Maintaining Integrity Constraints in Spatiotemporal Databases

Kyung Do Moon, SungKu Woo, ByungCheol Kim, KeunHo Ryu
Database Laboratory, Chungbuk National University
Cheongju Chungbuk, 361-763, Korea
kdmoon@chol.com, wsk527@naver.com, bckim@ok.ac.kr, khryu@dblab.cbu.ac.kr

Abstract: Spatiotemporal phenomena are ubiquitous aspects of real world. In the spatial and temporal databases, integrity constraints maintain the semantics of specific application domain and relationship between domains when proceed update in the database. Efficient maintenance of data integrity has become a critical problem, since testing the validity of a large number of constraints in a large database and after each transaction is an expensive task. Especially, in spatiotemporal domain, data is more complex than traditional domains and very active. Additionally, it is not considered that unified frameworks deal with both spatial and temporal properties to handle integrity constraints. Therefore, there need a model to maintain integrity constraints in the unified frameworks and enforcement and management techniques in order to preserve consistence.

Keywords: Spatiotemporal databases, Spatiotemporal predicate, Integrity constraints, Moving objects

1. Introduction

Spatiotemporal phenomena are ubiquitous aspects of real world. In today's computer systems, there are required to store more complex data such as temporal and spatial information. When spatial and temporal information are modeled into databases, it is important to note that the databases have to be consistence. In databases, integrity constraints maintain the semantics of specific application domain such as temporal and spatial and relationship between that domain when proceed update in the database [3, 4]. Up to now, most of databases systems can only provide integrity to keep consistence partially. Because, efficient maintenance of data integrity has become a critical problem, since testing the validity of a large number of constraints in a large database and after each transaction is an expensive task. Especially, in spatiotemporal domain, data is more complex than traditional domains and very activity. And also there does not consider about unified frameworks that deal with both spatial and temporal properties to handle integrity constraints. Therefore, there need a model to maintain integrity constraints in the unified frameworks and enforcement and management techniques in order to preserve consistence.

In this paper, we propose a spatiotemporal data model to represent integrity constraints that define the types of integrity constraints and manipulate operations in the unified frameworks. We also describe specifying integrity constraints to language that is a user interface in order to handle integrity constraints enforcement and management. Integrity constraints enforcement means that databases must include some method to ensure that integrity constraints are always satisfied after the application of a transaction. Integrity constraints management includes verification, repairing and handling. When new integrity constraints are defined on a databases by user, it has to be checked if the constraints themselves are validate syntactically and semantically. Therefore, implementation of integrity constraints through enforcement and management that based on the proposed model, we can expect to maintain integrity constraints in spatiotemporal databases.

The paper is organized as follows. Section 2 we mention some related work on the databases to represent integrity constraint. Section 3 defines the integrity constraints in the spatiotemporal databases. Section 4 present spatiotemporal integrity constraints specification language. Section 5 describes the integrity constraints managing module that consist of definition and validation. Section 6 applies the integrity constraints module to moving object management system. Section 7 contains our conclusion.

2. Related works

In general, the focal point of maintaining integrity constraints in database system depends on the domain, relationship and business rules [4]. Integrity constraints are classified by inherent integrity constraint, implicit integrity constraint and explicit integrity constraint that are in terms of where they are applied [3, 4]. Implicit integrity constraint is concerned about data model to represent domain, and relationship between them. Explicit integrity constraint is to describe data definition language to specify integrity constraints. Explicit integrity constraint, called business rule, depends on specific application such as geographic information system, environmental information system, multimedia information system and traffic control system. Most of previous works were concerned about implicit integrity constraint and explicit integrity constraint that how to enforce and manage integrity constraints in databases [1, 2, 5].

In cases of relational databases, there exist integrity constraints that domain and key constraints [3]. In cases of active databases, there can be thought of as coupling databases and rule based programming. The active databases rules enable many desired databases feature such as integrity constraint checking and enforcement, derived data maintenance, alerts, authorization checking and versioning [4]. In case of spatial databases, there have a lot of works that concerned about spatial relationship such as spatial object, topological relationship between two objects [2]. In cases of spatiotemporal databases, there exist previous works that how to express spatiotemporal predicate into query language and to define spatiotemporal operator that deal with geometry change over time [1, 2]. And also there exist some previous works that moving object on road network to restrict behavior of moving direction or distance through object moving over time [5].

3. Representation of integrity constraints in spatiotemporal data model

In this chapter, we describe spatiotemporal data model to represent integrity constraints and we show how to classify and define each of integrity constraints.

1) Spatiotemporal data model

Spatiotemporal databases deal with geometries change over time [1]. There can be considered about point, line, and region to lift temporal that we called spatiotemporal data type. The basic types of spatiotemporal data are moving point and moving region.

mpoint = *time* → *point*
mregion = *time* → *region*

More generally, we can introduce a type constructor τ that transforms any given atomic data type α into a type $\tau(\alpha)$ with semantics as $\tau(\alpha)$ = time → α and we can denote the types *mpoint* and *mregion* also as τ(*point*) and τ(*region*), respectively.

Spatiotrmporal operations are projection into domain and range, interaction with values from domain and range and operations related to rate of change that show as Table 1 [5].

2) Integrity constraints

Integrity constraints applicable at each of these levels of spatiotemporal information systems development are inherent integrity constraints, implicit integrity constraints, and explicit integrity constraints respectively. Especially, explicit integrity constraints that called user-defined integrity constraints are classified such as spatiotemporal integrity constraints, topological integrity constraints and aggregate integrity constraints. spatiotemporal integrity constraints consist of local and global integrity constraints that interact between unary or binary object. Local integrity constraints restrict derived values from single spatiotemporal object. Global integrity constraints restrict derived values from relationship between two spatiotemporal objects. Topological integrity constraints are special case of relationship between spatial objects change over time and deal with spatiotemporal predicate. Spatiotemporal predicates can be classified by basic and advanced predicates. Basic spatiotemporal predicate can be defined by temporal lifting and aggregation. The basic spatiotemporal predicates are *Disjoint, Meet, Overlap, Equal, Covers, CoveredBy, Contains, Inside*. The choice of defaults is to some degree debatable, for example, the symmetric definitions for *Meet* and *Overlap* are considered too restrictive, so we can easily add further predicates with different aggregation behavior as *Meets, MetBy, Overlaps, OverlappedBy*. A further basic spatiotemporal predicate is the *True* that yield true for two arbitrary spatiotemporal objects. Advanced spatiotemporal predicates are sequences of basic spatiotemporal predicates and can be defined by instant predicates and period predicates. Instant predicates can be true for an instant in time and period predicates can only hold for a period of time. Aggregate integrity constraints restrict derived values through aggregate function from spatiotemporal objects.

Integrity constraints can be defined by spatiotemporal function, spatiotemporal predicate, spatiotemporal constraint, and constraint instance.

Table 1. Operations for spatiotemporal objects.

Class	Operations
Projection to Domain/Range	deftime, rangevalues, location, trajectory, routes, traversed, inst, val
Interaction with Domain/Range	atinstant, atperiods, initial, final, present, at, atmin, atmax, passes
Rate of Change	derivative, speed, turn, velocity

Definition 1 (Constraint instance).
Let $S = S_1, S_2, ..., S_n$ be a schema for constraint and $s_1, s_2, ..., s_n$ be the respective instances of the constraints. There exist constraint instance s in the S, then violate semantic integrity constraints.

We refer to constraint instance s of S, there need to interface to define S and to validate s.

Definition 2 (Spatiotemporal constraint).
A spatiotemporal constraint is denoted by a clause $P_1 \vee ... \vee P_k$ where P_i's are spatiotemporal predicates. We assume there are no duplicate predicates in a constraint. The same constraint may also be represented in conjunctive form as $\neg P_1 \wedge ... \wedge \neg P_k \rightarrow$, where "$\rightarrow$" denote the logical implication.

Definition 3 (Spatiotemporal predicate).
A spatiotemporal predicate is of the form $X \theta Y$, where X is an attribute, θ is a comparison operator from $\{\geq, >, \neq, =, <, \leq\}$, and Y is a constant.

Definition 3 (Spatiotemporal function).

We utilize the fact that spatiotemporal object take value from comparison, spatiotemporal operation, topological operation that can only identify spatiotemporal attribute and relationship.

4. Specifying integrity constraints

After defining and representing integrity constraint, specification of integrity constraint is in hand. Integrity constraint specification includes enforcement(checking and maintenance) and management. Specification of integrity constraint to the language is decided by requirement of real world operation that related with information about business rule. These requirements are different from each of application, but generally include the features such as specification of integrity constraints, verification of integrity constraints, definition of integrity constraints, and maintaining concurrency. In this chapter, we present specifying integrity constraints to extend SQL. There have been two types of specifying integrity constraints that are specifying definition of integrity constraints and specifying validation of integrity constraints.

1) Specifying definition of integrity constraints

Integrity constraint specification means that databases must incorporate some mechanisms to ensure that integrity constraints are always satisfied after the application of transaction. It is possible to represent integrity constraint in spatiotemporal databases using definition of integrity constraint to specify that. The expressive power of integrity constraint reflects to reduce performance of system and to increase complexity of system. So, we have to consider about some feature when we design the integrity constraints that are described as following.

1. Event: for the definition of specifying integrity constraints, event means that time when there validate integrity constraint. There are some types of integrity constraints that are data modification such as insert, delete, update and data retrieval such as select and time such as absolute time, repeated time, periodic time and user call.
2. Condition: Condition is to validate databases state that verify before event termination. The method of specifying conditions are databases predicate, restricted predicate, databases query and application procedure.
3. Action: There exist a value that is true where validate integrity constraint when there occurred event. The types of operation are data modification operation, data retrieval operation, transaction command, wrong, and application procedure.

Semantics of integrity constraints describe meaning of integrity constraints that depend on taxonomy of constraints. The taxonomy of spatiotemporal integrity constraint is described as following.

1. Local integrity constraints: restrict value for operation that derived from single spatiotemporal object. The types of operations that using this constraint are *SIZE, EXTENT, PERIMETER, LENGTH, DURATION, AREA, TRAJECTORY, TRAVERSED, ROUTE, LOCATION, SPEED, TURN.*
2. Global integrity constraints: restrict value for operation that derived from two spatiotemporal objects. The types of operations that using this constraint are *DISTANCE, DIRECTION, INTERSECTS.*
3. Topological integrity constraints: restrict value for operation that derived from topological relationship between the two spatiotemporal objects. The types of operations that using this constraint are *DISJOINT, MEET, OVERLAP, EQUAL, COVERS, COVEREDBY, CONTAINS, INSIDE.*
4. Aggregate integrity constraints: aggregate value for operation that derived from between the spatiotemporal objects. The types of operations that using this constraint are *MIN, MAX, AVG CENTER.*

2) Specifying validation of integrity constraints

Integrity constraint management includes verification, repairing and handling. When new integrity constraints are defined on a databases, it has to be checked if the constraints themselves are validate syntactically and semantically. This process is called integrity constraint verification. Verification of implicit integrity constraints based on the defined event on constraint base. Verification of explicit integrity constraints based on the user call to validate specific database schema and integrity constraints.

5. Integrity constraints managing module

In this chapter, we present a module that is to define and validate integrity constraint. There have two modules that are definition module and validation module that depicted as fig 1.

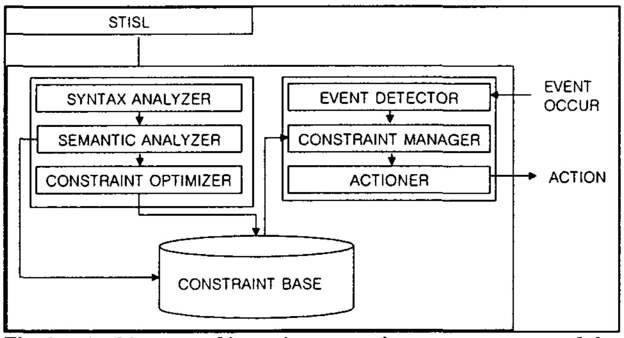


Fig. 1. Architecture of integrity constraints management module

Integrity definition module is to analyze specifying integrity constraints language that defined section 4 and store that constraint to the constraint base. Integrity validation module is to validate integrity constraint when event occur.

1) Integrity constraint definition module

Integrity constraints definition module consist of syntax analyzer and semantic analyzer. Syntax analyzer is to analyze syntactic error for specifying integrity constraint language and include LEXER and PARSER to generate parser tree. Semantic analyzer is to analyze semantic error for specifying integrity constraint language and include TRANSFORMER and SEMANTIC OPTIMIZER. Semantic constraint optimization is a technique that uses integrity constraints to improve the efficiency of constraint evaluation. Most of the semantic constraint optimization is applied to a query after the query is posed to a database, but before the syntactic constraint optimization is performed.

2) Integrity constraint validation module

Integrity constraints validation module include event detector, constraint manager, and actioner. Event detector is to identify event that occurred from query processor or user call or time period. Constraint manager is to load constraint from constraint base and return the constraint that received from event detector. Actioner executes the operations that notify on console mode to user and proceed ABORT.

6. Application

In this chapter, we apply presented integrity constraint module to moving object in networks. Moving object model offers paths over the network graph as a basic concept that called route. It allows one to distinguish between simple and dual routes and to describe possible violation for moving object such as vehicles, at junction.

At a junction, depict in Fig 2, between route A and B various transitions may be possible or routing violation for moving object. For example, in the transition matrix, below the Fig 3, only 1 can be possible for routing, otherwise routing violation. The routing violation can be expressed by *DIRECTION, ENTER* predicate.

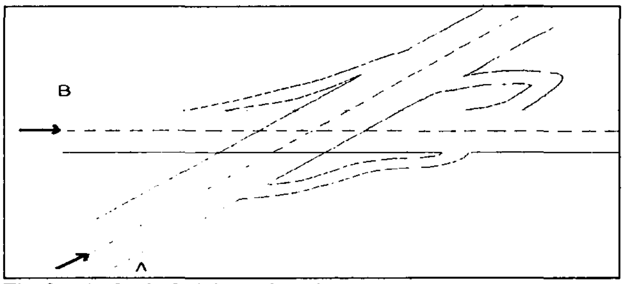
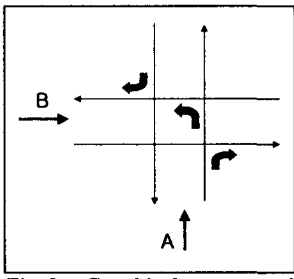


Fig. 2. A physical highway junction



	Au	Ad	Bu	Bd
Au	1	0	1	1
Ad	0	1	0	1
Bu	0	0	1	0
Bd	0	0	0	1

Fig. 3. Graphical representation and transition matrix

The *DIRECTION* predicate indicates direction that is up and down in dual highway network. The *ENTER* predicate indicates moving behavior that expressed by sequence of predicate as *Disjoint, meet, Inside*. The *Disjoint* predicate holds time period during some time as $[t_1, t_2]$. The *meet* predicate holds specific time point as t_2. The *Inside* predicate holds time period during some time as $[t_2, t_4]$. So, if the user defines the integrity constraint with above predicate, then there can be avoided violations such as routing. Therefore, using integrity constraint managing module, we can define the integrity constraints for specific application, thus there can be preserved integrity constraint in the databases.

7. Conclusions

Integrity constraints maintain the semantics of specific application domain such as geographic information system, environmental information system, multimedia information system and traffic control system. This paper considers about management of integrity constraints module that is defining and validating integrity constraints. And we apply this module to moving object system to control the moving behavior. However, it dose not support all of the spatial, temporal, and spatiotemporal functions. So, further work is need to extend for spatial, temporal, and spatiotemporal functions. In addition, effective optimize technique that proceed specifying integrity constraints language for performance evaluation can be suggested future research.

References

[1] M. Koubarakis, T. Sellis et al., 2003. Spatio-Temporal Databases, Springer, Germany.

[2] M. Erwig, M. Schneider, 2002. Spatio-Temporal Predicates, *IEEE Transactions on Knowledge and Data Engineering, Vol. 14*, No. 4, pp.881-901.

[3] Y. G. Lee et al., 1998. Spatial Data integrity in Spatial Information System, *Proc. Intl. Conf. on Applied Modeling and Simulation*, pp.267-271.

[4] S. Cockcroft, 1998. User Defined Spatial Business Rules : Storage, Management and Implementation, *Proc. Spatial Information Research Centre's 10th Colloquium*, New Zealand , pp. 73-82.

[5] R. H. Güting, V. T. de Almeida, Z. Ding, 2004. Modeling and Querying Moving Objects in Networks, *Informatik-Report 308*, Fernuniversität Hagen, Germany.