International Journal of Latest Trends in Engineering and Technology
Vol.(8)Issue(4-1), pp.294-300
DOI: http://dx.doi.org/10.21172/1.841.47
e-ISSN:2278-621X

LITERATURE SURVEY OF TEMPORAL DATA MODELS

Lalit Gandhi[1]

Abstract: Time is a standout amongst the most difficult aspect to deal with, in real time applications. The goal of this paper is to give a brief outline on need of temporal databases and survey the work done in this field. We utilized the most essential temporal dimension to explore their execution. A relative investigation of various temporal models carried out basis upon few parameters. Finally, all the temporal models are abridged and a concluded with the future work suggestions in the field of temporal database.

1. INTRODUCTION

Relational database is the most popular database nowadays. The main reason for popularity is Codd's rule and associated relational algebra. Relational data model considers only two dimensions i.e. rows (tuples) and columns (attributes). Time of transactions has no importance in relational model. Day to day real time applications like airline/ railway reservation system, banking/insurance, post office applications need to save transaction time in the database. Therefore, time of transaction plays important role and classified as third dimension of data models.Databases, that stores time dimension values along with transactional data, called as temporal databases.

2. NEED OF TEMPORAL DATABASE

Over the most recent two decades, the relational data model has picked up prominence because of its effortlessness and strong scientific establishment. Relational model as statedby Codd, does not consider the temporal dimension of data. It represents state of data only at solitary snapshot of time. The variation in the contents of databaseis *change*, when new data is included, erasing the old &obsolete data from the database.

Figure 1, Employee Table of University consists of columns Employee_ID, Counter,Employee_Name, Designation and Branch of *University* Database. (Employee_ID, Counter)acts as the primary key of the table. Counter column stores the number of transactions performed on the Employee table. Figure 2 shows updated table, adding information of transfer of employee with change in Branch as a new tuple, results in increasing the size of table.

Employee_ID	**Counter**	**Employee_Name**	**Designation**	**Branch**
111	0	Amit	Clerk	Registration Branch
112	0	Pankaj	Assistant	Academic Branch

Employee Table (Before Transaction) Figure 1

Employee_ID	**Counter**	**Employee_Name**	**Designation**	**Branch**
111	0	Amit	Clerk	Registration Branch
111	1	Amit	Clerk	Computer Center
112	0	Pankaj	Assistant	Academic Branch
112	1	Pankaj	Assistant	College Branch

Employee Table(After Transaction) Figure 2

The present data caught as a preview, disposes of the time part of past data. This is not appealing forthe applications that need to keep past, present, or probably future data values. This emerges the need to utilize temporal database, *which can store the time-variant data* without disposing of past esteems.

3. TEMPORAL DATABASECONCEPTS

Temporal database underpins three sorts of time dimensions: user-defined, valid time and transaction time. User-defined time portrayal meant to satisfy user needs. Valid time is,when certain conditions in the real world were, are or will be valid. Valid time can be characterized with single-chronon identifiers (event timestamps), with intervals (as intervals time-stamps), or as valid-time elements (which are finite set of intervals). Transaction time automatically captures changes made to the time-

[1] *Research Scholar, Department of Computer Science & Engineering, UIET,MaharshiDayanand University, Rohtak, India*

variant data in a database and record time of occurrence.Transaction time helps to maintain versioning. Another type of time-variant data model, called as bi-temporal data model is union of valid time and transaction time information.
Time stamping is the "value of time associated with a data value", that can be a temporal element, time interval or time point. There are two types of time stamping: Attribute time stamping& Tuple time stamping.

a) *Attribute Time stamping*: In attribute time stamping, value of timestamp is associated with attributes of relation (database table). The following are characteristics of attribute time stamping-
   i) Time Variant Data is stored using Non-First Normal Form.
   ii) It utilizes single relation/tuple to store time changing attributes. It does notdistinguish the time-variant and time invariant data into distinct tuples.
   iii) It keeps away from redundancy and is more expressive.
   iv) Attribute time stamping is more appropriate to execute legitimate time inquiries as it executes them quicker, but minimum reasonable for bi-temporal queries and performance degrades as the nesting level of queries increase.

Employee_ID	**Employee_Name**	**Designation**	**Branch**		
			Value	**Time Start**	**Time End**
111	Amit	Clerk	Registration Branch Computer Center	01/01/2009 01/01/2015	31/12/2014 31/12/9999
112	Pankaj	Assistant	Academic Branch College Branch	01/01/2009 01/01/2015	31/12/2014 31/12/9999

Attribute Time stamping Figure3

b) *Tuple Time stamping*: In tuple time stamping,Time Start, Time End is associated with the tuples of the database table. The different qualities of tuple time stamping are:
   i) It depends on the First Normal Form (1NF) to store time-variant information.
   ii) It contains high data redundancy in view of the reiterations of different tuples of a table because of progress in characteristic esteems over the time.
   iii) It utilizes relational tables to represent time-invariant along with time-variant data. It parts both time-variant and time-invariant data into few tuples.
   iv) It performs better for more complex nesting structure and bi-temporal query, consequently less reasonable to execute substantial time queries as it executes them gradually.

Employee_ID	**Counter**	**Employee_Name**	**Designation**	**Branch**	**Time Start**	**Time End**
111	0	Amit	Clerk	Registration Branch	01/01/2009	31/12/2014
111	1	Amit	Clerk	Computer Center	01/01/2015	31/12/9999
112	0	Pankaj	Assistant	Academic Branch	01/01/2009	31/12/2014
112	1	Pankaj	Assistant	College Branch	01/01/2015	31/12/9999

Tuple Time stamping Figure 4

4. REVIEW OF TEMPORAL DATA MODELS

The temporal models supervise two sorts of information: time-variant and in addition time in variant. Time can be included in any kind of model like entity-relationship, semantic data models, knowledge based data models and deductive databases. Relational and object oriented models extended to have time dimensions over past two decades. In this section,we have reviewed these models,basis on valid time or transaction time and then bi-temporal data models.

A. Valid Time Temporal Data Models

There are three approaches to represent the valid time in any temporal data model - single chronon (event timestamp), intervals (interval timestamp), or as valid-time elements (finite set of intervals). Valid time is related in three ways -with individual attributes value, attribute groups, or with the whole tuple.

Brooks[1]took first place to reflect time in the database. Frederick Brooks Jr. proposed three-dimensional view of valid time database, extending two-dimensions (tuples & attributes) of relational database. Subsequently Ahn, Clifford-1 and McKenzie stressed on "*cubic*" resemblance.

Wiederhold[3]developed the temporal data model to support medical applications specifically. This data model is *Time Oriented Data Base* where relation is sets of time-value-entity-attribute quadruple. Time stamping indirectly represents visit number on each visit. Segev's[14] model explored it more as *time sequences.*

Jones[5]proposed LEGOL 2.0. This is a language designed specifically for law-making rules writing and high-level system spec. First *time-oriented algebra*wasdefined in this language. Objects in LEGOL 2.0 are relations with addition of two implicit time attributes start and stop. Single-chronon represents time attribute values in this model.

Clifford-1[6] [8]is a*Historical Data Model*, with an additional chronon. Attribute is valuedas STATE. A corresponding Booleanfield EXISTS is added to designate whether a specific row exists corresponding to STATE.

Ariav[11]proposed *Temporally Oriented Data Model*. The specialist implemented relational data model as the base model.Sequence of snapshot relation statesindexed by valid time is called as *data cube*. It endorse the storage and retrieval of data using a query language, TOSQL.

Navathe[15] [33]developed *Temporal Relational Model* as an extension of relational model, along with algebra to support TSQL – a temporal extension of SQL. This model used valid time dimension to store validity of data allowing both time-varying and non-time-varying attributes. Time-start and Time-End attributes logs the valid time.

Sadeghi[17] designed a temporal data model featuring HQL, which is a calculus based valid-time query language.Objectsin this model are depictedas valid time relations. Two inherent fields Start and Stop,which record the end -points of each tuple's validity interval. This model required coalescing.

Sarda[24]designed a temporal data model incorporating HSQL, which is a calculus based valid-time query language. Objects in this model are characterized as snapshots or valid time relations. Attribute named *Period*, is used to record the valid time. Period is implicit, non-atomic field. "Period" is not closed at its right boundary.

Segev[14]*Time Sequence* is primary structure of the temporal data model, which recognizes objects by sequence of time-value pairs. Variety of time sequences are stepwise constant behavior, discrete behavior, and continuous behavior assumed for different applications. A set of time sequences is time sequence collection (TSC).

Clifford-2 [13]*Historical Relational Data Model*, associates the timestamps with both distinct tuples and with discrete attributes. This model allows two types of objects: lifespan - set of chronons, and a valid time relation. A lifespan assigned to everyattribute and tuple. Relations in this model consist of set of attributes, set of key attributes, function that maps attributes to lifespan, function which maps attributes to domain values. A tuple is an order pair consist of tuple's value and its lifespan. Relations have key fields and no two tuples can have same value for key attribute at same chronon.

Tansel[12]Designed temporal model, which incorporates *HQuel*, calculus based query language and *time-by-example* language. It supports only valid time relations. Four types of attributes – atomic, set valued, non-time-varying as well as time varying. Fields of the relation can be of multiple categories & values in a given tuple can be heterogeneous.

Gadia-1 [19]*Homogeneous model* supports two types of objects: valid-time elements and relations. Valid time elements are closed under complement, difference and union operations. The model insists homogeneity, all attribute values for a specific row be functions on same valid-time element.

Gadia-2 [22] [30]designed such that temporal elements can be of more than one dimensional, to model different facets of time. Attribute values are functions from temporal elements onto attribute value domain, but not on the same temporal element. Hence, lack in temporal homogeneity. Relations are association of key fields with condition that these fields are single valued.

Lorentzos[20-21]First Temporal Relational Model to integrate *nested specs* of timestamps using values of multiple granularity and to support periodic events. It uses attribute timestamping. No implicit timestampsassumed and used numeric explicit valuesfor updated attributes. Timestamps can be chrononor boundary point of validity interval of one or more attributes.

B. Transaction Time Temporal Data Models:

Principally, versioning distinguishes Object oriented data models. Transaction time data models supports versioning. These data models permit self-assertive, user-defined identifiers related with versions. If required, a whole version hierarchyassigned to data model.

Kimball[4]associates facts with time implicitly. For any update operation, timestampis not explicitly mentioned. It portrays only the snapshots pulled out from transaction time relations.There is no notion for explicit timestamp, as association of time and facts is implicit. Similarly, transaction-time event, interval or element stamping usagecannot be observed.Event-stamped tuples and pointers to predecessor tuples can do an implementation of DATA model.

Stonebraker[16]*Postgre* data model upkeeps transaction-time, built on object oriented data model. In this model,the display is not limited to snapshot states. Relations containing all tuples are a sequence of states. Intervals represent time and uses *Postquel* language based upon *Quel* for data operations.

Jensen [26] [27]*DM/T model* contains a system generated and maintained relation (transaction time) called as *backlog*. This backlog contains time markedchange history of the associated user defined relations. Backlog tuples are impressed with a single time value along withmodified field values. A time slice of a backlog collected for the portion that existed at that time interval.Thus, timestamps accessed like other explicit attributes.

C. Bi-temporal Data Models

Bi-temporal data models backs both valid time and transaction time, implemented using relational databases. Bi-temporal structures are basic components of bi-temporal models that result in consistent temporality.

Ben-Zvi[7]*Time Relational Model*, first bi-temporal model, with two types of relations- snapshot & bi-temporal. Bi-temporal relations have five inherent fields. Effective-time-start and Effective-time-stop are the endpoints of validity interval. Registration-time-start and Registration-time-stop are transaction time values. The deletion-time field contains the time when logically delete tuples. This model uses 1NF to store atomic attribute values, intervals are pair of chronons.
Ahn[9] proposed four-dimensional Model, differentiates valid & transaction time. Relational instances are itself two-dimensional sequences stamped with individual transaction time (as third dimension).In addition to this valid time is termed as fourth dimension &tuples were timemarked by intervals.

Snodgrass [9] [27-29]Bi-temporal Conceptual Data Model (BCDM) features TQuel, four inherent attributes added to the relations supporting valid and transaction time. Transaction time is when a tuple is inserted or deleted. Valid time tuples stamped in reality & stopped being valid in reality.
McKenzie[10] [18] Time stampsattribute value, but restricts that attribute to be single valued. Objects in this model featured as snapshots & valid time relations. Attribute value in the valid time relation is pair of attribute valuesof domain of attribute values along with a set of crayons.

Gadia-3 [19] [22] [30]is a temporal extension of the SQL Model proposed by Navathe. It is associated with *TempSQL – calculus based query language*. Attribute values are time marked with unions of rectangles in valid& transaction time. It can allow more than one occurrence of value equivalent events in the same partition of time.

Vincent S. Lai, Jean Pierre Kuilboer and Jan L. Guynes[32]proposed extension of the EER Model by addition of the time dimension, named Temporal Enhanced Entity Relationship (TEER). This model uses both valid time and transaction time dimensions. This model stores full history of every entity and its relationship.

Debabratadey, Terence M. Barron and Veda C. Storey[34]proposed Temporal Event Entity Relationship Model (TEERM). It is bitemporal in nature. New relationships like static, dynamic, quasi-static relationships and capturing the real world aspects, introduced in this model. Representation of events lead to redundancy into tables.

A temporal data model should meet the specific objectives, which can prompt the best conceivable results. The modelled application should have its semantics captured clearly and precisely. It must represent all the time-variant and time-invariant attributes. In any case, this is most likely impractical to have every one of these components while outlining a temporal data model. There exists an expansion of SQL i.e. TSQL2 which utilizes Bi-temporal Conceptual Data Model that uses the basic semantics of time- varying relations. A diverse model i.e. the representational data model used for guaranteed high execution.

Another data model, i.e.presentational data model portrays the time-varying conduct for the user. In this manner, no single data model can reach every one of the objectives that is the reason a combination of data models used to satisfy all.

D. Comparison

Temporal data models are compared basis upon Representation of Valid time, Representation of Transaction time, Attribute, Value representation and Homogeneity or coalescing.

*Valid time***:** Valid time representation is with single chronon, intervals or valid time elements. Table 1 compares models basis on valid timestamped attribute values & valid timestamped tuples. [35]

	Event	**Interval**	**Valid Time Elements**
TimeStamped Attribute Values		Gadia-2, McKenzie,Thompson, Tansel and Lorentzos	Brooks, Clifford-2, Gadia-1, Gadia-3
TimeStamped Tuples	Ariav, Clifford-2, Segev	Ahn, Ben-Zvi, Jones, Navathe, Sadeghi, Sarda, Snodgrass, Wiederhold	

Table 1

*Transaction Time***:** Transaction time can be represented with Single Chronon, an interval, threechronons and transaction-time element. Table 2 compares temporal models, basis on transaction timestamped attribute values, tuples and set of tuples. [35]

	Single Chronon	**Interval (Pair of Chronons)**	**Three Chronons**	**Element (Set of Chronons)**
TimeStamped Attribute Values				Gadia-3
TimeStamped Tuples	Jensen and Kimball	Snodgrass and Stonebraker	Ben-Zvi	
TimeStamped set of Tuples	Ahn and Thompson	McKenzie		

Table 2

*Homogeneity and Coalescing***:** Table 3 compares the models in aspects of Homogeneity and Coalescing, whether a model is homogenous in valid time. All models are homogeneous in transaction time. [35]

Valid-Time Homogeneous	**Valid-Time Coalesced**
Ahn, Ariav, Ben-Zvi, Clifford-1, Gadia-1, Jones, Navathe, Sadeghi, Sarda, Segev, Snodgrass, Thompson, Wiederhold	Ahn, Gadia-2,McKenzie,Navathe, Sadeghi, Snodgrass

Table 3

*Attribute Values***:** Table 4 compares the models with respect to attribute values, whether Atomic in 1-Normal Form, Functional or Ordered pair.

Attribute Values		
ATOMIC (1NF)	**FUNCTIONAL**	**ORDERED_PAIRS**
Ahn, Ariav, Ben-Zvi, Brooks, Clifford-1, Jensen, Jones, Kimball,Navathe,Sadeghi,Sarda,Segev, Snodgrass,Wiederhold, Stonebraker, Snodgrass, Stonebraker, Thompson,Wiederhold	Clifford-2, Gadia-1, Gadia-2,	McKenzie,Tansel,Wiederhold

Table 4

5. CONCLUSION AND FUTURE AREA OF WORK

The exploration in the territory of temporal database has been going on from last two decades. Codd proposed the relational model with relational algebra, because of which the fundamental concentration was just on the relational model. When it was required to include time with the data in a composed way, it brings about temporal relational databases. After a few decades, requirement emerges to capture real world elements with this time attribute, which brings temporal object-oriented data models. A discussion focuses on the following:

- A concise talk on the fundamental relational databases, which are usedall around.
- An inspection on need of temporal databases, as what issues were there in consequence of which we require temporal database.
- Examination and understanding of various time dimensions or time aspects of the temporal models.
- Many temporal data models analyzed. Some of which are quite recently implemented and some are not implemented
- A lot of research done on both types of temporal: relational and object oriented models, but none of the models satisfies all the desired objectives as an individual model. So, use a set of data models for the best outcomes.

Areas that should be tended to in not so distant future

- Performance tuning is a big task that needs to be think through in temporal databases. Volume of temporal database is huge, which takes a lot of time for processing. Data may be heterogeneous in nature, which will again affect the performance of operations on temporal database. Temporal joins, temporal indexes, parallel computing are the areas where performance tuning of temporal databases can be done.
- Temporal data models incorporates tuple time stamping, which enhances redundancy in the database and hence increasing size of database & hence tables. To work upon, it will be more difficult to have full tables in main memory. Temporal data models implementation using Distributed environments and parallel computing can resolve the same.

6. REFERENCES

[1] Brooks, F. P. "The Analytic Design of Automatic Data Processing Systems." PhD. Dissertation. Harvard University, May 1956.

[2] Codd, E. F. "Further Normalization of the Data Base Relational Model," in Data Base Systems. Vol. 6 of Courant Computer Symposia Series. Englewood Cliffs, N.J.: Prentice Hall, 1972.

[3] Wiederhold, G., J.F. Fries and S. Weyl. "Structured Organization of Clinical Data Bases," in Proceedings of the AFIPS National Computer Conference. AFIPS. 1975, pp. 479–485.

[4] Kimball, K. A. "The DATA System." Master's Thesis, University of Pennsylvania, 1978.

[5] Jones, S., P. Mason and R. Stamper. "LEGOL 2.0: A Relational Specification Language for Complex Rules." Information Systems, 4, No. 4, Nov. 1979, pp. 293–305

[6] Clifford, J. "A Model for Historical Databases," in Proceedings of Workshop on Logical Bases for Data Bases. Toulouse, France: Dec. 1982.

[7] Ben-Zvi, J. "The Time Relational Model." PhD. Dissertation. Computer Sc. Department, UCLA, 1982.

[8] Clifford, J. and D. S. Warren. "Formal Semantics for Time in Databases." ACM Transactions on Database Systems, 8, No. 2, June 1983, pp. 214–254.

[9] Snodgrass, R. T. and I. Ahn. "A Taxonomy of Time in Databases," in Proceedings of ACM SIGMOD International Conference on Management of Data. Ed. S. Navathe. Association for Computing Machinery. Austin, TX: May 1985, pp. 236–246.

[10] McKenzie, E. L. "Bibliography: Temporal Databases." ACM SIGMOD Record, 15, No. 4, Dec. 1986, pp.

[11] Ariav, G. "A Temporally Oriented Data Model." ACM Transactions on Database Systems, 11, No. 4, Dec. 1986, pp. 499–527.

[12] Tansel, A. U. "Adding Time Dimension to Relational Model and Extending Relational Algebra." Information Systems, 11, No. 4 (1986), pp. 343–355.

[13] Clifford, J. and A. Croker. "The Historical Relational Data Model (HRDM) and Algebra Based on Lifespans," in Proceedings of the International Conference on Data Engineering. IEEE Computer Society. Los Angeles, CA: IEEE Computer Society Press, Feb. 1987, pp. 528–537.

[14] Segev, A. and A. Shoshani. "Logical Modeling of Temporal Data," in Proceedings of the ACM SIGMOD Annual Conference on Management of Data. Ed. U. Dayal and I. Traiger. Association for Computing Machinery. San Francisco, CA: ACM Press, May 1987, pp. 454–466.

[15] Navathe, S. B. and R. Ahmed. "TSQL-A Language Interface for History Databases," in Proceedings of the Conference on Temporal Aspects in Information Systems. AFCET. France: May 1987, pp. 113–128.

[16] Rowe, L. and M. Stonebraker. "The POSTGRES Papers." Technical Report UCB/ERL M86/85. University of California. June 1987.

[17] Sadeghi, R., W. B. Samson and S. M. Deen. "HQL — A Historical Query Language." Technical Report. Dundee College of Technology. Sep. 1987.

[18] McKenzie, E. L. "An Algebraic Language for Query and Update of Temporal Databases." PhD. Dissertation. Computer Science Department, University of North Carolina at Chapel Hill, Sep. 1988.

[19] S.K. Gadia, "A homogeneous relational model and query languages for temporal databases," ACM Trans. Database Systems, vol. 13, no. 4, pp.418-448, Dec. 1988.

[20] Lorentzos, N. A. "A Formal Extension of the Relational Model for the Representation of Generic Intervals." PhD. Dissertation. Birkbeck College, 1988.

[21] Lorentzos, N. A. and R. G. Johnson. "Requirements Specification for a Temporal Extension to the Relational Model." Data Engineering, 11, No. 4 (1988), pp. 26–33.

[22] Bhargava, G. and S. Gadia. "Achieving Zero Information Loss in a Classical Database Environment," in Proceedings of the Conference on Very Large Databases. Amsterdam: Aug. 1989, pp. 217–224.

[23] Navathe, S. B. and R. Ahmed. "A Temporal Relational Model and a Query Language." Information Sciences, 49 (1989), pp. 147–175.

[24] Sarda, N. L. "Extensions to SQL for Historical Databases." IEEE Transactions on Knowledge and Data Engineering, 2, No. 2, June 1990, pp. 220–230.

[25] Thompson, P. M. "A Temporal Data Model Based on Accounting Principles." PhD. Dissertation. Department of Computer Science, University of Calgary, Mar. 1991.

[26] Jensen, C. S., L. Mark and N. Roussopoulos. "Incremental Implementation Model for Relational Databases with Transaction Time." IEEE Transactions on Knowledge and Data Engineering, 3, No. 4, Dec. 1991, pp. 461–473.

[27] Jensen, C. S., M. D. Soo and R. T. Snodgrass. "Unification of Temporal Relations." Technical Report 92-15. Computer Science Department, University of Arizona. July 1992.

[28] Snodgrass, R. T. "Temporal Databases," in A. U. Frank, I. Campari, and U. Formentini (eds.), Theories and Methods of Spatio-Temporal Reasoning in Geographic Space. Vol. 639 of Lecture Notes in Computer Science. Springer Verlag, 1992. pp. 22–64.

[29] Snodgrass, R. T., S. Gomez and E. McKenzie. "Aggregates in the Temporal Query Language TQuel." IEEE Transactions on Knowledge and Data Engineering, 5, Oct. 1993, pp. 826–842.
[30] S. Jensen, J.Clifford, R.Elmasri, S.K. Gadia, P. Hayes, and S.Jajodia "A consensus glossary of temporal database concepts", eds., Technical Report R 93-2035, Dept. of Mathematics and Computer Science, Inst. for Electronic Systems, Denmark, Nov. 1993.
[31] Jensen, C. S., J. Clifford, R. Elmasri, S. K. Gadia, P. Hayes and S. Jajodia [eds]. "A Glossary of Temporal Database Concepts." ACM SIGMOD Record, 23, No. 1, Mar. 1994, pp. 52–64.
[32] Vincent S. Lai, Jean-Pierre Kuilboer and Jan L. Guynes, "Temporal Databases: Model Design and Commercialization Prospects", DATABASE, Vol. 25, No. 3, August 1994.
[33] R. Elmasri and S.Navathe, "Fundamentals of Database Systems", Benjamin/Cummings 1994.
[34] DebabrataDey, Terence M. Barron, and Veda C. Storey, "A conceptual model for the logical design of temporal databases", Decision Support Systems 15 (1995), pp. 305-321, Elsevier Science B.V 1995
[35] Christian S. Jensen, Richard T. Snodgrass, and Michael D. Soo, "The TSQL2 Data Model" pp. 361-395 1995
[36] C.S. Jenson and R. T. Snodgrass, "Temporal Data Management", IEEE Transactions on Knowledge and Data Engineering, 11(1):36-44, January/ February 1999.
[37] H. Gregersen and C. S. Jensen, "Temporal Entity-Relationship Models-A Survey", IEEE Transactions on Knowledge and Data Engineering, Vol. II, Issue 3, pp. 464-497, 1999.
[38] Shailender Kumar, Rahul Rishi, "A relative analysis of modern temporal data models", 3rd International Conference on Computing for Sustainable Global Development, pp. 2851- 2855, March 2016
[39] Shailender Kumar, Rahul Rishi, "Retrieval of Meteorological Data using Temporal Data Modeling", Indian Journal of Science and Technology, Vol 9(37), October 2016.