TOWARDS AN ALGEBRA FOR SPATIO-TEMPORAL DATABASE

Karine Reis Ferreira

Qualificação do Curso de Doutorado em Computação Aplicada.

INPE
São José dos Campos
2009

Summary

1 INTRODUCTION

The recent technological advances in geospatial data collection, such as Earth observation and GPS satellites, wireless and mobile computing, radio-frequency identification (RFIDs) and sensor networks, have motivated new types of applications which handle spatio-temporal information. Examples include recording of animal tracking, transport systems, oil slicks on the ocean, and tracking land change objects. To satisfy this demand, there has been research on how to represent spatio-temporal information in geographical information systems (GIS). According to Worboys (2005), there are four stages in introducing temporal capacity into GIS: (0) static GIS, (1) temporal snapshots, (2) object change, and (3) events, actions and processes. Most current proprietary technologies are in stage zero, that is, they do not deal with spatio-temporal information. This is partly due to the lack of consensus on how to represent spatio-temporal information in computational systems.

Static geospatial information is represented in GIS following well-established ideas. These ideas include object-based and field-based models, vector and raster data structures, topological operatores, spatial indexing and spatial joins and operations (Couclelis, 1992) (Rigaux et al., 2002) (OGC, 2006). In recent years, database management systems (DBMS) have been extended to handle geospatial information. Examples include Oracle Spatial (Oracle, 2003), DB2 Spatial Extender (IBM, 2006) and PostGIS (Refractions, 2008).

In the GIS literature, there are many proposals of spatio-temporal database models. Each of these models defines an ontology of space and time and its representation through data types, relationships and operations among them. However, according to Pelekis et al. (2004), a serious weakness of existing spatio-temporal database models is that each model focuses on certain types of applications. Therefore, the models are not general enough to be a basis for a spatio-temporal GIS.

To improve this, there is a need for a general-purpose spatio-temporal database model that can be used for a new generation of dynamic GIS. Thus, this work aims to provide a critical analysis of some spatio-temporal database models from the literature, and to define a first version of a spatio-temporal algebra for dynamic geographical processes. This review is presented in Chapter 2 and the algebra initial version in Chapter 3. Finally, a conclusion of this work is presented in Chapter 4.

2 SPATIO-TEMPORAL DATABASE MODELS

During the past two decades, many spatio-temporal database models have been proposed in the literature (Pelekis et al., 2004). Most of these define an ontology of space and time and its representation through data types, relationships and operations over them. In the next sections, some models, shown in Figure 2.1, are briefly described. Finally, the last section presents a critical analysis.

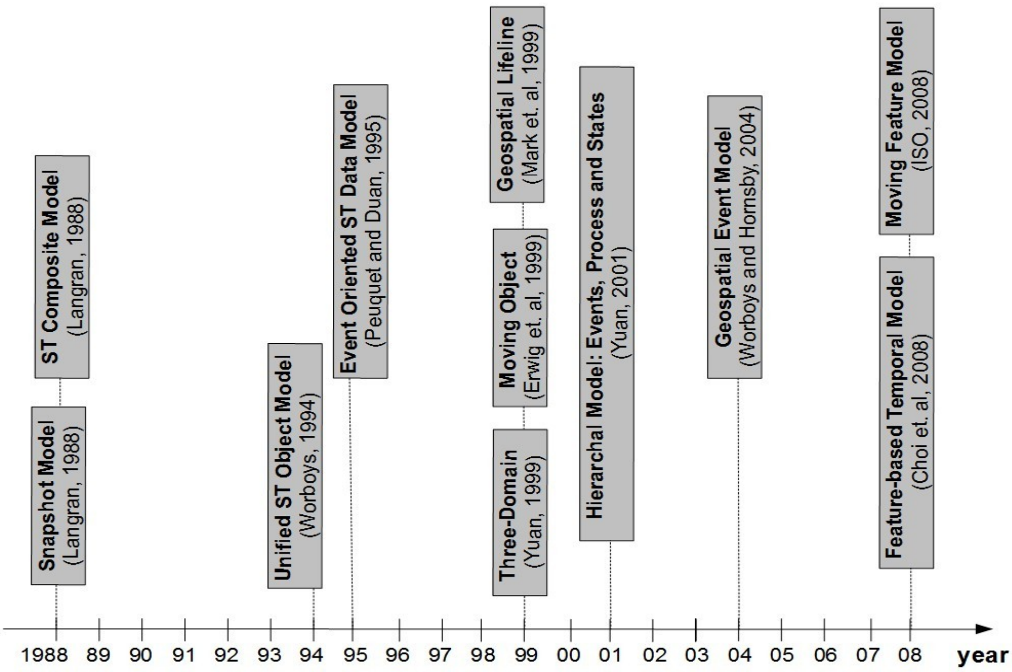


Figure 2.1 - Spatio-temporal database models.

2.1 Time-Slice Snapshot and Space-Time Composite Models

The Time-Slice Snapshot model is the simplest spatio-temporal model which includes temporal information into spatial data model by using time-slice snapshots (Langran, 1988). This model works with a set of snapshots, where each snapshot is a raster layer which represents a state of the real world at a given time, like a photograph. In short, in this model, each snapshot is a collection of temporally homogeneous units and there are no explicit temporal relations among snapshots.

As an evolution of the Snapshot Model, Langran (Langran, 1988) has proposed the Space-Time Composite (STC) model, by considering vector objects which change over time instead of raster time-slice layers. The mechanics of this model begin with a base layer which represents the objects at some starting time. After this, each change decomposes the space over time into increasingly smaller fragments (objects with geometries) with its own distinct history.

2.2 Unified Spatio-Temporal Object Model (STOM)

Worboys (1994 (a)) proposed the Unified Spatio-Temporal Object model (STOM) which merges the two-dimensional space and two-dimensional time of a geographical object in a spatial-bitemporal object. This model uses the concept of "geographical object" to represent real world entities. An object has spatial, graphical, temporal, and textual/numerical parts (Worboys, 1994 (b)).

A spatial-bitemporal object is a unified object which has both spatial and bitemporal extents. The spatial extent is represented as *simplicial complexes*, which themselves are made of non-overlapping *simplexes*. A *simplex* is either a single point, or a finite straight line segment or a triangular area. The bitemporal extent is represented by bitemporal elements (BTE) which are composed of event time and transaction time. Event time is when events occur in the real world.

Transaction time is when the database registers the event. Thus, a BTE is the union of a finite set of Cartesian products of event and transaction times.

To represent spatial-bitemporal objects, BTEs are attached as labels to *simplicial complexes*, originating two new data types called *ST-simplexes* and *ST-complexes*. ST-simplex is an ordered pair<*S*, *T*>, where *S* is a *simplex* and *T* is a BTE. A ST-complex is a finite set of ST-simplexes. Examples of these types are shown in Figure 2.2. Besides the data types, this model defines a set of spatio-temporal operations, such as ST-Union, ST-intersection, and ST-difference (Worboys, 1994 (a)).

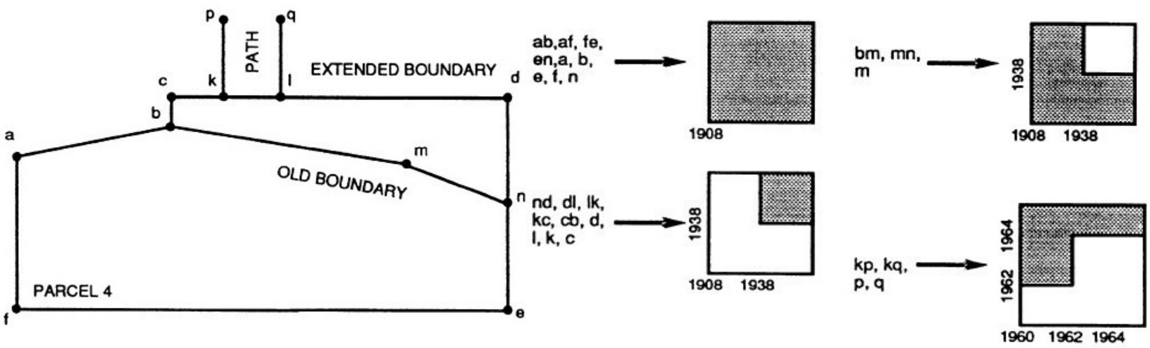


Figure 2.2 - A parcel boundary evolution represented as ST-complex.
Source: Adapted from Worboys (1994)

2.3 Event oriented Spatio-Temporal Data Model (ESTDM)

The ESTDM model takes a time-based approach, using time as its organizational basis (Peuquet and Duan, 1995). Its main idea is to group changes by time of occurrence, as shown in Figure 2.3 (a).

This model orders changes in locations within a prespecified geographical area. The time associated with each change, called event, is stored in increasing order from initial time t_0 to the latest time t_n. Thus, changes which occurred between t_{i-1} and t_i are associated with the time t_i. The only exception is the time t_0 which is the starting world state. The set of changes C_i recorded for any time t_i consists of the set of each location (*x, y*) which changed since t_{i-1}, and its new value *v,* as shown in Figure 2.3 (b). The two main characteristics of this model are: (1) the events are recorded when changes occur, that is, in any temporal resolution; (2) a value *v* is recorded only when it is different from the last one found along the scan line.

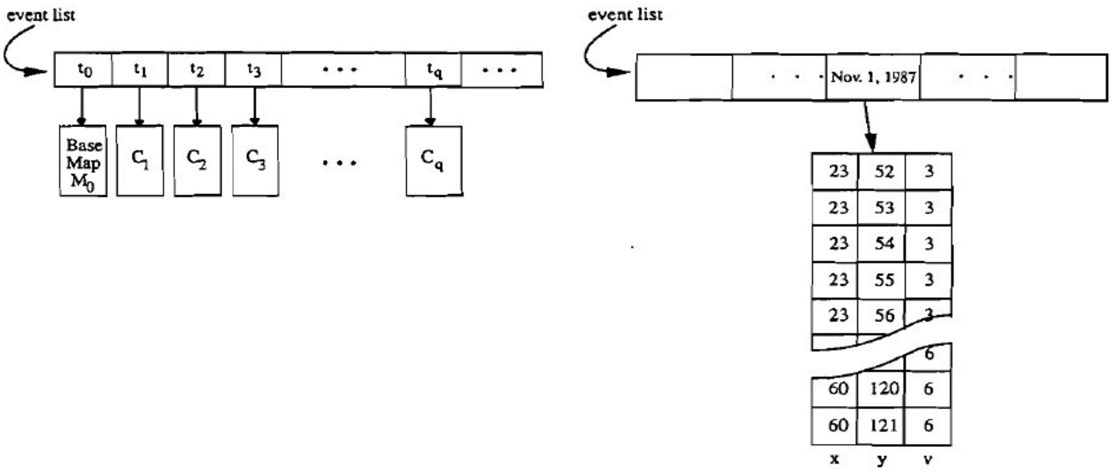


Figure 2.3 - (a) Representation of changes organized as a function of time. Here, t_i contains a time value (e.g. day, month and year) and C_i contains all changes that occurred at time t_i. (b) Detail of (a) showing the contents of C_i stored as individual x, y locations that changed and its new value *v*.

Source: Adapted from Peuquet and Duan (1995).

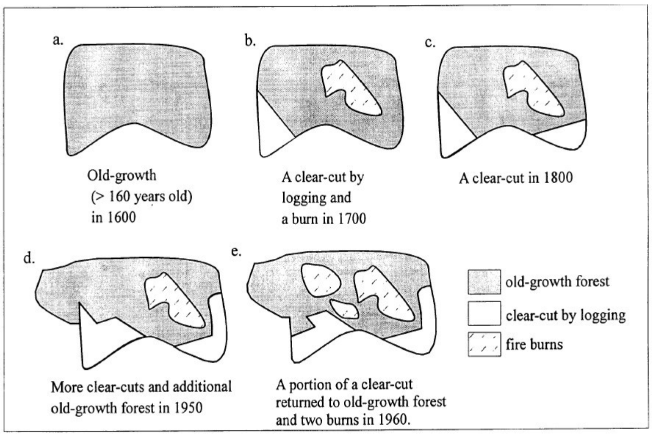
Peuquet and Duan (1995) describe algorithms for three fundamental temporal queries: (1) retrieving location(s) which changed to a given value at a given time; (2) retrieving location(s) which changed to a given value over a given temporal interval; and (3) calculate the total area which changed to a given value over a given temporal interval.

2.4 Three-Domain Representation Model

Yuan (1999) has proposed a three-domain model. It divides the spatio-temporal information in three different domains (semantic, temporal, and spatial) and links them. The semantic domain has objects representing categories, concepts, entities, events, and process. The temporal domain has objects representing time instants, intervals, temporal geometry, and temporal topology. And, finally, spatial domain consists of objects representing locations, spatial extent, spatial geometry, and spatial topology. Besides that, spatial domain includes a spatial graph to record transitions of spatial objects. In short, temporal and spatial objects represent the temporal and spatial properties of geographic semantics (including themes, entities, events, and processes).

Domain links represent associations among these objects. A link can be one of three types: (1) pointer that associate objects of different domains to represent spatio-temporal facts; (2) a function that indicate spatio-temporal behaviors; and (3) physical models that predict trends or processes in space and time.

Moreover, a database schema to represent this model has been proposed. It consists in four tables, one for each domain (semantic, temporal and spatial) and another for the domain link. An example of spatio-temporal transitions in a forest and the database schema to represent them is shown in Figure 2.4.



a. Semantics Table

Sem. ID	Landcover	Management	Address
1	Old Growth	USFS	12 Forest Rd.
2	Clear-cut	A. Log Co.	3 Clear Dr.
3	Burn	USFS	12 Forest Rd.
4	Clear-cut	B. Log Co.	45 Pine Ave.

b. Time Table

Time ID	Time	Operator ID
1	1600	2439
2	1700	2439
3	1800	7473
4	1950	1029
5	1960	1029

c. Space Table

Space ID	Area	Perimeters
4	A_1	P_1
6	A_2	P_2
8	A_3	P_3
9	A_4	P_4
10	A_5	P_5
11	A_6	P_6
12	A_7	P_7
13	A_8	P_8
14	A_9	P_9
15	A_{10}	P_{10}

d. Domain Link Table (Links among temporal, semantic, and spatial objects)

Sem. ID	Time ID	Space ID List
1	1	1
1	2	2
2	2	3
3	2	4
1	3	5
2	3	3, 6
1	4	7, 10
4	4	8, 9
1	5	10, 11, 13
2	5	6, 12
3	5	4, 14, 15

Figure 2.4 - An example of land-cover transitions in a forest area and its database schema, based on three-domain model.

Source: Adapted from Yuan (1999).

2.5 Moving Objects Model

Moving Objects refers to entities whose geometries change continuously over time, such as, cars, aircraft, ships, mobile phone users, polar bears, hurricanes, forest fires, or oil spills in the sea (Forlizzi et. al, 2000) (Guting et. al, 2003).

There are two different approaches leading to moving objects: *location management* perspective and *spatio-temporal data* perspective (Guting and Schneider, 2005). The first one goes about managing the positions of entities which are moving around right now. Thus, its main challenge is to deal with current and near-future movements. On the other hand, the last approach goes about modeling and querying histories of movements or evolution of spatial objects over time.

In order to support location management perspective, Guting and Schneider (2005) have proposed a model called MOST (Moving Objects Spatio-Temporal Model) to describe current and expected future movement. The basic idea of this model is to represent a moving object by its motion vector instead of its position directly. A motion vector consists in a position together with a speed and direction at a time. Besides that, the motion vector of each moving object is updated from time to time, without keeping the history of movement. Associated with this model, the authors define a query language called FTL (future temporal logic), which allows us to express queries about future movement, for example, “will truck T70 reach its destination within the next half hour?”.

Moreover, in order to represent and query histories of movements, Guting et. al (2003) have proposed an algebra which defines a set of data types and operations for moving objects. This algebra defines two main types to represent moving entities: *moving points* and *moving regions*. A moving point represents an entity moving around in the space, for which only its position, but not its extent, is relevant. Some examples of moving points are cars, trucks and polar bears. On the other hand, a moving region represent a moving entity which changes its position as well as its extent, such as oil spills in the sea. Besides these main types, this algebra defines a large number of auxiliary data types, such as, *line* to represent the projection of a moving point into a plane and *moving real* to represent the time-dependent distance of two moving points. All data types are shown in Figure 2.5.

Moving regions and moving points, as well as, moving real and moving booleans, are called temporal types, that is, types whose values are functions from time (*instant*) to some domain. All operations over temporal and nontemporal types are presented in Table 2.1.

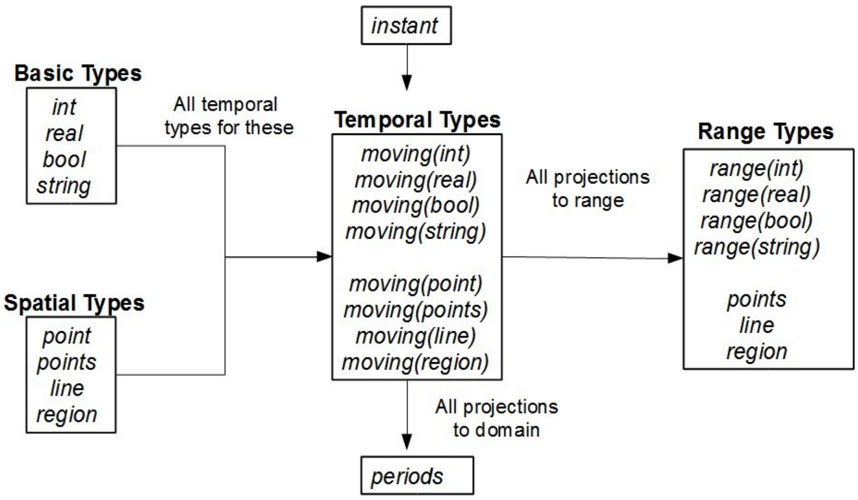


Figure 2.5 - Moving Objects Data Types.

Table 2.1 - Moving Objects Operations.

	Classification	***Operations***
nontemporal types	Predicates	isempty, =, <>, intersects, inside, <, >, <=, >=, before, touches, attached, overlap, on_border, in_interior
	Set Operations	intersection, union, minus, crossings, touch_points, common_border
	Aggregation	min, max, avg, center, single
	Numeric	no_components, size, perimeter, duration, length, area
	Distance and Direction	distance, direction
	Base Type Specific	and, or, not
	Projection to Domain and Range	deftime, rangevalues, locations, trajectory, routes, traversed, inst, val
	Interaction with Domain	atinstant, atperiods, initial, final, present, at, atmin, atmax, passes

temporal types	and Range	
	Rate of Change	derivative, speed, mdirection, turn, velocity
	Lifting	(all new operations inferred)
	When	when

The Moving Object algebra is presented by its authors in two levels of abstraction: abstract model and discrete model. Abstract model focus on the essential concepts without worry about implementation details. It is continuous, infinite and simpler than discrete model. However, it is possible only to store and handle discrete and finite information in computers. So, it is necessary to develop a discrete model which focuses on implementation, defining discrete and finite representations.

The discrete model can be implemented in an extensible database manage system (DBMS). The data types and operators can be embedded into its data model and query language in order to represent and query moving objects. In this case, consider the relation:

```
flights (id:string, from:string, to:string, route:mpoint)
```

The following question can be answered by using the operators of Table 2.1:

(a) "How far does flight LH 257 travel in French air space":

```
LET route257 = ELEMENT(SELECT route FROM flights
                       WHERE id = 'LH257');
length(intersection(France, trajectory(route257)));
```

(b) "What are the departure and arrival times of flight LH 257":

```
min(deftime(route257));  max(deftime(route257));
```

(c) "When and were did flight 257 enter French air space":

```
LET entry = initial(at(route257, France));
inst(entry); val(entry);
```

As a prototype of spatio-temporal database, the moving object model has been implemented in SECONDO, a database system that is extensible by algebra modules (http://dna.fernuni-hagen.de/Secondo.html/).

2.6 Geospatial lifeline

Mark et. al (1999) define a model for continuous movement in space-time based on the geospatial lifeline concept. A geospatial lifeline models a individual movement and is represented as a time-stamped record of locations that an individual has occupied over a period of time. The basic element of lifeline data is a space-time observation consisting of a triple <*Id*, *Location*, *Time*>, where *Id* is a unique identifier of the individual, *Location* is a spatial descriptor (such as a coordinate pair, a polygon and a street address), and *Time* is the time stamp when the individual was at that particular location (such as a clock time in minutes or event time in years).

Continuous processes are typically observed through a series of discrete samples, which are ordered by the times of their observations. These observations can be approximated for the actual movement of an object by different ways, taking into account different levels of granularities (Hornsby and Egenhofer, 2002). Granularity refers to the level of details at which phenomena is perceived. Consequently, depending on the desired granularity, a lifeline can be modeled as threads, beads, necklaces, convex hulls, tubular approximations, or trace.

A lifeline thread, Figure 2.6 (a), refers to a linear approximation of an ordered sequence of space-time samples. On the other hand, a lifeline bead, Figure 2.6 (b), represents the set of all possible locations that an individual could feasibly pass, based on a starting point (X_0, Y_0, T_0), an ending point (X_1, Y_1, T_1) and the maximum travel speed. A lifeline necklace, Figure 2.6 (c), is a sequence of beads and a convex hull lifeline, Figure 2.6 (d), is computed from geometric properties of the necklace.

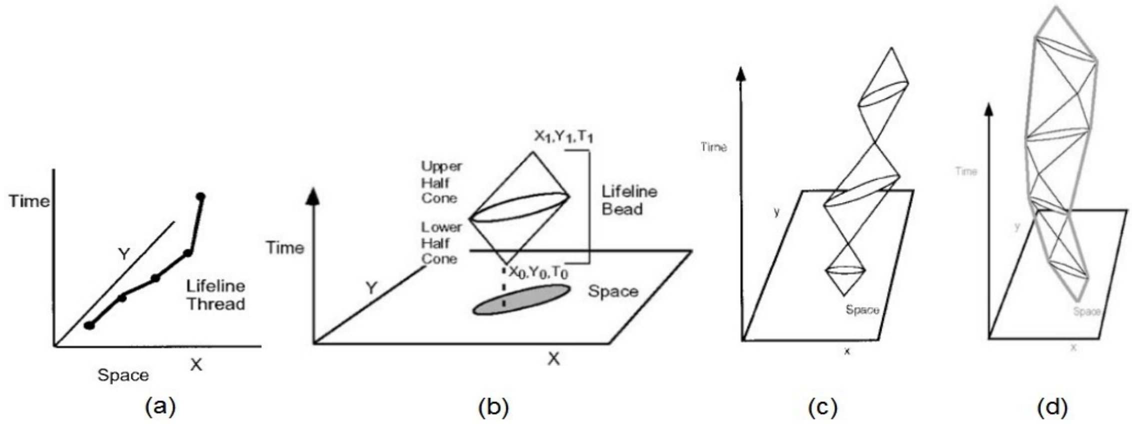


Figure 2.6 - Types of geospatial lifelines (a) thread (b) bead (c) necklaces and (d) convexhull.
Source: Adapted from Hornsby and Egenhofer (2002).

2.7 Hierarchal model: Events, Process and States

Yuan (2001) has proposed a conceptual framework which organizes complex geographic phenomena as a hierarchy of events, process and states. The author defines as complex all dynamic geographical phenomena which posses both field and object characteristics. For example, a wild-fire is in some sense a discrete object with a clear fire-front line, but there is an identifiable spatial and temporal variation within a fire.

The author defines an event as an occurrence of something significant, whereas a process as a sequence of dynamically related states that shows how something evolves. A process is in a continuing course of development, involving many changes in space and time, and is often captured by states. An event may consist of one or multiple process, a process may relate to multiple events, and a state may consist of footprints from one or more process.

In this model, the hierarchical representation consists of three data tiers: an event-composite layer, process-composite layers, and state layers, as shown in Figure 2.7. The event-composite layer records all event objects and their attributes, such as starting and ending time. Each event is associated with a set of processes in a process-composite layer, which is composed of process objects and their attributes. Each process object is associated with a set of state layers and a process attribute table which is built to record characteristics for individual processes. So, the object-like properties are stored with events and processes, and field-like properties are recorded on the state layers (Yuan, 2001).

In order to validate this model, the author has used it to represent and handle 882 digital precipitation arrays (DPAs) from April 15 to May 22, 1998, for the state of Okahoma, USA. Besides that, a prototype GIS has been implemented.

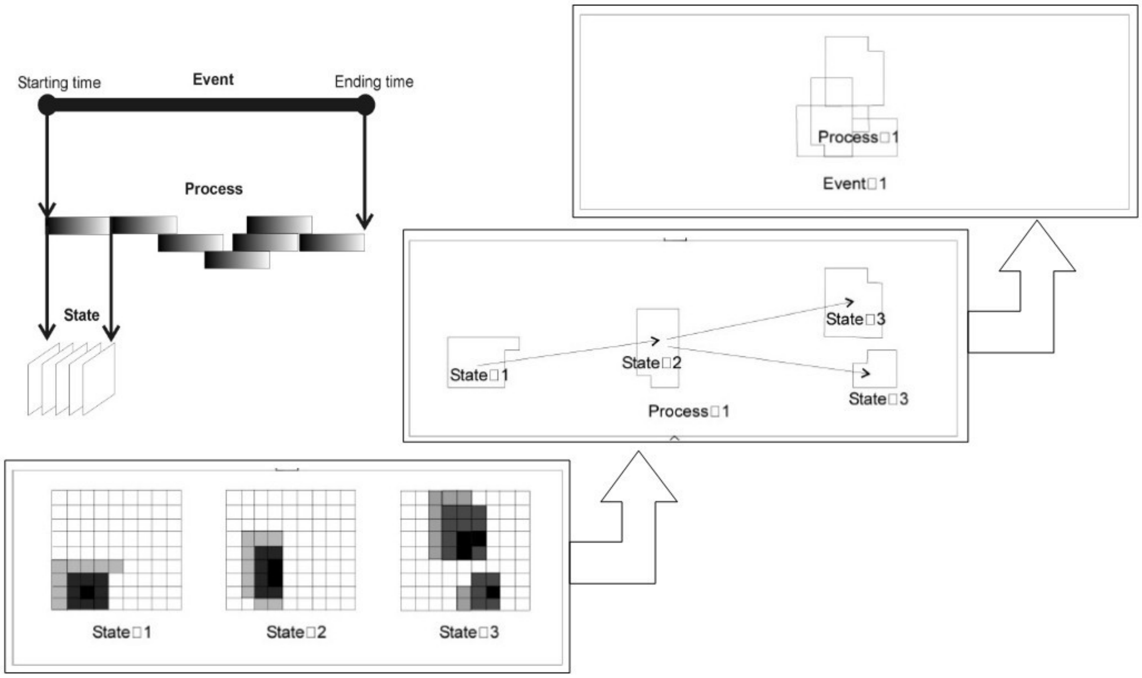


Figure 2.7 - Hierarchal model: Events, Process and States.
Source: Adapted from Yuan (2001).

2.8 Geospatial Event Model (GEM)

The GEM model, proposed by Worboys and Hornsby (2004), suggests an approach for dynamic geospatial domain based on the concepts of object and event. This model extends the traditional object-based geospatial models by introducing events into them. Besides the event concept, it defines two kinds of relationships: object-event and event-event.

In short, it delineates three categories of entities: objects (GeoObjects), events (GeoEvents) and settings (GeoSettings). Each object or event has a unique setting. A setting can be spatial (e.g., point, line and region), temporal (e.g., instant, interval and period) or spatio-temporal (e.g., trajectories, histories or geospatial lifeline). A spatio-temporal setting is a function from a temporal to a spatial setting. Finally, a geospatial event is associated to a spatio-temporal setting. Figure 2.8 shows the GEM scope.

In addition, this model defines two classes of relationships, one between two events (event-event) and the other between an object and an event (object-event). These relationships are described in Table 2.2.

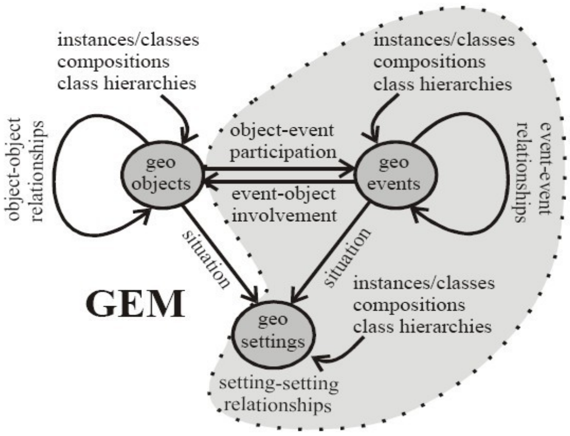


Figure 2.8 - GEM model: objects, events and their interaction.
Source: Adapted from Worboys and Hornsby (2004)

Table 2.2 – GEM model: relationships between event-event and object-event.

	Name	*Description*
event-event	Initiation	The occurrence of event A starts event B.
	Perpetuation / facilitation	The occurrence of event A plays a positive role in the initiation or continuation of event B.
	Hindrance / blocking	The occurrence of event A plays a positive role in the weakening, temporary stoppage, or termination of event B.
	Termination	The occurrence of event A allows/forces event B, already initiated, to terminate.
	Creation	An event that results in the creation of an object.
	Sustaining in being	An event that results in the continuation in existence of an

object-event		object.
	Reinforcement / Degradation	An event that has positive / negative effect on the existence of an object.
	Destruction	An event that results in the destruction of an object.
	Splitting / Merger	An event that creates/destroys a boundary between objects.

2.9 Feature-Based Temporal Model (FBTM)

The main characteristic of the feature-based temporal model is to represent the changes in space and theme of a feature independently (Choi et. al, 2008). The authors define three types of feature changes: (1) change in feature geometry or location with change in its themes; (2) change in feature geometry or location without change in its themes; and (3) changes in feature themes without change in its geometry or location. Therefore, in order to support these three types of changes explicitly, this model proposes to represent the three feature dimensions (space, theme, and time) separately and to link them explicitly through temporal relationships, as shown in Figure 2.9. In short, at a given time (T), a feature (f) is a set of temporal space (S_T) and temporal themes (H_T), that is, f : T → (S_T, H_T).

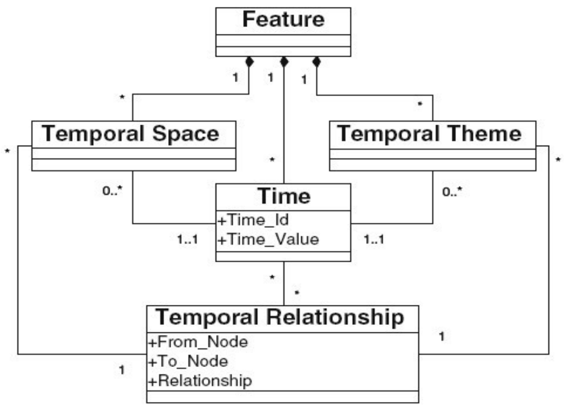


Figure 2.9 - Conceptual framework of feature-based temporal model.
Source: Adapted from Choi et. al (2008).

This model adopts and extends the key concepts of the STOM model and three domain model, describes in section 2.2 and 2.4, respectively. It represents features following the three-domain model approach, but uses a unique feature identifier to connect space, theme and time, as proposed in STOM model, instead of domain link table. In order to track spatial change history, three-domain and STOM models use a tree structure and persistence object identifier, respectively. The FBTM model also maintains persistent identifiers for spatial changes, but extends the STOM model by adding an object identifier for thematic changes. Besides that, the FBTM model defines explicit temporal relationships to represent temporal topology and types of changes.

Moreover, this model adopts and modifies ISO's temporal schema in order to represent time. The *Time* class, shown in Figure 2.9, stores temporal instants as TM_Instance class in ISO's temporal schema. And, the *TemporalRelationship* class is similar to TM_Topological class in ISO' temporal schema. But this class also includes an explicit temporal relationship, which extends the ISO' temporal schema.

The FBTM concepts has been materialized in a prototype system called Feature History Management System (FHMS).

2.10 Moving Feature Model

The International Organization for Standardization (ISO) has proposed a conceptual schema for moving feature (ISO, 2008). The term “feature” refers to an abstraction of real world phenomena and “moving feature” refers to features whose geometries move over time. This schema includes a set of classes, attributes, associations, and operations which provides a common conceptual framework to deal with feature geometry which moves as a rigid body. Therefore, it supports changes of location, translation and rotation of a feature, but not deformation.

This schema is based on the concept of one parameter set of geometries which can be viewed as a set of leaves or a set of trajectories. One parameter set of geometries is a function *f* from an interval $t \in [a, b]$ such that $f(t)$ is a geometry. And, for each point $P \in f(a)$ there is a one parameter set of points (called the trajectory of P) $P(t) : [a, b] \rightarrow P(t)$ such that $P(t) \in f(t)$.

Considering time as one parameter, a leaf represents the geometry of the moving feature at a particular value of time (e.g., a point in time) and a trajectory is a curve that represents the path of a point in the geometry of the moving feature. Moreover, this schema defines the prism and foliation concepts, as illustrated in Figure 2.10. In this figure, a 2D rectangle moves and rotates. Each representation of the rectangle at a given time is a leaf. The path traced by each corner point of the rectangle (and by each other point) is a trajectory. The set of points contained in all of the leaves, and in all of the trajectories, forms a prism. The set of leaves also forms a foliation.

A simple version of the class diagram proposed by the feature moving schema is shown in Figure 2.11. The classes form an inheritance hierarchy that has its source in the classes GM_Object and GM_Curve, specified in ISO 19107. The second level consists of a set of classes which describe a one-parameter geometry. These classes might be used to describe the movement of a feature with respect to any single variable, such as pressure, temperature, or time. Thus, the third level specializes these classes to describe motion in time.

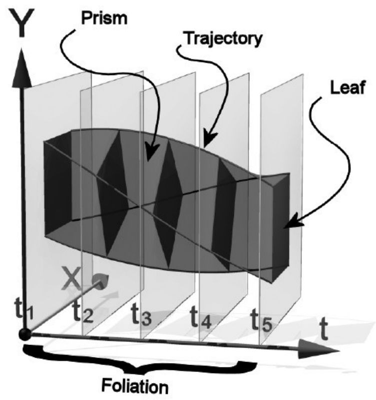


Figure 2.10: Feature movement as foliation.
Source: Adapted from ISO (2008).

Besides the data types, this model proposes a set of operations to handle moving features, such as, *leafGeometry* that returns a temporal geometry at a given time, *trajectory* that returns a temporal trajectory of a given point, *nearestApproach* that returns the nearest approach of the temporal geometry to any other given geometric object, *intersection* that computes the intersection between the temporal geometry and a given temporal object and *timeToDistance* that returns a graph of the time to distance.

This ISO model does not address other change types to the features, such as, the feature deformation and changes in non-spatial attributes of a feature.

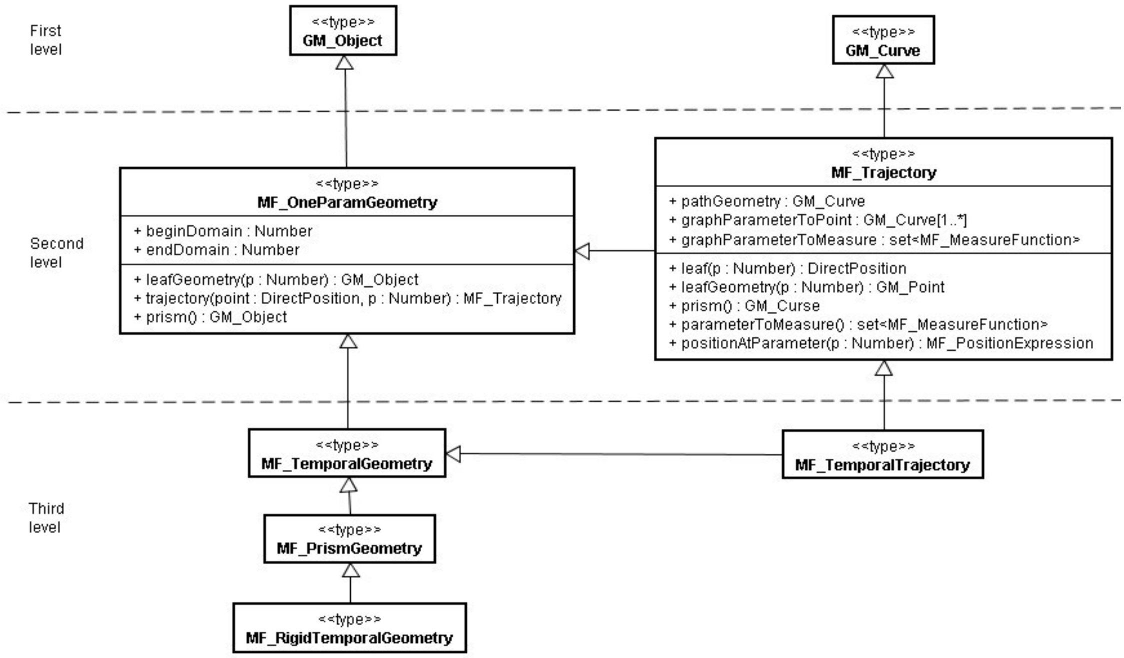


Figure 2.11 - Classes of feature moving schema.

2.11 Critical analyses

Based on the dichotomy, fields and objects, to represent geographical data (Couclelis, 1992), the models presented in this chapter can be classified in two classes: models which represent fields and the ones which represent objects that change over time. Still in the second class, there is a subset of models specialized in representing objects whose geometries change continuously over time, that is, objects in movement. This classification is shown in Table 2.3.

The Snapshot is the simplest model in representing fields which change over time. It has two main limitations: (1) operations among snapshots must compare them exhaustively; and (2) redundant storage because a complete snapshot is produced at each time slice, duplicating all unchanged data. The ESTDM model does not have the two limitations of Snapshot model because it stores only the changed cells by each event, reducing data volume and increasing computational efficiency. Besides that, ESTDM model defines a very simple event concept. In this model, an event is only a time occurrence. It does not explore the event concept, such as, event semantics or relationships among events.

Table 2.3 – Classification of existing spatio-temporal database models.

Fields which change over time	Objects which change over time	
	Discrete geometry change	Continuous geometry change
Snapshot Model ESTDM Model Hierarchal Model	STC Model STOM Model Three-domain Model GEM Model FBTM Model	Moving Object Model Geospatial lifeline Moving Feature Model

The Hierarchal model provides an interesting way of organizing complex geographical phenomena, that is, dynamic geographical phenomena which posses both field and object characteristics, in hierarchical layers. It is based on a sequence of snapshots called state layers, which represents fields which vary over time. Therefore, it has redundant storage problem like the Snapshot model. Besides the snapshots, this model also stores the objects which represent the phenomena. These objects are extracted from the state layers and associated to process layers. Thus, these two representations of phenomena, fields and objects, are used to improve the spatio-temporal query processing and operations. Finally, this model also defines the concepts of event and process and uses them in order to organize the data layers in different levels. So, events and processes are used like filters in order to determine which snapshots or state layers need to be processed, reducing the number of data layers to be searched in a query processing.

The STC model is an evolution of Snapshot model. Despite being very simple, it is important because it introduces the idea of representing spatial objects which vary over time.

The STOM is a model which represents discrete changes in object geometries over time. It defines two spatio-temporal data types, *ST-simplexes* and *ST-complexes*, and a set of operations over them, such as *ST-Union*, *ST-Intersection* and *ST-Difference*. However, this model does not consider changes in object attributes, that is, in the textual and numerical extents of geographical objects. For example, in the moving object model there are data types, such as *moving real* and *moving string*, to represent non-spatial attributes which vary over time. Moreover, using this model, it is impossible to represent objects whose geometries change continuously over time.

The Three-domain model focuses on how to represent objects which vary over time in a relational database system by using four normalized tables and a spatial graph as well as on how to query them by using SQL language. The proposed database schema can also be implemented in spatial DBMS, as PostGIS and Oracle Spatial, by using its support to deal with spatial information. It is a very simple model, without defining spatio-temporal data types and operations. It only uses the data types provided by DBMS and its query language.

Moving Object is a model for objects whose geometries change continuously over time. It is the most complete model presented in this work because it defines a robust algebra, data types and operations, in two levels of abstraction, abstract and discrete. Both levels of abstraction are essential because only discrete models can be implemented in computational systems. On the other hand, if we restrict attention directly to discrete models, there is a risk of missing a conceptually simple and elegant design of query operations. This is due to representation problems which might lead us to prematurely discard some options for modeling.

The principal disadvantage of this model is not to consider fields which vary over time. For instance, a hurricane must be represented in this model as a moving region. However, in some applications, the best representation of a hurricane is a field which varies over time. Besides that, this model is specific for continuous geometric changes, without considering discrete geometric changes, such as land parcel history.

Likewise Moving Object, Geospatial lifeline also models moving objects. Nevertheless, it defines only a simple data type to represent moving points or regions and some different types of lifelines or trajectories extracted from moving points. Depending on the desired granularity and on the application type, distinct types of trajectories are essential. For example, in animal tracking monitoring, the convexhull trajectory is necessary in order to define an animal habitat. Another example is infected persons monitoring. In this application, in order to identify all possible areas where a person might have passed by, it is necessary to consider a necklace trajectory. So, although it does not define operations for moving objects, it defines important different types of trajectories. In the Moving Object model only the linear or thread trajectory is extracted from moving points through the operator *trajectory*.

The GEM model is interesting because it introduces an event concept and relationships between events and objects in a model based on spatial objects. It defines different types of relationships, as shown in Table 2.2, following the idea that an event can affect or be associated to one or more objects of different types. In short, it is a simple model which defines only data types but not operations.

The FBTM represents the three dimensions (space, theme, and time) of a feature individually, like three-domain model. It also adopts and modifies ISO's temporal scheme to provide details on time dimension. This model focuses on representing and querying the history of objects over time. It does not define more complex operations like st-crosses and st-touches. Moreover, it represents only discrete changes in geometries.

Finally, the Moving Feature models features whose geometries move over time. In this model, the state of a geometry at a given time is called *leaf*, the set of all states of a geometry is called *foliation* and each point of a geometry in movement has its own *trajectory*. The main advantage of this model is to define a generic type called one-parameter geometry which represents the variation of feature geometry with respect to any single variable, such as pressure, temperature, or time. However, its main disadvantages are not to consider feature geometry deformation and changes in feature non-spatial attributes. For instance, due to these limitation, it is not possible to represent an oil slick moving on the ocean.

3 TOWARDS AN ALGEBRA FOR SPATIO-TEMPORAL DATABASE

As shown in the previous chapter, the existing spatio-temporal database models are specific to represent either fields or objects which vary over time. Besides that, models which represent objects that change over time are specific to represent either discrete or continuous geometry changes. Finally, there are models which do not represent non-spatial attribute changes of objects.

However, in order to represent dynamic geographical processes, a model is necessary to represent fields and objects which change over time, considering discrete and continuous geometry and non-spatial attribute changes. For example, a hurricane or a volcanic eruption can be represented by fields which vary over time. On the other hand, an animal life or a flight can be represented by objects whose geometries and attributes change over time. Finally, a forest deforestation process can be represented by a simple time series associated to each forest region or cell.

Therefore, this work aims at constructing an initial version of an algebra for dynamic geographical processes. The idea is to define a robust, clear and formal algebra for spatio-temporal database, as the one defined in the moving object model shown in the last chapter. Like the abstract model of the moving object algebra, this work is going to present only an abstract or conceptual model, without considering implementation details for now.

In order to accomplish this goal, this work follows the steps defined in the four universes paradigm for geoinformatics (Camara, 1995), as shown in Figure 3.1 . This paradigm defines four steps between the real geographical world and its computational representation. So, before defining data types and operations to represent and handle dynamic geographical phenomena, this work firstly reviews philosophical and geographical ontologies driven to represent it.

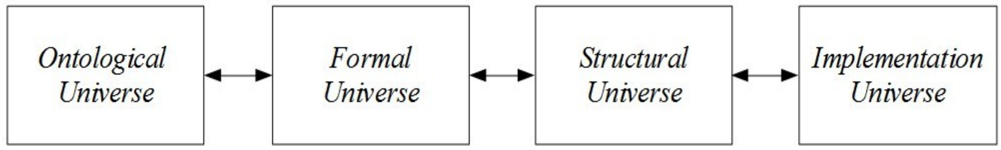


Figure 3.1 - Four universes paradigm.

3.1 Philosophical and Geographical Ontologies

According to Galton (2008), in philosophical ontology there is a long-standing and well-established classification of real world phenomena into *continuants* and *occurrents*. The former class represents entities that endure in the world through time while the latter represents entities that happen or go on in time. So, processes and events are considered *occurrents*. The main characteristics of a *continuant* are: (a) can undergo changes, (b) has spatial parts but not temporal part, and (b) is wholly present at each moment of its existence. Otherwise, the main features of an *occurrent* are: (a) can not undergo change, (b) has temporal parts, and (c) is not wholly present at any time short of its entire durations.

Some examples of *continuants* are: a person, an aircraft, and a volcano. While some examples of *occurrents* are: a persons' life, a flight and an eruption. Figure 3.2 shows an eruption of Karthala volcano through a sequence of six images, taken at different times, from Meteosat Second Generation (MSG) satellite. This figure illustrates that it is not possible to represent the whole eruption by considering only one image at a time. That is, it is necessary to consider its entire duration, the six images, in order to represent the whole process. Due to this characteristic, an eruption is considered an *occurrent*.

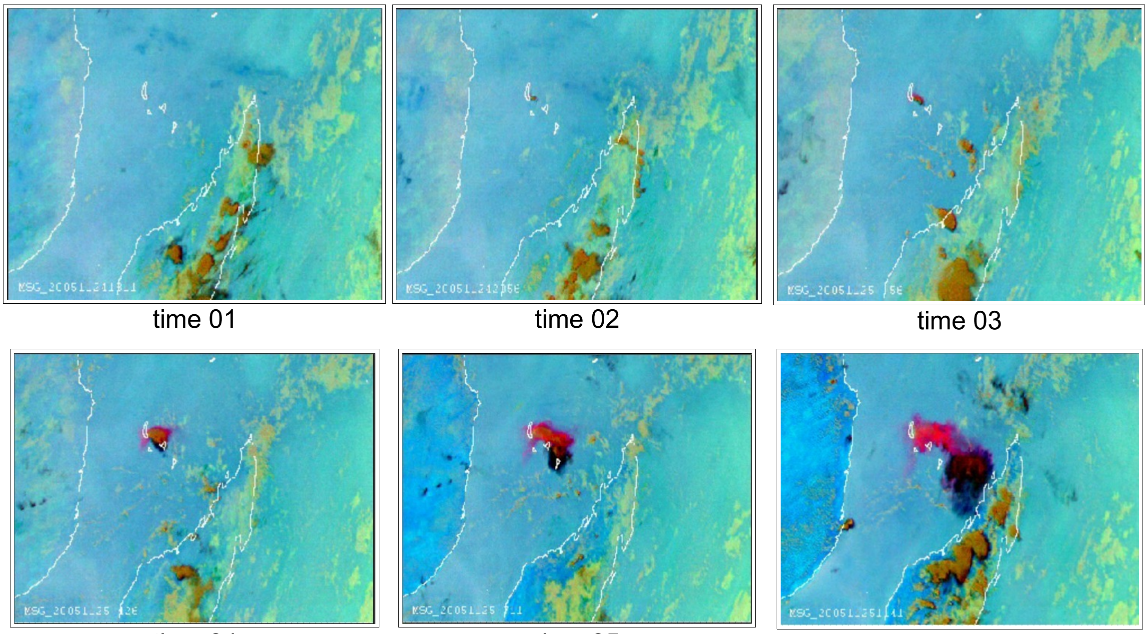


Figure 3.2 - Eruption of Karthala volcano.

Based on this classification, Grenon and Smith (2004) have proposed two ontologies to represent dynamic features of reality, SNAP for *continuants* and SPAN for *occurrents*. Each SNAP ontology is indexed by a single time instant, by a specific domain, and by a level of granularity. It recognizes only enduring entities at the time of its index, that is, entities which typically have already existed for some time in the past and will go on existing in the future. In other words, entities which are not instantaneous and not have temporal parts. For example, a SPAN ontology of the zoological domain, with index *now* would contain no entity of the type *dinosaur*. On the other hand, each SPAN ontology is indexed by a single time interval, and also by a specific domain and by a level of granularity. It recognizes entities which unfold themselves through a time interval in their successive temporal parts.

Since *continuants* can change over time, different SNAP ontologies are required, each one to represent a snapshot of the world at a time. In this case, changes are represented in differences between successive snapshots. On the other hand, the SPAN ontology encapsulates these changes as *occurrents*. Thus, there is just one SPAN ontology associated to a complete historical succession of SNAP ontologies. As SPAN ontology is analogous to snapshot of reality, so SPAN ontology is analogous to videos spanning time.

Besides these two ontologies, the authors defines trans-ontological relations in order to integrate distinct ontologies. There are three types of trans-ontological relations: SNAP-SNAP, SPAN-SPAN, and SNAP-SPAN. In the last relation, each SNAP entity is related to an unique SPAN entity which is its history or life. An an example, Figure 3.3 illustrates how to represent the volcanic eruption, shown in Figure 3.2, by using these ontologies.

For each ontology, the authors define different entity categories. There are SNAP categories, such as *spatial regions,* and SPAN categories, such as *spatiotemporal regions*, *temporal regions* as well as *process* and *events*. Process is defined as temporally extended *occurrents* and events as entities within processes which happen in single instants of time.

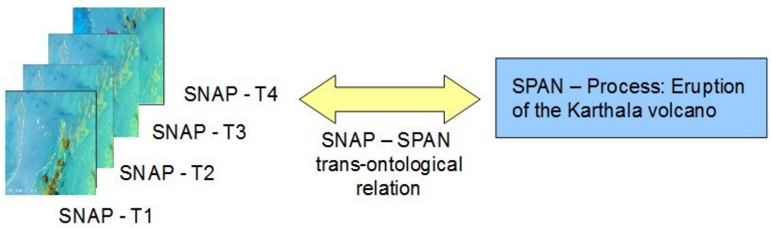


Figure 3.3 – SNAP/SPAN ontologies.

Finally, the general SNAP and SPAN ontologies have been applied to the geography domain, resulting in a geographical ontology. Geographical entities – both *continuants* and *occurrents* – are those entities appearing at a certain level of granularity and which have a certain relation to the Earth. Some examples of SNAP geographical entities are: mountains, volcano and lakes, valley and deltas and land parcels and cities. And of SPAN geographical entities are: a volcanic eruption, movement of air fronts, tornadoes, epidemic transmissions of diseases, and mountain erosions. The relationships among these ontologies are shown in Figure 3.4.

In the SNAP/SPAN ontology (Grenon and Smith, 2004), processes and events are considered SPAN or *occurrent* entities, that is, entities which can not change. Nevertheless, Galton (2008) has presented some cases where processes can change, for instance, “My life is becoming harder” and “The protest became violent”. Thus, based on these cases, he has rejected the identification of process as *occurrents* and has proposed two new ontologies called EXP/HIST instead.

The EXP ontology relates to the world as we experience it, when it is present and in constantly changing. It contains time-dependent entities, such as objects and processes, and their properties. In contrast, the HIST ontology relates to the historical record. It is used to describe synoptic overviews that span a succession of instantaneous experiential, EXP, snapshots. It contains entities which do not themselves change, such as events (Galton, 2008).

3.2 Formal Universe

The formal or conceptual universe contains formal abstractions (data models and algebras) which are necessary to represent entities of the ontological universe. Considering the geographical ontology presented in the previous section, it is necessary to have formal abstractions to represent SNAP and SPAN geographical entities. These formal abstractions as well as philosophical and geographical ontologies are shown in Figure 3.4.

In order to represent SNAP geographical entities, object-based and field-based approaches have been used by the majority of geographical information systems (GIS). This dichotomy, discussed by Couclelis (1992), recognizes the importance of two different kinds of entities, objects and fields, to better represent discrete and continuous geographical phenomena, respectively. Although some research in literature points out the inefficacy of fields and objects to represent some kinds of geographical phenomena (Burrough and Frank, 1996), this work is based on this dichotomy in order to move on towards a representation of spatio-temporal information. The idea is to discuss this inefficacy in future research, even in relation to dynamical geographical processes.

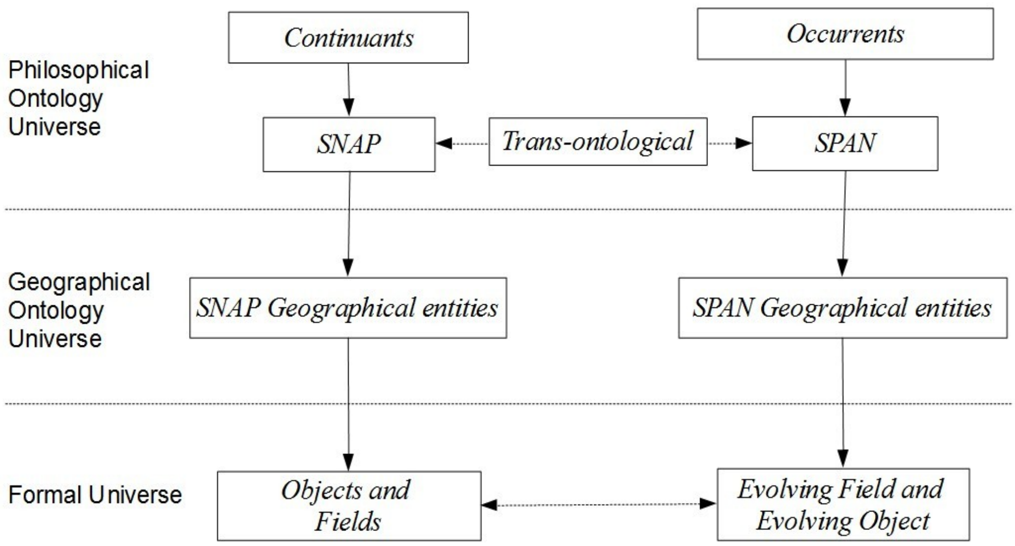


Figure 3.4 – Philosophical and geographical ontologies and formal universe.

Fields deal with real-world phenomena that continuously vary over space, such as, elevation and temperature. In GIS practice, a variety of terms have been used to refer to fields, such as *coverage* and *surface*. In short, a field ($Field$) can be represented as a function from spatial locations (S) to a set of attribute values (

A) from some domains:

$$Field: \; S \rightarrow \{A\}$$

An object, on the other hand, is an individual identifiable entity to which various attributes can be ascribed, including its location or form. Again, in GIS practice, several terms have been used to refer to spatial object, including *entity* and *feature*. In short, a object ($Object$) can be represented by a pair of spatial locations or forms (S) and a set of non-spatial attribute values (A) from some domains. Thus, a set of objects is represented as a function from object identity (I) to object:

$$Object: \; (S, \{A\})$$

$$Objects: \; I \rightarrow Object$$

3.2.1 Evolving Field and Evolving Objects

In order to represent SPAN geographical entities or dynamic geographical processes, this work introduces two main data types: *evolving field* and *evolving object*. Evolving field to represent the variation or evolution of a field over time and evolving object, of objects.

A volcano is a *continuant* and, therefore, can be represented as an object with its own properties, such as its location or form, its name, and the city it belongs to. However, a volcanic eruption is a dynamic process or an *occurrent* associated to a specific volcano. Considering that an eruption is measured by three attributes: temperature, SO_2, and CO_2 emission, a data type is necessary to represent the variation of these attributes in space over time. Thus, this data type is called evolving field.

As an example, the eruption presented in Figure 3.2 could be represented by six distinct fields, each one in a different time, instead of an evolving field. That means to represent it by six snapshots, using six SNAP ontologies, each one indexed by a time instant. In this case, the values of these three attributes for each space or cell must be represented in each snapshot, even those values which do not change for the same space between two snapshots. Therefore, the advantages of representing the whole eruption as an evolving filed are: (1) the use of only one SPAN ontology; (2) the evolving field data type can be optimized to represent only the attribute values which change between two successive times for the same space; and (3) the possibility to define specific and optimized operations over an evolving field data type, such as, “extract the time series of SO_2 emission from this eruption”.

Two other examples of evolving field are the process of mountain erosion and the process of forest deforestation. In the former example, a mountain is a *continuant* and can be represented by a SPAN geographical entity. However, its erosion process is an *occurrent* and can be represented by an evolving field, considering the variation of the lost soil over time. In the latter example, it is possible to represent the forest deforestation by using an evolving field generated from satellite images which identify this process.

On the other hand, there are *occurrents*, such as flight or animal tracking, which need to be represented by entities whose locations or forms and attributes vary over time. For example, in animal tracking monitoring, the temperature and location of an animal can be measured at different times, as shown in Figure 3.5. Another example is the history of a land parcel where its boundaries and owners vary over time, as illustrated in Figure 3.5. So, in order to represent these entities, this work proposes the evolving object data type. It is important to note that an evolving object represents space varying in a continuous way, such as in animal monitoring, as well as in a discrete way, such as the parcel history.

Galton (Galton, 2004) has discussed two approaches, 'three-plus-one'-dimensional and four-dimensional, to represent spatio-temporal information in GIS.

The first approach is more conservative in which the temporal dimension is held separate from space, distinguishing sharply between spatial and temporal objects. In the latter approach, purely temporal objects and spatial regions are replaced by “chunks” of space-time, where time is considered a fourth dimension.

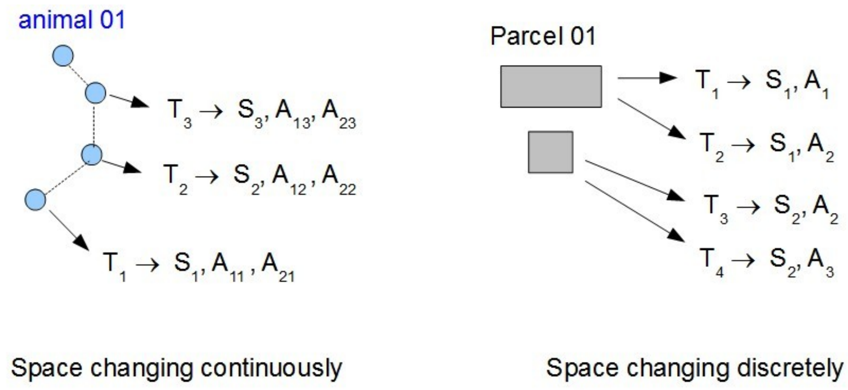


Figure 3.5 – Animal tracking and parcel history.

The algebra proposed in this work follows a 'three-plus-one'-dimensional approach to represent spatio-temporal data, as presented in the next section. We believe that space and time have to be separately handled because both exhibit essential different characteristics. For instance, we can ask if time T_1 is less than time T_2, but this question does not make sense for two spaces S_1 and S_2.

3.3 Towards an algebra for spatio-temporal database

This section presents a first version of an algebra to represent and handle SPAN geographical entities or *occurrents*. This algebra consists in a set of data types to represent *occurrents* and a set of operators to handle and query them.

All data types of this algebra are shown in Figure 3.6. We can observe in this figure that there are three base groups of data types: basic, temporal and spatial.

Basic data types, called A, are: *int*, *real*, *bool* and *string*. Temporal data types, called T, are: *instant* which represent a single instant of time and *period* which represents a time interval composed of two time instants. Finally, spatial data types, called S, are: *point*, *line*, *polygon* and *cell*.

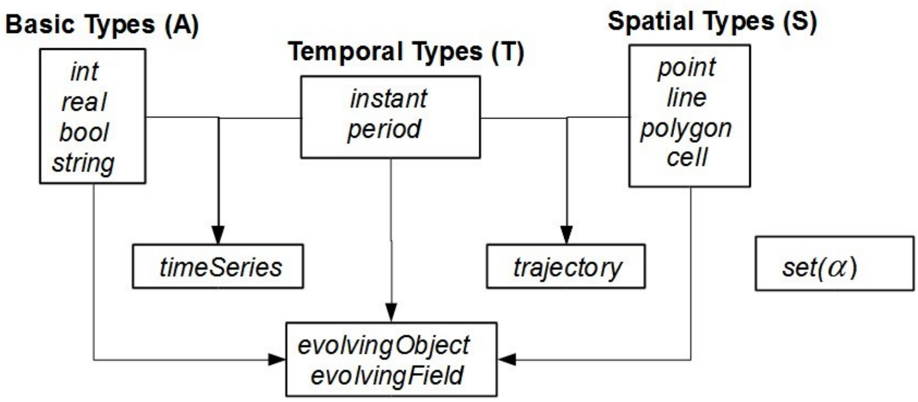


Figure 3.6 – Algebra data types.

The data types of these base groups are combined to generate four new ones: *timeSeries*, *trajectory*, *evolvingField*, and *evolvingObject.* A time series is a function from time (temporal type T) to attribute values from a base data type (A). A trajectory is a function from time (temporal type T) to spatial locations or forms (spatial type S) . An evolving field and an evolving object represent a field and an object which vary or change over time, respectively. An evolving field is a function from time (T) to spatial locations (S) to a set of attribute values from base data types (A). Finally, an evolving object is a function from time (T) to pairs of spatial locations or forms (S) and a set of attribute values from base data types (A).

In order to represent a set of values of a specific domain, we have proposed the data type *set(α)*, where α can be a base, spatial or temporal data type:

$$\texttt{timeSeries}: T \rightarrow A$$

$$\texttt{trajectory}: T \rightarrow S$$

$$\texttt{evolvingField}: T \rightarrow S \rightarrow \{A\}$$

$$\texttt{evolvingObject}: T \rightarrow (S, \{A\})$$

$$\texttt{set}(\alpha): \{\alpha\} \quad \alpha \in \{A, T, S\}$$

A time series is a sequence of measures, where each one is associated to a specific time. For instance, a sensor which collects humidity and temperature per hour in a city generates two time series, one which contains a humidity measure per hour and the other a temperature measure per hour. In this example, we are considering a fixed sensor, that is, a sensor whose location does not change over time. If this sensor is not fixed, it must be considered as an evolving object in order to represent its location and measures which vary over time.

Another example of application which generates time series is the monitoring of dengue fever in some Brazilian cities, based on egg traps for *Aedes aegypti* and *Aedes albopictus* mosquito. This monitoring consists in giving out egg traps in different locations around a city and in counting the infected eggs from each trap weekly. In this case, each egg trap generates a time series of the number of infected eggs per week.

A trajectory represents a space variation, location or form, over time. For instance, the trajectory of a flight or an animal tracking consists of the set of airplane or animal locations at different times during the flight or tracking. The trajectory of a land parcel consists of a set of its boundaries at different times during its history. We can note that a trajectory can represent space varying in a continuous way, such as flight and animal tracking, as well as in a discrete way, such as land parcel history.

In order to handle and query *occurrents*, this work proposes a set of operators over the data types presented in Figure 3.6. All operators and their descriptions are shown in Table 3.1.

Table 3.1 – Algebra operators.

Operation	*Description*
state: evolvingObject × T → object	Returns the state of an evolving object in a given time.
snapshot: evolvingField × T → field	Returns the snapshot of an evolving field in a given time.
timeSeries: evolvingObject × atName → timeSeries timeSeries: evolvingField × atName × aggrOp → timeSeries	Returns a time series: (1) of a specific attribute of an evolving object or (2) of a specific attribute of an evolving field, by using a aggregation operator, such as, SUM, AVG, and COUNT.
trajectory: evolvingObject → trajectory	Returns a trajectory from an evolving object.
time:evolvingObject → set(T) time:evolvingField → set(T) time:trajectory → set(T) time:timeSeries → set(T)	Returns the set of time values associated to an evolving object, an evolving field, a trajectory and a time series.
range: timeSeries → set(A) range: trajectory → set(S)	Returns the set of range values of a time series and a trajectory.
selection: timeSeries × condition → timeSeries selection: trajectory × condition → trajectory	Returns a selection of a time series or a trajectory, based on a specific condition.
	Returns the intersection between a space (e g. a polygon or a line) and

intersetion: evolvingObject × S → {evolvingObjects} intersetion: evolvingField × S → evolvingField	an evolving object or evolving field.
distance: trajectory × trajectory → timeSeries distance: timeSeries × timeSeries → timeSeries	Returns, for each time, the spatial distances between two trajectories or the distances between attribute values of two time series.
area: trajectory → timeSeries	Returns, for each time, the spatial area of each geometry in a trajectory.
linearTrajectory: trajectory → line necklaceTrajectory: trajectory → polygonSet convexhullTrajectory: trajectory → polygon	Returns linear, necklace and convexhull trajectories, as described in section 2.6.
max, min, avg, ...: set(α) → α	Returns the maximum, minimum, and average value of a set of values.

Considering these data types and operators embedded into a DBMS data model as attribute data types and SQL operators, we can define the following relations:

egg_traps (id: string, address: string, location: point, infected_eggs: timeSeries)

parcels (id: string, history: evolvingObject)

animal_tracking (id: string, description: string, tracking: evolvingObject)

```
eruptions ( id: string, volcano: string,
            eruption:evolvingField)
```

Based on these relations, we can answer the following questions by using the operators shown in Table 3.1:

1) What is the average of infected eggs in trap T01? When was the biggest number of infected eggs collected in this trap?

```
LET tSeries = ELEMENT(SELECT infected_eggs FROM egg_traps
                      WHERE id = 'T01');
PRINT avg(range(tSeries));
LET maxVal = max(range(tSeries));
PRINT time(select(tSeries, RANGE_VALUE == maxVal));
```

2) When was parcel P01 adjacent to street S01?

```
LET historyPA01 = ELEMENT(SELECT history FROM parcels
                          WHERE id = 'P01');
LET interP01S01 = intersection(historyPA01, streetS01);
FOR EACH i IN interP01S01
     PRINT min(time(interP01S01[i]));
     PRINT max(time(interP01S01[i]));
```

3) Did animal A01 cross the natural reservation X (considering the convexhull trajectory)?

```
LET animalA01 = ELEMENT(SELECT tracking FROM animal_tracking
                        WHERE id = 'A01');
intersects(reserve_x,
convexhullTrajectory(trajectory(animalA01)));
```

4) When did animal A01 cross the natural reservation X? And what was its temperatures inside the reservation X? And what was its mean temperature inside this reservation?

```
LET animalA01 = ELEMENT(SELECT tracking FROM animal_tracking
                         WHERE id = 'A01');
LET interA01Resx = intersection(animalA01, reserve_x);
FOR EACH i IN  interA01Resx
  min(time(interA01Resx[i]));
  max(time(interA01Resx[i]));
  PRINT timeSeries(interA01Resx[i], temperature);
  PRINT avg(range(timeSeries(interA01Resx[i], temperature)));
```

5) When and where did animal A01 meet animal A02 (minimal distance between both is less than 2 meters)?

```
LET animalA01 = ELEMENT(SELECT tracking FROM animal_tracking
                         WHERE id = 'A01');
LET animalA02 = ELEMENT(SELECT tracking FROM animal_tracking
                         WHERE id = 'A02');
LET tSeries = distance(trajectory(animalA01),
                       trajectory(animalA02));
LET t = time(select(tSeries, RANGE_VALUE <= 2));
PRINT (state(animalA01, t));
```

6) When was the biggest SO_2 emission of Karthala volcano eruption?

```
LET eruption = ELEMENT(SELECT eruption FROM eruptions
```

```
                       WHERE volcano = 'Karthala');
LET tSeries = timeSeries(eruptionInCity, SO2, COUNT);
LET maxVal = max(range(tSeries));
PRINT time(select(tSeries, RANGE_VALUE == maxVal));
```

7) When did the SO_2 emission of Karthala volcano eruption reach the city?

```
LET eruption = ELEMENT(SELECT eruption FROM eruptions
                       WHERE volcano = 'Karthala');
LET eruptionInCity = intersection(eruption, city);
LET tSeries = timeSeries(eruptionInCity, SO2, COUNT);
PRINT min(time(select(tSeries, RANGE_VALUE > 0))
```

4 CONCLUSION

This work presents a review of the principal existing spatio-temporal database models, providing a critical analyses, and an initial version of an algebra for dynamic geographical processes.

According to Pelekis et al. (2004), spatio-temporal research has focused on a number of specific areas, including: (a) the ontology of space and time, (b) development of efficient and robust space-time database models and languages, (c) inexactness and scaling issues, (d) graphical user interfaces and query optimization, and (e) indexing techniques for space-time databases. Considering this classification, this work focuses on (a) and (b).

The main goal of this work is to present a lack of clear, robust and formal algebra able to represent different types of dynamic geographical processes and, therefore, to propose an initial version of an algebra for spatio-temporal database. The idea is to define an algebra as clear and robust as the one defined in the moving object model. Thus, we intend to continue working on the data types and operators of this algebra initial version in order to get the expected clearness and robustness.

As future work, our objective is to extend the algebra with: (a) process and event concepts in order to represent the semantic of dynamic geographical processes and their relationships, for instance, "Katrina hurricane **started up** a flooding process"; (b) operators between evolving object and evolving field data types; and (c) operations over a set of evolving object and evolving field.

5 REFERENCES

BURROUGH, P.A.; FRANK, A.U. (Eds.) **Geographic Objects with Indeterminate Boundaries**. Bristol: Taylor and Francis, 1996.

CÂMARA, G. **Modelos, Linguagens e Arquiteturas para Bancos de Dados Geográficos**. 1995. Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 1995. Available at: <www.dpi.inpe.br/teses/gilberto/>. Access at: 27/01/2009.

CHOI, J.; SEONG, J. C.; KIM, B.; USERY, E. L. Innovations in Individual Feature History Management—The Significance of Feature-based Temporal Model. **Geoinformatica**, v. 12, n. 1, p. 1-20, 2008.

COUCLELIS, H. People manipulate objects (but cultivate fields): beyond the raster-vector debate in GIS. In: FRANK, A. U.; CAMPARI, I.; FORMENTINI U. (Eds). **Theory and Methods of Spatio-Temporal Reasoning in Geographic Space**. Berlin: Springer, 1992. p. 65–77

FORLIZZI, L.; GUTING, R. H.; NARDELLI, E.; SCHNEIDER, M. **A Data Model and Data Structures for Moving Objects Databases**. In: ACM SIGMOD International Conference on Management of Data. **Proceeding...** Dallas, Texas, United States, 2000. p. 319-330. ISBN:1-58113-217-4

GALTON, A. Fields and Objects in Space, Time and Space-Time. **Spatial Cognition and Computation Journal**, v. 4, n. 1, p. 39-68, March 2004.

GALTON, A. Experience and History: Processes and their Relation to Events. **Journal of Logic and Computation**, v. 18, n. 3, p. 323-340, June 2008.

GRENON, P.; SMITH, B. SNAP and SPAN: Towards dynamic spatial ontology. **Spatial Cognition and Computation Journal**, v. 4, n. 1, p. 69-104, March 2004.

GUTING, R. H.; BOHLEN, M. H.; ERWING, M.; JENSEN, C. S.; LORENTZOS, N.; NARDELLI, E.; SCHNEIDER, M.; VIQUEIRA, J. R. R. Spatio-temporal Models and Languages: An Approach Based on Data Types. In: KOUBARAKIS, M.; SELLIS, T. et. al (Ed.). **Spatio-Temporal Databases: the CHOROCHRONOS Approach**. Germany, 2003.

GUTING, R. H.; SCHNEIDER, M. **Moving Objects Databases**. San Francisco, CA: Morgan Kaufmann, 2005. p. 389.

HORNSBY, K.; EGENHOFER, M. J. Modeling Moving Objects over Multiple Granularities. **Annals of Mathematics and Artificial Intelligence**. v. 36, n. 1-2, p. 177-194, September, 2002.

INTERNATIONAL BUSINESS MACHINES CORPORATION (IBM). **DB2 Spatial Extender and Geodetic Data Management Feature**: User's Guide and Reference. IBM, 2006. 536 p. (S517-8556-00). Available at: <www.ibm.com/shop/publications/order>. Access at: 27/01/2009.

INTERNATIONAL STANDARD ORGANIZATION (ISO). **Geographic information — Schema for moving features** (ISO 19141), June, 2008.

LANGRAN, G.; CHRISMAN, N. R. A Framework For Temporal Geographic Information. **The International Journal for Geographic Information and Geovisualization**, v. 25, n. 3, p. 1-14, 1988.

MARK, D.; EGENHOFER, M. J.; BIAN, L.; HORNSBY, K.; ROGERSON, P.; VENA, J. **Spatio-temporal GIS analysis for environmental health using geospatial lifelines**. In: 2nd International Workshop on Geography and Medicine (GEOMED'99). **Proceeding...** Paris, France, 1999.

OPEN GEOSPATIAL CONSORTIUM (OGC). **OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture**. OGC, 2006. 95 p. (OGC 06-103r3). Available at: <www.opengeospatial.org/>. Access at: 27/01/2009.

ORACLE CORPORATION (ORACLE). **Oracle Spatial User's Guide and Reference**: 10g Release 1 (10.1). Oracle, 2003. 602 p. (B10826-01). Available at: <www.oracle.com/technology/products/spatial>. Access at: 27/01/2009.

PELEKIS, N.; THEODOULIDIS, B.; KOPANAKIS, I.; THEODORIDIS, Y. Literature Review of Spatio-Temporal Database Models. **The Knowledge Engineering Review**, v. 19, n. 3, p. 235-274, 2004.

PEUQUET D. J.; DUAN, N. An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. **International Journal of**

Geographical Information Science, v. 9, n. 1, p. 7-24. 1995.

REFRACTIONS RESEARCH. PostGIS 1.3.5 Manual. REFRACTIONS, 2008. 78 p. Available at: <http://postgis.refractions.net/>. Access at: 27/01/2009.

RIGAUX, P.; SCHOLL, M.; VOISARD, A. **Spatial Databases**: With Application to GIS. San Francisco, USA: Morgan Kaufmann, 2002.

WORBOYS, M. Event-oriented approaches to geographic phenomena. **International Journal of Geographical Information Science**, v. 19, n. 1, p. 1-28, January 2005.

WORBOYS, M. F. A Unified Model for Spatial and Temporal Information. **The Computer Journal**. v. 37, n. 1. 1994.

WORBOYS, M. F. Object-oriented approaches to geo-referenced information. **International Journal of Geographical Information Science**. v. 8, n. 4, p. 385-399. July, 1994.

WORBOYS, M.; HORNSBY, K. **From Objects to Events: GEM, the Geospatial Event Model**. Lecture Notes in Computer Science, v. 3234, p. 327-343, 2004.

YUAN, M. Use of a Three-Domain Representation to Enhance GIS Support for Complex Spatio-temporal Queries. **Transaction in GIS**, v.3 n.2, p.137-159, 1999.

YUAN, M. Representing Complex Geographic Phenomena in GIS. **Cartography and Geographic Information Science**, v. 28, n. 2, p.83-96, 2001.