# CHOROCHRONOS

**A Network for Spatiotemporal Database Systems**

*National Technical University of Athens (NTUA)*
*Aalborg University (AALBORG)*
*FernUniversitaet Hagen (HAGEN)*
*University of L'Aquila (UNIVAQ)*
*University of Manchester - Institute of Science and Technology (UMIST)*
*Politecnico di Milano (POLIMI)*
*Institut National de Recherche en Informatique et en Automatique (INRIA)*
*Aristotle University of Thessaloniki (AUT)*
*Technical University of Vienna (TU VIENNA)*
*Swiss Federal Institute of Technology, Zurich (ETHZ)*

# TECHNICAL REPORT SERIES

TECHNICAL REPORT CH-98-08

## CONCEPTUAL DATA MODELING FOR SPATIOTEMPORAL APPLICATIONS

*NECTARIA TRYFONA and CHRISTIAN S. JENSEN*

*SEPTEMBER 1998*

**CHOROCHRONOS: TMR Research Network Project, No ERBFMRXCT960056**
**Contact Person: Prof. Timoleon Sellis, Dept. of Electrical and Computer Engineering, National Technical University of Athens, Zografou 15773, GR Tel: +30-1-7721601, FAX: +30-1-7721659, timos@cs.ntua.gr**

# Conceptual Data Modeling for Spatiotemporal Applications [1]

Nectaria Tryfona Christian S. Jensen

Department of Computer Science

Aalborg University

Fredrik Bajers Vej 7E, DK-9220 Aalborg ∅st, Denmark

{tryfona,csj}@cs.auc.dk

**Abstract**

Many exciting potential application areas for database technology manage time-varying, spatial information. In contrast, existing database techniques, languages, and associated tools provide little built-in support for the management of such information. The focus of this paper is on enhancing existing conceptual data models with new constructs, improving their ability to conveniently model spatiotemporal aspects of information. The goal is to speed up the data modeling process and to make diagrams easier to comprehend and maintain. Based on explicitly formulated ontological foundations, the paper presents a small set of new, generic modeling constructs that may be introduced into different conceptual data models. The ER model is used as the concrete context for presenting the constructs. The semantics of the resulting spatiotemporal ER model is given in terms of the underlying ER model.

## 1. Introduction

Improved support for the development of information systems involving time-varying geo-referenced information–so-called spatiotemporal information–has been a long-term user requirement in a variety of areas, such as cadastral systems that capture the histories of landparcels, routing systems computing possible routes of vehicles, and forecast-prediction systems. This paper concerns the conceptual database design for such spatiotemporal information systems.

In order to enjoy the benefits of portability, expandability, ease-of-maintenance, and–most importantly–correctness, the design and development of an information system

[1] This research was supported in part by the Danish Technical Research Council through grant 9700780, the Danish Natural Science Research Council through grant 9400911, and the CHOROCHRONOS project, funded by the European Commission DG XII Science, Research and Development, as a Network Activity of the Training and Mobility of Researchers Program, contract no. FMRX-CT96-0056.

should follow a complete methodology cycle ([3]), including, among others, the phases of requirements analysis, conceptual and logical modeling. For conventional administrative systems, exemplified by the "supplier-supplies-parts" paradigm, the available set of modeling notations and techniques to support these phases is mature and adequate. However, this is not the case in non-standard systems managing spatiotemporal, multimedia, VLSI, image, and voice data. Rather, these pose new and unmet requirements to modeling techniques.

We aim to develop new techniques to accommodate the peculiarities of the combined spatial and temporal information that take their outset in existing ones and minimally extend them. In this paper, we present an ontological foundation for the spatiotemporal domain, covering concepts such as objects, attributes, and relationships. Based on this foundation, the paper then proposes a small set of constructs aimed at improving the ability to conveniently model spatiotemporal information at the conceptual level. These constructs may be included in a wide range of existing conceptual data models, improving their modeling capabilities without fundamentally changing the models.

We incorporate the proposed modeling constructs into the Entity-Relationship (ER) model ([5]), resulting in the semantically richer Spatio-Temporal ER (STER) model. The extension of the ER model to the STER model is just a prototypical example–other conceptual data models may be extended similarly.

The paper’s contribution is to some extent related to the *design patterns* that have been explored extensively in the context of object-oriented systems development (e.g., [15]). The proposed spatiotemporal modeling constructs effectively serve as abbreviations for patterns or small diagram fragments that occur frequently in ER diagrams when modeling spatiotemporal information. While not explored here, it would be possible to retain the patterns and not extend the ER model, but instead offer facilities in the associated ER modeling tool for directly working with the patterns.

The STER model facilitates an integration of two disjoint views ([24]) of spatial and spatiotemporal information, namely the (a) field-based view, suitable for the representation of properties like "temperature" or "vegetation" and (b) the object-based view, which captures spatial information over time in terms of identified objects, like "landparcel". More specifically, fields and objects appear together in the same graphical representation, as well as in the textual syntax of STER, and can be combined to express restrictions of fields in objects’ positions, such as "vegetation of landparcel."

A wide range of aspects of spatial and temporal databases have been studied in isolation for more than a decade. Only more recently has the combination of spatial and temporal data, i.e., spatiotemporal data, been subject to scrutiny, and we are not aware of any other work dealing with spatiotemporal modeling at the conceptual level. Below, we review the most related works, point out the relation to the paper's contribution.

[19] propose a design support environment for spatiotemporal databases, focusing on the integration of time with application data. Temporal classes, events, and states are emphasized as components of an ideal environment that better supports the developer's better understanding of spatiotemporal data and applications. [2] present a generic model consisting of objects, states, events, and conditions for explicitly representing links within a spatiotemporal GIS. This work focuses on the interaction among the components over time rather than on presenting them as the appropriate mechanisms for representing spatiotemporal data. [25] proposes a unified model for spatiotemporal data with two spatial and two temporal dimensions (database and event times). This model is not aimed at conceptual design, but is a mathematically-oriented model. [6] present a set of design patterns for spatiotemporal processes expressed in an object-relationship data model. The focus is on the analysis of spatiotemporal processsses and on the properties of object-oriented and entity-relationship data models. This work focuses on the design of processes rather than on the design of modeling constructs. Moreover, spatial issues such as the representation of time-varying fields (or spatial attributes) are not considered. Finally, mathematical models have been developed based on a systems perspective ([16]) with the purpose of understanding and, sometimes, predicting a wide variety of natural phenomena. Such models almost always incorporate notions of change over time.

The research reported here integrates previous, independent studies of spatial and temporal aspects of data by the authors. [12] [21] analyze and present requirements posed by geographic applications at the semantic level of design. The mathematically defined IFO model ([1]) and the Entity-Relationship model are extended with a small set of constructs handling spatial peculiarities. Based on this research, [22] apply the same set of constructs in the Object Model of the Object Modeling Technique (OMT) ([18]) and the Kappa System ([13]). These models' abilities to capture also behavior allows for the extension of their operations to better support spatial aspects. This research effort draws from previous theoretical and applied research on the subject ([23], for the development of a utility network management system; [4], for the design and implementation of the Greek cadastral system).

[9] and [10] present an exhaustive survey of temporally extended ER models and evaluate these according to a collection of criteria, pointing out the weaknesses and strengths of each model. The continuation ([11]) defines a graphical, temporally extended ER model that aims to satisfy the identified evaluation criteria.

Finally, the present paper takes into account, results from other theoretical works ([26], [14]) and surveys ([17], [20]).

The remainder of the paper is organized as follows. Section 2 describes the requirements of spatial data varying over time and gives a spatiotemporal ontological foundation based on concepts such as objects, attributes, and relationships. Then Section 3 presents a small set of modeling constructs consistent with the ontology that meet the requirements. The new constructs are incorporated into the ER model, resulting in the Spatio-Temporal ER (STER) model. STER's applicability and ease-of-use is the topic of Section 4, which gives examples taken from real applications. In Section 5, STER's semantics are given in terms of the ER model, and an annotated syntax is presented. Additionally, in Section 5.2 one of the examples of Section 4 is described syntactically. Section 6 concludes and points to directions for future research.

## 2. Spatiotemporal Information Requirements

To set the stage, a categorization of spatiotemporal information is discussed first. Then, an ontological foundation for spatiotemporal information is presented. Specifically, the concepts of objects, attributes, and relationships, with or without spatiotemporal extent, are defined. Then Section 3 presents modeling constructs and integrates these into the ER model, in a manner consistent with the ontology.

### 2.1 Types of Spatiotemporal Data and Applications

Spatiotemporal applications can be categorized according to the types of data they manage.

(a) Applications may involve *objects with continuous motion*; for example, navigational systems manage moving objects. In this type of application, objects change position, but not shape. For example, a "car" moves in a road network, but its shape remains unchanged.

(b) Applications dealing with *discrete changes of* and *among objects*. These applications involve objects located in space, whose characteristics, such as shape, as well as their position may change discretely in time. For example, "landparcels" or "rivers" change positions in cadastral applications, or neighboring "landparcels" change their common border, but these changes occur only discretely.

(c) Applications may manage objects integrating continuous *motion* as well as *changes of shape*. For example, a "storm" is modeled as a "moving" object with changing properties (e.g., intensity) and shape over time in an environmental application.

In this paper, the focus is on navigational and cadastral systems keeping track of moving vehicles and landparcels, respectively; analogies to applications of type (c) can be easily made. We will use excerpts from a real cadastral system for exemplification ([4]).

In a cadastral system, spatiotemporal objects may evolve as follows (see Figure 1).

- During census year 1980, the system records that landparcel *A* has common borders with *B* and river *R* runs through both *A* and *B*. Landparcel *A* has soil type "clay," while *B* has soil type "forest."

- In 1990, landparcel *B* was split into *B'* and *B"*. Landparcel *B'* has soil type "high-density forest" and *B"* has soil type "sparse forest."

- In 1995, river *R* changed its position and became *R'*.

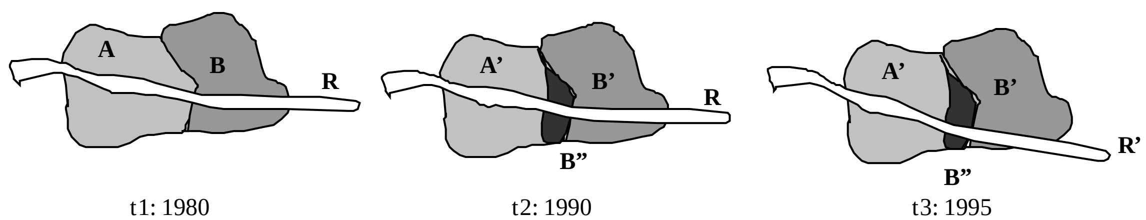



**Figure 1: Evolving spatial objects in time**.

Similarly, in a navigational system, typical queries would be: (a) what is the exact position of vehicle *v* in time *t*, given its previous position at time *t'* ? or (b) "will vehicles *v* and *v'* collide?"

It is clear from these examples that spatiotemporal information management poses new modeling requirements. The change over time of objects' positions in space must be captured along with the resulting changes of their properties (e.g., "soil type") as well as their spatial distribution. Additionally, properties such as "soil type" are involved in intricate integrity constraints as they relate to the underlying space, rather than to the objects embedded in the space. These types of properties are difficult to capture with existing modeling notations.

## 2.2 Spatiotemporal Ontological Foundations

When designing a conceptual data model, criteria such as *orthogonality* and *generality* are important design criteria that espouse, e.g., that independent concepts should be combinable in exactly every meaningful way. The ontological foundation presented next serves to explicitly state which concepts are independent and, by implication, which are not. It thus provides a guide for applying the design criteria.

*A. Objects, Attributes, and Relationships*

A database is a collection of objects $o_i$ that represent real-world entities. We will assume that each object $o_i$ belongs to an object class $O_i$. An object class $O_i$ is characterized by a set of attributes, or properties, $A_j^i$. Each attribute $A_j^i$, has a domain D($A_j^i$). Domains are implemented by data types. So, each object in a database instance has a set of attribute values, each belonging to the domain of the corresponding attribute of the object. Object classes may be interrelated via relationship sets. A relationship instance $r = <o_1,o_2,.,o_k>$ of a relationship set $R$ of degree $k$ interrelates $k$ objects from up to $n$ ($n \leq k$) different object classes.

*B. Spatial Aspects*

Most real-world objects have a *position* or a *spatial extent*. In a spatiotemporal database application, the positions of some objects matter and should be recorded, while the positions of other objects are not to be recorded. The former objects, we term *spatial*, or *geographic*, *objects* (*GO*).

The function *p* (*position*) takes spatial objects as arguments and returns the positions of the objects. Positions are parts of space and may be points, lines, regions, or combinations thereof, and are called *geometric figures*. So, function *p* is defined as follows.

$p: GO \rightarrow G$, where $G$ is the domain of geometric figures.

Objects with similar properties and the same type of geometric figures belong to the same *spatial* or *geographic object class*.

There is also a need to model the embedding *space*, in order to locate the objects in it. Space is a set, and the elements of space are called points. Many different set will do for space, but for practical reasons, space is modeled as a subset of $R^2$ or $R^3$ in current spatial applications, and we use $R^2$ as our space; this does not affect the generality of the proposed approach.

It is also an inherent property of spatial objects that their positions may be viewed at different granularities and that the granularity affects the concrete data type of the position. For example, a "landparcel" may be seen as a point, a region, or both, depending on the granularity requirements of the application at hand. Such different *object views* have to be integrated in one database schema.

Spatial objects have *descriptive attributes*, such as the "owner's name" or the "cadastral-id" of a landparcel and *spatial attributes*, such as the "soil type" of a landparcel. Spatial attributes are properties of the embedding space that indirectly become properties of the spatial objects via their position in space, i.e., the spatial objects inherit them from space. For example, although one application may view the "soil type" of a landparcel as an attribute of the landparcel, it is clear that: (a) the "soil type" is defined whether or not the landparcel exists at that position in space, and (b) when the landparcel moves (or changes shape), the landparcel's "soil type" will not remain unchanged; rather the "soil type" attribute inherits new values from the new position.

The spatial attributes of objects may be captured independently of the objects using so-called *fields* (the term *layer* is also used). Formally speaking, a field can be seen as either a function from geometric figures to a domain of descriptive attribute values or as a relationship with the geometric figure as the key attribute ([7]).

$$f_1: G \rightarrow D_1 x D_2 x \ldots x D_k$$

where $G$ is the set of geometric figures and the $D_i$ are (not necessarily distinct) domains. In other words, a field is a set of geometric figures with associated values.

There are two basic types of fields:

(a) those that are continuous functions, e.g., "temperature," or "erosion," and

(b) those that are discrete functions, e.g., "county divisions" represented as regions.

In case (a) we visualize a field as a homogeneous (or continuous) area, while in case (b) a field represents a set of areas with different values of the same attribute or positions of objects in space.

Finally, geographic objects may be related to each other in space via *spatial* (or geographic) *relationships*. For example, "the fjord Limfjorden *traverses* the city of Aalborg." Spatial relationships among geographic objects are actually relations on the objects' positions.

The set of spatial relationships is subdivided into three subsets: *topological* (e.g., "inside," "outside," etc.), *directional* (e.g., "North of," "North-East of," etc.) and *metric* (e.g., "5 km away from") relationships. Spatial relationships are further translated into spatial integrity constraints on the database. A conceptual design model should provide built-in support for representing such spatial database relationships.

*C. Temporal Aspects*

We define a spatiotemporal database as a set of objects with attribute values and relationships. The attribute values capture properties of the objects, and the relationships capture associations among the objects. In combination, the attribute values and relationships record *statements*, or *facts*, about the objects.

For example, a database of countries may include a "capital" attribute for the countries. The "Copenhagen" value of the attribute "capital" associated with "country" Denmark denotes the fact that "Copenhagen is the capital of Denmark." Precisely everything that can be assigned a truth value is a fact. For example, "Denmark is south of Greece" is a fact; it can be assigned the truth value "false." The sentence "Denmark and Greece are South" is not a fact.

Three temporal aspects have been the focus of attention in the research literature; they are universal, and applications frequently require that these be captured in the database:

(a) the *valid time* aspect applies to facts: the valid time of a fact is the time when the fact is true in the modeled reality. For example, the valid time of "Copenhagen is the capital of Denmark" is the time from year 1445 until the present day.

(b) the *transaction time* aspect applies not only to facts, but to any "element" that may be stored in a database: the transaction time of a database element is the time when the element is part of the current state of the database. Put differently, the transaction time of element *e* is the valid time of "*e* is current." Transaction time is important in applications that demand traceability and accountability.

(c) the *existence time* aspect applies to objects: the existence time of an object is the time when the object exists. Again, this aspect can be formulated in terms of valid time. The existence time of object *o* is thus the valid time of "*o* exists."

We will assume that it only makes sense for an object to have properties and participate in relationships *when the object exists*. This implies that the valid times of facts associated with objects must be contained in the existence times for those objects. The same holds for transaction time in place of valid time.

Time values are drawn from a domain of time values, with the individual values being termed *chronons*. All three temporal aspects have duration, and they may be captured using time intervals, where a time interval $[t_s, t_e]$ is defined to be a set of consecutive or chronons. We call $t_s$ and $t_e$ the start and the end chronon of the interval, respectively.

## 3. Spatiotemporal Conceptual Modeling Constructs

The following section extends the ER model in accordance with the spatiotemporal requirements and ontological foundation outlined in the previous section. More specifically, following a brief presentation of the ER model, a small set of modeling constructs is integrated into the model. Examples show the applicability of this proposal. Section 3.3 discusses the option of adding transaction time to the spatiotemporal ER model, while Section 3.4 gives STER's syntax and semantics.

### 3.1 The Entity Relationship Model

The Entity Relationship (ER) model ([5]) is arguably the first conceptual model that appeared in the literature. This easy-to-use model, consisting of very few modeling constructs, has gained an unparalleled, widespread popularity in industry. The model's basic constructs include the following: (a) entity sets that represent object classes and are depicted by rectangles; (b) relationship sets that represent associations among entity sets and are illustrated as diamonds; and (c) attributes of entities and relationships that capture properties of the objects and associations and are represented graphically as ovals. Relationships can be 1:1 (one to one), 1:M (one to many), and N:M (many to many). Some variations of the ER model include the special kind of relationship set, ISA, that is used to model one entity set being a subset of another; entities in the subset inherit all the properties and associations specified for the superset.

### 3.2 Extending the ER with Spatiotemporal Conctructs[2]

In order to conveniently capture spatial and temporal aspects of information, the spatiotemporal concepts presented in Section 2 are applied to the ER constructs, resulting in a new data model that we will term the Spatio-Temporal ER (STER) model. STER includes constructs with built-in spatial, temporal, and spatiotemporal functionality. A construct that captures a temporal aspect is called *temporal*; if it has built-in support only for a spatial aspect, it is termed *spatial*; and if it has both, it is *spatiotemporal*. The upper-right corner of each extended construct indicates its temporal support. The bottom-right corner indicates the

---

[2] Here, we explore all the constructs of the ER model that gain special meaning when they are related to space and time. Complex constructs such as "spatial aggregation" and "spatial grouping" are not discussed. For a specialized investigation on these, see [12].

spatial support. For each STER construct, we give its corresponding representation in the ER model.

While all basic constructs of the ER model can have spatial and/or temporal extent, not all types of time can be assigned to each construct. There is a semantic explanation for this: existence time indicates the existence of "something," and this has only meaning for identifiable ontologies or, in other words, entities (or objects). An entity set may be given attributes that describe the properties of the set's entities. In the ontology, we stated that valid time is meaningful only for *facts*. When assigning valid time to an attribute of an entity set, we indicate that the valid times of the facts, that entities in the set are associated with specific values for this attribute, are to be captured in the database. The same applies to attributes of relationship sets in place of entity sets. Finally, valid time may be assigned to a relationship set, indicating that the time when each relation in the set is true in the miniworld is to be captured in the database. Transaction time applies to any "element" stored in the database, regardless of whether or not it may be assigned a truth value. So unlike valid time, transaction time applies to entity sets. Table 1 shows the meaningful combinations of temporal aspects and modeling constructs.

| | **entity sets** | **attributes** | **relationship sets** |
|---|---|---|---|
| **existence time** | Yes | No | No |
| **valid time** | No | Yes | Yes |
| **transaction time** | Yes | Yes | Yes |

**Table 1: Assigning temporal aspects to ER constructs.**

As the next step, we illustrate in more detail how to assign existence and valid time to the ER constructs. Transaction time is covered separately, in the next section. The abbreviations "et," "vt," "tt," and "bt" are used for "existence time," "valid time," "transaction time," and "bitemporal time," respectively. Abbreviation "bt" is a shorthand for the combination of "vt" and "tt" that occurs often in a spatiotemporal database.

*A. Entity Sets*

Entity sets represent object classes.

*(i) Temporal Entity Sets*

Entity sets can be assigned existence and transaction time. Support for existence time for an entity set is indicated by placing an "et" in a circle in the upper-right corner for the entity set's rectangle as indicated in Figure 2. Figure 2(b) shows that for car entities we keep track of their existence time.

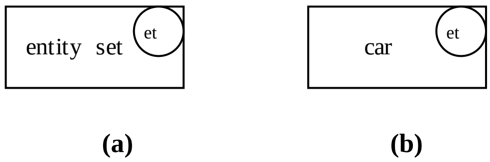



**Figure 2: (a) capturing existence time in STER, (b) representing car entities and their existence time.**

This notation is in effect a shorthand for a larger ER diagram. This shorthand is convenient because it concisely states that the existence times of the entities in the entity set should be captured in the database. The more verbose ER diagram corresponding to the STER diagram in Figure 2(a) is given in Figure 3. Attributes connected to each other denote composite attributes ([8]). Thus "time/id" values consist of pairs of "time" and "id" values.

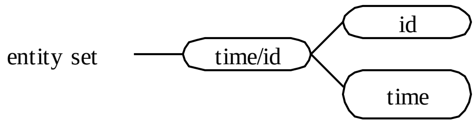



**Figure 3: Capturing existence time in the ER model.**

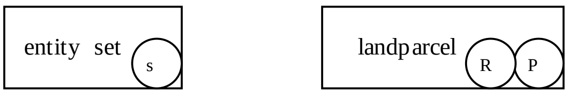
*(ii) Spatial Entity Sets*

Spatial objects have a position in space, and it is frequently necessary to capture this position in the database. The first step to support this is to provide means for representing the space in which the objects are embedded. The next is to provide means for indicating that the objects' positions in this space is to be captured. For these purposes, we introduce the following special entity and relationship sets.

(a) The special entity sets SPACE, GEOMETRY, POINT (or "P"), LINE (or "L"), REGION (or "R"). Entity set GEOMETRY captures the geometrical position of the entity set and can be (i.e., is-a) POINT, LINE, REGION, or any other geometric type (or geometry). For simplicity we use only POINT, LINE, REGION, and their combinations.

(b) The special relationship set "is_located_at" that associates a spatial entity set with its geometry. The cardinality of this set is 1:M because a spatial entity may have more than one geometry when multiple granularities are employed. The relationship set "belongs_to" between GEOMETRY and SPACE with cardinality constraint M:1 is also included.

The letters "s," "P," "L," or "R" in a circle in the lower-right corner of an entity set rectangle specifies the spatial support. Letter "s" stands for SPATIAL and is used to indicate a spatial entity set whose exact geometric type is unknown. Letters "P," "L," "R," and their combinations specify geometric types as indicated above. These annotations may occur simultaneously and represent then different views of the same object class. A spatial entity set is depicted as shown in Figure 4(a), and its meaning in terms of the ER model is given in Figure 5. Figure 4(b) illustrates the spatial entity set "landparcel" with simultaneous geometries point and region; in this case, the representation in the ER model will have only REGION and POINT as geometries.




**(a)** **(b)**

**Figure 4: (a) spatial entity sets in STER, (b) a landparcel as POINT or REGION.**

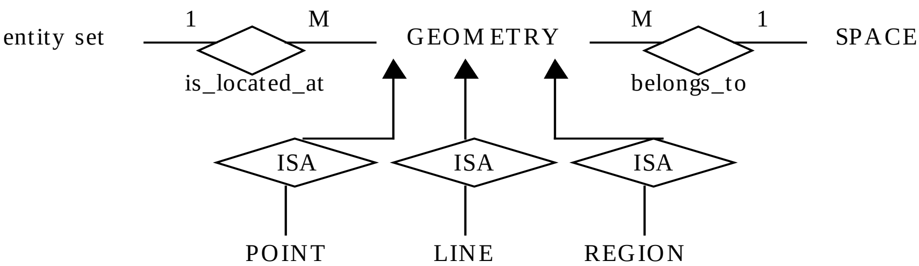



**Figure 5: Representation of spatial entity sets in the ER model.**

*(iii) Spatiotemporal Entity Sets*

When an object changes its position over time, it is the geometry that changes rather than the object itself. To capture a temporal aspect of the positions of the objects in an entity set, an "svt," an "stt," or an "sbt" is placed in a circle in the lower-right corner of the entity set's rectangle. The first annotation indicates valid-time support: the objects' current positions as well as their past and future positions are to be captured. This is illustrated in Figure 6(a), while Figure 6(b) gives an example. The second annotation (i.e., "stt") indicates transaction-time support: the current positions as well as all positions previously recorded as current in the database are to be captured. The third annotation (i.e., "sbt") indicates support for both valid and transaction time. If the geometric type of the entity set is known, the "s"-part is replaced by "P," "L," "R," or a combination of these.


**(a)** **(b)**

**Figure 6: (a) a spatiotemporal entity set with valid time support, (b) recording car's position in time.**

The meaning of the spatiotemporal entity set in Figure 6(a) is given in Figure 7 in terms of the ER model. So, referring back to the example of Figure 4(b), we now record a "city" as a REGION and POINT in time (i.e., valid time).

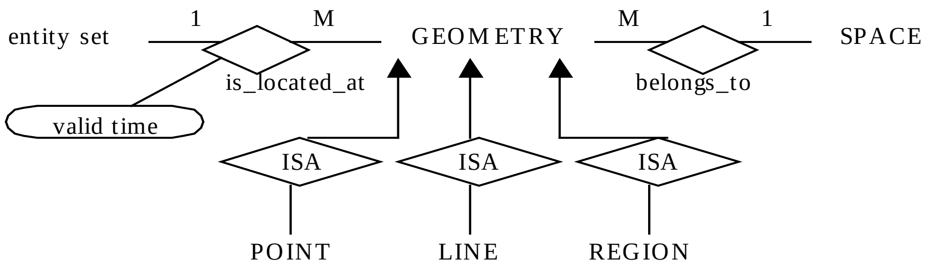



**Figure 7: ER-model representation of Figure 6(a).**

It is important to point out the difference between keeping track of (a) a spatial entity set in time, and (b) the position of a spatial entity set in time. Figure 8(a) shows an entity set "landparcel" as a spatial entity set (indicated by an "s"), for which a cadastral database captures transaction time (indicated by "tt"). In contrast, Figure 8(b) illustrates an entity set "landparcel," for which the database captures transaction time as before (indicated by "tt"), as well as its position over time (valid time, indicated by "svt").


**(a)** **(b)**

**Figure 8: (a) a landparcel in space and time, (b) a landparcel in time, with position in time.**

*B. Attributes of Entity and Relationship Sets*

As mentioned in Section 2.2, entity sets have two types of attributes: (a) *descriptive* attributes, such as the "cadastral-id" of a landparcel, and (b) *spatial* attributes, such as the "soil type" of a landparcel. The values of descriptive attributes for an entity (or a relationship) often change over time, and it is often necessary to capture this in the database. Spatial attributes with a temporal dimension are termed *spatiotemporal attributes*.

*(i) Temporal Descriptive Attributes*

Values of attributes of entities denote facts about the entities and thus have both valid- and transaction-time aspects. A circle with a "vt" or a "tt" in the upper-left corner of an oval denoting an attribute indicates that valid or transaction time, respectively, is to be captured. A circle with "bt" (bitemporal) indicates that both temporal aspects are to be captured. The sample STER diagram in Figure 9(a) contains an attribute with valid-time support, and Figure 10 gives the equivalent ER diagram. Figure 9(b) shows an example keeping track of cars' colors and their valid-time periods.

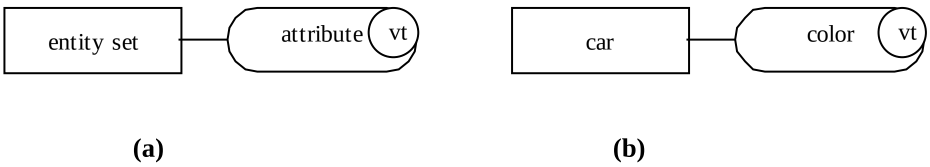



**Figure 9: (a) an attribute in STER with valid time support, (b) car "color" with valid time.**

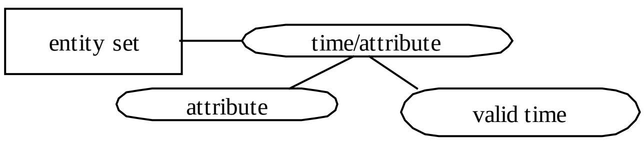



**Figure 10: The STER diagram of Figure 9(a) in the ER model.**

*(ii) Spatial Attributes*

Facts captured by attributes may also have associated locations in space, which are described as sets of geometric figures (Section 2). To capture this spatial aspect of an attribute, a circle with an "s" is used, as shown in Figure 11. Figure 11(a) depicts the general representation of a spatial attribute, while 11(b) shows that the "soil type" value of a landparcel is associated with a set of spatial regions ("R").

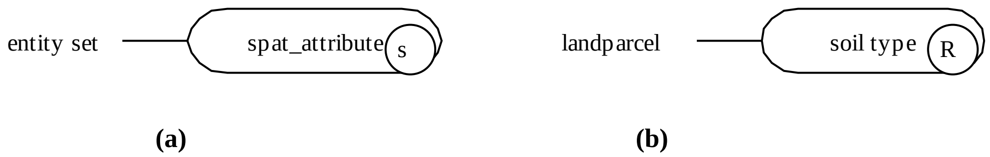



**Figure 11: (a) a spatial attribute in STER, (b) "soil type" as a spatial attribute.**

In terms of the ER model, a spatial attribute is modeled as an entity set connected to SPACE via the relationship set "has_spatial_attribute," see Figure 12. In this way, each part of space is assigned a specific value of the attribute. By connecting a spatial entity set to GEOMETRY (via the special relationship "is_located_at," see previous figures) and GEOMETRY to SPACE (via "belongs_to"), an object inherits spatial attributes. In other words, spatial attributes of entities are *derived properties* from space.

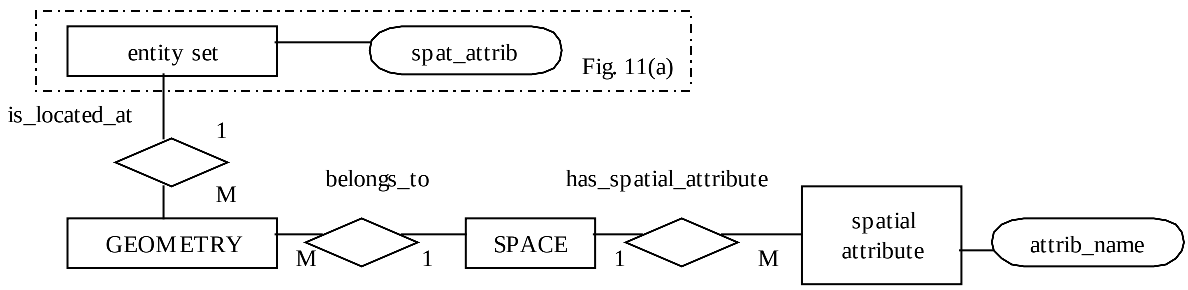



**Figure 12: Representation of spatial attributes in the ER model.**

*(iii) Spatiotemporal Attributes*

The temporal aspects (valid and transaction time) of spatial attributes are recorded by placing "svt," "stt," or "sbt" (and replacing the "s" with "P," "L," o "R," or a combination of these if the geometric type of the geometric figures of the attributes is known) in same way as for entity sets. This is illustrated in Figure 13(a), and Figure 14 gives the equivalent ER diagram. Figure 14(b) gives an example.

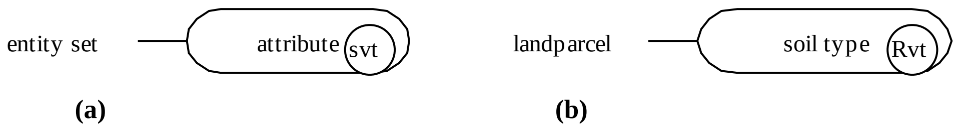



**Figure 13: (a) spatiotemporal attribute in STER, (b) "soil type" as a spatiotemporal attribute.**

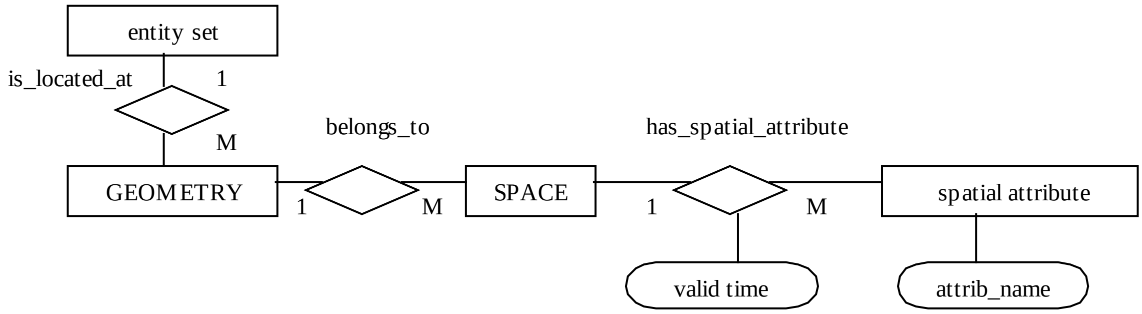



**Figure 14: ER diagram corresponding to the STER diagram in Figure 13.**

*C. Relationship Sets*

*(i) Temporal Relationship Sets*

By annotating a relationship set with a temporal aspect (valid time, transaction time, or both), we capture the changes of that temporal aspect for the set's relationships.

*(ii) Spatial Relationship Sets*

Spatial relationship sets are special kinds of relationship sets. In particular, they are associations among the geometries of the spatial entities which, for reasons of simplicity and ease of understanding, are described as relationships among the spatial entity sets themselves. For example, the relationship "traverses" between cities and rivers relates the geometries of entities of these two spatial entity types.

*(iii) Spatiotemporal Relationship Sets*

A spatiotemporal relationship set is a spatial relationship set with time support. In particular, by annotating a spatial relationship set with a temporal aspect we capture the changes of the spatial relationship over time. Figure 15(a) shows the general representation of a spatiotemporal relationship set, while Figure 15(b) depicts changes of the relationship "traverses" between cities and rivers are recorded in time. Figure 16 gives the equivalent of 15(a) in terms of ER diagram.

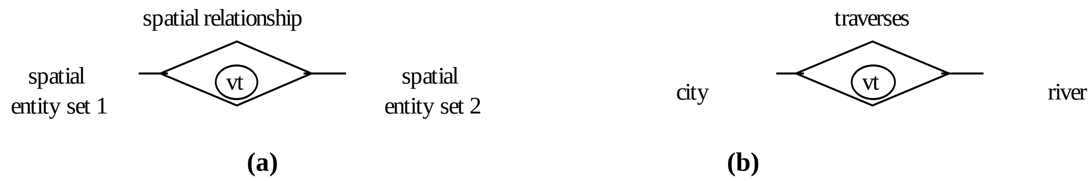



**Figure 15: (a) spatiotemporal relationship sets in STER, (b) "traverses" as a spatiotemporal relationship.**

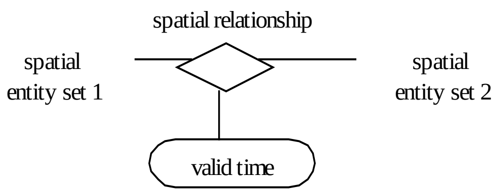



**Figure 16: Detailed representation of spatiotemporal relationship sets in STER.**

Finally, the previous discussion about temporal, spatial and spatiotemporal attributes applies also to attributes of relationship sets.

### 3.3 Adding Transaction Time to STER

In the description of STER so far, we have primarily focused on adding existence and valid time to the ER model constructs. The temporal aspect of transaction time is not strictly part of capturing the modeled reality. However, capturing transaction time, as reflected in the systems requirements, is important when designing a real-world information system, and an ER model should also provide built-in support for the capture of this aspect.

Transaction time is applicable in STER exactly everywhere where valid time or existence time is applicable. To capture the transaction time of a relationship set (Figure 15), all that is needed is to add "tt" to the relationship set construct.

## 4. Examples of Usage

The following examples present small excerpts of the conceptual schemas for a cadastral application ([4]) and a navigational system, designed in the STER model. Spatial (derived) attributes of entity sets are shaded. The goal is to give a more complete picture of the use of the STER diagram with many spatiotemporal constructs, for better understanding.

The diagram in Figure 17 states that for "landparcels," existence time is recorded; the positions of landparcels can be either points or regions. Moreover, "landparcels" have a spatiotemporal attribute "soil type," for which both valid and transaction time is recorded. "soil type" is of type set REGIONS ("R"), and both valid and transaction time for the soil types of landparcels are captured. Moreover, the "bt" in the upper corner of "soil type" shows that we also record "soil type" as descriptive attribute of "landparcel" and we keep track of both valid and transaction time. Additionally, "landparcels" may be traversed by "rivers." For the spatial relationship "traverses," we capture transaction time. For "rivers," transaction and existence time is captured. Finally, "rivers" are represented as lines ("L").

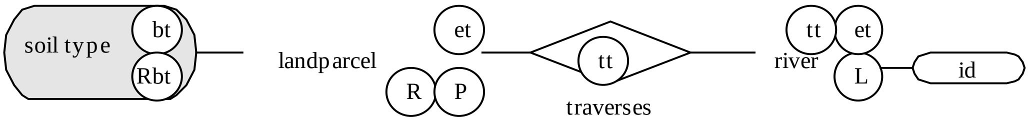



**Figure 17: Using STER to model an excerpt of a cadastral application.**

The next figure illustrates cars modeled in STER using a spatiotemporal entity set. A "car" entity has an existence time and a transaction time. A "car" also has a position in space, which is a point, for which transaction and valid time is captured ("Pbt"). Finally, cars have license plates, for which transaction time is supported. A moving car "follows" a "route", and we capture the transaction time of relationship set between cars and routes, as well as the transaction and existence time of the route itself. A "route" is a line ("L").

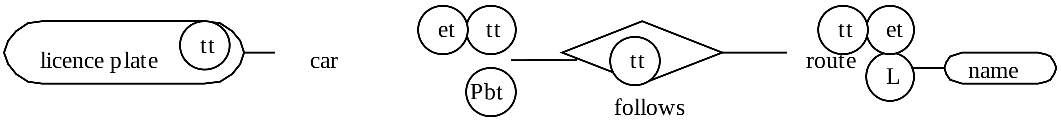



**Figure 18: An excerpt of a spatiotemporal application designed in STER.**

## 5. A Textual Syntax of STER

A tool that supports the use of a graphical model such as STER maps the diagrams designed by the user into an internal, textual representation. As a step towards the realization of an STER design tool, this section outlines a textual representation of STER. Specifically, Section 5.1 gives a textual representation for entity and relationships sets in the STER model and informally describes the semantics. Section 5.2 takes one of the examples presented in Section 4 and describes its translation into the textual representation.

### 5.1 The Syntax

The definitions use the following meta syntax. A `typewriter-like` font indicates elements of the data definition language; syntactic definitions are given with the normal font; *italics* are used for semantic explanations. Upper case words are reserved words, and lower case words are variables that represent arbitrary names. For example, `attr_name`$_i$ stands for any alphanumeric string that is not a reserved word. Capitalized words in lower case denote variables with restricted range. For example, "`Domain`" is one of INTEGER, STRING, DATE, etc. Optional elements are enclosed in "< >"'s, and "(, ...)" denotes repeatable elements; the notation { ... | ... } denotes selection (one of), and "( )" simply indicates grouping of arguments. A longhand notation (e.g., DEFINE instead of DEF) is used to facilitate a first reading; in practice one uses the shorthand.

### Definition of entity sets

```
DEFINE ENTITY SET entity_set_name
<TYPE Entity_construct (entity_set_name_i,…)>
<ATTRIBUTES
   (attribute_name_j
   <AGGREGATION_OF (attribute_name_k
                      <VALID TIME Valid_time>
                      <TRANSACTION TIME Transaction_time>,…)>
   <VALID TIME Valid_time>
   <TRANSACTION TIME Transaction_time>
   <GEOMETRY Geometric_type
           <VALID TIME Valid_time>
           <TRANSACTION TIME Transaction_time>>,…)>
<GEOMETRY Geometric_type
           <VALID TIME Valid_time>
           <TRANSACTION TIME Transaction_time>>
<EXISTENCE TIME Existence_time>
<TRANSACTION TIME Transaction_time>
<AS ISA OF (entity_set_name_l,…)> >
```

- `entity_set_name`, is an identifying alphanumeric string, i.e., different from any other used in the same syntactic position.
  *It is used as the name of the entity set being defined.*
- `Entity_construct` is one of PART_OF, GROUP_OF, SPATIAL_PART_OF, and SPATIAL_MEMBER_OF.
  *It is optional; it is used to define entity's complex type.*
- `entity_set_name`$_i$ is an identifying alphanumeric string. `i` is an integer.
  *It is used to define the constructs of the complex entity set.*
- `attribute_name`$_j$ is an identifying alphanumeric string. `j` is an integer.
  *It is used to define the name of the attribute of an entity set.*
- `attribute_name`$_k$ is an identifying alphanumeric string. `k` is an integer.
  *It is used to define the name of the attributes which composite a complex.*
- `Geometric_type` is one of P, L, or R or combination thereof.
  *This optional clause is used to define the geometric type of the entity set or the attribute (spatial) defined.*

- `Valid_time` is a time interval.

  *It is used to record the valid time of the corresponding attribute ($k^{th}$ or $j^{th}$) or geometry of the defined entity set or attribute.*

- `Transaction_time` is a time interval.

  *It is used to record the transaction time of the corresponding attribute ($k^{th}$ or $j^{th}$) or geometry of the defined entity set or attribute.*

- `Existence_time` is a time interval.

  *It is used to record the existence time of the corresponding entity set.*

- `Transaction_time` is a time interval.

  *It is used to record the transaction time of the corresponding entity set.*

- `entity_set_name`$_i$ is an identifying alphanumeric string and `i` is an integer.

  *It defines pre-existing classes used to construct the superset.*

**Definition of relationship sets**

```
DEFINE RELATIONSHIP SET relationship_set_name (entity_set_name_i,…)
TYPE Relationship_type
<ATTRIBUTES (attribute_name_k
           <VALID TIME Valid_time>
           <TRANSACTION TIME Transaction_time>,…)>
<VALID TIME Valid_time>
<TRANSACTION TIME Transaction_time>
```

- `relationship_set_name` is an identifying alphanumeric string.

  *It defines the functional relationship between entity sets.*

- `entity_set_name`$_i$ is an identifying alphanumeric string and `i` is an integer.

  *It defines entity sets which are related through the relationship.*

- `Relationship_type` is one of `ONE_TO_ONE`, `ONE_TO_MANY`, `MANY_TO_ONE`, `MANY_TO_MANY`.

  *It is used to define the relationship type.*

- `attribute_name`$_j$ is an identifying alphanumeric string and `j` is an integer.

  *It is used to relate possible attributes to the defined relationship.*

- `Valid_time` is a time interval.

  *It is used to record the valid time of the corresponding attribute, or relationship.*

- `Transaction_time` is a time interval.

*It is used to record the transaction time of the corresponding attribute or relationship.*

### 5.2 Example of Usage

Assume the example of Figure 17 of Section 4. The graphical diagram can be translated into the following textual description, provided by the textual syntax of STER. We informally explain the syntax line by line, and in close connection to what is depicted in Figure 17. We express that the entity set "landparcel" has "soil type" as attribute. For this, we record valid and transaction time, as well as the valid and the transaction time of its geometry, which is REGION. "landparcel" is of type REGION or POINT and we keep track of its existence time. Additionally, we have the entity set "river" keeping track of its existence and transaction time in the database. A "river" is of type LINE and has its identification ("id") as attribute. Landparcels and rivers are related via the spatial relationship "traverses," for which we record the transaction time ("tt") of the changes.

```
DEFINE ENTITY SET landparcel
ATTRIBUTES
   (soil type
    VALID TIME Valid_time>
    TRANSACTION TIME Transaction_time
      GEOMETRY REGION
            VALID TIME Valid_time
            TRANSACTION TIME Transaction_time)
GEOMETRY POINT, REGION
EXISTENCE TIME existence_time

DEFINE ENTITY SET river
ATTRIBUTES (id)
GEOMETRY LINE
EXISTENCE TIME existence_time
TRANSACTION TIME Transaction_time

DEFINE RELATIONSHIP SET traverses (landparcel, river)
TYPE MANY_TO_MANY
TRANSACTION TIME Transaction_time
```

## 6. Conclusions and Further Research

We view the design of spatiotemporal applications from the perspective of information systems development. Focusing on the special needs of spatiotemporal applications and based on an explicit ontological foundation, we have defined a small set of spatiotemporal conceptual modeling constructs. We employed the ER model as the concrete context for presenting the new constructs. The resulting Spatio-Temporal Entity-Relationship model captures more conveniently spatiotemporal aspects of information, which are prevalent in many information systems, than does the ER model. The modeling constructs may be added to a range of semantic models, thus improving also these models' ability to conveniently model spatiotemporal information.

STER was applied in a real, large-scale application ([4]), as well as in small prototypical examples, with quite encouraging results: diagrams were easy to understand by the users and the data modeling process proceeded more rapidly.

This work can also be seen from the perspective of design patterns for conceptual modeling for spatiotemporal applications. The set of constructs which are presented to handle spatiotemporal needs, can be considered as design patterns and may be incorporated into the design tools associated with existing conceptual models, not extending the models.

Some interesting directions for future research are drawn at this point: since the conceptual design level is followed by the logical, transformation rules must be produced for the conversion of STER diagrams to the relational model or to different implementation platforms at the logical level, such as object-oriented ones. This will lead to the construction of a Computer Aided Software Engineering (CASE) tool for spatiotemporal applications for the semi-automatable transition from the conceptual design to physical implementation.

## 7. References


1. Abiteboul, S., and Hull, R., 1987. *IFO: A Formal Semantic Database Model*. ACM TODS, 12(4):525-565.
2. Allen, E., Edwards, G., and Bedard, Y., 1995. *Qualitative Casual Modeling in Temporal GIS*. Proceedings of the International Conference on Spatial Information Theory (COSIT'95), Lecture Notes in Computer Science (988):397-412.
3. Brodie, M.L., 1982. *On The Development of Data Models*. In *On Conceptual Modeling*, Brodie, M.L., Mylopoulos, J.and Schmidt, W.J. (editors), (New York: Springer-Verlag).

4. Cadastral, 1997. *Definition of a Standard for the Exchange of Digital Cadastral Data*. National Technical University of Athens funded by CADASTRE SA, Greece.

5. Chen, P.S., 1976. *The Entity-Relationship Model: Toward a unified view of Data*. ACM TODS, 1(1):9-36.

6. Claramunt, C., Parent, C., and Theriault, M., 1997. *Design Patterns for Spatiotemporal Processes*. IFIP 1997. Chapman and Hall.

7. Delis, V., Hadzilacos, Th., and Tryfona, N. 1995. *An Introduction to Layer Algebra*. Proceedings of the 6th International Conference on Spatial Data Handling (SDH), Edinbrugh, UK, Taylor & Francis.

8. Elmasri, R., and Navathe, S., 1994. Fundamentals of Database Systems. Benjamin/Cummings.

9. Gregersen, H., and Jensen, C.S., 1997a. *Temporal Entity-Relationship Models–a Survey*. Submitted for publication.

10. Gregersen, H., Jensen, C., and Mark, L., 1997b. *Evaluating Temporally Extended ER Models*. In K. Siau, Y. Wang, and J. Parsons (eds.) Proc. of the 2nd CAiSE/IFIP 8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design, Barcelona.

11. Gregersen, H., and Jensen, C., 1998. *Conceptual Modeling of Time-Varying Information*. Working paper.

12. Hadzilacos, Th., and Tryfona, N., 1997. *Extending the Entity-Relational Model to Capture Spatial Semantics*, SIGMOD RECORD, 26,3.

13. IntelliCorp Inc., 1993. *Kappa Manual Set*, IntelliCorp Inc., 1975 El Camino Real West, Mountain View, CA 94040-2216, USA.

14. Nguyen, V.H., Parent, C., and Pirker, A., 1997. *Madstra: A Schema Translator for the Spatial Data Model MADS*. Technical Report EPFL-DI-LBD, June 1997.

15. Pree, W., 1994. *Design patterns for Object-Oriented Software Development*. Addison-Wesley.

16. Roberts. N., Andersen, D., Garet, M., and Shaffer, W., 1983. *Introduction to Computer Simulation: A System Dynamics Modeling Approach*. Addison-Wesley.

17. Roddick, J.F., and Patrick, J.D., 1992. *Temporal Semantics in Information Systems-a Survey*. *Information Systems*. 17,3.

18. Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., and Lorensen, W. 1991. *Object-Oriented Modeling and Design*. Prentice-Hall: Englewood Clifs, NJ.

19. Story, P.A., and Worboys, M.F., 1995. *A Design Support Environment for Spatio-Temporal Database Applications*. Proceedings of the International Conference on Spatial Information Theory (COSIT'95), Lecture Notes in Computer Science (988):413-430.
20. Theodoulidis, C.I., and Loucopoulos, P., 1991. *The Time Dimension in Conceptual Modeling*. Information Systems, 16(3):273-300.
21. Tryfona N., and Hadzilacos, Th. 1995. *Geographic Applications Development: Models and Tools at the Conceptual Level*, 3rd ACM Workshop in Geographic Information Systems, Baltimore, MD, pp:91-28
22. Tryfona, N., Pfoser, D., and Hadzilacos, Th., 1997. *Modeling Behavior of Geographic Objects: An Experience with the Object Modeling Technique*. Proceedings of the 9th International Conference on Advanced Information Systems and Engineering (CAiSE'97), Lecture Notes in Computer Science (1250):210-221.
23. UtilNets, 1994. Work Program of Brite-Euram Project 7120, DG XII, European Union, Brussels, Belgium.
24. Worboys, M., 1994a. *Object-Oriented Approaches to Geo-referenced information.* International Journal of GIS, 8,4.
25. Worboys, M., 1994b. *A Unified Model for Spatial and Temporal Information*. The Computer Journal, 37,1.
26. Zimanyi, E., Parent, C., and Spaccapietra, S., 1997. *TERC+: A Temporal Conceptual Model*. International Symposium on Digital Media Information Base (DMIB'97), Japan.