SPATIO-TEMPORAL INFORMATION SYSTEMS

Claudia Bauzer Medeiros
Institute of Computing, University of Campinas, Brazil

Keywords: database management systems (DBMS), geographic information systems, information system internals, spatial databases, spatio-temporal DBMS, scientific databases, temporal databases, versions

Contents

Summary

The notions of space and time are omnipresent in our life; we generate and are immersed in continuous streams of data that are spatial and that vary with time. Nothing is more natural, therefore, than to store these data and to develop information systems to help us manage and extract information from them—the so-called *spatio-temporal information systems*.

What kind of information do these systems help us get? They allow experts to derive spatio-temporal patterns from the stored facts, mingling past, present, and future. One major goal is to predict the future, by looking at the past—for example through trend analysis or behavior forecast systems. Another objective is to study the past to explain the present—for instance, using reasoning and diagnosis systems. Yet another possibility is to simulate the future to improve the present, through what-if scenario generation and decision support systems. Furthermore, spatio-temporal data allow us to document the present, as is done by engineering or cadastral systems.

Since we are surrounded by spatio-temporal data sources, it is easy for us to understand the utility and the goals of spatio-temporal systems. How are these systems, however, built inside? What makes them work, and what are the challenges in this domain? This text tries to answer some of these questions by analyzing the internals of spatio-temporal information systems from a computer science perspective. Since any

information system requires a database management system (DBMS) to manage the underlying data, the text concentrates on discussing research problems in spatio-temporal DBMS. It analyzes questions such as how to store spatio-temporal data, and how to manage and process these data in order to derive information from them. Most of the paper is directed towards geographic data, but it can be extended to other kinds of spatio-temporal data and systems.

1. Introduction

Spatio-temporal information systems are software packages that are developed to help us manage data about entities whose properties change with space and time. They are called information systems because they allow us to derive information by combining, analyzing, and aggregating these data, and visualizing them in multiple ways. They are spatio-temporal because of the nature of the data they handle. These data are stored in spatio-temporal databases; and the information systems rely on spatio-temporal database management systems (DBMS) to provide efficient and reliable data management functions.

In order to learn about spatio-temporal information systems, one must understand: how they *manipulate* data, by studying *spatio-temporal DBMS*; how these data are *stored*, by studying *spatio-temporal databases*; and how these data are *obtained*, by looking at some *spatio-temporal data sources*. The terms in italics in this paragraph are the key concepts that will be investigated in this paper.

The entire world is a complex system of spatio-temporal phenomena. What vary are the nature of the data being generated, and the temporal and spatial units considered. The spatial properties that change with time may be location, direction, size, or shape. This change may be continuous or discrete; cyclic or not; reversible or permanent.

Our bodies, for instance, are good sources for these data. The electric impulses of our brains can be charted on brain maps as spatial information; the changes in these maps through time—as we talk, move, hear, sleep, or think—are spatio-temporal data, for very small spatial and temporal units. At a different spatio-temporal scale, when we move—walking in our homes, crossing a street, driving a car, swimming at a beach, flying in a plane—the variation of our position with time is again spatio-temporal data. Medical information systems use brain maps, while mobile information systems deal with entities displaced in time.

We ourselves are sources of spatio-temporal data. Our acts (in constructing or dismantling entities) can also create these data. When we assemble a jigsaw puzzle, build or reform our homes, buy or join land parcels, plant one tree, grow vegetables, construct bridges, highways, dams, artificial lakes, we are creating spatio-temporal entities and changing their shape and size. These data are used in computer-aided design (CAD) systems, urban, transportation and environmental planning systems. These entities we create or destroy interact with each other and other entities, and produce further spatio-temporal data. The tree that we plant attracts birds and insects, and causes soil changes around it. The dam, the highway, and the lake have major impacts on the ecological balance of the area where they are situated, providing data which are studied in biodiversity information systems.

Nature is the most complex source of spatio-temporal data. Animals move and grow and construct things just like we do. Vegetation grows and changes with time. Seasons come and go, rivers flow, continents move and so does our planet. This paper is centered on geographic data, and thus the problems discussed concern primarily these kinds of data. However, most of the issues presented are valid for any kind of spatio-temporal data.

For this reason, the term "spatio-temporal data" is frequently used as a synonym for "geographic data." Indeed, geographic phenomena are prime sources for spatio-temporal data. The information systems that use geographic data receive a special name: GIS, or Geographic Information Systems (see *Advanced Geographic Information Systems).*

Examples of such systems are those dealing with urban planning, route optimization, public utility network management, demography, cartography, agriculture, natural resources administration, coastal monitoring, fire and epidemic control. Each type of application deals with different features, geographic and temporal scales and properties. Geographic data portray any feature that is related to the Earth. One can say that "Lake Titicaca" or "the Taj Mahal" relate to geographic data, because they refer to geographic entities, whereas "John Doe" is not usually seen in a geographic context.

The term "geo-referenced" refers to data about some geographic phenomenon associated with its location, spatially referenced to the Earth. Geographic entities are not considered to be geo-referenced until coordinates have been assigned to them, under some projection system. The Taj Mahal or Lake Titicaca can be stored in a geo-referenced data record if one field in this record contains the mausoleum's or the lake's location coordinates.

This text treats the terms "geographic" and "geo-referenced" as synonyms, in spite of this difference between the two concepts, and assumes that data records on geographic entities have been properly geo-referenced.

The concept of geo-referencing serves to distinguish between normal databases and spatial databases. For instance, any number of non-spatial databases may contain address data on "John Doe," including telephone directory or car licensing systems. If the "address" field is associated with the corresponding geographic coordinates, then the database becomes a spatial database: now one can establish any kind of spatial correlation between John Doe and his surroundings, whereas before geo-referencing one could only ask questions about his address. It becomes possible to compute the distance between his home and anywhere in the world, the probability that his home may be affected by some environmental disaster such as floods or tornados, or the need for him to take a preventative health check-up in case he lives within an epidemic risk area.

If furthermore the geo-referenced database keeps track of John Doe's address changes throughout the years, then it is a spatio-temporal database. Now one can compute the same facts but for each time period (e.g. what was the distance between John's home and the Taj Mahal in 1998? And in 1972?) Finally, if instead of geo-referencing John's address the database geo-references John's *location* at any time of the day (e.g. using a location sensor such as a GPS) then it is transformed into a mobile spatio-temporal database. Besides being able to answer all the previously mentioned questions, one can also ask questions about John's different itineraries between any two places, and help John avoid traffic jams or find the nearest gas station on a road.

What are the challenges in the development of spatio-temporal information systems? To study information systems, one must look at them from two angles: the kinds of information they generate (end-users' point of view) and how they manage data (information system programmer's point of view). The first aspect is discussed in *Advanced Geographic Information Systems*. This paper concentrates on the second aspect. Section 2 presents a brief history on DBMS for temporal, spatial, and spatio-temporal data. Sections 3 and 4 present the basic framework that will be used in the paper, characterizing a DBMS, temporal, spatial, and spatio-temporal DBMS. The subsequent sections analyze open problems under this framework. Finally, Section 10 points out some open problems and future trends in the field.

2. Historical Background

The development of spatio-temporal DBMS faced several obstacles. In the first place, hardware limitations prevented, until recently, the possibility of storing and retrieving massive volumes of data. It was only in the 1990s that storage device technology evolved to make it possible to store and access large data sets with an acceptable performance, thus setting up the computer technology infrastructure scenario for research and development in the domain of spatio-temporal DBMS.

Another problem concerned the management of the data attributes themselves. Typically, temporal and spatial data records require nested and composite fields. In the late 1980s, the management of nested data received a boost from research on object-oriented database systems, which allowed the management of what are known as "complex objects." In this context, a complex object is any record where fields may contain not only data values, but also pointers to other records, which in turn may also have pointers to further records. Thus, one record (i.e. a database object) is actually made up of several levels of nested records, which may furthermore come from several files. Objects are associated with functions that can be applied to them (called methods). For more on object-oriented databases, see *Spatial Query Languages.*

Finally, there is the nature of the data, which require custom-made functions to manage time and space properties. While present commercial DBMS can already handle massive data and complex records, they have little support for temporal and spatial analysis operations. Spatio-temporal database systems are based on work developed on temporal databases and on spatial databases.

Temporal database management: Temporal DBMS concern the handling of data that vary with time, but where the notion of space is ignored. Work in this domain has been largely centered on extending relational (tabular) databases, with a few proposals for object-oriented systems. Research in temporal DBMS started in the early 1980s, with hundreds of papers and tens of prototypes. The main motivation for this research direction was given by business and administrative applications. Stock and sales inventory are the most common applications of the first kind, whereas personnel management is an example of the second kind. In the mid-1980s temporal database systems began to be used to manage data generated by distinct kinds of scientific domains—the so-called scientific applications. This gave origin, in turn, to a new research field in computer science, that of scientific databases—those that store data generated by scientific research, for example in physics or biology. Information systems using these databases were called scientific information systems.

A large group of these scientific systems was concerned with the management of time series (e.g. in statistical studies). The novelty for the scientists who adopted scientific information systems resided in the fact that data were now managed by DBMS instead of being stored in unstructured flat files, thus supporting efficient data retrieval and update, and freeing the scientists from the task of managing these files. In the natural sciences, temporal databases began to be used to store environment-related measurements (rainfall, temperature, and so on), but in a limited way. Thus, more often than not, computer systems dealing with environmental temporal-varying data did not take advantage of research on temporal DBMS and moreover did not even rely on database systems. Rather, they were concerned with algorithmic and simulation aspects, privileging the mathematical modeling of phenomena and disregarding issues of efficient data storage and management. Most of the research on temporal databases remained restricted to business-oriented applications.

Spatial database management: The earliest examples of software to handle geographic data date back to the 1960s, but without using DBMS. A wide range of commercial geographic information systems (GIS) was developed in the following two decades, but DBMS use was very limited. Vendors developed proprietary solutions, with little flexibility. Most of these systems, for instance, did not support standard DBMS functions such as concurrent access to data or recovery from failures.

At the end of the 1980s, some systems started using relational DBMS, while research prototypes concentrated on object-oriented technology. The research field of spatial DBMS was consolidated in the 1990s. Here, the emphasis was on storing data about space, motivated by geographic studies and the design of engineering artifacts, but without concern for temporal issues. Research on GIS, up to then mostly restricted to geoscientists, was recognized by computer scientists as a worthwhile field. The term "GIS," from then on, would denote any kind of information system that manipulated maps and/or remote sensing data.

Spatial data management is a field that is receiving growing attention from computer scientists, motivated by global-scale problems such as global warming, pollution, and biodiversity preservation (see *Biodiversity Information Management).* At another scale, economic motivations are pushing forward the development of spatial information systems, notably in transportation and agriculture (see *Precision Farming and Geographic Information Systems).*

Spatio-temporal data management: One of the earliest examples of spatio-temporal information systems using DBMS dates back to the end of the 1970s. It concerns computer-aided design systems (or CAD). These information systems help users—such as engineers and architects—to design and test engineering artifacts. CAD systems can be used to design airplanes, buildings, automobiles, furniture, or electronic gadgets. These systems are spatial in the sense that they must cater to 3D features of artifacts and their parts, and temporal since they allow data versioning to accommodate the evolution of the design, as these artifacts undergo successive improvements. CAD databases store data on artifacts, but also on their components, their drawings and geometries, and materials used to construct them.

CAD information systems support several functions that are shared by other kinds of spatio-temporal information system. Design functions allow their users to assemble these parts and visualize the artifacts being designed, and compare alternative designs. Simulation functions let users run performance, stress, durability, and several other kinds of tests on these artifacts and their parts. Documentation functions support associating descriptive text to the parts and the artifacts, which indicates how they were constructed.

Another early example of spatio-temporal information systems occurred in the domain of VLSI (very large scale integration) chip design. In this context, the database stores millions of blueprints, circuit layouts and performance data on electronic chip components, and is used by electronic engineers to design and test chip layouts. Still other examples of spatio-temporal information systems help users manage biochemical components and their molecular structures (biochemistry information systems), analyze the universe (astronomy information systems) or investigate the human body (anatomy/physiology information systems).

This work notwithstanding, from a data management perspective two research and experimentation lines have been pursued in parallel, one concerning temporal evolution, regardless of space, and the other dealing with spatial characteristics, for a fixed moment in time. The mid-1990s saw the creation of the Chorochronos research network in Europe. This network allowed researchers working on spatial and temporal databases to cooperate and integrate their results and methodologies. This network has fostered several proposals of models and languages for spatial, temporal, and spatio-temporal systems, and developed some prototypes and testbeds.

Nevertheless, all this research effort has barely affected commercial DBMS. Still in 2006, spatial DBMS limit end-users to nontemporal data management. In order to detect spatio-temporal patterns, these users basically generate static views of spatial data—such as maps—for different points in time, and compare these views by visual inspection—for example, by overlaying them—in order to figure out temporal evolution. Sometimes, the management of these static snapshots may be helped by versioning mechanisms.

One can also create animations of spatial data evolution, either by associating static maps to video frames or by running animated simulations of real world phenomena. However, spatio-temporal data management is still restricted to research that concerns these kinds of data —in database systems, in human–machine interfaces, in computer graphics and in several scientific and technological domains.

Thus, at the beginning of the twenty-first century, there is no satisfactory solution to the problem of managing spatio-temporal data, as opposed to the wide range of generic DBMS for business data sets. It is true that solutions exist for specific problems, and one can find software that has been custom-built to deal with a given question. Examples are systems that manage data concerning water or air pollution, developed to store these data, analyze, and use them under distinct kinds of spatio-temporal simulation. The dissemination of mobile devices has introduced a strong motivation for the development of applications that keep track of displacement of objects on time (see *Issues in Spatio-Temporal Database Systems: Models, Languages and Moving Objects).*

3. Basic Framework

As was shown in the Introduction of this paper, all information systems rely on a DBMS. Sales figures for DBMS comprise a major percentage of the world's software market. Spatio-temporal DBMS are first and foremost database management systems that manage a specific kind of data: those that reflect phenomena that vary in space and in time.

This definition, seemingly obvious, requires the understanding of a considerable range of concepts, and is the key for understanding information systems' internals. First, one must know how a database management system works; second, one has to understand how spatio-temporal data are stored in databases; finally, there is the need to grasp the DBMS functions used to manage such data. This section is concerned with establishing this basic framework. It starts by giving an overview of generic DBMS, and follows by describing temporal, spatial, and spatio-temporal databases. This framework is used in the rest of this paper.

3.1. Characterizing a DBMS

A DBMS is a set of interconnected software modules, each of which has to perform several database management tasks. The advantage of using a DBMS as a building block for constructing information systems is that it liberates the computer experts who program these systems from the task of data management, allowing them to concentrate on developing the data analysis and visualization functions. Figure 1 portrays the modules of a DBMS schematically, in a four-layer architecture: interface, query processing and optimization, transaction management, and input/output (I/O) handling.

The top layer, "interface", is responsible for offering to information system developers adequate tools to develop such systems. There are basically two kinds of facility offered at this level: query languages (such as the standard Structured Query Language, SQL) and application programming interfaces (API). There are furthermore additional tools that help design the database.

The interface layer is followed by the "query processing" layer. The goal of the latter is to transform requests posed at the interface (queries, updates, and programs embedding both) into code that can be executed efficiently by the DBMS. The "transaction management" layer is responsible for ensuring concurrency control, thereby allowing multiple applications and users to query and update data simultaneously. It is also responsible for providing data recovery when there are system or power failures. Finally, the "I/O handling" layer coordinates data transfers between the main memory (where the code is executed) and the secondary storage (the actual databases). Traditionally, indexing issues and file structures are also considered at this level**.**

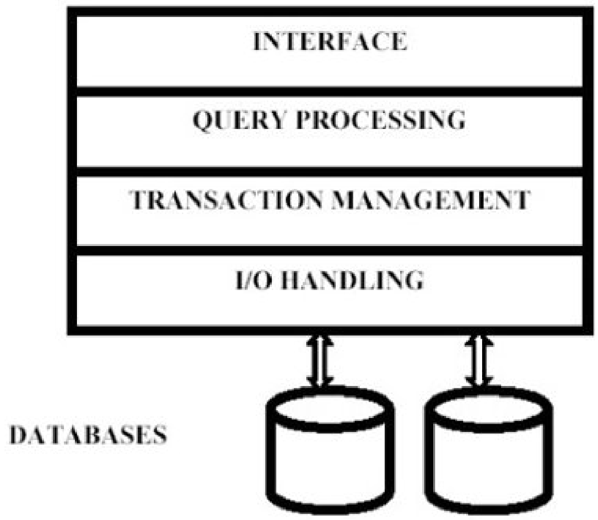

Figure 1. Schematic presentation of a DBMS, consisting of four layers. Requests at the "interface" level are translated into the DBMS internal code and passed on to the "query processing" level, where they are optimized. The "transaction management" layer ensures concurrent execution of data requests and updates. The "I/O handling" layer is responsible for transferring data between the main memory (where the code is executed) and the databases (where the data are stored).

The database (or databases) managed by the DBMS consists of a large collection of interrelated data files organized especially for rapid search and retrieval. Databases are constructed in three phases: database modeling, schema specification, and data loading. The first phase consists of specifying all the characteristics of the entities that will be stored in the database, together with their relationships and constraints. The result is then translated into a "database schema," which is basically the definition of each database file in terms of its record structure: record attributes, attribute domain, and constraints. One important constraint is called the "key," which is a set of attributes that uniquely identify a record: for example, the social security number of a person. The schema also specifies how the database files are interrelated. The third stage consists of entering data into the files previously defined. These data must conform to the schema specification. In the case of spatio-temporal databases, attributes may be of three kinds: temporal attributes, discussed next; spatial attributes, discussed in Section 3.3; and descriptive attributes. For more on spatial data modeling, see *Conceptual Modeling of Geographic Applications.*

-
-
-

Bibliography

Abraham T. and Roddick J. (1999). Survey of spatio-temporal databases. *GeoInformatica*, **3**(1), 61–99. [A good overview of spatio-temporal databases, centered on data structures and applications.]

Chakrabarti C. and Mehrotra S. (1999). Efficient concurrency control in multidimensional access methods. *Proceedings ACM SIGMOD Conference*, ACM: New York, 25–36. [One of the few papers that discusses transaction-level issues for multidimensional data.]

Egenhoffer M.J. and Golledge R.G. (eds.) (1998). *Spatial and Temporal Reasoning in Geographic Information Systems*, 275 pp. New York: Oxford University Press. [Collection of papers written by experts in temporal and spatial reasoning, modeling and cognition issues in geographic space and GIS.]

El Masri R. and Navathe R. (1999). *Fundamentals of Database Systems*, 960 pp., 3rd edn. New York: Addison Wesley. [A well-structured undergraduate level text on how to design and construct databases, with a good presentation on the internals of database management systems.]

Frank A., Campari I., and Formentini U. (eds.) (1992). *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space.* 434 pp. Lecture Notes in Computer Science 639, Berlin: Springer–Verlag. [Set of papers of a conference on spatio-temporal reasoning, which discuss a wide set of issues about early problems of modeling and implementing geographic systems.]

Gaede G. and Gunther O. (1998). Multidimensional access methods. *ACM Computing Surveys*, **30**(2): 170–271. [A comprehensive survey on indexing and spatial access methods for multidimensional data.]

Guting R.H. (1994). An introduction to spatial database systems. *VLDB Journal*, **3**(4), 357–400. [An introduction to spatial database systems, from a database point of view.]

Langran G. (1992). *Time in Geographic Information Systems*, 220 pp. London: Taylor and Francis. [A pioneer text in the study of temporal problems in the management of geographic data.]

Longley P.A., Goodchild M.F., Maguire D.J., and Rhind D.W. (eds.) (1999). *Geographical Information Systems: Principles, Techniques, Applications and Management*. 453pp. 2nd edn., New York: John Wiley. [This classic two-volume book provides a comprehensive overview of GIS theory and applications, written by experts in the field.]

Manolopoulos Y., Theodoridis Y., and Tsotras V. (1999). *Advanced Database Indexing*, 312pp. Kluwer International Series on Advances in Database Systems. Boston: Kluwer Academic. [Text on several kinds of index structures, and their applications, with a section dedicated to spatio-temporal indexing.]

Ooi B. C. (1990). *Efficient Query Processing in Geographic Information Systems*, 208pp. Lecture Notes in Computer Science 471. Berlin: Springer Verlag. [One of the first comprehensive studies on query processing techniques for spatial databases, with implementation discussions.]

O'Neil P. and O'Neil E. (2001). *Database Principle, Programming and Performance*. 960pp. Morgan Kaufmann Series in Data Management Systems. 2nd edn. San Francisco: Morgan Kaufmann. [Very good text on database internals for information system developers. Provides a good comparison of several commercial DBMS, and discusses object-relational databases as implementation platforms.]

Salzberg B. and Tsotras V. (1999) Comparison of access methods for time–evolving data. *ACM Computing Surveys*, **2**(31), 158–221. [A good survey of temporal access methods.]

Samet H. (1990). *The Design and Analysis of Spatial Data Structures*. Reading: Addison-Wesley. [A classic textbook on spatial data structures.]

Snodgrass R. (ed.) (1995). *The TSQL2 Temporal Query Language.* 704 pp. Boston: Kluwer Academic Publishers. [Set of papers on the foundations of standards in temporal query languages, and the establishment of the TSQL2 standard.]

Stonebraker M., Frew J., Gardels K., and Meredith J. (1993). The Sequoia 2000 Benchmark. *Proceedings of the ACM SIGMOD Conference*, pp. 2–11, ACM: New York. [One of the few benchmarks proposed to test the performance of queries on geographic databases.]

Biographical Sketch

Dr. **Claudia Bauzer Medeiros** is a full Professor of Computer Science at the Institute of Computing of the Universidade Estadual de Campinas (UNICAMP), Brazil. She received her electronic engineering degree in 1976 and her M.Sc. in Informatics in 1979 from Pontificia Universidade Catolica do Rio de Janeiro, Brazil. Her Ph.D. was awarded by the University of Waterloo, Canada, in 1985, in the area of database systems. She is the head of the database research group in UNICAMP, and leads projects on different aspects of geo-spatial databases, including query processing, versions, spatial decision support systems, and the management of integrity constraints and temporal data. Other research interests include image databases and documentation management. One of her major concerns is the design and development of applications centered on geo-referenced databases, aggregating interdisciplinary teams, for environmental and urban planning. She has coordinated several large research and development projects around GIS technology in Brazil, and was the senior database consultant for the development of many GIS applications, in the domains of urban planning, She was for six years the Editor in Chief of the *Journal of the Brazilian Computer Science Society*, the main scientific publication of the Brazilian Computer Society (SBC), and is on the editorial board of *GeoInformatica* and the *VLDB Journal*. She has been a keynote speaker or participated in program committees of major database and GIS conferences, having been the program chair of the Seventh ACM GIS Symposium. She is one of the co-founders of the Brazilian Special Interest Group on Databases, and a member of ACM, IEEE and SBC. She was awarded the 1996 and the 2001 Academic Merit Prize of the University of Campinas, and the 2000 Newton Fuller Prize of the Brazilian Computer Society.