Requirements, Definitions and Notations for Spatiotemporal Application Environments

Dieter Pfoser Nectaria Tryfona

Department of Computer Science
Aalborg University, Fredrik Bajersvej 7E,
DK-9220 Aalborg Øst, Denmark

{pfoser,tryfona}@cs.auc.dk

1. ABSTRACT

Modeling spatiotemporal applications is a complex task involving intricate issues, such as the representation of objects' position in time, and spatial attributes that change values depending on specific locations in time periods. Due to this complexity, the analysis of users' requirements–as the first phase of an application development methodology–is often neglected, focusing, mainly, on physical design aspects. In this paper, we address the set of spatial, temporal and spatiotemporal concepts as they are drawn from users' needs. The goal is to support the developer's better understanding about spatiotemporal applications, by providing the concepts and the notations needed in such environments; these concepts will, later, be translated into specific constructs and implementation issues. More specifically, *space*, *spatial objects*, *spatial attributes*, *fields*, *time* and *models of time* are presented and then *combined* to accommodate spatiotemporal peculiarities, resulting into the new, spatiotemporal, concepts of *snapshots*, *changes*, and *versions* of objects and maps, *motion* and *phenomena*. Examples taken from two real large-scale applications show the necessity and adequacy of the presented concepts.

1.1 Keywords

spatiotemporal requirements, space, spatial objects, spatial attributes, fields, time, models of time, snapshots, changes, versions of objects and maps, phenomena.

2. INTRODUCTION

Spatiotemporal applications enjoy a lot of attention over the last years. The combination of space and time, expressed by–just to mention a few–the need to observe objects' motion in space over time and then predict possible routes of them, or by studying phenomena, such as a "storm," which move over time, appears emerging.

In the last decade, the information systems, and more specifically, the database research community, gave valuable results in modeling and retrieving, on the one hand, spatial objects [9] [15] [23], and on the other, objects in a temporal framework [10] [11] [12] [16]. Some other efforts fall in the spectrum of the spatiotemporal systems, i.e., deal with combined spatial and temporal information: [17] proposes a design support environment for spatiotemporal databases focusing on the integration of time application data. [1] presents a generic model consisting of objects, states, events and conditions for explicitly representing casual links within a spatiotemporal GIS. [23] proposes a unified model for information which is referenced to two spatial dimensions and two temporal dimensions (database and event times). [6] presents a set of design patterns for spatiotemporal processes expressed in an object-relationship data model. Finally, in this spectrum, domain experts give their own solutions and modeling techniques to these issues, mainly driven by the need for fast and applicable answers. The result is spatiotemporal systems tightly coupled with specific software and hardware (see for example [2] [3] [24] for the use of Arc/Info in environmental modeling).

It is clear that despite the many efforts, the spatial and temporal research areas have not, yet, met satisfactorily. The main reasons lie in the complexity of their components: space by itself is a complex and intricate issue, involving, among others, position of objects, and spatial attributes that change values depending on specific locations. On the other hand, research on temporal issues is related mainly to physical design aspects and improvements of systems performance [13].

In this paper we address the concepts involved in a spatiotemporal environment, as they are drawn in users' requirements. The goal is to support the developer's better understanding of spatiotemporal applications, by providing the concepts and the notations needed in such environments; these concepts, will later be translated into specific constructs and implementation issues. More specifically, the concepts of *snapshots*, *changes*, and *versions of objects* and *maps*, *motion* and *phenomena* are presented and then combined to accommodate spatiotemporal needs.

The contribution of this work is described as follows:

(a) The recording and presentation of user requirements for spatiotemporal applications, based on the experience from two real large-scale projects ([22] for the design and development of a network utility management system, and [5] for the design of a cadastral database). We see the users' requirements analysis as one of the phases of an application development methodology, including, among others, conceptual and logical modeling, and implementation. This phase is essential, as methodological and correct analysis of requirements leads to a set of concepts, necessary for the representation of the application's information. It is important to point out that we only deal with *modeling requirements*. We are not aware of any relevant work at this level; no recording of spatiotemporal users' requirements exists in literature.

(b) An exhaustive presentation and discussion of the necessary spatial and temporal concepts one should take into account, when he/she deals with the modeling of user requirements. Here, we clarify concepts and expressions for further communication without ambiguities.

(c) The combination of the spatial and temporal concepts, resulting into *new* spatiotemporal concepts, such as *snaphots*, *changes* and *versions* of objects and maps, *motion* and *phenomena*. Then, we show how these concepts meet the previously listed user needs.

The rest of the paper is organized as follows: Section 3 discusses the set of requirements for a spatiotemporal application environment. In Section 4, the spatial, temporal and spatiotemporal concepts are presented and analyzed, and formal expressions are given; moreover, analogies between the spatial and temporal dimension are presented. Section 5 shows how these concepts meet the modeling requirements of spatiotemporal applications. We conclude, in Section 6, with the summary of this work and future research plans.

3. SPATIOTEMPORAL APPLICATIONS

Spatiotemporal applications fall into the category of data intensive applications, often referred to as "non-standard", including, among others, multimedia, VLSI design, and artificial intelligence based systems. They differ from business data processing–exemplified by the "supplier-supplies-parts" paradigm–in a variety of ways, centered around the support of complex objects and relationships among them, and long transactions. In addition, spatiotemporal applications deal with objects whose *position* in space, as well as the *change* of it, over time, *matters*. Modeling space is, conceptually, an open problem, and computationally very difficult. As a consequence to be positioned in space, each object is related to every other in complex ways, partially captured by spatial relationships. To make things even more complicated, there is the need to record past states such as history of "forest fires," and to study geographical phenomena, such as the birth, motion and death of a "storm" over time.

Three different kinds of spatiotemporal applications can be distinguished, based on the type of data they manage:

(a) applications dealing with *moving* objects, such as navigational; in these, objects change position in time, for example, a moving "car" on a road network,

(b) applications involving objects located in space, whose characteristics, as well as their position, may *change* in time; for example, in a cadastral information system, "landparcels" change positions by changing shape, but they do not "move," and

(c) applications dealing with objects which integrate the above two behaviors; for example, in environmental applications, "pollution" is measured as a *moving* phenomenon which *changes* properties and shape over time.

For the design and implementation of such complex environments, it is important to follow structured application development methodologies; the benefits of using them (ease of maintenance, portability, and expandability among others) have been praised in theoretical [4] [7], as well as in practical works [14] [22]. A typical application development methodology includes–among others–the phases of requirements analysis, conceptual and logical modeling and implementation. Models, tools and techniques are available in each phase for the transition to the next one. Requirements analysis plays an essential role, as its main objective is to collect the set of concepts and operations on them that are involved in an application environment, based on description given by the users. Later, these concepts are translated into specific modeling constructs provided by semantic and logical models, and finally into implementation issues and details.

In the next section we give a set of spatiotemporal modeling requirements, taking examples from the above three application categories. Later (in Section 4), we analyze and combine the concepts involved into these requirements.

3.1 Spatiotemporal Modeling Requirements

Based on theoretical research [19] [21] as well as applied experience ([22] for the design and development of a network utility management system, and [5] for the design of a cadastral database) the following set of spatiotemporal requirements, at the modeling level, is drawn:

- Need for representation of objects with position in space and existence in time. Consider the example taken from a utility management information system in which "waterpipes" are objects occupying certain parts of space at certain time periods.

- Need to capture the *change of position* in space over time. Depending on the type of application we are interested, two different changes of positions matter:

  (a) *continuous change*, which results into *motion*. For example, in a navigational system, the continuous change of a "vehicle's" position.

  (b) *discrete change*, such as, the *change* of the shape of objects over time. An example taken from a cadastral system [5] reflects this need: a "landparcel" has a position in space at some point in time. When it changes *shape* (e.g., a new part of land is attached to it) its position changes; however, it is not considered a moving object, as in the previous case. Shape is one of the four components, which fully and non-redundantly define position; the others are *size*, *orientation* and *centroid*. Similarly to shape, the other three components of position can change over time.

- Need for the definition of attributes of space (*spatial attributes*) and organization of them into *layers* or fields, i.e., thematic maps. For example, "soil erosion" is a property of space organized in a layer, representing sets of regions (with different values). Spatial attributes can be visualized as *continuous* (e.g., "temperature") or as *discrete* ("soil type").

- Need to capture the change of spatial attributes over time. Changes–like changes of positions, previously–can be *discrete* (e.g., changes on a map of "landparcels" or "vegetation") or *continuous* (change of "temperature").

- Need to connect spatial attributes to objects. For example, a landparcel has "soil type" as an attribute. The "soil type" is an attribute of space and landparcels inherit part of it.

- Need for the representation of spatial relationships among objects in time. For example, "in 1990, "waterpipe" WP890 crosses "landparcel" LP3760."

- Need for the representation of relationships among spatial attributes in time. For example, the "soil type" is a result of the combination of the "acidity" and the "corrosivity" of soil.

- Need to specify spatiotemporal integrity constraints, imposed either by the user, or by the designer for the integrity of the database. For example, "replace all "waterpipes" which are more than 20 years old, located in areas with "soil erosion" higher than 60%.

4. CONCEPTS OF A SPATIOTEMPORAL ENVIRONMENT

In this section, we discuss concepts involved in a spatiotemporal database environment via the users' requirements presented above. Firstly, the spatial and temporal concepts are given independently, and then, are combined.

A *database* is a collection of *objects*, which represent part of the real world. Each object belongs to an *object class*, which is characterized by a set of properties or *attributes*. Each attribute is associated with a *domain*, which is a set of values. So, each object in a database instance is represented by a set of values, each belonging to the domain of the corresponding attribute of the object class. A database is called *spatial*, *temporal*, or *spatiotemporal* if it manages spatial, temporal, or spatiotemporal concepts, respectively. Next, we describe these concepts.

4.1 Spatial Concepts

In order to start a discussion about spatial concepts, we, first, need to refer to *space*. Then, we shall locate objects on it (spatial objects). Moreover, space has *attributes*, which are represented as *layers*. Objects and layers are *orthogonal* and *complementary* views of space.

4.1.1 Space

Space is a set. The elements of space are called *points*, while finite sets of points (i.e., subsets of space, which can be point, lines or regions) are called *geometric figures*. Any set will do for space; however for practical purposes of current spatiotemporal applications, space is modeled as a subset of R^3. Z^3, Z^2, Z, R^2 and R are the most common subsets used in practice. All specific examples in this paper use R^2 as space, without affecting any discussion.

4.1.2 Spatial Objects

Objects in real world have a *position* in space. In specific application environments, objects' position in space *matters* and these objects are called *spatial objects*. For example, a moving "car" in a navigational system has position, as well as a "landparcel" in a cadastral system. Position *p* of objects is a function from objects to parts of space.

$p: spatial\_objects \rightarrow G$

where, *G* is a finite set of geometric figures.

4.1.3 Spatial Attributes

As said before, objects have attributes which characterize them. Spatial objects have, apart from descriptive attributes, also *spatial attributes*; for example "vegetation" of a "landparcel." Values of spatial attributes depend on the referenced position and *not* on the object itself. If the spatial object "landparcel" changes position, then the value of "vegetation" will change. More specifically, *spatial attributes are properties of space*, and spatial objects located in specific positions inherit parts of these attributes.

However, not all spatial objects have spatial attributes; this depends on the application requirements. For example, no spatial attribute can be assigned to a moving "car", while many (e.g., "vegetation", "soil type") can be assigned to a "landparcel."

4.1.4 Layers or Fields

Spatial attributes refer to the whole space and can be represented as *layers* (or *fields*) representing one theme (i.e., *thematic maps*). Informally speaking, a *layer L* is a representation of a spatial attribute. Formally speaking, a layer is a function from geometric figures to attribute domains:

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

where, G is a finite set of geometric figures and D_i, with $1<i<k$, are domains of values. In other words, a layer is a set of geometric figures (which are points, lines, regions or combinations of them) with associated values.

There are two basic types of layers: (a) those representing functions with continuous range; for example, "temperature," or "erosion," and (b) those representing functions with range of discrete values; for example, "vegetation" represented as set of regions.

The concepts of spatial objects and layers are orthogonal. Consider the example illustrated in Figure 1: a "landparcel" has "vegetation," "soil type" and "elevation" as attributes. While layers (e.g., "vegetation") exist all over the space, objects (e.g., "landparcel") exist in certain positions and inherit parts of layers' values; just those that refer to the objects' exact position.

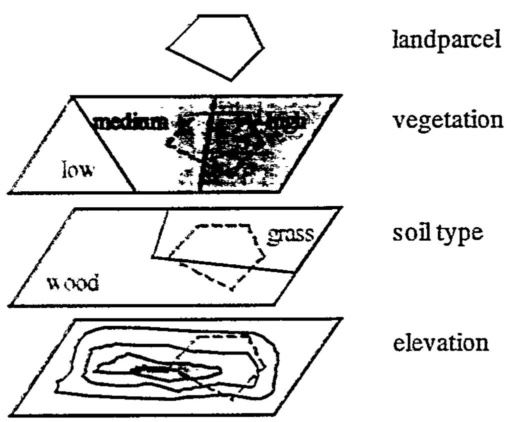

Figure 1: Spatial objects and layers are orthogonal in space.

4.2 Temporal Concepts

The discussion about temporal concepts includes the assumption we make about time, aspects of time that are recorded in the database, as well as models of times that are used.

4.2.1 Time

In literature many different models of time are presented. Some authors [8] even propose taxonomies of time. In our work we assume a *linear ordered time line*, isomorphic to a finite subset of the natural numbers. The elements of this set are termed *chronons.*

4.2.2 Aspects of Time

When talking about time in databases, we refer to the temporal aspects of the real world that can be captured in the database. In literature the temporal aspects of *transaction time* and *valid time* are of foremost importance. Valid time shows when a *fact* is true. For example, "landparcel LP9870 was split into two others in 1990." In terms of implementation, it can be viewed as a classical attribute of a tuple, denoting when the information recorded by the explicit attributes of a tuple is true in the modeled reality. Transaction time, on the other hand, shows when information is recorded in the database. For example, the information that "landparcel LP9870 was split into two others, is recorded in the database in 1995." Transaction time is a system attribute, keeping track of when tuples were inserted and "deleted" in the database. Deletion, here, stands not for physical deletion but for assigning an end transaction time to a tuple that is different from the value *now*, denoting present time.

4.2.3 Models of Time: Time points vs. Time intervals

Two basic models of time are used to record facts and information of a database: *time points* and *time intervals.* A *time point* t_l is considered as one chronon, while a *time interval* $[t_k,t_m]$, with t_k,t_m time points and $m\leq k$, has duration and is defined as set of chronons. Time points and time intervals can record (or, represent) valid or transaction time. Finally, we define a *time period* as a set of time points, time intervals or any combination thereof.

There are two basic facts for which we want to record time: *events* and *states.* An *event* occurs at an exact time point, i.e., an event has no duration. Example events are a "car crash," "sunrise," etc. A *state* is defined for each chronon in a time interval, hence it has a duration. For example, a "meeting" takes place from 9am until 11am.

4.3 Spatiotemporal Concepts

In the previous sections we described the concepts of the spatial and temporal domain, separately. In the following, we combine them. We start by simply recording the spatial view (i.e., objects and layers) in time (time point and time interval); the result is new, *spatiotemporal*, concepts. Since, in this paper, we are only interested in the geometry of spatial objects and layers, for reasons of simplicity, we use the word "spatial object" instead of "position of spatial object."

4.3.1 Spatial Objects in Time points

Recording a spatial object in a time point results into taking a snapshot of it. For example, for a moving "vehicle" in a route network we can record its "new" position in time points $t_1,\ldots,t_n$. In the same way, as a "landparcel" changes its shape (e.g., split, expanded etc), we capture snapshots of it in time.

4.3.2 Spatial Objects in Time intervals

Recording a spatial object in a time interval is translated into capturing its *evolution* over time, i.e., capturing the possible *changes* of its position over time.

Depending on the *type of change*, different spatiotemporal concepts are involved:

(a) if the change is *continuous*, we capture *motion* of objects. For example, a car changing its position continuously, is *moving*. We refer to the change of the position on a continuous basis as *motion*.

(b) if the change is *discrete*, we capture *changes* of objects, through this time interval. Consider the example of recording a "landparcel" in [1995,1997], which changes shape. The changes of shape are discrete, and only the *states* of the landparcel *before* and *after* the change *matters*, not during the change. One could see the discrete changes of an object during a time interval as shapshots of that object, taken in time points of this time interval.

4.3.3 Spatial Objects in Time periods

If we record an object in time points of *the same time period*, then we capture *versions* of that object; in other words, *versions* of an object, are *related* (through its identity) *snapshots* of it. For example a "landparcel" changing its shape in the period [1990,1998] gives different versions of it.

However, we may want to record specific appearances of an object in some time points of a time period, without relating them to previous or next ones. In this case, the result is snapshots of the object. For example, in [1990, 1998] the changes of a "landparcel" give different versions of it, but when it changes ownership, it gets new identity, becoming a new object.

Similarly, if we record an object in time intervals of the same time period, depending of the nature of the object we capture *changes* (e.g., change on a "landparcel") or *motion* (change on "car's" position) of it.

4.3.4 Layers in Time points

This combination results into taking *snapshots* of a layer. For example, the recording of the "vegetation" (of an area of interest) in May 5, 1990, gives a specific vegetation map of that area. In other words, to make the analogy to layers (Section 3.1.4), here, we talk about *temporal layers TL*, where *TL* is:

$TL: T \rightarrow L$ and $L: G \rightarrow D_1 x D_2 x \ldots x D_k$

where, T is a finite set of time points of the time line.

4.3.5 Layers in Time intervals

Similarly to Section 4.3.2, recording a layer in a time interval is translated into capturing its *evolution* over time, i.e., capturing the possible *changes* of it.

Depending on the *type of change*, we deal with different spatiotemporal concepts:

(a) if the change is *continuous*, then it is understood as *phenomenon* [18]; for example, the birth, spread and death of a "storm." So, we refer to the continuous change of a layer as *phenomenon*.

(b) if the change is *discrete*, we capture *changes* of the layer, through the time interval. Consider the example of observing the map of "vegetation" (of an area of interest) in [1980,1990]. One could see the discrete changes of a layer during a time interval as shapshots of it, taken in time points of this time interval.

4.3.6 Layers in Time periods

If we record a layer in time points of the same time period, then we capture *versions* of that layer; in other words, *versions* of a layer, are *related* (through its identity) *snapshots* of it; for example, the appearance of "fires" in Greece in [1990, 1998]. Again, a layer in a specific time point of the time period is a *snapshot*.

Finally, observing a layer in time-intervals of a time period results into observing a *phenomenon*, or *discrete changes* of it.

The following table shows all the combinations of the spatial and temporal domain, resulting in the new, spatiotemporal, concepts of *snapshot*, *change*, *motion*, *version* and *phenomenon*.

	Spatial object	**Layer**
Time point	snapshot	snapshot (temporal layer)
Time interval	motion change	phenomenon change
Time period	snapshot version motion change	snapshot version phenomenon change

Table 1: Combining spatial and temporal concepts.

4.4 Analogies Between the Spatial and Temporal Domain

Although not always clear, space and time have a set of analogies, in terms of concepts needed to capture spatial and temporal information, respectively. Here, we help the developer's better understanding about spatiotemporal applications, by showing that the concepts we chose to model and interrelate are not only the result of the users' requirements analysis phase, but are also governed by an analogous rationale.

Space has attributes (i.e., spatial attributes) which are represented in layers. A layer is a set of points, lines, and regions or any combination thereof, with associated values, which are the values of the represented spatial attribute. Spatial objects occupy positions in space. Spatial objects are orthogonal to layers, as they inherit parts of their values, depending on their position in space.

Analogously, time is represented in time lines. In practice, we consider different time lines for different application environments. For example, the time line for a forest fire management system is different from the time line for a set of events leading to a car accident. A time line has time points and time intervals, or any combination of them, as elements. Events and states take place in time intervals and time points of a time line, respectively. Objects are orthogonal to time lines, as they participate in states and events of the corresponding time lines. Table 2 shows these analogies.

Spatial Domain	Temporal Domain
layer	time line
objects have a position in space	objects exist in time
position of objects can be points, lines, regions or a combination thereof	existence of objects in time can be in time points, time intervals, or a combination thereof
spatial objects and layers are orthogonal	temporal objects and time lines are orthogonal

Table 2: Analogies between space and time.

5. MEETING USERS MODELING REQUIREMENTS

Spatial objects, spatial attributes, layers, time points, time intervals, and their combination provide an elegant way of dealing with users modeling requirements:

- The representation of objects with position in space, and existence in time results in the representation of spatial objects in time points, i.e., snapshots of spatial objects in time.
- The need to capture the *change* of position in space over time is met by capturing objects in time periods, which consist of time points and time intervals. If the change of object's position is continuous, then we have motion. If the change is discrete, then we have snapshots, versions or changes of the object.
- The definition of attributes of space (*spatial attributes*) in time is given in terms of temporal layers, i.e., *snapshots* of layers.
- The need to capture the *change* of spatial attributes over time is met by capturing layers in time periods. Depending on the application domain, this requirement is translated into *snapshots* or *versions* of maps (or layers), or the life of a *phenomenon*, or simple, *discrete changes* on the layer.
- The connection of spatial attribute and objects is achieved by the orthogonal relationship between layers and objects (see Figure 1).
- Spatial relationships among objects in time are reduced to algebraic or geometric conditions (integrity constraints) among their positions (which are geometric figures).
- Relationships among spatial attributes in time are reduced to algebraic or geometric conditions (integrity constraints) among layers.
- Finally, spatiotemporal integrity constraints–imposed either by the user or the designer–for the integrity of the database are translated into constraints among objects, layers, and their combinations.

6. CONCLUSIONS

Spatiotemporal applications design has hitherto focused on the physical data organization and quite understandably so, because of the volume and complexity of the data. In this paper, we focus on the phase of requirement analysis for spatiotemporal applications and we address the involved spatial, temporal and spatiotemporal concepts. The goal is to support the developer's better understanding about spatiotemporal applications, by providing the concepts and the notations needed in such environments; these concepts, will later be translated into specific constructs and implementation issues.

Firstly, we present a set of user requirements for spatiotemporal applications, based on the experience from two real large-scale projects ([22] for the design and development of a network utility management system, and [5] for the design of a cadastral database). We deal only with *modeling requirements.* Then, an exhaustive presentation of the necessary spatial and temporal concepts one should take into account, when he/she deals with modeling user requirements, is given. More specifically, *the concepts of space, spatial objects, spatial attributes, fields, time and models of time* are presented and then combined, resulting in the *spatiotemporal* concepts of *snapshot, changes,* and *versions* of objects and maps, *motion,* and *phenomena,* which are essential to accommodate spatiotemporal needs. Finally, we show how the presented concepts meet the previously listed users' needs.

This work is part of a larger effort for the analysis, design and implementation of spatiotemporal applications. As requirements analysis is the first phase of an application development methodology, followed by the conceptual and logical modeling and the implementation, the next step is to translate the concepts presented here, into specific constructs of semantic models, to capture spatiotemporal information. We are currently working on the extension of the Entity-Relationship model towards this direction [21] and its translation into an extended relational model [20] to accommodate the needs of the spatial and temporal domain.

7. ACKNOWLEDGEMENTS

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. ERBFMRX-

CT96-0056. The authors wish to thank the Computer Technology Institute, Greece, and the National Technical University of Athens, Greece, for providing the environment for the case-studies taken from real applications ([22] and [5] respectively). Discussions with Dr. Th. Hadzilacos and Dr. M. Kavouras were the starting point of this work.

8. REFERENCES

[1] Allen, E., Edwards, G., and Bedard, Y., 1995. *Qualitative Casual Modeling in Temporal GIS*. Proc. of the International Conference on Spatial Information Theory (COSIT'95), Lecture Notes in Computer Science Vol. 988, Berlin: Springer-Verlag.

[2] Batty, M., and Xie, Y., 1994a. *Modeling Inside GIS: Part 1. Model Structures, Exploratory Spatial Data Analysis and Aggregation.* International Journal of GIS, 8, 3.

[3] Batty, M., and Xie, Y., 1994b. *Modeling Inside GIS:. Part 2. Selecting and Calibrating Urban Models Using Arc/Info.* International Journal of GIS, 8, 3.

[4] Brodie, M., 1982. *On the Development of Data Models.* in On Conceptual Modeling, Brodie, M., Mylopoulos, J., and Schmidt, W.J., (editors), NewYork: Springer-Verlag.

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

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

[7] Elmasri, R., and Navathe, S.B., 1994. Fundamentals of Database Systems. The Benjamin/Cummings Publishing Company, Inc., Redwood, CA.

[8] Frank, A.U., 1998. *Different Types of "Times" in GIS.* In Spatial and Temporal Reasoning in Geographic Information Systems, M. Egenhofer and R. Golledge (editors). Oxford University Press, New York, 1998.

[9] Guetting, R., and Schneider, M., 1995. *Realm-Based Spatial Data Types: The Rose Algebra.* VLDB Journal, 4.

[10] Jensen, C.S., Clifford, J., Elmasri, R., Gadia, S.K., Hayes, P., and Jajodia, S. (eds), 1994. *A Glossary of Temporal Database Concepts.* ACM SIGMOD Record, 23, 1, 52-64.

[11] Jensen, C.S., and Snodgrass, R.T., 1996. *Semantics of Time-Varying Applications.* Information Systems, SIGMOD RECORD 21, 4, 311-352.

[12] Lorentzos, N.A., and Manolopoulos, Y., 1995. *Functional Requirements for Historical and Interval Extensions to the Relational Model.* Data and Knowledge Engineering 17, 59-86.

[13] Ozsoyoglu, G., and Snodgrass, R., 1995. *Temporal and Real-Time Databases: A Survey.* IEEE Transactions on Knowledge and Data Engineering, 7, 4.

[14] Pelagatti, G., Belussi, A., Negri, M., and Sbatella, L., 1991. *Design and Implementation of a Geographic Information System: the case of Regione Lombardia.* Proc. of the 2nd European Conference on GIS, Brussels, Belgium, April 1991.

[15] Scholl M., and Voisard, A., 1989. *Thematic Map Modeling.* SSD'89, Springer-Verlag, Lecture Notes in Computer Science, Vol. 409. Berlin: Springer-Verlag.

[16] Snodgrass, R.T., 1992. *Temporal Databases.* Invited paper in the International Conference GIS: Theories and Methods of Spatio-Temporal Reasoning. Lecture Notes in Computer Science, Vol. 639. Berlin: Springer-Verlag.

[17] Story, P.A., and Worboys, M.F., 1995. *A Design Support Environment for Spatio-Temporal Database Applications.* Proc. of the International Conference on Spatial Information Theory (COSIT'95), Lecture Notes in Computer Science, Vol. 988. Berlin: Springer-Verlag.

[18] Tryfona, N., 1998. Modeling Phenomena in Spatiotemporal Applications: Desiderata and Solutions. Proc. of the 9th International Conference on Database and Expert Systems Applications (DEXA'98). Lecture Notes in Computer Science. Berlin: Springer-Verlag.

[19] Tryfona, N., and Hadzilacos, Th., 1995. *Geographic Applications Development: Models and Tools at the Conceptual Level.* Proceedings of the 3rd ACM International Workshop on Advances in Geographic Information Systems. Baltimore, Maryland, 1995.

[20] Tryfona, N., and Hadzilacos, Th., 1998. Logical Data Modeling for Spatio-Temporal Applications: Definitions and a Model. International Database Engineering and Applications Symposium (IDEAS), 1998. IEEE Press Proceedings.

[21] Tryfona, N., and Jensen, C. S., 1998. *Conceptual Data Modeling for Spatio-Temporal Applications: Requirements and Solutions.* Under submission.

[22] UtilNets, [1994-1997]. Work Program of Brite-Euram Project 7120, DG XII, European Union, Brussels, Belgium.

[23] Worboys, M., 1994. *A Unified Model for Spatial and Temporal Information.* The Computer Journal, 37,1.

[24] Yuan, M., 1994. Wildfire Conceptual Modeling for Building GIS Space-Time Models. GIS/LIS.