AN OBJECT-ORIENTED MODELING AND IMPLEMENTATION OF SPATIO-TEMPORAL KNOWLEDGE DISCOVERY SYSTEM

K.Venkateswara Rao[1], Dr.A.Govardhan[2] and Dr.K.V.Chalapati Rao[1]

[1]Department of Computer Science and Engineering, CVR College of Engineering,
Ibrahimpatnam RR District, Andhra Pradesh, India
kvenkat.cse@gmail.com, chalapatiraokv@gmail.com
[2]JNTUH College of Engineering, Jagityala, Karimnagar Dist, Andhra Pradesh, India
*mailto:*govardhan_cse@yahoo.co.in

ABSTRACT

Spatiotemporal databases deal with changes to spatial objects with time. The Spatiotemporal knowledge discovery process involves the integration of such database systems with data warehousing, data mining and online-analytical processing technologies. The applications in this domain will process spatial, temporal and attribute data elements and use spatio-temporal relationships among these elements. These applications deal with the evolution of spatial objects and changes in their topological relationships, associations along with time. These advanced database applications require storing, management and processing of complex spatiotemporal data. In this paper we consider the classification and modeling requirements for spatiotemporal applications, System modeling, preprocessing spatiotemporal data, discovering spatiotemporal topological relationships and extension of apriori algorithm for mining spatiotemporal frequent predicates. The design of an appropriate system that can capture spatiotemporal features of datasets is discussed. Prototype implementation of the system is carried out on top of open source object relational spatial database management system called postgresql and postgis. The algorithms are experimented on historical cadastral datasets which are created using OpenJump. The results that are visualized using OpenJump software are presented.

KEYWORDS

spatiotemporal database, spatiotemporal application, spatiotemporal data modeling, spatiotemporal relationships, spatiotemporal data mining, spatiotemporal data analysis.

1. INTRODUCTION

Spatio-Temporal applications like climate change modeling and analysis[1], transportation systems, forest monitoring[2], diseases spreading[3], temporal geographic information systems[4] and environmental systems[5] process spatial, temporal and attribute data elements for knowledge discovery. The spatial objects are characterized by their position, shape and spatial attributes. Spatial attributes are properties of space and spatial objects located in specific positions inherit these attributes. Spatial attributes refer to the whole space and can be represented as layers, each layer represent one theme. The temporal objects are characterized by two models of time that are used to record facts and information about spatial objects. The two models of time are time points and time intervals. A time point is considered as one chronon, while a time interval has duration and is defined as set of chronons. Time points and time intervals can represent valid or transaction time. Valid time shows when a fact is true. There are two basic facts, events and states, for which time is recorded. An event occurs at an exact time

DOI : 10.5121/ijcsit.2011.3205

point, i.e., an event has no duration. Example events are "car crash," "sunrise," etc. A state is defined for each chronon in a time interval, hence it has duration. For example, a "meeting" takes place from 9am until 11am.

Spatiotemporal object captures simultaneously spatial and temporal aspects of data and deal with geometry changing over time. It can be represented by a four tuple - object id, geometry, time and attributes[6]. Recording a spatial object at a time point results in a snapshot of it. For example, capturing snapshots of a "landparcel" that changes its shape (e.g., split, expanded etc) during certain period of time. Recording a spatial object in a time interval is translated into capturing its evolution over time, i.e., capturing the possible change*s* of its shape over time. Consider the example of recording a "landparcel" in [2006, 2009], which changes shape.

Recording a layer at a time point results in a temporal layer of it. For example, the recording of the "vegetation" (of an area of interest) in March 25, 2009, gives a specific vegetation map of that area. Recording a layer in a time interval is translated into capturing its evolution over time, i.e., capturing the possible changes of it. Consider the example of observing the map of "vegetation" (of an area of interest) in [2000, 2009]. One could see the discrete changes of a layer during a time interval as snapshots of it, taken in time points of this time interval.

Modeling spatiotemporal applications is a complex task involving intricate issues[7] such as the representation of geometry of objects as well as its change in time, and spatial attributes that change their values depending on specific locations in time periods.
Topology describes spatial relationships like intersects, touches etc, between spatial objects. The spatial objects may be point, line or polygon. Different types of spatial topological relationships between two objects are shown in fig 1.

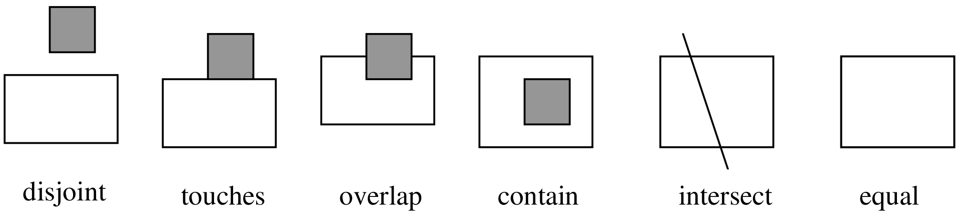


Fig 1: The set of spatial topological relationships

But spatiotemporal topological relationships are far beyond this. Based on the semantics of spatiotemporal changes in the real world, they are classified into two categories[8].

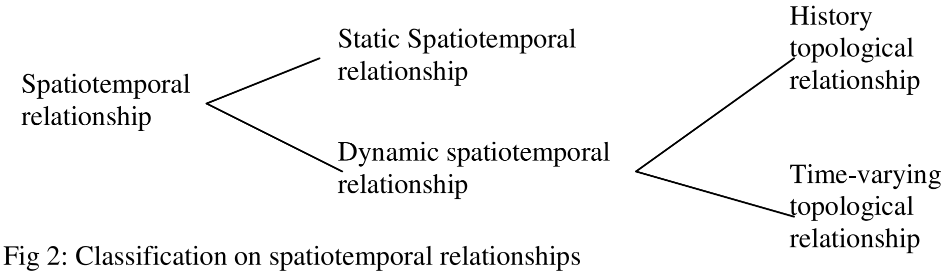
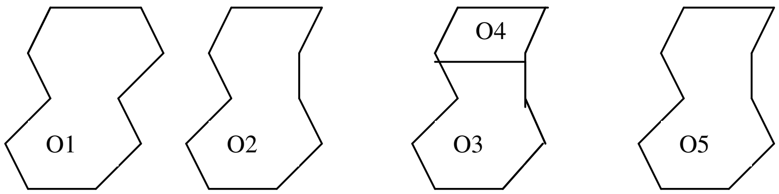

Spatiotemporal relationship
Static Spatiotemporal relationship
Dynamic spatiotemporal relationship
History topological relationship
Time-varying topological relationship

Fig 2: Classification on spatiotemporal relationships

The static spatiotemporal relationship between two spatiotemporal objects, which is at a certain time instant, refers to the spatial topological relationship between these two objects at that time instant. It can be any one of the topological relationships shown in fig1. Dynamic spatiotemporal relationships refer to the topological relationship of spatiotemporal objects along the time line. This is mainly because a spatiotemporal object has life and topological changes. History topology is the spatiotemporal relationship between a spatiotemporal object and its "parents and children". It means where and how a spatiotemporal object comes from, and its life history. The history topology of land parcel objects from time t1 to t4 is shown in fig2. Time-varying topology is the changing history of spatial topological relationship between two spatiotemporal objects in a given time interval. It is shown in fig4 for two objects in a time interval [t1, t4].



Object O1 at time t1

O1 changes at time t2

O2 splits at time t3

O3 and O4 merges at t4

Fig 3: History topology among land parcels in an interval [t1, t4]

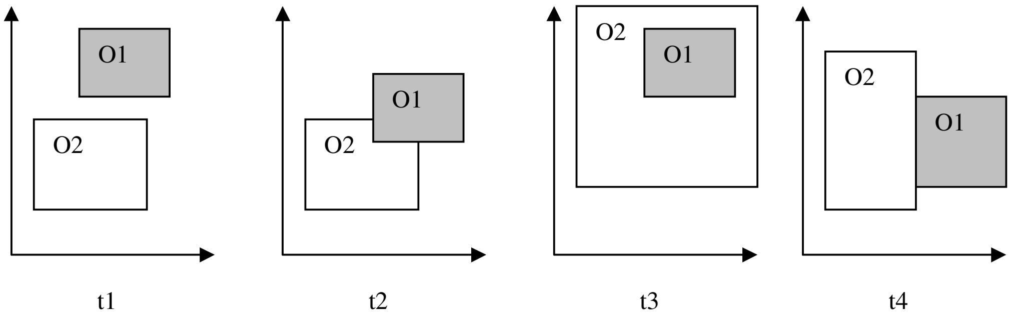


Fig 4: Time-varying Topology in an interval [t1, t4]

Spatiotemporal data mining refers to the extraction of knowledge, spatiotemporal relationships, or other interesting patterns not explicitly stored in spatiotemporal databases. It is an emerging area dedicated to the development and application of novel computational techniques for the analysis of very large spatiotemporal databases. It presents a number of challenges due to complexity of geographic domains and mapping of all data values into spatial and temporal frameworks.

The purpose of this research is to model and design an extendable object-oriented model, developing a methodology to discover spatiotemporal relationships and mining association rules from spatiotemporal data bases. When a spatiotemporal dataset D and the set of properties T of the dataset in both space and time are given, a spatiotemporal association rule takes the form X=>Y (s%, c%, $[t_s,t_e]$) where X, Y belong to T ,X and Y are disjoint, X = P1ΛP2 Λ … ΛPm, Y=Q1ΛQ2 Λ … ΛQn and at least one of the predicates P1,P2, … ,Pm, Q1, Q2, … , Qn is a spatial predicate such as adjacent, inside, intersects etc. s% and c% are support and confidence of the rule respectively. The Support of the rule X=>Y is the probability of (XUY) and the confidence is conditional probability of (Y/X), t_s and t_e are starting and ending time of the time interval in which the rule is valid.

2. SPATIOTEMPORAL APPLICATIONS CLASSIFICATION & REQUIREMENTS

2.1 Classification

Spatiotemporal applications can be classified into three major categories[9] based on the type of data they manage.

- Applications dealing with location change of spatial objects with time. In this type of application, objects change position, but not shape. For example, Navigational System, moving automobile on a road network.
- Applications dealing with shape change of spatial objects with time. For example, change of geometry of “land-parcels” or “rivers” in cadastral information system.
- Applications dealing with both location and shape change of spatial objects with time. For example, a ‘storm’ or a cyclone is measured as a moving phenomenon, which changes its location and shape over time.

2.2 Spatiotemporal Modeling Requirements

spatiotemporal information management poses new modeling requirements. The list of spatiotemporal applications requirements, at the modeling level, given in [9] are extended to cover knowledge discovery process and discussed below.

- Need for representation of objects with position in space and existence in time. An example is "waterpipes" occupying certain parts of space at certain time point in utility management information system.
- The need to capture the change of position in space over time. If the change of object’s position is continuous, then it results into motion. For example, in a navigational system the continuous change of a “vehicle’s” position. If the change is discrete, then it results into snapshots, versions of the object. An example is a “land parcel” having a position in space at some point in time. When it changes shape (e.g., a new part of land is attached to it) its position changes.
- Need for the definition of spatial attributes in time and organization of them into temporal layers or fields, i.e., snapshots of thematic maps. For example, "soil erosion" is a property of space organized in a layer, representing sets of regions (with different values). Spatial attributes can be visualized as continuous (e.g., "temperature") or as discrete ("soil type").
- Need to capture the change of spatial attributes over time. The changes□ can be discrete (e.g., changes on a map of "land parcels" or "vegetation") or continuous (change of "temperature").
- □Need to connect spatial attributes to objects. For example, a land-parcel has “soil type” as an attribute. The “soil type” is an attribute of space and land-parcels inherit part of it.
- Need for the representation of different spatiotemporal topological relationships among spatial objects in time for analysis.
- □Need for the representation of relationships among spatial attributes in time. For example, the “soil type” is a result of the combination of the “acidity” and the “corrosively” of soil.
- Need to specify spatiotemporal integrity constraints, imposed either by the user, or by the designer for the integrity of the database.
- Need to represent the orientation and direction features of spatio-temporal objects during their change.

- Need to represent events and list of all changes associated with each event. An event may represent abrupt change or it may have duration.
- Need for the representation of multiple granularities for spatio-temporal objects. For example, when tracing modifications to spatial areas, the history of the areas under observation has to be maintained and retrieved at multiple temporal granularities (e.g., years, months, decades).
- Need for representation of spatiotemporal data in multi-dimensional model for analytical processing.
- Need for the representation of concept hierarchies for the dimensions.

3. ARCHTECTURE AND DESIGN OF THE SYSTEM

3.1 Architecture

The architecture of the system that meets the modeling requirements for spatiotemporal data analysis is given in fig 5. The spatiotemporal database can be created and managed using postgresql [10] and postgis [11] server. The mapping software modules translate database from relational structure to object oriented structure vice versa. Data processing, analysis and mining involve set of software modules which are developed using spatial data processing capabilities of postgis. OpenJump[12] is used to create data sets for testing as well as visualization of the results stored in the database.

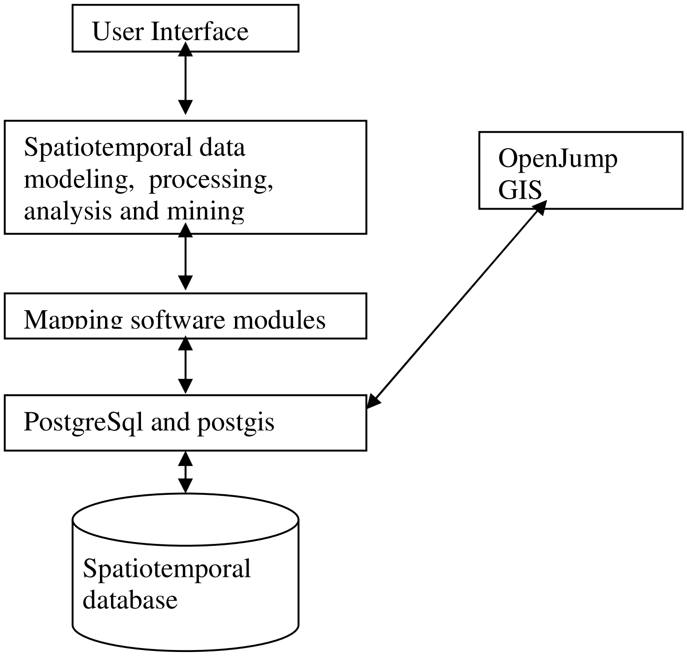
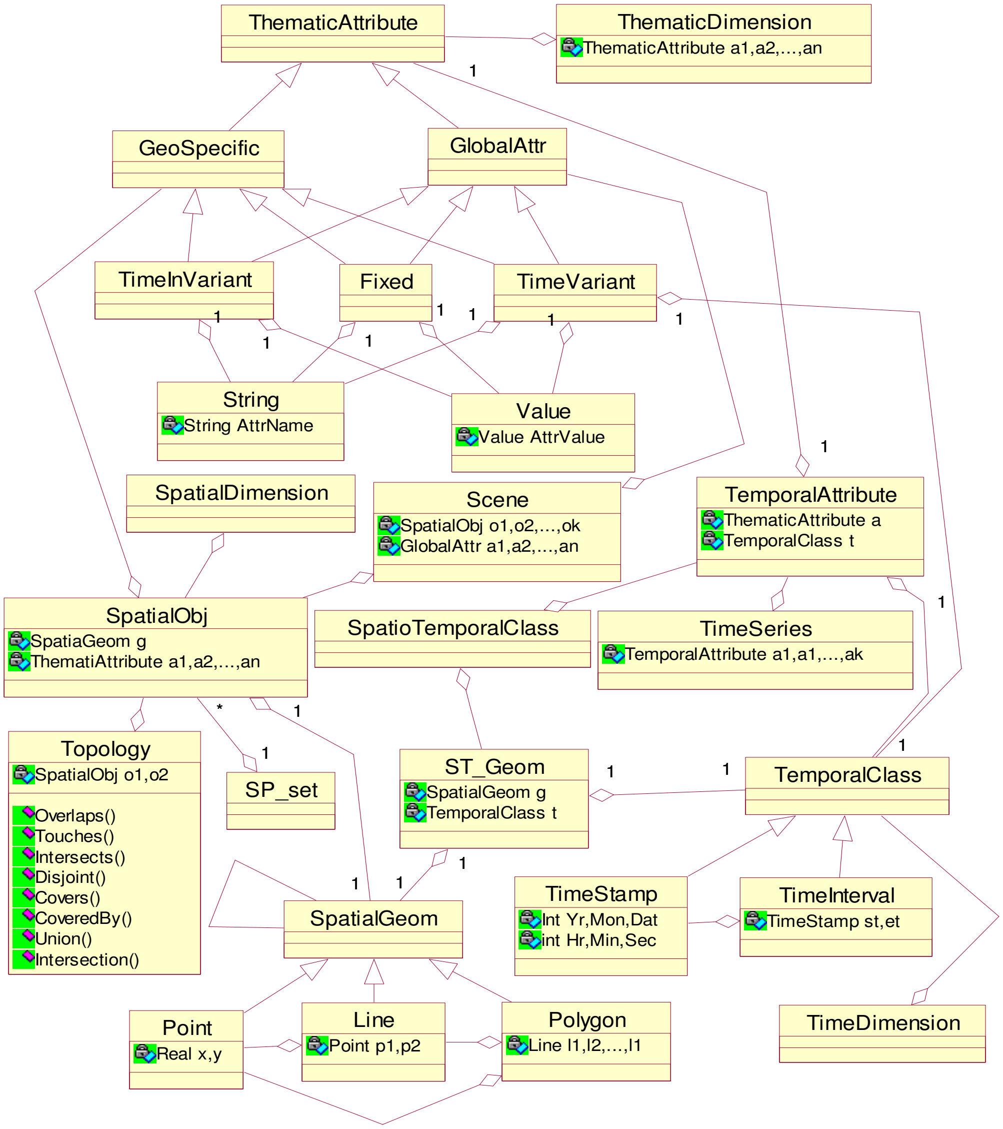


Fig 5: Architecture of the system

3.2 Object Oriented Model

The requirements are analyzed to identify different classes, attributes and methods of each class, relationships among the classes and general methodology for different kinds of spatiotemporal analysis. The class diagram that can be extended to any application domain by adding domain specific classes is shown in fig 6.

The spatiotemporal applications involve geo-referenced objects and phenomena (any observable occurrence or event) changing over time. The spatiotemporal phenomenon is the aggregation of spatiotemporal class, event class and process class. Each of these classes is an aggregation of spatial, temporal and thematic attribute classes. The spatial class manages location and shape of the spatial objects and thematic attribute class deals with the features of the objects. The temporal class maintains time attributes of the objects. The event class records the causes of the changes to the spatial object and the process class deals with the effect of changes to the object. Both event class and process class have location, time attributes and other features which are specific to the kind of the event and the process. For example an event rainfall over a period of time may have caused the spatial object like a land parcel to change its shape and also its surface features like crop or vegetation. The event rainfall may have caused process flood at particular location for some time. This spatiotemporal phenomenon can be analyzed by applications developed using the classes which are described in fig 6.



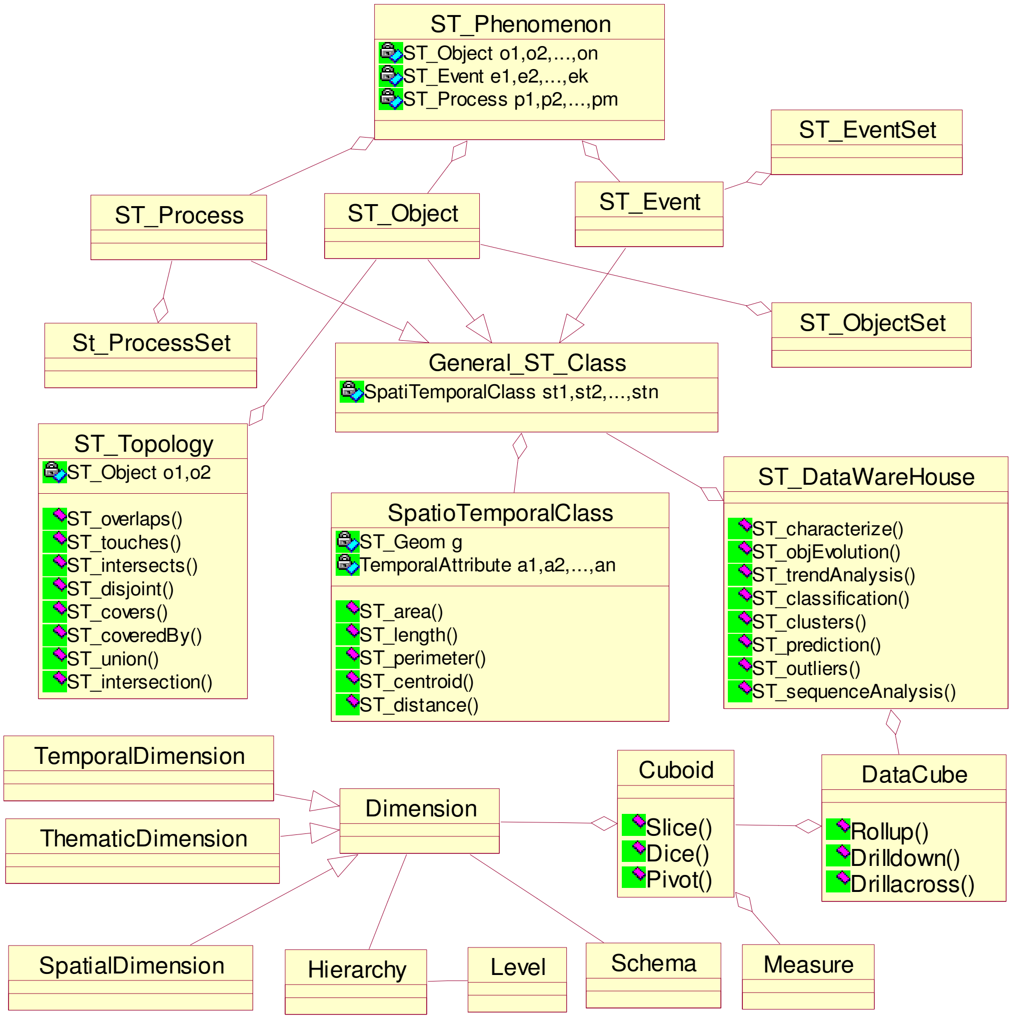


Fig 6: Class diagrams of the system

The knowledge discovery process involves multidimensional modeling, analysis and mining of the spatiotemporal data at different levels of abstraction. The multidimensional model views data in the form of a data cube which can be constructed using a set of dimensions and a set of measures. The dimension provides semantic information and it consists of a set of attributes, schema, hierarchy and levels. The schema is in a form of tree structure that specifies the overall structural relationship among the attributes of the dimension. The hierarchy is a path from root to a leaf node in the schema and it contains a set of levels. The schema can have one or more hierarchies. The level in the hierarchy can be used to indicate a different degree of abstraction of the objects. The measures are the functions which compute the values of cells in the multidimensional model. A data cube can also be viewed as a set of cuboids where each cuboid is obtained by a groupby operation on dimensions of the multidimensional model at a particular level of the dimension hierarchy of each dimension. The multidimensional model is the basis on which a data warehouse and many data analysis and mining tools can be developed. The spatiotemporal data mining tasks[13,14] include characterization, classification[15], trend prediction, clustering, outlier analysis and frequent pattern analysis.

3.3 Spatiotemporal database design:

An object relational spatiotemporal database which consists of a set of tables and relationships among them is designed to facilitate the spatiotemporal data analysis and mining. Different sorts of spatiotemporal data to be handled are states, events and episodes. A state represents a version of an entity in a given moment. States can consist of different versions of an individual entity. An event is the moment in time an occurrence takes place. Event causes one state to change to another. An episode is the length of time during which change occurs, a state exist or an event lasts. Two main strategies to represent multiple versions of an object are tracking the versions either at the level of objects or attributes. First one involves a different identifier (oid) to each new version and chaining the older versions to new oid. Second one involves a single object identity (oid) with versions actually associated with attributes. The attributes of spatiotemporal objects can be categorized as version significant, non-version significant and invariant. The version significant attribute values are to be updated in non-destructive manner, the non-version significant attribute values are to be updated in a destructive manner and invariant attributes values are not allowed to be changed. Following entities are designed to address these requirements.

1. Temporal_tab : This table stores timestamps which correspond to time at which change to any spatial object has taken place.
2. Spatial_obj_tab : This table contains spatiotemporal objects with unique identifier, geometry and existence time which has from time(f_t) and to time(t_t) as attributes. It also has other attributes to indicate category and type of spatial object, type of change. The events and episodes or processes are also considered as spatiotemporal objects and stored in this table.
3. split_tab: This composite entity is used to record splitting of any spatial object into multiple objects. It has object identifier that got split and new object identifiers for objects derived due to split and timestamp attribute that records time of split.
4. Merge_tab: This composite entity keeps the data related to merging of two or more objects into single object. It maintains object identifiers which are merged, new object identifier for object derived due to merge and timestamp attribute that records time of merge. The new objects created due to split or merge are stored in spatial_obj_tab with their new object identifiers.
5. Geom_version : This table records geometry changes by creating new object identifier for each change to the geometry of the object. It has oid of the object changed, new object identifier and timestamp attribute that records time of the change.The new object is stored in spatial_obj_tab table.
6. VsAttr_tab: This table manages version significant attributes of all objects in spatial_obj_tab. It has oid, attribute name, timestamp and attribute value as its fields.
7. VinAttr_tab: This table manages version insignificant and invariant attributes of all objects in spatial_obj_tab. It has oid, attribute name and attribute value as its fields.
8. Result_tab : This entity is used by analysis algorithms to store results back into database. This table can be accessed using OpenJump (An Open source GIS software) to visualize the results.

4. KNOWLEDGE DISCOVERY METHODOLOGY AND IMPLEMENTATION

4.1 Methodology

The knowledge discovery considered in this paper are discovering evolution of spatial objects with time, changes in topological relationships between pairs of spatial objects with time, frequent patterns involving topological predicates and spatiotemporal association rules[14,16].

This requires preprocessing the database objects and mapping it to spatiotemporal data structures. The algorithms designed for this purpose are described below.

4.1.1 Algorithms

```
Algorithm 1: Tracking Spatial object or History Topology
Input: Spatiotemporal dataset (D),Spatial Object Identifier (oid), from time(f_t) and to time(t_t)
Output: The Object (oid) details changing with time.
Method: Process(oid)
Begin
  Obj = getobjdetails(oid) /* connect to dataset D get spatial object details and load them into variable  obj */
  Display(Obj)
  If ( Obj.et  <  t_t) Track(Obj)
End

Display(Obj)
Begin
 If ( Obj.st > f_t and Obj.et < t_t )
   Print or save id, geom, st, et, area, centroid and perimeter of Obj. The attribute ChangeType of Obj indicates type of change the Obj has undergone at time et.
 Elseif ( Obj.st > f_t and Obj.et >= t_t )
   Print or save id, geom, st, t_t, area, centroid and perimeter of Obj.
 Elseif ( Obj.st <= f_t and Obj.et >= t_t )
   Print or save id, geom, f_t, t_t, area, centroid and perimeter of Obj.
 Elseif ( Obj.st <= f_t and Obj.et < t_t )
   Print or save id, geom, f_t, et, area, centroid and perimeter of Obj. The attribute ChangeType of Obj indicates type of change the Obj has undergone at time et.
 Else Print " Obj is not valid between f_t and t_t"
End

Track(Obj)
Begin
Next = Obj.changeType
Switch(Next)
Begin
Case C:
        Look into Geom_version table and find new object identifier (n_oid).
        Process(n_oid)
        Break;
Case S:
        Look into split_tab and find list L of identifiers of objects which are result of split of the Obj.
        For each object identifier e_oid in L, Process(e_oid)
        Break
Case M:
        Look into Merge_tab and find the new object identifier (n_oid) which is the result of merge of Obj with some other spatial object.
        Process(n_obj)
        Break
Case default: break
End
End
```

Algorithm 2: Finding change in topological relationships, intersection, Union, distance between two objects.
Input: : Spatiotemporal dataset (D), Object Identifier (oid1, oid2), from time(f_t) and to time(t_t)
Output: Topological relationships, Intersection, Union, distance changes between the oid1 and oid2 from f_t to t_t
Method :

```
Begin
    1. Using Algorithm1,  track the objects oid1, oid2 and record all time points at which either oid1 or oid2 or their siblings have changed between f_t and t_t. Also use special data structure that manages valid identifiers of the objects for each of the time points.
    2. For each of the time points, create an object of type GeometryRelation and use its methods to compute topological relationships, intersection, union and distance between the relevant pair of objects which are valid for the time point.
    3. Display or store the results for the given duration.
End
```

```
Public class GeometryRelation
{
        PGgeometry obj1, obj2;
        Methods:
        PGgeometry Union();
        PGgeometry Intersection();
        Float Distance();
        Boolean isintersects()
        Boolean istouches()
        Boolean isequals()
        Boolean isdisjoint()
        Boolean iscrosses()
        Boolean isoverlaps()
        Boolean iscovers()
        Boolean iscoveredby()
}
```

All the methods of GeometryRelation class can be implemented using Postgis application programming interface.

Algorithm 3: Mining spatiotemporal frequent itemsets and association rules
Input: spatiotemporal database (D), Target location geometry(tlg), two types of object type(c1,c2), from time (f_t), to time (t_t), minimum support (m_s), minimum confidence (m_c)
Output: Association rules of form : X=>Y(s%,c%,$[t_s,t_e]$)
Method:

```
Begin
ST_set1={Obj/ Obj belongs to D and Obj.geom inside tlg, and Obj.type = c1 and (f_t<= Obj.ts<=t_t) and (f_t <=Obj.te<=t_t)}
ST_set2={Obj/ Obj belongs to D and Obj.geom inside tlg, and Obj.type = c2 and (f_t<= Obj.ts<=t_t) and (f_t <=Obj.te<=t_t)}
TP_set= { t/t belongs to temporal_tab table and (f_t <= t <= t_t )}
Targetdb = preprocess(St_set1,ST_set2,TP_set)
ST_minedb = GenerateTemporalDataset(TargetDB)
For each distinct value t of temporal_field of ST_minedb /* the distict value may be time point or time interval */
  Begin
```

```
        Task_relevant_db = { obj / obj belongs to ST_minedb and Obj.temporal_field = t }
        L = Frequent_item_set= apriori(Task_relavant_db,m_s,m_c)
        Generate strong association rules for every frequent itemset. /*rules of form x=>y(s%,c%) generated here */
        Asscociate t to every association rule. /* rules becomes x=>y(s%,c%,[$t_s$,$t_e$]) in this step */
  End
End
```

```
Algorithm 4: preprocess(ST_set1,ST_set2,TP_set)
Begin
For each t belongs to TP_set
  For each object i belongs to ST_set1
    For each object j belongs to ST_set2
        Begin
            tp=Topology(i,j)
            StoreInDB(i,j,t,tp)
        End
End
```

4.2 Implementation

The system is implemented using postgresql, postgis, Java, JDBC and OpenJump technologies. The object relational database is created and used in prototype implementation of the system for historical cadastral data analysis, static and time-varying spatiotemporal topological relationships discovery and frequent predicate analysis. The versions created due to split and merge processes or evnts are managed by considering each version as a new object. The linkage between parent and children objects is maintained using primary key foreign key relationship among the appropriate tables. Following is an example of table creation.

```
create table spatial_obj_tab(spobjid integer, category integer, st timestamp, et timestamp, objtype varchar(10),Changetype interger);
select AddGeometryColumn('spatial_obj_tab','objgeom',-1,'POLYGON',2);
```

The sample database objects are generated using OpenJump and loaded into the database. The modules for accessing the database, mapping the database object structure to spatiotemporal object structure and the designed algorithms are implemented in JAVA. The application programming interface provided by the postgis is used for all geometric related computations. The modules for loading the results back into the database for effective visualization using OpenJump software are developed.

5. RESULTS

Case 1: Results for History Topology: Single Object Varying With Time:
Given object Id : 91
Starting time : 2000-01-01 12:12:12
Ending time:2000-12-31 12:12:12
Output:
The object 91 is changed to 151 at “2000-01-15 12:12:12”. Then the object 151 is changed to 173 at “2000-04-15 12:12:12” and the object 173 is changed to 224 at “2000-08-26 12:12:12”.

```
Object Id=91:
Ending time=2000-08-26 12:12:12.0
POLYGON((526 411,540 420,560 420,560 400,526 411))
```

Area: 430.0
centroid: POINT(547.2713178294573 411.72868217054264)
Perimeter: 92.378456

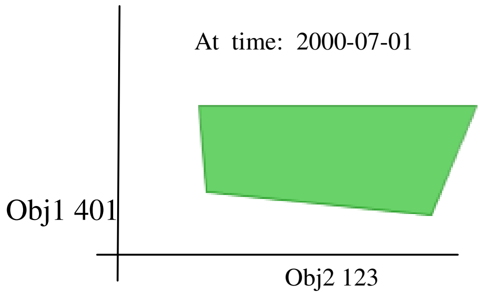
Object id=151:
Ending time=2000-04-15 12:12:12.0
POLYGON((526 411,540 420,555.6048020833342 411.7092239583335,560 400,526 411))
Area: 303.14026
centroid: POINT(544.3809692703338 410.3903727435555)
Perimeter: 82.55591

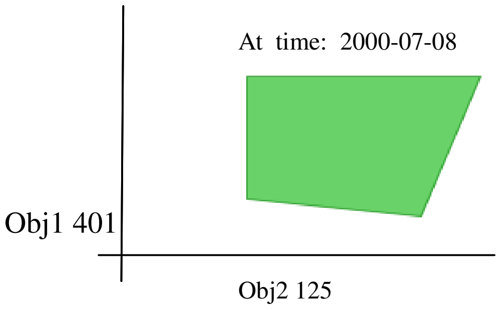
Object id=173:
Ending time=2000-08-26 12:12:12.0
POLYGON((526 411,540 420,555.9061650065141 417.272132796836,560 400,526 411))
Area: 361.783
centroid: POINT(545.6312040877306 411.0948951800529)
Perimeter: 86.2675

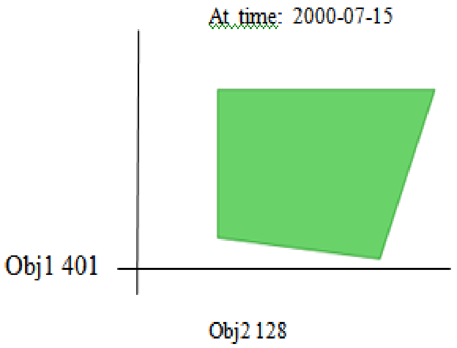
Object id = 224:
Ending time=2000-12-31 12:12:12.0
POLYGON((526 411,540 420,554.1993333333334 429.72733333333343,560 400,526 411))
Area: 469.26666
centroid: POINT(546.7926981736673 413.5161907152215)
Perimeter: 99.87812

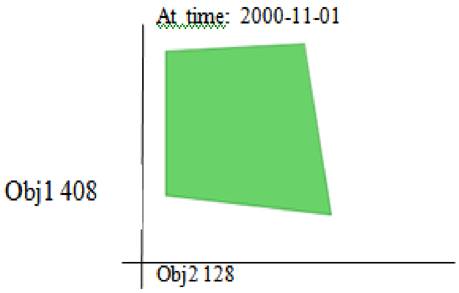
Case 2: Find the change in intersection of two given object geometries from t1 to t2.
Given object1 Id : 401
Object2 Id : 401
Starting time : 2000-07-01 12:12:12
Ending time:2000-11-10 12:12:12
Output:
Geometry for object 401 is :
POLYGON((509.3403692762187 193.5578729689808,500.75703999999996 305.7642400000001,660.4122933333333 394.80274666666674,703.3964 239.75293333333337,618.9633333333334 60.140773333333314,509.3403692762187 193.5578729689808))
Geometry for object 123 is:
POLYGON((500 240,580 240,568 211,480 220,480 240,500 240))

The intersection of area of objects 401 and 123 at different points of time is visualized using OpenJump and is shown below.









Case 3: Results of static spatiotemporal topological relationships:

Given Two objects 4 and 5 and timestamp as input to algorithm. It generates following output.

Obj1	Obj2	intersects	contains	equals	touches	disjoint	overlaps	time
4	5	0	0	0	1	0	0	2000-01-08 12:12:12

Case 4: Results of time-varying spatiotemporal topological relationships:

Given Two objects 45 and 83 and two timestamps as input to algorithm. It generates following output. The object changes are tracked by creating new object in database for each change to its geometry.

Obj1	Obj2	intersects	contains	equals	touches	disjoint	overlaps	From time	To time
45	83	0	0	0	0	1	0	2000-01-08 01:10:15	2000-01-08 09:10:15
45	85	0	0	0	1	0	0	2000-01-08 09:15:15	2000-01-09 01:10:15
45	95	0	0	0	0		1	2000-01-09 01:15:15	2000-01-10 01:10:15
54	95	0	0	0	0	1	0	2000-01-10 01:15:15	2000-03-10 01:10:15

The results can be used in applications like detection of forestation & deforestation, estimation of flood damage by using spatiotemporal overlay analysis with forest geometry and flood geometry respectively.

Case 5: Results of apriori algorithm :

Given two categories of spatial objects are Landparcel and Floodregion
These results consist of frequent predicates which are satisfying minimum support. Only few results are provided.

```
Landparcel,Floodregion,overlaps:122, 2000-08-01 12:10:15
Landparcel,Floodregion,coveredby:39, 2000-08-01 12:10:15

Landparcel,Floodregion,overlaps:180, 2000-08-03 12:10:15
Landparcel,Floodregion,coveredby:78, 2000-08-03 12:10:15

Landparcel,Floodregion,overlaps:98, 2000-08-05 12:10:15
Landparcel,Floodregion,coveredby:23, 2000-08-05 12:10:15

Landparce,Floodregion,overlaps:180, 2000-08-01 12:10:15, 2000-08-03 12:10:15
Landparce,Floodregion,coveredby:78, 2000-08-01 12:10:15, 2000-08-03 12:10:15

Landparcel,Floodregion,overlaps:98, 2000-08-03 12:10:15, 2000-08-05 12:10:15
Landparcel,Floodregion,coveredby:23, 2000-08-03 12:10:15, 2000-08-05 12:10:15

Landparcel,Floodregion,overlaps:98, 2000-08-01 12:10:15, 2000-08-05 12:10:15
Landparcel,Floodregion,coveredby:23, 2000-08-01 12:10:15, 2000-08-05 12:10:15
```

The association rules generated are of type

```
LandParcel,Floodregion => overlaps (10%,60%,[2000-08-01 12:10:15, 2000-08-05 12:10:15])
LandParcel,Floodregion => overlaps (10%,30%,[2000-08-01 12:10:15])
```

6. RELATED WORK:

A Dynamic Object-Oriented model proposed by Yangming JIANG et al [17] considered spatiotemporal class as a super class of four classes, ZeroTObject(ZTO), OneTObject(OTO), TwoTObject(TTO), ThreeTObject(THTO) where ZTO is a temporal node, OTO is a temporal arc, TTO is a temporal polygon and THTO is a temporal cube. This model is used to simulating changes in digital earth. The spatiotemporal data model proposed by ZHANG Ruiju et al [18] is based on space-time composite model in which spatiotemporal object has a unique identifier irrespective of changes it experiences. S.Nadi and M.R Delavar [19] reviewed the spatiotemporal models and proposed a five table database schema as a general spatiotemporal database structure. Nikos pelekis et al [6] provided extensive literature review of various spatiotemporal database models. The problems in spatiotemporal data model are discussed and a general object-oriented spatial Temporal model is proposed by Bonan Li and Guoray Cai [7]. Different approaches to representing changes of geographical phenomena for analyzing and tracking the evolution of objects are discussed by Southeil khaddaj et al [20].

There has been work on spatial association rules [21] and temporal association rules [22] but very little work has addressed both spatial and temporal dimensions. The work by Tao et al [23] addressed the problem of spatiotemporal association rules in spatiotemporal domain. They consider association rules of the form $(r_i,T,p) \Rightarrow r_j$ with the interpretation " If an object is in region r_i at some time t, then with probability p, it will appear in region r_j by time (t + T)". When interpreted in traditional sense, p is really the confidence threshold of the rule and there is no support defined. That is the number of objects for which the rule is applied is ignored. Our research work addressed the object-oriented modeling of spatiotemporal knowledge discovery which includes modeling, analysis and mining of spatiotemporal databases.

7. CONCLUSION AND FUTURE WORK

In this paper, we investigated the problem spatio-temporal knowledge discovery and the role of computer science in terms of the requirements that determine the priorities of modeling spatio-temporal applications, modeling spatial and temporal dimensions independently and incorporating them into the database design. Due consideration is given in modeling, design, representation and implementation for spatio-temporal objects and their topological relationships discovery. The research focused on high level abstraction of various data mining tasks. Accordingly, an appropriate architecture is described, and object-oriented modeling for representing the structure of spatio-temporal knowledge discovery and the database design are discussed. The system is implemented using open source software postgresql, postgis and OpenJump. The results of different analysis like history topology, time-varying topological relationships and association analysis for historical cadastral database are provided. The system can be extended to provide multidimensional analysis at multiple granularities and mining different kinds of knowledge patterns such as clustering, sequence analyses, classification and multi-level association rules.

REFERENCES

[1] Auroop R Ganguly and Karsten Steinhaeuser, (2008) ”Data Mining for climate change and impacts”, *IEEE international conference on data mining workshops,ICDMW*,15-19,Dec,2008,Italy.

[2] T. Cheng and J. Wang, (2006) “Applications of spatio-temporal data mining and knowledge discovery (STDMKD) for forest fire prevention”, *ISPRS Commission VII Mid-term Symposium "Remote Sensing: From Pixels to Processes*, Enschede, the Netherlands, 8-11 May 2006

[3] Diego Ruiz-Moreno, Mercedes Pascual, Michael Emch, Mohammad Yunus, (2010) “Spatial clustering in the spatio-temporal dynamics of endemic cholera“, *BMC Infectious Diseases*, Volume 10, 2010.

[4] Yang ping , Tang Xinming , Wang Shengxiao, (2008) “Dynamic cartographic representation of Spatio-Temporal data“, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVII. Part B2. Beijing 2008

[5] Z. Obradovic , D. Das, V.Radosavljevic, K.Ristovski, S.Vucetic, (2010) “Spatio-Temporal characterization of aerosols through active use of data from multiple sensors, “*ISPRS TC VII Symposium – 100 Years ISPRS*, Vienna, Austria, July 5–7, 2010

[6] Nikos Pelekis et al, (2004) “Literature Review of Spatio-Temporal Database Models“, *The Knowledge Engineering Review*, 2004, 235-274 Cambridge university Press.

[7] Bonan Li and Guoray Cai, (2002) ”A General Object-Oriented Spatial Temporal Data Model”, *Symposium on Geospatial Theory, Processing and Applications*, Ottawa, 2002.

[8] Peiquan Jin and Lihua Yue, (2006) ”A Framework for the Description of Spatiotemporal Relationships”, *Proceeings of the Joint conference on information Sciences*, October, 2006.

[9] Dieter Pfoser and Nectaria Tryfona, (1998) ”Requirements, Definitions, and Notations for Spatiotemporal application environment” *, GIS’98 Proceedings of the 6th ACM international symposium on Advances in geographic information systems*, ACM Digital Library.

[10] http://www.postgresql.org/docs/manuals/

[11] http://postgis.refractions.net/documentation/

[12] http://www.openjump.org/

[13] Hao-En Chueh, (2010) “Mining Target-Oriented Sequential Patterns with time-intervals*”, International journal of computer science & information Technology (IJCSIT*) Vol.2, No.4, August 2010.

[14] Irshad Ullah, (2010) “Data Mining Algorithms And Medical Sciences”, *International Journal of Computer Science & Information Technology (IJCSIT),* Vol 2, No 6, December 2010.

[15] P.Bhargavi and S.Jyothi,, (2010) “Soil Classification Using GAtree”, *International journal of computer science & information Technology (IJCSIT)* Vol.2, No.5, October 2010.

[16] Sujni Paul, (2010) “An Optimized Distributed Association rule mining algorithm in Parallel and distributed data mining with XML data for improved response time”, *International Journal of Computer Science and Information Technology*, Volume 2, Number 2, April 2010.

[17] Yangming JIANG and Siwen BI, (2008) ”Dynamic Object-Oriented Model and its Applications for Digital Earth”, *Digital Earth Summit on Geoinformatics*, Nov,12-14,2008,Germany.

[18] ZHANG Ruiju et al, (2005) “An Object Oriented Spatio-Temporal Data Model”, *Proceedings of International Symposium on Spatio-Temporal Modeling, Spatial Reasoning, Analysis, Data Mining and Data Fusion*, 27-29, Aug,2005, peking University, China.

[19] S. Nadi and M.R.Delavar, (2005) “Toward a General Spatio-Temporal Database Structure for GIS applications”, *Proceedings of International Symposium on Spatio-Temporal Modeling, Spatial Reasoning, Analysis, Data Mining and Data Fusion*, 27-29, Aug,2005, peking University, China.

[20] Souheil Khaddaj,Abdul Adamu and Munir Morad, (2005) “Construction of an Integrated Object Oriented System for Temporal GIS”, *American Journal of Applied Sciences* 2(12), 2005, pp.1584-1594, ISSN 1546-9239.

[21] Salvatore Rinzivillo and Franco Turini, (2005) ”Extracting spatial association rules from spatial transactions”, *Proceedings of GIS’05 13th annual ACM international workshop on geographic information systems*,Germany.

[22] Schluter T and Conrad S, (2010) ”Mining Several kinds of Temporal association rules enhanced by Tree structures”, *Second international conference on Information,Process and Knowledge Management*,2010, Saint Maarten.

[23] Tao Y et al, (2004) ”Spatio-temporal aggregation using sketches”, *20th international conference on Data Engineering,IEEE*(2004) 214-225.

Authors

Mr. K. Venkateswara Rao is working as Associate professor in Computer Science and Engineering Department at CVR College of Engineering. He is pursuing his part time PhD from Jawaharlal Nehru Technological University. He received his M.Tech in Computer Science and Engineering from Osmania University. Previously he worked as Software Specialist at Wipro Technologies and as Scientist at ISRO Satellite Centre. His areas of interest includes Databases, Data Warehousing & Mining, Operating systems and Realtime Systems.

Dr.A.Govardhan: received Ph.D. degree in Computer Science and Engineering from Jawaharlal Nehru Technological University in 2003, M.Tech. from Jawaharlal Nehru University in 1994 and B.E. from Osmania University in 1992. He is Working as a Principal of Jawaharlal Nehru Technological University, Jagitial. He has published around 108 papers in various national and international Journals/conferences. His research of interest includes Databases, Data Warehousing & Mining, Information Retrieval, Computer Networks, Image Processing, Software Engineering, Search Engines and Object Oriented Technologies

Dr K.V. Chalapati Rao is a Professor of Computer Science & Engg., and Dean, Academics at CVR College of Engineering. Prior to joining the CVR, he served Osmania University as a Professor & Head, Department of CSE and Dean of Engineering. After obtaining his PhD, Dr. Rao joined Electronics Corporation of India Limited and worked in various capacities for 16 years, before joining the Osmania University. He guided number of PhD scholars in areas of Real time systems, Operating Systems, Software Engineering, Distributed Systems, Knowledge and Data Engineering.