Moving Objects Databases:
Achievements and Challenges

M. Andrea Rodríguez-Tastets.
Department of Computer Science,
University of Concepción, Chile
and
Center for Web Research,
University of Chile, Chile

INTRODUCTION

Moving objects databases are particular cases of spatio-temporal databases that represent and manage changes related to the movement of objects. Unlike spatio-temporal applications associated with geographic phenomena where the identity of geographic features may change over time, in moving objects databases the objects maintain their identities but change their locations or shapes through time. That is, it is the geometric aspect of an object that changes rather than the object itself. Within this domain, the most suitable applications are those where objects are cars, airplanes, or any object with regular movements.

Traditional DBMSs are not well equipped to handle data about moving objects. One of the reasons is that DBMSs assume that data is constant unless an explicit modification occurs, and this assumption is not adequate for handling continuously changing data such as the locations of moving objects. In traditional DBMSs it is difficult to specify queries about spatial and temporal information. For example, a query such as “retrieve the cars that will intersect at a particular location in an hour” is not easily expressed with SQL. Finally, location of a moving object is inherently imprecise because the location stored in the database cannot always be the actual location of the object (Wolfson, 2002).

Unlike traditional database applications, moving objects applications involve the following requirements, which are a subset of spatio-temporal applications requirements (Pfoser & Tryfona, 1998):

- the need for representing objects, such as moving cars, with a position in space and an existence in time;
- the need to capture the change of position over time. This change of position may be continuous or discrete;
- the need for representing spatial relations among objects in time; and
- the need to specify spatio-temporal integrity constraints.

BACKGROUND

Initially, research in databases handled time and space separately. It was only in the nineties that spatio-temporal databases become an area of active research. The evolution of spatio-temporal databases and therefore, of moving objects databases, involves issues at different levels. For example, at the ontological level, the semantics built into a spatio-temporal database must be in agreement with the ontological concepts related to the space and time of moving objects (Frank, 2003). At the conceptual level, spatio-temporal requirements are expressed in terms that are independent of any particular data model. Conceptual modeling of spatio-temporal databases should consider spatial and temporal aspects associated not only with objects, but also with attributes and relationships (Tryfona et al., 2003). At the level of data models and languages, the typical components for representing moving objects are abstract data types and object classes that incorporate spatial and temporal aspects (Güting et al., 2003).

The basic components of spatio-temporal databases are spatial objects, which are finite sets of points in a space (Pfoser & Tryfona, 2001). From a temporal perspective, properties and relationships are considered facts of objects and, therefore, they can be assigned truth-values. There are different types of temporal aspects that have been traditionally discussed:

> *valid time* is the time when a fact is true in a modeled reality;
> *transaction time* is the time when an element in the database, which is not necessarily a fact, is part of the current state in the database; and
> *existence time* of an object refers to the time when the object exists in reality.

In the context of moving objects, the valid time of a given object is the current, past and future position that has been recorded. The transaction time of a position refers to the current and past positions that were recorded as current in the database. The existence time is associated only with the existence of an object and not with its position.

Different types of models for time are time points versus time intervals. A time point is an instant in time, whereas a time interval is defined by a start and end point in time. A spatio-temporal database may store events or states. An event usually occurs at a specific time point, without duration; for example, an event may be that a car crashed. A state, in contrast, has duration and is defined for each time point within an interval; for example, the state of a car that is parked.

ACHIEVEMENTS OF MOVING OBJECTS DATABASES

This Section reviews the development of moving objects databases that concern conceptual modeling, logical models and query languages, and spatio-temporal access methods. This review will be basis for presenting trends for future moving objects databases.

Conceptual Modeling

Conceptual modeling aims at providing a direct mapping between the perceived real world and its representation. To fulfill this goal, conceptual models should offer constructs sufficiently powerful to express a model of reality. The current proposals of such constructs include, at a minimum, objects types, relationship types and attributes. For spatio-temporal databases, these constructs are associated with spatial or temporal concepts (Tryfona et al., 2003).

The traditional strategy for spatio-temporal conceptual modeling has been to extend existing models with constructs that accommodate the requirements of spatial and temporal information. An extension of the Entity-Relation model (ER) to a spatio-temporal ER (STER) (Tryfona & Jensen, 1999; Tryfona et al., 2003) incorporates temporal, spatial and both spatio-temporal aspects in the specification of constructs. Indeed, the STER allows one to model spatial, temporal and spatio-temporal cases of entities, attributes and relationships. STER facilitates the integration of file-based and object-based views of the space (Shekhar *et al.*, 1997), the explicit representation of topological relations between objects, as well as, the representations with multiple granularities.

A different approach to modeling moving objects databases is to consider an extension to UML (Price et al., 2000). The Spatio-Temporal UML supports changes of the instantiated UML elements (i.e., objects, associations, and attributes instances) to provide for associated time periods or spatial extents (Tryfona el at., 2003). The extension incorporates a minimum set of constructs for spatial, temporal and thematic data, which can model temporal changes in spatial extents or location, changes in the value of attributes across time or space, and composite data whose components varying depending on time or location. These constructs can then be applied to any UML class diagram and UML model element. The specification of the spatio-temporal semantics of time units (e.g., instants and intervals), time and space dimensions (e.g., existence, valid and transaction time), models (e.g., object versus field-based space models) and interpolations (e.g., discrete, linear, or spline) are given in specification boxes, which can be associated with any icons or combinations using a unique naming label.

A hybrid ER/OO model for applications with spatio-temporal features, called Modeling of Application Data with Spatio-temporal features (MADS), was explored in (Parent et al., 1999). In this approach, an object-based model is extended with pre-defined hierarchies of spatial and temporal abstract data types and spatial complex data types to describe the properties of attributes (i.e., name, cardinality, temporal and spatial dimensions). Spatio-temporal features in MADS can also be associated with objects, attributes, and relationships. The spatial features of MADS supports the discrete or continuous view of space where the spatial domain for any space-varying information is the geometry of any selected item. Relationships in MADS may be of different types, such as is-a relationships, aggregation relationships, constraint relationships (i.e., spatial relationships and synchronization relationships) and dynamic relationships (i.e., transition and generation relationships). This model does not directly support data elements associated with several different spatial extents. In such cases, the data element must be modeled as an association of spatial objects.

Data Models and Languages

Some practical solutions to the modeling and querying of moving objects propose extensions based on abstract data types (ADTs) (Forlizzi et al., 2000; Güting et al., 2000; Güting et al., 2003). They model *moving points* and *moving objects* as three-dimensional (2D + time) or higher dimensional entities whose structure and behavior is captured in an ADT. Designing types and operations to represent moving objects, however, may also require types other than moving points and moving regions (e.g., lines for modeling trajectory). Once the ADTs are defined, they can be integrated into relational or object-oriented databases and their operations can be used in queries.

The ADT approach focuses on modeling spatial and temporal relationships that can be described with algebraic geometry. Additional operations are also included into ADTs for computing velocity, derivative, turn, and speed. An extension to ADTs that deals with the modeling of moving objects is to define spatio-temporal predicates and their composition in order to describe the development of relationships between moving objects (Güting et al., 2003). For example, two moving objects may be first 10 miles from each other, then intersect, and finally be 10 miles from each other again.

Constraint databases represent another alternative to data modeling for spatial and temporal data. The idea is to use a mathematical formulation to represent spatial and temporal data as infinite collection of points that satisfy first-order formulae. The constraint-based model allows a uniform representation of all kinds of spatio-temporal data and supports declarative query languages that are well-suited for complex spatio-temporal queries (Grumbach et al., 2003).

There are different constraint-based models for spatio-temporal data. In the original constraint data model, constraints are expressed as linear equations or inequalities (Kanellakis et al., 1995). The indefinite constraint data model (Koubarakis, 1994) extends the original constraint model to handle indefinite information by defining *possible worlds* as linear or polynomial constraint relations that use variables to handle vague or imprecise values. An extension of the linear constraint model uses differential geometry (Su et al., 2001). Within such a approach, simple primitives of velocity and acceleration along with vector operators are sufficient to express relations about movements, and the topological and temporal relations among objects. Finally, another extension to the original constraint model treats spatio-temporal data as (possibly infinite) sets of points in a multidimensional space of rational numbers, with no limitation of the space dimension (Grumbach et al., 2003). These sets are then used as values in tuples.

Access Methods

Traditional access methods do not support spatio-temporal data; therefore, different proposals have been developed to simultaneously support time and space. They aim at indexing objects that move in a two-dimensional space (Ghanem & Aref, 2003). Most of these methods extend spatial access methods to include temporal components. These methods may be classified based on the type of data about moving objects that they deal with. Some methods focus on the retrieval of historical data with queries defined by *timeslices* or *intervals*. Examples are RT-Tree (Xu et al., 1990), HR-Tree (Nascimiento et al., 1999), and their variations. A second type of access methods focuses on the trajectory of moving objects, such as, TB-Tree and STR-Tree (Pfoser et al., 2000). Finally, some methods are used to retrieve future positions of moving objects based on the current positions and movement patterns (Agarwal et al., 2000, Kollios et al.,1999).

A different classification of access methods considers how time is included in the spatial structure. Considering time as another dimension, the 3D-Rtree (Theodoridis et al., 1996) treats spatio-temporal data as 3-dimensional structures, such that, the spatial locations are depicted, say, on the xy-plane, and the z-axis is the time dimension. Another strategy incorporates temporal data in the nodes of the spatial structure, such as the case of the RT-Tree (Xu et al., 1990). Other approaches take into account the overlapping of common data across time, thus reducing unnecessary duplication of data. This is the type of approach used by HR-Tree (Nascimiento et al., 1999) and OLQ (Tzouramanis et al., 2000).

In addition to the general spatio-temporal indexing methods, some access methods have been designed taking into consideration specific requirements of the applications where they are used (Frentzos, 2003, Pfoser et. al, 2001). Such special requirements may include movements within partitions of the space, movements at different speeds, objects as points regardless their shapes, and dynamic indexing when new objects are included in the system.

FUTURE TRENDS

Moving objects databases is an important area of database research that has become especially active since the nineties. Although there have been advances in the different aspects involved in the design of moving objects databases, there are still many challenges that current solutions have either only partially addressed or have not addressed at all. These challenges are summarized in Table 1.

Table 1. Challenges for moving objects applications

Uncertainty
Moving objects applications should include aspects of uncertainty and quality of data, as well, as more semantics through constraints.

Discrete observations versus continuous movement
Data comes as discrete observations of a continuous movement. This is still a challenging issue for moving objects applications that need to define the frequency of observations.

Lack of standardization
There are still a large number of unstandardized features being used to model spatio-temporal data. In this context, creators of spatio-temporal models should consider how their models may be integrated with ISO TC 211, GML (The Geographic Markup Language) o related developments.

Data integration
Like traditional databases, data integration in moving-objects

applications presents issues at the semantic, logical and physical levels of a database.

Granularity
Not only spatial data, but also temporal data can be viewed at multiple granularities. Such granularities offer multiple representations of moving objects data, which have an impact on the consistency and the integration of databases.

Consistency
Management of consistency is usually accomplished by defining integrity constraints. There are no studies that address deeply enough the specification and modeling of spatio-temporal constraints (rules).

Query processing
Query processing needs to take into account more than indexes. In particular, it also needs to consider optimizers and join algorithms.

Performance evaluation
More research is necessary to verify the performance of existing or new indexing methods in realistic scenarios.

Distributed systems
Centralized databases may be impractical, in which case a distributed solution would be needed. Such distributed solution requires a system to allocate, update and query moving objects or trajectories in distributed data repositories.

Moving-objects data mining
The mining of motion patterns means the detection of periodicity in the movements of objects. Such periodicity may be cyclic (e.g., monthly, diary, and son on) or not, which is useful in the prediction of motion.

Aggregation and visualization
Spatio-temporal OLAP involves the study of aggregation functions, as well as, the visualization of multidimensional data.

User interface
Although there exist several approaches to visual query languages for spatial databases, they all allow querying of only static spatial situations. Querying and visualization of object changes and changes of objects' relationships are the goal for a spatio-temporal language.

C O N C L U S I O N

Moving objects databases are a challenging area for further research. Emerging commercial location-based services that make use of devices such as smart cell phones, wireless modems and GPS devices, whose prices are dropping rapidly, suggest that the use of this technology will spread worldwide. The issues discussed in this paper about spatio-temporal concepts, conceptual modeling, data modeling, and access methods emphasize the current state of the start of moving objects databases. Given that moving objects databases are still a young technology, there is a need for further studies that address not only the already-covered aspects of modeling and indexing structures, but also new considerations about data mining, warehousing, query processing, consistency, data integration, and user interfaces.

R E F E R E N C E S

Agarwal, P., Arge, L. & Erickson, J. (2000). Indexing moving points. *Symposium on Principles of Database Systems* (pp. 175-186). ACM Press.

Di Pasquale, A., Forlizzi, L., Jensen, C., Manolopoulos, Y., Nardelli, E., Pfoser, D., et al. (2003). Access method and query processing techniques. In K. Koukarakis, R. Sellis, A. Frank, S. Grumbach, R. Güting, C. Jensen, N., et al., (Eds.), *Spatio-Temporal Databases: The Chorochronos Approach,* LNCS 2520. (pp. 203-262). Berlin: Springer-Verlag.

Frank, A. (2003). Ontology for spatio-temporal databases. In K. Koukarakis, R. Sellis, A. Frank, S. Grumbach, R. Güting, C. Jensen, N., et al., (Eds.), *Spatio-Temporal Databases: The Chorochronos Approach,* LNCS 2520. (pp. 9-78). Berlin: Springer-Verlag.

Forlizze, L., Güting, H., Nardelli, E. & Schneider, M. (2000). A data model and data structure for moving objects databases. *ACM SIGMOD* (pp. 319-330). ACM Press.

Fretzos, E. (2003). Indexing objects moving on fixed networks. In T. Hadzilacos, Y. Manolopoulos, J. Roddick & Y. Theodoridis, (Eds.), *Advances in Spatial and Temporal Databases,* LNCS 2750. (pp. 289-305). Berlin: Springer-Verlag.

Grumbach, S., Koubarakis, M., Rigaux, P., Scholl, M. & Skiadopoulos, S. (2003). Spatio-temporal models and languages: An approach based on constraints. In K. Koukarakis, R. Sellis, A. Frank, S. Grumbach, R. Güting, C. Jensen, N., et al., (Eds.), *Spatio-Temporal Databases: The Chorochronos Approach,* LNCS 2520. (pp. 177-201). Berlin: Springer-Verlag.

Güting,R., Böhlen, M., Erwing, M., Jensen, C., Lorentzos, N., Nardelli, E., et al. (2003). Spatio-temporal models and languages: An approach based on data types. In K. Koukarakis, R. Sellis, A. Frank, S. Grumbach, R. Güting, C. Jensen, N., et al., (Eds.), *Spatio-Temporal Databases: The Chorochronos Approach,* LNCS 2520. (pp. 117-176). Berlin: Springer-Verlag.

Güting, R., Böhlen, M., Erwing, M., Jensen, C., Lorentzos, N., Schneider, M., et al. (2000). A foundation for representing and querying moving objects. *ACM Transactions on Database Systems*, *25*, 1-42.

Kanellakis, P., Kuper, G. and Revesz, P. (1990). Constraint query languages, *ACM Symposium on Principles of Database Systems* (pp.299-313). ACM Press.

Kollios, G., Gunopulos, D. and Tsotras, V. (1999). On Indexing Mobile Objects, *ACM Symposium on Principles of Database Systems* (pp. 261-272). ACM Press.

Koubarakis, M. (1994). Database models for infinite and indefinite temporal information. *Information Systems, 19,* 141-173.

Mokbel, M.F. & Aref, W.G. (2003). Spatiotemporal access methods. *IEEE Data Engineering Bulletin, 26*, 40-49.

Nascimiento, M., Jefferson, R., Silva, J. & Theodoridis, Y. (1999). Evaluation of access structures for discretely moving points. *Workshop on Spatiotemporal Database Management* (pp. 171-188).

Parent, C., Spaccapietra, S. & Zimányi, E. (1999). Spatiotemporal conceptual models: data structure + space + time. In C. Medeiros, *ACM GIS* (pp.26-33). ACM Press.

Pfoser, D. & Tryfona, N. (2001). Requirements, definitions and notations for spatiotemporal aplication environments. *ACM GIS* (pp. 124-130). ACM Press.

Pfoser, D., Jensen, C. & Theodoridis, T. (2000). Novel approaches in query processing for moving object trajectories, *VLDB Journal*, 395-406.

Price, R., Tryfona, N. & Jensen, C. (2000). Extended spatiotemporal UML: Motivations, requirements and constructs. *Journal of Database Management*, *11*, 14-27.

Shekhar, S., Coyle, M., Goyal, B., Liu, D.-R. & Sakar, S. (1997). Data models in geographic information systems. *Communications ACM*, *40*, 103-111.

Su, J., Xu, H. & Ibarra, O. (2001). Moving objects: logical relationships and queries. In C. Jensen, M. Schneider & J.T. Vassilis (Eds.), *Advances in Spatial and Temporal Databases,* LNCS 2121. (pp. 3-19). Berlin: Springer-Verlag.

Theodoridis, Y., Vazirgiannis, M. and Sellis, T. (1996). Spatiotemporal indexing for large multimedia applications. *International Conference on Multimedia Computing and Systems* (pp. 441-448).

Tryfona, N, Price, R. & Jensen, C. (2003). Conceptual models for spatio-temporal applications. In K. Koukarakis, R. Sellis, A. Frank, S. Grumbach, R. Güting, C. Jensen, N., et al., (Eds.), *Spatio-Temporal Databases: The Chorochronos Approach,* LNCS 2520. (pp. 79-116). Berlin: Springer-Verlag.

Tryfona, N. & Jensen, C. (1999). Conceptual data modeling for spatiotemporal applications. *GeoInformatica, 3,* 245-268.

Tzouramanis, T., Vassilakopoulos, M. & Manolopoulos, Y. (2000). Overlapping linear quadtrees and apatiotemporal query processing. *The Computer Journal, 43,* 325-343.

Xu, X., Han J. & Lu, W. (1990). RT-Tree: An improved R-Tree index structure for spatiotemporal database. *4th International Symposium on Spatial Data Handling* (pp. 1040-1049).

Wolfson, O. (2002). Moving objects information management: The database challenge. *5th Workshop on Next Generation Information Technology and Systems* (pp. 75-89). Berlin: Springer-Verlag.

Terms and Definitions

Valid Time: The time when a fact (i.e., a statement with an associated truth value) is true in the modeled reality.

Transaction Time: The time when an element (i.e., anything that may be stored in a database) is part of the current state of the database. It can be seen as the valid time of an element that is currently in the database, that is, valid time of "element *e* is current in the database."

Existence Time: The time when an object exists in the reality. It can be seen as the valid time of the existence of an object, that is, the valid time of "object *o* exists."

Interval-based Model of Time: temporality is specified using regular or irregular intervals or periods, which are durative temporal references.

Point-based Model of Time: temporality is specified using explicit occurrences of an event, observation or action, which are punctual occurrences

Timeslice or Snapshot Query: Refers to a query that asks for data as of a given transaction time. For example, find all objects alive during a time interval.

Bitemporal Query: Refers to a query that involves both valid and transaction time points or intervals.

Spatio-temporal Selection Query: It refers to a query that asks for data that intersect a spatial and temporal window.

Spatio-temporal Join Query: It refers to a query that asks for data that satisfy a spatial relation and a temporal window. For example, find all cars within 10 miles of each other during a given hour (or at 1 pm.).