Identity Based Change Represented in the Three Domain Model: An Efficient System for Modeling Real World Phenomena

By

TONY H. WAKIM

A Thesis

Submitted in Partial Fulfillment of the
Requirements for the Degree of Master of
Science in Computer Science

Department of Computer Science
Faculty of Natural and Applied Sciences
Notre Dame University – Louaize
Zouk Mosbeh, Lebanon

June 2000

Identity Based Change Represented in the Three Domain Model: An Efficient System for Modeling Real World Phenomena

By

TONY WAKIM

Approved:

Fouad Chedid: Associate Professor of Computer Science and Chairperson.
Advisor.

Khaldoun El Khaldi: Assistant Professor of Computer Science.
Member of Committee.

Marie Khair: Assistant Professor of Computer Science.
Member of Committee.

Nasser Saad: Assistant Professor of Mathematics.
Member of Committee.

Date of Thesis Defense: 10 July 2000

ACKNOWLEDGEMENTS

"Security and clear guidance bring true wisdom, and wisdom becomes the spark or catalyst to release and direct power." *Stephen R. Covey*

Thank You GOD.

I would like to express my sincere thanks and gratitude to Dr. Fouad Chedid for his support and guidance throughout my work on this thesis.

I would like to express my deepest appreciation and gratitude to my family for their patience and support throughout my study and work on this thesis. Their encouragement and sacrifice have always been a major motivation in my work.

I feel a deep sense of gratitude:

- To my boss Mr. Jacques Ekmekji who was of great help and support.
- To all my professors at NDU, thank you for everything.
- To Dr. M. Yuan, for providing me with the literature about the three-domain model.
- To my colleagues at work especially my team members.
- To my friends, Mounia Bedran Saba, Pepa Sadek, Hisham Abdel Massih, Ghassan Eid, George Yared, Rana El Eid, Samer Wakim, and specially to Carine F. Khoury, Roula Chehab, and Bassam Farah.

ABSTRACT

Geographic Information Systems (GIS) have gone a great way since it has started as a science. GIS is making headlines in a number of disciplines and, through its advancement, has improved our ability to model our real world. In the past years the need to support the storage and retrieval of dynamic changes, which occur to our data, has increased which consequently led to the interest in spatio-temporal behavior in GIS. These years have also seen the birth of several spatio-temporal data Models, whose aim is to support this spatio-temporal behavior. The main aim behind Spatio-Temporal support is to be able to answer a set of spatio-temporal queries. The ultimate goal behind these models is to support exploration, explanation, prediction, and planning. In this thesis we will review these models, then concentrate on the three-domain model for we believe its flexible structure makes it an excellent model to support the storage and retrieval of spatio-temporal data. In addition this thesis investigates a special type of change named Identity Based Change and show how to efficiently implement identity based change operations within the three-domain model framework. Modeling spatio-temporal change will help to further reflect real world behavior, in other words improve our ability to model the real world.

TABLE OF CONTENTS

List of Figures

List of Tables

CHAPTER I
INTRODUCTION

In this chapter we introduce the topic of GIS highlighting a number of applications that make use of it. Then we distinguish between spatial databases and temporal databases and motivate the need for creating spatio-temporal databases. Also, this chapter defines the problem that this thesis is concerned with. Mainly we investigate the use of the three-domain model for representing identity based change operations. The chapter concludes with an outline of the rest of the thesis.

1.1. MOTIVATION

The topic of Geographic Information Systems (GIS) is currently about 3 decades old. GIS as a science is mostly still associated with the production of maps; and those who are using it for other than map production are still to discover its full potential. It is an emerging science whose potential is being unleashed little by little and whose implementations are growing as its demand is growing. However if we look at the wider picture of the definition of GIS, it is considered to be "a collection of information technology, data, and procedures for collecting, storing, manipulating, analyzing, and presenting maps and descriptive information about features that can be presented on maps" [22]. Another perspective of looking at GIS considers it to be an aggregation of people, data, hardware and software; the sum of which when put collectively gives us the components that GIS as a science requires to function.

There is a wide range of applications covered by GIS. More and more work fields and organizations are recognizing GIS with time. As a consequence, the range of users is growing steadily. Currently the implementation of GIS include fields such as surveying and mapping, forestry, agriculture, fire fighting, utilities such as water, gas, electricity, oil and petroleum; others include transportation, insurance and finance, retail sales, government and military, landscaping, health care, banking, real estate, etc. [22]. The implementation of GIS in each of these sectors vary based on the need, where there could be certain common points or total diversification of GIS implementations.

The increased demand and recognition for GIS is mainly due to the benefits that GIS has to offer its users. The uses range from doing more with less resources, to deriving greater benefit from material at hand and from staff activities; it also includes improved effectiveness, reflected in improved information flow to management, between departments, better utilization of assets, and better decision making.

A main disadvantage of existing GIS systems is that they store the latest state of the data; that is the GIS database reflects the latest changes [10][19][20][21]. Since the database represents data for the current time, any analysis involving old data is a major problem to deal with. However the need for storing data related to past times is being acknowledged more and more, as the need for that data increases. For example in a maintenance system one would expect that he would have the ability to retrieve information about past maintenance operations for a certain feature. In an existing GIS systems this would not be possible since only the last state of the feature is stored and any previous data is overwritten. The need also goes far beyond the normal explanation of data, it goes into exploration, prediction and planning. Planning is commonly used in GIS systems, however the ability to retrieve past information gives us the means to discover patterns in previous data so that explanation and prediction are made possible.

Recently, efforts are made to integrate spatial and temporal databases into what is currently known as spatio-temporal databases. Several spatio-temporal models have been established [1][2][3][7][8][11][12][13][16][17][19] for the purpose of answering spatio-temporal queries; that is queries that are related to space, time and space-time related events. These models vary in their approaches whether storing the whole database at a given time, or time stamping the features, or restructuring the database model to store the required information. The aim stays the same, to be able to bring real world changes into the database.

1.2. DEFINING THE PROBLEM

In this thesis we will investigate further the Three-Domain model [19][20][21], proposed for modeling spatio-temporal phenomena. This will be done within the concept of identity based changes as proposed by M. J. Egenhofer and K. Hornsby in their work on qualitative representation of change [4][5].

The rest of this thesis is organized into five chapters. Chapter II discusses some background issues of related topics. Chapter III discusses single object identity based change operations and demonstrates how to implement them in the framework of the Three-Domain model. Chapter IV constitutes our case study where a GIS has been created to keep track of the locations of different transformers that are (were) part of the EDL (Electricity of Lebanon) Network for the municipal area of Beirut. Chapter V concludes the thesis with a summary and directions for future research.

CHAPTER II
BACKGROUND

In this chapter we will discuss the notion of GIS. We explain its components, people, data and types of data, hardware and software and software operations, mainly spatial operations, that are implemented in GIS. Then we explain what we mean by static change representation and compare it to dynamic change representation. Some key terminologies related to spatio-temporal modeling are then introduced, these include time semantics and model semantics. We also explain the types of queries and give examples about them. The recently proposed spatio-temporal models are then described. In particular we focus on the three-domain model and elaborate on how space, time and semantic objects are stored separately and how can this be a great advantage in storing change and supporting spatio-temporal queries. Finally, we conclude this chapter by describing the notion behind identity based changes, explain the types of single object identity-based changes and composite object identity-based changes.

2.1. What is GIS?

We recall that the definition of GIS is an aggregation of human resources, data, machines and software, which collectively form GIS system [22]. (*See Figure 2.1*)

Figure 2.1: GIS components

In the following sections we give a brief description of each of the components that appear in figure 2.1.

People function on different levels from surveying needs of an institution to data collection and acquisition, to data entry, to those who tailor applications based on the collected needs and others who operate on these applications to obtain the requested information.

The data collected can be widely diversified, based on the desired output. Mainly it consists of geographic data, tabular data, in addition to other kinds of information. The geographic data in general is divided into what is known as separate layers that constitute a separate related group of features, such as buildings, roads, customers, vehicles, etc.(see Figure 2.2)

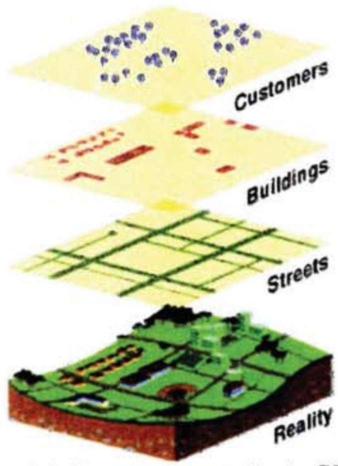


Figure 2.2: Layer representation in GIS.

GIS features can be represented in a GIS system using two ways: the first is using raster data and the second is using vector data. Raster data is represented by a matrix of cells (pixels for 2D, and voxels for 3D). Usually a certain feature is represented by a collection of contiguous cells sharing similar properties. The size of the cell affects the analysis done and the resulting maps generated. Large cell sizes can cause some data to be lost where as small cell sizes would result in more data precision but also with more data/information to store. At the same time smaller cell sizes would cause analysis to consume more time than large cell sizes.

On the other hand in a vector format, features are stored as a collection of (X,Y) coordinates that define the shape of the feature. In this format there are three basic feature types: point, line and polygon (*see Figure 2.3*).

Figure 2.3: Basic types and their representation

In a vector data, a feature is represented by one row in a table. This row contains the shape (collection of x,y points) and other attribute data that are correlated with the feature (see Figure 2.4). Vector representation makes more efficient use of computer storage as they require only the useful data and do not need to represent the whole plane or space.

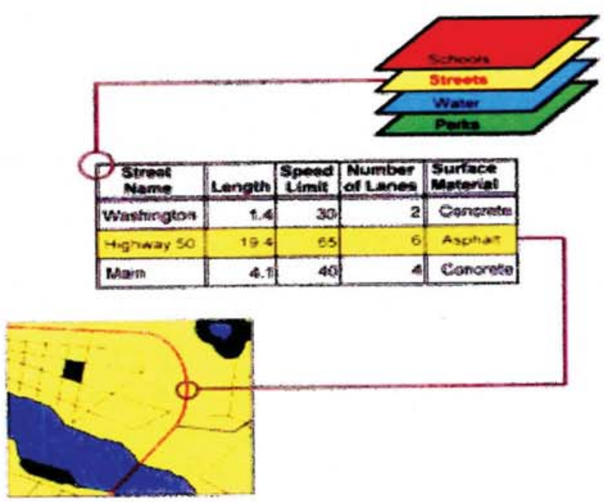
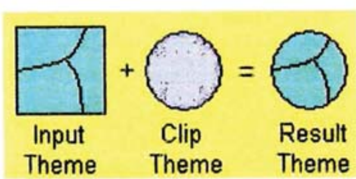


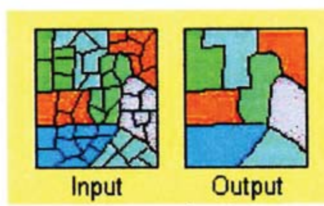
Figure 2.4: Spatial tabular correlation.

The hardware consists of the machines that store the data and that contain the software that operates the GIS system. In addition, hardware can consist also of the tools that are used to obtain the data such as scanners, digitizers, and GPSs (Geographic Positioning Systems) or any other tools used for outputting the results such as printers, plotters, etc.

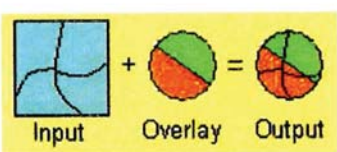
Software on the other hand consists of the tools used by the people to operate on the data and perform the requested analysis. There are several kinds of software from different providers. They range from simple interfaces to preview whatever data we have to more advanced tools that do analysis on the data. A more advanced type of software is the one that can be customized and tailored based on the needs to do the requested analysis. Operations on the data can vary from simple retrieval of the data to identifying certain features, to querying the data for certain data values to operation like summarize operations and performing calculations. Other somewhat more advanced operations include thematic classification, showing graduated symbols, graduated colors, charts, contours and 3D perspective views. Finally coming to some of the advanced geo-processing spatial operations such as clip, dissolve, intersect, merge, etc.



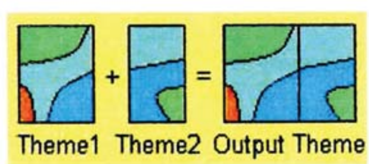
(a): "Clip"



(b): "Dissolve"

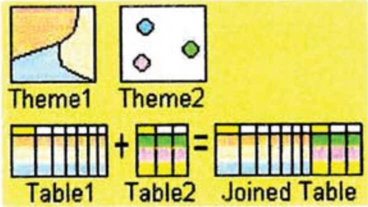


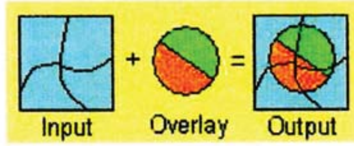
(c) "Intersect"



(d): "Merge"

Figure 2.5





(e): "Spatial Join" *(f): "Union"*

Figure 2.5

2.2. Static Verses Dynamic Changes

A standard GIS system normally stores the latest version of the data, when a change to a certain feature occurs, it overwrites the old data. This is known as static data representation. Dynamic data representation is acquired when the change to the data is stored [19][20][21]. Dynamic data representation varies based on the model used. Each model has its unique way of representing the data, which determines its ability to answer queries differ. In the next section we describe a number of models proposed to handle dynamic change.

2.3. Key Terminology and Query Definitions

Before we discuss the proposed models we explain some key terminology that are used in these models. These key terminologies [10] describe some of the characteristics of the models. They deal with time semantics, model semantic and finally query capabilities.

The usage of time semantics differs from model to model based on the way the designer of the model wishes to use them; for example:

- *Time Density* describes whether time should be modeled as discrete of continuous.
- *Time Reference* describes whether the time should be absolute or relative.
- *Time Order* has two views: linear and non-linear. In the linear model time advances from past to present then to future. In the non-linear model, time advance from past to present then branches to several future time lines depending on the events that happen in the present.
- *Granularity* describes a time slice, defined by a fixed point and a partitioning length. Different models require different granularity sizes.
- *Transaction Time* describes when the time of a certain event is entered into the database.
- *Valid Time* is the time that the event really occurred in life.

Model semantics on the other hand describe other points related to the model itself. For example how to incorporate time into the model and what to time stamp. Representing the data in the model is another point, whether to represent it as vector or raster. What data types to use also is described in the model semantics. Factors like how to best model the real world and how to best capture movement and change also are parts of the model semantics.

Query Capabilities as described in [10][21], are classified into four categories of spatio-temporal queries. (1) Queries about attributes of entities, (2) queries about locations, spatial properties, and spatial relationships, (3) queries about time, temporal properties and temporal relationships, and (4) queries about spatio-temporal behaviors and relationships.

Next we give examples of the types of queries introduced in each category.

(1) *Queries about attributes of entities*. These ask questions about attributes and relationships that might be independent of time and space. An example of a simple attribute query would be "Who is the owner of this car?"

(2) *Queries about locations, spatial properties, and spatial relationships*. They answer queries about where and what. Questions would be like "Where are parcels with an area greater than 2000 square meters?"

(3) *Queries about time, temporal properties, and temporal relationships.* A simple temporal query example would be "what is the state of a feature at time t?" A temporal range query would be "what happens to a feature over a given period?"

(4) *Queries about spatio-temporal behavior*. A simple spatio-temporal query seeks information about a state at a given time, an example would be "what is the state of a region at time t?" and a sample spatio-temporal range query would be "what happens to a region over a period of time?"

2.4. Proposed Models

This section describes a number of spatio-temporal models recently proposed in the literature [1][2][3][7][8][11][12][13][16][17][19], these models are:

2.4.1. The Snapshot Model:

This model is one of the simplest models. The main concept behind this model is to time-stamp the whole layer requested. What happens is that at certain time interval a copy of whole layer is taken and time-stamped. Now, the stamped layer may contain some features that have not changed. This means that redundancy of data may frequently occur regardless of the amount of change to the data. Another main point here is that time intervals between any two copies of a layer may vary, and there is no reference to changes that occur to the layer between these two times. The model uses discrete time intervals, and time reference is absolute. Only valid time is supported. This model can support simple types of queries, such as attribute queries, temporal queries and spatial queries, but it cannot answer queries (without exhaustive analysis) about spatial relationships, temporal relationships, temporal range queries, for example: (Who owned that house between time T1 and time T2?), spatio-temporal range queries, for example: (Where was the max fire spread between T1 and T2?) or spatio-temporal behavior queries.

2.4.2. The Space-Time Composite Model:

This model is based mainly on the idea that all features across time are projected to a spatial plane creating a polygon mesh. Each polygon in this polygon mesh has its attribute history associated to it. Time in this model is linear, discrete and the reference is relative. Both valid and transaction time, are supported by this model. One major problem with this model is that whenever an object is split into two parts, the old object is replaced with the two new objects with two new identifiers. Consequently, the old identifier must be replaced in the whole database with one or both of the new identifiers. This model has good support for attribute and temporal queries as well as spatio-temporal queries and spatio-temporal range queries. One main difficulty in this model is the handling of queries about spatio-temporal behavior and relationships.

2.4.3. Data Models Based on Simple Time-Stamping:

In this model, the main concept is to timestamp each object with its creation time and its cessation time. Cessation time for current objects can be stored as a special value, usually one of "now", "current", or "null". This model uses discrete time with absolute reference; time is linear and it supports only valid time (transaction time is not supported). The strength of this approach is that it is relatively easy to answer questions about the state of an object at a certain time; but at the same time its weakness is seen in the difficulty to trace history of single objects. This model can support attribute queries, spatial queries, simple temporal queries, and simple spatio-temporal queries, but its weakness is revealed when dealing with relationships whether spatial or temporal and when dealing with range queries and spatio-temporal behavior.

2.4.4. Event Oriented Models:

This model represents events explicitly. It stores a fixed initial state of the spatial layer and then as events occur only these events are stored by storing the changes that occur to the spatial feature related. The time density is discrete, and the time reference is relative; to get the status of a feature at a certain time you would have to trace back in the changes until you reach the time requested. This model supports only valid time. A main advantage behind this model is that data redundancy is minimal since only the changes to the spatial features are stored. One main advantage that we have not seen in the past models is that errors can be trapped because improbable events can be isolated and corrected; whereas in the past models there was no reference to the events themselves. The model is strong in questions like "What has happened in the area in this period?" and weak in spatial queries, relationship queries, temporal range queries and spatio-temporal behavior queries.

2.4.5. The History Graph Model:

This model is concerned with the ability to represent objects as well as events that occur on these objects. It is noted that in the previous models change (if represented) is considered to be a result of an event that took place instantaneously; that is without any time duration. This is not the case in real life. For example, if a road is being built, it would require some time to finish it. This model takes care of representing the state of all objects before, during, and after this event occurred. There are three types of behavior identified. Continuously changing object, events with duration and finally sudden events (see *figure 2.6*)

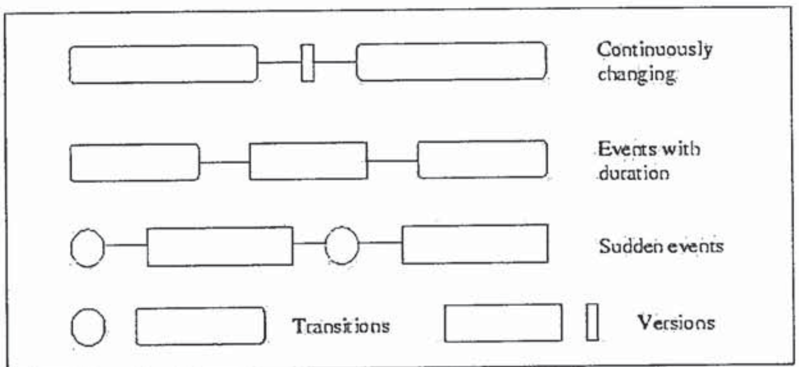


Figure 2.6: Behavior types of temporal objects.

The changes that occur are identified to be one of the following:

Creation: Where an object is created from an event.

Alteration: Where an object undergoes a certain event and changes to another object.

Cessation: Where an object ceases to exist after a certain event occurs.

Reincarnation: Where a previously existing (and ceased) object comes back to existence after a certain event occurred.

Split/Deduction: Where one object, after being exposed to an event, is split into more than one objects.

Merge/Annexation: Where several objects combine into one object after a certain event occurs.

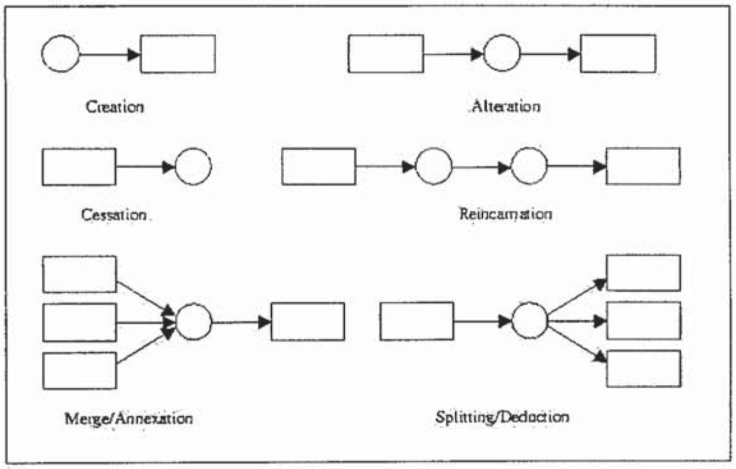


Figure 2.7: Types of change

The History graph model can be discrete or continuous, it can also have absolute or relative time reference, and can support both transaction and valid time. A main advantage is that it can support all types of queries identified previously.

2.4.6. The Three-Domain Model:

In this model, in addition to the space and time domains, a third domain named the semantic domain is introduced. Each of these three domains is placed in a separate table with unique identifiers for its objects. In addition this model introduces a fourth table, the relation table, which provides the links between spatial, temporal and semantic objects. Any semantic, spatial or temporal change will be logged into the relation table, where each entry is a triplet made of semantic-id, time-id and space-id. One of the model's main advantages is its ability to handle movement as well as change. A main point in this model is that every spatial object has a unique identifier throughout its life. And to keep track of the relation among these objects the model provides a spatial tree for that. Although this model distributes data throughout separate tables, it still allows the user to compose any query of interest; this model supports the previously identified four types of queries. In this model, time is discrete, the reference can be either absolute or relative, and it can support both valid and transaction time.

2.4.7. Spatio-Temporal Data Models with Moving Objects:

The main idea behind this type of modeling is to model moving object. If we are dealing with position movement then the representation is that for moving points; if we are dealing with extent then the representation is for moving regions, which could also include growing or shrinking regions. Time is considered continuous and time reference is absolute. One of the proposed models in this category represents an attribute of an object by three sub-attributes: a value, an update-time and a function. It is defined as follows: at time "update-time" the value of the object is "value", and until the next update, the value of the object at time "update-time"+t0 is provided by the function which takes as parameter the time (update-time + t0). In other words, this model has the ability to project future locations of the objects based on current status and a given function. However this specific model has deficiency in supporting relationship queries, as well as spatio-temporal queries.

2.4.8. The Spatio-Temporal Entity-Relationship Model:

Entity Relationships are commonly used in the conceptual stage of a database design. Recently, efforts were made to extend it to the capture of spatio-temporal information. The main idea is to study the requirements of spatial information, temporal information, and then combine them both into a higher level of abstraction. Information like size, location, and orientation form the link between a shape and positions entities. Time is discrete, and reference is absolute. This type of model supports only valid time. This category supports all types of queries except those related to spatio-temporal behavior queries.

2.4.9. Spatio-Temporal Object Oriented Data Models:

This category of models uses the concepts of object oriented paradigm; these include objects, classes, inheritance, encapsulation and polymorphism. Different approaches for time stamping can be used, some models time stamp the entire geographical entity, others time stamp the primitive spatial object (point, line, polygon) and a third option is to time stamp the space at the point level, and then to propagate this to a higher level object. The basic time density used is discrete, and time reference is absolute. And mostly the ability is to support valid time only. Good support for most queries is provided, except in some cases related to temporal relationship and spatio-temporal behavior queries.

2.5. The Three-Domain Model:

The three-domain model was introduced in [10][19][20][21]. The author structures her model into three separate domains: semantic domain, which contains semantic objects, temporal domain, which contains temporal objects, and spatial domain, which contains spatial objects. These Three domains are highly interconnected in the form of a separate domain that represent the dynamic links among semantic, temporal, and spatial objects.

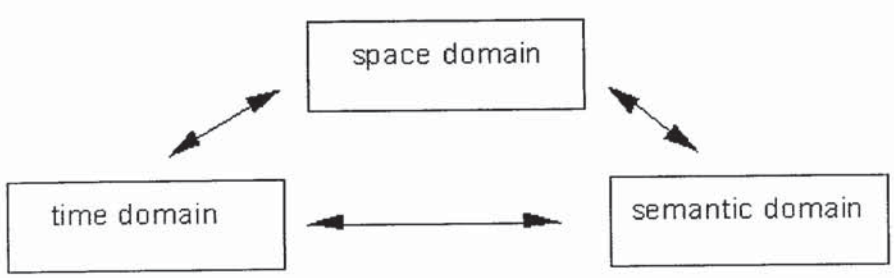


Figure 2.8: Conceptual framework of the three-domain model.

In [21], the author specifies the links among her domains to be one of three types: (i)-pointers that associate objects of different domains to represent spatio-temporal facts, (ii)-mathematical functions that indicate spatio-temporal behaviors, and (iii)-physical models that predict trends or processes in space an time. The author restricts her study to the first representation of domain links although she mentions that it could be implemented in any one of the three.

A table containing a list of all semantic objects represents the semantic domain. Every semantic object has a unique identifier that represents it. An example of this is shown in table 2.1 below [21], where the data represents transactions that occur on a forest area.

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

Table 2.1: The Semantic Table (three-domain model)

The semantic table could have links to other tables if we need to represent any additional information. Second, an example of the time table is shown in table 2.2 below.

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

Table 2.2: The Time Table (three-domain model).

In a traditional GIS system, time is represented as an attribute of the spatial object, but not in this model. Here time is considered to be as a separate entity, an object with its unique identifier. The time table holds the time values and the unique identifiers representing these time values. In table 2.2 the time values are (1600, 1700, 1800, 1950, 1960) and their respective time ids values are 1 through 5. The model assumes no change in the spatial domain for any time not represented in the time table. The time table contains all time values at which any change occurs.

The Space Table (see table 2.3) holds the spatial objects with their unique identifiers. Only those spatial objects that represent the latest configuration of the data are found in the spatial table, normally with their coordinates, geometry and topology.

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

Table 2.3: The space table (three-domain model)

The domain links (see table 2.4) represent association between the semantic, time and space objects.

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

Table 2.4: Domain Links, The Relation Table (three-domain model)

For example the first record in the above table shows that the feature whose space-id is 1 (represented by area = A1, and perimeter = p1) was created at time whose time-id is 1 (1600) and whose semantic-id is also 1 (landcover = Old Growth, ...). The relation table enables us to keep the three domains (space, time and semantics) separated and at the same time be able to represent the links between them. This separation gives us a level of abstraction that can be useful in many ways. For example keeping the semantic reference independent enables us to keep track of semantic objects through time independent of the change that may occur to the space object related to them. In addition keeping the space object independent also allows us to upgrade the system to represent 3D space instead of 2D without affecting the system at hand.

Since only features that are current are represented in the space table, the model uses a separate spatial tree to represent the links between spatial objects and thus gives us the ability to infer past spatial objects that have undergone change. Figure 2.9 shows an instance of a tree.

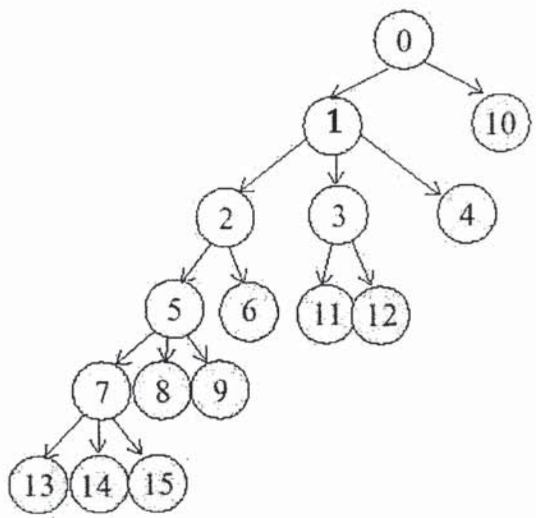


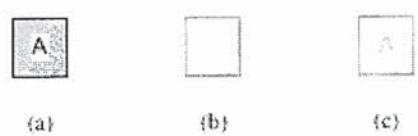
Figure2.9: The Spatial Tree

In this spatial tree, the leaf nodes (shown in gray) represent the most recent objects or in other words the spatial objects that are represented in the space table. The space object referenced by id 3 is not a current space object; but if we want to retrieve its shape we can deduce it by aggregating spatial objects 11 and 12. Thus by going back the spatial tree we can reconstruct any spatial object we want.

In regard to the model's capability to answer spatio-temporal queries, it was shown in [10][21] that this model supports all four types of spatio-temporal queries mentioned before.

2.6. Identity Based Change

In [4] and [5], Hornsby and Egenhofer considered a new approach to change based on object identity. As a matter of fact, there is a rich set of change semantics, as perceived by many people that regard change to an object as something that either preserves or changes the identity of that object. In [4], the authors began by defining two basic identity-based change primitives named create and destruct. To describe these primitives the authors use some basic simbology as represented in Figure 2.10.



Basic symbols used for (a) object existence, (b) non-existing object without history, and (c) non-existing object with history.

Figure 2.10: Identity based changes – basic symbols

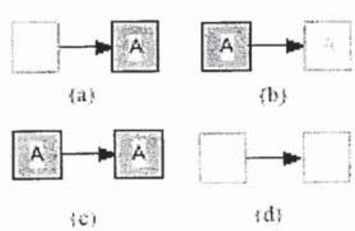
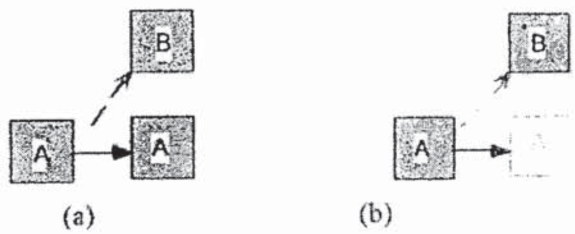
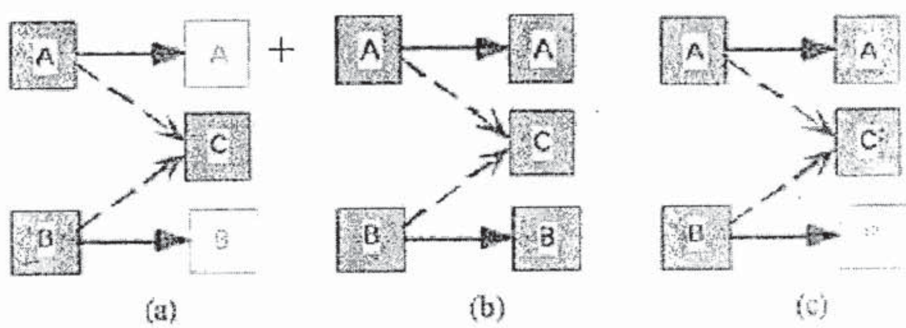


Figure 2.11: Identity operations on single objects: (a) create, (b) destruct, (c) continue existence, (d) continue non-existence.

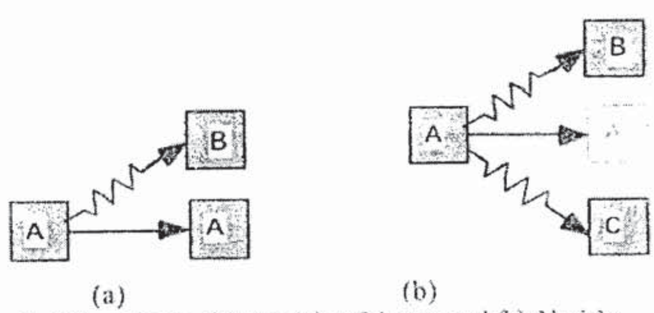
Figure 2.11 shows the basic representation of the 'create' and 'destroy' operations. Then, the authors developed a whole set of operations, built on top of create and destruct, to model real-world phenomena that involve single as well composite objects. Besides create and destruct, identity operations on single objects included issue operations (spawn, metamorphose - Figure 2.12), combine operations (merge, generate, mix – Figure 2.13), and split operations (splinter, divide - Figure 2.14).



Operations involving the transition type *issue*:
(a) spawn and (b) metamorphose.
Figure 2.12.

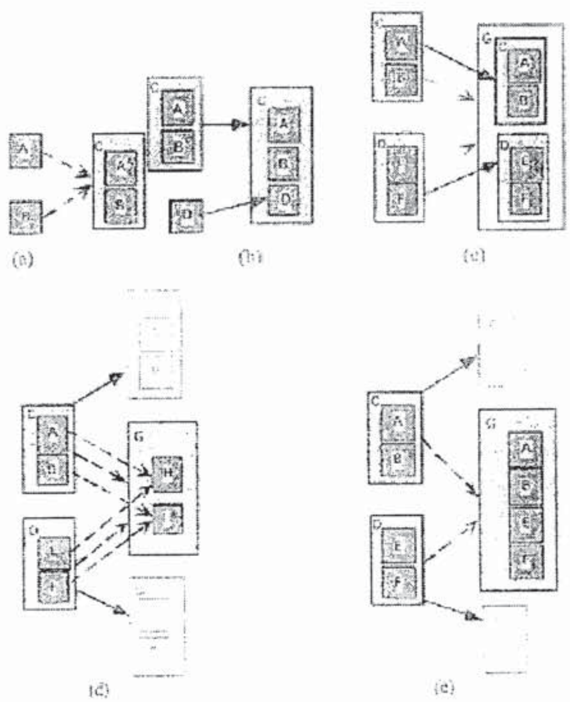


Combining single objects: (a) merge, (b) generate, and (c) mix.
Figure 2.13



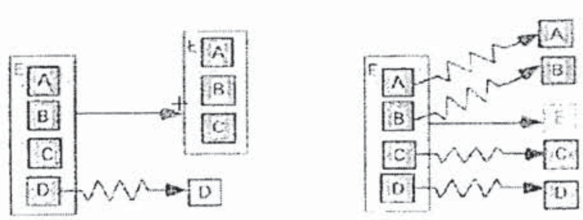
Splitting single objects: (a) splinter and (b) divide.
Figure 2.14

Identity operations on composite objects included join operations (aggregate, compound, unite, amalgamate, combine – Figure 2.15) and split operations (secede and dissolve – Figure 2.16) (we refer the readers to [4] and [5] for a full description of these operations). However, in presenting these various types of operations, Hornsby and Egenhofer used an iconic, visual language with no consideration as to how these operations can be actually implemented at the system level. In the next section, we develop a system based on the three-domain model that supports all these identity-based change operations efficiently. We also consider further this change-based approach to data modeling in general.



Forming composite objects: (a) aggregate, (b) compound, (c) unite, (d) amalgamate, and (e) combine

Figure 2.15



Splitting composite objects: (a) secede and (b) dissolve.

Figure 2.16

Chapter III

THE THREE DOMAIN MODEL FOR IDENTITY BASED CHANGES

In this chapter we describe the identity based change operations and show how to represent them in the three-domain model. We show that the three-domain model can work as a system-level framework for supporting identity-based changes. We take the example provided by Hornsby and Egenhofer [4] in their work on identity based changes and implement it using the three-domain model framework.

3.1 Single Object Identity Based Changes

We recall that part of the three-domain model is the concept of a semantic domain. As explained in [10], the semantic domain holds uniquely identifiable objects that correspond to human concepts independent of their spatial and temporal location. Also as quoted by Hornsby and Egenhofer in [4], "object identity has been defined as the trait that distinguishes an object from all others [6]." They continue to quote "Identity provides a way to represent the individuality or uniqueness of an object, independent of its attributes and values. This concept aids the idea of an object being a stable and enduring element, something on which we can have a perspective [14]. This identity may be implemented at the system level by ensuring that each system object has a unique identifier, created when the object is created, never altered, and only removed once the object has been destroyed [18]." It should be clear now that all these authors are talking about the same concept, which is so elegantly incorporated into Yuan's three-domain model in the form of a semantic domain.

The rest of this section is devoted to a system implementation of identity-based change operations within the framework of the three-domain model. Unique to our implementation is that the unique semantic-id of an object is never destroyed, even when the object is destroyed (the semantic-id is kept in the database in the object's memory!).

We use this idea to better capture the semantics of a reincarnation. The destruction of an object is modeled in our system by setting its space-id to -1. This is to say that the object no longer exists physically. This idea can be used to capture the semantics of an object being either physically destroyed or being hidden away only to reappear at a later time. This allows for a better analogy between the identity concepts of existence and non-existence as compared to the visual domain where existence can be equated with visible and non-existence (or hiding) with invisible.

A. Create

The create operation creates a new object that didn't exist previously. In [4], it is described visually as

In our implementation, this causes the following data to be recorded in the database. We suppose the object was created at time t1.

Semantic Table

Id	Reference
1	A

Space Table

Id	Shape	Previous	Next
1	object-shape	-1	-1

Time Table

Id	Time
1	t1

Relation Table

Semantic-id	space-id	Time-id
1	1	1

B. Destruct

The destruct operation destroys an object that was previously created. In [4], it is described visually as:

Suppose the object was created at time t1 and destroyed at time t2.

Semantic Table

Id	Reference
1	A

Space Table

Id	Shape	Previous	Next
1	object-shape	-1	-1

Time Table

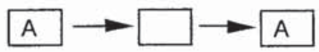
Id	Time
1	t1
2	t2

Relation Table

Semantic-id	space-id	Time-id
1	1	1
1	-1	2

C. Reincarnate

This depicts a sequence of operations in which an object is destroyed and subsequently recreated (with the same or different spatial features). In [4], it is described visually as:



Suppose the object was created at time t1, destroyed at time t2, and then recreated at time t3.

Semantic Table

Id	Reference
1	A

Space Table

Id	Shape	Previous	Next
1	Object type	-1	-1
2	Object type	-1	-1

Time Table

Id	Time
1	t1
2	t2
3	t3

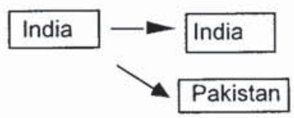
Relation Table

Semantic-id	space-id	time-id
1	1	1
1	-1	2
1	2	3

Notice that it is the same object (semantic-id = 1) that was created at time t1, destroyed at time t2, and then recreated with a new space-id at time t3.

D. Spawn

This depicts the type of change where a new object is issued from an existing object. The new object has a new identity that is unique and separate from that of the original object. The original object and identity continue to exist. For example, following [4], at the time India gained independence from Britain, Pakistan was spawned, while India's identity continued to exist.



We suppose that India was created at time t1 and Pakistan was spawned at time t2.

Semantic Table

Id	Reference
1	India
2	Pakistan

Space Table

Id	Shape	Previous	Next
1	polygon	-1	2
1	polygon	-1	3
2	polygon	1	-1
3	polygon	1	-1

Time Table

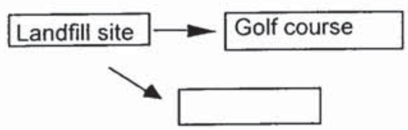
Id	Time
1	t1
2	t2

Relation Table

Semantic-id	space-id	time-id
1	1	1
1	2	2
2	3	2

E. Metamorphose

This depicts another type of change where a new object is issued from an existing object, but in this case, the original object ceases to exist. For example, following [4], when a landfill site is reclaimed and becomes a golf course.



We suppose that the landfill site was created at time t1 and became a golf course at time t2.

Semantic Table

Id	Reference
1	Landfill site
2	Golf course

Space Table

Id	Shape	Previous	Next
1	polygon	-1	2
2	polygon	1	-1

Time Table

Id	Time
1	t1
2	t2

Relation Table

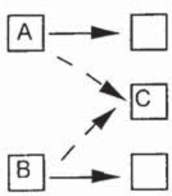
Semantic-id	space-id	time-id
1	1	1
1	-1	2
2	2	2

F. Join

This depicts the type of change when two objects join together and form an object with a new identity. The following presents three different classes of joins as they appeared in [4]:

F.1. Merge

In the case of a merge, the original objects are destroyed at the time of the merge and an object with a new identity is issued. In [4], it is described visually as:



We suppose that objects A and B were created at time t1 and the merge took place at time t2.

Semantic Table

Id	Reference
1	A
2	B
3	C

Space Table

Id	Shape	Previous	Next
1	object type	-1	3
2	object type	-1	3
3	object type	1	-1
3	object type	2	-1

Time Table

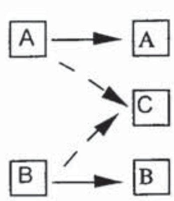
Id	Time
1	t1
2	t2

Relation Table

Semantic-id	space-id	time-id
1	1	1
2	2	1
1	-1	2
2	-1	2
3	3	2

F.2. Generate

This type of change is similar to a merge, but in this case, the original objects continue to exist. Parenthood is the classic example of this type of issuance. In [4], it is described visually as:



We suppose that objects A and B were created at time t1 and C was created at time t2.

Semantic Table

Id	Reference
1	A
2	B
3	C

Space Table

Id	Shape	Previous	Next
1	object type	-1	3
1	object type	-1	5
2	object type	-1	4
2	object type	-1	5
3	object type	1	-1
4	object type	2	-1
5	object type	1	-1
5	object type	2	-1

Time Table

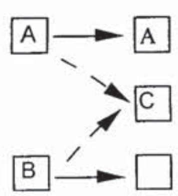
Id	Time
1	t1
2	t2

Relation Table

Semantic-id	space-id	time-id
1	1	1
2	2	1
1	3	2
2	4	2
3	5	2

F.3. Mix

This is the third type of join where one of the parent objects is destroyed in the join, but not both as with a merge.



We omit the implementation of mix as it can be easily derived from the implementations of merge and generate.

3.2. Example: Tracking a convoy of vehicles

In this section, we borrow the example used in [4] to show how to model a scenario of changes involving appearance and disappearance of objects.

Suppose we have at time t1 a birds-eye view of four vehicles that collect to form a convoy at time t2, and then another vehicle joins the convoy at time t3. Now consider another change in events, as the convoy disappears from view when the road winds through a forest at time t4 and then partially emerges from the forest at time t5.

In our system, this scenario can be represented as follows. First, a set of four individual vehicles is visible on the roads at time t1.

Semantic Table

Id	Reference
1	A
2	B
3	C
4	D

Space Table

Id	Shape	Previous	Next
1	point	-1	-1
2	point	-1	-1
3	point	-1	-1
4	point	-1	-1

Time Table

Id	Time
1	t1

Relation Table

Semantic-id	space-id	time-id
1	1	1
2	2	1
3	3	1
4	4	1

The vehicles collect to form a convoy at time t2 and a new car E appears far in the distance.

Semantic Table

Id	Reference
1	A
2	B
3	C
4	D
5	Convoy
6	E

Space Table

Id	Shape	Previous	Next
1	Point	-1	6
2	point	-1	7
3	point	-1	8
4	point	-1	9
6	point	1	-1
7	point	2	-1
8	point	3	-1
9	point	4	-1
5	line	-1	-1
10	point	-1	-1

Time Table

Id	Time
1	t1
2	t2

Relation Table

Semantic-id	space-id	time-id
1	1	1
2	2	1
3	3	1
4	4	1
1	6	2
2	7	2
3	8	2
4	9	2
5	5	2
6	10	2

The addition of the vehicle E to the convoy at time t3 can be modeled as follows:

Semantic Table

Id	Reference
1	A
2	B
3	C
4	D
5	Convoy
6	E

Space Table

Id	Shape	Previous	Next
1	point	-1	6
2	point	-1	7
3	point	-1	8
4	point	-1	9
6	point	1	12
7	point	2	13
8	point	3	14
9	point	4	15
5	line	-1	11
10	point	-1	16
12	point	6	-1
13	point	7	-1
14	point	8	-1
15	point	9	-1
16	point	10	-1
11	line	5	-1

Recall that the most current configuration of the system can be derived from the space table by only considering those records whose Next field is set to -1.

Time Table

Id	Time
1	t1
2	t2
3	t3

Relation Table

Semantic-id	space-id	time-id
1	1	1
2	2	1
3	3	1
4	4	1
1	6	2
2	7	2
3	8	2
4	9	2
5	5	2
6	10	2
1	12	3
2	13	3
3	14	3
4	15	3
5	11	3
6	16	3

At time t4, the convoy has entered the forest and no vehicles are visible.

Semantic Table

Id	Reference
1	A
2	B
3	C
4	D
5	Convoy
6	E

Space Table

Id	Shape	Previous	Next
1	point	-1	6
2	point	-1	7
3	point	-1	8
4	point	-1	9
6	point	1	12
7	point	2	13
8	point	3	14
9	point	4	15
5	line	-1	11
10	point	-1	16
12	point	6	-1
13	point	7	-1
14	point	8	-1

15	point	9	-1
16	point	10	-1
11	line	5	-1

Time Table

Id	Time
1	t1
2	t2
3	t3
4	t4

Relation Table

Semantic-id	space-id	time-id
1	1	1
2	2	1
3	3	1
4	4	1
1	6	2
2	7	2
3	8	2
4	9	2
5	5	2
6	10	2
1	12	3
2	13	3
3	14	3
4	15	3
5	11	3
6	16	3
1	-1	4
2	-1	4
3	-1	4
4	-1	4
5	-1	4
6	-1	4

At time t5, the convoy partially emerges from the forest. Two vehicles are now visible.

Semantic Table

Id	Reference
1	A
2	B
3	C
4	D
5	Convoy
6	E

Space Table

Id	Shape	Previous	Next
1	point	-1	6
2	point	-1	7
3	point	-1	8
4	point	-1	9
6	point	1	12
7	point	2	13
8	point	3	14
9	point	4	15
5	line	-1	11
10	point	-1	16
12	point	6	-1
13	point	7	-1
14	point	8	-1
15	point	9	-1
16	point	10	-1
11	line	5	-1
17	line	-1	-1
18	point	-1	-1
19	point	-1	-1

Time Table

Id	Time
1	t1
2	t2
3	t3
4	t4
5	t5

Relation Table

Semantic-id	space-id	time-id
1	1	1
2	2	1
3	3	1
4	4	1
1	6	2
2	7	2
3	8	2
4	9	2
5	5	2
6	10	2
1	12	3
2	13	3
3	14	3
4	15	3
5	11	3
6	16	3
1	-1	4
2	-1	4
3	-1	4
4	-1	4
5	-1	4
6	-1	4
1	18	5
2	19	5
5	17	5

Chapter IV

CASE STUDY

In this chapter we introduce the GISEL project (Geographic Information Systems Electricity of Lebanon), along with its applications. We import the data from the GISEL project into a three-domain model based system. We take one of the layers, the transformer layer, represent it in the three domain model, describe a set of transactions that can occur on these transformers and see how these transactions can be implemented in the three domain model. The chapter concludes with a number of typical spatio-temporal queries along with the SQL statements which provide the answers.

4.1. The GISEL Application

This case study uses an application named GISEL that was originally developed by the GIS group at Khatib & Alami Consolidated Engineering Company (K&A). GISEL consists of several applications (Facility Siting, Trouble Call, Maintenance, Collection Management, Work Order, ArcFM, Switching, etc.) for the area of municipal Beirut. Mainly, a GIS system was developed to model the existing network of the EDL (Electricity of Lebanon). This includes parcels, buildings, and substations, MT/BT (Medium-Tension, to Low-Tension transformer rooms), transformers, medium tension lines, low tension distribution lines, etc.

The next figure below shows a capture of municipal Beirut with its 60 sectors.

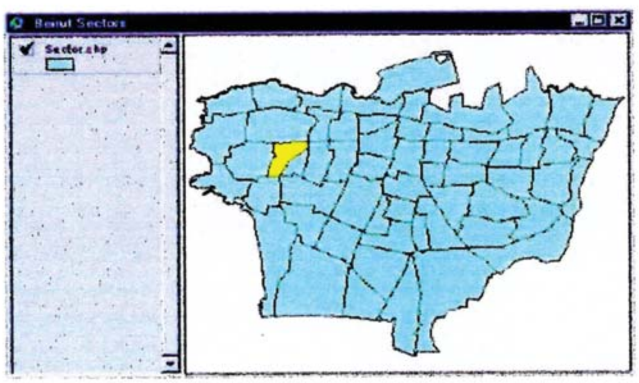


Figure 4.1: Municipal Beirut

Another figure shows the above yellow sector with its parcels.

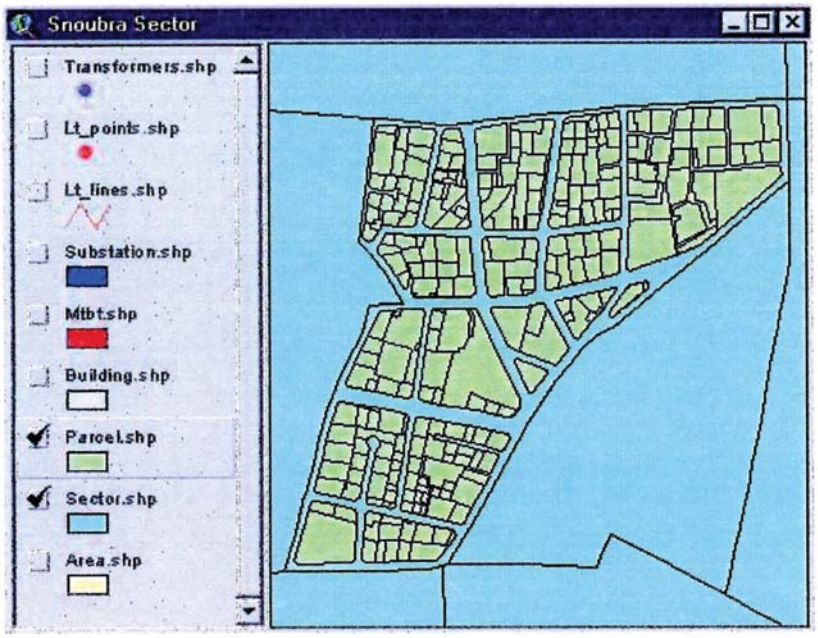


Figure 4.2: Parcel representation in one sector in Municipal Beirut

4.2. GISEL Implemented in the Three-Domain Model

In this section, we consider further this application and develop a system based on the three-domain model to keep track of the locations of the different transformers that are part of the EDL network. This allows us to answer a number of spatio-temporal queries that are of great importance to some of the EDL applications such as the maintenance application, the Trouble Call, The Collection Management, and The Switching application.

System Setup: Our application is initially written for Arcview 3.1 software, and then customized using the avenue code. The data used is that of the entire area of municipal Beirut, but only one sector is used in the examples for this paper. This sector includes 30 MT/BT (Medium-Tension to Low-Tension Transformer Rooms), and 36 transformers located in these MT/BTs.

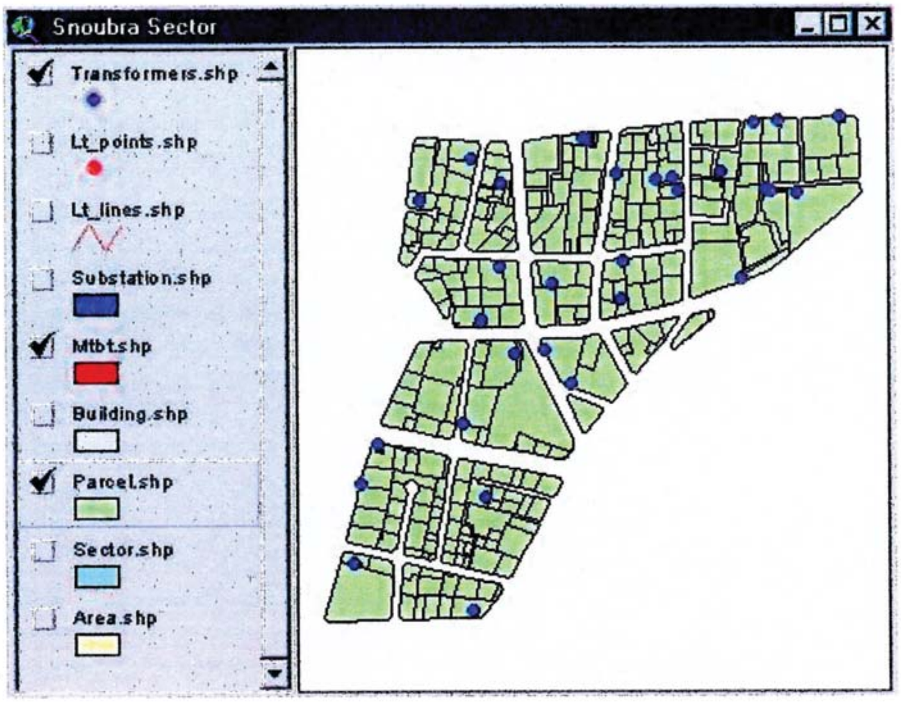


Figure 4.3: Transformer distribution

Some of the possible scenarios that are possible to occur on our network include transactions that can be done on transformers from simple switching operations to maintenance to change location or removal. Our model adopts all these operations, from changing related attributes of the transformers to changing their spatial locations. The transactions are reflected in the model studied. The model will be able to log and retrieve these transactions. It will also be able to answer many other queries about our database for the period when these transactions occurred. We will see sample data behind these operations and examples of queries that can be resolved using this model.

Assume the following sequence of events…

1999: Transformer # 18 has been moved to MT/BT # 2697

A New transformer has replaced the old one.

2000: Transformer # 24 has been removed from MT/BT # 2590

2001: Transformer (previously # 18) has been moved to MT/BT 2698.

A new upgraded transformer has been installed

The following figures show captures of the views and the underlying tables before and after the changes (Note that in this implementation we have chosen to store the current configuration in a separate table.)

The next figure shows the state of the concerned transformers at time 1998

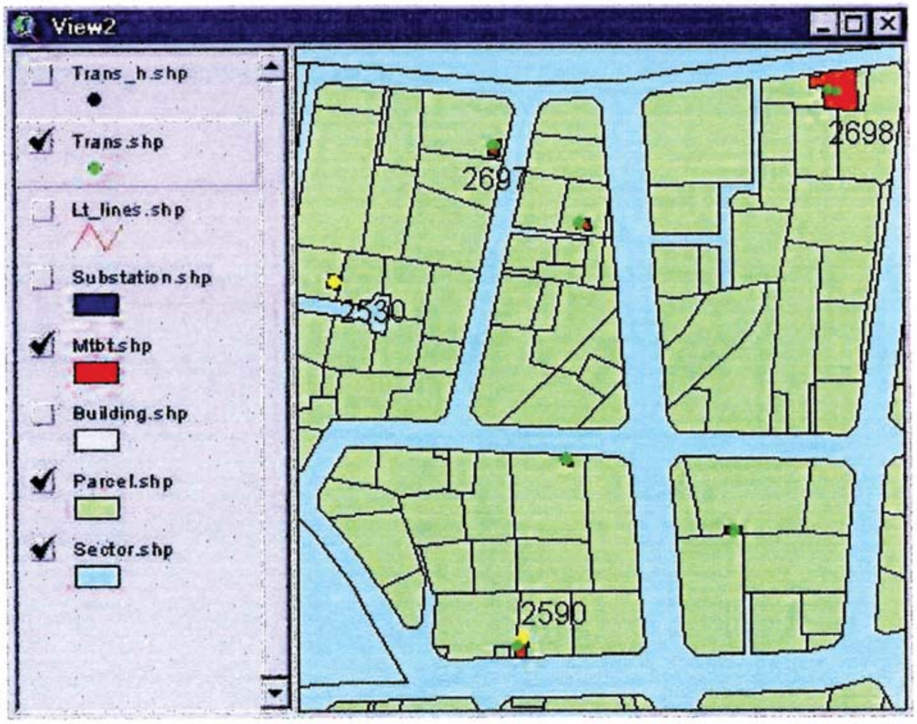


Figure 4.4: Transformer state 1998

The corresponding tables are shown in the figures below…

Attributes of Trans.shp

Shape	Id	Pt no	Pt ty	Status	Trans_no	Maintr_no	Dev_no	Mtbf_no	Manuf	Op_v
Point	17	172983	TRN	IN	2	0	0	2743	Matelec	220
Point	18	173004	TRN	IN	1	0	0	2530	Matelec	110
Point	19	173076	TRN	IN	1	0	0	702	Matelec	220
Point	20	173082	TRN	IN	1	0	0	2507	Matelec	220
Point	21	173092	TRN	IN	1	0	0	274	Matelec	220
Point	22	173099	TRN	IN	1	0	0	2506	Matelec	220
Point	23	173125	TRN	IN	1	0	0	2559	Matelec	220
Point	24	173169	TRN	IN	1	0	0	2590	Elprom	110
Point	25	173177	TRN	IN	2	0	0	2590	Matelec	110

Table 4.1: 1998 - the Space Table

trans_semantics.dbf

Reference	Sem_id
212001	17
500002	18
3055007	19
3034016	20
1518004	21
32530	22
37082	23
57701	24
2830015	25
2391001	26

Table 4.2: 1998 – The Semantic table

trans_time.dbf

Date	Time_id
1984	12
1985	13
1988	14
1990	15
1992	16
1993	17
1994	18
1996	19
1997	20
1998	21

Table 4.3: 1998 – The Time Table

trans_relation.dbf

Sem_id	Time_id	Space_id
16	7	16
17	16	17
18	13	18
19	18	19
20	18	20
21	15	21
22	10	22
23	11	23
24	3	24
25	18	25
26	19	26
27	8	27

Table 4.4: 1998 – Relation Table

Attributes of Trans_h.shp

Shape	Id	Prev_id	Next_id
Point	18	-1	37
Point	37	18	39
Point	38	-1	-1
Point	39	37	-1
Point	40	-1	-1
Point	1	-1	-1
Point	2	-1	-1
Point	3	-1	-1

Table 4.5: 1998 – The History table

Next we will see the **2001** data status

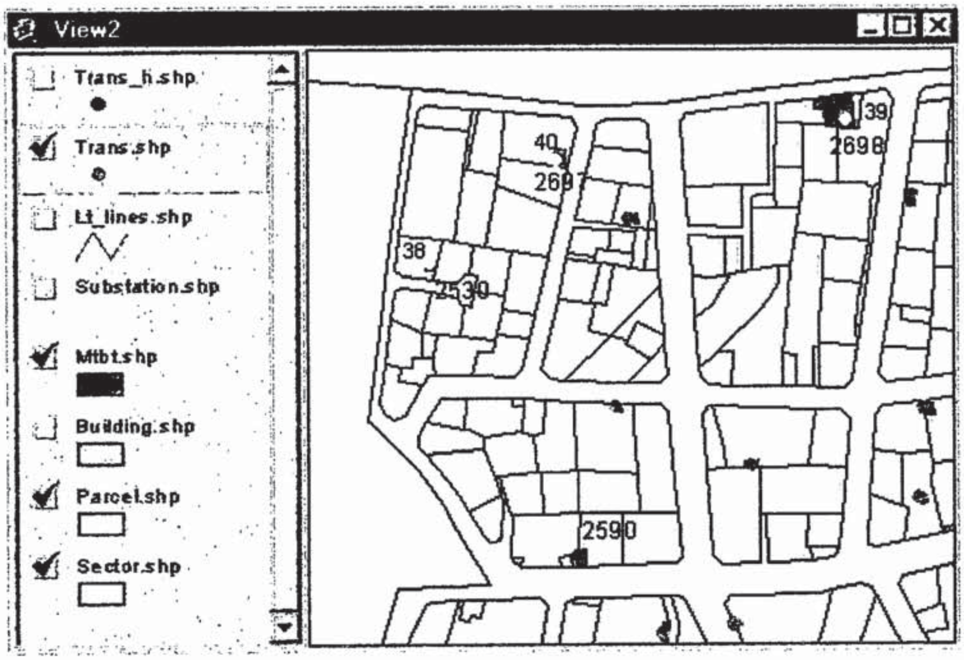


Figure 4.5: Transformer state 2001-View

Attributes of Trans.shp

Shape	[illegible]	Pt_no	Pt_t	Status	Trans_nc	Mainbr_nc	Dev_nc	Mtbt_nc	Manuf	Op_v
Point	32	173437	TRN	IN	1	0	0	2554	Matelec	220
Point	33	173459	TRN	IN	1	0	0	2816	Matelec	220
Point	34	173546	TRN	IN	1	0	0	2553	Matelec	220
Point	35	173632	TRN	IN	1	0	0	2568	Matelec	220
Point	36	182928	TRN	OU	1	0	0	2743	Elprom	220
Point	38	0	TRN	IN	1	0	0	2530		220
Point	39	173004	TRN	IN	1	0	0	2530	Matelec	110
Point	40	0	TRN	IN	1	0	0	2697		220

Table 4.6: 2001 – The Space Table

trans_semantics.dbf

Reference	Sem_id
424040	32
2349022	33
39904	34
1914053	35
55796	36
333225	37
222336	38

Table 4.7: 2001 – The Semantic Table

trans_time.dbf

Date	Time_id
1997	20
1998	21
1999	22
2000	23
2001	24

Table 4.8: 2001 – The Time Table

trans_relation.dbf

Sem_id	Time_id	Space_id
35	16	35
36	2	36
18	22	37
37	22	38
24	23	-1
18	24	39
38	24	40

Table 4.9: 2001 – The Relation Table

Attributes of Trans_h.shp

Shape	Id	Prev_id	Next_id
Point	17	-1	-1
Point	18	-1	-1
Point	19	-1	-1
Point	20	-1	-1
Point	21	-1	-1
Point	22	-1	-1
Point	23	-1	-1
Point	24	-1	-1
Point	25	-1	-1

Table 4.10: 2001 – The History Table

Sample Queries:

Queries about attributes of entities:

These queries seek information about attributes of features and entities. Example: What is the status of transformer X? What was the rating of Transformer Y on 21/5/1999?

Queries about locations, spatial properties, and spatial relationships

They answer questions about where and what. Where was Transformer X at 1995? Which Transformers were in MT/BT Y at time January 1996?

Queries about time, temporal properties, and temporal relationships

What happened to Transformer X from 1996 to 1997? In Which Month did we have most number of switching operations?

Queries about spatio-temporal behaviors and relationships

What happened to the transformers of MTBT Y from Jan. to April 1999?

In which Month and In Which Mt/Bt did we have the largest number of repairs?

To answer:

What is the status of transformer X?

Select Transformer ID X from Current Space Table,
Query Value for selected record for status (In Service/Out of Service)

Where Was Transformer X in 1995?

From relation Table select space id = X.
Get Time ID
From Time Table check if time <= 1995 then go to Space table and get its location (Shape).
Otherwise, go to History table and get its previous id value. If previous-id = -1 then the transformer X did not exist in 1995.
Go to the beginning and check again

What happened to transformer X from 1996 to 1997?

Select Transformer X from Relation Table. Get its Time ID.
From Time table, if time < 1996, then transformer X attributes have not changed values during the given interval. In this case, go to the space table and get its location (shape); stop.

```
(a) If 1996 <= time <= 1997, then
   select Transformer from History table
   Get its previous ID.
   Get its (previous ID) time from the Relation table.
If time <=1996 then
      select Transformer (corresponding to previous id) from History table
      stop
   otherwise goto (a)
if 1997 < time then
   Get its previous ID.
   Get its (previous ID) time from the Relation table.
   If time <=1997 then goto (a)
```

CHAPTER V

CONCLUSION AND FUTURE RESEARCH

In this thesis, we described the concepts behind spatio-temporal modeling in GIS. We started by describing the difference between static and dynamic change representation and then moved to spatio-temporal models. Among these models we chose the three-domain model and described a system based on it to provide a system-level implementation for identity-based changes. In addition, we showed the model's capability in supporting spatio-temporal queries. The added flexibility provided by the three-domain model, where the space, time, and semantic domains are dealt with separately in three different tables and then linked together via a fourth table, gives this model a tremendous maneuver capability in handling real-world scenarios.

Next we presented a real case study, implemented on GISEL project, to show the capability of the three-domain model to support spatio-temporal queries. We provided a set of transactions that occur on the transformer network for electricity of Lebanon, and implemented these transactions in the three-domain model. We presented some spatio-temporal queries and showed how to solve these queries using typical SQL statements.

Directions for future research could include:

(i) The issue of displaying the answers to the queries visually rather than using SQL statements.

(ii) The implementation of identity-based change operations in relation to composite objects.

(iii) The issue of an efficient recovery of old versions, especially when dealing with very large databases.

References

[1] F. Bonfatti and P.D. Monari, "Spatio-Temporal Modeling of Complex Geographical Structures", Computer Support for Environmental Impact Assessment: Proceedings of the IFIP TC5/WG5.11 Working Conference on Computer Support for Environment Assessment, CSEIA 93, Como, Italy, Elsevier, 1994

[2] C. Claramunt and M. Thuriault, "Managing Time in GIS: An Event-Oriented Approach," Recent Advances on Temporal Databases, Clifford J. and Tuzhilin A. (eds.), Springer-Verlag, pp. 23-42, 1995

[3] M. Erwig, R. H. Guetting, M. Schneider, and M. Vazirgiannis, "Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases," Chorochronos, Technical Report, CH-97-08, 1997

[4] K. Hornsby and M. J. Egenhofer, “Qualitative Representation of Change,” in: S. Hirtle and A. Frank (eds.), Spatial Information Theory: A theoretical Basis for GIS, Proceedings of the International Conference COSIT’97, Lecture Notes in Computer Science 1329, Springer-Verlag, pp. 15-33, 1997

[5] K. Hornsby and M. J. Egenhofer, "Identity-Based Change Operations for Composite Objects," Eight International Symposium on Spatial and Data Handling, Vancouver, Canada, T. Poiker and N. Chrisman (eds.), pp. 202-213, July 1998

[6] S. Khoshafian and G. Copeland, “Object Identity,” SIGPLAN Notices 21: 406-416, 1986

[7] G. Langran, "A Framework For Temporal Geographic Information Systems," *Cartographica*, 25 (3), 1988

[8] G. Langran, "Time in Geographical Information Systems," (ed.) Taylor and Francis, London, 1992

[9] D. Peuquet, "It's About Time: A Conceptual Framework for the Representation of Temporal Dynamics in GIS", Annals of the American Association of Geographers, 84 (3):441-461

[10] A. Pavlopoulos and B. Theodoulidis, "Review Of Spatio-Temporal Data Models," Timelab Technical Report, Tr-98-3, October, 1998

[11] A. Renolen, "History Graphs: Conceptual Modeling of Spatio-Temporal Data," in: GIS Frontiers in Business and Science, Vol. 2, International Cartographic Association, Brno, Czech Republic, 1996

[12] A. Renolen, "Conceptual Modeling and Spatio-Temporal Information Systems: How to Model the Real World," in: Hans Hauska, Editor, SCANGIS'97, Proceeding of the 6th Scandinavian Research Conference on Geographical Information Systems, Stockholm, Sweden, June 1997

[13] A. Renolen, "The Spatio-Temporal Object Model: A Conceptual Model Based on Ontology," Article submitted to Geoinformatica, 1997

[14] B. Smith, On the Origin of Objects, Cambridge, MA, The MIT Press, 1996

[15] K. Al-Ţaha and R. Barrera, "Identities Through Time," in: M. Ehlers and D.R.Steiner (eds.), International Workshop on Requirements for Integrated Geographic Information Systems, New Orleans, LA, pp. 1-12, 1994

[16] N. Tryfona, D. Pfoser And Th. Hadzilacos, "Modeling Behavior of Geographic Objects: An Experience With the Object Modeling Technique," The 9th International Conference on Advanced Information Systems and Engineering (CAISE), Lecture Notes in Computer Science, Barcelona, Catalonia, June 1997

[17] M.F. Worboys, H.M.Hearhshow, D.J.Maguire, "Object-Oriented Modeling for Spatial Databases," *Int. Journal of GIS*, Vol. 4, No. 4, 1990

[18] M. Worboys, "Object-Oriented Approaches to Geo-Referenced Information," International Journal of Geographical Information Systems 8 (4):385-399, 1994

[19] M. Yuan, "Wildfire Conceptual Modeling for Building GIS Space-Time Models, " Proceedings: GIS/LIS'94, pp. 860-869, 1994

[20] M. Yuan, "Modeling Semantic, Temporal, and Spatial Information in Geographic Information Systems," in: M. Craglia And H. Couclelis (eds.) Geographic Information Research: Bringing The Atlantic, pp. 334-347. London, Taylor and Francis Inc., 1996

[21] M Yuan, "Use of Three-Domain Representation to Enhance GIS Support for Complex Spatiotemporal Queries, *Transactions in GIS* 3 (2): 137-160

[22] William E. Huxhold, Allan G. Levinshon, "Managing Geographic Information System Projects", Oxford University Press, 1995