Spatio-Temporal Database

Doctoral Course: Conceptual Modeling

Januray 25, 2005

Jiyong Zhang(jiyong.zhang@epfl.ch)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
School of Computer and Communication Sciences
CH-1015 Lausanne, Switzerland

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Recent News: U.S. Submarine hit undersea mountain

- Time: January 8, 2005
- The accident occurred about 350 miles south of Guam
- 1 sailor died, 24 injured

San Francisco, the nuclear-powered attack submarine

The undersea map was drawn on 1989

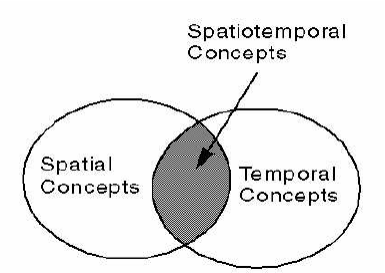
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Outline

- Introduction
- Spatial Database
- Temporal Database
- Spatio-Temporal Database
- Open issues for STDB Systems
- Summary

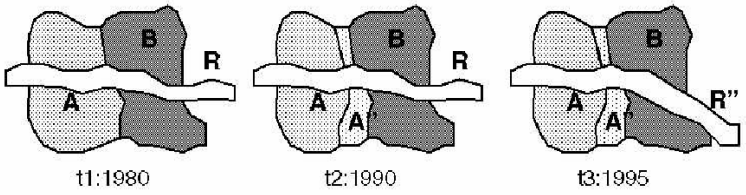
Introduction

- Spatio-Temporal Database:
  - A database that embodies spatial, temporal, and spatiotemporal database concepts, and captures spatial and temporal aspects of data
  - Dealing with geometry changing over time



Introduction (cont'd)

- Example of Spatiotemporal Applications
  - Year 1980: landparcel A has common borders with B; river R runs through A and B; A has soil type "clay", B has soil type "forest"
  - Year 1990: A was divided into A and A'', B has soil type "high-density forest", A'' has soil type "sparse forest"
  - Year 1995: river R changed its position and become R"



Spatial Database: Introduction

- Many applications in various fields require management of *geometric, geographic or spatial* data (data related to space)
  - A geographic space: surface of the earth
  - Man-made space: layout of VLSI design
  - Model of the human brain
  - 3-D space representation of the chains of protein molecules
- The Common challenge:
  - Dealing with large collections of relatively simple geometric objects: e.g., 100,000 polygons

Spatial Database: Definition

- A spatial database system:
  - Is a database system (with additional capabilities for handling spatial data)
  - Offers spatial data types (SDTs) in its data model and query language
    - Structure in space: e.g., POINT, LINE, REGION
    - Relationships among them: e.g., *a* intersects *b*
  - Supports SDT in its implementation
    - Spatial indexing: retrieving objects in particular area without scanning the whole space
    - Efficient algorithm for spatial joins

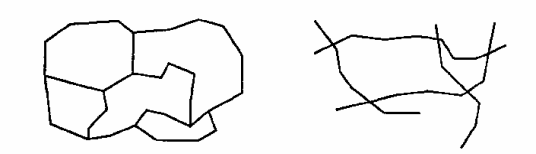
Spatial Database: Modeling

- Assume 2-D GIS application, two basic things need to be represented:
  - Objects in space: cities, forests, or rivers
    distinct entities arranged in space, each of which has its own geometric description
    =>modeling single objects
  - Space: describe the space itself
    say something about every point in space
    =>modeling spatially related collections of objects

Spatial Database: Modeling (cont'd)

- Fundamental abstractions for modeling single objects:
  - Point, Line, Region

- Spatially related collections of objects:
  - Partition, Networks



Spatial Database: Spatial Data Types and Operations

- A sample System: ROSE (Guting and Schneider, 1993)
- Three data types:
  - Points, lines, regions
- Define two type sets:
  - EXT={lines, regions}, GEO={points, lines, regions}

- Four classes of operations:

1. Spatial Predicates for topological relationships:

$\forall$ *geo* in GEO. $\forall$ ext_1, ext_2 in EXT. $\forall$ *area* in *regions*

geo × *regions*	→	*bool*	**inside**
$ext_1 \times ext_2$	→	*bool*	**intersects, meets**
area × *area*	→	*bool*	**adjacent, encloses**

Spatial Database: Spatial Data Types and Operations (cont'd)

- 2. Operations returning atomic spatial data type values:

  $\forall$ *geo* in GEO.

lines × *lines*	→ *points*	**intersection**
regions × *regions*	→ *regions*	**intersection**
geo × *geo*	→ *geo*	**plus, minus**
regions	→ *lines*	**contour**

- 3. Spatial operations returning number:

  $\forall$ $geo_1 \times geo_2$ in GEO.

$geo_1 \times geo_2$	→ *real*	**dist**
regions	→ *real*	**perimeter, area**

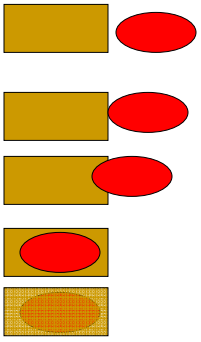
- 4. Spatial operations on set of objects:

  $\forall$ *obj* in OBJ. $\forall$ *geo*, geo_1, geo_2 in GEO.

set(*obj*) × (*obj* → *geo*)	→ *geo*	**sum**
set(*obj*) × (*obj* → geo_1) × geo_2	→ *set*(*obj*)	**closest**

Spatial Database: Spatial relationships

- Topological relationships
  - Disjoint, touch, overlap, in, cover, equal
- Direction relationships
  - Above, below, north_of, southwest_of,…
- Metric relationships
  - Distance



Spatial Database: Querying

- Two main issues:
  - 1. Connecting the operations of a spatial algebra to the facilities of a DBMS query language.
  - 2. Providing graphical presentation of spatial data (i.e. results of queries), and graphical input of SDT values used in queries.

Spatial Database: Querying (cont'd)

- Fundamental spatial algebra operations:
  - **Spatial selection**: returning those objects satisfying a spatial predicate with the query object
    - Example: All big cities no more than 300Kms from Lausanne
    - SELECT cname FROM cities c WHERE dist(c.center, Lausanne.center) < 300 and c.pop > 500K
  - **Spatial join:** A join which compares any two joined objects based on a predicate on their spatial attribute values
    - For each river pass through Switzerland, find all cities within less than 50KMs
    - SELECT c.cname FROM rivers r, cities c WHERE r.route intersects Switzerland.area and dist(r.route, c.area) < 50KM

Spatial Database: Querying (cont'd)

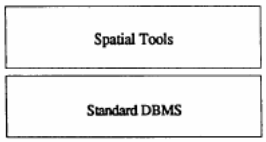
- Requirements for spatial querying
  - Spatial data types
  - Graphical display of query results
  - Graphical combination of several query results
  - Display of context
  - A facility for checking the context of display
  - Extended dialog
  - Varying graphical representations
  - Legend
  - Label placement
  - Scale Selection
  - Subarea for queries

Spatial Database: System Architecture

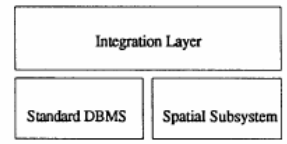
- Extensions required to a standard DBMS architecture
  - Representations for the data types of a spatial algebra
  - Procedures for the atomic operations,
  - Spatial index structures,
  - Access operations for spatial index,
  - Filter and refine techniques
  - Spatial join algorithms
  - Cost functions for all these operations (for query optimizer)
  - Statistics for estimating selectivity of spatial selection and join
  - Extensions of optimizer to map queries into the specialized query processing method
  - Spatial data types & operations within data definition and query language
  - User interface extensions to handle graphical representation

Spatial Database: System Architecture (cont'd)

- Previous approaches to GIS architecture
  - Built directly on top of file system
  - Using a Closed DBMS
    - 1. **Layered architecture**



    - 2. **Dual architecture**



Spatial Database: System Architecture (cont'd)

- Using an Extensible DBMS
  - There is no difference in principle between:
    - A standard data type such as a STRING and a spatial data type such as REGION
    - Same for operations: concatenating two strings or forming intersection of two regions
    - Sort/merge join and spatial join
    - Query optimization
- Current commercial solutions are OR-DBMSs:
  - NCR Teradata Object Relational (TOR)
  - IBM DB2 (Spatial extenders)
  - Informix Universal Server (Spatial datablade)
  - Oracle 8i (spatial cartridges)

Temporal Database: Introduction

- Most applications of database technology are temporal in nature:
  - Financial apps.: portfolio management, accounting & banking
  - Record-keeping apps.: personnel, medical record and inventory management
  - Scheduling apps.: airline, car, hotel reservations and project management
  - Scientific apps.: weather monitoring
- Definition:
  - Temporal DBMS manages time-referenced data, and times are associated with database entities

Temporal Database: Introduction (cont'd)

- Modeled reality
- Database entities
- Fact: any logical statement than can meaningfully be assigned a truth value, i.e., that is either true or false
- **Valid Time (*vt*)**
  - Valid time is the collected times when the fact is true
  - Possibly spanning the past, present & future
  - Every fact has a valid time
- **Transaction Time *(tt)***
  - The time that a fact is current in the database
  - Maybe associated with any database entity, not only with facts
  - TT of an entity has a duration: from insertion to deletion
  - Deletion is pure logical operation

Temporal Database: Introduction (cont'd)

- Time domain may be discrete or continuous
- Typically assume that time domain is finite and discrete in database
- Assume that time is totally ordered

- Uniqueness of "*NOW*"
  - The current time is ever-increasing
  - All activities is happed at the current time
  - Current time separates the past from the future
- "NOW" <> "HERE"
  - Time cannot be reused!
  - A challenge to temporal database management

Temporal Database: Modeling

- More than 24 extended relational models proposed
- Bitemporal Conceptual Data Model (BCDM):
  - timestamps tuples with sets of (tt, vt) values

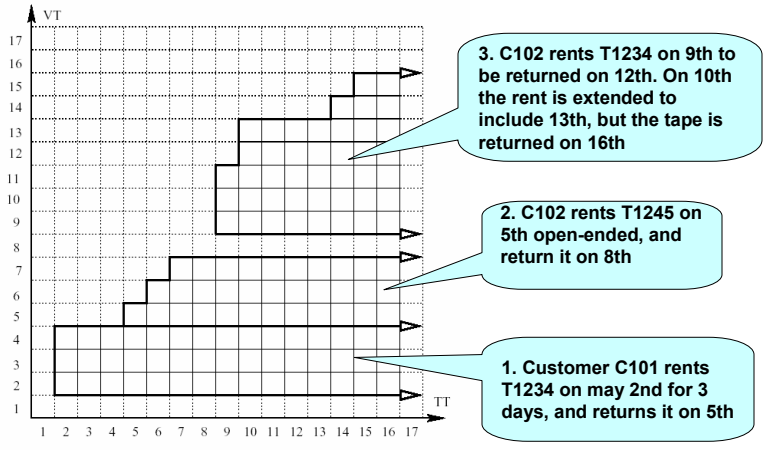
CustomerID	TapeNum	T
C101	T1234	{(2, 2), (2, 3), (2, 4), (3, 2), (3, 3), (3, 4), . . . , (UC, 2), (UC, 3)(UC, 4)}
C102	T1245	{(5, 5), (6, 5), (6, 6), (7, 5), (7, 6), (7, 7), (8, 5), (8, 6), (8, 7), . . . , (UC, 5), (UC, 6), (UC, 7)}
C102	T1234	{(9, 9), (9, 10), (9, 11), (10, 9), (10, 10), (10, 11), (10, 12), (10, 13), . . . , (13, 9), (13, 10), (13, 11), (13, 12), (13, 13), (14, 9), . . . , (14, 14), (15, 9), . . . , (15, 15), (16, 9), . . . , (16, 15), . . . , (UC, 9), . . . , (UC, 15)}

UC: until changed

  - Customer C101 rents T1234 on May 2nd for 3 days, and returns it on 5th
  - C102 rents T1245 on 5th open-ended, and return it on 8th
  - C102 rents T1234 on 9th to be returned on 12th. On 10th the rent is extended to include 13th, but the tape is returned on 16th

Temporal Database: Modeling (cont'd)

- Graphical Illustration of the Timestamp Values



Temporal Database: Modeling (cont'd)

- BCDM pros:
  - Simple, while also capturing the temporal aspects of the facts stored in a database.
  - Since no two tuples with mutually identical explicit values are allowed in BCDM relation instance, the full history of a fact is contained in exactly one tuple.
- BCDM cons:
  - Bad internal representation and display to users of temporal info
  - Varying length and voluminous timestamps of tuples are impractical to manage directly
  - Timestamp values are hard to comprehend in BCDM format

Temporal Database: Querying

- Temporal queries can be expressed in conventional query language such as SQL, but with great difficulty
- Language design must consider
  - Time-varying nature of data
  - Predicates on temporal values
  - Temporal constructs
  - Supporting states and/or events
  - Supporting multiple calendars
  - Modification of temporal relations
  - Cursors, views, integrity constraints,
  - handling now, aggregates, schema versioning, periodic data
- Some 40 temporal query languages have been defined
- More recent language: TSQL2
  - Extension to SQL-92

Temporal Database: DBMS Implementation

- **Integrated approach:** internal modules of a DBMS are modified or extended to support time-varying data
  - Efficiency
- **Layered approach:** a software layer interposed between the user applications and DBMS that converts temporal query language statements to conventional statements
  - More Realistic for short and medium term

Spatiotemporal Database: Applications

- Three Types of Spatiotemporal Applications
  - 1. Applications may involve objects with continuous motion
    - Navigational systems manage moving objects
    - Objects change position, but not shape
  - 2. Applications dealing with discrete changes of and among objects
    - Objects' shape and their positions may change discretely in time
  - 3. Applications may manage objects integrating continuous motion as well as changes of shape
    - A "storm" is modeled as a "moving" object with changing properties (e.g., intensity) and shape over time

Spatiotemporal database: modeling requirements

- Need for representations of objects with position in space and existence in time
- Need to capture the change of position in space over time
  - Continuous change, or discrete change
- Need for the definition of attributes of space and organization of them into layers or fields
- Need to capture the change of spatial attributes over time
- Need to connect spatial attributes to objects
- Need for the representation of spatial relationships among objects in time
- Need for the representation of relationships among spatial attributes in time
- Need to specify spatiotemporal integrity constraints, imposed either by the user, or by the designer for integrity of the database

Spatiotemporal database: Querying

- Spatial operators(Faria1998)
  - NORTH(A,B)
  - AREA(A)
  - LENGTH(A)
  - DISJOINT(A,B)
- Temporal operators:
  - BEGIN(A), END(A)
  - T_BEFORE(A,B)
  - INTERVAL (start-time, end-time)

Spatiotemporal database: Querying (cont'd)

- Spatio-temporal Operators
  - Location-temporal Operator ST_SP(A, T)
    - Returns the spatial representations of object A valid at time T
  - Orientation-temporal Operators
    - Return a boolean value indicating whether there exists specific relationship between two objects (A and B)
    - ST_NORTH(A,B) or ST_EAST(A,B), etc
  - Metric-temporal Operators
    - The metric of object A at a time value T, ST_AREA(A, T)
    - Distance between two spatial components A and B at time T: ST_DISTANCE(A,B,T)
  - Topologic-temporal Operators
    - Return a boolean value indicating the topologic relationship between A and B during the time T: ST_DISJOINT(A, B, T)

Spatiotemporal database: Querying (cont'd)

- Querying examples

1. "Select the farms that contain electricity poles."

   $f \mid f \in \text{Farm} \wedge p \in \text{Pole} \wedge p.\text{type} = \text{"electricity"} \wedge \text{INSIDE}(SP(p), SP(f))$

```
select f
from f in Farms, p in Poles
where p.kind = "electricity" and inside(p->sp, f->sp)
```

2. "What was the area occupied by farms from 01/01/97 to 01/01/98?"

   $\text{ST\_AREA}(f, \text{INTERVAL}(01/01/97, 01/01/98)) \mid f \in \text{Farm}$

```
select tuple (farm: f,
              area:f->st_area(interval("01/01/97", "01/01/98")))
from f in Farms
```


Spatio-Temporal Database Systems Architecture

- Standard Relational DBMS with Additional Layer
  - Implement a spatiotemporal layer on top of a standard relational database
- Combination Architecture
  - Standard DBMS
  - Other storage components (such as file system) are used to store the spatial and temporal data and indexes
- Extensible DBMS
  - Object-Relational DBMS

Spatiotemporal database: open issues

- Database size
  - Spatial databases contain large amounts of information, temporal information further increases the database size
  - Increased difficulty of rapid data retrieval
- Legacy systems
  - Using STDB to Replace old systems & data
  - Building new STIS on existing SIS
  - Utilization of a data warehouse, enabling several legacy systems to be incorporated in a data-supply role
- Data quality
  - Errors exist in data gathering
  - discrete representation of numbers in computer
  - Temporal dimension further this problem

Summary

- Spatio-Temporal Information systems improve the existing spatial information system by handling temporal information.
- Most existing prototype systems are extensions of existing spatial systems
- Mainly for specific purposes (such as global change research)
  - It's unclear if a generic spatio-temporal information system will be commonly used
- Would be benefit from research in both spatial database and temporal database

References

1. Ralf Hartmut Guting, An introduction to Spatial Database Systems, VLDB Journal 3, 357-399 (1994)
2. Christian S. Jensen, Introduction to Temporal Database Research, *Temporal Database Management*, 2000.
3. Tamas Abraham and John F. Roddick, Survey of Spaio-Temporal Databases, GeoInformatica 3:1, 61-99 (1999)
4. Dieter Pfoser and Nectaria Tryfona: Requirements, Definitions and Notations for spatiotemporal Application environments. ACM GIS'98
5. Nectaria Tryfona and Christian S. Jensen, Conceptual Data Modeling for Spatiotemporal Applications, GeoInformatica 3:3, 245-268 (1999)
6. Glaucia Faria, Claudia Bauzer Medeiros, Mario A. Nascimento, An Extensible Framework for Spatio-Temporal Database Applications (1998)

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

~END~

Q&A

Januray 25, 2005

Jiyong Zhang(jiyong.zhang@epfl.ch)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
School of Computer and Communication Sciences
CH-1015 Lausanne, Switzerland

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE