

# AN ARCHITECTURE FRAMEWORK FOR SPATIOTEMPORAL DATA MINING SYSTEM

K.Venkateswara Rao<sup>1</sup>, A.Govardhan<sup>2</sup> and K.V.Chalapathi Rao<sup>1</sup>

<sup>1</sup>Department of Computer Science and Engineering, CVR College of Engineering, Ibrahimpatnam RR District, Andhra Pradesh, India

kvenkat.cse@gmail.com

chalapatiraokv@gmail.com

<sup>2</sup>JNTUH, Hyderabad, Andhra Pradesh, India

govardhan\_cse@yahoo.co.in

## ABSTRACT

*Spatiotemporal data mining is an emerging area of research. Applications for mining different kinds of spatiotemporal patterns and trends are being developed by researchers in various domains. However, there is a need for underlying architecture framework for these applications to provide reusability of analysis and design. This research paper addresses such a requirement by adopting conceptual model of architecture framework described in ISO/IEC 42010:201X standard for Systems and Software Engineering - Architecture Description. This research identifies stakeholders and their concerns, describes context, integration, information, functional, concurrency and technology viewpoints as architectural viewpoints which provide reusability of analysis and design for spatiotemporal data mining applications. The correspondence rules which describe correspondence among the viewpoints are also described.*

## KEY WORDS

*Architecture Framework, Spatiotemporal Data Mining, Viewpoints, Data Mining Issues*

## 1. INTRODUCTION

An interest to analyze data has grown significantly in recent years as businesses in all sectors have recognized the competitive advantages that just-in-time information can bring in the decision-making process. Additional features, such as huge volumes of historical data that many organizations own and better quality of software that manages this data have played significant role in increasing this interest too. A wide variety of applications need to capture spatial and time-varying attribute data of the entities they model. With the advent of location-based services and mobile computing, even more amount of spatial data is being collected and stored in business information systems. Therefore spatiotemporal data are becoming a corner stone for decision-makers to analyze business data in a spatial context and the exploitation of spatial data within the decision making process needs research. These huge collections of spatiotemporal data often hide possibly interesting information and valuable patterns and trends. It is obvious that a manual analysis of these data is impossible and data mining can provide useful tools and technology in this context. Data mining is partially automated search for hidden patterns in large and multi-dimensional databases. It involves integration of techniques from multiple disciplines such as database and data warehouse technology, machine learning, statistics, high performance computing, data visualization, pattern recognition, neural networks, spatial and temporal analysis.

The availability of gigantic volume of spatiotemporal data in future, often continually updated, poses great challenges to our ability to digest the data and to gain useful knowledge that would otherwise be lost [1]. With increase in the size of data sets, data mining has recently become an important area of research and is receiving substantial interest from both academia and industry. Spatiotemporal data mining is an emerging research area dedicated to the development and application of novel computational techniques for analysis of large spatiotemporal databases. The research community in spatiotemporal data mining in different areas is investigating new techniques for addressing the issues and challenges in their respective domains. Without an underlying framework, a data mining application is like a computer program without an operating system [2]. Without a proper framework, a considerable portion of the development time of data mining application will be spent in implementing functionality common to all applications in the data mining domain. However, no redundant work would be necessary if a sufficiently generic framework [1] that can be easily tailored to meet application specific needs is available. The benefits of using a framework for application generation [1] are numerous because a typical data mining application has to handle a large number of variables and transformations.

Architecture of a system is its fundamental structure that brings all the architectural elements of the system together. An architectural element is a fundamental piece from which a system can be considered to be constructed. An Architecture Framework expresses a fundamental paradigm for structuring software. It determines the basic structure of an application and can be viewed as a template for creating software architectures. According to ISO/IEC 42010:201X [3], Systems and Software Engineering – Architecture Description document, an architecture framework establishes a common practice for creating, interpreting, analyzing and using architecture descriptions within a particular domain of application or stakeholder community. The motivation of this definition is to provide a means of defining existing and future architecture frameworks in a uniform manner to promote sharing of information about systems, architectures and techniques for architecture description. The conceptual model of an architecture framework according to ISO/IEC FDIS 42010 is described in Fig. 1.

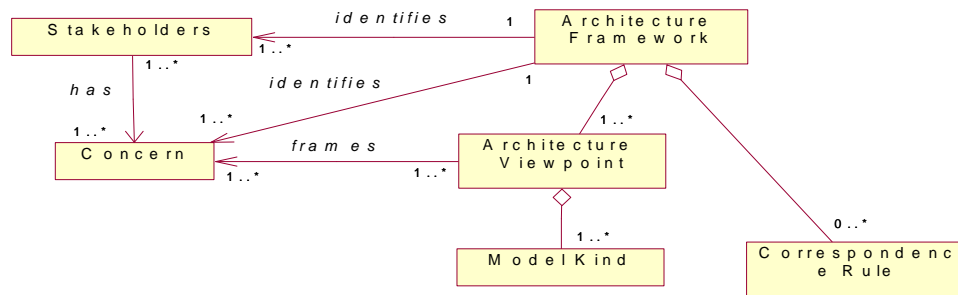


Fig. 1: Conceptual model of Architecture Framework

An Architecture framework shall include information identifying the architecture framework, the identification of one or more concerns, the identification of one or more stakeholders having those concerns, and one or more architecture viewpoints that frame those concerns and any correspondence rules.

A stakeholder in the architecture of a system is an individual, team, or an organization having an interest in the realization of the system. A concern about architecture is a requirement, an objective, an intention, or an aspiration a stakeholder has for that architecture. According to ISO/IEC FDIS 42010, a viewpoint refers to the conventions for expressing architecture with respect to a set of concerns. A viewpoint is a way of looking at systems and the application of a viewpoint to a particular system is referred as an architecture view. Views and viewpoints and their application to software architecture are elaborated in [4,5]. The reusable nature of

architecture viewpoints (and architecture frameworks, as coordinated viewpoint sets) highlights their utility as mechanisms for capturing strategic architectural knowledge, within an organization or within the larger architecting community. Different viewpoint sets and their use to describe architecture framework are reported in [6]. A model has a subject and it can be a concept or a work product. If model is a concept, then the architecture is a conception or mental model of a system. If the model is a work product, then its subject is determined by its model kind. Model kind is defined as conventions for type of modeling. Examples of model kinds include: data flow diagrams, class diagrams, Petri nets, balance sheets, organization charts and state transition models. A correspondence defines a relation between Architectural Elements such as Stakeholders, Concerns, Architecture viewpoints and model kinds. A correspondence rule expresses a constraint to be enforced on correspondence. Examples of correspondence rules are described in [7]. Both correspondence and correspondence rules are used to express and enforce architecture relations such as composition, refinement, consistency, traceability, dependency, constraint and obligation.

The remaining paper is organized as follows. Section 2 describes adaptation of conceptual model of architecture framework in ISO/IEC 42010:201X to spatiotemporal data mining. Section 3 describes stakeholders and their issues. Section 4 elaborates Architectural Viewpoints of spatiotemporal data mining system. Section 5 briefs correspondence rules among the Viewpoints. Finally conclusions are given in Section 6.

## 2. ARCHITECTURE FRAMEWORK FOR SPATIOTEMPORAL DATA MINING

An architecture framework for spatiotemporal data mining is described by applying the conceptual model for architecture framework defined in ISO/IEC FDIS 42010. It involves identification of one or more concerns, the identification of one or more stakeholders having those concerns, and one or more architecture viewpoints that frame those concerns and any correspondence rules. The result of performing these activities is the architecture framework for spatiotemporal data mining as shown in Fig. 2 and is described in remaining sections of this paper.

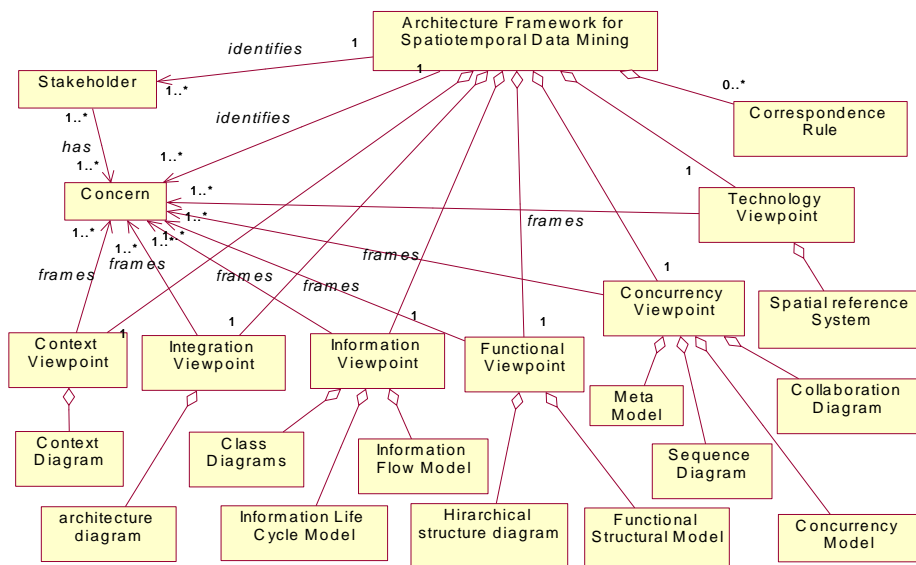


Fig. 2: Architecture Framework for Spatiotemporal Data Mining System

### **3. STAKEHOLDERS AND CONCERNS**

#### **3.1 Stakeholders**

A stakeholder in the architecture of a system is an individual, team, or organization having an interest in the realization of the system [7]. Various stakeholders of architecture framework for spatiotemporal data mining are software engineers, designers, developers, testers, spatiotemporal database designers, knowledge engineers, and online analytical processing tools, domain specialists or users such as meteorologist, archaeologist, animal movement analyst, forester, ecologist, disaster information analyst, geologist, Land use analyst, agronomist, and Road traffic analyst.

#### **3.2 Concerns**

A concern about architecture is a requirement or an issue, an objective, an intention, or an aspiration a stakeholder has for that architecture [3]. Various concerns of the stakeholders are identified after extensive literature survey [8,9] and categorized as follows.

##### **3.2.1 Information Structure and Content related concerns**

Design and development of robust spatiotemporal representation and data structures is the fundamental issue for spatiotemporal data handling, analysis and mining. It has to address following requirements [10, 11].

- Representation of objects with position in space and existence in time.
- Capturing the change of position in space over time: If the change of object's position is continuous, then it results in motion.
- Definition of spatial attributes in time and organizing them into temporal layers or fields, i.e., snapshots of thematic maps.
- Capturing 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").
- Connection of spatial attributes to spatial objects.
- Representation of relationships among spatial attributes in time.
- Specifying spatiotemporal integrity constraints: The constraints are imposed either by the user or by the designer for the integrity of the database.
- Representation of events and list of changes associated with each event: An event may represent abrupt change or it may have duration.
- Representation of multiple granularities for spatiotemporal objects.
- Representation of spatiotemporal data in multi-dimensional model for analytical processing.
- Representation of concept hierarchies for the dimensions.

##### **3.2.2 Spatiotemporal data semantics related concerns**

- The unique characteristics of spatiotemporal datasets are that they carry distance and topological information which requires geometric and temporal computation. Therefore it is required to compute different spatiotemporal topological relationships among spatial objects in time for analysis and mining.
- Spatial and temporal relationships are implicitly defined. They are not explicitly encoded in a database. These relationships must be extracted from data. There is a trade-off

between preprocessing them before the actual mining process starts and computing them on-the fly as and when they are actually needed.

- The attributes of neighboring patterns may have significant influence on a pattern and should be considered. For example, spatiotemporal event like hurricane will have influence on traffic jam pattern.
- Many rules of qualitative reasoning (e.g. transitive property) on spatial and temporal data provide a valuable source of domain independent knowledge that should be taken into account when generating patterns. How to express rules and how to integrate them with spatiotemporal reasoning mechanism is an issue.

### **3.2.3 Spatiotemporal Data management concerns**

- DBMS has to manage spatiotemporal data since it is voluminous. What capabilities are required for the DBMS is a concern.
- How spatiotemporal data mining system is to be integrated to the DBMS so that primitive functions provided by the DBMS are used in analysis and mining.

### **3.2.4 Mining Methodology related concerns**

- System has to mine different kinds of knowledge.
- System needs to have a provision to incorporate background knowledge.
- System should eliminate uninteresting patterns and trends through knowledge evaluation.

### **3.2.5 User Interaction related concerns**

- Interactive mining of knowledge at different levels of abstraction. This requires development of efficient techniques for visualization of spatiotemporal knowledge and interaction facilities for gaining an insight of underlying phenomena represented by the knowledge.
- Presentation and Visualization of spatiotemporal data mining results. This involves development of effective visual interfaces for viewing and manipulating the geometrical and temporal attributes of spatiotemporal data and knowledge.

### **3.2.6 Performance and Technology related concerns**

- The unique characteristic of spatiotemporal datasets requires significant modification of data mining techniques so that they can exploit the rich spatial and temporal relationships and patterns embedded in the datasets. The data mining algorithms must be efficient, scalable and able to use parallelism for maximizing the performance of the system.
- The hardware and software used to develop and implement spatiotemporal data mining system has to support parallelism and the Database Management System must support spatial reference system.

## **4. ARCHITECTURAL VIEWPOINTS**

A viewpoint is a way of looking at system, and application of a viewpoint to a particular system is referred to as an architecture view. The reusable nature of architecture viewpoints (and architecture frameworks, as coordinated viewpoint sets) highlights their utility as mechanisms for capturing strategic architectural knowledge. The viewpoints identified to address the concerns are Context viewpoint, Integration viewpoint, Information viewpoint, Functional viewpoint, Concurrency Viewpoint and Technology Viewpoint as described in the following sections.

#### 4.1 The Context Viewpoint

The context viewpoint is defined and explained in [12]. It describes the relationships, dependencies, and interactions between spatiotemporal data mining system and its environment as shown in Fig. 3.

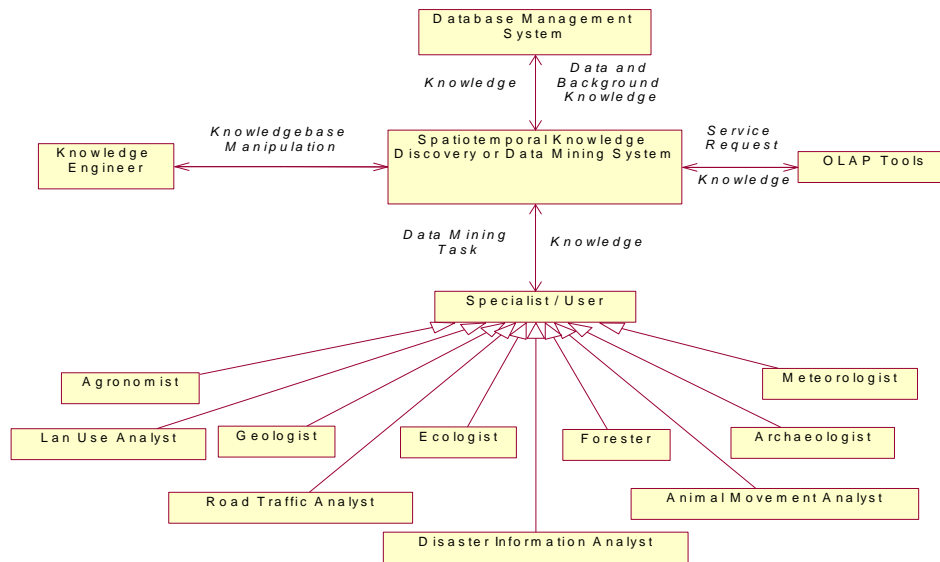


Fig. 3: Context Diagram of the Architecture Framework

The database management system manages the spatiotemporal data and background knowledge. Intermediate results of data preprocessing algorithms and results of the data mining methods can also be stored in the database for further analysis and mining. The knowledge engineer interacts with domain experts and provides domain specific knowledge or background knowledge to the system. This knowledge is used for guiding the search process and to evaluate the discovered knowledge for its usefulness. OLAP tools send request to the system to discover knowledge at different levels of abstraction of the task relevant data. Specialists or users belonging to various domains specify their data mining tasks. The system should provide an appropriate interface so that they specify task relevant data, kind of knowledge to be mined, background knowledge, interestingness measures and form in which the discovered knowledge is visualized. The spatiotemporal data mining system should use information specified as part of the data mining task and discover appropriate knowledge. It should display the discovered knowledge to the user as per his visualization requirement.

#### 4.2 Integration Viewpoint

The integration viewpoint describes how to couple architecture framework for spatiotemporal data mining with spatial database or data warehouse server. The degree of coupling decides to what extent the data and functionalities provided by such servers can be utilized in spatiotemporal data mining. There are four possible schemes of integration, - no coupling, loose coupling, semi-tight coupling and tight coupling [13]. Semi-tight coupling is considered and used for integration of architecture framework for spatiotemporal data mining with spatial database server because it facilitates access of task relevant data from the database, use of built-in functions of the database server for spatial data processing and storing of intermediate and end results in the database.

### 4.3 Information viewpoint

The information viewpoint describes the way the system stores, manipulates, manages, and distributes information. This viewpoint develops a complete but high level view of static data structure and information flow. It focuses on the kinds of information handled by the system, the semantics of the information, and the interpretation of that information. It describes the information managed by the system along with the structure, content, semantics, type, relationships, and constraints on the data used within the system. Design of an object relational spatiotemporal database and its usage for capturing the evolution of spatiotemporal objects are described in this section.

#### 4.3.1 Spatiotemporal Database

The requirements are analyzed and different classes, attributes and methods of each class as well as relationships among the classes are identified [10, 11]. 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 exists 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 attribute values are not allowed to be changed. The Spatiotemporal database entities that are designed and relationships among those entities are described below.

**Spatial\_objects\_Table:** This entity maintains spatial objects and it has following attributes.

Attribute Name	Attribute Type	Description
Obj_id	Numeric	Spatial object identifier
Obj_geom	Geometry	Represents geometry like point, line, polygon of the spatial object or event or episode.
f_t	timestamp	Creation time of Spatial object ( from time )
t_t	timestamp	Cease time of Spatial object. ( to time )
Obj_Category	Categorical	Used to identify group of objects of same type in given area. Example are road network, water supply system, parks, forests.
Obj_Type	Categorical	Indicates type of the spatial object. Examples are flood, land parcel, well, village, city, district, state, country, forest.
Change_type	Char	It indicates type of change the spatial object has undergone before or at cease time. S for Split, M for Merge, C for geometry change that needs to be recorded with new object identifier, L for Location change, G for geometry change that is to be recorded with old object identifier only and M for both geometry and location change.

**Split\_table:** This entity maintains information about spatial object which got split into two or more spatial objects. If any object has got split into n objects, then this table contains n objects with same obj\_id but each one of them having unique new\_obj\_id representing one of the objects derived due to split. Each new object created due to split is maintained in spatial\_objects\_table to record its geometry and existence.

Attribute Name	Attribute Type	Description
Obj_id	numeric	Object identifier of the spatial object that got split. References Obj_id of Spatial_objects_table.
New_obj_id	numeric	Identifier of the object derived due to split.
S_t	timestamp	Time at which split occurs.

**Merge\_table:** This entity maintains information about the creation of new spatial object derived due to merging of two or more spatial objects.

Attribute Name	Attribute Type	Description
Obj_id	numeric	Object identifier of the spatial object that has participated in the merge. References Obj_id of Spatial_objects_table.
New_obj_id	numeric	Identifier of the object created due to merge.
m_t	timestamp	Time at which merge occurs.

If n objects are merged into a single object, then this table contains n objects with same New\_object\_id but each one with unique obj\_id representing one of the n objects participated in merge. The new object created due to merge is maintained in spatial\_objects\_table to record its geometry and existence

**Geom\_Version\_table:** This entity maintains object identifiers of spatial objects whose geometry is changed. It is designed based on principle that object identifier changes whenever its geometry is changed. The changed geometry is maintained in spatial\_objects\_table.

Attribute Name	Attribute Type	Description
Obj_id	numeric	Identifier of the object whose geometry is changed. References Obj_id of Spatial_objects_table.
New_obj_id	numeric	New object identifier assigned after the change.
C_t	timestamp	Time at which the geometry got changed.

**Attribute\_Info\_Table:** This entity maintains meta data about attributes of spatial objects.

Attribute Name	Attribute Type	Description
Obj_id	numeric	Identifier of the spatial object. References Obj_id of Spatial_objects_table.
Attr_id	numeric	Identifier of the attribute of the spatial object identified by Obj_id.
Attribute_Name	Categorical	This attribute holds name of the attribute of the spatial object.
Attribute_type	numeric	This attribute holds a value that indicates type of the attribute which is held in Attribute_Name. 0 for integer, 1 for real, 2 for char, 3 for string and 4 for Boolean.
Attribute_Category	numeric	The value held by this attribute indicates category of the attribute which is held in the field Attribute_Name. 0 for fixed, 1 for version insignificant and 2 for version significant.



**Version\_Insignificant\_Attribute\_Table:** This entity maintains all the information about version insignificant and fixed value attributes of all the objects in spatial\_objects\_table.

Attribute Name	Attribute Type	Description
Obj_id	numeric	Identifier of the spatial object. References Obj_id of Spatial_objects_table.
Attribute_id	numeric	Identifier of an attribute. References Attri_id of the Attribute_Info_Table.
Attribute_Value	Categorical	This attribute holds the value of an attribute identified by the field Attribute_id.

**Version\_Significant\_Attribute\_Table:** This entity maintains all the information about version significant attributes of the all spatial objects in spatial\_objects\_table.

Attribute Name	Attribute Type	Description
Obj_id	numeric	Identifier of the spatial object. References Obj_id of Spatial_objects_table.
Attribute_id	numeric	Identifier of an attribute. References Attri_id of the Attribute_Info_Table.
Attribute_Value	Categorical	This attribute holds the value of an attribute identified by the field Attribute_id.
v_f_t	timestamp	Valid from time.
v_t_t	timestamp	Valid to time.

**Temporal\_table :** This table stores timestamps which correspond to time at which change to any spatial object in spatial\_objects\_table has taken place. It is a single attribute table.

Attribute Name	Attribute Type	Description
Change_occur_time	timestamp	Time of change of some spatial object.

Application of data mining techniques on spatiotemporal data stored in above tables yields spatiotemporal patterns and trends. Suitable table structures need to be designed if results are to be stored in database for further processing or any other reason. The intermediate results can also be stored in the database by designing appropriate tables so that large volumes of data can be handled.

#### 4.3.2 Capturing Spatial Object Evolution

How spatiotemporal database could capture various changes of spatiotemporal objects during its life time is described in this section.

The values of geometry, location and non-spatial version significant attributes of the spatial objects may change with time. Whenever the value of the version significant attribute of any spatial object is changed, a new record is created with the values of the object identifier, attribute identifier, new value and time of change. This new record is stored in Version\_Significant\_Attribute\_Table of the spatiotemporal database. How changes of geometry and location of a spatial object are captured in spatiotemporal database is described in Fig. 4.

The geometry of the spatial object when first created is stored in Spatial\_Object\_Table with its creation time. Whenever the object ceases to exist, then its cease time is recorded in the same table. Changes that occur to spatial object during its life time can be categorized as

- Spatial object splits into two or more objects.
- Two or more spatial objects merge to form a new spatial object.
- The Geometry of the spatial object changes.
- The location of the spatial object changes.
- Both geometry and location of the spatial object change.

Whenever any spatial object gets split into two or more objects, the details of split are recorded in Split\_Table, the object that got split ceases to exist and the new objects created due to split are recorded in Spatial\_Objects\_Table. Whenever two or more spatial objects got merged to form new spatial object, the details of the merge are recorded in Merge\_Table, The object which participated in merge ceases to exist and the new object created is stored in Spatial\_Objects\_Table.

The geometry change occurring to spatial object is handled in two ways as follows.

1. The time of change to the geometry of the spatial object is recorded in Geom\_Version\_Table and new geometry of the object is stored as a new spatial object in Spatial\_Object\_Table. The link between old object identifier and the new object identifier is maintained in Geom\_Version\_Table. The old object is now ceased to exist.
2. A new object with old object identifier is created to represent the geometry change and this new object is stored in Spatial\_Object\_Table. The creation time of old and new object is different even though their object identifier is same.

The location change occurring to the spatial object is captured by forming a new version for the object with same identifier but with different time of creation and storing it in Spatial\_Object\_Table. The creation time of old and new versions of the object is different even though their object identifier is same. If both location and the geometry of the object are changed, then this is also captured by forming a new version for the object with same identifier but with different time of creation and storing it in Spatial\_Object\_Table.

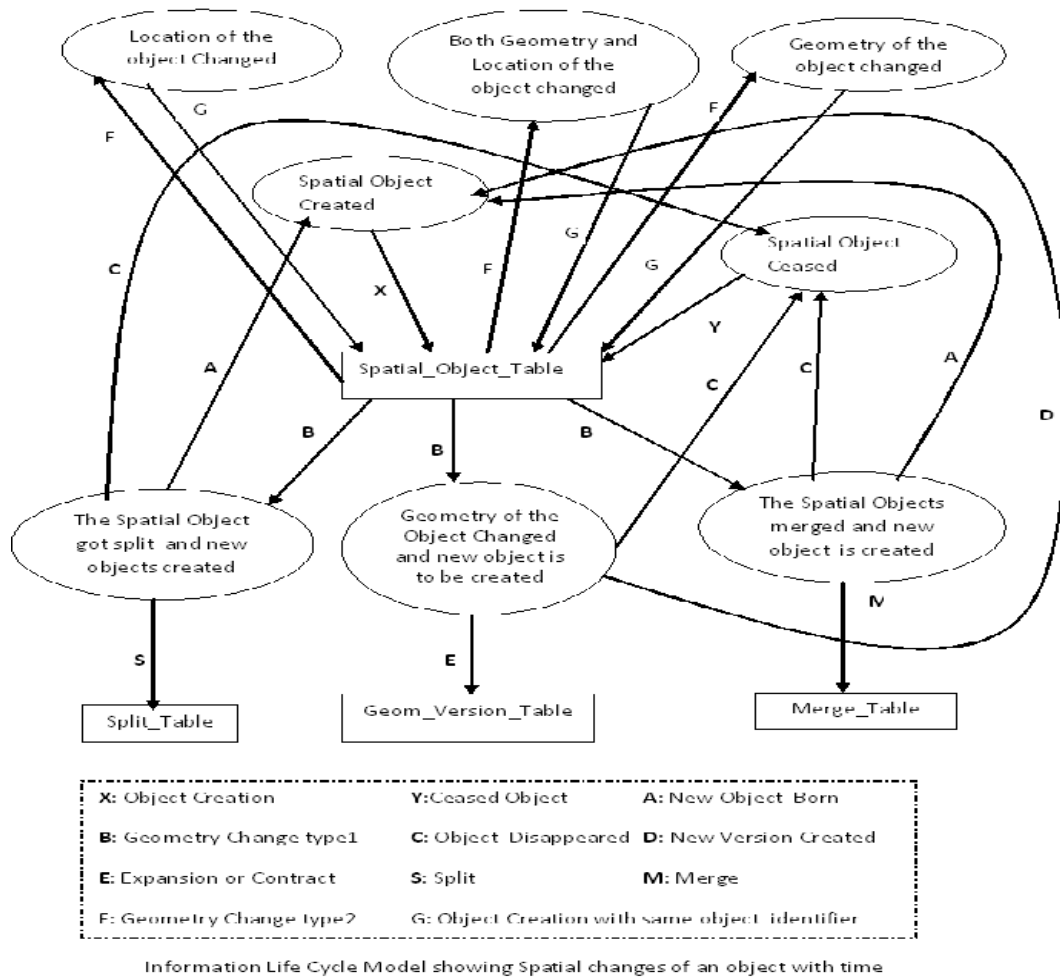


Fig. 4: Information Life Cycle Model for Spatiotemporal Object

#### 4.4 Functional viewpoint

The functional viewpoint describes the system's runtime functional elements, their responsibilities, interfaces, and primary interactions. The functional structure model and responsibilities of various components are described in this section.

##### 4.4.1 Functional Structure Model

It identifies the architectural elements of the system and relationships among them as shown in Fig 5.

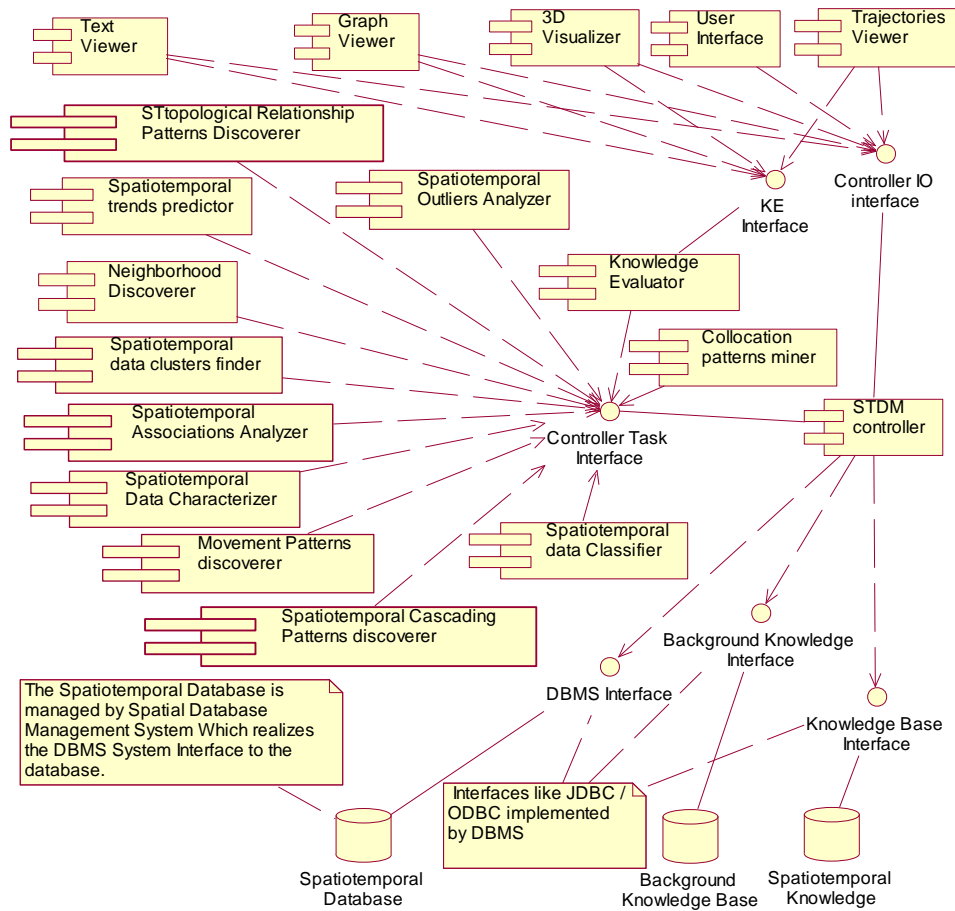


Fig. 5: Functional Structure Model of the Architecture Framework

Following entities are not shown in Fig. 5 to reduce the complexity of the framework and to increase ease of interpretation.

1. The interfaces provided by knowledge discovery components such as ST topological Relationship Discoverer, Spatiotemporal trends predictor, Spatiotemporal data clusters finder, Spatiotemporal Associations Analyzer, Spatiotemporal Data Characterizer, Movement Patterns Discoverer, Spatiotemporal Outlier Analyzer, Collocation Patterns miner, Spatiotemporal Cascading Patterns discoverer and Spatiotemporal Classifier are not shown in the diagram. Each of these components realizes an interface and uses the interfaces provided to access spatiotemporal Database, Background knowledge base and spatiotemporal knowledge.
2. The Knowledge Evaluation component uses the interfaces provided by knowledge discovery components, Background knowledgebase and spatiotemporal knowledgebase. It provides an interface so that knowledge presentation components such as Text Viewer, Graph Viewer, Trajectory Viewer and 3D Visualizer gets the knowledge to be displayed.

But all the interfaces of ST\_topological relationship Patterns discovery [14, 15] component and their usage in discovering Spatiotemporal Topological Relationships patterns are shown in Fig. 6 for visualization of complete functional structure model of the framework.

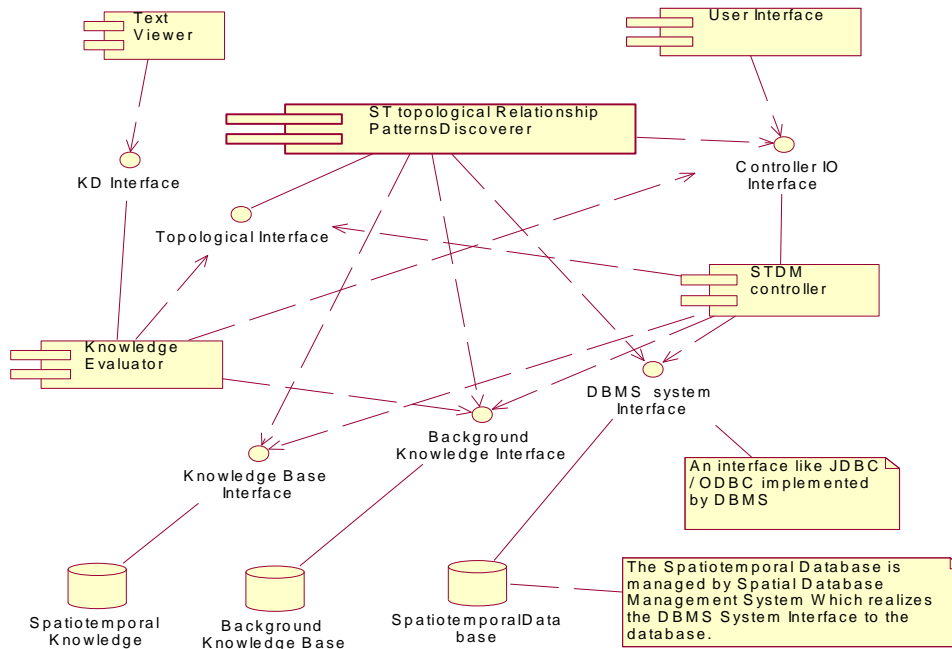


Fig. 6: Functional Structure Model for Spatiotemporal Topological Relationships patterns

#### 4.4.2 Components Responsibilities

Responsibilities and interfaces of different components in functional structure model are described below.

**User Interface:** This component accepts a data mining task from user. It should provide facilities to specify task relevant data, background knowledge, kind of knowledge to be discovered, interestingness measures and methods for visualization of discovered knowledge. It communicates the data mining task to **STDM controller** through **Controller IO interface**.

**Text Viewer:** This component should display the discovered knowledge in the form of text.

**Graph Viewer:** This component is responsible to present the discovered knowledge in the form of graphs, e.g. how traffic patterns are changing in a road network of a particular area over a period of time.

**3D Visualizer:** This component is responsible for showing patterns and trends in the form of data cube.

**Trajectories viewer:** This component has to display trajectories of moving objects and moving patterns.

**STDM controller:** This component is responsible for analyzing data mining task and delegate the task to the appropriate component through the interface provided by that component. It uses

interface provided by the Spatiotemporal Database, Background Knowledge base and Spatiotemporal Knowledge to carry out its responsibilities.

**Spatiotemporal Database:** This component is the implementation of spatiotemporal database schema and holds the spatiotemporal data. It is managed by spatial database management system which is external to the spatiotemporal data mining system. It is accessed by different components through an interface realized by the external Spatial DBMS.

**Background Knowledge base:** This component stores concept hierarchies and user interesting measures for different kinds of knowledge. It is managed by Knowledge base management system or DBMS which implements an interface to access the knowledge base. Different components use this information to guide the search for patterns and trends. The knowledge evaluation component uses interesting measures to evaluate interestingness of the patterns or trends.

**Spatiotemporal Knowledge:** This component holds intermediate results or end results of different knowledge discovery components. It is to be managed by external system like DBMS or KDBMS.

**Knowledge Evaluation Component:** This component is responsible to accept the knowledge discovered by any of the knowledge discovery components and to evaluate its usefulness or interestingness to the user based on the interestingness measures provided by the background knowledgebase. It sends the interesting patterns to the visualization components through its interface.

**Neighborhood Discoverer:** This component is responsible for finding spatiotemporal neighborhoods [9] from spatiotemporal database. This is used by many other components such as outlier analyzer, clusters finder, cascading patterns discoverer, collocation patterns miner.

**ST Topological Relationship Patterns [14, 15] Discoverer:** This component when activated by the STD controller component computes spatiotemporal topological relationships among the specified types of spatial objects. It stores the spatiotemporal topological relationships [14] in database. It then applies the topological relationship pattern [15] discovery techniques to find patterns.

**Spatiotemporal Trend [9] predictor:** This component uses regression, auto regression or neural network based techniques to predict spatiotemporal trends. This gets information to discover such trends from STD controller.

**Spatiotemporal Data Clusters [9] finder:** This component uses neighborhood discoverer to generate a neighborhood set and then stores it in the database. It groups the spatiotemporal objects based on similarity using appropriate methods to find spatiotemporal clusters [9] such as moving clusters, trajectory clusters.

**Spatiotemporal Association [9] Analyzer:** This component discovers associations among moving spatiotemporal objects or among spatiotemporal objects whose topological relationships with other objects change with time. It also finds associations among changes of thematic attribute values of spatiotemporal objects.

**Spatiotemporal Data Characterizer:** This component computes characteristics [9] of spatiotemporal objects, for example spatial objects evolution, characterization of spatial objects whose attribute values change with time.

**Movement Patterns [9] discoverer:** This component performs required computations for discovering grouping dynamics of moving objects along their trajectories. This uses neighborhood discoverer component services for computing spatiotemporal neighborhood set.

**Spatiotemporal Cascading Patterns [9] discoverer:** This component is responsible for discovering partially ordered subsets of event types whose instances are located together and occur serially. It uses neighborhood discoverer to find spatiotemporal neighborhoods.

**Spatiotemporal data Classifier:** This component is responsible for building and validating the spatiotemporal data classification [9] Model. This uses supervised learning techniques such as decision trees, Bayesian belief networks, and neural networks with back propagation learning method.

**Collocation patterns miner:** This component computes the spatiotemporal collocation patterns and collocation episodes [9]. It uses the services of the neighborhood discoverer component to discover spatiotemporal neighborhood of the different types of objects involved in collocation patterns and episodes.

**Spatiotemporal Outlier Analyzer:** This component is responsible for computing and discovering spatiotemporal outliers [9] which deviate too much from other spatiotemporal objects. The services of neighborhood discoverer are required for this component.

#### 4.5 Concurrency Viewpoint

The concurrency viewpoint describes the concurrency structure of the system. It maps the functional elements onto runtime execution entities to clearly identify the parts of the system that can execute concurrently and shows how this is coordinated and controlled. The concurrency model contains processes, process groups, threads and inter-process communication details. Fig. 7 shows the meta-model for the concurrency viewpoint.

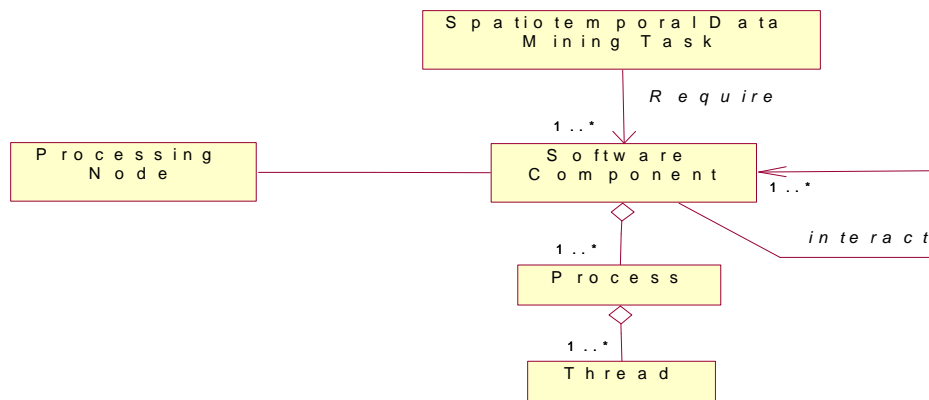


Fig. 7: Meta Model for the Concurrency Viewpoint

Different processes and the functional elements assigned to them to identify design level concurrency are described in the following table. All the processes described in the table can run simultaneously.

<b>Process Name</b>	<b>Process Description</b>	<b>Components associated to the process</b>
STDM_task	This process accepts spatiotemporal data mining task from user.	User Interface component.
Display	This process presents the discovered knowledge to the users. This creates threads or sub processes depending on visualization requirements of the components associated to it.	Text Viewer, Graph Viewer, 3D Visualizer, Trajectory Viewer components.
STDM_controller	This process receives data mining task details from STDM task process. Then it creates a sub process called Validate task and delegates the responsibility of validating the user specified task with respect to the information in the database.	STDM controller component.
Validate_Task	This is a child process of STDM controller process. It validates data mining task and communicates to appropriate data mining process if the task is valid. This process sends validated task information and required communication to one of the other processes such as classification, Cascading Patterns, Movement patterns, characterizer, clustering, outliers, prediction, Topological Relations, Association analysis and collocations.	DBMS Interface, Background Knowledge Interface.
Knowledge_Evaluation	This process evaluates the discovered knowledge. It gets communication from data mining process when it has discovered some knowledge that needs to be shown to user. It creates threads and delegates the responsibility of evaluating the user interestingness of the patterns discovered by each data mining process.	Knowledge Evaluator component.
classification	This process spawns threads to construct and validate classification models.	Spatiotemporal data classifier component.
Cascading_patterns	This process executes spatiotemporal cascading patterns discoverer components. It creates thread to compute neighborhoods.	Spatiotemporal cascading Patterns discoverer and Neighborhood discoverer components.
Movement_patterns	This process runs to discover spatiotemporal movement patterns. It	Movement Patterns discoverer and Neighborhood



	creates threads to speed up neighborhood computation.	discoverer components.
characterizer	This process is to do spatiotemporal data characterization.	Spatiotemporal data characterizer.
clustering	This process performs spatiotemporal data clustering. It spawns required number of threads to compute spatiotemporal similarity of every pair of spatiotemporal objects.	Spatiotemporal data clusters finder and Neighborhood discoverer components.
Outliers	This is the process that discovers spatiotemporal outliers.	Spatiotemporal Outliers Analyzer and Neighborhood discoverer components.
predictor	This process builds models for spatiotemporal data prediction.	Spatiotemporal trends predictor component.
Topological_Relations	This process is designed to call and execute Spatial DBMS functions to find spatial topological relationships and organizing them into spatiotemporal dimensions. It also finds spatiotemporal topological relationship patterns.	ST topological Relationship Patterns Discoverer component.
Association_analysis	It is used to find different types of spatiotemporal association rules.	Spatiotemporal Association Analyzer component.
collocations	This process is to discover spatiotemporal collocation patterns and episodes. It creates threads to speed up neighborhood computation.	Collocation patterns miner and Neighborhood discoverer components.

#### 4.6 Technology Viewpoint

The Technology viewpoint describes the specific technology that will be used to implement the system. The spatiotemporal data mining system possesses multiple tasks that can run concurrently. Each task may also have sub tasks that can be executed concurrently as multiple child processes or threads. The Spatiotemporal data mining system also depends on database management system because both the systems are integrated using semi-tight coupling method. The underlying DBMS also may execute queries and operations on the database concurrently for maximizing its performance. Simultaneous execution of multiple tasks requires use of multiple processors inside a computer, operating independently of one another. These processors might be integrated in one chip or communicating on the same bus / high speed communication link. The hardware architectures, operating system features and DBMS features required for efficient implementation of spatiotemporal data mining system are described below.

##### 4.5.1 Hardware Architectures

Advances in computer architecture, database technology, and data storage devices all contribute to the parallel processing that makes it possible to query the massive databases and achieve acceptable response times. It is a proven technology today to combine dozens or hundreds of central processing units (CPUs) together to build a supercomputer to run queries and perform calculations against a multi-terabyte database.

There are four hardware architectures for parallel processing:

1. Symmetric Multiprocessor (SMP) system, also known as Shared Everything system, is a set of tightly coupled CPUs that share memory and disk. The Operating system of SMP contains special code to allow the different CPUs to access central structures such as job schedule queues and memory address tables in a controlled manner. This means that a process can run on different CPUs at different times in a system.
2. Massively Parallel Processor (MPP) system, also known as Shared Nothing system, is a set of loosely coupled nodes, each of which has its own memory and disk. Unique instances of the operating system and application run on each node. No resources are shared between the nodes and communication between the nodes is done through high speed connection used by operating systems for passing messages between the nodes.
3. Clustered SMP system, also known as shared disk system, is a set of loosely coupled nodes connected by a high speed interconnect. Each node is an SMP system, having its own CPU and memory but they share access to disk.
4. Non-Uniform Memory Access (NUMA) system, also known as distributed shared memory system, is a hybrid of the clustered SMP and MPP platforms.

#### 4.5.2 Operating System features

The operating systems designed for these different architectures should provide advantages of hardware platform to applications if the applications are developed using the application programming interface provided by the Operating system. However, if the applications are developed using higher level languages such as Java, which provide communication mechanisms inside the language environment, then degree of parallelism and performance of the application may be limited by the virtual machine (such as Java Virtual Machine) capabilities.

Parallelism in an application can be detected at several levels, - Large grain (task level), Medium grain (control level), Fine grain (data level) and Very fine grain level ( multiple instruction issue). Following table describes code size and kind of parallelism.

Grain Size	Code Size	Parallelized by
Large	Program or separate heavy weight process	Programmer
Medium	Function or module of standard size	Programmer
Fine	Loop / Instruction block	Parallelizing Compiler
Very fine	Instruction	Processor

#### 4.5.3 DBMS features

Parallel relational databases such as NCR Teradata, Informix XPS, Sybase IQ12-Multiplex and IBM DB2 UDB Enterprise-Extended Edition enable parallel query execution via simultaneous and concurrent execution of SQL on separate CPUs, each extracting data from its own disk which holds a portion of the database. A defining feature of a parallel database is that the tables are distributed across the available disks. Allocating the data in a predetermined manner across the nodes balances the workload and maximizes the memory buses for sequential reads of large, uninterrupted blocks of data.

The DBMS must have supported coordinate reference system which manages the metadata required to interpret spatial location information unambiguously and timestamp data type to represent temporal data associated with spatial objects. The programming language like Java must provide constructs by which different components can be implemented. Database Access systems like JDBC, Java Database Connectivity, must have been implemented by the DBMS so that data can be accessed and processed in application. The DBMS also should have implemented data types to handle geometric objects and open GIS functions for handling and processing geometry objects in applications.

#### 4.5.3.1 Spatial Reference System

The high level abstract model for spatial referencing by coordinates is shown in Fig. 8. Coordinate reference system consists of one coordinate system that is related to the earth through one datum. The coordinate system is composed of a set of coordinate axes with specified units of measure. This concept implies the mathematical rules that define how coordinate values are calculated from distances, angles and other geometric elements and vice versa. A datum specifies the relationship of a coordinate system to the earth, thus ensuring that the abstract mathematical concept “coordinate system” can be applied to the practical problem of describing positions of features on or near the earth’s surface by means of coordinates. The resulting combination of coordinate system and datum is a coordinate reference system.

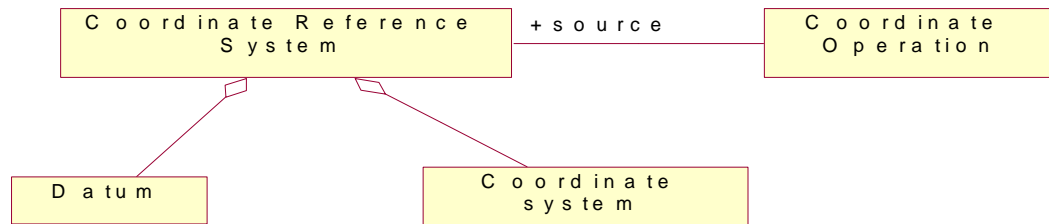


Fig. 8: Spatial Reference System

Commercial database management systems such as NCR Teradata and Oracle are capable of managing huge volumes of database. The Teradata Database is supported on both NCR SMP and NCR MPP systems with UNIX MP-RAS or Windows 2000 Operating System. It also supports geospatial data types and temporal data types in its databases. Oracle RDBMS also takes the advantage of SMP, MPP, Clustered and NUMA technologies and runs on all these platforms. It supports spatial and temporal data types in its databases. These two systems are suitable for development and management of high performance spatiotemporal data mining systems.

## 5. CORRESPONDENCE RULES

A correspondence expresses relationship between architectural elements across viewpoints. A correspondence rule specifies a constraint to be enforced on a correspondence. Following correspondence rules are identified across viewpoints of spatiotemporal data mining architecture framework.

- Every data mining component specified in functional viewpoint has to get task relevant data from the database whose structure is defined in information viewpoint.
- Every component in the component diagram of functional viewpoint should be associated with at least one process in concurrency viewpoint.

- All columns of type “ Geometry “ in spatiotemporal database designed in Information viewpoint must be associated with one spatial reference system described in Technology Viewpoint.
- The spatiotemporal database referred in Integration viewpoint should have the similar structure or schema described in Information Viewpoint.
- The Knowledgebase referred to in context viewpoint must be accessed and used by Knowledge Evaluator component of component diagram in Functional viewpoint.

## 6. CONCLUSION

The rapid growth of spatiotemporal datasets due to widespread use of sensor networks and location aware devices as well as domain specific features associated with such dynamic datasets demand research in spatiotemporal data mining tasks. Spatiotemporal data mining poses many challenges but has promising applications in various domains. It is still a largely unexplored area of research. This paper elaborated the significance of architecture framework for spatiotemporal mining in addressing various issues and concerns related to representation, processing, analysis, mining and visualization. Nature of spatiotemporal data, how complex it is and need for scalable and efficient algorithms are described. Spatiotemporal data mining tasks such as multidimensional analysis, characterization, classification, clustering, association analysis, outlier analysis, collocation patterns, episodes, cascading spatiotemporal patterns, movement patterns, trends and topological relationships are addressed through components and processes.

The role of computer science in terms of applying systems and software engineering to the problem of spatiotemporal knowledge discovery is investigated in this research paper. The conceptual model of architecture framework standardized and described in ISO/IEC 42010:201X is applied and various stakeholders and their concerns as requirements are identified. Modeling spatial, temporal dimensions and incorporating them into spatiotemporal database design is done in information viewpoint. The research focused on high level abstraction of various data mining tasks. Accordingly, appropriate components and their responsibilities are identified for spatiotemporal data mining in functional viewpoint. The system context, the rationale behind its integration to external database management system using semi-tight coupling and design level concurrency are discussed in respective viewpoints. The technology for handling and processing of large spatiotemporal databases is elaborated in technology viewpoint. The correspondence rules across various viewpoints are described.

The future work involves

- Detailed requirements analysis and development of efficient methods and techniques for each of the spatiotemporal data mining tasks.
- Evaluation of the techniques with large datasets in different domains at multiple spatial and temporal granularities.
- Identifying quality measures specific to each of the spatiotemporal data mining tasks.
- Extension of the architecture framework for distributed spatiotemporal data mining.
- Design and development of efficient methods and techniques for distributed spatiotemporal data mining.

Future work requires interdisciplinary collaboration of data miners with researchers in different domains to evaluate data mining methods and the discovered results. It is expected that integration of spatiotemporal data mining to digital earth [16] which can integrate information relating to the atmosphere, hydrosphere, lithosphere, and biosphere of the earth [17] is likely to gain importance in future. Multidimensional modeling, analysis and mining of spatiotemporal

data plays a major role in realization of highlighted applications of digital earth related to global climate change, natural disaster prevention, new energy-source development, agriculture and food security, and urban planning and management [17, 18].

## REFERENCES

- [1] K.Venkateswara Rao, Dr. A.Govardhan & Dr.K.V.Chalapati Rao, (2008) "A Generic Framework for Spatio-Temporal Data Mining System", Journal of Andhra Pradesh Society for Mathematical Sciences, Vol 1, No 2, July.
- [2] Laurinen P, Tuovinen L and Roning J (2005), " Smart archive: a component-based data mining application framework", 5th International Conference on Intelligent Systems Design and Applications, 8-10, September, IEEE explore, Digital Library.
- [3] ISO/IEC FDIS 42010 (2011) "Systems and software engineering — Architecture description".
- [4] Rich Hillard (1999) " Views and Viewpoints in Software Systems Architecture", WICSA 1, San Antonio, TX, 22-24, February.
- [5] Nick Rozanski and Eoin woods " Applying Viewpoints and Views to Software Architecture", <http://www.viewpoints-and-perspectives.info>
- [6] Eoin Woods (2004) " Experiences Using Viewpoints for Information Systems Architecture: An Industrial Experience Report", Springer LNCS volume 3047, pp 182-193.
- [7] David Emery and Rich Hillard, (2009) " Every Architecture Needs a Framework: Expressing Architecture Frameworks Using ISO/IEC 42010" , WICSA/ECSA 2009, Cambridge, UK, 14-17, September.
- [8] Jiawei Han, (2003) "Mining Spatiotemporal Knowledge: Methodologies and Research Issues", A position paper, KDV workshop.
- [9] K.Venkateswara Rao, Dr. A.Govardhan & Dr.K.V.Chalapati Rao, (2012) "Spatiotemporal Data Mining: Issues, Tasks and Applications", International Journal of Computer Science and Engineering Survey ( IJCSES), Vol 3, No 1, Feb
- [10] K.Venkateswara Rao, Dr. A.Govardhan & Dr.K.V.Chalapati Rao, (2011) "An Object-Oriented Modeling and Implementation of Spatio-Temporal Knowledge Discovery System", International Journal of Computer Science & Information Technology(IJCSIT), Vol 3, No 2, April.
- [11] K.Venkateswara Rao, Dr. A.Govardhan & Dr.K.V.Chalapati Rao, (2011) "An Object-Oriented Modeling of Spatiotemporal Knowledge Discovery System", International Conference On Advances in Computing and Communication (ICACC – 11), 8-10, April, NIT, Hamirpur.
- [12] Eoin Woods and Nick Rozanski (2009) " The System Context Architectural Viewpoint", IEEE/IFIP WICSA/ECSA, 14-17, September, Cambridge.
- [13] Jiwei Han and Micheline Kamber (2006) " Data Mining Concepts and Techniques", second edition, Morgan Kaufmann Publishers, pp:34-36.
- [14] K.Venkateswara Rao, Dr. A.Govardhan & Dr.K.V.Chalapati Rao, (2011) "Discovering Spatiotemporal Topological Relationships", The second international workshop on Database Management Systems, DMS-2011, July, Chennai, India, Springer Proceedings LNCS-CCIS 198.
- [15] K.Venkateswara Rao, Dr. A.Govardhan & Dr.K.V.Chalapati Rao (2012) "Mining Topological Relationship Patterns from spatiotemporal Databases", International Journal of Data Mining and Knowledge Management Process, IJDKP, Vol-2, no-2, March.
- [16] Karl E Grossner, Michael F Goodchild & Keith C Clarke, (2008) "Defining a Digital earth System " , Transaction in GIS, 12(1):145-160.
- [17] H D Guo, Z Liu & L W Zhu, (2010) "Digital Earth: decadal experiences and some thoughts", International Journal of Digital Earth, Vol 3, No 1, March, 31-46.
- [18] Huadong Guo, (2010) "Understanding global natural disasters and the role of earth observation", International Journal of Digital earth, Vol 3, No 3, Sept.

## **Authors**

1. 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 Hyderabad. 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.

2. 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 a Professor in CSE and Director of Evaluation of Jawaharlal Nehru Technological University Hyderabad, He has published more than 152 research papers in various national and international Journals/conferences. He guided seven PhD thesis and number of post graduate projects. His research of interest includes Databases, Data Warehousing & Mining, Information Retrieval, Computer Networks, Image Processing, Software Engineering, Search Engines and Object Oriented Technologies.

3. 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.