STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume LII, Number 1, 2007

THE 3SST RELATIONAL MODEL

ANDREEA SABAU

Abstract. In order to represent spatio-temporal data, many conceptual models have been designed and a part of them have been implemented. This paper presents the relational model of the 3SST conceptual model, as the corresponding implementation of it on a relational database platform. The property of generality invoked during the conceptual modeling operation is inherited by the relational model. The concrete model is able to represent thematic, spatial, temporal, spatio-temporal and event-type objects, as the conceptual model. The spatial objects can be represented as points or objects with shape and the evolution of the spatio-temporal objects can be implemented as discrete or continuous in time, on time instants or time intervals. More than that, different types of spatial, temporal, ST and event-based queries can be performed on represented data. Therefore, the 3SST relational model can be considered the core of a ST data model.

1. Introduction

Spatio-temporal databases (STDB) deal with spatial objects that are changing over time and space. In other words, these objects are characterized by spatial and temporal attributes. Yet, it is important to mention that these are not static objects: the spatial characteristics are changing in time. Another condition to consider a database to be spatio-temporal, is to store not only the last state (or the current state) of objects, but their evolution in time as well. Actually, the temporal characteristic of the stored data consists of this condition [4, 9].

After years during which the spatial and temporal databases were studied and developed independently, the need of using spatial and temporal data in the same application appeared. The first attempts consisted in adding one dimension to the other: including temporal data into a spatial database or adding spatial attributes to the temporal objects. Later, other models joined space and time into one unified spatio-temporal view [11].

There are many domains where the spatio-temporal (ST) data is used today:

Received by the editors: September 1, 2007.

2000 *Mathematics Subject Classification.* 68P05, 68P20.

1998 *CR Categories and Descriptors.* H.2.1 [**Information Systems**]: Database Management – *Physical Design*; H.2.8 [**Information Systems**]: Database Management – *Database Applications* .

cadastral applications, military operations, weather systems, multimedia presentations, moving objects etc. Some of the objects managed in these applications have associated the location and the shape as spatial attributes and both of these may evolve in time. Others, like the moving objects, have no need to store their extent, the only important spatial information being the position in space. It can be noticed that some of these applications require data that evolve discretely in time. For example, the cadastral applications or multimedia presentations deal with spatial objects (the shape of the land parcels, and the position and shape of multimedia objects, respectively) that are not changing continuously. Therefore, for such an application, the manner in which the timestamps are associated with spatial objects is straightforward. On the other side, there are also objects that may suffer a continuous evolution in time. The transportation systems that are monitoring moving objects (cars, ships etc.) have to deal with permanently changing positions in the working space.

The main decision that has to be taken during the modeling operation of ST data is how to put together the space and time elements. The simple addition of one dimension to the other may lead to different advantages in performing queries given to one domain (spatial or temporal) to the others detriment. The real challenge is combining space and time in a way that does not put to advantage one of them, as for the storage of data and the allowed operations and queries to be performed on represented data.

Following different approaches in perceiving ST data, modeling techniques and database models, many conceptual models have been designed and concrete applications have been implemented. Some of the models represent space and evolving spatial objects organized in time-stamped layers (see *The Snapshot Model* in [5]). One layer contains the state of a geographic distribution at a moment of time, but there are no explicit temporal connections between layers. One of the earliest data models that can represent spatial and temporal information into a unified view is the Worboy's ST data model [11]. It is an object-oriented data model, within which the main entity is the *ST-atom*. The ST-atom encloses unchanging spatial information during a certain time interval. A ST object is represented by a set of ST-atoms as a *ST-complex*. These elements can be seen as the temporal extension of the spatial simplices and simplicial complexes [3]. The discrete evolution of spatial objects (with or without shape) can be represented using this data model.

An original approach is found in [12]: the *Three-Domain Model* separates semantic domain from spatial and temporal domains. The advantage of this model arises from the independence of the three domains at semantic and behavioral level. There are links from semantic and temporal objects to spatial objects and from spatial and temporal objects to semantic objects. Assuming that a spatial object is located in time, there are no direct links from semantic to spatial domain. The particular case of objects without temporal measures is marked with a null time value. The ST data is organized within four relations: three relations that correspond to the three domains and a relation that links the semantic objects, the time elements and the spatial entities. These structures store the discrete evolution of a region's partitions, as the land usage within a certain area. Therefore, the spatial domain is the same, and only the partition changes over time. These changes (the splitting and the aggregation of land parcels) are maintained using a spatial graph. The space table records only the current spatial elements, because the older spatial objects are determined using the transitions modeled within the spatial graph.

A parametric ST model called *Parametric k-Spaghetti* is introduced in [2]. The evolving spatial data can be of type point, line segment or region. One geometry element is represented by one or more triangles (degenerate in the case of points and line segments). Therefore the ST information is stored within tuples which contain the object's id, the parametric coordinates of one triangle and a valid time interval as timestamp. Though the structure of the relation is relatively simple, the represented information can capture the continuous evolution of spatial objects in time.

Moving Object Data Models have been developed to deal explicitly with continuously moving objects. The Moving Objects Spatio-Temporal data model (MOST) [8, 10] introduces the notion of dynamic attribute represented as functions of time in order to denote an attribute that change continuously. The considered dynamic attributes are the spatial coordinates of the position of moving objects; therefore the model can represent the continuous evolution of spatial objects of type point. Each dynamic attribute A is composed by three sub-attributes: $A.value$, $A.updatetime$, and $A.function$, where $A.value$ represents the value of the corresponding dynamic attribute at time $A.updatetime$ and $A.function$ gives the evolution of the attribute's value until the next update time.

The conceptual ST data modeling process proposed in [7] materialized into a data model called 3SST. Using this model, the designer is allowed to include ST objects, but also thematic objects, without any spatial or temporal attributes. Depending on the application, events may be modeled using particular event-type objects. Discrete and continuous evolutions are allowed for the spatial objects which can be points, lines or simple polygons (therefore, spatial objects with or without shape).

In this paper the work on the 3SST conceptual model is continued by proposing a concrete model in order to be implemented on a relational database system.

The paper is organized as follows: the next section presents the characteristics and the diagram corresponding to the final step of the 3SST conceptual data modeling process. The concrete relational 3SST model is introduced in Section 3 and Section 4 presents a classification of spatial and temporal operations. The final section contains conclusions and proposed future work.

2. The 3SST Conceptual Data Model

A weakness of many existing models is that each of them deals with some common characteristics found within specific applications. The paper [7] proposed a modeling process of ST data in three steps: the construction of an entity-relationship ST model, the specification of the domain model and the design of a class diagram which includes the objects characteristic to a ST application and other needed elements. The modeling phases do not take into account a certain ST application, but tries to identify and use the objects and elements needed within an application dealing, among others, with ST data. The modeling process was called 3SST (*Three Steps Spatio-Temporal* modeling process) and materialized into a data model called 3SST as well.

During the construction of the diagrams corresponding to the three modeling steps, the concrete example of a meteorological application was considered. The main entity of such an application is the meteorological phenomenon. This is one of the most complex ST objects because it is a spatial object with both position and extent as *spatial attributes*, both of them usually evolving in time. Besides these attributes, the so-called *thematic characteristics* (non-spatial attributes) of objects are considered. For example, a meteorological phenomenon may have associated: a type (rain, drizzle, fog, snow, hail, glazed frost, storm), which is a *non-temporal attribute*; different meteorological parameters (atmospheric pressure, air temperature, soil moisture, visibility, wind speed), which can be *temporal attributes* if their evolution in time is recorded. The former kind of attribute will also be called *static attribute* (an attribute whose values does not evolve in time), and the later - *dynamic attribute* (its values are changing in time and we are interested in keeping knowledge about the temporal trajectory of these values).

In order to generalize the domain of the problem and to achieve a fairly comprehensive model, the set of object types to be considered was enlarged. For example, the domain of a meteorological application may contain spatial objects with no temporal characteristics (like table-land or town), temporal objects without spatial attributes (for example the usage of equipments or the measurement of the values corresponding to different parameters) or objects with no spatial or temporal attributes (for example the persons who are studying and analyzing the meteorological information).

Another aspect taken into account was the possibility of having knowledge about the context in which the state of an object changed. If a database contains only ST objects (the attributes and the evolution of their values in time), the types of queries that may be efficiently answered are object-oriented, spatial-oriented, time-oriented and combinations of these. Yet, it cannot be known what caused the change of the state. Therefore, am other object considered was the event in order to incorporate this information [6]. An event object has usually associated as attributes the time instant and the position where the event occurred. It is important to notice that the event objects have no evolution in time and that an event is not a ST object, even if it has associated spatial and time elements.

An observation is made in order to clarify the difference made by this paper between time and temporal elements. It is called *temporal* an element (object or attribute) whose state (value) is changing over time. The timestamps associated with evolving elements are simply called *time* elements.

After the first two modeling steps (the construction of the E-R and the 4-Domain diagrams), the presented data model was able to represent four types of objects:

- *Non-spatial non-temporal objects* (noted here as *thematic objects*): the objects that do not have any spatial or temporal attribute;
- *(strict) Spatial objects*: they have at least one spatial attribute, but its evolution in time is not recorded, and do not have any temporal attribute;
- *(strict) Temporal objects*: these objects do not have any spatial attribute, but they have associated at least one valid time or transaction time attribute;
- *Spatio-temporal objects*: they have at least one spatial attribute whose evolution in time is recorded.

The class diagram corresponding to the 3SST data model is depicted in figure 1. This is the result of the normalization operations over the object classes [1], and only the class model in 3ONF (the 3rd Object Normal Form) is presented. The normalization process at object class level was used during the refining operations because it has the main advantage over the relation normalization the fact that it makes possible to identify independent objects not only at characteristics level, but at behavioral level as well. In this way, the obtained model is closer to a concrete one. For example, the space and time objects might be treated in a similar fashion regarding the resemblance between the spatial and temporal dimensions characteristics; nevertheless, the two domains present major differences at the behavioral level.

Some observations have to be made regarding the class diagram presented in figure 1:

- A geometric object is represented by one or more n-dimensional points (in the conceptual view). Thus, such an object can represent a point, a line segment (if there are associated two points) or any region implemented as a polygon having at least three vertices.
- The points of a polygon are stored in counterclockwise order, in order to facilitate the implementation of different computations, like area, direction, intersection, or triangulation of regions; the attribute Next_point is a link to the next point within the current list of points (if it is not the last point of the list).

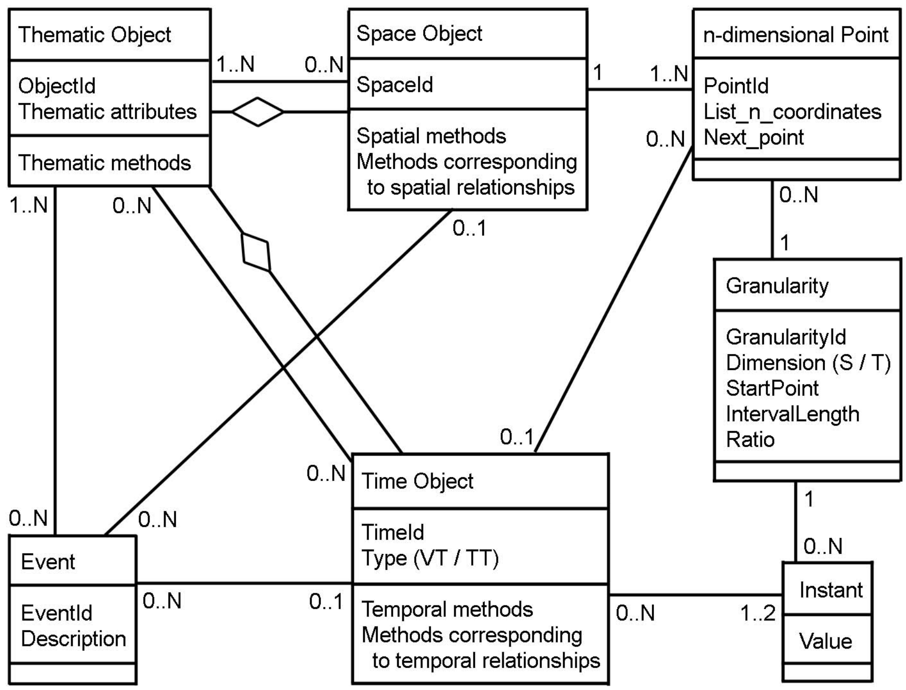

FIGURE 1. The class diagram of the 3SST conceptual data model.

- The presented spatio-temporal model allows both types of time objects to be used: the *valid time* (the time when the fact is true in the modeled reality) and the *transaction time* (the time when a fact is stored in database).
- The time elements can be instants or intervals.
- Two different relations between thematic and space objects, and between thematic and time objects are depicted in figure 1: the case of a thematic object having a time attribute or a spatial attribute that does not evolve in time implies the relation of association; if the object has thematic or spatial dynamic attributes, then the composition relation is considered.
- The entity *Granularity* is included in order to express spatial, time and numerical data in association with different measurement units.
- The methods that define the behavior of thematic objects, space objects and time objects are not presented in detail: the methods of thematic objects may be implemented according to the nature of managed data, and the methods for spatial and temporal data correspond to different

spatial and temporal operators. A classification of spatial and temporal operators is given in Section 4.

- The values of a spatio-temporal attribute may evolve discretely or continuously. On the other hand, regarding the spatial attributes of objects, the changes that may occur are on shape and / or position. These kinds of changes are represented by the data members of the *Point* class: if a point object is represented by a scalar value or a constant function of time, the evolution of that point is discrete on a corresponding time interval or it is a static object; the continuous evolution of a point object during a time interval might be represented by a non-constant function of time.

3. The Relational 3SST Data Model

Some aspects that had been considered during the design and implementation operations of the relational structures are mentioned next:

- The embedding space is considered to be two-dimensional during the implementation phase, because of some limitations of the used Transact SQL language and the implementation of some spatial operators.
- The time elements are enriched with an attribute that represents the corresponding time zone, if needed. For example, the timetable of airplanes uses the local time for arrivals, but, in order to be able to compute the duration of a flight, the difference between the time zones is needed to be known.
- It is known that most of the spatial databases that use the vector data model represent spatial entities by their approximations: a line is represented by as set of connected line segments and a region is modeled as polygon. The implementation of the 3SST model makes use of the same technique. The set of polygons that are represented by a set of points (the vertices) are convex or non-convex and have to be simple, non-self-intersecting polygons. The next sub-section contains the definitions of the basic spatial data types used within the 3SST model: point, line segment, line, and polygon.

3.1. **The Represented Spatial Data.** The space *Sp* that includes the spatial objects is considered theoretically to be the Euclidian n-dimensional space. Because of implementation reasons, the domain of values corresponding to the coordinates of points is limited to the *Real* type of the system.

According to the above observation, $Sp = R^2 = \{(x_1, x_2) \mid x_1, x_2 \in R\}$. Let $P = (x_1,\ x_2)$, $x_i \in R,\ i := 1..2$ be a *point* of the considered space.

Let $P_1,\ P_2 \in Sp, P_1 \neq P_2$ be two points. The *line segment* defined by P_1 and P_2 is given by $S = \{P_s \mid P_s = a \times P_1 + (1-a) \times P_2, a \in [0,1]\}$.

In order to define the line and polygon entities, the *oriented line segment* is considered to be the vector determined by two points P_1 and P_2. Therefore, if $P_1,\ P_2 \in Sp$, $P_1 \neq P_2$, and $SO_1 = (P_1,\ P_2)$ and $SO_2 = (P_2,\ P_1)$ are two oriented segments, then $SO_1 \neq SO_2$

The *line* is defined as a set of oriented segments, $L = (SO_1,\ SO_2,\ ...,\ SO_l)$, such as:

PL1: $\forall i := 1..(l-1)$, $SO_i.P_2 = SO_{i+1}.P_1$ (the segments are connected at their end points);

PL2: $\forall i, j := 1..l$, $i \neq j$, $SO_i \cap SO_j = \oslash \vee SO_i \cap SO_j = \{P\}$ (the segments are not overlapping, partially or totally).

Let $Pg = (SO_1,\ SO_2,\ ...,\ SO_p)$, $p \geq 1$, be a set of oriented segments. Pg is a *simple polygon* if the following conditions are fulfilled:

PP1: $\forall i := 1..p$, $SO_i.P_2 = SO_{(i+1)MODp}.P_1$ (the segments are connected at their end points);

PP2: $\forall i := 1..(l-2)$, $j := (i+2)..l$, $SO_i \cap SO_j = \oslash$ (any two non-consecutive segments are not intersecting);

PP3:

$$\sum_{i:=2}^{p-1} A(\triangle P_1 P_i P_{i+1}) > 0 \quad (1)$$

($A(\triangle P_1 P_i P_{i+1})$ is the signed area of the triangle $\triangle P_1 P_i P_{i+1}$, i:=2..(p-1); the sum of the triangles signed areas represent the signed area of the polygon; the positive sign assures the counterclockwise orientation of the vertices of the polygon).

3.2. **The Time Elements.** The type of time objects that can be used as associated timestamps to thematic or spatial values is instant or interval. It is considered that the time domain is the time of reality, and not simply a surrogate temporal axis, as the real numbers.

The evolution of an object O is given by a sequence of states (S^1, S^2, ..., S^n), each state being defined over a certain time interval (in the case of evolutions represented on time intervals). Let I^k with the end points t_1^k and t_2^k, $t_1^k < t_2^k$, be the time interval corresponding to S^k state, k:=1..n. If the lifespan of O is continuous and there cannot exists two different states of O at the same time, then any two time intervals I^k and I^j, $k, j := 1..n$, $k \neq j$, must be disjoint and they are closed at their "left" end point and open at the "right" end point. Figure 2 depicts a discrete evolution of O, which consists of four states (S^1, S^2, S^3, S^4), during time interval $[t^1,\ t^5)$.

3.3. **The Relational Implementation of the 3SST Model.** The structure of the 3SST relational model described in this section (see figure 3) corresponds to the presented conceptual ST model. The property of generality invoked during the conceptual modeling operation is inherited by the relational model. Therefore, the 3SST relational model can be considered the core of a ST data model. For example, the current structure contains one relation *Object* which includes the

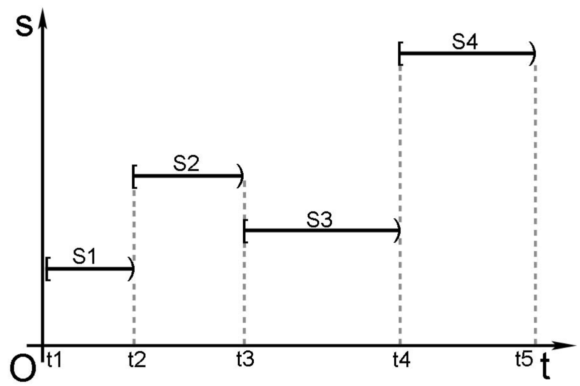

Figure 2. The four states of an object's discrete evolution over the time interval [t1, t5).

static data of the application domains entities. Depending on the set of objects, more *Object*-like relations can be included in the database.

A set of observations and comments about the diagram structure depicted in figure 3 are given next:

- A measurement unit is included into a single family of granularities, each of these having a parent granularity.
- Each spatial element has associated a unique ID (SOID) and each point identified by PID corresponds to a certain spatial element (see the foreign key *Point* (SID) referencing *Spatial_Obj* (SOID)).
- The type of the ROW_ID columns is *Timestamp*. The chosen data type assures that the contained values are unique within the entire database, not only within a relation. Another advantage is that their values are automatically managed by the server.
- A static point does not have associated any tuple within the relation *Evolution*.
- Any point is identified by the value PID, but a state of a point is identified by ROW_ID. Therefore, the evolution of a point is given by the set of tuples of the *Point* relation that contain the given PID value.
- The relation *Evolution* contains the complete evolution of all application's objects (non-spatial and spatial).

4. Operations on Spatio-Temporal Data

This section presents a classification of spatial and temporal operations that can be usually performed on ST data, and examples of operations are given for each category of operations. The implementation of these operations is already

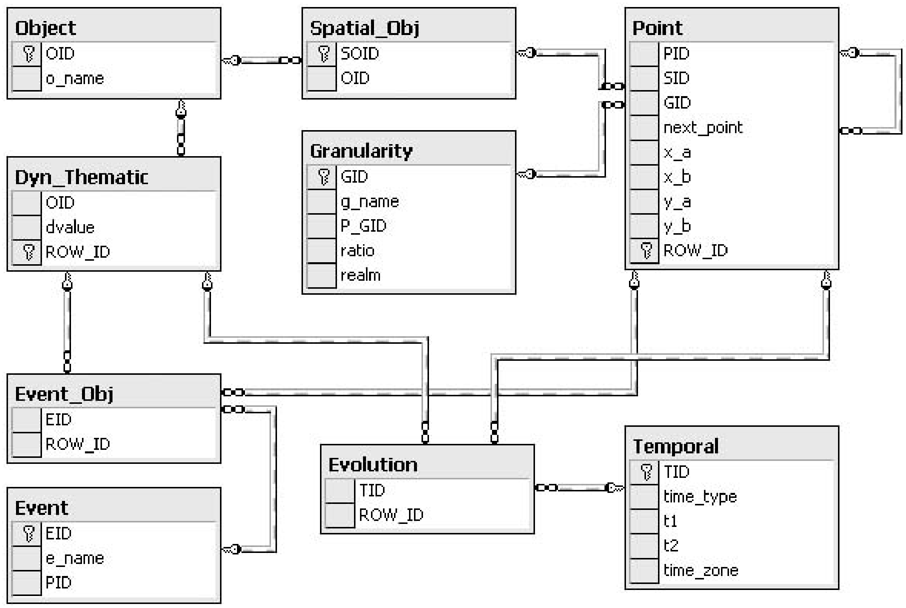

FIGURE 3. The diagram of the 3SST relational model.

accomplished or it is in progress (the aggregation operations).

The spatial and temporal operations on ST data are:

- Operations with numerical result:
  - Simple:
    - Spatial (the distance between two points, the perimeter of a polygon, the area of a triangle etc.);
    - Temporal (the length of a time interval etc.);
  - Aggregation:
    - Spatial (the total area covered by the evolution of an object of type region);
    - Temporal (the average length of a set of time intervals);
- Operations with Boolean result (predicates):
  - Topologic:
    - Spatial (the intersection of two polygons, the inclusion of a point into a polygon)
    - Temporal (a time instant is included into a time interval, the adjacency of two time intervals, the intersection of two time intervals)
  - Metric:

      * Spatial (in_circle, in_square)
      * Temporal (in_neighbourhood)
  - Directional
      * Spatial: the current implementation includes eight functions with result of type Boolean, corresponding to the directional operations mentioned next (is_north, is_south_east etc.)
      * Temporal (an instant of time is before a time interval)
- Operations with result of type Direction - only for spatial data of type point; there are eight values used of type Direction E, NE, N, NW, W, SW, S, SE. Given two points, P_1 and P_2, the result is returned according to the angle θ, where θ is determined by the two semi-lines $[P_1P_1'$ and $[P_1P_2$, $P_1'.x = P_1.x + 1$, $P_1'.y = P_1.y$;
- Operations with result of type spatial or time:
  - Selection of an object's component (the geometric state of a given object at a certain time instant);
  - Simple, construction (the intersection, reunion, difference - spatial or temporal - between two spatial or temporal elements);
  - Aggregation, construction:
      * Spatial (the trajectory of a mobile object as the projection in the 2D space of its spatial evolution)
      * Temporal (the lifespan of a given object)

These operations are used within the implementation of what is considered to be ST operations; therefore, the later ones will not be explicitly mentioned. For example, a ST window query like "Which are the objects that passes through region R during time interval T?" is solved by querying objects whose 3D trajectory intersects the 3D region defined by R X T. Therefore, the mentioned query is reduced to an intersection query of the valid trajectory during T with the region R.

The output of an operation with a complex result is represented as a recordset (for example: a set of points representing the state of a geometric object at a given time instance).

According to the author's knowledge, this is the first paper that introduces the data type *Direction* and mentions operations that return a value of this type.

The operation routines and queries are currently written using the MS SQL-Server's Transact-SQL language.

5. Conclusions and Future Work

The presented paper shows the capability of a relational database system to store ST data with discrete or continuous evolution in time. The spatial attributes of considered objects may be of type point, line or simple polygon. The implementation of spatial evolution allows the defining points of the geometric objects to change independently, with a different frequency, on different time intervals. Also, the implemented model is able to perform different spatial, temporal and ST operations and queries on the stored data, with the help of a set or routines written in the standard query language.

The current work will be continued by the implementation of aggregation operators and visual interface. In order to implement in a more elegant and efficient fashion the data and the corresponding routines presented in the conceptual model, the proposed future work also includes the implementation on top of an object-relational database system and the study of queries performance on a large set of data.

References

[1] S. W. Ambler, *Introduction to Class Normalization*, Available at www.agiledata.org, last updated 2006.

[2] J. Chomicki, P. Revesz, *Constraint-Based Interoperability of Spatiotemporal Databases*, Proc. of the 5th Intl. Symposium on Large Spatial Databases, Springer-Verlag, LNCS 1262, 1997, pp. 142-162.

[3] M. Egenfofer, A. Frank, J. P. Jackson, A Topological Data Model for Spatial Databases, Proc. of the 1st Intl. Symposium on Large Spatial Databases, Santa Barbara, 1989, pp. 271-286.

[4] C. S. Jensen, R. T. Snodgrass, Temporal Data Management, TimeCenter TR-17, 1997.

[5] G. Langran, N. R. Chrisman, A Framework for Temporal Geographic Information Systems, Cartographica, 25, 3, 1988, pp. 1-14.

[6] D. Peuquet, N. Duan, An Event-Based Spatio-temporal Data Model (ESTDM) for Temporal Analysis of Geographical Data, Intl. Journal of Geographical Information Systems, 9, 1, 1995, pp. 7-24.

[7] A. Sabau, The 3SST Model: A three Step Spatio-Temporal Conceptual Model, Proc. of the 2nd AIS SIGSAND European Symposium on System Analysis and Design, Gdansk, 2007, pp. ?-?.

[8] A. P. Sistla, O. Wolfson, S. Chamberlain, S. Dao, Modeling and Querying Moving Objects, Proc. of the 13th Int. Conf. on Data Engineering (ICDE13), 1997, Birmingham, 422-432.

[9] R. T. Snodgrass, Of Duplicates and Septuplets, Database Programming and Design, 11(6), 46-49, 1998.

[10] O. Wolfson, B. Xu, B., S. Chamberlain, L. Jiang, L., Moving Objects Databases: Issues and Solutions, Proc. of the 10th Intl. Conference on Scientific and Statistical Database Management (SSDBM98), 1998, pp. 111-122.

[11] M. F. Worboys, A Unified Model for Spatial and Temporal Information, The Computer Journal, 37, 1, 1994, pp. 27-34.

[12] M. Yuan, Use of a Three-Domain Representation to Enhance GIS Support for Complex Spatiotemporal Queries, Transactions in GIS, 3(2) , 1999, pp. 137-159.

Faculty of Mathematics and Computer Science, Babes-Bolyai University, Cluj-Napoca
E-mail address: deiush@cs.ubbcluj.ro