spatiotemporal needs [2, 17]. Having Fig. 5 as a starting point, a translation to any other language (e.g., ONTOLIGUA, XML) can be made. For example, the UML eXchange Format (UXF) for the translation of UML to XML is already available allowing UML to be highly interoperable [20].

3.3 The Role of Ontologies in Interoperation – Two Examples

In the next section, we describe two example scenarios based on experiments from the mobile world, to illustrate the information sharing. Fig. 6(a) shows the

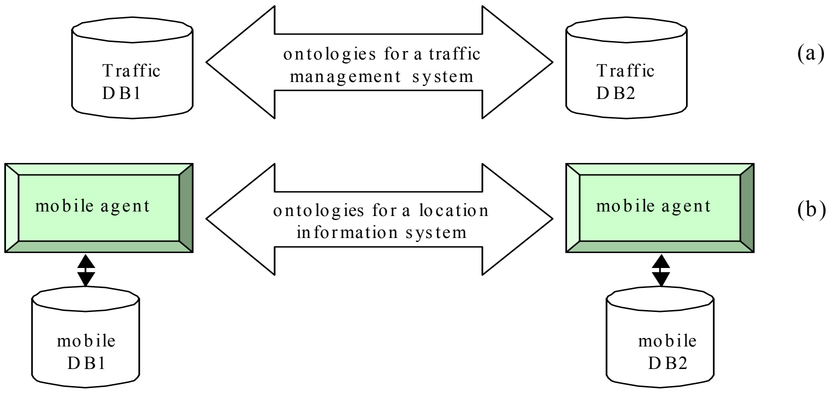

Fig. 6. Two scenarios of exchanging ontologies: (a) between traffic databases (b) between mobile agents.

communication between two traffic management systems, operating in two different areas such as two neighboring city areas, e.g., Athens and Piraeus in Greece. Besides the expected behavior that each system retrieves data about traffic flow, trajectories, moving cars, and current traffic conditions in each area, it is crucial to share critical information with each other, such as related to vehicles leaving Athens and heading towards Piraeus, e.g., their speed, and relationships among them, e.g., how close they are to each other. This is accomplished by communicating this information in terms of ontologies as directories in XML.

Fig. 6(b) depicts a scenario from the mobile communication world. In this application area, mobile, personalized agents and user profiles become a fundamental architectural component. A mobile agent is an application oriented towards the satisfaction of user profile and preferences. For example, the user of a mobile phone can define via Event-Condition-Action rules [13] that he/she wants to know at specific time points his/her position (i.e., which is the position of the mobile phone) in relation to specific museums, parks, parking lots, or other regions (see Fig. 5 for 3D-regions in time). The agent accomplishes this task, by exchanging information about the user's current position and trajectory. Note that the user might want to continue moving and getting information about the spatial relations as described in Sect. 3 (Fig. 5) by interacting, either directly with the server of the local database, or with other agents, for example, the agent of a local system, specializing in interesting tourist information. Again, the communication and exchange of information can be done without ambiguities by using mobile ontologies, such as trajectories, 3D-regions and their (topological) relations.

4. Architectural Issues

After having studied and registered the ontological domain of moving objects applications, we describe an architectural proposal towards handling ontologies.

4.1. Architectural Issues

In this section, we discuss some architectural issues relevant for an implementation of an ontology-based system (see Fig. 7). Several different proposals towards this

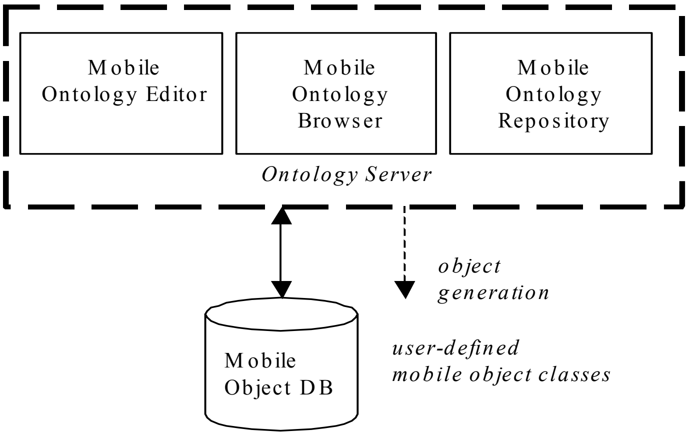

Fig. 7. An architectural proposal for mobile ontology-based applications.

direction *do* exist. Our goal is to show the basic components of such a system. The *Mobile Ontology Server* consists of a Mobile Ontology *Editor*, which is used to create and edit mobile ontologies, a Mobile Ontology *Browser* to browse and to choose the appropriate ontologies, and a Mobile Ontology *Repository*.

Typical languages to be used for the realization of mobile ontologies handled by these components are spatiotemporal UML [17], ONTOLIGUA, XML, Geo-XML [21], or other XML-based languages, such as the OIL ontology representation language [8].

After ontologies have been realized, objects and, further, relations, events, and processes can be defined. Object classes, such as a *trajectory* or a *moving object* are generated using ontologies. A moving object class can inherit characteristics from more than one mobile ontology, for example a *city area* can inherit properties from a 3D-region ontology and a 3D-line ontology, which is used to represent the river network in the area and needs to be taken into account when creating cities. This is a common issue already used in many other application environments, e.g., urban ontologies in GIS [9].

The issue in generating object classes from multiple ontologies can be easily resolved by an application that takes ontologies as inputs and outputs object classes. The creation of a new object class derived from ontologies is a semi-automatic process, whereas the definition of a subclass is derived from one or more super classes in a strict way. The Mobile Object DB (Fig.7) can be realized in any spatial DBMS such as Oracle 8i Spatial, while the ontology-object translation can be supported by tools such as, the Oracle XML SQL utility [14].

The dashed box in Fig. 7 contains the Ontology Server and its components. Connected to this is a database application containing objects that are created using ontologies (object generation) and thus allow for the exchange of data between the ontology server and the database.

Fig. 8 illustrates the relation between the Mobile Ontology Editor and the Mobile Object Database. As said, object classes are created from ontologies in a semi-automatic process.

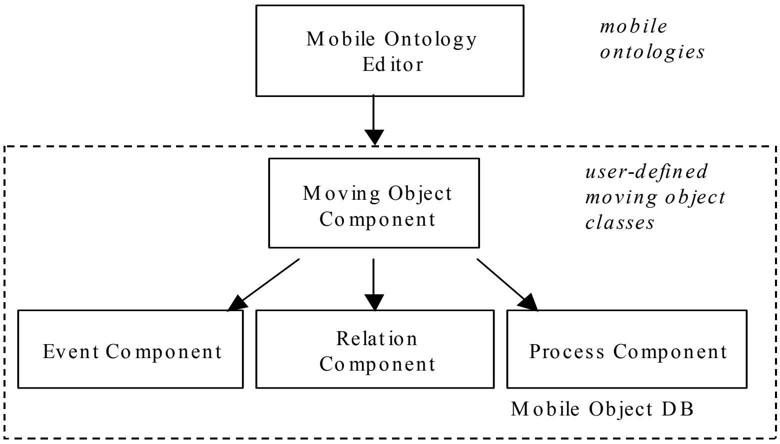

Fig. 8. Relations among ontological and database components.

5. Conclusions and Future Work

As the need for exchanging and sharing information related to movement becomes apparent, while on the other hand the use of ontologies for interoperability purposes has been long-praised, we focus on the comprehension and registration of the fundamental mobile ontologies which can be used to exchange useful information.

We use our experience from studies on traffic management systems and mobile information management systems that have been focused on requirements and implementation issues [13, 15]. To understand the domain of mobile applications, we concentrate on its basic characteristics and analyze the fundamental aspect of 'movement' of objects. This serves as the guideline to build the corresponding ontological world, i.e., mobile ontologies, their properties and their relations in UML. We use UML as a prototypical approach, and we further argue towards its use for this purpose. We present an architectural proposal combining ontologies and objects and show the interactions between different components of the ontological and object database. Finally, we show the applicability of this approach by illustrating two scenarios of the mobile world, each one using and exchanging ontologies at a different communication level.

This work is part of a larger project focusing on the information exchange of mobile systems [4]. Several research lines can be drawn at this point. Currently, we are working on the definition of a spatiotemporal XML based on an already existing spatiotemporal UML [17] and the transformation rules from one language to the other, in order to provide the Mobile Editor in UML. The definition of not only a Data Manipulation Language but a Query Language at an abstract level is also essential. Furthermore, we have been focusing on the design and implementation of mobile agents to exchange information with local servers in location-dependent service scenarios (Fig. 6(b)). After having understood the mobile domain and the needs of mobile ontologies, it is convenient to also adopt ontological languages. In this framework the adoption and enrichment of the promising OIL approach [8] with mobile features is one of the next goals.

References

1. Bartels, R.H., Beatty, J.C., and Barsky, B.A., 1987. An Introduction to Splines for Use in Computer Graphics & Geometric Modeling. Morgan Kaufmann Publishers, Inc.
2. Brodeur, J., Bédard, Y., Proulx, M.J., 2000. Modeling Geospatial Application Databases using UML-based Repositories Aligned with International Standards in Geomatics. *Proceedings of the 8th ACM Symposium on Advances in Geographic Information Systems (ACMGIS), Washington D*C.
3. COMPASS, 2001. IST Project Proposal, 7th Framework, European Commission (under submission).
4. DBGlobe, (2001-2003). A Data-Centric Approach to Global Computing. IST Project, European Commission.
5. Egenhofer, M., 1991. Reasoning about Binary Topological Relations Reasoning about Binary Topological Relations. *Proceedings of the 2nd Symposium on Large Spatial Databases, Zurich, Switzerland. Lecture Notes in Computer Science. Vol. 525. Springer-Verlag*.
6. ΙΧΝΗΛΑΤΗΣ, 2001. A System to Support Interoperability among Moving Objects Applications. Project proposal, Computer Technology Institute, Athens, Hellas.
7. Frank, A., 1997. Spatial Ontology: A Geographical Point of View. In *Spatial and Temporal Reasoning*, Kluwer, Dordrecht.
8. Fensel, D., Crubezy, F., van Harmelen, F., and Horrocks, M.I., 2000. OIL and UPML: A Unifying Framework fro the Knowledge Web. Proceedings of the Workshop on Applications of Ontologies and Problem-solving Methods. *14th European Conference on Artificial Intelligence (ECAI), Berlin, Germany*.
9. Fonseca, F., Egenhofer, M., Davis, C., and Borges, K., 2000. Ontologies and Knowledge Sharing in Urban GIS. *Computer, Environment and Urban Systems (to appear)*.
10. Gruber, Th., 1992. A Translation Approach to Portable Ontology Specifications. Knowledge Systems Laboratory – Stanford University, Stanford, CA, Technical Report KSL 92-71.
11. Gruber, Th., 1993. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In *Formal Ontologies in Conceptual Analysis and Knowledge Representation*. Kluwer Academics.
12. Güting, R.H., Böhlen, M., Erwig, M., Jensen, C.S., Lorentzos, N., Schneider, M., and. Vazirgiannis, M., 2001. A Foundation for Representing and Querying Moving Objects. *ACM TODS* 25(1):1-42.
13. Olesen, H., Tryfona, N., and Vinther, P., 2000. A Framework for User Profiles and Personal Agents in Mobile Communications. Aalborg University Technical Report.
14. Oracle XML SQL Utility for Java, http://technet.oracle.com/tech/xml/oracle_xsu. Current as of 10-9-01.
15. Pfoser, D., 2000. Issues in the Management of Moving Point Objects. PhD Dissertation, Aalborg University.
16. Pfoser, D., and Theodoridis, Y., 2000. Generating Semantics-Based Trajectories of Moving Objects. *International Workshop on Emerging Technologies for Geo-Based Applications, Ascona, Switzerland.*
17. Price, R.J., Tryfona, N., and Jensen, C.S., 2000. Extended Spatiotemporal UML: Motivations, Requirements, and Constructs. *Journal of Database Management. Special Issue on UML, 11, 4, (Oct.-Dec)*.
18. Rumbaugh, J., Blaha, M., Premerlani, W., Eddu, F., and Lorensen, W., 1991. Object-Oriented Modeling and Design. Prentice-Hall, Englewood Cliffs, NJ.
19. Smith, B., and Mark, D., 1998. Ontology and Geographic Kinds. *Proceedings of the International Symposium on Spatial Data Handling (SDH), Vancouver, Canada.*
20. Suzuki, J., and Yamamoto, Y., 1998. Making UML models exchangeable over the Internet with XML: UXF approach. *Proceedings of UML'98: Beyond the Notation*. Mulhouse, France.
21. Zhang, J., Javed, M., Shaheen, A., and Gruenwald, L., 2000. Prototype for Wrapping and Visualizing for Geo-Referenced Data in A Distributed Environment Using XML. *Proceedings of the 8th ACM Symposium on Advances in Geographic Information Systems (ACMGIS), Washington DC.*