Moving Objects Databases

M. Andrea Rodríguez-Tastets
University of Concepción and 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 1990s that spatio-temporal databases became an area of active research. The evolution of spatio-temporal databases, 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, Price & Jensen, 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, 1998). From a temporal perspective, properties and relationships are considered facts of objects; 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 associ-
Moving Objects Databases

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 the 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, Tryfona & Jensen, 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 et al., 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 vary 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, Spaccapietra & Zimányi, 1999). In this approach, an object-based model is extended with predefined 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 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) (Forlizze 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
Related Content

Disclosure Control of Confidential Data by Applying Pac Learning Theory
[www.igi-global.com/article/disclosure-control-confidential-data-applying/47422?camid=4v1a](www.igi-global.com/article/disclosure-control-confidential-data-applying/47422?camid=4v1a)

Politically Oriented Database Applications
[www.igi-global.com/chapter/politically-oriented-database-applications/20706?camid=4v1a](www.igi-global.com/chapter/politically-oriented-database-applications/20706?camid=4v1a)

Comparing Object-Oriented and Extended-Entity-Relationship Data Models
[www.igi-global.com/article/comparing-object-oriented-extended-entity/51151?camid=4v1a](www.igi-global.com/article/comparing-object-oriented-extended-entity/51151?camid=4v1a)

Adoption, Improvement, and Disruption: Predicting the Impact of Open Source Applications in Enterprise Software Markets
Michael Brydon and Aidan R. Vining  (2008). *Journal of Database Management* (pp. 73-94).
[www.igi-global.com/article/adoption-improvement-disruption/3386?camid=4v1a](www.igi-global.com/article/adoption-improvement-disruption/3386?camid=4v1a)