Adapting Temporal Aspects to Static GIS Models

K.C.S. Murti and Viksit Agarwal

Birla Inst. of Technology & Science, Pilani,
Hyderabad Campus
Jawahar nagar, Hyderabad, 500078, India
{chandra.kavuri,viksit.agarwal}@gmail.com

Abstract. Conventional GIS data models emphasize static representation of real world Geographic features. There are several real world problems where this assumption is not valid. For example the boundary of a lake changes with time depending on the inflow due to rains. The feature's geometry, its attributes and the topology with respect to its adjacent features are temporal. Current models used in commercial GIS do not support efficient persistence of the temporal history, provide temporal queries and deal with time variant topological changes. Study of current spatio-temporal models has been done and suitable techniques to enhance current spatial models with temporal aspects without breaking existing functionality has been explored. Moving object data model and its variations have been found to be an appropriate candidate for such enhancement. This paper describes the extension of spatial data model and its schema of a commercial GIS tool for supporting moving object data and creation of typical temporal analysis and query commands.

Keywords: Temporal Geographical Information System (TGIS), Spatio-Temporal data, Topology, Spatio-Temporal Models (STM), Moving Object Model.

1 Introduction

Geographic Information System (GIS) is a computer system capable of assembling, storing, manipulating, and displaying geographically referenced information, i.e. data identified according to their locations. GIS represents graphic reality i.e. location, geometry; links attribute data to the objects and builds spatial relationship among objects. GIS has made significant impacts on the ways we handle spatial data. These impacts are reflected not only by the wide range of GIS applications in various fields but also through the development of new data representation methods and analysis capabilities available to the GIS users. GIS still needs robust models to solve real world problems . These models have to handle geographic complexity, temporal aspects, scale differences, generalization, and accuracy in a robust way.

Most of commercial GIS tools handle the real world geographic features in a static way. In fact, no geographic feature is static. Either they will be fast moving or slow moving. So every feature's geometry, its attributes and the topology with respect to its adjacent features is temporal. The temporal granularity can be discrete or

N. Meghanathan et al. (Eds.): CCSIT 2012, Part II, LNICST 85, pp. 424–434, 2012.

continuous. The real world objects can be broadly classified as continuously changing objects (Moving vehicles..), objects that are basically static, but they are changed by events that have duration (Lake boundaries..) and objects that are always static and change only by sudden events (land parcels..) Current models used in GIS do not support efficient persistence of the temporal history, provide temporal queries and deal with time variant topological changes. Hence, spatio-temporal models come into play.

An excellent review of Spatio-Temporal database models is presented by Pelekis et all [1] with typical applications in LIS. The paper summarizes the capabilities of different models and considers Moving Object model as a promising candidate for robust implementation. Geographic Markup Language (GML) is published by OGC (Open GIS Consortium) for representation of geographic data and is a standard for exchange of spatial data. The standard has evolved further to represent Spatio-Temporal supporting (i.e) Moving Object models in later versions.[6] GML now supports Dynamic Features to support temporal information to some extent. Heavy software base is developed by commercial vendors supporting static GIS models. Rudimentary changes cannot be done for temporal support in commercial packages, except to adapt STM in a loosely coupled way. This paper discusses a possible strategy in this direction.

2 Characteristics Spatio-Temporal Data Models

Spatio-Temporal data models are the core of a Spatio-Temporal Information System (STIS); they define object data types, relationships, operations and rules to maintain database integrity. A rigorous data model must anticipate spatio-temporal queries and analytical methods to be performed in the STIS. Spatio-temporal database models are intended to deal with real world applications, where spatial changes occur over the time line. A serious weakness of existing models is that each of them deals with few common characteristics found across a number of applications. Thus the applicability of the model to different cases fails on spatio-temporal behaviors not anticipated by the application used for the initial model development.

The study of the literature of the domain highlighted a set of precise characteristics of existing models that stand for the requirements of spatio-temporal database community. These requirements fall in four categories.

The first category deals just with the nature of time including the basic features that are used to describe it. The second category handles the pure spatial aspects of the existing approaches. The third deals with the unified spatio-temporal semantics, while the last category considers the query capabilities of the models. If these requirements are followed carefully in the process of designing a spatio-temporal database model, a robust and expandable model can be achieved, capable of dealing with most of the real world spatio-temporal processes (Pelekis et. al. [1]).

2.1 Temporal Semantics

Granularity: Granularity is specified by an anchored point on the time axis and a partitioning length. The anchored point denotes where the partitioning begins while the partitioning length denotes the size of each granule. Different applications require different levels of granularity.

Temporal operations: A series of specific operations describing temporal relationships have been proposed and proved necessary in handing any time-referenced information (e.g. timepoint T "inside" temporal period A which "meets" period B).

Time density: Time density is closely related to the types of changes/events that can occur to the value of a thematic or spatial characteristic. This issue arises whether time should be modeled as discrete or as continuous elements.

Representation of time: Time is represented by timestamps, where representation methods are different for each model. This criterion allows us to compare each modeling technique, by whether maintaining the duration of the status of an object or recording events that imply status change.

Lifespan: This factor shows if a model supports and deals with the duration of an event. This also concerns whether a model keeps track of the history of the real world objects, in terms of storing the lifespan of a discrete phenomenon or the temporal differences for a continuous one.

2.2 Spatial Semantics

Structure of space: This criterion represents the two basic approaches for computer storage of geographic data, which are the raster and vector spatial data models. Raster data are structured as an array of cells, pixels or voxels for 2D or 3D representations respectively. Vector techniques describe each spatial object in terms of start and end points.

Orientation/Direction: This standard demonstrates whether a model supports the orientation and the direction features that real world objects show in space (e.g. on the left side of, to the right).

Measurement: This issue examines whether it is possible to get a value of a spatial object (e.g. length, perimeter, distance etc.) using a particular model or if a model supports comparative operations such as bigger, longer.

Topology: This criteria establishes whether the model supports different topological relationships for the real world spatial objects.

2.3 Spatio-Temporal Semantics

Data types: Examples of spatial data types are the point, line and region whereas temporal point and interval are samples of temporal data types. Moving point and moving region are characteristic cases of unified spatio-temporal data types.

Type of change: This norm compares the models if they are able to deal with changes in shape and size of the objects. Models are also evaluated whether a change in the description of a spatio-temporal object can be combined with a synchronous representation of the change of an object's position. This norm further considers whether a model supports spatio-temporal real world objects that change continuously or just objects that are subject to discrete changes. Another criterion that further categorizes existing approaches that follow the continuous paradigm is whether the latter can deal with the movement of the spatial objects over time.

Evolution in time & space: This factor shows if there are defined functions like evolution, creation, fusion etc. to observe and describe the movement or change of objects in space, independently from their object identification. The norm is also applied to compare models on the existence of operations able to calculate the velocity and/or the acceleration of the movement of spatio-temporal objects.

Space-time Topology: This criterion sets a standard whether models can estimate metrics like values of distance, direction and change in size of a particular object. It further evaluates the ability of the models to represent topological relationships between (in particular continuously) evolving spatial objects for a certain period of time.

Object identities: Another issue that can be employed to evaluate the modeling ability of existing spatio-temporal data models is the manipulation of the identity of an object. In particular, the lifespan of an object is an important application dependant variable. The question is when does "change" affect an object so as not to be called the same object any more? Some times it may be more appropriate to destroy the original instance of an object and re-create a new one, due to an extensive change. Another critical issue is that of splitting or unifying objects.

Dimensionality: With this criterion, models are examined whether they support 2 dimensions to model the spatio-temporal objects, as traditional GIS do. Although two and half dimensional solutions exist (perspectives, stereo views etc.), volumetric 3 dimensional GIS provide advantages in displaying spatio-temporal data. In more recent approaches, relegating the attribute value associated with grid locations to a fourth dimension, time can be introduced as a fifth.

2.4 Query Capabilities

Existing spatio-temporal database models can be characterized based on their query capabilities:

Static spatial Queries : Queries about locations, spatial properties, and spatial relationships. The queries are on stationary spatial objects involving the location, topology and the attributes of the object.

Temporal properties and relationships: The queries are temporal value, time range and temporal relationships. (ex: What is the location of object at time t? What is the region in which the object moved during this period? What are the objects which are within 1 Km range at time *t*? etc.

Spatio-temporal behaviors and relationships: This set of queries involve space and time on discretely or continuously changing objects. (ex: Find the location where the train with this ID is at time *t*. It can be spatio-temporal range queries (ex. Find all the goods trains around the city moving this night?)

3 Moving Object Data Models

When we try integration of space and time, we are dealing with geometries changing over time. In general, geometries cannot only change in discrete steps, but continuously, and then we are talking about moving objects. If only the position in space of an object is relevant, then moving point is a basic abstraction; if the extent is also of interest, then the moving region abstraction captures moving as well growing or shrinking regions.

Researchers Erwig et.al. [2], have tried to model such spatio-temporal databases using this concept of moving object. While there is extensive literature on the evolution of Spatio-temporal models, let us restrict to moving object models, which is the appropriate model planned for implementation. Moving points and moving regions are viewed as three-dimensional (2D + time) or higher dimensional entities whose structure and behavior is captured by modeling them as abstract data types. The objective is to integrate such abstract data types into any extensible DBMS. This approach models the time as integral part of the spatial entities. The time dimension is based on the linear, discrete/continuous, absolute time model and initially only valid time is considered. The model captures both change and movement ex: The trajectory of a moving point can be described either as a curve, or as a polygonal line in 2D space.

3.1 ADTs for Moving Objects

Base types: The base types are int, real, string, and bool. All base types have the usual interpretation, except that each domain is extended by the value “undefined”.

Spatial Types: Basic are point, line, and region. A value of type point represents a point in the Euclidean plane or is undefined.. A line value is a finite set of continuous curves in the plane. A region is a finite set of disjoint parts called faces, each of which may have holes. The point set paradigm expresses that space is composed of infinitely many points.

Time Type: Type “instant” represents a point in time or is undefined. Time is considered to be linear and continuous, i.e., isomorphic to the real numbers.

Moving Types: From the base types and spatial types, we derive corresponding temporal types. The type constructor moving is used for this purpose. For all “moving” types we introduce extra names by prefixing the argument type with an “m”, that is, mpoint, mpoints, mline, mregion etc. The temporal types obtained through the moving type constructor are functions, or infinite sets of pairs.

3.2 Temporal Operations

Relevant operations [3] can be defined on temporal objects based on context and application. Such an operation may return a spatial or temporal or moving types.

Some examples are illustrated here with. If a moving point (or point set) changes its position only in discrete steps, then operation_locations returns its projection as a points value. Operation_routes similarly returns the projection of a discretely moving line value. Some practical examples of moving objects is given below

_ Land_Parcel (name: string, owner: string, area: mregion)
_ Lake (name: string, area: mregion)
_ Road (name: string, route: mline)
_ Tree (name: string, centre: mpoint)

Examples Typical Operations of Moving Objects

_Operation (mregion-> mreal) area applied to a time-varying real number representing the size of the lake at all times
_Operation (mregion x mpoint -> mboolean) inside applied against a land_parcel and a tree computes a time-varying boolean representing when the tree has been moved or planted in a parcel
_ Operation (mpoint X mpoint -> mreal) distance calculates the time-varying distance between two trees at all times.

4 Methodology and Implementation

Review of current methodologies [4] and models was done keeping the suitability and adaptability of the methods for implementation. Moving Object Data model is found to be most suited because: This model is close to reality and abstraction level is much higher and supports representation of continuous changes in geometries. The representation is data-type oriented, with emphasis on generality, closure and consistency. It models the time as integral part of the spatial entities. It can be integrated into any extensible DBMS. There is redundancy in storage of temporal data.

4.1 Overview of GeoMedia® Data Model

GeoMedia® Professional [5] is the next generation Geographic Information System (GIS) product. GeoMedia is an enterprise GIS tool for capturing Geo referenced data, creating an enterprise database as warehouses, providing spatial queries, spatial manipulations and publishing maps. This product has unique feature which enables to view geographic data from different sources, in a unified way irrespective of the source formats. Users can do spatial analysis from heterogenous sources seamlessly in a single environment.

Geographic Data Objects (GDO) are programmable database objects based on the Microsoft Data Access Object (DAO) model (Fig. 1). The Microsoft Component Object Model (COM) design provides the automation for data access and data update. GeoMedia® Professional combines DAO concepts and COM with geographic aspects of geodetic coordinate systems, geometry, and graphics display.

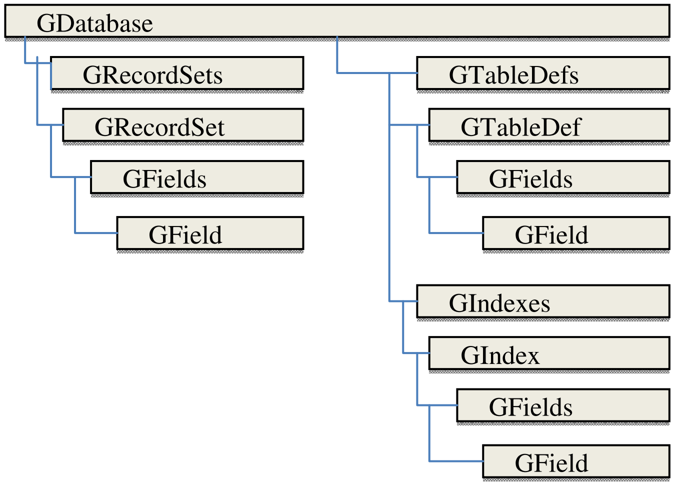

Fig. 1. Geographic data Objects (GDO) of GeoMedia©

Expansions to the DAO model allow database definition, population, and update through the GDO model. These expansions consist of the following.

Geometry fields are defined to store geometric information. Standard Tables are defined to aid in manipulating and updating geographic information. Coordinate Systems Table provides coordinate system information used to project geometry. Metadata is table data about the GDO structure and contents. It is used by automation and built-in GeoMedia® Professional commands to return supplementary information for feature geometry and attributes. For example, it returns the name of the primary geometry field, a list of all feature tables, and so forth.

GeoMedia® Professional talks to the various data sources using GDO. Intermediate components called as GDO data servers read the native data and serve data to GeoMedia® Professional client using GDO. Each data server is specific to a native data type. Information about and manipulation of a GDatabase object is performed through its methods and properties. The GTableDef object is the stored definition of a database table. GTableDef that has specific metadata defined are recognized as feature classes in GeoMedia® Professional. GField represents a column of data similar to DAO but extended to support spatial data types.. Each feature class is designated a primary geometry type (Point, Line, Area, Compound, Text or None). Geometry data is stored in geometry GFields in GTableDef in the form of GDO blobs.

4.2 Temporal Schema Design

The static point feature class resides in GeoMedia® static warehouse. In reality, the temporal information about the moving point will be an input in the form of stream of data samples (t1, x1, y1), (t2, x2, y2), … and so on.

The source of this temporal data can be from GPS. To simulate this behavior in the present work, a linear feature has been used to define the trajectory of moving point that is moving with a uniform speed. The linear feature class that is used to input the trajectory of a moving point resides in a GeoMedia® warehouse static data model. In this table, each moving point feature is uniquely identified by a primary key "ID". A new metadata table is created that stores the temporal table name corresponding to this linear feature class. In the proto-type implementation, since there is no relevance of presence of static point feature class table, the metadata table stores the mapping of input linear feature class table name and the temporal data table. But in reality the metadata table should store the temporal table name against the static feature class table name. In the temporal table, the "ID" field of the linear feature class is made a foreign key that establishes a relationship between the two tables. Once the temporal data is generated it has no relation with the static table. In other words, all temporal analysis will be done using only the temporal data table.

The metadata table is used for book-keeping of all the temporal data tables present in the warehouse. Also, if a temporal table needs to be updated or refreshed, the mapping will be used by the Create Temporal Data command to show the already existing temporal data table name.

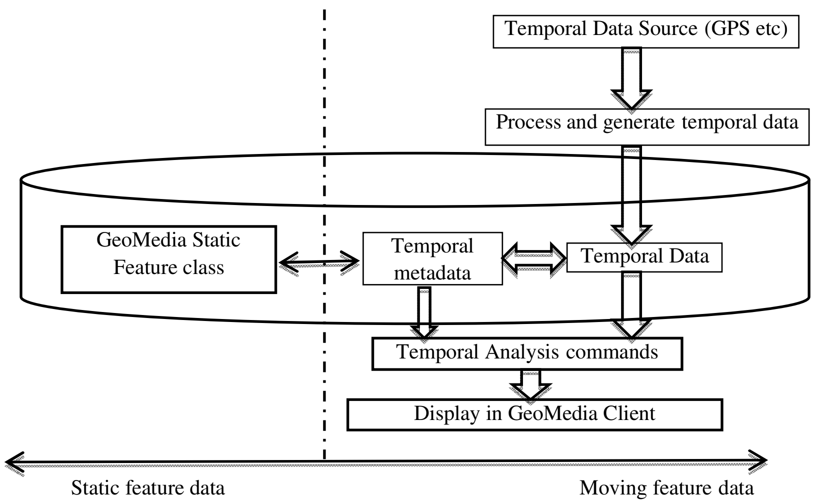

Fig. 2. Data Flow and relation of static and temporal data

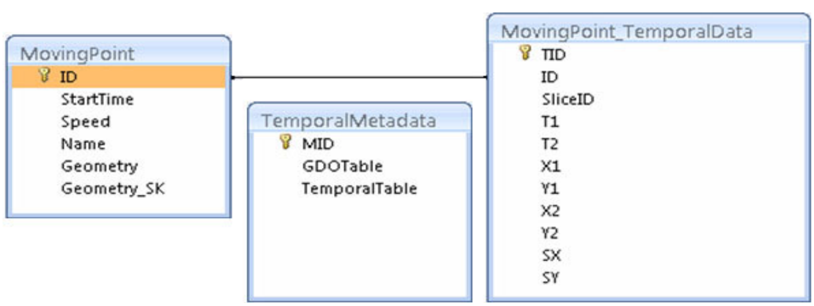

Fig. 3. Relationship among temporal data tables in GDO

4.3 Temporal Data Creation

Data input is done using Insert Feature command of GeoMedia® Professional in the form of linear feature class that represents the trajectory of the moving point. At the time of digitization of the linear feature, the uniform speed and start time of the moving point are also input as the attribute values. A command has been developed to perform the rest of processing and prepare the temporal data for further analysis. The command takes the following input:

- Input linear feature class that represents the trajectory of moving point.
- Read write connection name that contains the linear feature class and will be used to create temporal data and metadata tables.
- Name of the temporal data table to be created for a moving point.

The following flow chart explains the processing and data flow during the data input and preparation step:

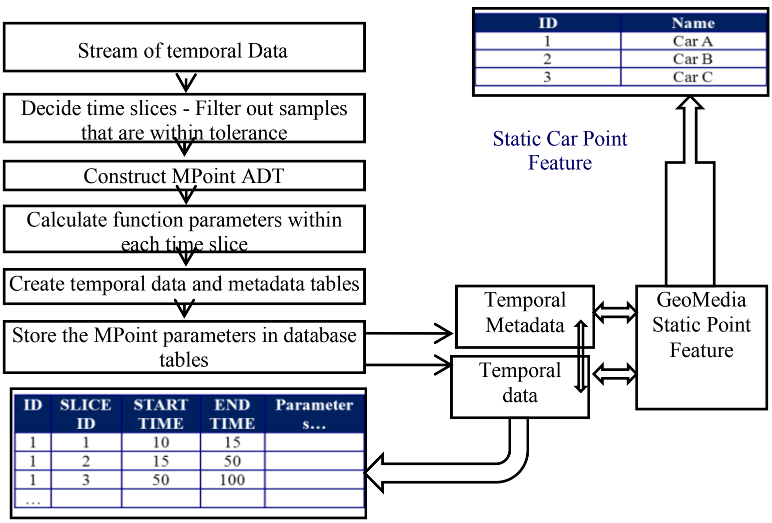

Fig. 4. Create Temporal Data command processing flow

4.4 Results

Two temporal analysis commands – "Display Moving Points" and "Profile Moving Point" have been developed. Display Moving Points command creates the point geometries at a given instant and displays them in map window. The below flow chart explains the processing of the command:

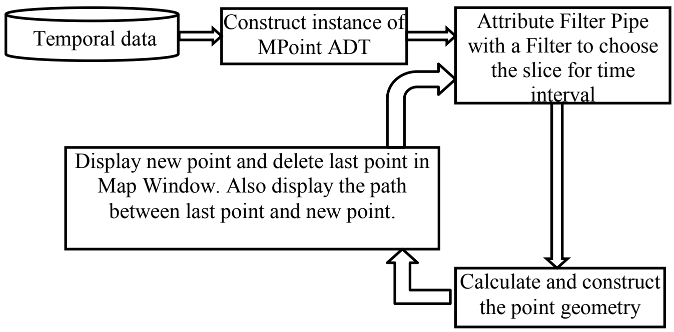

Fig. 5. Profile moving point processing

Profile moving parts command creates point geometries at a given regular interval of time, and displays them in map window at refresh intervals. When it adds the new point, it deletes the old point and draws a path between the two points; this makes the point appear moving in the GeoMedia® Professional map window. The road network considered in this example is a real data of Interstates (highways connecting different states) within Alabama, USA. On each of the interstates, one car is moving at pre-defined randomly selected speeds forming a trajectory.

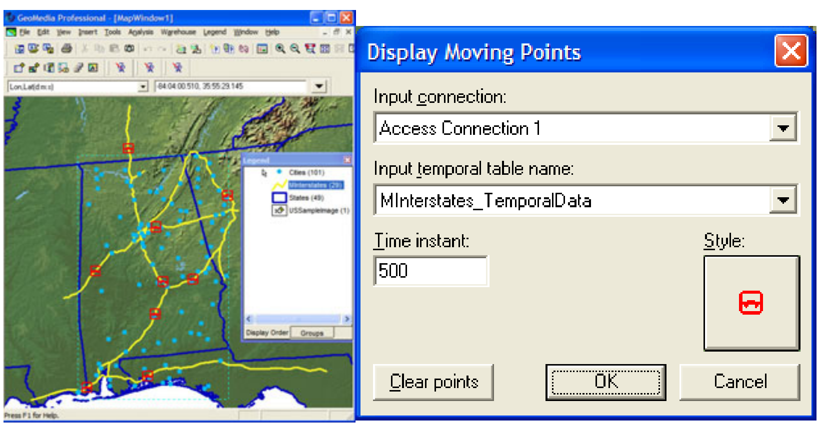

Fig. 6. Display moving points screen and command parameters

5 Conclusions

Moving Object Data model requires very less storage space to store temporal data compared to other models. The model requires only the function parameters to be stored with in each slice that describes the motion of the moving object. The model can be incorporated in commercial GIS products without affecting its existing static data and functionality. Implementations does not have dependency on any specific DBMS. Temporal queries can be handled similar to any spatial queries and persisted.

References

1. Pelekis, N., Theodoulidis, B., Kopanakis, I., Theodoridis, Y.: Literature Review of Spatio-Temporal Database Models. Knowledge Engineering Review 19(3), 235–274 (2004)
2. Erwig, M., Guting, R.H., Schneider, M., Vazirgiannis, M.: Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. GeoInformatica 3(3), 265–291 (1999)
3. Guting, R.H., Bohlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M.: A Foundation for Representing and Querying Moving Objects. ACM Transactions on Database Systems 25(1), 1–42 (2000)
4. Langran, G., Chrisman, N.R.: A Framework for Temporal Geographic Information Systems. Cartographica 25(3), 1–14 (1988)
5. GeoMedia® Professional developers' and users' help documents
6. TGML - Extending GML by Temporal Constructs - A Proposal for a Spatiotemporal Framework in XML