2006

# Spatio-temporal applications

Swati Prabhu
*San Jose State University*

# SPATIO-TEMPORAL APPLICATIONS

A Thesis

Presented to

The Faculty of the Department of Geography

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Arts

by

Swati Prabhu

August 2006

UMI Number: 1438589

## INFORMATION TO USERS

APPROVED FOR THE DEPARTMENT OF GEOGRAPHY

Dr. Richard Taketa

Prof. Gary. Pereira

Jay Marianowits, Stanford University

APPROVED FOR THE UNIVERSITY

# ABSTRACT


## SPATIO-TEMPORAL APPLICATIONS

by Swati Prabhu

This thesis addresses the topic of GIS (Geographic Information Systems) systems that deal with issues of spatial and temporal applications. It examines some of the categories of spatio-temporal issues and the broad concepts of approaches various applications utilize to address them.

Research on this subject reveals that although very interesting advancement has been made in the field of theoretical progress, real world solutions are still lacking in complementary development. The study focuses on analyzing the real world software solutions available and applies the three basic concepts of potential solutions to the case study. The conclusion arrived at for the case study is proved to not be applicable in general for all other spatio-temporal issues.

# ACKNOWLEDGEMENTS

This project would not have been possible without the support of many people. Many thanks to my advisor, Dr. Richard Taketa, who patiently read my numerous revisions and provided support all along. Also thanks to my committee members, Gary Pereira and Jay Marianowits, who offered guidance and support and made this possible.

I take this opportunity to thank my parents for dreaming and encouraging me. And finally, thanks to my husband for enduring this long process with me, always offering support and love.

TABLE OF CONTENTS

## LIST OF TABLES

LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

This study addresses issues of maintaining and representing changes over time for spatially dynamic entities whose attributes also change. There are various types of such spatio-temporal changes that could occur. Some of the issues faced today in various GIS applications relating to spatio-temporal conditions include:

- Attributes change over time for a given spatial entity or feature.
- The geometries of the entity change over time but attributes remain the same.
- Geometries morph over time and so do the attributes.

Obviously, the third item would be the most complex in terms of the database design, graphical representation, and modeling. Examining the morphing of the geometries leads us to determine the change types. The changes could be in specific or random directions. They could be between two objects that are interdependent where a change in one would necessitate a change in the other. Below are outlined some instances of changes in both spatial and attribute data over time that may be expected.

The geometric entity representing the phasing area for a construction site could become larger during the course of a project or reduce in size towards the close of the project. The extents of the feature would most probably reduce as the project nears completion. In most cases, it is possible to predict the extents of this laydown area based on the various phases along the timeline of the project. Tracking this information is critical from a planning perspective for logistics and to communicate impacts to the general public.

Tracking all the areas of construction activity poses a significant challenge. The temporal element further complicates the graphical representation and database design of these entities. Areas or polygons on a map may change their functionality over time from being a lay down area to a construction site and sometimes back to a lay down area. The attributes associated with the entities change based on its functionality. Occasionally, the extents of an area for a construction site may increase or decrease, although, all the attributes of the site remain the same. Also, a single polygon can morph in size and shape over time concurrently changing its attributes.

Another instance of a spatio-temporal change type are interdependent geometric elements that require greater planning and coordination, although it has an underlying similarity to the previous example. Changing one element needs to be reflected on its neighboring elements as well. Examples include land parcels that merge for a time period only to be divided again, or the layout of a cubicle space in an office that may change needing to be reflected on adjoining spaces, all within the confines of the exterior extents of the building. The geometrical change to the cubicle walls is contained within the solid outer walls. The direction of the change in this case is restricted by, and dependent on the features that make up the solid walls.

Another example is space planning and changes to the attributes with geometric layout changes on occasion. In a given year, room usage can change and so can the geometry of the space based on that usage. The space usage in a facility and particularly in a University environment change frequently. A given room can be split many number of times in a year or two rooms can be merged for a particular function. For example, a large room can be divided into office spaces and converted back into one large laboratory. Tracking this information is extremely important as the space usage reports are periodically audited for cost recovery to the government, for research grants, as well as for space planning. Representation and reporting these changes both graphically and in a database presents a huge challenge. Keeping track of the physical room numbers, the area or geometry of a room and its unique ID are only some of the issues to be addressed.

Maintenance of the floor plans to reflect these changes is an ongoing effort and minor room alterations may not even be captured graphically. The database that tracks attributes about these spaces, including room usage, also needs to be continually updated. Archiving of annual data, although in its place makes reporting of data from backups difficult.

A third instance, and by far the most complicated, involves continuously changing geometry with the smallest time interval between changes. These include features whose geometry continually changes but history of the dataset might not be critical in the reporting.

An example includes emergency management related situations like a spreading fire or flood areas where the geometry morphs in random directions in a very short time interval. Further examples are observed in transportation management and vehicle tracking analysis with point features depicting vehicles that are constantly moving spatially.

Table 1 reviews some of these spatio-temporal examples of real world challenges particularly faced in a facilities and infrastructure planning environment.

Table 1. Typical categories of Spatio-temporal issues.

| **Change type Scenario** | **Geometric configuration (spatial aspect)** | **Temporal Granularity** | **Notes** |
|---|---|---|---|
| Areas for Construction projects have geometric changes and also change in attributes. | The change is random in direction and not dependant on other features. Could be predicted in advance and predetermined. There may not be any inter-dependencies although there could be overlaps. | The temporal granularity of the change is down to a monthly level. | Spatial representation of the general area is needed and is flexible, but temporal and attribute data demands greater accuracy. |
| Parcels in a given area may merge into one APN number for a time period to be split up later. Attributes belonging to both entities need to be conveyed in one result or output. | Geometries are interdependent. Changes to one almost always need to be reflected on adjoining geometries in one or more feature classes. | May be down to an annual level but not finer in most cases. | Greater accuracy in spatial and temporal elements is important to be tracked. |
| Functions or usage of a particular room may change based on movable partitions, although the larger area in question may stay the same. Example: large conference rooms that may be used for smaller events by using the movable partitions. | Geometry may stay the same unless the temp partitions are graphically depicted. May be useful to depict it in some cases. | Changes may occur as often as everyday or several times in a day. | Is there a real need for a graphical representation? Would a tabular interface for scheduling be sufficient? |

Table 1 (continued)

| **Change type Scenario** | **Geometric configuration (spatial aspect)** | **Temporal Granularity** | **Notes** |
|---|---|---|---|
| Numerous examples exist in the field of transportation that have a spatio-temporal issues. | Locations of moving elements need to be spatially located and they may be point features. Some amount of dependency exists. | Temporal granularity is very fine with changes to be tracked down to a minutes. | Granularity and geometric configuration can be predicted to a large extent except during accidents/ detours |
| In Emergency management scenarios, a hazard (fire, flood, bio hazardous fumes) may spread in various directions. | The geometry of this could be in random direction without extensive predetermination. No dependencies exist. | Granularity varies based on the event. Can be down to a minute or be at an hourly granularity. | Growing emphasis on emergency preparedness warrants a need for tracking this spatial information temporally. |
| A space in room may get redefined. Example: by moving the semi-permanent cubicle walls. | Geometries are interdependent with tighter constraints enforced on the direction by the bounding walls. | The changes typically may vary over a few months. | How important is it to keep track of previous configurations? If not critical do we care about temporal data and could the record be overwritten every time change occurs? |

Some of the questions that this study will examine are outlined below.

What is an ideal GIS environment in terms of the database and graphical representation for elements whose spatial representation and attributes change dynamically? This study examines some of the existing solutions for representing temporal variations over time.

Does temporal granularity say more about the change needed? If a particular example has a smaller granularity, does it warrant a different type of database design versus a larger granularity?

Are there inherent similarities between the various issues listed above? If seemingly different issues can be grouped together based on temporal granularity and/or spatial configuration, could all of them be resolved with a single methodology?

To address the questions posed, research and literature reviewed in this subject are followed by an in-depth case study of one of the above scenarios.

# CHAPTER II

## LITERATURE REVIEW AND RESEARCH

Significant research has been ongoing in the field of temporal issues in GIS. However, not many commercial software applications have developed concurrently to support these theories. The complexity of the issue at hand and the various subtle nuances in the requirements makes commercial development a challenging endeavor.

The fundamental functions of any temporal GIS, according to Langran (1995), are inventory, analysis, updates, quality control, scheduling, and display. Graphical representation of temporal changes continues to be a challenge. Bertin's (1983) system of graphic variables was the first attempt to define elements of symbology that can be manipulated to convey some of the information. He did not explicitly address the temporal dimension. Morrison (1974) has included further variables; Caivano, DiBase et al. (1992) added three explicit temporal variables followed by three more by MacEachren (1995). But, however extensive the research in symbology and graphical representation, it is the database or data structure that drives the system.

A temporal GIS must be able to record temporal changes in both spatial objects and their attributes. Two kinds of time dimensions are important in constructing a temporal database: the world time or the valid time, and the database time. The valid time is the time period for which a spatial phenomenon is considered and database time is the time when a database transaction is recorded.

Many data models have been proposed for recording changes in spatial objects. Six of the most popular and supported ones are briefly explained below.

**Data Models**

The snapshot model stores spatio temporal information in a series of map layers depicting the same phenomenon over the entire space, one for each time slice. This is the most common and widely used form of data models implemented in applications today.

The update model stores only one full version of a data set, with new changes added as updates to be stored separately whenever they occur (Langran, 1995, Peuquet, 1994). A third option, the space-time composite model is similar to update model but stores both past and present data in the same layer and the topology is constantly maintained (Langran & Chrisman, 1988). This model allows historical information to be preserved by identifying spatial units that have unique attributes and existence in terms of time, but does not solve the problem of spatial objects that decompose progressively into smaller objects, thus requiring retroactive adjustment to their identifiers.

The 3D/4D model treats time as a fourth dimension and every spatial object is defined by coordinates in the form of either (x, y, t) or (x, y, t, z) (Hazelton et al. 1991). Another approach, the integrated model combines some of the above models to take advantage of their individual strengths and overcome some of their weaknesses (Peuquet, 1994, Kelmelis, 1991).

Finally, a three-domain model proposed by Yuan (1994) incorporates three domains (semantic, temporal, and spatial) and is designed to represent features that change characteristics and locations continually.

The snap shot model is what current GIS technology vastly supports. The update model may be implemented as an extension to some current GIS. The update data model (also named "base state with amendments") can be used on top of most data structures. The main disadvantage of this is that there is no easy way to store spatial topology changes over time (Hazelton, 1990).

The 3D/4D model is much more complex and requires a GIS to be developed from scratch. The ideal TGIS would have a true 4-dimensional vector data structure, but this requires a major software engineering effort.

**Relational Databases**

Simple tables, hierarchical and network models are considered obsolete in a database context and have progressed to relational databases and the newer concept of object relational databases. Relational Database Management Systems (RDBMS) can adequately represent only data that is precise and homogeneous. This is not an issue with most business applications, which have the same characteristics. Because of the linear nature of time, extensions of the relational model to include temporal information for traditional business applications have been developed. But for Geographic data, this poses many disadvantages. There have been a number of experimental implementations of geographic representations using RDBMS (e.g., Maguire, Stickler and Browning (1992), Shapiro and Harlick (1982), van Roessel (1987), Waugh and Healey (1987)). Detailed analyses have shown that the need to appropriately represent characteristically irregular geographic boundaries within the relational model imposes an inherent and significant performance handicap. Another difficulty is that intrinsically spatial operations (distance and direction calculations, spatial overlay etc.) are not part of native Structured query language (SQL). Egenhofer (1992) suggested that this is the result of attempting to incorporate spatial concepts into a conceptual data model framework that is fundamentally ill suited for representing spatial information.

Most recently, object-relational and object-oriented database management systems utilizing an object-oriented approach have been developed. These were originally developed to provide programming tools at the implementation level, so that representation of entities as understood in a normal, conceptual sense could more easily be built. The basic principles, particularly encapsulation and inheritance, represent an advance and significant departure from traditional data modeling approaches. Data elements are distinct and unique to each object. In this model, geographic data elements are still conceptually divided into the same point, line, area or grid cell units, but they are now considered objects.

These systems have been criticized for a lack of a firm theoretical foundation, and for taking a step back into ad hoc database system implementation (Darwen & Date, 1998, Stonebraker et al. 1990). Donna Peuquet (2001) rightfully asks, "Why is the GIS field in a primitive and artificial ontology of points, lines, polygons and pixels in computer representation?" Why does the data model used for computer representation drive the user view and the kinds of analyses that can be performed?

Various experts have extensively addressed the notion of two kinds of views in a temporal GIS. There is a distinct difference between time as a snapshot view (discrete view), which is a series of views when a change is noted versus the continuous view, which encompasses all phenomenon for the given entity at any point of time.

In the discrete view, distinct entities are the basis of representation. Their spatial and temporal extents are denoted as attributes attached to these entities. In the continuous view, the basis of the representation is space and time. Individual objects are denoted as attributes attached to a given location in space and time. Using land ownership information, the particular parcel's number would be an attribute of the entire space it occupies, with locations denoted in some continuous coordinate field. Because these two views are duals of each other, the same spatio-temporal phenomenon can be described as either discrete or continuous. Computer data models should provide the choice between the two views as dual models within the database design (Peuquet, 2001).

Current Database Management Systems (DBMS) can handle discrete changes however continuous change can only be implied from one recorded state to the next for any given entity that is tracked within the database. There is no commercially available RDBMS with more than rudimentary temporal capabilities. But there is a growing consensus on a unifying conceptual data model, known as the bi-temporal conceptual data model (Jensen and Snodgrass, 1999). This has the particular advantage of being compatible with existing relation approach and ease of implementation on an existing non-temporal DBMS. The severe disadvantage this model poses is that it only captures discrete changes in a spatial context for rigid or point objects.

Continuous change requires a different representational and data model approach. There are numerous other examples that involve change of objects in space as they both move and change shape (forest fires, urbanized areas). One exception is cadastral systems, in which property boundary changes are recorded in discrete transactions.

Existing GIS incorporate temporal data by simply recording a sequence of "snapshot" images in the same way that a series of attribute "layers" are recorded. This model is more restrictive than the discrete changes of temporal transaction in a DBMS. Not only is change limited to representation in discrete increments but also temporal increments apply to every location whether or not a change has occurred at a given location.

The snap shot approach maintains the conceptual simplicity of the raster model and the world state for any stored time that can be retrieved directly; the changes between world states are not explicitly stored and can only be interchanged from two successive stored world states. In particular, a snapshot misses the key nature of change; the linkage to processes beyond the forms detected on the images. As already mentioned, changes cannot be recorded selectively. This means that there is redundant data if a location has not changed from one snapshot to the next. Thus a snapshot model fools the existing GIS, with the additional drawback that there is no query support for temporal relationships.

Another concept of amendment vectors, involves tracking changes in the initial location, time, and geometry. These changes are recorded as amendments to the original recorded map vectors and previous lines. This is a transaction-oriented representation of time. Beyond the limitations of handling only discrete changes and only boundary changes, the storage of these amendments becomes unwieldy as individual objects evolve over time, because these changes alter the topology of connected map vectors. The problem is compounded when various new objects come into existence, then perhaps disappear and then reappear (Peuquet and Qian, 1996).

Extending a vector model to incorporate an attribute of time is simple conceptually but becomes overly complex at implementation level. Research has shown that extending the spatial data model to include temporal data or vice versa will result in a form of implementation that is both complex and voluminous (Peuquet, 1994). Recent research has focused on the use of object-oriented approaches. Emphasis must shift from organizing space over time to representing a real world phenomenon in space and time (Wachowicz, 1999).

**Query Language in a Temporal GIS**

Most current GIS do not have a query language as such. Relational databases do use a language interface, Standard Query Language (SQL) that is supported by every RDBMS package. Its principle component is known as relational algebra, which involves a set of operators that take relations as their operands and return a new relation as a result. While SQL is based on relational algebra, there is a theoretical equivalent form called relational calculus. The key difference as described by Date (2000), is that relational algebra is prescriptive whereas relational calculus is descriptive. In relational algebra, the procedure needed is expressed whereas in relational calculus the desired result is described.

Example: List records for all houses on east Main Street that contain a commercial enterprise.

Relational algebra:

Join structure_inventor and land_use over street_Address, then,

Select from result where (struct = house) and land_use = commercial)

Expressed in the style of the relational calculus:

Get street_address for properties such that there exists a commercial enterprise in a house.

A query language called QUEL based on relational calculus did compete with SQL initially. Attempts to extend SQL to include spatial operators have not been very successful (Egenhofer, 1991).

**Space Dominant Views versus Time Dominant Views**

Torsten Hagerstrand unfolded the time Geography approach in early 1960. In this framework, an entity follows a space-time path, starting at the point of birth and ending at the point of death. Time and space are seen as inseparable. This is conceptualized as a succession of changes of location and events over a space-time path.

A space dominant view is characterized by viewing space as a container and elements exist only when associated to a layer or theme. Analysis is based on similarity or dissimilarity between aggregations at different points of time. Almost all GIS products have adopted the space dominant view within their data models (Wachowiz, 1999).

Most examples extend the relational database model by creating new versions of the table, column or attribute every time a change occurs. Their conclusion is that change is best incorporated as a component of the database at the attribute level, rather than at the column or table level. Langran (1995) also concludes that attribute versioning offers the most adequate approach for GIS applications presenting spatial dominance. A time dominant view is characterized by viewing time on a time line and events or actions are associated to the time line. Space is not an entity by itself.

Peuquet proposed the first attempt at integrating these two approaches in the method referred to as TEMPEST in 1994. TEMPEST uses a "triad model" (Peuquet, 1994) to query and manipulate time-based datasets. Other advances in manipulating time-based datasets have been made outside the GIS world. At a recent international workshop in Switzerland, software designers from around the world discussed issues ranging from Temporal Query Languages to alternative views on temporal data models (Clifford and Tuzhilin, 1995). Unfortunately, the direction for software developers is about as clear as the definition for time itself. With the exception of TEMPEST and a few modified query languages, the absence of adequate software tools to manipulate temporal data poses a significant limitation to widespread Temporal-GIS (TGIS) applications.

Depending on the number of time slices used to record events, the TGIS application may approach a real-time dynamic system. As the number of time slices increases, data volume increases as well. Likewise, larger data volumes mean a greater demand for CPU power. Even today's fastest supercomputers cannot adequately model the complexity of a developing thunderstorm in real time.

**Other Approaches and Research**

Animation is another method used to display temporal dimension. Dynamic maps use the snapshot data model creating frame-by-frame “map movies.” Animated maps have generated interest in the context of geographic visualization but as Peterson notes:

"What happens between each frame is more important than what exist on each frame" (1994, p.48) this reveals the problem of space-time representation visually, but this is a limitation of the underlying data representation.

Recently, a growing effort has included the areas of geographic visualization and virtual reality for geographic exploration and analysis. Both scientific visualization or VISC and virtual reality rely on sophisticated, modern computer graphics and fast processors. They attempt to integrate computer technology with human vision and cognition. VISC had its beginning in a report by McCormick, Defanti and Brown (1987). It focuses on the issue of 3-D and dynamics in displays for the purpose of understanding physical objects by allowing visually realistic renderings of them to be inspected.

The use of Virtual reality (VR) is a current area of interest. It refers to 3-D graphical rendering of a portion of a real or imagined environment either through a despot environment or utilizing specialized devices such as the Power wall or the CAVE. The intent is to place the user in a simulated environment.

These graphics based solutions do not provide as much flexibility and are still in their infancy requiring huge system resources without necessarily addressing all temporal issues.

# CHAPTER III

# STUDY OF EXISTING SOFTWARE AND SOLUTIONS

To further understand the real world possibilities this study inspects some of the solutions that exist today. The solutions can be broadly categorized into three areas, a snapshot approach, a versioning approach, and a spatial database approach.

## Snapshot Approach

The snapshot approach is the most rudimentary solution offered by most vendors. Some of the examples include ESRI's ArcMap and Autodesk's AutoCad. As discussed in the prior section there are several drawbacks to this solution for a spatio-temporal application, but this approach is still vastly used. The approach involves creating snapshots at a regular interval as a snapshot of the database state at a point in time. This type of information does not track intermediate changes that occur between the snapshots.

A change event is the thing that moves a database from one point in time or state to another point in time or state. Temporal granularity is the frequency between change events. The finest granularity at which temporal data can be recorded is at a transaction level. A change event is typically created for each database transaction. The temporal granularity dictates what information is captured and archived.

Snapshots may misrepresent the amount of change by missing multiple changes or by detecting no change when a change has occurred and then returned to an earlier condition. Currently, snapshots are created as discrete time chunks but, because of this, many changes in between are not captured or reported.

### Versioning Approach

The versioning approach involves storing data in the database and tracking changes using delta tables. ESRI's tool for versioning in a geodatabase is one of these potential solutions. Can versioning be a possible solution by not only facilitating as a series of temporary what if scenarios but as a more permanent record? Could versions be used as a permanent record of state instead of being a temporary tool for making edits to arrive at the final product?

"Modeling and using history in ArcGIS (May 2003)" – An ESRI technical paper deals extensively with the versioning scenario in temporal applications. The paper describes in detail the basic concepts and the means to possibly use versions to deal with historic information suggesting that the same applies to other applications.

History has been traditionally achieved through a workaround strategy of custom applications that store deleted features in a special history layer or by maintaining tables with active dates for each row. The geodatabase versioning model may provide an alternative to this method.

A lineage search may ask the question "Show how the feature "y" has changed through time?" Or "Show what is space of feature "z" at a time "b"?" One of the important points of how a versioned database maintains its edits is that objects are never deleted. All changes to a versioned database are inserts into delta tables. This aspect of the versioned database allows maintenance and temporal querying.

A version can be used to represent the state of a database at a specific point in time. The versioning model will maintain the old representation of objects, deleted objects, and the time that these events happened without having to manage special layers or date stamps on features.

A versioned database is a database that can have multiple persistent representations of its contents without a need for data replication. To manage versions, a versioned geodatabase contains a collection of states. States are a discrete snapshot of the database. There are two delta tables for each feature class or table: the add table and the deletes table. Compressing a versioned database simplifies the state tree and moves rows from the delta tables to the base tables of versioned feature classes and tables

In "Modeling our World" (Zeiler, 1999), five possible versioning workflows are presented.

- Direct editing: Maintenance of a single version of a database. The obvious drawback is that previous versions of features are not maintained.
- Two level tree: A series of single level of versions each of which is based on the default version. Changes need to be reconciled and posted to default versions.
- A multilevel tree: Child versions can themselves have children.
- Cyclical tree: A single chain of versions each of which has a single child that represents a step in the progression.
- Extended history: A series of versions, each of which represents the state of the geodatabase at a particular point of time. The main drawback of this

model is that the “add” and “delete” tables are never flushed during database compression.

Versioning is plainly a new version of the information being produced every time there is a change in the database. Generating a new version of the entire database for every change is unrealistic. New versions may be generated for tables, records and attributes associating one set of time stamps with the entire table, with each row in a table or each attribute in a table respectively.

Versioning at a table level results in a high degree of duplication but information retrieval will be simplest since the entire database table may be queried based on a given time slice. Record level versioning will reduce duplication but requires more processing. Versioning at attribute level is the most compact database but associated operations are most complex (Zhao, 2002).

The concept of Versioning is based on the premise that all temporal data is a snapshot in time. Each version is but a scenario at a given chunk of time. The issue this study addresses deals with time as a continuously changing element. Another aspect is the creation of delta tables, which by definition are temporary tables that are placeholders. Data has to reside in the original table and features are overwritten once reconciled with the parent table. This study needs to maintain changing attributes as well as the actual geometry. Creation of another feature is not a solution as the polygon is inherently the same with attributes and geometry translated loosely.

The whitepaper titled “Using ArcSde versions in real world” (Ewing, 2001) describes a Palm Beach county property Appraiser’s implementation of a feature management system using an ArcSde versioned geodatabase. The requirements were to provide for a means to track work in progress, to allow users to retrieve an historical view of certified tax roll for past years, to allow point in time views of map status at tax roll events and to support multiple events. He summarizes that the versioning workflows supported by ArcSde do not provide a sufficiently robust control to meet the needs of the appraiser's application. Versioning is a discrete incremental concept and not a continuous function. It failed because of two primary reasons: firstly ArcSde versions are not true versions but merely a collection of states. Secondly, in a geodatabase with a large number of versions the number of states and supporting database rows will increase much faster than the number of features. They fall short in two main areas: performance and flexibility. To address this, an application was built using feature level attributes to control geodatabase aspects such as history and point in time snapshots and to use a single default ArcSde version to provide work in progress visibility and maximize long-term geodatabase performance. The solution at Palm Beach County Property Appraisers Office was to use a combination of ArcSde Versioning and a set of feature attributes to allow a user to dynamically define a true version of the geodatabase.

This goes to prove further that ESRI's versioning, although a promising start in the field of spatio-temporal problems, may not address much of the issues involved. Both the snapshot and versioning approaches present two potential problems. Firstly there is the inflexibility for most temporal datasets to decide the granularity. Secondly, there is the issue of analyzing problems that span across two such discreet data chunks.

**Database Approach**

The final approach is the database centric approach of capturing and manipulating both spatial and temporal attribute data completely within the database. There are various vendors that offer solutions using this approach. Oracle Spatial provides a good framework for this approach. Vendors supporting this include Geomedia and MapInfo. Some other vendors seeking to transition include ESRI's ArcSde module that connects to Spatial as well as Autodesk's Spatial extension. DB2 is another object oriented spatial database that is developing recently.

A spatial database approach, like Oracle Spatial, provides an open and direct access to the simple feature geometries (points, lines, and polygons). This means that application developers and other vendors' products that also support Oracle Spatial can make use of the simple features in the central spatial database. The other benefits include interoperability with other reporting applications, and combining spatial data and attribute data in the same database.

This approach also provides value in terms of accommodating concurrent multi-user transactions and offers scalability in its architecture. Some vendors also support a data type like Spatial Data Object (SDO) that is native to the structure. This allows for standard SQL queries against the dataset. The other potential advantages to this approach include storing and indexing vector as well as raster data types in the database.

Typically, complex GIS data with many coordinates perform slower (e.g., retrieval and query) when stored with the spatial types of Oracle Spatial. This has to do with a number of functions including indexing and the way Oracle structures core data types.

Designing the database "intelligently", structuring the relationships and key fields, and leveraging the indexing capabilities can overcome some of the performance issues.

Oracle Spatial in particular includes R-tree indexing that can be applied to any data in relational databases. An R-tree index approximates each geometry with the smallest single rectangle that encloses the geometry called the minimum bounding rectangle, or MBR (Figure 1).

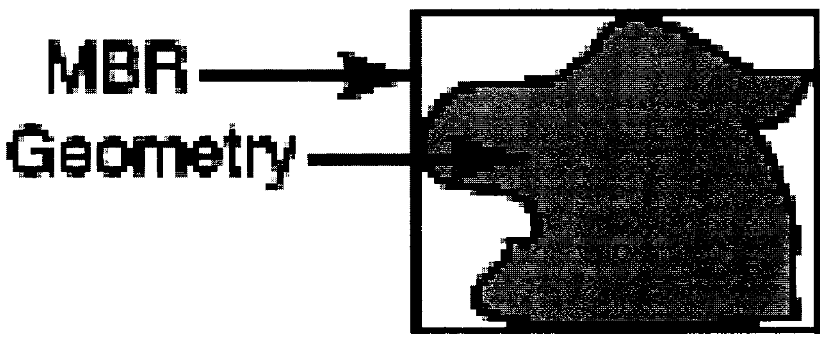


Figure 1. Oracle Spatial's minimum bounding rectangle concept for R-tree indexing.

For a layer of geometries, an R-tree index consists of a hierarchical index on the minimum bounding rectangles of the geometries in the layer. Because R-tree indexes are fast and work directly on geodetic data they are the preferred indexing mechanism for working with spatial data stored in the database. (Data sheet provided by Oracle, 2003).

A database centric concept can be implemented with many other software solutions provided by other vendors like DB2, ESRI, Geomedia to reap similar potential benefits.

# CHAPTER IV

# CASE STUDY

This section focuses on a detailed case study of one of the various examples of spatio-temporal issues addressed at the onset of this work. The postulate is that an in-depth analysis of one example and pursuing a case study might be applicable to other categories.

The criteria used to narrow down this focus are considerations about the flexibility in temporal granularity, the need for tracking modified data (archiving historic information about transactions) and finally, considerations about the need to have varying periods of dormancy. The final criteria addresses certain elements of the dataset that may need to remain inactive, while there is active progress in the temporal and attributed dataset for other elements. Table 2 reviews the examples for the above criteria.

Table 2. Categories of Spatio-temporal applications and analysis.

| **Category** | **Flexibility in temporal granularity?** | **Need for tracking modified data?** | **Varying Periods of active and inactive status of individual objects?** |
|---|---|---|---|
| Construction projects and laydown areas. | Quite flexible and dependant on user requests and resources availability. | The project managers making changes are fully responsible for the changes and information may be required occasionally, but is not relied on extensively. | Yes. Redundancies exist in duplicate geometries being created over multiple periods of active and inactive construction activity. |
| Parcels in a given area merge into one APN number for a time period to be split up again later. Attributes belonging to both entities need to be conveyed in one result or output. | The temporal changes need to be tracked very accurately in terms of the incident time but there is room for flexibility in the transaction recording time. | The database could be the system of record and all modifications and updates need to be tracked. | Yes. The entity and its record may be dormant but accurate history of that entity needs to be preserved along with its ID, as it is a single object. Data duplication is not possible. |
| Geometric configuration changes in large indoor spaces or, spaces may have various functions over a given period. | The changes are approximated to the nearest time periods and the transaction record time is posted regularly, however data reported may not be as regular. | All approved modifications for functional use needs to be tracked and the entire database can be archived. Need for archival reporting possible. | Yes. Most spaces/ records are typically active and the attributes are associated with the entity at all times. The physical ID may change over time and the tracking may be similar to the construction project tracking. |

Table 2 (continued).

| **Category** | **Flexibility in temporal granularity?** | **Need for tracking modified data?** | **Varying Periods of active and inactive status of individual objects?** |
|---|---|---|---|
| In Emergency management scenarios, a hazard (fire, flood, bio hazardous fumes) that spread in various directions. | The smallest possible granularity in terms of both transaction time and the incident time. | The system needs to be the system of record and there may be a need to know when the status of incident changed. | Possible, however at a given period of emergency response, the status of the incidents for a geometric location would typically be active until entire emergency situation is controlled. |
| Vehicular tracking in Transportation management field | No flexibility in the finest granularity on both incident and transaction time. Data redundancy is unavoidable | Historical tracking is purely for archival purposes | Possible for vehicles, but geometry is typically point features tracked along its path and not as complex. |

After reviewing the various criteria to be addressed, the construction projects tracking issue was chosen for the case study. This issue involved most of the considerations and struck a middle ground to the others. Furthermore, the geometric or spatial aspects of the elements would potentially encompass challenges posed by the other instances. The issue seemed to have the most potential for successfully implementing the three broad software approaches and practically analyzing them. Coincidentally, there were administrative benefits at the Stanford University campus as well.

The problem at hand was to keep track of the changing aspects of the phasing areas (lay down areas) and actual construction extents. Some of the needs for tracking this information include efficient project management and a planning tool that could result in significant saving in time and money for the staff. Another reason is better communication with the community and dependant populations for better activity planning. In an environment with constant reconstruction, remodeling and changes, a robust and reliable GIS can be a critical piece. The challenge is to manage this GIS in terms of the technology to be used, the graphical representation, information gathering, and the database design to store all this data. Some of the methods employed are detailed below in chronological order of development starting from its infancy.

Initially, a map for locating construction activities was developed only on demand as requested by a group. The project boundaries or extents were drafted in AutoCAD version 2000, with the temporality existing at different layers. Layers were named by years and the polygons depicting construction zones assigned to various layers (Figure 2).

Projects that spanned multiple time periods were either incorrectly represented or repeated on two layers.

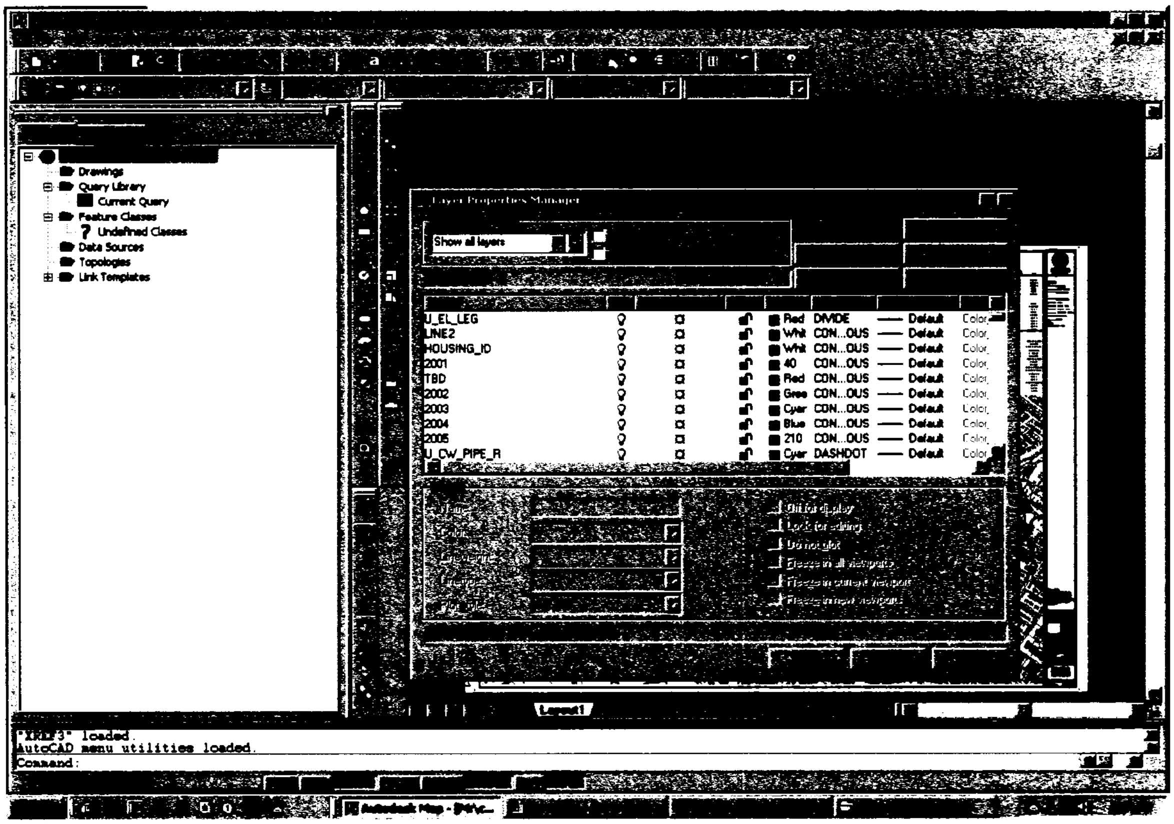


Figure 2. Traditional approach with AutoCad 2000.

There was an obvious need to integrate this into a GIS and establish a scheduled information release. The first task at hand was to develop a map once every 4 months that depicted upcoming events known at the time of data gathering. A project was created in ESRI's ArcView 3.2 for that quarter and the map published. For the next quarter another snapshot view with the attribute data attached to the shapefiles was set up. Some of the attributes the map was themed on included the department funding the project and the type of activity i.e., laydown area versus actual construction site. After the map was developed and distributed, this snapshot ArcView project was archived.

The next release brought along another project with most of the drafted geometry copied over from the earlier archived dataset (Figure 3).

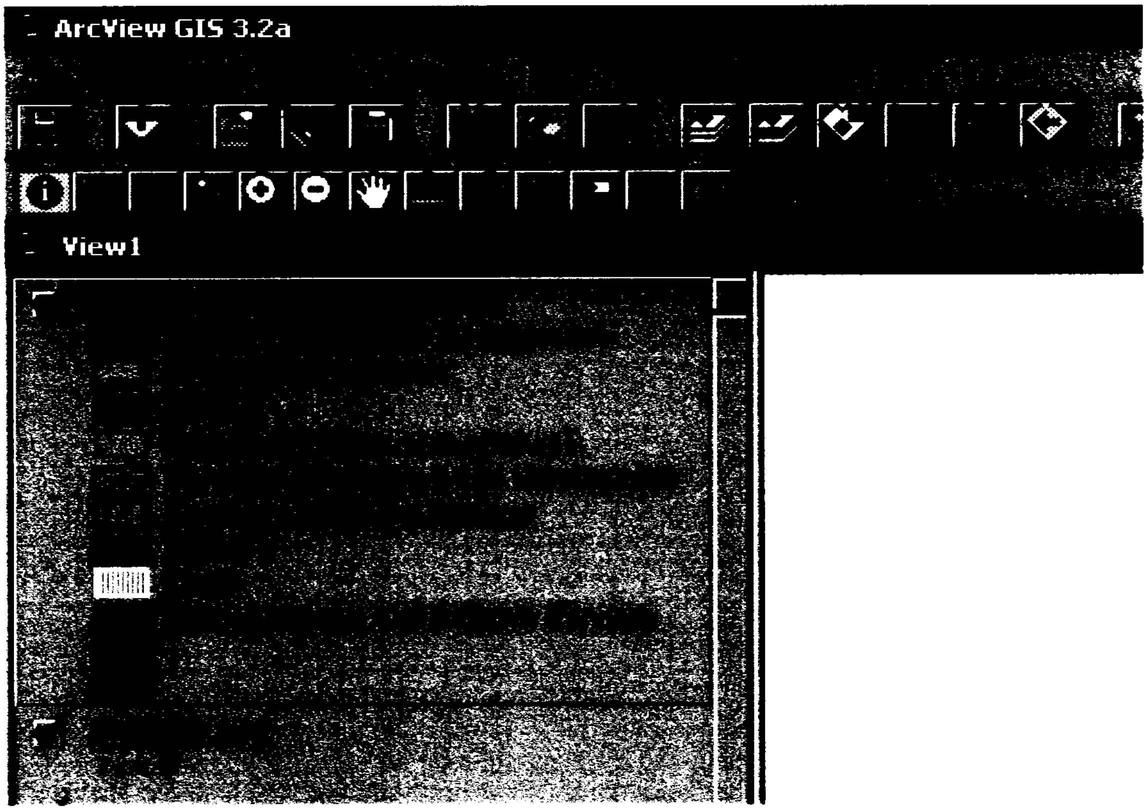


Figure 3. Traditional snapshot approach in a GIS with ArcView 3.2.

The obvious drawback is evident in the redundancy of the data. Many of the construction activities that lasted over several quarters had to be recreated in every release. In spite of this drawback, the value of a graphical representation of this spatio-temporal dataset was evident and an additional requirement to map the laydown and contractor parking areas was developed. This was simply setup as another theme of laydown areas.

Another requirement that developed in during this period was a need for archival data especially in terms of the construction projects. This information is required for deferred maintenance reporting and planning. This further necessitated the association of all geometries related to the process into a single shapefile – past, present and future.

The first step moving towards a more intelligent system was to combine the various temporal layers into one shape file and to move the attribute information into an external DBF file. There was sufficient evidence that the database would drive the system for this set of requirement. In addition, there were separate layers for lay downs, contractor parking and construction projects, all of them with one geometry type - polygons. The logical step was to combine these three datasets into one shapefile. The shapefile was still archived at every iteration, but some redundancy was eliminated. The external file served as a cumulative list at least for the year.

Around this time ESRI's ArcGIS 8.0 was introduced and the project was migrated over with the attribute data being tracked in an external database and the shapefile carrying an intelligent ID to link to the attribute table. The focus now shifted to a database driven model for the attributes. Attributes became part of the database and the shapefile was a continuing accumulation of all projects (Figure 4).

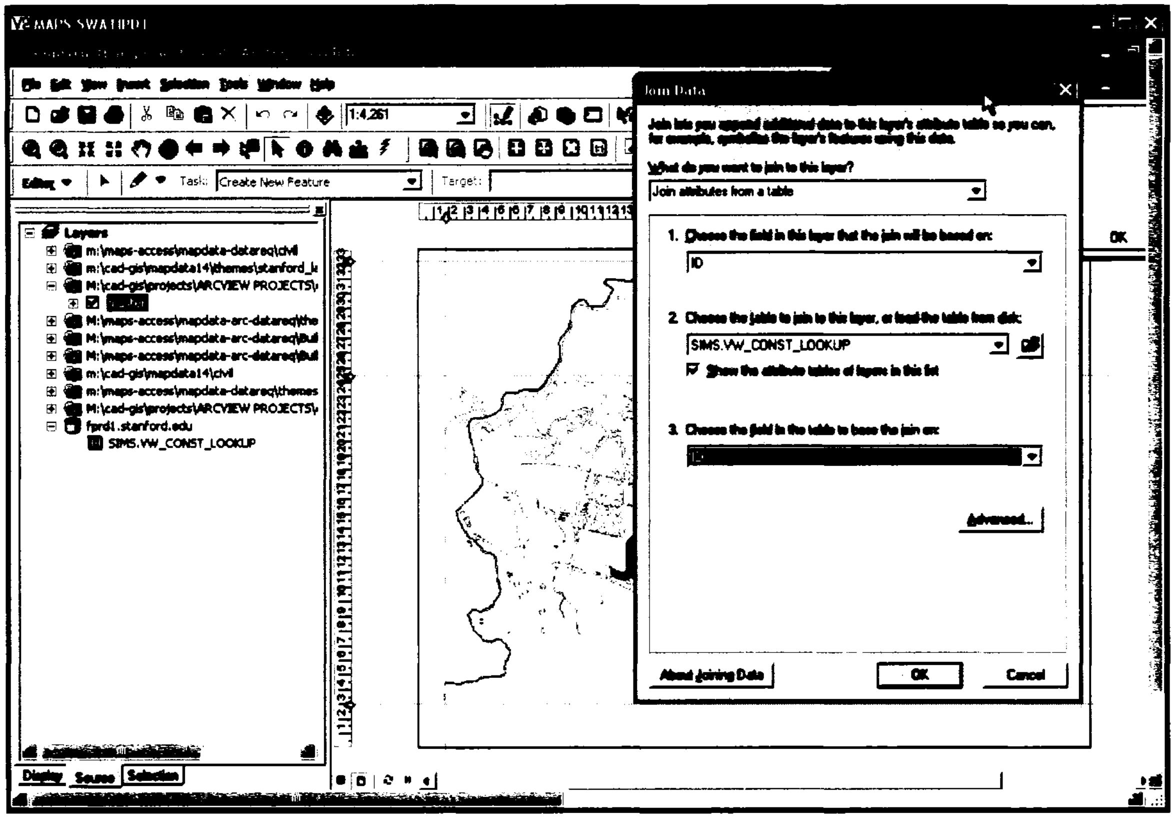


Figure 4. A pseudo database-centric solution with ArcMap 8.0 to track the shapefile and Oracle 9.0 for attributes and temporal elements.

To further interact with the temporal element of the dataset, a SQL view was established in the database. This view reflected all data form the current time to 6 months ahead.

SQL Statement:

```
SELECT * FROM current_construction
WHERE (( CONST_START <= SYSDATE
   and  CONST_END >= SYSDATE+180 )
OR (CONST_START >= SYSDATE
   and CONST_START  <=  SYSDATE+180)
```

```
    OR (CONST_END >= SYSDATE
   and CONST_END <= SYSDATE+180 )
OR  CONST_START  IN ( SYSDATE)
OR CONST_END IN (SYSDATE+180)));
```

The need for this granularity can be debated, it was chosen for convenience but any granularity can be chosen. The main purpose of the view is to narrow down projects to a relevant timeframe.

The map allows a direct ODBC link to the view and the graphical component was now reduced to a shapefile with polygons having a unique ID. Quarterly updates now involved adding new projects to the shapefile and updating the attributes in the database. A quarterly meeting with all the department managers was setup to gather the information pertaining to the physical boundaries and attribute data. However information not conveyed during these meetings or unexpected developments that were not recorded led to data discrepancy. One could not completely rely on this GIS system for accurate information due to lack of resources to keep the system current.

Similarly, in terms of communicating the information, the granularity chosen was not suitable for every need. Soon after, it was obvious that no single granularity would meet the needs of all users and the system needed to be more dynamic. The granularity was dependant on the user and the ideal system would be more interactive. Furthermore, although the map was excellent for getting information about the immediate future, there was a need to query both the past for reference information and the future for planning and logistical purposes.

The data model lends itself to meet the requirement as all the information was in the Oracle table and appropriate queries could derive the desired information. The graphical representation aspect of dynamic display was more challenging. The solution had now evolved from a snapshot approach to a pseudo - database centric technology.

The next step in this system development was to provide for dynamic querying and displaying of the boundaries for various users. A web based interactive solution was chosen for this aspect of the project. Autodesk's Mapguide product allows for seamless interaction between the graphical elements and Oracle database using technologies like Cold Fusion and Javascript.

The web application allows the user to choose a range of dates to narrow down the scope of their query. The SQL statement consists of two variables to define the start and end dates of the request and the cold fusion application runs this query against the Oracle database to return the Unique Ids of the projects that meet the criteria. This Id along with all the associated attribute information from the database can be formatted for display on the web based report.

SQL Statement:

```
SELECT * FROM current_construction
WHERE(
((CONST_START <= to_date('#STARTDATE#.#STARTYEAR#', 'MM YYYY'))
and (CONST_END >= to_date('#ENDDATE#.#ENDYEAR#', 'MM YYYY')) )
OR( (CONST_START >= to_date('#STARTDATE#.#STARTYEAR#', 'MM
YYYY'))
```

and (CONST_START <= to_date('#ENDDATE#.#ENDYEAR#', 'MM YYYY')))

OR( (CONST_END >= to_date('#STARTDATE#.#STARTYEAR#', 'MM YYYY'))

and (CONST_END <= to_date('#ENDDATE#.#ENDYEAR#', 'MM YYYY')))

OR CONST_START IN (TO_DATE('#STARTDATE#.#STARTYEAR#','MM YYYY'))

OR CONST_END IN (TO_DATE('#ENDDATE#.#ENDYEAR#','MM YYYY'))

Graphically displaying the selected projects only within the Mapguide viewer was more challenging. A temporary layer that highlighted geometries on the map based on selected criteria was developed (Figure 5).

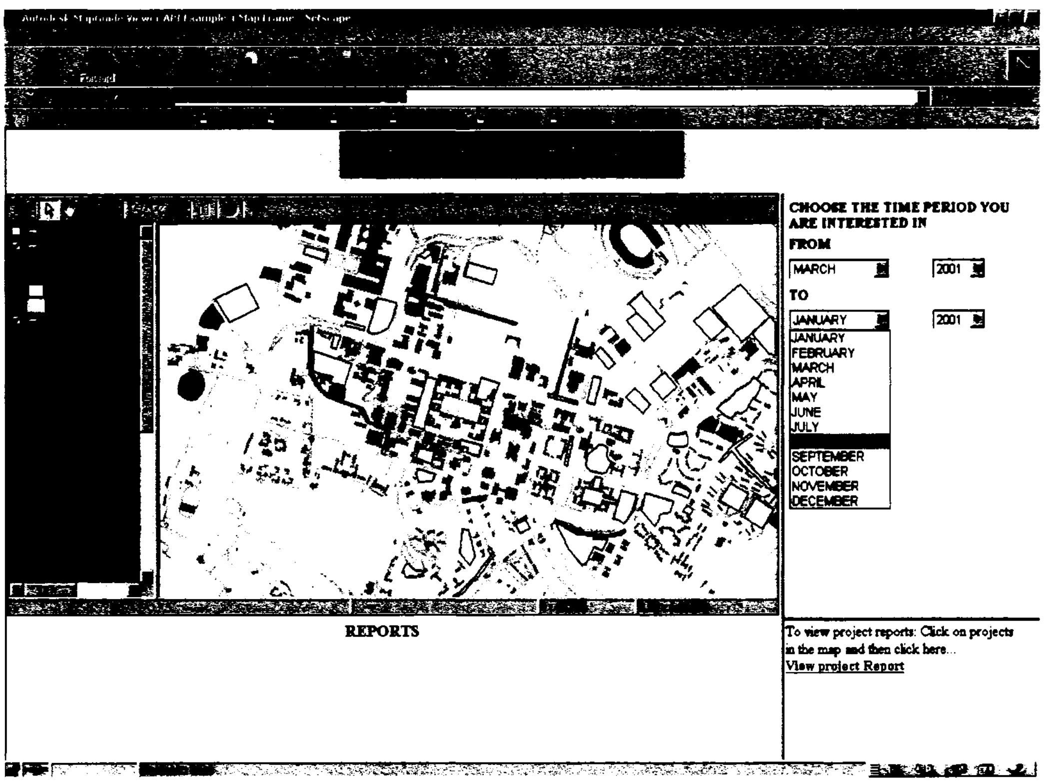


Figure 5. Web based solution using Autodesk Mapguide 6.0 for display/analysis.

Now that the challenge of dynamic user interaction of the information was met, the next step was to make the data entry and updating more dynamic as well. Information queried from the system is only as good as the data available. The requirement was for an easy to use application that allows project managers or representatives to easily draft project extents and update the attribute information.

An application was developed using ArcPad to allow the user to either provide updates in the field or back at the desk on a tablet PC. The updates were uploaded to the server when the tablet PC was synchronized in the form of a shapefile with the attributes. After a quick overview by the GIS staff of this updated shapefile and few error checks to ensure closed polygons, the shapes file was posted on the web server. The attribute information was then uploaded to the Oracle database with SQL loader. However, every change was a new record in the database and a given project could result in multiple records with different modify dates. For this application, only the most recent change was critically desired in the reporting and when the project information did change and it was archived, audit trails would suffice. An interim solution was to create another Oracle table to store these proposed changes. A trigger would have to compare the two tables and update the master table.

The interactive data updates and queries allowed for a relatively smaller temporal granularity and the temporal discrepancy now was between the data update and the synchronization of the servers. Ideally the data would be updated directly by the user and the need for the interim data loads eliminated. Further, the need for on site updating was questioned for this particular application, as drafting accuracy was not critical. The construction zones were simply a depiction of the region of activities and not measured representations.

ArcGIS and the versioning approach seemed to be another alternate direction but as outlined in the literature review, it would not address this particular temporal issue. Like the snapshot model, versions also require specific choices for establishing the version granularity. This raises the same concerns as the snapshot approach, about determining this specific choice given the varying levels of granularity. It also does not address elements that span across versions. Hence, this specific software solution was not reviewed in this case study.

The developments made it clear that we were indeed moving towards a database-centric approach, where both spatial and attribute information could be stored in a central repository. Relative popularity and acceptance by various users, availability of the software and research about potential vendors further suggested that a prototype for this approach could be attempted with Oracle's Spatial module. The introduction of Oracle Spatial technology seemed to be an ideal environment for moving ahead with the prototype. The polygons were converted to Oracle's Spatial objects and the attributes associated with these objects. The need for shapefiles or dwgs is eliminated and updating as well as reporting technologies can tap into the spatial information. Further, Autodesk's Mapguide 6.5 supports web based drafting and updating of Oracle Spatial objects. ESRI's ArcIMS also offers similar functionality but for the purpose of this study, Autodesk's Mapguide technology was employed (Figure 6).

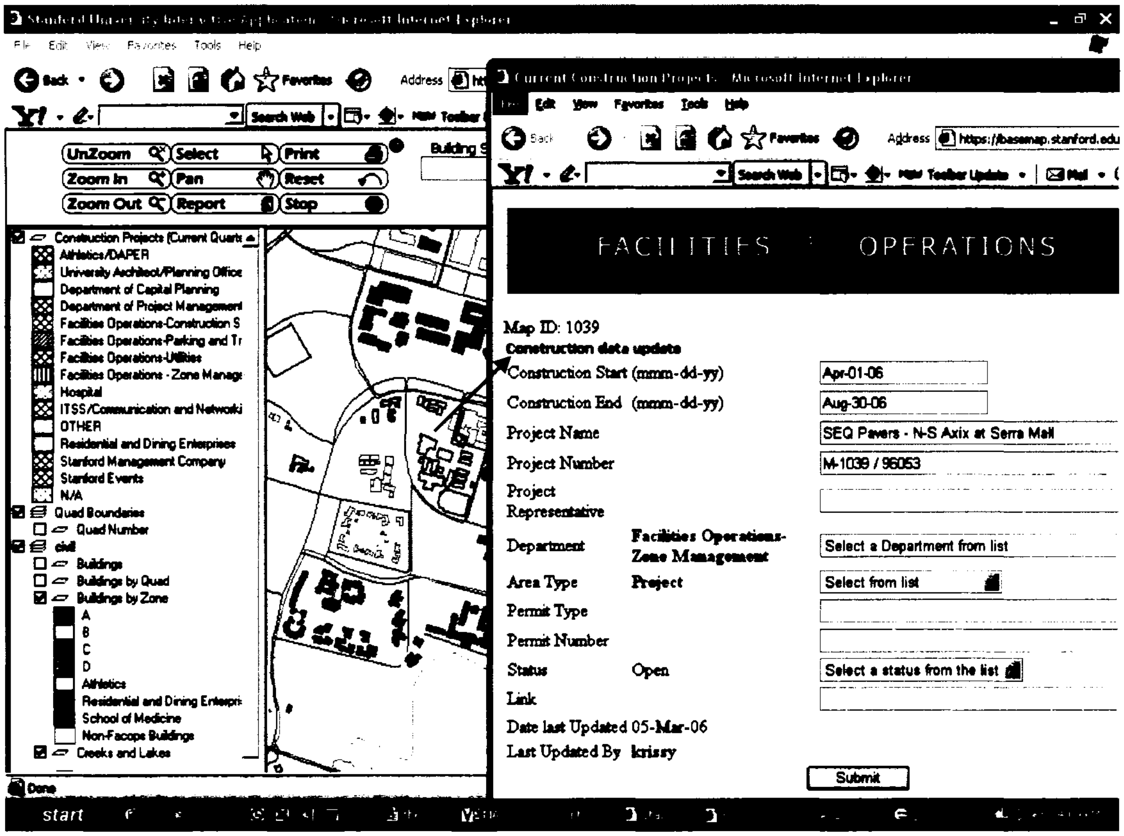


Figure 6. Web-based solution for digitizing extents and updating attributes using Mapguide 6.5 and Oracle Spatial 10g.

This further supports the development of an Oracle Spatial data model for the construction areas and the associated attribute information. Project managers can access the secure website and make updates directly to both the spatial component and associated attribute data. Information is now truly real time and dynamic, while updates can be queried through the reporting interface immediately.

Updates and modification made by the project managers can be retrieved from the audit trails if needed. However, the basic premise is that they can make changes directly to the system without a need for any routing and approval or the need for previously forecasted attributes. This flexibility in requirement for this particular category makes it easier to arrive at a possible resolution to the problem at hand.

The data being tracked in Oracle Spatial does not fully eliminate redundancy although it makes this particular problem manageable. The resulting solution offered in this case study addresses most of the spatio-temporal requirements defined. For the purposes of this study the database centric concept of tracking spatio-temporal data proved to be the best fit.

# CHAPTER V

# FINDINGS AND CONCLUSION

The process for developing this system for the specific temporal issue of construction activities and situation revealed the ongoing research and development in working with temporal data. The study mainly underlines the lack of sufficient technological advancements to support a spatio-temporal data model in the real world. There has been a great amount of research and theoretical studies on this subject, but this momentum has not been carried into the practical aspects of real world implementation.

One of the major questions this study asked, in detailing the various instances of spatio-temporal datasets, was if there could be inherent similarities between the various issues listed below. If seemingly different issues can be grouped together based on temporal granularity and/or spatial configuration, could all of them be resolved by a single methodology? At the onset, it would seem that most spatio-temporal problems could be addressed with a generalized system. However after pursuing the detailed case study and reviewing the issues below, the conclusion is rather obvious. The approach of a database centric solution worked very well for the case study and the issues it addressed but cannot be extrapolated to being suitable for all spatio-temporal issues.

Table 3 revisits the categories and analyzes the solution from the case study as applicable to each instance.

Table 3. Categories of Spatio-temporal issues and conclusions.

| Category | Does solution from case study address this spatio-temporal issue? |
|---|---|
| Construction projects and laydown areas. | Yes. Spatial aspect of the data has a greater tolerance for accuracy. Temporal granularity is flexible and data maintenance is distributed allowing for automation. |
| Geometric configuration changes in large indoor spaces. | No. Geometry of the data is not flexible and requires accuracy of measurements. Transactional and incident granularity can be flexible |
| Rooms/spaces may have various functions over a given period | Perhaps. Geometry of the data is not extensively dynamic. Temporal granularity is quite large and data maintenance can be distributed and automated. |
| Parcels in a given area merge into one APN number for a time period to be split up again later. | No. Geometry of the data is not flexible and requires accuracy of measurements. Temporal granularity cannot be well defined but real time data critical. Transactional and incident granularity is not well defined. |
| In Emergency management scenarios, a hazard (fire, flood, bio hazardous fumes) that spread in various directions. | Perhaps. Generation of geometry data can be automated with GPS/ imagery solutions Temporal granularity cannot be well defined but real time data is critical. Database design very challenging with huge system resource overheads |
| Vehicular tracking in Transportation management field. | Perhaps. Geometry of data can be automated with GPS/imagery solutions. Temporal granularity very fine and well defined. Real time data is critical. Database design is very challenging with huge system resource overheads. |

Another question this study posed at the beginning attempted to identify existing software solutions in the field of GIS that were suitable for spatio-temporal issues. The research and case study that ensued underlined the growing need for software development in this field. The complexities of this problem and the lack of potential solutions available today were further spotlighted by this research. Fortunately, there is a growing awareness about the need to accommodate spatio-temporal issues in the software industry.

There are three broad categories of software solutions offered today. Both the snapshot and versioning approaches present two potential problems. Firstly, there is the inflexibility for most temporal datasets to decide on the granularity. Too fine a granularity increases system resources and redundancy, and too big the temporal range causes many temporal elements to be lost. Secondly, there exists the issue of analyzing problems that span across two data chunks and fall between two temporal "images". The spatial database approach seems to be a promising direction for these problems. The database approach allows one to retrieve all features that existed during a given time span, even if they existed in different forms. The challenge is in the design of the database and key fields. Also system resources for large datasets and the application's ability to handle the returned dataset pose some hurdles.

The issue of spatio temporal systems is further complicated by the vast differences between what appears to be similar problems. This research attempted to pursue one category of the scenarios identified in the case study to draw relevant conclusions that perhaps could be applied across the various issues. The process of progressing through the various practical aspects of implementation with available technologies provided valuable insight.

The other question this study asked was whether temporal granularity had any bearing on the design of the potential solution. The study showed that the pattern of analysis for seemingly similar temporal problems could be varied. In the case of the requirements for this case study, some of them turned out to be quite flexible. The temporal granularity could be fairly adaptable based on the resources available. Furthermore, although the information was of a temporal nature the past updates to a record did not need tracking. The modification could be directly posted to the database.

The temporal granularity of the dataset in question dictates to a large extent the methodology adopted for the design and implementation. The finer the granularity the more challenging it is to implement a solution, perhaps making it unfeasible. Flexibility in the granularity and criticality of data lost between time periods further contributed to the challenge for some issues. A well-defined time interval is critical to the successful analysis of time-based data (Peuquet, 1994). There are instances when the granularity cannot be well defined for a spatio temporal problem that might lead to unnecessarily choosing the finest granularity.

The minimum requirements that are required of a spatio-temporal dataset dictate the design and implementation of the application. Although any GIS problem should have well defined precise requirements, the complexity of spatio-temporal data further underscores its importance. Given the heavy load on resources and the data unwieldiness of highly ambitious endeavors for some of the categories, consideration to redefining and simplifying requirements is paramount.

## REFERENCES


Bertin, J. (1983). *Graphics and graphic information processing.* Berlin: de Gruyter.

Clifford, J., & Tuzhilin, A. (1995). Recent advances of temporal databases. *Workshops in Computing Series,* 194-213. Berlin: Springer-Verlag.

Darwen, H., & Date, C.J. (1998). *Foundation for object relational databases: The third manifesto.* MA: Addison-Wesley.

Date, C.J. (2000). *An introduction to database systems.* MA: Addison-Wesley.

Egenhofer, M.J. (1992). Why not SQL!. *International Journal of Geographic Information Systems, 6(2),* 71-85.

Egenhofer, M.J., & Golledge, R.J. (1998). *Spatial and temporal reasoning in Geographic Information Systems.* New York: Oxford University Press.

Ewing, J.(2001). *Using ArcSDE Versions in the Real World.* Retrieved January, 2006, from http://gis.esri.com/library/userconf/proc01/professional/abstracts/a232.html

Hazelton, N.W.J., & Williamson, I.P. (1990). On the design of a temporally referenced 3D Geographical Information Systems; *Proceedings of GIS/LIS Annual Conference, Anaheim, CA, 1,* 357-372.

Hazelton, N.W.J. (1991). *Integrating time, dynamic modeling and Geographic Information Systems: Development of four dimensional GIS.* Unpublished masters thesis, Department of Surveying and land information, University of Melbourne, Australia.

Ingram, K., & Phillips, W. (1987). Geographic information processing using a SQL-based query language. *Proceedings of the American Society for Photogrammetry and Remote Sensing, Baltimore, MD, 8,* 326-335.

Jensen, C.S., & Snodgrass, R. (1999). Temporal data management. *IEEE Transactions on Knowledge and Data Engineering, 11(1),* 36-44.

Kelmelis, J. (1991). *Time and space in geographic information: Toward a four-dimensional spatio-temporal data model.* Unpublished doctoral dissertation, Pennsylvania State University, University Park, PA.

Langran, G., & Chrisman, N.R. (1988). A framework for temporal geographic information. *Cartographica. 25(3),* 1-14.

Langran, G. (1995). A review of temporal database research and its use in GIS applications. *International Journal of Geographic Information Systems, 3(3),* 215.

MacEachren, A.M. (1995). *How maps work: Representation, visualization and design.* New York: Guilford Press.

Maguire, D., Stickler G., & Browning G. (1992). Handling complex objects in geo-relational GIS. *Proceeeding at the Fifth International Symposium on Spatial Data Handling,(2),* 652-661.

McCormick, B.H., Defanti, T.A, & Brown, M.D. (1987). Visualization in scientific computing. *IEEE Computer Graphics and Applications, 7(7),* 61-70.

Morrison, J.L. (1974). A theoretical framework for cartographic generalization with the emphasis on the process of symbolization. *International Yearbook of Cartography. (14),* 115-27.

Ott, T., Swiaczny, F. (2001). *Time integrative Geographic Information Systems, Management and analysis of spatio-temporal data.* Berlin: Springer-Verlag.

Peterson, M. (1994). Spatial visualization through cartographic animation: theory and practice. *Proceedings of the GIS/LIS Annual Conference, St.Paul, MN.* 619-628.

Peuquet, D. J. (1990). *A conceptual framework and comparison of spatial data models. Introductory readings in GIS.* London: Taylor and Francis.

Peuquet, D. J. (1994). It's about time: a conceptual framework for the representation of temporal dynamics in geographic information systems. *Annals of the American Association of Geographers, 84,* 441-461.

Peuquet, D.J., & Qain, L. (1996). An integrated database design for temporal GIS. *Seventh International Symposium on Spatial Data Handling. Delft: Delft University of Technology, I(2),* 1-11.

Peuquet, D.J. (2001). Making space for time: Issues in space time data representation. *Geoinformatica, 5(1),* 11-32.

Peuquet, D.J. (2002). *Representations of space and time.* New York: The Guilford Press.

Raper, J. (2000). *Multidimensional Geographic Information Science.* New York: Taylor and Francis.

Shapiro, L.G., & Harlick R. (1982). An experimental relational database system for cartographic applications. *Proceeding at the American Society for Photogrammetry and Remote Sensing/ American Congress of Surveying and Mapping Conference, Bethesda, MD.*

Stonebraker, M., Rowe, L.A., Lindsay, B.G., Gray, J., Carey, M.J., & Beech, D. (1990). Third-generation database system manifesto. *ACM SIGMOD Record,19(3)*, 31-44.

van Roessel, J.W. (1987). Design of a spatial data structure using the relational normal forms. *International Journal of Geographic Information Systems. 1(1),* 33-50.

Wachowiz, M. (1999). *Object oriented design for temporal GIS.* London: Taylor and Francis.

Waugh, T.C., & Healey, R.G. (1987). The geoview design: A relational database approach to geographical data handling. *International Journal of Geographic Information Systems, 1(2),* 101-118.

Yuan, M., (1994), Wildfire conceptual modeling for building GIS space-time models. *Proceedings of the GIS/LIS Annual Conference, Phoenix, Arizona ,* 860-869.

Zeiler, M. (1999). *Modeling our world.* Redlands: ESRI publications.

Zhao, F. (2002). *Transportation application of temporal GIS.* White Paper, Department of Civil and Environmental Engineering Florida International University.