A Dissertation On
"A Study Of Spatio-Temporal Database Management Systems"

For The Award Of The Degree
Master Of Philosophy
In Information Technology

By

Ms. Shilpa M. Pund

Under The Guidance Of
Dr. M.S.Prasad

Bharti Vidyapeeth University
Institute Of Management And Enterpreneurship Development Pune-411038
2007-08

Accreditated By NAAC With "A" Grade

BHARATI VIDYAPEETH UNIVERSITY, PUNE

INSTITUTE OF MANAGEMENT AND

ENTREPRENEURSHIP DEVELOPMENT PAUD ROAD,

ERANDAVANE, PUNE- 411038

CERTIFICATE

Certified that

*MR/MRS/MSS _ **Shilpa M. Pund** is a bonafiide student, pursuing Degree of Master of Philosophy program of the Bharti Vidyapeeth University at the Institute of Management and Entrepreneurship Development, Pune for the academic year 2007-2008.*

The Dissertation titled " A Study of Spatio-Temporal Database Management Systems " is a bonafied work, the student has carried out under the guidance of Dr. M.S. Prasad.

(Teacher Guide)	**(Course - Coordinator)**	**(Director)**

Date: 24/4/2008.

ACKNOLEDGEMENT

*I would like to express my gratitude towards my **college Modern College Of Arts , Science and Commerce,Pune-5** where I am working as a Lecturer in Computer Science department for allowing me to persue valuable degree of M.Phil.*

*I would also like to express sincere thanks towards **Principal Mr.A.G.Gosavi ,Mr.S.S.Deshmukh** ,Head, Department of Computer Science and my collogues who always helped me through out my study.*

*I would like to express my gratitude to **Bharti Vidyapeeth's Institute of Management and Entrepreneurship Development** for allowing me to do M.Phil and submit this dissertation*

*I would like to express my special thanks to my guide **Dr. M.S. Prasad** for his endless support and right guidance.*

*I would like to express my thanks to **coordinator Dr. Abhyankar and Dr. M.V. Shitole** who always help me and encourage me.*

*I would like to express my special thanks to my **parents**, my **husband** and my 1 year old son **master Shlok** , for their coordination.*

Thank you,

DECLARATION

I have carried out this work. This is my own and has not been carried out by any other candidate.

Place : PUNE

Date: 24/4/08

Shilpa

Signature

Ms. Shilpa M Pund

"A Study of Spatio-Temporal Database Management Systems"

Index

Chapters Page No.

1. A Conventional Database Management System

A database management system (DBMS) is a system software used to manage the organization, storage, access, security and integrity of data in a structured database. The nature of database management systems has dramatically since the 1960 as the demand for data storage have increased and the technology to store data has plummeted.

A database can be a set of flat files stored on computer tape or disk or it could consist of database tables that are managed by a Database Management System (DBMS). There are different types of DBMS products: 1) relational, network and hierarchical.

The most widely commonly used type of DBMS today is the Relational Database Management Systems (RDBMS). Some Data Base Management Systems can be accessed directly using programming languages such as COBOL while others provide their own programming language for interacting with the database. Many DBMS applications also provide reporting and query tools to view data

In the early days of computing, disk storage was extremely expensive. Most application systems ran in batch mode using data that was stored on magnetic tape. Data had to be read sequentially from flat files. For performance reasons, the management of data was tightly integrated with the application system.

As the cost of disk storage fell, opportunities to store data for real-time access arise. Specialized DBMS software emerged during the 1960s for the sole purpose of managing data. Application systems were then able to focus on the user interface, screen navigation, data validations etc. and could leave the data management tasks to the specialized DBMS technology. The application system simply had to call the DBMS when it needed to read or store data.

The early DBMS systems required structured data which is easy to store and/or access. Data was stored in database records that were linked to related data via "pointers". (i.e. hierarchical and network databases). Although speed of access was good, flexible access to data was not.

As the cost of data storage fell, it became feasible to store data in tables. This eliminated much data redundancy and provided much more flexible data access. Most DBMS databases today are relational.

One very important role of a database management system is to maintain the data definitions for each table and columns in the database. Each piece of data must be assigned a name, a data type (e.g. date, alphanumeric, numeric) and a mandatory/optional status. Most DBMS packages will enforce these rules when rows are added to the database. DBMS packages can also enforce domain rules. For example, domains for marital status, state codes or country codes could be defined to ensure that only valid values are stored.

Most DBMS software will also enforce cardinality and optionality rules that govern relationships between things of interest (i.e. tables). Examples of such rules are as follows:

1. Each customer ***may*** have ***zero, one or many*** orders.
2. One and only one customer must initiate an order.
3. An order ***must have one or many*** order items.

When rows are deleted from the database, the DBMS will ensure that all dependent rows are also deleted, leaving the database in a stable state. For example, if an order is deleted, the DBMS would ensure that all related order items are also deleted. Enforcing referential integrity rules is an important role of database management systems.

Another important role of a Database Management System is to enforce data security. Based on the assigned roles of users, a DBMS system can ensure that a given user only has read and/or update access to appropriate columns in the database. This ensures that private data such as salary information is only accessible to the appropriate parties.

Data access can be restricted via database "views" that filter out sensitive data and by other means. Many DBMS applications can track changes made to tables in the database. Along with the prior version of the data, the DBMS will record the identity of the person who altered the data.

Maintaining audit trails on important data is another important role for database management systems. Following are some database management Systems

1) **DB2** is a relational database management system that was introduced by IBM in 1983. The product was initially developed to run on the MVS operating system. A distributed version of DB2 was then released on OS/2 in 1987. Today DB2 is widely used by large organizations and can run on wide variety of UNIX, Linux and Windows platforms.

IBM markets different "editions" DB2 with different licensing arrangements. The cost of ownership therefore is based on the options that are needed. DB2 Editions include the following:

1. DB2Workgroup
2. DB2 Workgroup Unlimited
3. DB2 Enterprise Server Edition
4. DB2 UDB Data Warehouse Enterprise Edition (DWE)

The DB2 UDB Data Warehouse Enterprise Edition includes business intelligence capabilities such as ET, data mining and online analytics.

Microsoft Access 2003 is a low cost relational database management system (RDBMS) used mainly by home users and small businesses with one or two system users. It can run on a personal computer or server and provides an easy-to-use interface for designing simple databases, reports and data entry forms.

The current version of MS Access allows you to save, read, and interchange data in XML format. The DBMS also has good ODBC capabilities (for connecting to other data sources) and can grab data from external databases and spreadsheets. Microsoft provides a wizard to replicate an MS Access database to Microsoft SQL Server.

MySQL is a popular database management system that is used in some six million installations. It is widely used for web applications, especially in concert with PHP.The MySQL DBMS is owned and sponsored by the Swedish company MySQL AB. The company sells support and service contracts, as well as commercial licenses.

Oracle Corporation launched its first commercial SQL relational database management systems in 1979. The DBMS runs on Unix, Linux and Windows platforms and is widely used around the world.

Oracle is the world's largest enterprise software company and sells many applications system packages that run on its Oracle database. In late 2005, the company purchased PeopleSoft and rights to its CRM, Purchasing, Financials, Human Resource, Supply Chain Management software packages.

Commercial database management systems (DBMS) such as Oracle, Sybase, Informix and O2 allow the storage of huge amounts of data. This data is usually considered to be valid now. Past or future data is not stored. Past data refers to data, which was stored in the database at an earlier time instant, and which might has been modified or deleted in the meantime. Past data usually is overwritten with new (updated) data. Future data refers to data, which is considered to be valid at a future time instant (but not now).

A DBMS stores the data in a well-defined format. A relational DBMS, for example, stores data in tables (also called relations). Thus, a relational database actually contains a set of tables. Each table contains rows (tuples) and columns (attributes). A row contains data about a specific entity, for example, an employee. Each column specifies a certain property of these entities, for example, the employee's name, salary etc. The following table stores data about employees:

EmpID	Name	Department	Salary
10	John	Sales	12000
12	George	Research	10500
13	Ringo	Sales	15500

Object-oriented DBMS store data about entities in objects. So each employee is actually an object. The type of an object specifies the properties the object has. An employee object thus has properties such as a name, a salary etc. Sets of objects of the same type are called collections. Thus - in an object-oriented DBMS - a database contains a set of collections.

2. Limitations of Conventional DBMS and need for Spatio-Temporal DBMS

The existing relational DBMS (RDBMS) technology has been successfully applied to many application domains. RDBMS technology has proved to be an effective solution for data management requirements in large and small organizations, and today this technology forms a key component of most information systems. However, the advances in computer hardware, and the emergence of new application requirements such as multimedia and mobile databases, GIS, large size databases produced a situation where the basic underlying principles of data management need to be re-evaluated. Some technology observers see RDBMS technology as obsolete in the context of today's application requirements and advocate a shift towards Object-Oriented (OO) databases. While complex data types play an important role in many applications, they represent only a subset of the problems which database technology needs to address. I attempt to provide a balanced view on new trends in database technology. Here the requirement for high-volume databases and their application areas, which are being successfully addressed by extending relational database technology, are discuss.

For most of the history of databases, the type of data stored in a database is relatively simple and hence there is a limited support for the data types in earlier versions of SQL.

In past few years, there has been increasing need for handling new data types in databases, such as temporal data, spatial data , multimedia data and so on. Hence there is a through study of Spatio-Temporal Database Management System.

Miniaturization of computing devices, advances in wireless communication, and development of positioning systems created a wide range of database applications such as location-based services, vehicle tracking and air polution movement that has to deal with Moving Objects (MO). Database Management System (DBMS) technology provides a potential foundation upon which to develop these applications, however, DBMS's are currently not tuned for this purpose. The reason is that there is a critical set of capabilities that are needed by moving objects database applications that are not supported in existing DBMS's.

This dissertation presents behavioral aspect of the spatio-temporal databases, for managing and querying moving objects. The objective is to evaluate and develop a framework for managing and querying moving object database. A commercial software package has then been selected to design and develop the capabilities required for moving object's querries. In order to examine the developed system, the querries poses by fleet management companies are selected. The results have been evaluated and showed the superior functionalities of the system.

3 . Current Database Technology Trends

Commercially available database technology supports the requirements of most business-style applications. Such applications use structured data, i.e information that can be represented as records with pre-defined standard data types (e.g. character, number, date, etc.). Business applications are characterized by well-defined (short) transactions in which individual users update simple records; typically, a single row in a database table (e.g. debit/credit banking transactions). While the management of structured data remains important other types of information including image data, audio, video, and text are used increasingly in applications today.

New Application Requirements

A number of new application areas have emerged recently which require data management support that extends well beyond the traditional data management functions. Also importantly, advances in computer hardware, network and user interface technology have re-defined the context in which data management needs to performed. In this section we discuss three new database application areas: complex object databases, high-volume databases, and mobile databases.

Complex Object Database Applications

Application such as CAD/CAM (Computer Aided Design/Computer Aided Manufacturing), document management, and multimedia, are forcing the developers of DBMS technology to re-assess the basic underlying database principles and produce new technical solutions. The distinguishing features of these applications are complex objects and unstructured data, which are difficult to represent as tables in a relational system. Applications that use complex objects often involve complicated, long-duration transactions that can take several days or even weeks to complete. For example, transactions encountered during design activities tend to involve co-operation between several users and cannot be served effectively by traditional transaction mechanism, which relies on locking and transaction rollbacks to resolve contentions. Many database researchers regard the relational model as unsuitable for applications of this type and advocate the use of object-oriented technology. OO technology has been applied successfully to programming and user interface development, but its impact on database is still rather unclear. Several pure OODBMS (Object-Oriented DBMS) products are available commercially (e.g. Versant, O2, Object Store), but they remain focused on specialized applications and have not achieved wide market acceptance. Some research problems and many practical issues remain to be resolved before OODBMS technology can be applied to large-scale, mission-critical applications. Hybrid solutions which support both structured and unstructured information using relational and OO technologies are being developed for commercial use by database vendors. For example, Oracle Media Server combines Oracle database server technology with text and multimedia servers.

High-Volume Database Applications

Another active area of database research and development concerns applications which use very large volumes of data. Examples of such applications include retail chain applications which store every cashier transaction in a historical database and perform ad-hoc analysis to determine customer buying patterns, and the popularity of individual products. Traditionally, the main factor limiting database performance was the speeds of disk read and write operations (I/O throughput). Advanced cashing techniques used in modern database server systems have resolved the I/O bottleneck and produced CPU-bound systems. An obvious solution to the CPU bottleneck is to use multiple-processor systems configured as either SMP (symmetric multiprocessor) or MPP (massively parallel processor) systems. Parallel computing architectures have been used in scientific computing for some time, and they are now beginning to impact on database applications. Proprietary database machines (e.g. Terradata, Tandem, etc.) built using specialized hardware and software were the first parallel database technologies on the market. The use of multiprocessor machines for database applications is becoming an attractive solution to performance problems associated with high-volume databases as the cost of multiprocessor hardware falls. MPP architectures present a particularly cost-effective solution as MPP systems are constructed using commodity processors (i.e. Intel 486 and 586) and offer excellent price-performance ratios and scallability. Several complex technical problems need to be overcome to enable database server systems to achieve good scallability on MPP architectures. To start with, MPP systems are based on shared-nothing architectures; each processor has its own RAM (Random Access Memory) and communicates with other processor via an interconnection network. Consequently conventional concurrency techniques, which rely on shared memory, do not work in this message-based environment. Also, to take full advantage of MPP architectures all database operations need to run in parallel. This applies in particular to query and update operations, but also to table and index creation, database loads and backups. Interestingly, the SQL language, because of its non-procedural nature and theoretical foundation in relational algebra, lends itself to parallelizations. SQL queries can be expressed in functional form and operations such as aggregation, sorts, joins, and table scans can be run simultaneously on multiple processors. Production versions of database server technology are available today which perform at least some database operations in parallel .The impact of parallel computing on database processing is likely to be highly significant creating new opportunities for applications which cannot be accommodated with single processor architectures.

Mobile Databases

With the emergence of mobile computing the corporate information system also includes notebook computers and other portable devices. Similar to desktop computing users, mobile users need access to information stored on corporate database servers. In some cases, mobile users carry fragments of the corporate database with them in notebook databases and from time to time need to synchronize their information with the information held on the remote server. Mobile computing can be regarded as a special type of distributed environment and presents a number of technical problems for database implementation. Firstly, mobile computing is characterized by relatively slow communications based on wireless networks (2 - 9 kbps) or modems using phone lines (up to 28.8 kbps). Mobile communications are non-continuous and generally unreliable causing poor system availability. Synchronous client/server computing techniques which need fast and reliable networks are not suitable for communications between the server and the client application running on a mobile workstation. Asynchronous processing using store and forward techniques are best suited to this type of environment. During periods when the mobile station is disconnected, messages are stored in a queue and transmission is resumed when communications are re-established. In situations where the mobile station also carries data, the resulting environment can be characterized as a distributed database. Asynchronous replication techniques play an important role in maintaining information stored on mobile stations. Database and communication technologies suitable for mobile operation were announced recently and will be available commercially in the near future.

These new types of applications have created a situation where the existing database server technology is no longer sufficient. Both the size and the complexity of databases have dramatically increased forcing database researchers to develop new data and transaction models. The result of this research is now beginning to impact on commercially available database server technology. In this dissertation I have briefly reviewed the achievements of database research and development of new database application area. High-volume database applications are already a reality in many organizations today and parallel database technology is beginning to be used to address the requirements of such applications. Complex object databases and mobile databases are likely to become significant areas of activity in the future.

4 .Literature Review

In order to understand the concept of Spatio-temporal databases, first it is needed to understand the basis behind the concept. "A temporal data means a data with respect to time and a spatial data means a data with respect to space and a collaboration of Space and Time with each other".

Spatio-temporal databases deal with applications where data types are characterized by both spatial and temporal semantics. Development and research in this area started decades ago, when management and manipulation of data, relating to both spatial and temporal changes, was recognized as an indispensable assignment. However, spatio-temporal data handling was not a straightforward task due to the complexity of the data structures requiring careful analysis in structuring the dimensions, together with the representation and manipulation of the data involved. Therefore, the earlier work in this area begins from separate research in both temporal [1993] and spatial [1994] databases. This effort later became the basis for spatio-temporal database models.

Since the integration of spatial and temporal database models into spatio-temporal database models, a number of new approaches have been proposed. At the same time, reviews of these works have classified and compared the existing spatio-temporal models. Currently, domain experts are trying to achieve more effective integration of the spatial and temporal aspects providing practical, unified spatio-temporal data modeling, and clarifying the direction for further research and development. Standing at this point the contribution and contemporaneously the aim of this [redacted] is to provide a complete review of existing spatio-temporal database models developed or suggested in recent decades and for the first time to critically compare and evaluate them in terms of some universal criteria, in order to identify the trend as well as the needs for further research in the area.

The attempt to classify most of the existing data models was facilitated by previous reviews in the field. One of the most significant contributors of the domain has been Gail Langran who first looked at the aspects of time in Geographic Information Systems (GIS) [1992]. Although Langran's work covers many of the most important issues of spatio-temporal systems, a number of new proposals have emerged since. Earlier review works are considered and included in order to capture all the trends and the ideas proposed in the domain of spatio-temporal database modeling. [1992] was one of the first attempts to present the poor (till then) theories and methods of reasoning in the time varying spatial space, while a bibliography on spatio-temporal databases until 1994 was published , which contains interesting pointers for further reference. More fruitful reviews of the domain were available the forthcoming years in [1996], [1997], [1999], [1999], [2001] and especially in [1998] where is introduced the classification of spatio-temporal database models used in the current survey of the area.

Definitions on Spatio-Temporal Data Modeling

Evolving research on space-time representation has focused on a number of specific areas, including:

(1) the ontology of space and time and the development of efficient and robust space-time database models and languages,

(2) inexactness and scaling issues,

(3) graphical user interfaces and query optimization,
(4) indexing techniques for space-time databases.

Spatio-Temporal data models are the core of a Spatio-Temporal Information System (STIS); they define object data types, relationships, operations and rules to maintain database integrity. A rigorous data model must anticipate spatio-temporal queries and analytical methods to be performed in the STIS. Spatio-temporal database models are intended to deal with real world applications, where spatial changes occur over the time line. A serious weakness of existing models is that each of them deals with few common characteristics found across a number of specific applications. Thus the applicability of the model to different cases, fails on spatio-temporal behaviours not anticipated by the application used for the initial model development.

The study of the literature of the domain highlighted a set of precise characteristics of existing models that stand for the requirements of spatio-temporal database community. These requirements implicitly form an evaluation norm for spatio-temporal data modeling. From this study common directions of modeling in the area were identified and weak and strong points of different research approaches were also detected. Consequently, this allows to recognise the achievements of previous works, to identify the issues where we should concentrate to and as such choose the routes for subsequent improvements.

The above-mentioned requirements fall in four categories. The first category deals just with the nature of time including the basic features that are used to describe it. The second category handles the pure spatial aspects of the existing approaches. The third deals with the unified spatio-temporal semantics, while the last category considers the query capabilities of the models. As we believe, if these requirements are followed carefully in the process of designing a spatio-temporal database model, a robust and expandable model can be achieved, capable of dealing with most of the real world spatio-temporal processes.

Temporal Semantics

***Granularity*:** Granularity is specified by an anchored point on the time axis and a partitioning length [1996]. The anchored point denotes where the partitioning begins while the partitioning length denotes the size of each granule. Different applications require different levels of granularity.

Temporal operations: In the literature a series of specific operations describing temporal relationships have been proposed and proved necessary in handing any time-referenced information. Allen [in 1983] was the first who introduced such kind of operations (e.g. timepoint *T "inside"* temporal period *A* which *"meets"* period *B*).

***Time density*:** This issue arises whether time should be modeled as discrete elements (isomorphic to integers) or as continuous elements (isomorphic to real numbers). More specifically, time density is closely related to the types of changes/events that can occur to the value of a thematic or spatial characteristic. Stable features that are exposed to sudden events present *stepwise* constant values (e.g. position of tectonic rocks). Continuously changing attributes can be divided into two sub-categories according to their pattern of change. As such we notice either *uniform* (e.g. ships, airplanes) or *irregular* (e.g. taxis moving in a city centre) types of continuous changes. A third type of entities exhibits *discrete* values (e.g. seismograms that interrelate measurements of earthquakes at different locations), which are collected on a periodical or irregular basis. Additionally some entities are static and never change (e.g. historic battlefields) while other entities may be measured or depend on the time itself.

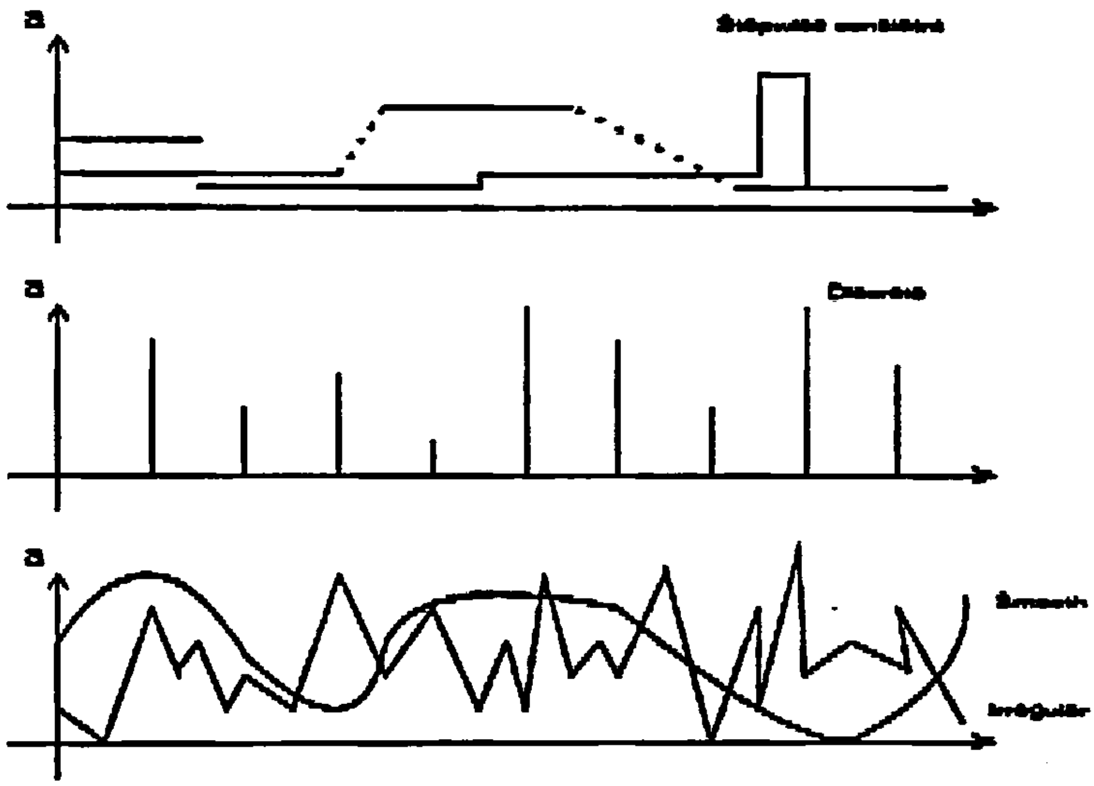


Figure 1 Types of changes according to time density

Figure 1 Types of changes according to time density

Representation of time:

In a model time is represented by timestamps, where representation methods are different for each model. This criterion allows us to compare each modeling technique, by whether maintaining the duration of the status of an object or recording events that imply status change. A practical question that arises with the representation of spatio-temporal data is what to timestamp or on which level to associate the temporal references. In [1994] Worboys depicts two approaches of amalgamating spatial and temporal references in a computer-based system:

- The first option is to timestamp the entire geographical object. This is likely to be the most inexpensive of the options in terms of computer storage. However, it allows only a limited expression of the temporal properties of the object throughout its life.

- The second option is to fuse time and space at the primitive spatial object level (point, polygon).

Although this option leads to a greater storage overhead, it allows a much finer granularity of expression of temporal variability within objects.

Transaction / Valid time: There are two different clauses that a model utilizes to associate time with spatial changes-processes. The transaction time (or registration time) indicates the time an event is actually recorded in the database. The valid time (or real-world time) describes the time that an event actually happened in the real world. A spatio-temporal model that supports both transaction and valid time is said to maintain bitemporal time.

Time order: Observes two major metaphors/criteria for describing the perspective of time, that of time as an arrow, representing progress, and as a cycle, representing constancy and continuity.
They are complemented by two other (non-linear) metaphors, namely branching and multidimensional time.

Lifespan: This factor shows if a model supports and deals with the duration of an event. This also concerns whether a model keeps track of the history of the real world objects, in terms of storing the lifespan of a discrete phenomenon or the temporal differences for a continuous one.

Spatial Semantics

Structure of space: This criterion represents the two basic approaches for computer storage of geographic data, which are the *raster* and *vector* spatial data models. *Raster* data are structured as an array of cells, pixels or voxels for 2D or 3D representations respectively. Space is partitioned into grids where each cell is addressed by its position in the raster array. On the other hand, *vector* techniques describe each spatial object in terms of start and end points. Vector representations make more efficient use of computer storage as they utilize only useful data and not the entire plane.

Orientation/Direction: This standard demonstrates whether a model supports the orientation and the direction features that real world objects show in space (e.g. *on the left side of, to the right*).

Measurement: This issue examines whether it is possible to get a value of a spatial object (e.g. *length, perimeter, distance* etc) using a particular model or if a model supports comparative operations such as *bigger, longer*.

Topology: With this criterion we distinguish existing spatio-temporal modeling techniques according to whether they support different topological relationships for the real world spatial objects lists all possible topological relationships between objects of various dimensions in up to 4- dimensional space.

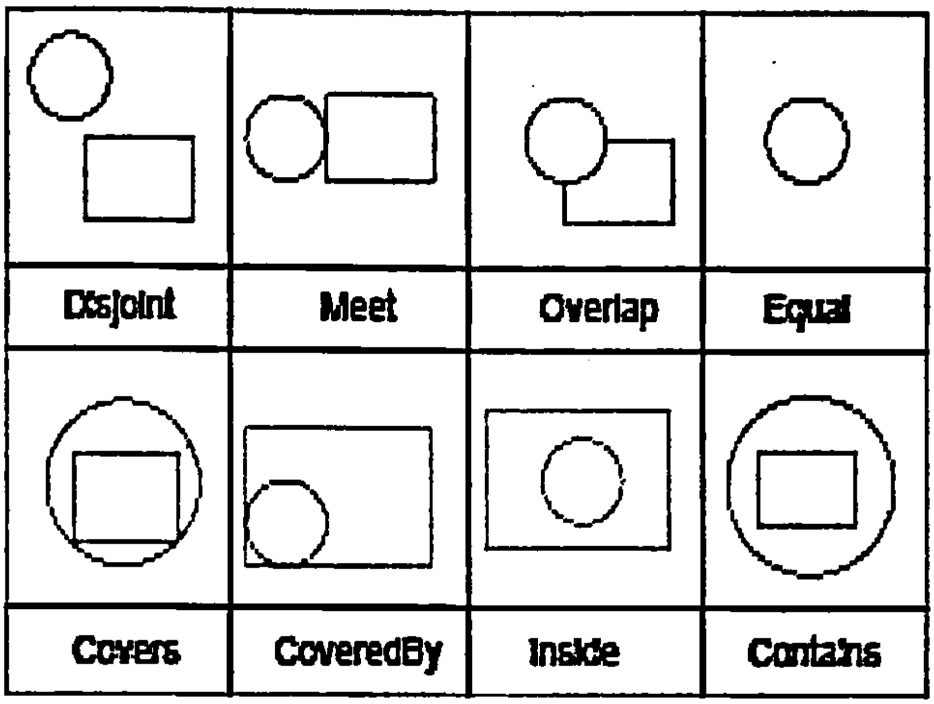


Figure 2 Topological relationships

The Spatio-Temporal Semantics

Data types: This issue refers to the basic spatial, temporal or spatio-temporal data types adopted by each model. Examples of spatial data types are the point, line and region whereas temporal point and interval are samples of temporal data types. Finally, moving point and moving region are characteristic cases of unified spatio-temporal data types.

Primitive notions: This criterion specifies the abstraction of the real world used by each model. Each model concentrates on different aspects of the real world to represent spatio-temporality in the information system. The notions vary depending not only on the method used but on the observations and choices of the particular modeller as well

Type of change: This norm compares the models if they are able to deal with changes in shape and size of the objects. Models are also evaluated whether a change in the description of a spatio-temporal object can be combined with a synchronous representation of the change of an object's position.
Consequently, the morphology, topology and attributes of a spatio-temporal object may or may not change over time, allowing for eight different scenarios

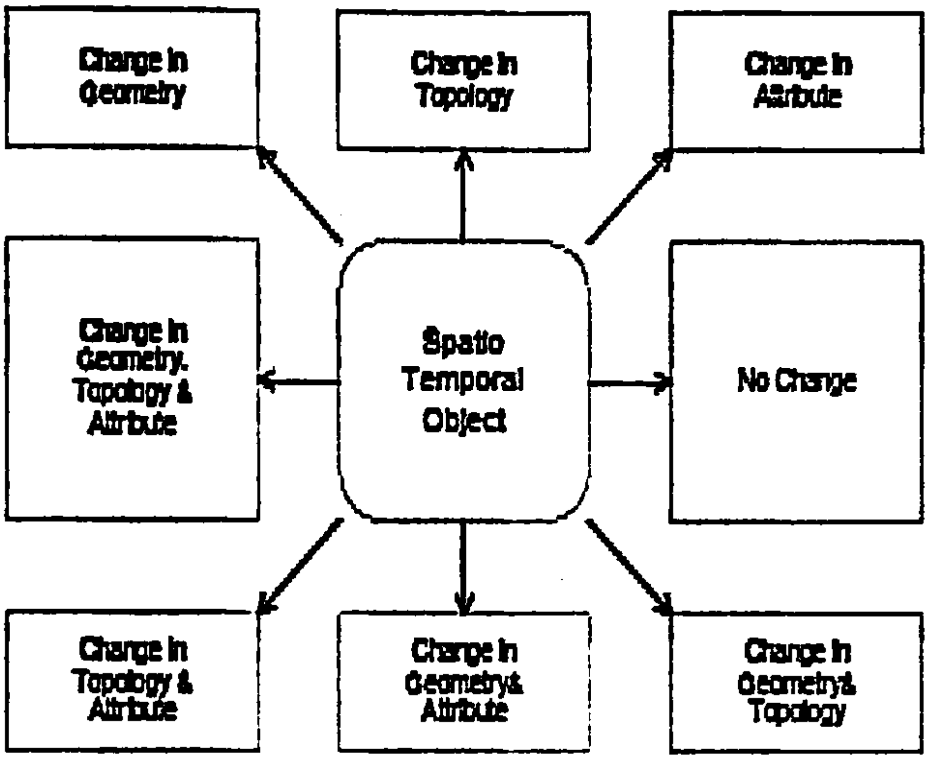


Figure 3 The eight possible types of change of a spatio-temporal object

This norm further considers whether a model supports spatio-temporal real world objects that change continuously or just objects that are subject to discrete changes. An additional emerging criterion that further categorizes existing approaches that follow the continuous paradigm is whether the latter can deal with the movement of the spatial objects over time. This is an extra decisive factor that differentiates models that support changes upon the position and/or the extent of objects in the unified space-time continuum.

Evolution in time and space: This factor shows if there are defined functions like *evolution, creation, fusion etc.* to observe and describe the movement or change of objects in space, independently from their object identification. The norm is also applied to compare models on the existence of operations able to calculate the velocity and/or the acceleration of the movement of spatio-temporal objects. There are specific difficulties presented by the evolution of objects at different speeds. Normally, a physical object is thought to continuously move/change during its lifespan. On the other hand, a reoccurring event must be viewed discretely. Slow continuous evolution of an object may also be disrupted by sudden drastic changes, some of which may be permanent, others temporary, or even part of a cycle. To model such evolution, a modeller may choose a discrete representation or alternatively he may use step semantics or some kind of interpolation to represent the unrecorded states. Similarly, extrapolation can be utilized to predict the short-term future.

***Space-time Topology*:** This criterion sets a standard whether models can estimate metrics like values of distance, direction and change in size of a particular object. It further evaluates the ability of the models to represent topological relationships (figure 2) between (in particular continuously) evolving spatial objects for a certain period of time. In other words, it considers relationships that are defined as the combination of the spatial topological relationships with Allen's temporal relationships.

Object Identities: Another issue that can be employed to evaluate the modeling ability of existing spatio-temporal data models is the manipulation of the identity of an object. In particular, the lifespan of an object is an important application dependant variable. The question is when does "change" affect an object so as not to be called the same object any more? Some times it may be more appropriate to destroy the original instance of an object and re-create a new one, due to an extensive change. Another critical issue is that of splitting or unifying objects.

Dimensionality:
With this criterion models are examined whether they support 2 dimensions to model the spatio-temporal objects, as traditional GIS do. Although "2.5" dimensional solutions exist (perspectives, stereo views etc.), volumetric 3 dimensional GIS provide advantages in displaying spatio-temporal data. In more recent approaches, relegating the attribute value associated with grid locations to a fourth dimension, time can be introduced as a fifth.

Query Capabilities
This section classifies existing spatio-temporal database models in terms of their query capabilities. The proposed categorization is a superset of a similar benchmarking framework presented in.

Queries about locations, spatial properties, and spatial relationships: Queries of this category involve stationary reference objects. Examples include attribute of entities independent of space and time (e.g. who is the owner of this parcel?), as well as point (e.g. where is this building?), range / distance-based (e.g. find gas stations in this rectangular area / in this circle), nearest-neighbor (e.g. find the closest gas station) and topological queries (e.g. find streets crossing a particular area).

Queries about time, temporal properties, and temporal relationships: These queries can be simple temporal queries (e.g. what is the state of a spatial feature at time t?), temporal range queries (e.g. what happens to that feature over a given period?) and temporal relationship queries (e.g. find stadiums that were built in Athens concurrently and their construction took less than six months).

Queries about spatio-temporal behaviors and relationships: This set of queries is further classified into three sub-categories: (1) *Simple* spatio-temporal queries on discretely changing (e.g. what is the state of a parcel at time t?) or moving reference objects; examples include distance-based (e.g. find humans passed close to me yesterday) and similarity-based queries (e.g. find a similar trajectory to the one I followed today).

(2) Spatio-temporal *range* queries (e.g. what happens to a region over a given period?) and/or *join* queries; examples include distance-join (find the three closest restaurants to my fleet) and similarity-join queries (find the two most similar pairs of trajectories in month January).

(3) Spatio-temporal *behavior* queries involving unary operators, such as traveled distance or speed (e.g. find the average speed on Saturday nights, when/where did the fire reach its maximal rate of spread?). Based on related research work, the above queries constitute a minimum functionality a spatio-temporal system should provide and we expect that soon coming releases of commercial DBMSs will partially support them.

5. Objectives And Importance

Objectives:

1. to describe the limitations of current database management system.
2. to describe the changing current database trends
3. to understand the need of new data types for the different type of versatile data
4. to classify the data with respect to space and time
5. to understand the concept of spatial database
6. to understand the concept of temporal database
7. to understand the spatio-temporal database management system
8. to introduced the use of spatio-temporal dbms in GIS

Importance:

1. Spatio-Temporal database is helpful for wide variety of applications where there is a need of complex data types. It has a wide scope.
2. Spatial data models are required to support cartography, topography, cadastral and other relevant applications.

6. Introduction to Temporal and Spatial Databases

6.1 What are Temporal Databases?

- Non-Temporal Databases
- Temporal Databases
- Different Forms of Temporal Databases

Non-Temporal Databases

Commercial database management systems (DBMS) such as Oracle, Sybase, Informix and O2 allow the storage of huge amounts of data. This data is usually considered to be valid now. Past or future data is not stored. Past data refers to data which was stored in the database at an earlier time instant and which might has been modified or deleted in the meantime. Past data usually is overwritten with new (updated) data. Future data refers to data which is considered to be valid at a future time instant (but not now).

A DBMS stores the data in a well-defined format. A relational DBMS, for example, stores data in tables (also called relations). Thus, a relational database actually contains a set of tables. Each table contains rows (tuples) and columns (attributes). A row contains data about a specific entity, for example, an employee. Each column specifies a certain property of these entities, for example, the employee's name, salary etc. The following table stores data about employees:

EmpID	Name	Department	Salary
10	John	Sales	12000
12	George	Research	10500
13	Ringo	Sales	15500

Temporal Databases

Temporal data strored in a temporal database is different from the data stored in non-temporal database in that a time period attached to the data expresses when it was valid or stored in the database. As mentioned above, conventional databases consider the data stored in it to be valid at time instant now, they do not keep track of past or future database states. By attaching a time period to the data, it becomes possible to store different database states.

A first step towards a temporal database thus is to timestamp the data. This allows the distinction of different database states. One approach is that a temporal database may timestamp entities with time periods. Another approach is the timestamping of the property values of the entities. In the relational data model, tuples are timestamped, where as in object-oriented data models, objects and/or attribute values may be timestamped.

What time period do we store in these timestamps? As I mentioned already, there are mainly two different notions of time which are relevant for temporal databases. One is called the **valid time**, the other one is the **transaction time**. Valid time denotes the time period during which a fact is true with respect to the real world. Transaction time is the time period during which a fact is stored in the database. Note that these two time periods do not have to be the same for a single fact. Imagine that we come up with a temporal database storing data about the 18th century. The valid time of these facts is somewhere between 1700 and 1799, where as the transaction time starts when we insert the facts into the database, for example, January 21, 1998.

Assume we would like to store data about our employees with respect to the real world. Then, the following table could result:

EmpID	Name	Department	Salary	ValidTimeStart	ValidTimeEnd
10	John	Research	11000	1985	1990
10	John	Sales	11000	1990	1993
10	John	Sales	12000	1993	INF
11	Paul	Research	10000	1988	1995
12	George	Research	10500	1991	INF
13	Ringo	Sales	15500	1988	INF

The above valid-time table stores the history of the employees with respect to the real world. The attributes ValidTimeStart and ValidTimeEnd actually represent a time interval which is closed at its lower and open at its upper bound. Thus, we see that during the time period [1985 - 1990), employee John was working in the research department, having a salary of 11000. Then he changed to the sales department, still earning 11000. In 1993, he got a salary raise to 12000. The upper bound INF denotes that the tuple is valid until further notice. Note that it is now possible to store information about past states. We see that Paul was employed from 1988 until 1995. In the corresponding non-temporal table, this information was (physically) deleted when Paul left the company.

Different Forms of Temporal Databases

The two different notions of time - valid time and transaction time - allow the distinction of different forms of temporal databases. A **historical database** stores data with respect to valid time, a **rollback database** stores data with respect to transaction time. A **bitemporal database** stores data with respect to both valid time and transaction time.

As we mentioned above, commercial DBMS are said to store only a single state of the real world, usually the most recent state. Such databases usually are called **snapshot databases**. A snapshot database in the context of valid time and transaction time is depicted in the following picture:

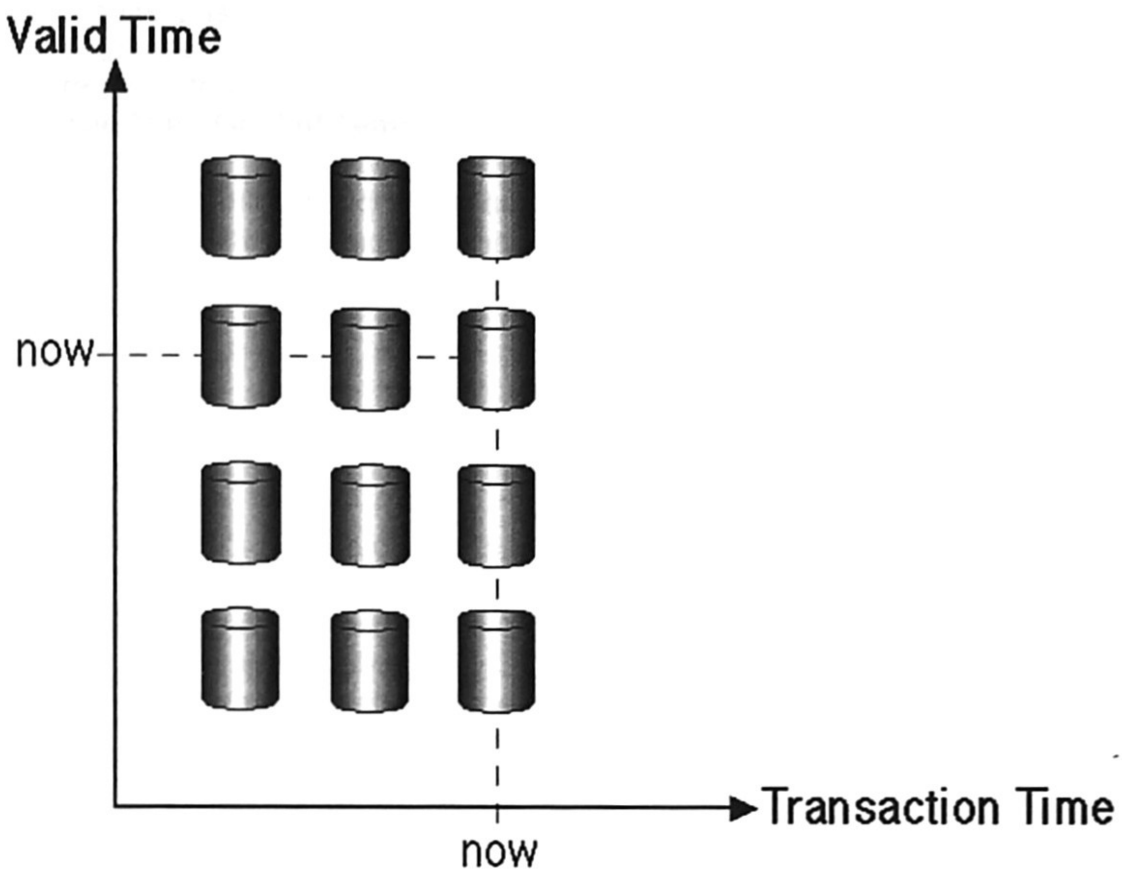


Fig 4: Valid Time And Transaction Time

On the other hand, a **bitemporal DBMS** such as TimeDB stores the history of data with respect to both valid time and transaction time. Note that the history of when data was stored in the database (transaction time) is limited to past and present database states, since it is managed by the system directly which does not know anything about future states.

A table in the bitemporal relational DBMS TimeDB may either be a snapshot table (storing only current data), a valid-time table (storing when the data is valid wrt. the real world), a transaction-time table (storing when the data was recorded in the database) or a bitemporal table (storing both valid time and transaction time). An extended version of SQL allows specifying which kind of table is needed when the table is created. Existing tables may also be altered (schema versioning). Additionally, it supports **temporal queries, temporal modification statements** and **temporal constraints.**

The states stored in a bitemporal database are sketched in the picture below. Of course, a temporal DBMS such as TimeDB does not store each database state separately as depicted in the picture below. It stores valid time and/or transaction time for each tuple, as described above.

1. **Why we need Temporal Database:**
   In the last two decades, the relational data model has gained popularity because of its simplity and solid mathematical foundation. However, the relational data model as proposed by Codd [1970] does not address the temporal dimension of data. Variation of data over time is treated in the

same way as ordinary data. This is not satisfactory for applications that require past persent, and/or future data values--to be dealt with by the database, in real life such applications abound. In fact, most application requires temporal data to a certain extent.

2. **The Main Goal of Temporal Database:**

- Identification of an appropriate data type for time
- Prevent fragmentation of an object description
- Provide query algebra to deal with temporal data
- Compatible with old database without temporal data

3. **What we can do by Temporal Database:**

- It's easy to deal with temporal data
- Record the data changed with time is more convenient
- An object description can be well defined without fragmentation
- Having relation model to describe temporal data
- Having query algebra to deal with temporal data
- It works to handle static data (without time dimension) in temporal database
- The traditional database algebra still work in temporal database
- The new query algebra to control time dimemsion is similar to tranditional database algebra

6.2 What are Temporal Database Management Systems?

Non-Temporal Database Management Systems (DBMS)

- Temporal DBMS

Non-Temporal DBMS

Commercial database management systems (DBMS) such as Oracle, Sybase, Informix andO2 are non-temporal DBMS since they do not support the management of temporal data. A temporal DBMS should support temporal data definition language, a temporal data manipulation language and a temporal query language, temporal constraints.

Although some of the DBMS support data types for dates and time, they cannot be considered to be temporal DBMS. For example, the specification of a query considering several different database states (the history of data) is left to the user, without any support by the system.

Temporal DBMS

A temporal DBMS such as TimeDB supports

1. a temporal data definition language,
2. a temporal data manipulation language,
3. a temporal query language, and
4. temporal constraints (such as temporal referential integrity).

TimeDB supports SQL, however in an extended form. Basically, two keywords are added (VALIDTIME and TRANSACTIONTIME).

Temporal Data Definition Language

In TimeDB, a bitemporal table can be created the following way:

```
CREATE TABLE Employees (EmpID INTEGER, Name CHAR(30), Department CHAR(40), Salary INTEGER)
AS VALIDTIME AND TRANSACTIONTIME;
```

Temporal Data Manipulation Language

The following statement inserts temporal data about John:

```
VALIDTIME PERIOD '1985-1990'
INSERT INTO Employees VALUES (10, 'John', 'Research', 11000);

VALIDTIME PERIOD '1990-1993'
INSERT INTO Employees VALUES (10, 'John', 'Sales', 11000);

VALIDTIME PERIOD '1993-forever'
INSERT INTO Employees VALUES (10, 'John', 'Sales', 12000);
```

Temporal Query Language

To query the data, the same keywords are used:

```
VALIDTIME
SELECT * FROM Employees;
```

This query returns the history of the employees with respect to valid time (when were they employed). The following query find out when the tuples in table Employees were stored in the database :

```
TRANSACTIONTIME
SELECT * FROM Employees;
```

To find out both valid time and transaction time, a combination of the keywords can be used:

```
VALIDTIME AND TRANSACTIONTIME
SELECT * FROM Employees;
```

In fact, any legal standard SQL query can be extended with one of the combinations VALIDTIME, TRANSACTIONTIME or VALIDTIME AND TRANSACTIONTIME.

Temporal Constraints

Temporal integrity constraints can be expressed similarly. For example, a referential integrity constraint demanding that at each time instant an employee is a member of a department, the corresponding department itself must exist, can be expressed the following way:

```
CREATE TABLE Employees (EmpID INTEGER, Name CHAR(30), Department CHAR(40) VALIDTIME REFERENCES Departments(department), Salary INTEGER) AS VALIDTIME AND TRANSACTIONTIME;
```

6.3 An Introduction to Spatial Database Systems

A definition of a spatial database system as a database system that offers spatial data types in its data model and query language and supports spatial data types in its implementation, providing at least spatial indexing and spatial join methods. Spatial database systems offer the underlying database technology for geographic information systems and other applications.

In various fields there is a need to manage *geometric*, *geographic*, or *spatial* data, which means data related to *space*.
For example,
The two-dimensional abstraction of the surface of the earth – that is, geographic space and a man-made space like the layout of a VLSI design.
A volume containing a model of the human brain, or another 3d-space representing the arrangement of chains of protein molecules.

Characteristic for the technology emerging to address these needs is the capability to deal with *large collections of relatively simple geometric objects*, for example, a set of 100 000 polygons.
This is somewhat different from areas like CAD databases (solid modeling etc.) where geometric entities are composed hierarchically into complex structures, although the issues are certainly related.

Several terms have been used for database systems offering such support like *pictorial*, *image*, *geometric*, *geographic*, or *spatial database system*. The terms "pictorial" and "image" database system arise from the fact that the data to be managed are often initially captured in the form of digital raster images (e.g. remote sensing by satellites, or computer tomography in medical applications). The term "spatial database system" has become popular during the last few years, to some extent through the series of conferences "Symposium on Large Spatial Databases (SSD)" held bi-annually since 1989 and is associated with a view of a database as containing sets of objects in space rather than images or pictures of a space. Indeed, the requirements and techniques for dealing with objects in space that have identity and well-defined extents, locations, and relationships are rather different from those for dealing with raster images. It has therefore been suggested to clearly distinguish two classes of systems called ***spatial database systems*** and ***image database systems***, respectively. Image database systems may include analysis techniques to extract objects in space from images, and offer some spatial database functionality, but are also prepared to store, manipulate and retrieve raster images as discrete entities.

Another definition of spatial database system is as follows:

(1) A spatial database system is a database system.
(2) It offers spatial data types (SDTs) in its data model and query language.
(3) It supports spatial data types in its implementation, providing at least spatial indexing and efficient algorithms for spatial join.

Let us briefly justify these requirements. (1) the fact that spatial or geometric information is always connected with "non-spatial" (e.g. alphanumeric) data. Nobody cares about a special purpose system that is not able to handle all the standard data modeling and querying tasks. Hence a spatial database system is a full-fledged database system with *additional* capabilities for handling spatial data.

(2) Spatial data types,.e.g. POINT, LINE, REGION, provide a fundamental abstraction for modeling the structure of geometric entities in space as well as their relationships (*l intersects r*), properties (*area*(*r*) > 1000), and operations (*intersection*(*l*, *r*) – the part of (*l* lying within *r*). *Which* types are used may, of course, depend on a class of applications to be supported (e.g. rectangles in VLSI design, surfaces and volumes in 3d). Without spatial data types a system does not offer adequate support in modeling.

(3) A system must at least be able to retrieve from a large collection of objects in some space those lying within a particular area without scanning the whole set. Therefore spatial indexing is mandatory. It should also support connecting objects from different classes through some spatial relationship in a better way than by filtering the Cartesian Product (at least for those relationships that are important for the application).

The purpose is to present a coherent way of the fundamental problems and their solutions in spatial database systems. The focus is on describing solutions that have been found rather than on listing many open problems. We consider spatial DBMS to provide the underlying database technology for geographic information systems (GIS) and other applications. As such, they can offer only some basic capabilities; it is not claimed that a spatial DBMS is directly usable as an application-oriented GIS.
In the following four sections we consider modeling, data structures, data types and data representation for spatial database systems.

Modeling

What needs to be represented?

The main application driving research in spatial database systems are GIS. Hence we consider some modeling needs in these areas which are typical also for other applications.
Examples are given for two-dimensional space, but almost everywhere, extension to the three- or more-dimensional case is possible. There are two important alternative views of what needs to be represented:

(i) *Objects in space:* We are interested in distinct entities arranged in space each of which has its own geometric description.
(ii) *Space:* We wish to describe space itself, that is, say something about every point in space.

The first view allows one to model, for example, cities, forests, or rivers. The second view is the one of thematic maps describing e.g. land use or the partition of a country into districts. Since raster images say something about every point in space, they are also closely related to the second view. We can reconcile both views to some extent by offering concepts for modeling.

(i) *single objects*, and
(ii) *spatially related collections of objects*.

For modeling *single objects*, the fundamental abstractions are *point, line*, and *region*. A point represents (the geometric aspect of) an object for which only its location in space, but not its extent, is relevant. For example, a city may be modeled as a point in a model describing a large geographic area (a large scale map). A line (in this context always to be understood as meaning a curve in space, usually represented by a polyline, a sequence of line segments) is the basic abstraction for facilities for moving through space, or connections in space (roads, rivers, cables for phone, electricity, etc.). A region is the abstraction for something having an extent in 2d-space, e.g. a country, a lake, or a national park. A region may have holes and may also consist of several disjoint pieces. Figure 5.shows the three basic abstractions for single objects.

Figure 5: The three basic abstractions point, line, and region

The two most important instances of *spatially related collections of objects* are *partitions* (of the plane) and *networks* (Figure 6).
A *partition* can be viewed as a set of region objects that are required to be disjoint. The adjacency relationship is of particular interest, that is, there exist often pairs of region objects with a common boundary. Partitions can be used to represent thematic maps. A *network* can be viewed as a graph embedded into the plane, consisting of a set of point objects, forming its nodes, and a set of line objects describing the geometry of the edges. Networks are ubiquitous in geography, for example, highways, rivers, public transport, or power supply lines.

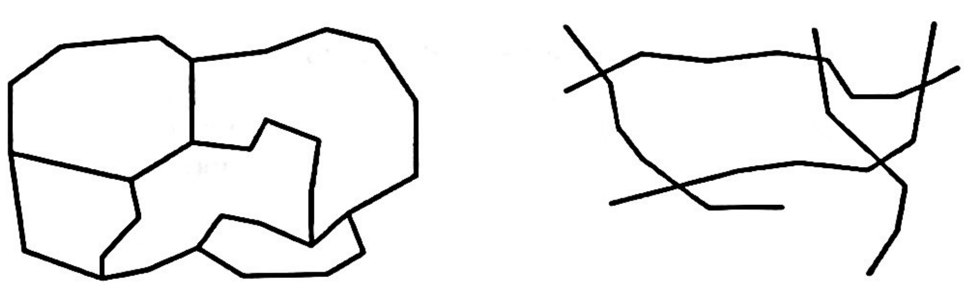


Fig 6: Partitions and networks

Obviously, we have mentioned only the most fundamental abstractions to be supported in a spatial DBMS (in case of GIS). For example, other interesting spatially related collections of objects are *nested partitions* (e.g. a country partitioned into provinces partitioned into districts etc.) or a digital terrain (elevation) model.

Representing Geographic Features:

How do we describe geographical features?

- by recognizing two *types of data*:
  - **Spatial data** which describes location (where)
  - **Attribute data** which specifies characteristics at that location (what, how much, and when)

How do we represent these digitally in a GIS?

- by grouping into ***layers*** based on similar characteristics (e.g hydrography, elevation, water lines, sewer lines, grocery sales) and using either:
  - **vector** data model (*coverage* in ARC/INFO, *shapefile* in ArcView)
  - **raster** data model (*GRID or Image* in ARC/INFO & ArcView)

- by selecting appropriate ***data properties*** for each layer with respect to:
  - ***projection, scale, accuracy,*** and ***resolution***

How do we incorporate into a computer application system?

- by using a relational Data Base Management System (DBMS)

GIS Data Structures

- Spatial Data Types And Attribute data types

Raster Data Structures: represent geography via grid cells
---- tessellations
---- run length compression
---- quad tree representation
---- BSQ/BIP/BIL
---- BMS representation
---- File formats

Vector Data Structures : represents geography via coordinates
--- whole polygon
--- point and polygon
--- node/arc/polygon
--- Tins
--- File formats

Spatial Data Types

- *continuous:* elevation, rainfall, ocean salinity
- *areas:*
  - *unbounded:* e.g. landuse, market areas,soils, rock type

- *bounded: e.g.* city/county/state boundaries, ownership parcels, zoning
- *moving: e.g.* air masses, animal herds, schools of fish

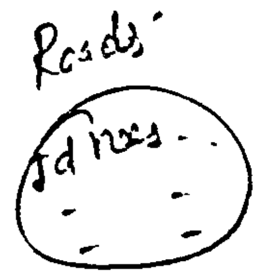
- *networks:* e.g. roads, transmission lines, streams
- *points:*
  - *fixed:* e.g. wells, street lamps, addresses
  - *moving:* e.g. cars, fish, deer

Attribute data types

Categorical (name):

—–nominal

- no inherent ordering
- land use types, county names



——ordinal

- inherent order
- road class; stream class

Often coded to numbers eg SSN but can't do arithmetic.

Numerical

Known difference between values

—— interval

- No natural zero
- can't say 'twice as much'
- temperature (Celsius or Fahrenheit)

—— ratio

- natural zero
- ratios make sense (e.g. twice as much)
- income, age, rainfall

may be expressed as *integer* [whole number] or *floating point* [decimal fraction]

Attribute data tables can contain locational information, such as addresses or a list of X, Y coordinates. ArcView refers to these as *event tables*. However, these must be converted to true spatial data (shape file), for example by geocoding, before they can be displayed as a map.

Data Base Management Systems (DBMS)

Parcel Table			
Parcel#	Address	Block	$ Value
8	501 N Hi	1	105,450
9	590 N Hi	2	89,780
36	1001 W. Main	4	101,500
75	1175 W. 1st	12	98,000

Contain Tables or *feature classes* in which:

——— *rows*: entities, records, observations, features:

- 'all' information about one occurrence of a feature

——— *columns*: attributes, fields, data elements, variables, items (ArcInfo)

- one type of information for all features

The *key field* is an attribute whose values uniquely identify each

Relational DBMS (RDBMS)

Tables are related, or *joined*, using a common record identifier (column variable), present in both tables, called a *secondary (or foreign) key*, which may or may not be the same as the key field.

Parcel Table			
Parcel #	Address	Block	$ Value
8	501 N Hi	1	105,450
9	590 N Hi	2	89,780
36	1001 W. Main	4	101,500
75	1175 W. 1st	12	98,000

Secondary or foreign key

Geography Table			
Block	District	Tract	City
1	A	101	Dallas
2	B	101	Dallas
4	B	105	Dallas
12	E	202	Garland

Goal: produce map of values by district/ neighborhoodProblem: no districtcode available in Parcel Table

Solution: join Parcel Table, containing values, with Geograpahy Table, containing location codings, using Block as key field

GIS Data Models:

Raster v. Vector : *"raster is faster but vector is corrector"* Joseph

- **Raster data model**
  - location is referenced by a grid cell in a rectangular array (matrix)
  - attribute is represented as a single value for that cell
  - much data comes in this form
    - images from remote sensing (LANDSAT, SPOT)
    - scanned maps
    - elevation data from USGS
  - best for continuous features:
    - elevation
    - temperature
    - soil type
    - land use

- **Vector data model**
  - location referenced by x,y coordinates, which can be linked to form lines and polygons
  - attributes referenced through unique ID number to tables
  - much data comes in this form
    - DIME and TIGER files from US Census
    - DLG from USGS for streams, roads, etc
    - census data (tabular)
  - best for features with discrete boundaries
    - property lines
    - political boundaries
    - transportation

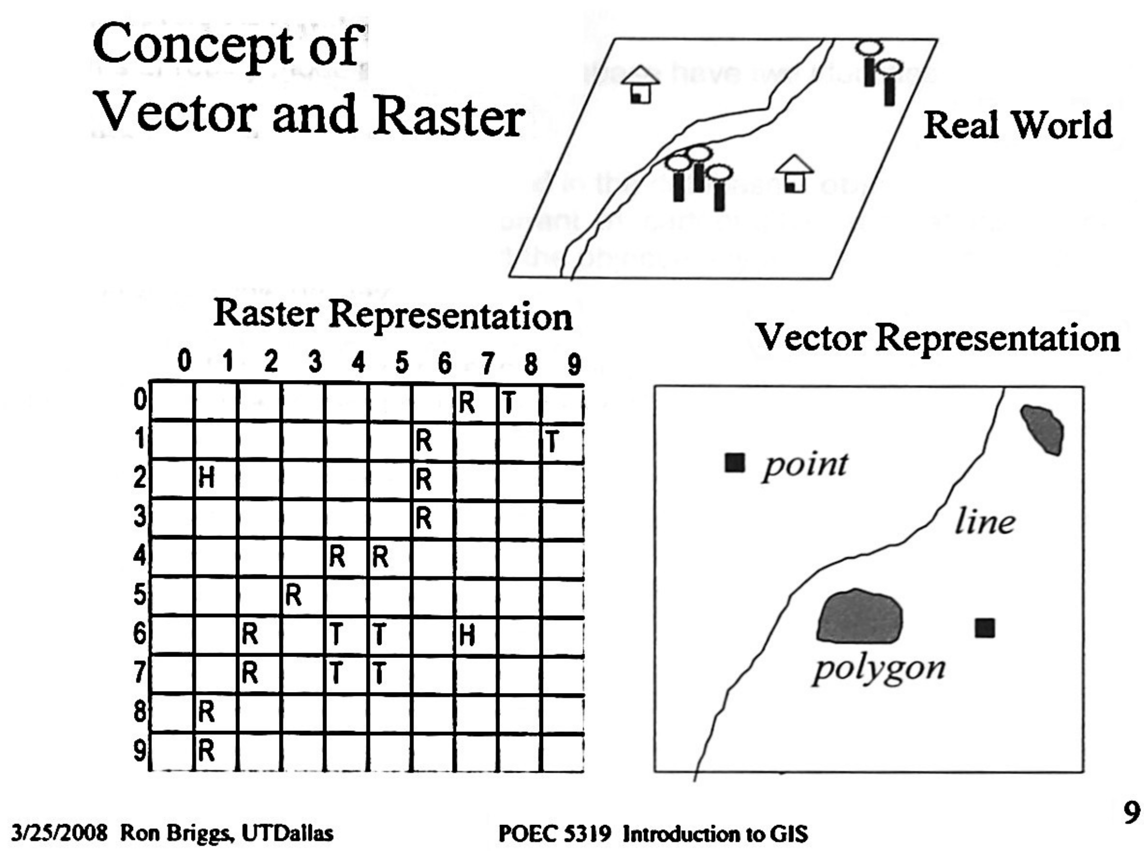


Fig 7: Vector and Raster Representation

The real world is too complex for immediate and direct understanding. We create "models" of reality that are intended to have some similarity with selected aspects of the real world. Databases are created from these models as a fundamental step in coming to know the nature and status of that reality.

So the another definition is : A Spatial Database in a collection of spatially refered data that acts as **a model of reality.**

A database is a model of reality in the sense that the database represents a selected set or approximation of phenomenon. These selected phenomenons are deemed important enough to represent in digital form. The digital representation meight be for some past, present or future time period.

For Example: Let us consider a wetlands

Wetlands data have different points of view:

Ecological organization might define wetlands as a natural resource to be preserved and restricted from development. So it requires details for describing the area's biology and physical resources.

A taxing authority might define a wetland to be a "wasteland" and of very little value to society. So it requires the description about the boundry of the "wasteland" in the database.

Fundamentals Of Database Elements:
Elements of reality modeled in a GIS database have two identities

1. the element in reality - **entity**
2. the element as it is represented in the database – **object**
3. a third identity that is important in cartographic applications is the **symbol** that is used to depict the object/entity as a feature on a map or other graphic display

Entity : An entity is "a phenomenon of interest in reality that is not further subdivided into phenomena of the same kind".

E.g. a city could be considered an entity and subdivided into component parts but these parts would not be called cities, they would be districts, neighborhoods or the like.

Object: An object is "a digital representation of all or part of an entity". The method of digital representation of a phenomenon varies according to scale, purpose and other factors.

E.g. a city could be represented geographically as a point if the area under consideration were continental in scale

Desirable database characteristics :

Database should be:

1. Contemporaneous - should contain information of the same vintage for all its measured variables

2. As detailed as necessary for the intended applications

3. The categories of information and subcategories within them should contain all of the data needed to analyze or model the behavior of the resource using conventional methods and models

4. Positionally accurate

5. Exactly compatible with other information that may be overlain with it

6. Internally accurate, portraying the nature of phenomena without error - requires clear definitions of phenomena that are included

7. Readily updated on a regular schedule accessible to whoever needs it.

7. Research Methodology

Research Methodology is a way to systematically solve the research problem. This is an Exploratory research study, also we can say it as Synthetic Research. The major emphasis in such studies is on the discovery of ideas and insights.

Here in this research, there is a survey of different aspects of spatial databases, temporal databases, and a composite study of Spatio-Temporal Databases. This research contains the review of available material and deriving the relevant hypotheses from it.

As it is explorative or synthetic research. So here in this study , first limitations of conventional databases are discussed and then the concept of spatio-temporal database is introduced. Many articles related to the field from journals, websites and books are read and understood. All information is compiled. Then it is decided to work on the theme to know "how spatio-temporal databases different from conventional databases ". In general "how spatio-temporal databases use in various growing sectors" was approach for this study to undertake. The theoretical approach for spatio-temporal database management system has been discussed so that further researchers would find a guideline to solve the shortcomings of the current spatio-temporal database management system.

exploratory research is a type of research conducted for a problem that has not been clearly defined.
exploratory research relies on sec. research such as reviewing the available literature &/or data

8. Spatio-Temporal Database Management System

Integrating space and time in databases is considered a very important research issue. The challenge in this field is to efficiently support queries that contain spatial and temporal predicates. Access methods that have been proposed so far, fail to provide satisfactory query processing in all cases. Another important issue is spatio-temporal query optimization techniques. The problem of spatio-temporal data management becomes even harder, if we take into account valid and transaction time. Although, interesting bi-temporal access methods have been proposed, the issue of spatial attributes has not been considered.

Spatio-temporal database is a database that embodies spatial, temporal, and spatiotemporal database concepts, and captures spatial and temporal aspects of data. Spatio-Temporal Database is a special temporal database and it has all the features of temporal databases. If we look at any timestamp only, it is almost the same as the conventional spatial database.

The importance of incorporating time in modeling data is widely recognized by many researchers, but most of the intentions mentioned are focused around the non-spatial domain. Particularly, temporality has been studied in banking, administrative and other commercial databases. Although the above concept of time handling is the same, the great difference between spatio-temporal and a spatial database is what they refer to. Since spatial data are spatially referenced, the spatial topology has to be maintained in the updating process. This maintenance of topology becomes more critical when different version of an object have to be accessible, for each given of time slices, and where correctness of topology is required.
Validity is one of the main concepts in spatial databases. Logically, no decision can be made when there is no valid data. If only the most recent data are kept, and out-dated ones are deleted, one can have the static database which is a snapshot of the real world (just like taking a photograph of a dynamic object). On the other hand, many spatial analysis applications do need to travel through the history of data and retain an impression about its old status. In this case, it can be seen that deleting data, even if it is non-valid, may be disastrous. In addition, many applications require estimation about the future. In this respect, one concludes that considering time as an independent dimension, which allows traveling alone the time line, is necessary. Keeping an active time domain in a spatial database, gives rise to the term spatio-temporal databases.

The ideal spatio-temporal database mentioned, besides having the normal functions essential to every spatial database, also has the ability for keeping track of changed data. In such databases, the mechanism of renewing the most recent spatial and temporal topologies is implemented. In addition to the process of updating geographical objects, keeping the valid topological (either temporal or spatial) relationships are also operational.
Spatio-temporal objects are limited to a given temporal validity. When new information is assigned, the database has to be changed, in order to add the new data. In such a case the database may have to rebuild completely for storing the new state, which causes a great amount of redundancy. Alternatively the database might be time stamped in order to indicate the temporal validity of its objects. The ability of versioning a database is essential

for all spatio-temporal applications, where knowledge of the real world is uncertain or changing through time.

The Architecture of a Flexible Query processor for Spatio-Temporal Databases

For designing a flexible query processor for spatio-temporal databases, a solution is to design a bottom-up interface, whose parameters handle different spatiotemporal applications. A relational schema is proposed that can cope with various spatio-temporal data types. Based on this model,there is a design of flexible query processor for spatio-temporal databases.

Emerging database management systems that can handle spatial data types have changed both GISystems and GIScience. System wise, this technology enables a transition from the current GIS technology to a new generation of *spatial information appliances*, tailored to specific user needs [1999]. For the GIScience community, it enables many theoretical proposals to face the crucial test of practice. One of the important challenges for the GIScience community is finding ways to use spatially enabled DBMS to build innovative applications which deal with spatio-temporal data [1999 and 2000]. Modeling spatio-temporal applications is a complex task that involves representing objects with spatial extensions and attribute values that change over time [2003]. To deal with spatio-temporal data, one alternative is building a specialized DBMS created for efficient support of spatio-temporal data types. When is not possible to use a specialized DBMS, one has to build a layered architecture on top of an existing object-relational DBMS.

In this case, one basic question arises: *how to design a flexible query processor for spatio-temporal data using object-relational DBMS?*
A flexible query processor needs to be able to cope with different applications of spatio-temporal data and their needs for queries and responses. To solve this problem, a popular approach in the literature is to provide specialized algebras for different applications. For example, Gũting, Bohlen et al. [2003] present a model for moving objects that includes moving points and moving regions. Hornsby and Egenhofer [2000] and Medak [2001] propose models for the life and evolution of socio-economic objects. These specialised models can lead to databases where each type of application is handled by a different query processor. Obviously, this is not desirable for developers of applications using spatio-temporal databases. Ideally, the architecture of the query processor would have a unified and flexible way of dealing with the different applications of spatio-temporal data.

Spatio-temporal data handling: top-down x bottom-up approaches

In this section, we discuss four different levels in a database design that address different aspects for handling of spatio-temporal data:

(a) A set of data types and an associated algebra;
(b) A conventional data model for spatio-temporal data;
(c) A spatiotemporal query language;
(d) An application programming interface with suitable parameters.

We consider the first three as "top-down" approaches and the fourth as a "bottom-up" choice. The first alternative is to define a set of spatio-temporal data types and operators.

The DBMS is extended to support these data types and operators and will provide an associated query language. This is the approach taken by Güting [2005] that defines algebra for moving objects. His spatio-temporal data types for moving objects are embedded in a query language to answer queries as: *"Given the trajectories of two airplanes, when they will pass over the same location?"* .Similarly, Medak [2001] proposes an algebra for modeling change in socio-economical units. Medak's algebra provides answers to queries such as: *"When was this parcel divided?"* The main challenge of this approach is finding a suitably small set of data types and operators for handling all types of spatio-temporal data. Currently, we only find spatio-temporal algebras for specialized applications (e.g., moving objects).
The second choice is to design a conventional model for spatio-temporal data. In this case, the designer starts from an external view of the problem and provides a set of classes (or an equivalent E-R model). These classes encapsulate abstractions such as geometry, attributes and their changes. Examples include STER [Tryfona and Jensen 1999] and MADS [Parent, Spaccapietra et al. 1999] (see also Pelekis et al [2004] for a review of similar models). The main drawback of these approaches is the large variety of different application semantics for combining space and time. These models work fine for some applications, but will not fit other cases well.
The third approach is the design of a general spatio-temporal query language, which needs a well-defined set of predicates for spatial, temporal, and spatio-temporal queries. For spatial data, topological and directional operators are already well established in the literature [Egenhofer and Franzosa 1991] [Papadias and Egenhofer 1997] [Clementini and Di Felice 1996]. Dealing with temporal data is also a well researched issue. The interval algebra for temporal operators is established [Allen 1983], as is the bitemporal model of Worboys [1994]. However, there are problems when trying to devise a unique canonical set of spatio-temporal predicates. As shown by Erwig and Schneider [2002], it is not practical to devise one such set because there are too many predicates that can be considered different. They propose two options. Either each application will develop a specialized subset of predicates, or the spatio-temporal database will provide *combinators* that allow the user to build up her or his own predicates.

Bottom-up alternative is to design a query processor as a parametrizable function. Taking in consideration the suggestion by Erwig and Schneider [2002] to design a combinator, this query processor is flexible and can be used by different applications.The set of parameters of the query processor works as a combination of spatio-temporal predicates.

The architecture of a spatio-temporal query processor

General view of a spatio-temporal database

In this section, a generic model for a geographical database, which is the basis for designing the query processor. Assume that a geographical database stores *layers*. A *layer* aggregates spatial information that covers a geographical region and has a common set of attributes and shares the same

spatial reference system. Layers supports both the object-based and field-based models of spatial information [1992].
The layer model is used by most spatial extensions of object-relational DBMS such as ORACLE SPATIAL and PostGIS. In this work we concentrate on the object-based layers.

ID_	NAME	POP_2004
261160605001	Recife	98361
261160605010	Cabanga	102548
261160605011	Sao Jose	1524
261160605012	Derby	157621
261160605013	Gracas	278951
261160605014	Jaquera	102200

Fig: 8 A Layer of Districts of Recife,PE

Figure 8 shows a layer of districts of the Brazilian city of Recife. Each district has a set of descriptive attributes, such as its name, or the population of the district in the census of 2004. The spatial extension of each district is a polygon that represents its boundaries.

Our model considers that a layer contains a set of *spatio-temporal objects (STObjects)*. An *ST-Object* is an entity that preserves its identity over time [Hornsby and Egenhofer 2000]. *Static layers* aggregate *ST-Objects* with geometry and attribute values that do not change. *Temporal layers* aggregate *ST-Objects* that change their attribute values or their geometry. The different versions of the same *ST-Object* as *spatio-temporal instances (ST-Instances) are refered here*. Each *ST-Instance* has an associated interval that is the validity time of that instance and knows its current spatial extension and its current set of attribute values.

A generic database model for spatio-temporal data

Static Layers
Our database model considers in a set of relations that include attribute relations, geometry relations and metadata relations. In a generic way, we represent the geometry and attribute relations as:

geometries (geomId:int, objId:string, spatialData: spatial)

attributes (objId:string, [att1:attType,...,attn:attType]).

Consider that each entity has a unique and persistent identification (objId). Each geometry also has a unique identifier (geomId). We use attType for conventional types such as int, double or string. We use the spatial keyword for types that can store a spatial extent. We consider these relations as *data relations* since they effectively store the spatial data. We also need to store metadata information on the database. These relations describe the geometry and attributes relations associated to each layer. Our metadata relations are:

layers (la erld: int, layerName: string)

representations (layerld: int, geomRelation: string)

attributesRel (layerld: int, attrRelation: string)

The layers relation provides a unique identifier (layerld) for each layer. It can also contains other attributes that are relevant to the layer, such as a name or a link to its spatial reference system. The representations relation associate, to each layer, its geometries relations. The model allows multiple spatial representations for the objects of the layer. The attributesRel relation points to the descriptive attributes relations associated to a layer.
This data model is suitable to store static layers. It allows more than one attribute relation for each object type, as needed by real data. Mapping the example shown in Figure to this database model, and including some data for clarity, we have the following data relations:

DistrictsG

geomId: int	objectId: string	spatialData: spatial
1	261160605001	*bbbbbbbbbbbbbbbbb*
2	261160605010	*bbbbbbbbbbbbbbbbb*
3	261160605011	*bbbbbbbbbbbbbbbbb*

DistrictsA

ID_: string	name:string	POP_2004:int
261160605001	Recife	98361
261160605010	Cabanga	102548
261160605011	Sao Jose	1524

The metadata relations would contains the following items:

layers

layerId: int	layerName: string
1	*Recife*

representations

layerId: int	geomRelation: string
1	*DistrictsG*

attributesRel

layerId: int	attrRelation: string
1	*DistrictsA*

Temporal layers

In this section, we consider how to extend the static model to deal with temporal data. Suppose that we want to keep track of changes in each district, to follow the evolution of its population and its boundaries. We want to be able to register all of these changes in the same database and to extract information about the spatial and temporal changes on the data. Since we consider that an *ST-object* preserves its identity over time, every change in its attributes or its geometries produces a new instance of this object. Since changes in attributes and geometries might be asynchronous, our generic database model needs two adjustments. First, we have to introduce a unique identifier in every attribute relation. This identifier allows the distinction of different instances of attributes to the same object. The second adjustment is including one more relation, a *status* relation. This relation describes which instances of geometries and attributes are valid in a given interval:

status(geomId: int, uniqueId: string, initialTime: time, finalTime: time)

The status relation maps every instance of geometry (identified by the field geomId) to an instance of attributes values (identified by the field uniqueId). Each mapping has an associated valid interval (identified by the field initialTime and finalTime). As is possible that a layer has more than one attribute and geometry relations, there should be one status relation to each combination of a geometry relation with an attribute relation. Returning to our example of the districts of Recife, and showing some data, we would have the following data relations:

DistrictsG

geomId: int	objectId: string	spatialData: spatial
1	261160605001	*bbbbbbbbbbbbbbbbb*
2	261160605010	*bbbbbbbbbbbbbbbbb*
3	261160605011	*bbbbbbbbbbbbbbbbb*
4	261160605001	*bbbbbbb*
5	261160605010	*bbb*

DistrictsA

ID_: string	name:string	POP_2004:int	unique_id: string
261160605001	Recife	98361	1
261160605010	Cabanga	102548	2
261160605011	Sao Jose	1524	3
261160605011	Sao Jose	2789	4
261160605010	Cabanga	106548	5

DistrictsStatus

attributeInst:string	geomInst:string	timeI:time	timeF: time
1	1	01/01/2003	31/12/2003
2	2	01/01/2003	31/12/2003
3	3	01/01/2003	31/12/2003
1	4	31/12/2003	
4	3	31/12/2003	
5	5	31/12/2003	

The status relation tracks both synchronous and asynchronous changes in the geometries or attribute values of the *ST-Objects*. It also allows the retrieval of individual *ST-Instances* of an *ST-Object*.

Examples of Spatio-Temporal Data

This section describes four sets of the spatio-temporal data that are representative of real world problems and demands. They are semantically different, and explain the requirements for the query processor and the expressive power of our generic database model. These three data sets are:

Crime events:

The object is the crime occurrence. Its geometry is a point representing the location of the event. Each new event has a unique identifier, therefore, is a new object. The main characteristic of this data set is that for each object there is only one spatio-temporal instance.

Land parcels:

This data is an experimental data that reflects the changes happening in land parcels of a neighborhood or city. Parcels are divided, sold or reacquired over time. This means that they can suffer changes in its geometry (for example, when the parcel is divided) or in their attributes (for example, when the parcel is sold). The object is the parcel, and each change in their attribute values or geometries generates a new instance of the parcel.

Satellite tracking animals:

This data results from a research project of surveying free ranging animals by radio transmitters. The transmitters are installed in collars attached to the animals and the signal is picked up by satellites. The object is the animal and each new signal detected is a new instance of the geometry of the object (or the animal location). This data represents the typical case of moving objects.

The Query Processor

A spatio-temporal database can be queried in different ways, according to the applications. For example, *"for each month, which changes occurred in the parcels?"*, *"Which crimes happened on Friday in the south zone of Recife?"*

To answer this demand, we have developed a flexible query processor, named Querier, able to deal with different applications of spatio-temporal data and their needs for queries and responses. The Querier is responsible for the link between applications and spatio-temporal databases, as shown in Figure 9.

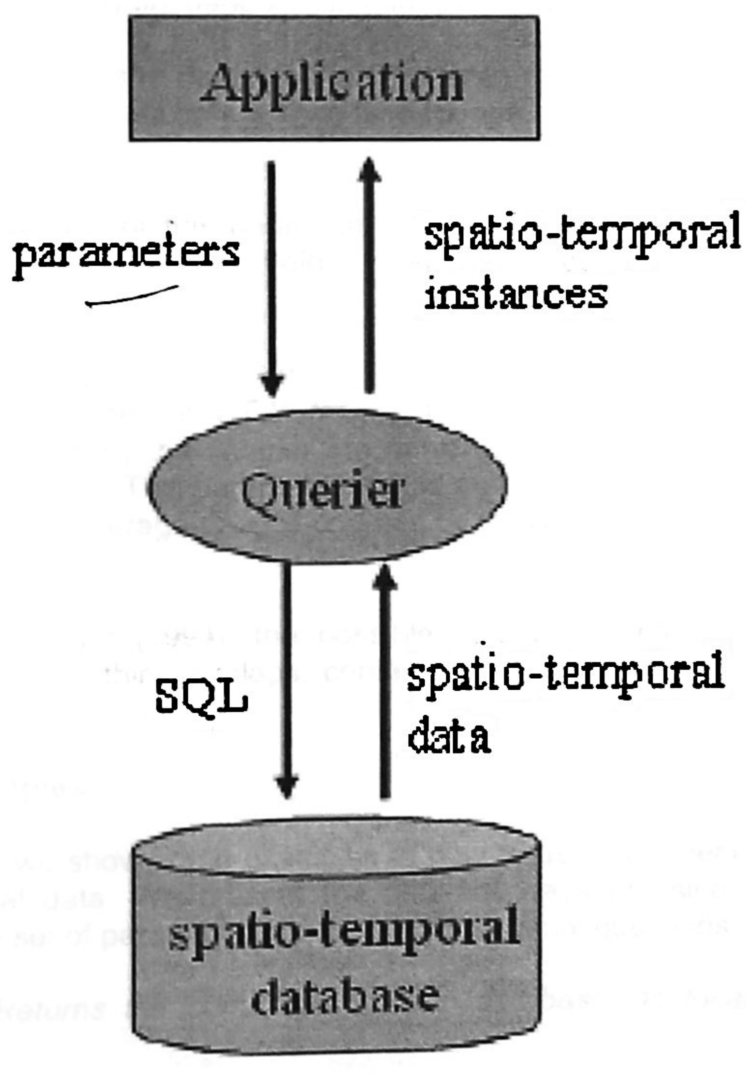


Figure 9. The Querier architecture.

The Querier receives a set of parameters that define its behavior. These parameters control how applications retrieve *ST-Instances* and *ST-Objects*.

Their combination provides a flexible way of querying a spatio-temporal database:

Layer: The source of the *ST-Instances* recovered by the Querier.

Chronon: Defines how the *ST-Instances* are split in time frames. The possible values of chronon are: second, minute, hour, day, month, year, day of week, day of month, day of year, month of year, week of year, hour of day, minute of hour or second of minute. For example, if you want to watch the crime events separated by weekday, the Querier returns seven time frames, one for each weekday. In this case, all the crimes that happened on Sunday are in the first time frame, the ones that happened on Monday are in the second time frame, and so on.

Which time frames will be retrieved:
Controls whether the Querier returns either every time frame existent in the interval or only the time frames where some change occurred. Suppose a user is querying a set of parcels in 2004 and Chronon month. Changes on the parcels occurred only in January and in September of 2004. If the user requests the Querier to return every time frame, she will get twelve time frames. Otherwise, if she requests the Querier to return time frames where a change occurred, she will only get two time frames.

Temporal predicate:
The possible values for this parameter are the temporal interval predicates defined by Allen [1983]: equals, before, after, meets, during, overlaps, ends or starts.

Aggregate functions:
Allows grouping a set of *ST-Instances* to produce a single value. The functions supported by the Querier are: maximum, minimum, average, sum, counting, or variance. This parameter should be used, for example, when one needs to know the average of eggs counted by trap is in each month.

Spatial predicate:
Based on Egenhofer [1994], the possible spatial predicates are: disjoint, touches, crosses, within, overlaps, contains, intersects, equals, covers, and covered by.

Querier Examples

In this section we show some examples of how to use the Querier to retrieve spatio-temporal data. We present the different ways of using the Querier combining the set of parameters to answer some usual questions.

Example 1) Returns the crime events from database, its location and its properties.

Step 1. Retrieve the layer information

```
CrimeLayer = database->getLayer ("Crimes");
```

Step 2. Set the parameters

```
Querier->setParams(loadGeometries, loadAllAttributes);
Querier->setParams(CrimeLayer);
```

Step 3. Load the instances

```
Querier->loadInstances();
```

Step 4. Consume the returned instance (ST-instances).

```
while (Querier->fetchInstance(sti))
{
    Geometry = sti.getGeometry();
    Properties = sti.getProperties();
    Time = sti.getTime();
}
```

In this example, the Querier returns the crimes from the layer "Crimes", its location (point geometry) and its attributes or properties. The first step loads the layer information from the database. The Querier behavior is defined by the parameters set in the step 2. Step 3 loads the instances and, in the final step, the Querier traverses every crime event and, for each one, gets its location (geometry), properties and time. *Example 2) Which crimes happened in 2003 in the district named "Cabanga" of Recife city?*

Step 1. Retrieve the layer information

```
CrimeLayer = database->getLayer ("Crimes");
```

Step 2. Get the geometry from the layer that will be used in the spatial predicate

```
DistrictLayer = database->getLayer ("Districts");
DistrictGeometry= DistrictLayer->getGeometry ("Cabanga");
```

Step 3. Set the parameters

```
Querier->setParams(loadGeometries, loadAllAttributes);
Querier->setParams(CrimeLayer, year, ChangedTimeFrames);
```

```
        Querier->setSpatialRestriction(DistrictGeometry, within);
Step 4. Load the instances of a specific time frame
        TimeFrame = Querier->getTimeFrame('2003', Year);
        Querier->loadInstances(TimeFrame);
Step 5. Consume the returned instance (ST-instances).
        while (Querier->fetchInstance(sti))
        {
            Geometry = sti.getGeometry();
            Properties = sti.getProperties();
            Time = sti.getTime();
        }
```

9. Integrating Space and Time in GIS

A spatio-temporal GIS extends a customary two or three dimensional spatial GIS on the time axis. A space-time path is defined by the changes of locations over time (figure 1). In a spatio-temporal framework space and time are equivalent features of an event or a process. Furthermore each entity which physically exists in the real world is defined in a spatio-temporal context which can be conceptualized using spatial, temporal and attribute information.

Every spatial object used in a GIS has a temporal validity as well as one or many attribute values. The entity of a spatio-temporal process may change its spatial representation over time as well as its spatial relationship to other entities. In addition, the related attribute information may be subject to changes throughout time.

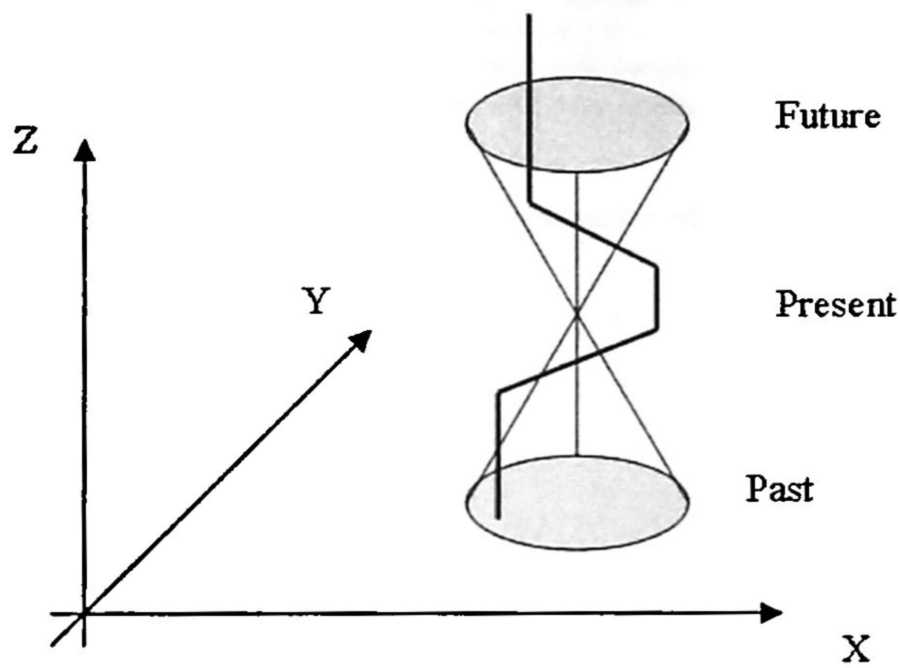


Fig 10: Space and Time path

"the concept of time implies that changes occur throughout the present, the past and the future of the life span of a real-world phenomenon. A temporal GIS will aim to undrestand these changes and their affects over time rather than simply reproducing them by displaying a sequence of snapshots"

Spatio-Temporal database features

There is a critical set of capabilities that are needed by moving objects database applications and Spatio-temporal database. Existing DBMS's are not well equipped to handle continuously changing data and they shoud be developed to be able to handle these data. Following are the features for database, that are use to manage spatio-temporal data:

- Elementary features (basic data types, SQL language features, declarative integrity constraints, programming abstractions, automatic generation of identifiers, national characters support).
- Transactions and multi-user access (transactions, locks, multi-user access).
- Programming in database (stored procedures and triggers).
- Elements of database administration (access control, backup copies, data migration).
- Portability and scalability
- Performance and VLDB (Very Large Database) issues (query optimization, structures supporting query optimization, support for analytical processing, allocation of disk space, data size limits, known VLDB implementations)
- DB2 Distributed databases (access to multiple databases, heterogeneous systems support)
- Special data types (large objects in database, post-relational extensions, support for special data types)
- Application development and interfaces (embedded SQL, standard interfaces, additional interfaces, interoperability with Web technology, XML, CASE)
- Reliability (failure recovery)
- Commercial issues (technical support available, market position)
- Spatial vision

10. Applications Of Spatio-Temporal Database Management Systems

Examples of application domains dealing with Spatio-temporal data are:

- **Financial Applications (e. g. history of stock market data)**
- **Insurance Applications (e. g. when were the policies in effect)**
- **Reservation Systems (e. g. when is which room in a hotel booked)**
- **Medical Information Management Systems (e. g. patient records)**
- **Decision Support Systems (e. g. planning future contigencies)**
- **Weather Forecasting System**
- **Various Geographical Information Systems.**

1 . Disaster Management

Effective disaster management requires getting the right information (often geographically related) to the right place at the right time. Minimising response times to incidents is therefore critical. Ordnance Survey's Mapping for Emergencies unit addresses this requirement through out-of-hours incident support. Additionally, a Pan-government agreement now gives a wide range of central government organisations direct access to a suite of Ordnance Survey's products, allowing better preparedness in the planning phases of disaster management.
Recent and anticipated (often disruptive) developments in spatial databases, GPS, wireless, mobile and computing technologies have changed, and will continue to change, the way in which geographic information (GI) can be collected, maintained, analysed, integrated and delivered to the end user in disaster management and other domains. GI is increasingly part of the information mainstream. These developments have changed the role of Ordnance Survey from being the nation's map maker to being the geographic information provider to the nation, with a substantial role in developing a geographic framework in which both geographic and related information can be efficiently integrated, exchanged and understood. These developments prompt a number of research challenges within the domain of disaster management. Three are considered and illustrated:

• The application of user-centred design techniques to user requirements and behaviours in order to identify where GI adds value in contributing to task efficiency
• Considerations of database modelling in 3 and 4 dimensions
• Exploring GI portrayal with new technology

In meeting the GI needs of the disaster management domain. It considers:
• The nature and scope of disaster management
• Ordnance Survey's historic support in meeting mapping and data requirements for emergency response and crisis management
• Recent developments in providing government with direct access to GI
• The development of the Digital National Framework™ (DNF®) as a consistent framework within which to use and link GI and related information

According to the Cabinet Office Civil Contingencies Secretariat (CCS) (www.ukresilience.info/handling.htm) the management of disasters, whether man-made or natural, involves:

- Risk assessment
- Risk prevention or mitigation
- Preparation including contingency planning, training, and exercising
- Emergency response and recovery

Role of a National GI Provider in Meeting Requirements for Disaster Management

Since everything happens somewhere, what, then, is the role of a national GI provider in meeting the above requirements for information within the domain of disaster management?

Access to National GI

One role is to provide better access to national GI by ensuring government departments and agencies have better access to Ordnance Survey products and services directly. In support of this, a Pan-government agreement launched in 2003 gives over 500 British government departments and agencies access to a wide range of digital map data. This is a significant step towards ensuring that common standards are consistently used for decision making.

If the requirements of disaster management actors and decision makers are: to have the right information communicated in the right way at the right time, what, then, are the research challenges of a national GI provider in trying to meet these requirements?

Over the last 10–15 years technological advances offered by GPS, spatial databases, the Internet, wireless and mobile devices have revolutionized the way location-based data is collected, stored, maintained, analysed and delivered to the user. The map is now just one expression of the spatial database. These advances have brought GI to the information mainstream and are enabling its increasingly pervasive use as a fundamental ingredient to effective decision making and task maagement. They have enabled a change in Ordnance Survey's role from that of the nation's map maker to that of fundamental provider of the nation's geographic reference framework, whilst maintaining its role in providing national map coverage.

These developments are enabling a move from a product centric to a database centric business, from which products and services can be derived.

Technology will enable virtually everything (objects and people) to be identified, tagged, sensed and monitored. Miniaturisation will allow processing on the device and geographic intelligence to be held on the network.

Whilst increased bandwidth and processing power will allow complex distributed analyses across trusted super computing environments.

In a spatial database GI is no longer confined to a 2-D planimetric representation of the world. Increasingly, applications – such as building information modeling (BIM), which joins up data in the architectural, engineering and construction domain, and contingency planning applications, for example – will require true 3-D data models, or the means to derive them. For many applications, including LBS, tasks and decisions are bounded by

events and, increasingly, the temporal (4th dimensional) aspects of GI will be required.
Disasters are dynamic events (or a series of dynamic events) that occur at particular locations over time. The management of disasters also requires actions at particular locations over time and therefore the incorporation of time as the fourth dimension in GI is important for effective disaster management.
A spatio-temporal data model should facilitate:

- An understanding of the rules that govern real-world change
- Explanation of the state of the world at the current time (now) and previous times
- Predictions of the state of the world in the future.

Cartography

Nowadays, many applications need data modeling facilities for the description of complex objects with spatial and/or temporal facilities. Responses to such requirements may be found in Geographic Information Systems (GIS)

Cartography is one of the major application areas using geographical databases. Whether it is for the business of producing paper maps for sale, or whether it is for displaying maps on a screen to visualize the result of a query, we need computer systems that know how to represent the same geographical area at different scales. The concept of multiscale database has become popular in the GIS domain as a way to enforce consistency between representations and reduce the global update load. Scaling, however, is just one of the facets that may lead to keeping several representations for the same real-world object. Viewpoint and classification are two major abstractions in the design process that also generate multiple representations. This dissretation investigates the generic issues and solutions to achieve flexible support of multiple representation in a GIS database

A Database for Bio-molecular Images

Information technology research has played a significant role in the genomics revolution over the past decade, from aiding with large-scale sequence assembly to automating gene identification to efficiently searching databases by sequence similarity. The tremendous amount of information gathered from genomics will be dwarfed in the next decade by the knowledge to be gained from comprehensive, systematic studies of the properties and behaviors of all proteins and other biomolecules. High resolution imaging of molecules and cells will be critical for understanding complex systems such as the nervous system, whether it be for the localization of specific neuron types within a region of the central nervous system, the branching pattern of dendritic trees, or the localization of molecules at the subcellular level. Furthermore, knowing how these distribution patterns and subcellular locations change as a function of time is critical to understanding how cells respond to stress, injury, aging, and disease. We are developing sophisticated information technologies for collecting and interpreting the enormous volume of biological image data. A major outcome of the research will be a unique, fully operational, distributed digital library of biomolecular image data accessible to researchers around the world. Such searchable databases will make it possible to optimally understand and interpret the data, leading to a more complete and integrated understanding of cellular structure, function and regulation.

Biomolecular images have a high processing and storage cost. A 2D protein localization image from confocal microscopy can require 4MB (1M pixels recorded in two channels) of storage. A 3D localization image can be 200MB (50 z-slices). A time series of 50 such 3D localizations that record dynamic information can be 10 GB. This is the result of acquisition from a single sample, and typical experiments involve dozens of samples for different proteins or under different conditions. AFM images require even larger amounts of storage capacity. But it is not just the needed storage that makes the problem of designing bioimage databases daunting. The images have to be analyzed, and visual descriptions extracted using image processing tools (manually or automatically), and these extracted metadata have to be associated with other sources of biological data such as genomics and proteomics . This analysis can lead to a multifold increase in the amount of storage and complexity. Clearly, the amount of information to be maintained and accessed in such a bioimage database is enormous.
Queries in a bioimage database can be divided into four classes based on the degree of semantics and interpretation.

1 . Metadata queries 2. Spatial Queries

3.Semantic Queries 4. Spatio-temporal queries

1. Metadata queries

Typical metadata fields from the experiments will be date, scientist, lab, experimental setup, microscope, light sources, filters, camera, experiment, species, antibodies for each channel, and experimental conditions (e.g., normal retina, retina detached for N days, retina reattached for N days, retina under increased oxygen concentration). Some specific queries in this class are as follows.

- Find all images from the same experiment as a given image ID.
- Find all experiments that contain normal cat images that have been labeled with calretinin (a calciumbinding protein) both under normal conditions and after 3 days of retinal detachment.

2. Spatial queries

Simple spatial features based on texture and shape can be extracted from the images. This can be done at multiple spatial resolutions to provide more flexibility for querying and browsing. The most important task will be to define the right metrics for comparing images based on the extracted features, especially since the images will be produced under different experimental conditions and will be of different subjects. The distance metrics will also need to be supplemented with a statistical model that defines the distribution of the distances. Finally, the extracted features will be highdimensional and one is faced with the usual challenges of content-based search in such spaces.
Some typical queries in this class are as follows:

- Find all images in which vimentin (a filament protein) has a spatial distribution similar to that in a given image.
- Find all pairs of images from the same experiment in which the distribution of vimentin changes as a result of detachment.

3. Semantic queries

Queries in this class are based on semantics extracted from the images. Typical examples of such semantics are the types of cells, their shapes, and their relative location. Semantics can be extracted manually or automatically. This process will be eased through an atlas that define the expected distribution of cells under different experimental conditions.Some examples of queries in this class are as follows:

• Find all normal retinal cell images that contain horizontal cells.

4. Spatiotemporal queries

Spatio-temporal queries consider the time-based evolution of cells and disease processes. Supporting such queries in a meaningful manner requires the extraction of appropriate temporal information from a set of images.
The system should provide tools for the modeling of cell behaviors, changes in protein localizations, and disease processes. Queries will typically examine correlations between sets of images or across images and cell/disease models.
The temporal aspect can be observed either by conducting an experiment at different time intervals (e.g., studying retinal images detached for different lengths of time), or by directly observing a change (e.g., movementof a microtubule). In the latter case, temporal features will be useful for an individual microtubule and also for groups of microtubules in order to understand their collective behavior. Some examples of spatio-temporal queries are as follows:

• Find all image datasets in which the change of vimentin within Muller cells is similar to that observed in the change of GFAP.
• Find patterns of apoptotic cell death within cell populations for a given set of images, and then search for similar patterns

Tools Used in spatio-temporal databases

A number of tools have been built to query the spatio-temporal database and display textual or graphical results. The tools communicate indirectly with the database server using a simple client application which passes queries to the server and returns results in a readily interpreted form. Most have been constructed using Tcl/Tk (Ousterhout, 1994). Tcl is a simple scripting language and Tk is a toolkit extension for building graphical user interfaces. The versions used here run under UNIX with Tk as an X Window System toolkit, but recent releases of Tcl/Tk are also available for Mac and PC platforms.

Fig 11 : (45kB)

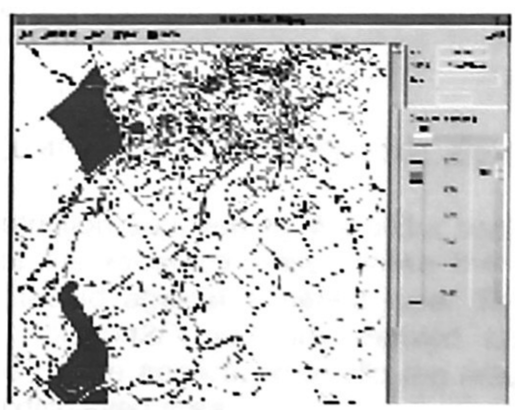


Figure 11 shows the main spatio-temporal query and map display tool. This functions in a similar way to many GIS display modules in that it allows different collections of geometric information to be retrieved and overlaid to form composite maps. If required, a composite result can be saved as a new MapLayer object in the database. Objects to be displayed are selected by composing and executing SQL queries which can be entered 'by hand' in a pop-up window. However, the interface also incorporates a number of aids to query composition so that part or all of the query can be generated using menu options or other interface elements. In due course, these capabilities may be extended to provide a more complete visual query composition tool.

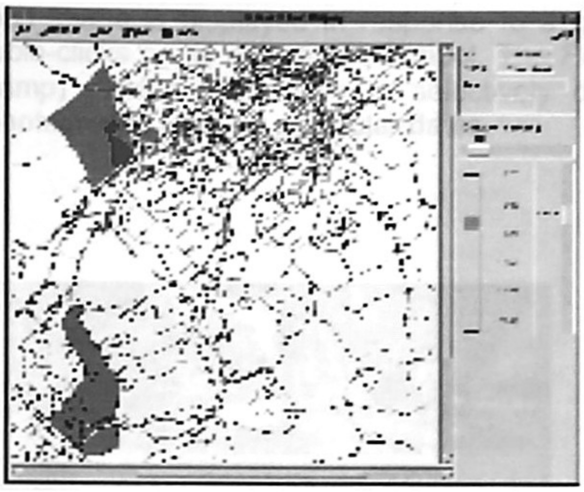
The temporal range of each query is set using the two slider controls to the right of the map window. These set the starting date and duration of the period of interest. The particular object class to which the temporal restriction is applied is chosen from a list of classes in the 'Database' menu. Figure 2 shows the display after several distinct queries have been executed. Initially, vector point and line date were retrieved for three MapLayer objects corresponding to the 1884 and 1953 cadastral maps, and some recent fieldwork observations. These queries were of the same form as the example discussed earlier.

The temporal constraints were then set to cover the period 1884-1893 and a further query executed to retrieve vector polygons corresponding to land parcels owned by one Jean Come Pietri at this time. Here, the temporal constraint was applied to *Propown*, a class that includes a valid time element and links *Owner* with *Property*. *Owner* contains a subset of all persons in the database, and *Property* contains groupings of land parcels that represent single 'units of ownership'. This class has data members holding details such as location, area and class of land, and is linked to the *Parcel* class, itself a derivative of *SpatialApplicationObject*. This final query was as follows:

```
select g from (
   select unique pa.geometry
   from Parcel pa, Property pr, Propown po, Owner o
   where o.owner = po.owner
   and po.prop = pr.prop
   and pr.prop = pa.prop
   and Overlap(po.valid, '1884~1893'::period)
   and o.snom = 'Pietri'
   and o.names = 'Jean Come'
) t, Polygon2D g
where t = g.oid
and Overlap(g, '(510700,4622800,512200,4624300)'::Box);
```

The temporal overlap restriction supplied by the slider controls is applied to the valid time element of the *Propown* class. Unlike the earlier *MapLayer* example, the overlap is tested against a period type. The query result is limited to instances of *Polygon2D* and any derived classes. A spatial restriction is also supplied by the application to clip the result to include only those polygons within the displayed area.

Fig 12:



Where necessary, further temporal constraints may be added by manually editing the query. In figure 12 the main period of interest has been changed to 1910-1919, and separate queries, similar to that above, have been used to retrieve the land owned by three of Jean Come's siblings. These queries were edited to add further constraints to select only the land that had been owned by Jean Come in 1884. Line and fill colours may be selected from a menu, or set to correspond to a column returned by the query. Here, the fill colour was changed using the menu before each new query was executed.

Separate queries were used here to show the land owned by each individual sibling. However, we anticipate that future versions of the application will include ways of using genealogical and inheritance information stored in the database to provide more direct access to groups of individuals and their properties.

As well as drawing maps in response to queries, the application provides the user with constant feedback including the current cursor position in world coordinates and the type and OID of any *GeometricObject* under the cursor. As the cursor is moved these objects are highlighted. Double-clicking on an object is used to retrieve details of any associated *SpatialApplicationObject*. In most cases, this information is in alphanumeric form and is displayed in the query window, but this mechanism can also be exploited to retrieve graphical or other representations. Figure 13 shows a general view of the application with several of its component windows, including four photographic images that have been retrieved in this way. Spatially and temporally located photographic information represents a particularly important source for anthropological research.

Four symbols (disks with direction of view arrows) can be seen on the map, two at the south-west end of the village and two near the north east corner of the window. These represent *OrientedPoint2D* objects, a sub-class of

GeometricObject. A *Photograph* class derived from *SpatialApplicationObject* holds details of all photographic images stored in the database. Instances of this class are linked to *OrientedPoint2D* objects to provide their geometric representation. Each image is displayed in response to a query generated when the user double-clicks on its associated symbol. The *Photograph* class includes a timestamp, making it possible to selectively display symbols corresponding to photographs taken at particular dates.

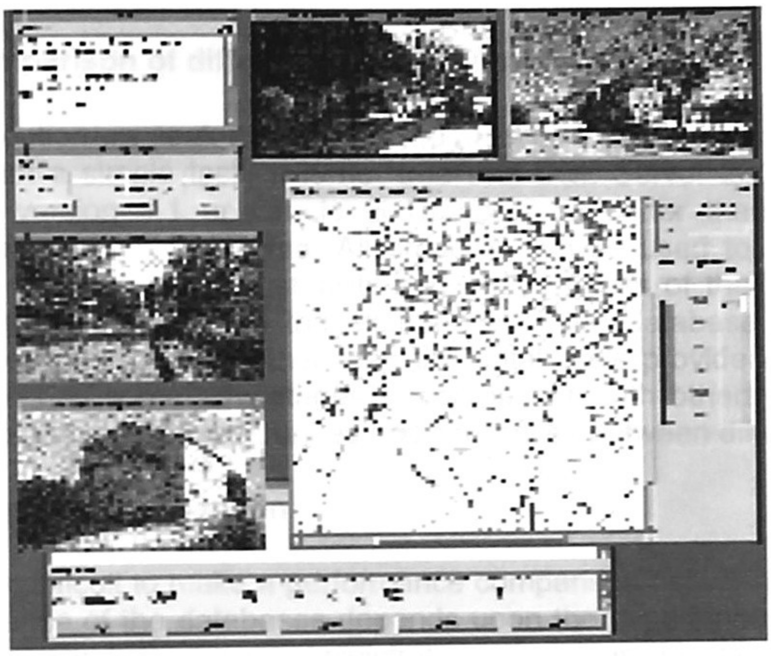


Fig 13:

11. Database Softwares

With consider to above mentioned and based on the capabilities that are needed by moving objects database applications, we choose some existance DBMS's and assessed them to find best DBMS to extend for design and development a moving object database.

DBMS such as Oracle that here , mean Oracle 9i and DB2 universal database that is short form of database 2 universal database and is an IBM production. SQL server version 2000 from Microsoft. And at last MySQL and PostgreSQL as Open source databases.

The comparison of different database software

In this section we seek to compare products of a similar nature. Unfortunately this is not a simple task. It is not true that SQL Server 2000 is better than MySQL version 4.1 or Oracle 9i Database is better than DB2 Universal Database v8.1 or vice versa. All products can be used to build stable and efficient systems and the stability and effectiveness of the applications and databases depends upon the experience of the database developers and database administrator rather than the database's provider. However, each database has some advantages in comparison with others and vise versa. Here are some of the differences and similarities between different databases:

Performance comparison

It is very difficult to make a performance comparison between databases. The performance of the databases depends upon the experience of the database developers and database administrator rather than the database's provider.

Platform comparison

All of Oracle 9i Database, DB2 Universal Database version 8.1, MySQL version 4.1 and PostgreSQL support all known platforms, including Windows-based platforms, AIX-Based Systems, HP-UX systems, Linux Intel, Sun Solaris and so on. SQL Server 2000 only works on Windows-based platforms, including Windows 9x, Windows NT, Windows 2000 and Windows CE.

Features comparison

All of Oracle 9i Database, IBM DB2 Universal Database v8.1, MySQL version 4.1 and SQL Server 2000 support the ANSI SQL-92 entry level and do not support the ANSI SQL-92 intermediate level.
In general, Oracle offers all the features required for information integration in a single unified product. The inherent reliability and security features of the Oracle database are automatically inherited by information integration applications. Oracle is clearly the leader in the distributed space with its single, unified solution to satisfy a complete spectrum of information integration requirements. All editions of Oracle include; the PL/SQL engine to develop stored procedures, triggers, and functions; a Java compiler and Virtual Machine (JVM) to develop Java stored procedures and triggers; XML support; an Apache Web server; and object-relational capabilities.Oracle9i Database offers a rich set of development and administration tools, and

recent releases have focused on security and scalability improvements. The implementation of various index types demonstrates how extensively Oracle9i Database focuses on fast data retrieval, especially in large databases such as data warehouses. In addition, the use of reverse key and function-based indexes, allows for a great degree in flexibility in quickly accessing data. In recognizing the high availability challenges every business faces, Oracle provides comprehensive, unique, powerful and simple to use capabilities to protect against most forms of unplanned downtime, including system faults, data corruption, human error and disaster. And, achieves this in an environment where the need for planned downtime is marginalized.
IBM on the other hand has a completely different model for information integration. They offer different products for different integration scenarios. This adds several layers of complexity, since each product has to be installed and configured separately.
Oracle and DB2 can scale to terabytes of data storage fairly easily. MySQL and PostgreSQL are known to run well into the hundreds of gigabytes. PostgreSQL and MySQL boast widespread use for relatively small databases (under 100GB, for example).
DB2 offers the very basic set of backup and recovery capabilities but lacks the completeness and depth of data protection required by most businesses today. In fact, DB2 is behind even Microsoft SQL Server when it comes to high availability.
The dialect of SQL supported by Oracle 9i Database is called PL/SQL. The dialect of SQL supported by IBM v8.1 is called DB2 SQL dialect. The dialect of SQL supported by Microsoft SQL Server 2000 is called Transact-SQL (T-SQL). The dialect of SQL supported by MySQL version 4.1 is called MySQL dialect. Transact-SQL dialect is a more powerful language than MySQL dialect.

Spatial } Extensions
Temporal }

Spatial Extender

In 1997, the Open GIS Consortium published the OpenGIS Simple Features Specifications for SQL, a document that proposes several conventional ways for extending an SQL RDBMS to support spatial data.

Use DB2 Spatial Extender to generate and analyze spatial information about geographic features and to store and manage the data on which this information is based. The DB2 Spatial Extender allows the user to include spatial attributes (distances, times, and geographical information) into business analysis. It confirms to ISO and OpenGIS Consortium (OGC) standards. Additionally the latest version of the spatial extenders includes an Index Advisor for tuning spatial indexes, includes additional spatial functions, and the ability to export SDE transfer data and spatial data to geobrowsers.
MySQL implements spatial extensions following the specification of the Open GIS Consortium (OGC).
A major feature of Oracle 9i is Oracle Spatial. This is a spatial extender that provides storage, indexing and proximity queries for location-based information, which may include road networks, wireless service boundaries and geo-coded customer addresses, for example. Oracle Spatial makes it possible to combine the relational power of a database with spatial data. The ability to use indexes, various queries, and functions means complex spatial

calculations may be pushed back onto large database servers. As mobile applications and technologies increase, so will the demands to store and analyze spatial data in a transactional setting.

Security
Database security is a very important aspect of any relational database management system to protect access to the database operations and the data.
At first glance, Oracle and IBM appear to offer similar security solutions, but with closer inspection, it is plain to see that the two companies approach security differently and ship solutions at vastly different levels of maturity. Independent evaluations and feature-for-feature comparisons prove that the Oracle9i Database is more secure than IBM's DB2 Universal Database. Overwhelming evidence supporting this assertion that Oracle security is far superior to DB2 security.

It is difficult to make up for a lack of security built into the core DB2 product set, but IBM offers a variety of packaged service plans to do so. Oracle's security solutions are much less expensive than IBM's because customers do not have to pay for additional software and services. IBM's security solutions are less secure than Oracle's because they rely on external solution and services to implement security they've neglected to build into DB2, which does not provide equivalently robust, mature security features that Oracle has been shipping for years.

The following table summarizes the database comparison.

Table Comparison of some existing DBMS's

Features	Oracle	DB2	MS SQL server	MySQL	PostgreSQL
Data types	Very good	Good	Good	Good	Very good
SQL language features	Very good	Very good	Very good	Average	Good
Declarative integrity constraints	Very good	Very good	Very good	Average	Very good
Primary key	Yes	Yes	Yes	Yes	Yes
Foreign key	Yes	Yes	Yes	No	Yes
Programming abstractions	Very good	Good	Good	Poor	Average
National characters support	Very good	Very good	Good	Good	Good
Transactions	Very good	Very good	Very good	Poor	Good
Locks	Very good	Good	Good	Poor	Good
Multiuser access	Very good	Good	Average	Average	Average
Stored procedures and triggers	Very good	Very good	Good	Poor	Very good
Access control	Very good	Very good	Good	Very good	Good
Backup copies	Very good	Very good	Good	Average	Average
Portability of DBMS	Very good	Good	Poor	Good	Good
Scalability	Very good	Very Good	Good	Good	Average
Query optimization	Very good	Very good	Good	Good	Good
Structures supporting query optimization	Very good	Good	Good	Poor	Good
Support for analytical processing	Very good	Very good	Good	Poor	Poor
Allocation of the disk space	Very good	Good	Good	Average	Average
Data size limits	Very good	Very good	Very good	Good	Average
VLDB (very large data base)	Very good	Very good	Good	Poor	Good

implementations					
Access to multiple databases	Very good	Very good	poor	Average	Average
Heterogeneous systems support	Good	Good	Poor	Poor	Poor
Large objects in database	Very good	Very Good	Good	Good	Average
User-defined data types	Yes	Yes	No	No	Yes
Object-relational extensions	Yes	Yes	No	No	Yes
Support for special data types	Very good	Very good	Good	Poor	Average
Complex Applications	Very good	Very good	Average	Poor	Good
Embedded SQL	Very good	Very good	Good	Poor	Good
Standard interfaces (ODBC, JDBC)	Very good	Very good	Good	Good	Good
Interoperability with Web technology	Very good	Very good	Good	Good	Good
XML	Very good	Very good	Very good	Poor	Poor
Recovery from failures	Very good	Very good	Good	Average	Average
Security	Very good	Good	Good	Average	Average
Ease of Use	Average	Poor	Very good	Good	Good
Mobile-commerce	Very good	Very good	Good	Average	Poor
Spatial extender	Very good	Very good	Good	Average	Poor
Prices	High	Average	Low	Free	Free
Technical support from manufacturer (Availability)	Very Good	Good	Good	Average	Poor
Position on the market	Very good	Very good	Good	Poor	Poor

12. Conclusion

We saw the limitations of conventional DBMS. No. of new applications are emerged and needs of some new complex datatypes are arise. That complex datatypes are not supported by the conventional DBMS. Hence there is a need to extend the current DBMS. Spatio-Temporal database Management System is an extended DBMS which plays an important role in many sectors. So a detail study of Spatio-Temporal Database Management Systems is the need for todays changing trends. Different database softwares are suggested as above which supports the spatio-temporal features and comparision .

From the above presented comparison, the following conclusions can be made:

- SQL Server 2000 holds the top TPC-C performance and price/performance results.
- SQL Server 2000 is generally accepted as easier to install, use and manage.
- MySQL version 4.1 requires little hardware resources.
- You can use MySQL version 4.1 without any payment under the terms of the GNU General Public License.
- Oracle 9i Database, DB2 Universal Database v8.1 and MySQL version 4.1 supports all known platforms, not only the Windows-based platforms.
- PL/SQL is the most powerful language
- Oracle9i Database offers a rich set of development and administration tools
- Oracle and DB2 can scale to terabytes of data storage fairly easily
- When it comes to backups, open-source databases like MySQL and PostgreSQL may not completely fulfill your needs.
- Oracle Spatial makes it possible to combine the relational power of a database with spatial data

With consider to above mentioned specifications of different commercial database management systems and based on the capabilities that are needed by moving objects database applications, it is recommend that Oracle is the first choice for managing spatio-temporal data. DB2 universal database and SQL server are the next alternatives. Also, Spatio-temporal database management system plays an important role in GIS.

13. GIS Softwares

Software	Software Maker/Distributor	Description
ArcCad	ESRI www.esri.com	PLATFORM : Windows (95, NT) "ESRI's ArcCAD software gives you full geographic information system (GIS) functionality within AutoCAD. ArcCAD provides powerful mapping, data management, and display tools that work directly with AutoCAD software's design and drafting tools. ArcCAD users can create, access, and exchange ARC/INFO data, the de facto standard for local, state, and national GIS installations."
ArcExplorer	ESRI www.esri.com	PLATFORM : Windows "ArcExplorer software is a lightweight GIS data explorer. ArcExplorer can work on its own with local data sets or as a client to Internet data and map servers. ArcExplorer is built using MapObjects, ESRI's powerful collection of GIS and mapping components for application developers. As a stand-alone application, ArcExplorer is a complete data explorer, allowing users to display and query a wide variety of standard data sources."
ArcFM	ESRI www.esri.com	PLATFORM : Windows (NT) "Arc Facilities Manager (ArcFM) is a powerful ARC/INFO-based application for the editing, maintenance, modeling, and data management of utility information. ArcFM is all-relational utility application software developed using

		ARC/INFO software's Open Development Environment and object-oriented programming."
Arc/Info	ESRI www.esri.com	PLATFORM : Most Unix platforms, Windows (NT) "Professional GIS offers advanced modeling and analysis capabilities"
ArcView Business Analyst	ESRI www.esri.com	PLATFORM : Windows (3.51, 95, 98, NT) "ArcView Business Analyst allows you to focus on getting the data analysis you need, rather than worrying about the details of the underlying technology. A wizard interface assists you with tasks ranging from basic, such as mapping customers, to complex, such as performing analyses or creating what-if scenarios."
ArcView GIS	ESRI www.esri.com	PLATFORM : Most Unix platforms, Windows (95, 98, NT) "The leading mapping and GIS software for the desktop."
Atlas GIS	ESRI www.esri.com	PLATFORM : Windows (95, 98, NT) "Business professionals worldwide rely on Atlas GIS to help them make decisions that improve the efficiency and profitability of their organizations. Atlas GIS is both powerful enough to perform sophisticated desktop mapping and intuitive enough that nontechnical professionals feel comfortable using it."
AutoCAD Map 2000	Autodesk www.autodesk.com	PLATFORM : Windows (95, 98, NT) "AutoCAD Map® 2000 is the Autodesk® software optimized for precision mapping and geographic information system (GIS) analysis in the AutoCAD® environment. It provides the

		GIS analysis tools and productivity improvements you need to easily create, maintain, and produce maps and geographic information.”
BusinessMAP PRO	ESRI www.esri.com	PLATFORM : Windows (3.1x, 95, NT) “BusinessMAP PRO is an affordable mapping application that transforms information from databases, contact managers, spreadsheets, and electronic phone books to pushpin and color-filled maps. It will turn static data into meaningful images to help you understand your business better and help with sales automation and mission-critical decision making processes.”
Data Automation Kit	ESRI www.esri.com	PLATFORM : Windows (3.x, 95, NT) “This software package perfectly complements ArcView GIS software and other desktop mapping software by allowing you to create fully intelligent (topological) data sets by digitizing or converting data from many formats, with functionality that previously existed only in a full ARC/INFO system.”
Didger	Golden Software www.golden.com	PLATFORM : Windows (3.51, 95, 98, NT) “Didger is a highly accurate digitizing program that will be an invaluable addition to your software library. In seconds, Didger precisely transforms points, lines, or areas from your paper maps, graphs, aerial photos, or other plotted information to a versatile digital format you can use with your other software. You name it and Didger can handle it quickly, accurately, and usefully. And

		don't be fooled by the low $119 price. With Didger's multitude of features and ease-of-use, this is an unbelievable value considering the time and effort savings you will realize! You will soon wonder how you have done your job without this indispensible tool."
ERDAS IMAGINE Professional	ERDAS www.erdas.com	PLATFORM : Windows (95, NT), Most Unix platforms "Quick image testing and modeling allows you to automatically and quickly generate broad land cover maps. Perform graphical spatial data modeling and integrate radar imagery. "
ERDAS MapSheets	ERDAS www.erdas.com	PLATFORM : Windows "All that work you put into gathering and analyzing geographic data... without a quick and simple way to visually present the information to your boss, your client or your prospective customer. Because it's fully Microsoft Office Compatible, MapSheets gives you the same look and feel of other Microsoft applications, as well as the power to create a convincing PowerPoint presentation, an effective Word report or a compelling Web page."
ER Mapper	Earth Resource Mapping www.ermapper.com	PLATFORM : Windows, Unix "Used in more than 142 countries world wide and supported by 514 ER Mapper resellers around the globe, ER Mapper is the world's most popular desktop integrated image processing software."
ER Viewer	Earth Resource Mapping www.ermapper.com	PLATFORM : Windows, Unix "ER Viewer is an easy to use image viewer featuring

		interactive roaming and zooming with very large image files. "
Geographic Calculator	Blue Marble Geographics www.bluemarblegeo.com	PLATFORM : Windows "The Geographic Calculator converts individual coordinates, point databases, and map files from virtually any coordinate system, datum, and map projection to any other."
Geographic Explorer	Blue Marble Geographics www.bluemarblegeo.com	PLATFORM : Windows (95, NT) "The Geographic Explorer extends the Windows 95 and NT Explorer to view, translate, and find map files. The Geographic Explorer provides a number of powerful utilities. You can explore map files directly on your desktop with the Geographic Explorer."
Geographic Tracker	Blue Marble Geographics www.bluemarblegeo.com	PLATFORM : Windows "The Geographic Tracker delivers all of the power and utility of GPS to the Windows environment by allowing you to link a GPS receiver to other Microsoft Windows applications through standard Windows DDE messaging. This standard "plug-and-play" GPS interface component supports most of the leading GPS receivers on the market. Simply plug your GPS receiver into the serial port of a computer running Windows and the Geographic Tracker will GPS-enable your applications."
Geographic Transformer	Blue Marble Geographics www.bluemarblegeo.com GEMI Store www.gemistore.com	PLATFORM : Windows $799 upgrade: $299 "The Geographic Transformer allows you to easily establish an "image-to-world" relationship between image and map coordinates and reproject an

		image into a georeferenced image map."
Geographic Translator	Blue Marble Geographics www.bluemarblegeo.com	PLATFORM : Windows "The Geographic Translator supports comprehensive data and coordinate system translation of map files in the AutoCAD DWG & DXF, Microstation DGN, MapInfo MIF & TAB, and ESRI Shape formats. "Hands free" batch mode operation is just one compelling reason to add this new Blue Marble tool to your GIS toolbox."
Geomatica FreeView	PCI Geomatics http://www.pcigeomatics.com	PLATFORM: Windows 95, 98, NT, 2000 "The Geomatica[tm] FreeView includes tools for fast image roaming and magnification, enhancements, numeric value display, and attributes table display."
Geomedia	Intergraph http://www.intergraph.com	PLATFORM : Windows "GeoMedia is your information integrator, serving as a visualization and analysis tool and a platform for custom GIS solutions. A universal GIS client, GeoMedia's live data connections allow you to maintain GIS data in native repositories, and ensure that your data is always up to date. You can integrate data automatically using on-the-fly coordinate transformation and feature definition. Perform simple and complex spatial analysis, such as incident and proximity analysis, to support decision-making activities."
Geomedia Viewer	Intergraph http://www.intergraph.com	PLATFORM : Windows "Create Thematic Maps... View and anlyze data in either Microsoft Access or Arc/View Shape format... Load a satellite

		raster image and easily manipulate your map with zoom-in, zoom-out capability... it's FREE!!!"
Manifold System	manifold.net http://www.manifold.net	PLATFORM : Windows "Manifold is a software package that combines database and mapping functionality. Manifold also includes a variety of statistics, logic, mathematics, and networking capabilities for use on maps, on databases, or all together at the same time."
MapInfo Professional	MapInfo www.mapinfo.com/	PLATFORM : Windows, Macintosh, Sun, HP "MapInfo Professional is the world's leading desktop mapping software that helps you see patterns and trends in your data. Trends that may otherwise be impossible to identify. "
MapInfo ProViewer	MapInfo www.mapinfo.com/	PLATFORM : Windows "MapInfo's easy viewing tool designed for MapInfo Professional users. With ProViewer, you can share your MapInfo Professional maps and analyses with clients and co-workers, who may easily view and manipulate your creations even if they do not have MapInfo Professional."
Maplex	ESRI www.esri.com	PLATFORM : Windows (NT) "Maplex is a fully automated cartographic name placement software. Name placement can be one of the most time-consuming aspects of map production, often requiring considerable manual intervention to ensure that names do not overlap and are clearly associated with the features they annotate. Maplex provides the means to overcome this bottleneck in

		map production."
MapMarker and MapMarker Plus	MapInfo www.mapinfo.com/	PLATFORM : Windows They both utilize "street maps to allow you to visually display all your geocoded data. MapMarker ships with MapMarker Streets - a five layer street display product based on TIGER data from U.S. Census Bureau. MapMarker Plus ships with MapMarker Plus Streets - a five-layer street display product based on data from GDT, Inc."
MAPublisher	Avenza www.avenza.com	PLATFORM : MacOS PowerPC, Windows (95, 98, NT) "MAPublisher 3.5 is a suite of Xtras for Macromedia FreeHand and Plug-ins for Adobe Illustrator 8.0.1 that bridges Geographic Information System (GIS) technology with high-end graphics software for high resolution printing and electronic publishing technology. "
MAPublisher GPS	Avenza www.avenza.com	PLATFORM : Windows (95, 98, NT) "MAPublisher GPS allows you to publish your data quickly, easily and intuitively while speaking to a GPS receiver through a satellite/radio network from the Illustrator graphics environment. This data can efficiently be made available on the internet/intranet by using Avenza's pdfPLUS technology."
Map Viewer	Golden Software www.golden.com	PLATFORM : Windows (95, 98, NT) "MapViewer is a thematic mapping package that creates full-color maps by linking worksheet data to areas or points on a map. MapViewer includes numerous boundaries and data files for the United

		States and World to get you started, or you can create your own boundaries for use with MapViewer. By placing your data in the full-featured MapViewer worksheet, you can easily link your data to areas or points on the map, and generate instructive, publication quality maps in virtually no time. With the numerous map types, overlays, and drawing tools you will create the most informative maps possible."
MicroStation GeoGraphics	Bentley Systems, Incorporated www.bentley.com	PLATFORM : Windows (3.1x, 95, NT), DOS. Releases are planned for Intergraph CLIX, Hewlett Packard HP/UX, Sun Solaris, DEC Alpha NT, Silicon Graphics, IBM RS6000, and Macintosh Power "MicroStation GeoGraphics is a comprehensive mapping toolbox that bridges the gap between CAD and GIS. MicroStation GeoGraphics combines MicroStation's industry-standard data capture and editing tools with a comprehensive database interface and a powerful spatial analysis engine. MicroStation GeoGraphics integrates seamlessly with MicroStation 95, and provides tools to input, validate, manage, analyze, and visualize geographical information. MicroStation GeoGraphics also has the ability to manage a diverse combination of vector data, raster data, and non-spatial attributes. "
OziExplorer	ESRI www.powerup.com.au/~lornew/oziexp.html	PLATFORM : WIN 95/98/NT4/NT5/WIN2000 "OziExplorer GPS Mapping Software will work with Magellan, Garmin, Lowrance and Eagle GPS receivers for the upload/download of data

		and most brand of GPS receivers for on screen tracking of position (Moving Map)."
PC Arc/Info	ESRI www.esri.com	PLATFORM : DOS, Windows (3.x, 95, 98, NT) "PC ARC/INFO is a series of six integrated software modules that combine fundamental geographic information system (GIS) tools and utilities for cartographic design and query, data entry and editing, data translation, polygon overlay and buffering, and network analysis and modeling."
Surfer	Golden Software www.golden.com	PLATFORM : Windows (3.1, 95, 98, NT) upgrade: $139 "Surfer is a contouring and 3D surface plotting program that runs under Microsoft Windows. Surfer quickly and easily converts your data into outstanding contour maps and surface plots. Virtually all aspects of your maps can be customized to produce exactly the presentation you want. Producing publication quality maps has never been quicker or easier. Both the 32-bit and 16-bit versions are included."
Terrain Professional	Maptech GEMI Store www.gemistore.com	PLATFORM : Windows (95, 98, NT) "High-quality USGS topographic maps on CD-ROM. Up to 300 exportable maps per CD, at 160 dpi image resolution. Advanced features let you export and reproject full, georeferenced maps for use in GIS applications. Annotate and print customized maps. NGS (National Geodetic Survey) Data Sheets included. "
TNTatlas	MicroImages, Inc. www.microimages.com	PLATFORM : Every Platform "a free product that lets you

		publish and distribute your spatial materials on CD-ROM. TNTatlas CD's contain multiple versions of the software so that a single CD can be used on any popular computing platform. TNTatlas stacks are prepared with the HyperIndex® feature in TNTmips."
TNTview	MicroImages, Inc. www.microimages.com	PLATFORM : Every Platform "TNTview gives you the same powerful visualization and interpretation tools contained in MicroImages' industry-leading TNTmips product. TNTview is easy to learn and provides low cost visualization of your organization's project materials at sites where the users' scope of work does not include the preparation, processing, or final production of your spatial data."
WinCATS	Applied Global Technologies, Inc. www.appliedglobal.com/frames.htm	PLATFORM : Windows "WinCATS is the cutting edge intel tool for displaying database information over a vast host of geographic backgrounds. Advanced enough for the most complex cartographic demands, yet simple enough for the everyday software user, WinCATS is a serious mapping tool that remains user friendly."

14. Summary

In common with many areas, a spatial and temporal database systems research should generally be characterized by research that:

• is collaborative with other research in cognate areas of information technology research, such as mobile telecommunications;

• integrates with other disciplines such as medicine, CAD /CAM, GIS, environmental science, molecular biology or genomics/bioinformatics;

• uses large, real databases—synthetic databases are often of less use in this area than in many others;

• where possible, attempts to change the accepted paradigm or approach to a given problem.

15. Future Activities

Management of spatial data is not yet satisfactorily simple for such end-users as cartographers and others. Hence, the anticipation is that friendly graphical user interfaces will have to be developed on top of DBMS that handle spatial data.

16. Bibloigraphy

Books :

1.Spatial Databases (Technologies , Techniques and Trends) ... By Yannis Manolopoulos ,Apostolos N Papodopoulos and Michel Gr. Vassilakopoulos
2. Database Management System By korth. Silberchaz
3. Spatial databases: with application to GIS.. By Philippe Rigaux, Michel Scholl.

Websites :

1. http://dna.fernuni-hagen.de/Tutorial-neu.pdf
2. http://en.wikipedia.org/wiki/Spatial_database
3. www.informatik.fernuni-hagen.de/import/pi4
4. http://www.timeconsult.com/TemporalData/TemporalDB.html
5. http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u10.html
6. http://en.wikipedia.org/wiki/Temporal_database
7. http://www.sigmod.org/record/issues/0406/RR1.SSTDpaper.pdf
8. http://books.google.co.in/books

Journals:

1. Burrough, P. A. and Frank, A. U., 1995, Concepts and paradigms in spatial information: are current geographical information systems truly generic? International Journal of Geographical Information Systems.

2. K. K. Al-Taha, R. T. Snodgrass and M. D. Soo, 'A Bibliography on Spatio-temporal Databases,' *International Journal of Geographic Information Systems*, Vol. 8, No. 1, January-February, 1994.

3. D. Peuquet and E. Wentz, 'An approach for time-based analysis of spatiotemporal data,' *Proc. of the 6th International Symposium on Spatial Data Handling*, Edinburgh, 1994.