Moving Objects in Databases and GIS: State-of-the-Art and Open Problems

Markus Schneider

University of Florida, Department of Computer & Information Science & Engineering
Gainesville, FL 32611, USA
mschneid@cise.ufl.edu

1 Introduction

The field of *moving objects databases* (Güting and Schneider 2005) has received a lot of research interest in recent years. This technology enables the user to model, store, retrieve, and query the movements of spatial objects over time, called *moving objects*, and to ask queries about such movements in a database context. A moving object represents the *continuous* evolution of a spatial object over time. In some cases, only the time-dependent locations are of interest, and we speak of *moving points*. Examples are mobile phone users, whales, ships, planes, terrorists, cars, spacecrafts, satellites, and missiles. In other cases, also the time-dependent shape and/or areal extent, which can grow or shrink, need to be handled, and we speak of *moving regions*. Examples are hurricanes, lakes, forest fires, oil spills, and the spread of diseases. In some rarer cases, the time-dependent shape and/or linear extent, which can lengthen or shorten, is of interest, and we speak of *moving lines*. Examples are snakes, the slowly retreating front of an army or a glacier, and the boundaries of moving regions in general. Much interest in moving objects databases has been spurred by current trends in consumer electronics, wireless communications, positioning technologies, and location-based services (Schiller and Voisard 2004). Corresponding hardware like wireless networking enabled and position-aware (i.e., GPS equipped) devices such as PDAs, on-board units in vehicles, sensors, or even special mobile phones have become affordable and will be in widespread use in the near future and trigger many new kinds of mobile, geographical applications. These applications will produce a huge volume of movement information that has to be managed and analyzed in database systems and be made available for spatiotemporal analysis in Geographic Information Systems (GIS). Unfortunately, current database technology and GIS technology are far away from being able to perform this task and thus require new data management and processing concepts and techniques.

The goal of this paper is to give an overview of the current state-of-the-art of moving objects databases and, in particular, to identify open research problems and indicate possible solutions.

Section 2 deals with moving objects in unconstrained environments. These are spatial objects that can freely change their location, shape, and extent. Section 3 describes moving objects in constrained environments. These are spatial objects whose temporal evolution is bounded due to spatial limitations like networks or labyrinths. In Section 4, we finally draw some conclusions.

2 Moving Objects in Unconstrained Environments

In the spatiotemporal database community, two main modeling paradigms have been proposed to characterize the movement of spatial objects. The first paradigm supports an *orthogonal* view. The concept is to handle multidimensional issues by decomposing them into different facets so that each facet (dimension) can be handled independently from the other. The benefit is that understanding and solving one facet at a time is much simpler than solving all facets together. For spatiotemporal issues this means that time and space are considered as separate facets and modeled as a part of type *time* × *space*. An approach supporting this paradigm is *MADS* (*Modeling Application Data with Spatio-temporal Features*) (Parent et al. 2006). The second paradigm supports a uniform view and emphasizes the functional dependence between time and space in spatiotemporal issues. This means that time and space are handled in an integrated way and modeled in terms of functions of type *time* → *space* as a particular subtype of *time* × *space* with special features like continuity. The author has contributed to and advocates an approach supporting this paradigm (Güting and Schneider 2005) since movement denotes the evolution of spatial objects over time.

Moving objects in unconstraint environments are either not impeded by spatial constraints (for example, the extension of hurricanes), or we are not interested in their spatial constraints (for example, the route of whales in oceans). The research literature has made a separation of this kind of moving objects into historical moving objects (Section 2.1) and predictive moving objects (Section 2.2).

2.1 Historical Moving Objects

Historical moving objects describe the temporal evolution of spatial objects *in the past* and are leveraged for spatiotemporal analysis. We first give an overview of available approaches and then identify some open research problems.

2.1.1 Overview

A widely accepted approach to modeling historical moving objects in databases introduces the fundamental concept of *spatiotemporal data types* (Erwig et al. 1999a). These data types enable the user to describe the continuous, dynamic, and time-dependent behavior and location change of spatial objects over time and to perform spatiotemporal analysis. That is, the spatial objects move, and they are therefore called *moving objects*. They are stored in special spatiotemporal databases called *moving objects databases* (Güting and Schneider 2005) and are designed as abstract data types, embedded as attribute types into a DBMS data model (relational, object-oriented, etc.), and equipped with a comprehensive collection of operations and predicates. Spatiotemporal data types are available for *moving points* (type *mpoint*), *moving lines* (type *mline*), and *moving regions* (type *mregion*). In case of moving regions, besides the movement aspect, one can also represent the change of their extent and shape over time. Conceptually, a moving point is a function f : *time* → *point*, a moving line is a function f : *time* → *line*, and a moving region is a function f : *time* → *region*. For example, for a moving region this means that at each time instant an object of type region has to be returned. The general ideas of this model have been presented in (Erwig et al. 1998, 1999a, 1999b). A formal specification of the spatiotemporal data types and operations has been given in (Güting et al. 2000).

Spatiotemporal predicates (Güting and Schneider 2005) describe changing topological relationships of moving objects over time. In the same way as spatial objects can change over time, the topological relationships between them can change over time. An example of such a predicate is the term *Disjoint* >> *meet* >> *Inside* >> *meet* >> *Disjoint* that is composed of a *temporal sequence* of the basic spatio-temporal predicates *Disjoint* and *Inside* as well as the topological predicate *meet*. The *temporal composition operator* is indicated by the symbol >>. This predicate could, for example, ask for a spatiotemporal pattern whether a plane is disjoint from a hurricane for some period, then meets the boundary of the hurricane at a time instant, is inside the hurricane for some period, meets the boundary of the hurricane again at a time instant, and is disjoint again from the hurricane for some period. The alternating sequence of topological predicates that hold for some period or for some time instant is characteristic for composite spatiotemporal predicates. A *spatiotemporal query language* called *STQL* has been introduced in (Erwig and Schneider 1999), and a *visual query language* called *Query-By-Trace* has been proposed in (Erwig and Schneider 2000, 2003).

Regarding implementation, data structures for moving objects have been presented in (Forlizzi et al. 2000) and algorithms for spatiotemporal operations have been designed in (Cotelo Lema et al. 2003). The author has co-authored a book titled *Moving Objects Databases* (Güting and Schneider 2005) covering all aforementioned topics.

Since then, the field has blossomed out, and much work has been done especially on implementation issues, e.g., on developing index structures (Aragwal et al. 2000; Hadjieleftheriou et al. 2002; Kollios et al. 1999; Pfoser et al. 2000), processing continuous queries (Song and Roussopoulos 2001; Tao and Papadias 2003), studying similarity of trajectories (Yanagisawa et al. 2003), developing test data generators (Theodoridis et al. 1999), to name only some of the areas.

2.1.2 Open Research Problems

From a modeling perspective, the author sees a general need to perform research on *spatiotemporal predicates* as the temporally lifted versions of spatial predicates. In a spatial database context, these predicates are needed as filter conditions for spatiotemporal joins and selections. While a model for the evolution of *topological relationships* over time is available (Erwig and Schneider 2002), models concerning the evolution of *directional relationships* (*cardinal directions*) over time have not been proposed. One reason might be that the purely spatial problem of modeling directional relationships in the two-dimensional space and the three-dimensional space has not been solved satisfactorily so far. In a spatiotemporal context, the problem is to detect directional patterns of moving objects. For example, a group of whales could have moved from location *X* in northwestern direction for a while, then turned to the south, then moved to the east, and finally moved to the north. It seems that from this directional information we can neither derive the overall cardinal direction seen from the starting point of the route nor the location of the whales.

From an implementation perspective, further work is needed to develop appropriate database-enabled data structures and algorithms for moving objects. Efficient algorithms are needed for spatiotemporal predicates of all kinds. For example, it is unclear how topological predicates over time can be evaluated (efficiently). Traditionally, work is needed that focuses on spatiotemporal index structures.

2.2 Predictive Moving Objects

Predictive moving objects describe the predicted temporal evolution of spatial objects *at the present time* and *in the near future*. We first give an overview of available approaches and then identify some open research problems.

2.2.1 Overview

The data model *MOST* (*Moving Objects Spatio-Temporal model*) (Sistla et al. 1997, 1998) is the only model so far that is able to describe current and expected future movement in a database context. All known application-specific models are independent of a database context. MOST enables the user to keep track of a set of time-dependent locations (i.e., moving points like mobile phone users) in a database in terms of *location-based management*. The model is based on the observation that one should not keep the positions directly in the database, leading to a very high volume of updates, but rather represent them by a *motion vector*. Only when the object's position predicted by the motion vector deviates from the real position by more than a threshold, an update needs to be transmitted to the database.

The fundamental idea is to introduce so-called *dynamic attributes* which change their values automatically with time. Not all attribute types are eligible to be dynamic. Such a type must have a value 0 and an addition operation. The dynamics is given by linear functions that describe *motion vectors* and avoid frequent database updates. Examples are the types *dynamic integer* and *dynamic real*. Unfortunately, there is no concept of dynamic spatial data types so that the only option to represent a “moving point” is to model it as a pair (*x* : *dynamic real*, *y* : *dynamic real*). Dynamic lines or regions cannot be modeled, and there is no concept of spatiotemporal data types available. If a query refers to a dynamic attribute A, its dynamic value is meant and used in the evaluation. Hence, the result depends on the time when the query is issued. If such a query is reevaluated on each clock tick, this query is called *continuous*.

A query language called *Future Temporal Logic* (*FTL*) (Ststla and Wolfson 1995; Sistla et al. 1997, 1998) allows one to express conditions about the future and specify temporal relationships between objects in queries. It especially introduces the temporal modal operators *until* and *nexttime* from which a collection of other temporal operators can be derived. This approach is restricted to moving points. An implementation of this model has been provided in a prototype called DOMINO and described in (Wolfson et al. 1998a, 1998b, 1999).

Another model uses cylindrical volumes to represent the uncertainty of the *trajectory* of a moving object (Trajcevski et al. 2002, 2004). Uncertainty is an inherent feature of current and near future locations of moving objects. This model takes into account the *temporal uncertainty* and the *spatial uncertainty* of objects. One can then ask for the objects that are inside a query region *sometimes* or *always* during a time interval (temporal uncertainty). Similarly, one may ask for the objects that are *possibly* or *definitely* inside a query region. A combination of both uncertainty aspects leads to a particular kind of *spatiotemporal predicates* like *PossiblySometimeInside* or *AlwaysDefinitelyInside*. Related work on uncertain trajectories includes (Mokhtar et al. 2002, Pfoser and Jensen 1999).

2.2.2 Open Research Problems

Despite some progress, research on predictive moving objects is still restricted. Current models do only refer to moving points which are modeled implicitly through dynamic attributes. Hence, the first research issue relates to the design and rigorous definition of a comprehensive data model for the predictive and near future evolution of moving objects. A requirement is that this data model supports a data type view on predictive moving objects, that is, predictive spatiotemporal data types, so that these data types can be later integrated into databases. A main challenge is the treatment of the inherent uncertainty that affects near future moving objects. A second issue refers to the design of spatiotemporal predicates for predictive moving objects. The main question here is to explore how moving objects with an individual, predicted behavior may behave towards each other. Such a behavior may comprise topological, directional, and distance relationships between these objects. Again, a challenge is the treatment of the uncertainty that also affects these relationships. A third issue is the design and implementation of effective and versatile data structures for predictive spatiotemporal data types and efficient algorithms for their operations and predicates. A fourth issue is the embedding of these novel concepts into a database query language and its integration into a database system. A fifth issue refers to a seamless integration of the data models and query languages for historical and predictive moving objects into database systems. The ultimate goal must be to obtain a data model that is capable of modeling historical *and* predictive movement in a single, seamless, consistent, and homogeneous framework. This will lead to a much more complete and powerful data model and allow queries that span the past and the future. An example is the query where a hurricane has been located since yesterday and where it will be in two days. Since all concepts are intended for a use in a database context, the issue is how the concepts for historical and predictive moving objects can be smoothly integrated into a database query language like SQL.

The author has started research on these issues which is led by the statement that it is *not* the task of a database system to predict the future movement of a moving object. The reason is that prediction models are tailored to specific application domains and based on external factors or domain-specific parameters which may significantly affect the future movements of moving objects. For instance, information such as atmospheric pressures, temperature zones, and wind and ocean currents plays a major role in predicting the future evolution of a hurricane. This requires highly specialized and sophisticated prediction models and algorithms beyond those in which only the past and current object movements are considered as system parameters. In fact, the development effort of such prediction methods is a whole discipline by itself, and this task belongs to domain experts, in this case meteorologists. Further, prediction models for different application domains are usually different. Therefore, a moving objects database system as a general-purpose tool should provide application-neutral modeling and persistence support for storing and retrieving predicted moving objects data and offer querying capability for retrieving such data. This means that it is solely reasonable to perform a separation between moving object models and prediction models with respect to representing future evolutions of moving objects. Since this separation has not been realized so far in existing future movement models, each of these models has only dealt with a specific problem area or object motion type while neglecting the problem of how a moving object database can model the future movements of moving objects in general.

The author proposes a novel algebra or type system called *Moving Balloon Algebra* for moving objects in unconstrained environments. So-called *balloon data types* enable the representation of both *historical*, *predictive*, and *time-spanning* (= historical + predictive) moving objects. The term *balloon* is used as a metaphor to model the nature of a moving object evolving in the past *and* extending to the future for a given current time

t^{now}. The *string* and the *body* of a *balloon object* represent its past and future movement respectively. For example, the past, known movement and the future, predicted movement of the eye of a hurricane are usually illustrated by using a shape that resembles a balloon. The past movement of the eye (a moving point) can be seen over time as the movement along a line or a curve which resembles the string of a balloon. On the other hand, the position of the eye at a time instant in the future can be anywhere within an area of uncertainty. Thus, the future movement of the eye can be seen as a moving region of uncertainty which resembles the body of a balloon. Furthermore, the connection point between the string and the body of a balloon object represents the present state of the moving object at t^{now}. From a data type perspective and ignoring the uncertainty aspect for a moment, we can describe the string by the spatiotemporal data type *mpoint* and the body by the type *mregion*. But this is not the only possible type combination. For example, assume a car driving on a road network has taken a particular route so far. At a junction where it is now at time t^{now}, it has several options since several roads may emanate from this junction. We assume that it is not exactly known which route the car will take. This situation can be modeled by a single-component *mpoint* object for the past and a multiple-component *mpoints* object for the possible future routes. Note the difference between the names and the semantics of the types *mpoint* and *mpoints* here. An interesting observation is that the dimension of the historical moving object representing the past of a time-spanning moving object is always less than or equal to the dimension of the predictive moving object representing the future. Due to uncertainty, the predictive object part cannot always be modeled with the same precision as the past object part.

So far, we have described that the set of potential future positions or the extent of a moving object can be modeled by a spatiotemporal data type like *mpoint* or *mregion*. However, this concept does not specify the relative change or degree of *confidence* with which a potential future position will eventually be the position of the moving object. To do this, we propose to employ a concept that we call *confidence distribution* such that each potential future position is associated with a degree of confidence. It is the task of an application-specific prediction model to provide an appropriate confidence distribution. Since the prediction model is unknown to a DBMS, our view is that the DBMS regards such a model as an *oracle* that can be consulted on request. This enables us for a future time instant to model the set of potential positions or the possible extent of an object by imposing a confidence distribution on a spatial object representing the future positions. We use the term *confidence* because we do not know how an application models uncertainty and because we permit any theory like probability theory, fuzzy set theory, or rough set theory. We assume that a confidence distribution is given by a function $f: \mathrm{IR}^2 \rightarrow [0, 1]$.

An interesting observation is that balloon objects are *static* although they describe the dynamic, temporal evolution of a moving object. The reason is that they represent snapshots describing what the past development and the predicted development of a time-spanning moving object at a time t^{now} is. Since the current time moves on, we obtain a time-spanning moving object for each current time instant t_i^{now}. This movement represents the dynamics, and we speak of *moving balloon objects*. So-called *spatio-temporal balloon data types* are supposed to enable us to represent them. If we compare t_i^{now} and t_{i+1}^{now}, at t_{i+1}^{now} we know more about the past. That is, the known past at t_i^{now} is contained in the past known at t_{i+1}^{now}. On the other hand, we will probably obtain different predictions at different current times. This is interesting from a querying perspective since it enables us to compare predictions at different current times and to make statements about their validity. Examples of interesting new queries are: (1) What area will potentially be affected by the hurricane XYZ in 10 hours from now? (2) What was thought 12 hours ago about the potentially affected area of XYZ 22 hours later? (3) Assess the match or similarity of both predictions. (4) What is the chance that XYZ will traverse the city ABC 16 hours from now? (5) Determine all cities for which the degree of confidence that they will be hit by XYZ is larger than 70%. (6) Identify all flight routes that are potentially affected by XYZ in the next 24 hours. (7) What is the development of XYZ from three days ago until in 8 hours?

3 Moving Objects in Constrained Environments

Many spatial objects like cars, planes, trains, and people in buildings are restricted in their possible motions since they move in *spatially embedded networks* like roads, highways, railways, lanes in factories for robots, buildings, or airplane routes. We speak of *moving objects in constrained environments*. Therefore, as a conclusion, spatial networks should be taken into account in a data(base) model and database query language for moving objects. This would make it possible to describe movement relative to a network rather than the general 2D space. It would also enable an easier formulation of queries and more efficient representations and indexing of moving objects for network-based database applications.

Section 3.1 deals with research on static spatial networks. Section 3.2 focus on moving objects in these networks.

3.1 Spatial Networks

Spatial(*ly embedded*) *networks* or *spatial graphs* are an important spatial concept in the geosciences and have been widely discussed in the literature. They describe a *spatial connectivity structure* (another important one would be *spatial partitions* (Erwig and Schneider 1997, McKenney and Schneider 2007)) and consist of a set of point objects representing their nodes and a set of line objects describing the geometry of their edges. Examples are transportation networks and supply networks. However, the modeling, representation, and integration of *spatial networks in databases* are a rather open research issue. We first give an overview of available approaches and then identify some open research problems.

3.1.1 Overview

Without going into detail, the manipulation of abstract, non-spatial graphs (networks) in databases has received quite a bit of attention in the database field. However, research on *spatial*(*ly embedded*) *graphs* or *spatial networks* is rather limited. Spatial graphs are graphs whose nodes and edges are annotated with a geometry and are thus embedded in the Euclidean space. Chapter 6 of (Shekar and Chawla 2003) gives a good overview of the current state-of-the-art of representing, querying, and implementing spatial graphs in databases. The current representation strategy of spatial graphs is the same as in the non-spatial case: Networks are modeled at a very low abstraction level (based on identifiers and (x, y)-coordinates) by explicit node and edge relations but they are not visible as self-contained entities in the database. Further, there is no explicit support of paths, which the author regards as important. The formulation of queries in SQL is cumbersome, even if a special *connect* clause in SQL2 and a *recursive* clause in SQL3 enable the recursive traversal of a graph structure by deriving the transitive closure of a relation. Commercial solutions like the ESRI Network Data Model (Zeiler 1999) or the Oracle Network Data Model (Oracle Corporation 2005, 2006) follow exactly that strategy. The consequence for applying network operations is that the network data have to be loaded from the node, edge, and many other database tables into a middleware layer outside the database system and that network operations are implemented and executed in this middleware layer. That is, mass data are handled outside the database, network data are kept twice, and updates have to be reported to the database in order to maintain consistency. All this causes a lot of unnecessary overhead. Core database functionality like transactions, query processing, concurrency control, and recovery cannot or can only very limitedly be performed or used in the middleware layer.

The reason of this development is the fact that graphs or networks are not (but should be) first-class concepts in DBMSs. The only approaches that go into his intended direction of treating graphs or networks as first-class objects can be found in (Amann and Scholl, 1992, Güting 1991, 1994). In (Güting 1991) relations and graphs coexist as modeling facilities but a graph consisting only of nodes is practically the same as a relation. If several graphs occur in a database, it is hard to separate them in the design. The approach in (Amann and Scholl 1992) offers node and edge objects but no path objects. The proposal in (Güting 1994) is the most interesting one and offers node, edge, and path objects in an object-oriented setting. A number of special implementation problems have been considered like query processing in spatial network databases (Papadias et al. 2003, Shekhar et al. 1993), matching different road networks (Chen et al. 2006), clustering objects in spatial networks (Yiu and Mamoulis 2004), processing of spatial network queries (Huang et al. 1996, 1997; Shaw et al. 2006), and k-nearest neighbor search in spatial networks (Almeida and Güting 2006).

3.1.2 Open Research Problems

From a conceptual standpoint, a first research issue is the modeling and formal definition of spatially embedded networks and operations on them. Standard databases only allow the user to model spatial networks by the standard facilities of a DBMS data model. This is rather problematic since such a model is unable to design networks as self-contained entities, to represent their spatial connectivity, and to specify and process operations on them (like shortest path). Spatial databases, which are an improvement regarding spatial data handling and offer spatial data types (Schneider 1997), are only able to represent the geometry (points, lines) of the components of spatial networks but do also not have a concept of their connectivity and are also unable to specify operations on them. Hence, the requirement is that such a data model should have an explicit concept of a network embedded in space. That is, spatial networks should be first-class objects in spatial databases. Fulfilling this requirement would lead to easier and more powerful formulations of queries. A second issue is that the classical modeling of networks through nodes and edges might be too simplistic. *Paths* over networks (graphs) are often, or even more, the main conceptual entities of interest. For example, in a road network, roads represent paths over such a network. Of less importance are nodes representing their junctions and edges corresponding to the parts between junctions. A third issue is that networks should not only represent geometric information but also be labeled by thematic information. Further, the thematic information about a network should be extensible and leverage the standard facilities of the DBMS data model. In a relational environment we should be able to create relations to add information relative to a network. For example, it should be possible to label the components of a highway network with information describing motels or speed limits. A fourth issue is the exploration of operations and predicates within a single network and between different networks. Operations on single networks include shortest path computations based on network distance or user-defined weights. Interesting questions on different networks are how they can be combined (union, intersection, difference) and which relationships we can find between them. A fifth issue is the design of a suitable DBMS query language that incorporates the operations and predicates on networks.

From an implementation standpoint, the first research issue relates to the integration of spatial networks into DBMS. Due to the fact that spatial networks are not first class objects in databases, available implementations are nowadays only provided as middleware layers *outside* the DBMS. The role of databases is limited to delivering basic geometric and thematic components of networks so that networks can be constructed and network operations can be executed in the middleware layer. However, this means that already available standard and spatial database functionality has to be sourced out into the middleware layer, network data have to be duplicated, large volumes of data have to be transferred, and network operations have to be processed in the middleware layer. All this overhead can be avoided if spatial networks and their operations are part of the spatial DBMS. A second issue refers to the design of effective data structures for spatial networks and efficient algorithms for network operations. Classical data structures for networks like the *doubly-connected edge list* (DCEL) or the *winged-edge data structure* are mainly main-memory structures. They have the negative features that they have to be stored in a number of tables and that they require random array access. This is difficult to implement in a DBMS context. Further, they do not harmonize with the concept of spatial data types. A persistent version of these data structures makes it very difficult to support the algorithms implementing the network operations. Hence, the task is to find a better representation of these data structures or even other data structures that support these operations better. Due to their importance, especially paths in networks should be supported. A third issue is the design of algorithms for operations and predicates between different networks.

The author has started some research on this topic and proposes a novel type system or algebra called *Spatial Network Algebra* (*SNETALG*) for modeling and implementing spatial networks in databases. This concept defines networks on the basis of *routes* and *junctions* (crossings) (and not nodes and edges). Routes correspond to roads, streets, or highways in real life and to paths in a graph. A route itself can have several properties. It can be bi-directional (dual route), one-directional (directed route), or the direction does not matter (simple route, example: pedestrian zone). Further, it can have several lanes, and the number of lanes can change within the same route. This is, e.g., interesting for traffic management. Junctions correspond to intersection or meeting points of routes. They play a special role since they are responsible for the connectivity in a network and determine the allowed motion directions ((no) U-turn, one-way). Routes usually have names (like road names) but junctions and parts of roads between junctions usually do not have names.

Abstract data types like *network*, *npoint* (network point), and *nline* (network line) are proposed to represent the network, a position within the network, and a section of the network respectively. A comprehensive collection of operations and predicates includes construction and transformation (import and export) operations from and to predefined network relations, operations on networks (e.g., shortest path), between networks (e.g., union), and between networks and spatial data types (e.g., part of network intersected by region).

3.2 Moving Objects in Spatial Networks

Adding movement relative to a spatial network leads us to *moving objects in networks*. That is, the motion of spatial objects (often point objects) is constrained by a static network. This shows great promise for interesting, new operations and predicates and thus new and, compared to the unconstrained case, different kinds of queries. We first give an overview of available approaches and then identify some open research problems.

3.2.1 Overview

Data models for *moving objects in networks* are rare. The MOST model (Sistla et al. 1997, 1998) uses a particular dynamic location attribute which includes a polyline representing a path over the network plus some additional information like start time, start position, and speed. However, the underlying network is not modeled in any way. Hence, there is no explicit relationship to the network any more. The problem of finding relationships between moving objects and the network in queries is not addressed. The approach in (Vazirgiannis and Wolfson 2001) considers modeling and querying moving objects in road networks. The network model is a relation representing “blocks” that are the edges of the network graph. Each tuple contains a polyline describing the geometry of the edge. The model basically corresponds to an undirected graph where nodes are street crossings and edges are city road blocks. The model is not defined formally but rather an application-specific model. The approaches in (Hage et al. 2003; Jensen et al. 2003b; Speicys et al. 2003) look at data modeling issues for spatial networks regarding possible uses for location-based services. These interesting application studies emphasize the large complexity of real road networks that cannot be adequately modeled by simple directed graph models; they are a good motivation for our planned work. The approach in (Güting et al. 2006) is so far the only one proposing data types for moving objects in networks. Implementation-oriented aspects have, e.g., dealt with index structures for moving objects in networks (Renzos 2003; Pfoser and Jensen 2003), query processing algorithms for networks (Jensen et al. 2003a; Shahabi et al. 2003), and building test data generators for network-based moving objects (Brinkhoff 2002).

3.2.2 Open Research Problems

Movement in networks can be mainly evoked by two kinds of spatial objects. Either point objects like cars or robots move, or line objects like traffic jams evolve in a network. In this case, we may use our well known spatiotemporal data types for modeling these moving objects, only with the restriction that their motion is constrained to a network.

Considering moving objects in networks results in a number of conceptual research issues. The first issue refers to the problem that both spatial networks and moving objects model geometry and locations. But it is sufficient to store geometry only once. Hence, the question is how moving objects should be modeled in a spatial network without redundant geometric information. This seems to indicate that a new and, compared to the unconstrained case, different data model of moving objects is needed in the network case. A second issue is that the model should allow the user to describe static or moving objects relative to the network, such as static positions (e.g., motels, restaurants, gas stations), static regions (e.g., construction sites, speed limit zones), moving positions (e.g., vehicles), and moving regions (e.g., traffic jams, network parts affected by inundations or hurricanes). A third issue relates to the possible transfer or adaptation of available operations and predicates for moving objects in unconstrained environments to the network case and to the identification and design of possibly new operations and predicates for moving objects in networks. For example, the Euclidean distance concept for unconstrained environments is not meaningful in spatial networks. A fourth issue is the design of a query language for moving objects in networks. A fifth issue is the seamless integration of the data models and query languages for moving objects in unconstrained and constrained environments into a homogeneous and consistent framework.

The consideration of moving objects in networks also leads to implementation issues. A first issue is that if moving objects in networks should have to be modeled differently, different data structures will be needed for them. The consequence would be that such moving objects would be bound to a network and could not be used in an unconstrained environment. A second issue is the immediate impact on the design of algorithms for the operations and predicates on such moving objects. A third issue is the consistent integration of moving objects in networks with their unconstrained counterparts as well as spatial networks. A fourth issue is the embedding of all these concepts into databases.

The author has started some research on this topic and proposes a novel type system or algebra called *Moving Objects in Networks Algebra* (*MONET Algebra*) to model and implement moving objects in spatial networks. The first task is to design appropriate abstract data types for the objects that can move in a network. We have seen that applying the temporal type constructor $\tau(\alpha) = time \rightarrow \alpha$ to the spatial data types $\alpha \in \{point, line\}$ leads to the spatiotemporal data types $\tau(point) = mpoint$ and $\tau(line) = mline$. In the same way, we can now obtain the spatiotemporal data types *mnpoint* and *mnline* whose movement is constrained to a network by applying τ to the network types *npoint* and *nline*. That is, $mnpoint = \tau(npoint)$ and $mnline = \tau(nline)$. The second task is to devise meaningful operations and predicates on these types. An issue is whether some of the operations on moving objects in unconstrained environments can be transferred to the network case. The third task is the design and implementation of effective data structures for the data types and efficient algorithms for the operations and predicates. The fourth task is the design of a query language that is in accord with the query language for networks. The fifth task is the database and query language integration of all concepts in accordance with the integration of spatial networks.

Interesting and new queries for spatial networks and moving objects in them will be possible. Example queries are: (1) Find all sections (edges) of University Avenue that are longer than one mile. (2) Determine the part of the Gainesville road network located east of Main Street. (3) Determine the location of postman Miller at 3pm last Friday. (4) In which part of the network did the postman deliver letters between 10 am and 1pm of last Monday? (5) How many cars have passed Main Street in the last five hours? (6) Which postman is currently nearest to Main Street and moving into that direction (to give him a letter)? (7) Compute a list that determines the current number of construction sites (modeled as *nline* objects) of all U.S. interstates in decreasing order. (8) What are currently the ten longest traffic jams (modeled as *mnline* objects) on the interstate network? (9) Which parts of the network are affected by fog?

The problem that both networks and moving objects in them have absolute locations is solved by only maintaining the absolute locations of networks and by modeling the positions of moving objects relative to the length of a route (rather than an edge of a graph). This is very similar to the concept of *linear referencing* widely used in the *GIS in Transportation* literature and available in commercial database products such as Oracle Spatial (Oracle Corporation 2000). If positions are given relative to edges, then, e.g., a car along an interstate at constant speed needs a change of description at every exit and junction because the edge identifier changes. If positions are given relative to routes, then the description only needs to change when the car changes or leaves the interstate.

4 Conclusions

Moving objects are ubiquitous in our life and a promising concept to adequately model and represent the changing space-time behavior of geometric objects in spatiotemporal applications. From a conceptual standpoint, this chapter gives an overview of the state-of-the-art of moving objects technology in the context of databases and GIS. It especially distinguishes moving objects in unconstrained environments (historical moving objects, predictive moving objects) and constrained environments (spatial networks, moving objects in spatial networks) and sketches a number of open problems in these research fields.

References

Agarwal PK, Arge L, Erickson J (2000) Indexing Moving Points. In: ACM Symp. on Principles of Database Systems, pp 175–186

de Almeida VT, Güting RH (2006) Using Dijkstra's Algorithm to Incrementally Find the k-Nearest Neighbors in Spatial Network Databases. In: ACM Symp. on Applied Computing, pp 58–62

Amann B, Scholl M (1992) Gram: A Graph Data Model and Query Language. In: European Conf. on Hypertext Technology

Brinkhoff T (2002) A Framework for Generating Network-Based Moving Objects. GeoInformatica 6(2): 153–180

Chen C-C, Shahabi C, Knoblock CA, Kolahdouzan M (2006) Automatically and Effciently Matching Road Networks with Spatial Attributes in Unknown Geometry Systems. In: Int. Workshop on Spatio-Temporal Database Management

Cotelo Lema JA, Forlizzi L, Güting RH, Nardelli E, Schneider M (2003) Algorithms for Moving Objects Databases. Computer Journal: 680–712

Erwig M, Schneider M (1997) Partition and Conquer. In: 3rd Int. Conf. on Spatial Information Theory (COSIT'03), LNCS 1329, Springer, Berlin Heidelberg New York, pp 389–408

Erwig M, Schneider M (1999) Developments in Spatio-Temporal Query Languages. In: IEEE Int. Workshop on Spatio-Temporal Data Models and Languages, pp 441–449

Erwig M, Schneider M (2000) Query-By-Trace: Visual Predicate Specification in Spatio-Temporal Databases. In: 5th IFIP 2.6 Working Conf. on Visual Database Systems

Erwig M, Schneider M (2002) Spatio-Temporal Predicates. IEEE Trans. on Knowledge and Data Engineering 14(4): 1–42

Erwig M, Schneider M (2003) A Visual Language for the Evolution of Spatial Relationships and its Translation into a Spatio-Temporal Calculus. Journal of Visual Languages and Computing 14(2): 181–211

Erwig M, Güting RH, Schneider M, Vazirgiannis M (1998) Abstract and Discrete Modeling of Spatio-Temporal Data Types. In: ACM Symp. on Geographic Information Systems, pp 131–136

Erwig M, Güting RH, Schneider M, Vazirgiannis M (1999a) Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. GeoInformatica 3(3): 265–291

Erwig E, Schneider M, Güting RH (1999b) Temporal Objects for Spatio-Temporal Data Models and a Comparison of Their Representations. In: Advances in Database Technologies (ER'98 Workshop on Spatio-Temporal Data Management), LNCS 1552, Springer, Berlin Heidelberg, New York, pp 454–465

Fenzos R (2003) Indexing Moving Objects on Fixed Networks. In: Int. Symp. on Spatial and Temporal Databases, pp 289–305

Forlizzi L, Güting RH, Nardelli E, Schneider M (2000) A Data Model and Data Structures for Moving Objects Databases. In: ACM SIGMOD Int. Conf. on Management of Data, pp 319–330

Güting RH (1991) Extending a Spatial Database System by Graphs and Object Class Hierarchies. In: Int. Workshop on Database Management Systems for Geographical Applications

Güting RH (1994) GraphDB: Modeling and Querying Graphs in Databases. In: Int. Conf. on Very Large Data Bases, pp 297–308

Güting RH, Schneider M (2005) Moving Objects Databases. Morgan Kaufmann Publishers

Güting RH, Böhlen MH, Erwig M, Jensen CS, Lorentzos NA, Schneider M, Vazirgiannis M (2000) A Foundation for Representing and Querying Moving Objects. ACM Trans. on Database Systems 25(1): 1–42

Güting RH, de Almeida VT, Ding Z (2006) Modeling and Querying Moving Objects in Networks. VLDB Journal 15(2): 165–190

Hadjieleftheriou M, Kollios G, Tsotras VJ, Gunopulos D (2002) Efficient Indexing of Spatiotemporal Objects. In: Int. Conf. on Extending Database Technology, pp 251–268

Hage C, Jensen CS, Pedersen TB, Speicys L, Timko I (2003) Integrated Data Management for Mobile Services in the Real World. In: Int. Conf. on Very Large Data Bases, pp 1019–1030

Huang YW, Jing N, Rundensteiner EA (1996) Path Queries for Transportation Networks: Dynamic Reordering and Sliding Window Paging Techniques. In: ACM Symp. on Geographic Information Systems, pp 9–16

Huang YW, Jing N, Rundensteiner EA (1997) Integrated Query Processing Strategies for Spatial Path Queries. In: Int. Conf. on Data Engineering, pp 477–486

Jensen CS, Kolavr J, Pedersen TB, Timko I (2003a) Nearest Neighbor Queries in Road Networks. In: ACM Symp. on Geographic Information Systems, pp 1–8

Jensen CS, Pedersen TB, Speicys L, Timko I (2003b) Data Modeling for Mobile Services in the Real World. In: Int. Symp. on Spatial and Temporal Databases, pp 1–9

Kollios G, Gunopulos D, Tsotras VJ (1999) On Indexing Mobile Objects. In: ACM Symp. on Principles of Database Systems, pp 261–272

McKenney M, Schneider M (2007) Spatial Partition Graphs: A Graph Theoretic Model of Maps. In: 10th Int. Symp. on Spatial and Temporal Databases, LNCS 4605, Springer, Berlin Heidelberg New York, pp 167–184

Mokhtar H, Su J, Ibarra OH (2002) On Moving Object Queries. In: ACM Symp. on Principles of Database Systems, pp 188–198

Oracle Corporation (2000) Oracle Spatial Linear Referencing System Release 8.1.6. User’s Guide

Oracle Corporation (2005) Oracle Database 10g: Oracle Spatial Network Data Model. An Oracle Technical White Paper

Oracle Corporation (2006) Oracle Spatial Topology and Network Data Models 10g Release 2. Oracle Spatial User’s Guide and Reference

Parent C, Spaccapietra S, Zimányi E (2006) Conceptual Modeling for Traditional and Spatio-Temporal Applications : The MADS Approach. Springer, Berlin Heidelberg New York

Papadias D, Zhang J, Mamoulis N, Tao Y (2003) Query Processing in Spatial Network Databases. In: Int. Conf. on Very Large Data Bases, pp 802–813

Pfoser D, Jensen CS (1999) Capturing the Uncertainty of Moving-Object Representations. In: Int. Symp. on Advances in Spatial Databases, pp 111–131

Pfoser, Jensen CS (2003) Indexing of Network Constrained Moving Objects. In: ACM Symp. on Geographic Information Systems, pp 25–32

Pfoser D, Jensen CS, Theodoridis Y (2000) Novel Approaches in Query Processing for Moving Object Trajectories. In: Int. Conf. on Very Large Data Bases, pp 395–406

Schiller J, Voisard A (eds) (2004) Location-Based Services. Morgan Kaufmann Publishers

Schneider M (1997) Spatial Data Types for Database Systems – Finite Resolution Geometry for Geographic Information Systems, LNCS 1288. Springer, Berlin Heidelberg New York

Shahabi C, Kolahdouzan MR, Sharifzadeh M. A Road Network Embedding Technique for K-Nearest Neighbor Search in Moving Object Databases. GeoInformatica 7(3): 255–273

Shaw K, Sample J, Ioup E, Abdelguerfi M, Mansion O (2006) Graph processing for spatial network queries. In: Int. Conf. on Information and Knowledge Engineering, pp 3–9

Shekar S, Chawla S (2003) Spatial Databases: A Tour. Prentice Hall

Shekhar S, Kohli A, Coyle M (1993) Path Computation Algorithms for Advanced Traveller Information Systems. In: Int. Conf. on Data Engineering, pp 31–39

Sistla AP, Wolfson O. Temporal Triggers in Active Databases. IEEE Trans. on Knowledge and Data Engineering 7(3): 471–486

Sistla AP, Wolfson O, Chamberlain S, Dao S (1997) Modeling and Querying Moving Objects. In: Int. Conf. on Data Engineering, pp 422–432

Sistla AP, Wolfson O, Chamberlain S, Dao S (1998) Querying the Uncertain Position of Moving Objects. In: Etzion O, Jajodia S, Sripada S (eds) Temporal Databases: Research and Practice, LNCS 1399, Springer, Berlin Heidelberg New York, pp 310–337

Speicys L, Jensen CS, Kligys A (2003) Computational Data Modeling for Network-Constrained Moving Objects. In: ACM Symp. on Geographic Information Systems, pp 118–125

Song Z, Roussopoulos N (2001) K-nearest Neighbor Search for Moving Query Point. In: 7th Int. Symp. on Spatial and Temporal Databases, pp 79–96

Tao Y, Papadias D (2003) Spatial Queries in Dynamic Environments. ACM Trans. on Database Systems 28(2): 101–139

Theodoridis Y, Silva JRO, Nascimento MA (1999) On the Generation of Spatiotemporal Datasets. In: Int. Symp. on Advances in Spatial Databases, pp 147–164

Trajcevski G, Wolfson O, Hinrichs K, Chamberlain S (2004) Managing Uncertainty in Moving Objects Databases. ACM Trans. on Database Systems: pp 463–507

Trajcevski G, Wolfson O, Zhang F, Chamberlain S (2002) The Geometry of Uncertainty in Moving Objects Databases. In: Int. Conf. on Extending Database Technology, volume LNCS 2287, Springer, Berlin Heidelberg New York, pp 233–250

Vazirgiannis M, Wolfson O (2001) A Spatiotemporal Model and Language for Moving Objects on Road Networks. In: Int. Symp. on Spatial and Temporal Databases, pp 20–35

Wolfson O, Chamberlain S, Dao S, Jiang L, Mendez G (1998a) Cost and Imprecision in Modeling the Position of Moving Objects. In: Int. Conf. on Data Engineering, pp 588–596

Wolfson O, Xu B, Chamberlain S, Jiang L (1998b) Moving Objects Databases: Issues and Solutions. In: Int. Conf. on Scientific and Statistical Database Management, pp 111–122

Wolfson O, Sistla AP, Chamberlain S, Yesha Y (1999) Updating and Querying Databases that Track Mobile Units. Distributed and Parallel Databases 7: 257–287

Yanagisawa Y, Akahani J, Satoh T (2003) Shape-based Similarity Query for Trajectory of Mobile Objects. In: Int. Conf. on Mobile Database Management, pp 63–77

Yiu ML, Mamoulis N (2004) Clustering Objects on a Spatial Network. In: ACM SIGMOD Int. Conf. on Management of Data, pp 443–454

Zeiler M (1999) Modeling Our World: The ESRI Guide to Geodatabase Design. ESRI Press