

Navigating through Hierarchical Change Propagation in Spatiotemporal Queries

Giorgos Mountrakis, Peggy Agouris, Anthony Stefanidis

*Dept. of Spatial Information Engineering and
National Center for Geographic Information and Analysis (NCGIA)
University of Maine*

5711 Boardman Hall #348, Orono, ME, 04469, USA

Phone: (207) 581-2188, Fax : (207) 581-2206

{giorgos, peggy, tony}@spatial.maine.edu

Abstract

In spatiotemporal applications, meaningful changes vary according to object type, level of detail, and nature of application. In this paper, we introduce a dynamic classification scheme and its interaction with a spatiotemporal model, to describe change propagation in spatiotemporal queries. Our classification is based on identifying three levels of change, ranging from simple verification of the continuous existence of an entity to the identification of change and the detailed description of this change. In order to provide a transformation function between these levels, we introduce a tree-based hierarchy in the spatial and temporal domains. This hierarchy is dynamic and continuously updated as new information arrives. The user has the flexibility to navigate through different resolution representations by manipulating two functions, the Minimum Spatial Element and the Minimum Temporal Element. In the temporal domain, queries are further more decoded as point-based and interval-based. In doing so we address the problem of time continuity within a spatiotemporal query context.

1. Introduction

A significant functionality of a spatiotemporal model is the ability to express explicitly changes that may be found implicitly in various sources and object representations. Defining a common measurement framework to express spatiotemporal change is a challenging task, as meaningful changes vary according to level of detail, nature of application, and type of object. Furthermore, the context of change, as determined by applications and user needs and queries, can be considered to be spatial in nature, temporal, or a

combination of both. The communication of this change requires the consideration of users who have diverse needs, and exhibit various levels of understanding of the content and organizational schemata of the databases that they query.

In this paper we address the problem of spatiotemporal database queries by assisting *user navigation* in multiple spatiotemporal granularities. We introduce a novel concept in detecting and propagating change, the *hierarchical classification of change resolution*. We present a dynamic classification scheme to support change monitoring and communication of spatiotemporal objects. We also employ a sliding rule paradigm to support navigation through multiresolutional data. Related work has been performed in identifying and formalizing operations in the temporal domain in order to support the integration of different databases [3] and also on granularity issues [10,14]. Our approach is novel as it incorporates the concept of granularity in the query process of a spatiotemporal model. In spatiotemporal systems resolution cannot be considered static, since change resolution is defined by the query in real-time mode. Our system's structure supports these multiresolutional queries, and to enhance the obtained results, the user can navigate through different resolution's representations and guide the query process, by interacting with the system.

The paper consists of four parts. We begin by presenting the structure of our spatiotemporal model (Section 2). This structure is based on the three different levels of detail that are analyzed in Section 3. Then we introduce a tree-based hierarchy of classes, and provide applications of propagation of change in queries at the spatial, temporal and spatiotemporal domains (Sections 4,5 and 6). We summarize with the conclusions and plans for future work in section 7.

2. Structure of our Model

The model we are proposing consists of a *Geographic Identity Register*, a *Change Indexing Register* and a number of *child databases*, in order to describe specific information aspects (Fig. 1). The structure is based on the need to support spatiotemporal

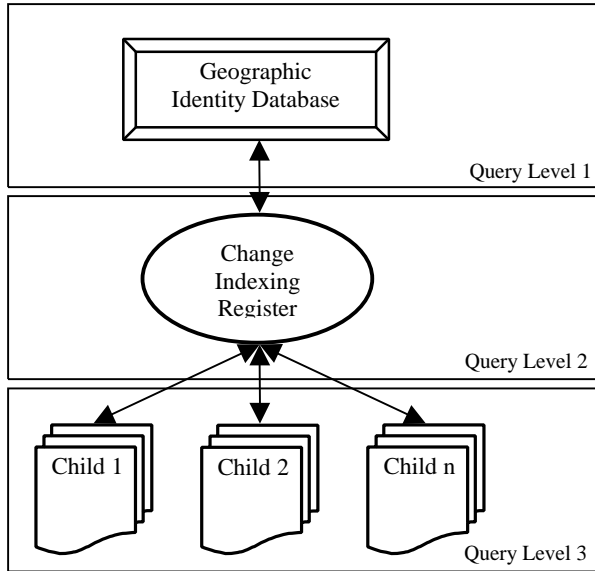


Figure 1. Model structure

queries of different levels of detail. Accordingly, change is decomposed in two levels: *entity* and *object*.

In the coarser level, geographic features are treated as entities. This means that only an abstract representation, joined with a lifespan [5] is stored, without explicit spatiotemporal information. The database that stores information about existence through time is the *Geographic Identity Database* (GID, query level 1 of Fig. 1). Actual characteristics of change at the object level, are expressed through *child databases* (query level 3 of Fig. 1). There is a parent/child relation between GID and child databases.

The selection of the child databases can be based on various criteria, depending on the application. Optimality in defining child databases is achieved if there exists:

- independence between them, to assure minimal redundancy, and
- completeness, to assure the adequate description by them of all essential aspects of change.

Accordingly, an optimal segmentation of the child database space will approximate semantically a base of the object's change space. Each database corresponds to a specific dimension of change. However, optimality is

not a necessary condition when deciding on the content of these child databases, and it is beyond the scope of this paper. Often, pre-existing constraints (e.g. standards and practice conventions) dictate the assumption of specific child databases. In our case we assume three independent components of change, generating three child databases:

- a Geometric one, containing information on outline (e.g. edges of an image)
- a Positional one, describing the position of the object (e.g. X,Y,Z coordinates and orientation information), and
- a Thematic one, describing use of the object (e.g. identifying building as residential).

The continuous communication and updates between parent and children, and consistency in corresponding records are established through the *Change Indexing Register* (CIR, query level 2 of Fig. 1). This register provides the essential multi-dimensional indexing mechanism [9], with flags to all attributes of the child databases that have been modified.

3. Categorizing Queries by Level of Detail

In this paper, we introduce a hierarchical structure to categorize spatiotemporal queries, according to their level of detail. The interaction of these three levels with the structure of the model is showed. The backbone of this hierarchy is based on the following three different types.

1) Existence of Entity

At the first level, a qualitative boolean query can be performed to find whether an object exists (Query Level 1). This is a single-source query, which can be applied repeatedly, but there is no dependency between the sources used. It is applied on the parent database, the GID. At this level, the object is treated as an entity, as described above, with no further spatiotemporal attributes. This type of queries has been addressed in the past, as identity-based queries [8].

2) Existence of Object's Change

At this level, the user can have a simple Yes/No answer to a query whether there exist any evidence of change for a specific object. This query is differential in nature, since two or more sources are required. Queries of this type [7] are executed in the Change Indexing Register, without the need to access any of the child databases.

3) Characteristics of Object's Change

This is the most detailed level, where a complete description of all the attributes of the object that changed, is provided. As this is also a differential query, at least two sources are needed. Based on the indexing structure of the Change Indexing Register, only the modified attributes of each database are accessed and returned.

4. Spatial Domain

In order to establish a common measurement framework to detect change in the spatial domain, we make use of a hierarchical structure, categorizing spatial classes, based on class containment relationships [1] (e.g. part/whole relations, container/containeed). It should be emphasized here that this does not imply the establishment of an exhaustive classification of all types of spatial classes. Instead, we aim at the establishment of a versatile, dynamic arrangement of a selected subset of spatial classes that may be of interest for the specific application. We should note that the purpose of this paper is not to establish or to redefine these spatial relationships, but rather to introduce them as constraints to activate change propagation.

The hierarchy dictated by these relationships leads to

(e.g. adding counties between state and road/lake/town), or we can even introduce new instances within existing levels (e.g. adding a forest class at the same level as road/lake/town). Furthermore, complete branches of the tree may be moved within this hierarchy (e.g. moving the road branch under town). This does not affect the performance of the presented approach. As we will see later on, the sliding rule focuses on the relative arrangement of entities, without considering absolute distances between successive nodes.

This hierarchy is a non-transitive, many-to-many relationship. The distance between the nodes is not important, only the number of nodes in-between. The only rule applied is that there should be no containment type relationship between classes of the same level, because in that case the hierarchy is dissolved. As mentioned above, this basic tree structure has a dynamic form, so when new classes are introduced, the model has the flexibility to update its structure. Furthermore, depending on the application, there are cases where same classes can lead to different structures, based on specific user needs.

In order to establish a resolution reference for change in the spatial domain, the term of the Minimum Spatial Element (MSE) is presented. The MSE expresses the spatial resolution of the recorded change and can have an absolute or a relative value. For example, the MSE of a building can be a cube of 1m x 1m x 1m volume. In this case, only changes larger than that element, are

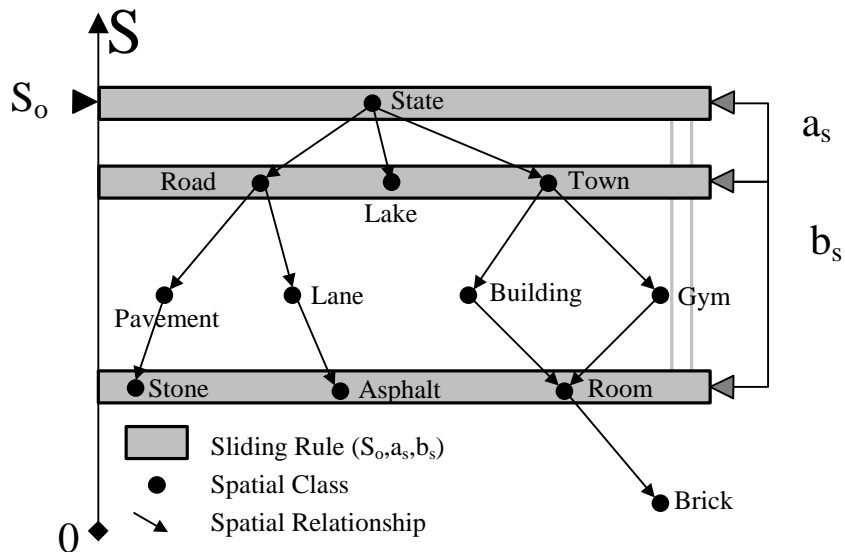


Figure 2. Sliding rule on spatial hierarchy

a tree structure of entities in the spatial domain. The resulting tree structure (Fig. 2) is dynamic. For example, we can introduce new levels in-between existing ones

returned to a query on whether something has changed in a spatial object. But the real flexibility that MSE provides is when a relative value from the previous tree

structure is assigned. For the same example, a building can have as a MSE a wing, a room, or a brick. If one expands this to other object types (classes) that might exist in the database (e.g. a town) and consider a building as a MSE for the town, then automatically a propagation process for change resolution is triggered.

The MSE functions are projected on the tree structure in a manner similar to a Sliding Rule (Fig. 2). This sliding rule has 3 parameters, namely S_o , a_s , b_s :

- S_o defines the original position of the rule and it reflects the relationship between the spatial extent of the query and its corresponding class, while
- a_s and b_s properties are zoom-in factors, describing the distance in number of nodes for the two MSE functions (Fig. 3).

Class_Building class might act as a MSE for the Class_Town class, but the object My_House, even though it is included in that database, does not have a satisfactory spatial relationship with any of the 2 objects classified as towns, simply because it is in another state.

The levels of detail and their implicit classification interact in a query process. They do so by decomposing the query into three parameters, namely resolution, spatial extent, and temporal extent, and by transforming the query into a range of resolutions:

Query[Level of Detail, Spatial Extent, Temporal Extent].

So if the query is “How has Orono changed last year?”, we have:

Hierarchy in Spatial Classes

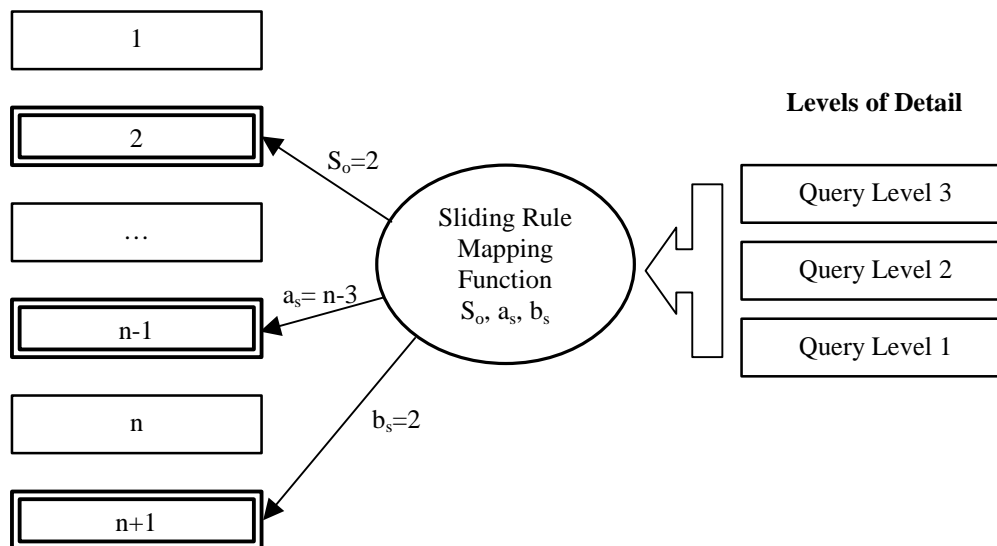


Figure 3. Sliding rule mapping function

Initial approximations for the two zoom-in parameters are set to 1. This corresponds to selecting the nodes in the two levels directly under the one corresponding to S_o to perform the query. The process of querying can be dynamic, with the user reentering combinations of parameters until the satisfied resolution is reached. Furthermore, specific branches may be excluded or introduced based on the obtained results.

In addition to the MSE function between the classes, the same type containment relationships between the objects should exist. By combining class level and object level, we manage to avoid conceptual conflicts. For example an object Memorial_Gym can have a part/whole relation with the object Orono_Town, but there is no MSE relation of their classes, Class_Gym and Class_Town respectively. Furthermore, the

Level of Detail = 3,
Spatial Extent = Orono,
Temporal Extent = Last Year

Leaving aside for the time the temporal extent, the rule that we follow for the query transformation in the spatial domain is straightforward. If the level of detail is one, meaning that the query refers to the question of existence of an entity, then the query is addressed at this level without using the MSE function or the tree structure. If the level of detail is two or three, then the approach that our system employs is to go down one class in the hierarchy and go up one level of detail.

Here is an application of the previously described propagation rule. Let's assume that our database comprises 2 towns, 450 buildings, 5400 rooms and 4

gyms. At the class level, through the sliding rule, the user defines:

Class_Building = MSE (Class_Town),
 Class_Room = MSE (Class_Building)

At the object level, spatial relationships between the objects from time t_1 until time t_2 exist:

Boardman_Hall = Part_of (Orono_Town, t_1, t_2),
 GIS_LAB = Part_of (Boardman_Hall, t_1, t_2)

After the above MSE classification, we have the following hierarchy:

- A) Class_Town
- B) Class_Building
- C) Class_Room

When a query is performed at the spatial domain, then a letter inherited from the class hierarchy and a number representing one of the three levels of detail is assigned. So if the query is “Does the town of Orono exist in 1921?”, the code for this query is A1: A for the Class_Town, and 1 for the level of detail, which in this case is Existence of Entity. All the queries of level 1 are addressed at that level, without using the MSE classification. If the query is “Has the town of Orono changed the past 4 years?”, the code for this query is A2. By following the propagation rule A becomes B, meaning that the query is transferred to the all the objects of the building class, that satisfy the required spatial relation, and 2 becomes 1, to compensate for change at the spatial class level. So the previous query is rephrased to “Do all the buildings of Orono exist and none was added for the past 4 years?”. One could claim that the answer to this question might be incomplete, since some buildings might have changed (e.g. add one roof), or a new gym might be built within the 4 years. That can be the case, but the way the user has defined the hierarchical resolution of change through the MSE functions, shows that only the Class_Building class is of interest to him/her, and that the class of the buildings is the only MSE for the class of the towns. If further information is required, a new scenario would be to terminate the two MSE functions and replace the with:

Class_Room = MSE (Class_Town),
 Class_Gym = MSE (Class_Town)

The last type of queries is a level 3 type, for example “How has the town of Orono changed from 1988 until 1996?”. We follow the same approach, and from A3 we go to B2, and after reapplying the propagation formula, B2 is converted to C1. So the query we answer to is “Do

all the rooms of Orono exist and none was added from 1988 until 1996?”.

5. Temporal Domain

From the spatio-temporal point of view, the goal of a model is to describe in an appropriate manner the spatial changes that appear over time, and the temporal changes that appear over space, based on the available information. In such a spatio-temporal system, a snapshot time model [12,13] is eventually chosen, as a result of the lack of continuous types of information in the temporal domain. Even video sequences or real-time updated databases, have a blank interval, where no information is gathered. To compensate for that, we project the continuous time model of the reality to the snapshot time model of our system, but at the same time we keep track of the relationship that this projection is establishing. Every time interval can be decomposed in two different time states, the sleep and action modes, partitioning the interval in one, two or three sectors, depending on the change format (Fig. 4).

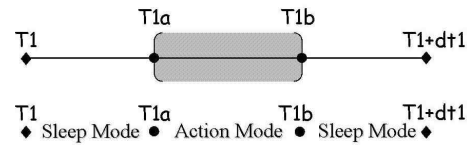


Figure 4. Sleep vs. action mode

In the above segmentation, one should note two things. First that this time line is for a specific object, and second that the interval (T_{1a}, T_{1b}) is treated like a “black box”, meaning that a label “action” is assigned to that, without including any temporal information about the behavior of change. It can be discrete, continuous, periodic or have any other form. The segmentation function is reapplied n-times, until a predefined interval, acting as a threshold is reached (Fig. 5).

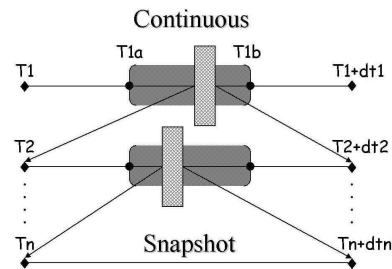


Figure 5. Segmentation based on the MTE

This predefined time interval that expresses the minimum duration of a change is introduced as the

Minimum Temporal Element (MTE). At this point we can reasonably claim that the continuous model of the reality is projected successfully on our system's snapshot model. Then, we back-project our snapshot model to the continuous time line, trying to reconstruct the object's temporal behavior, where information is available. Sequential sleep or action intervals are joined together, and the new time line is reconstructed (Fig. 6).

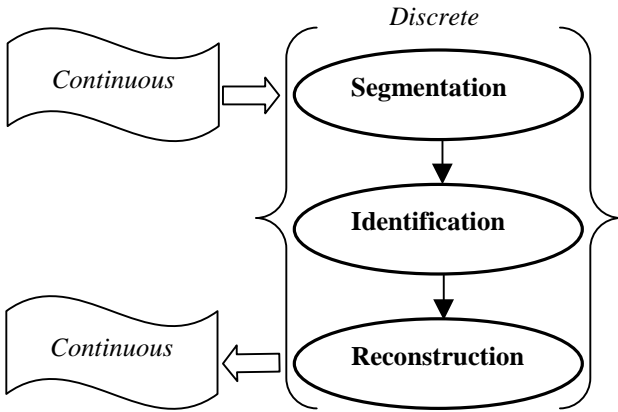


Figure 6. Reconstruction of continuous time line

With this, we employ a multi-resolution time line with a coarser resolution, where no change appears, and finer where changes are detected. When new information arrives, and T_{new} is the new time snapshot, found in the time interval (T_i, T_{i+1}) , then if

$$(T_{i+1} - T_i) < MTE$$

the above process is reapplied locally.

The Minimum Temporal Element can have an absolute or a relative value, similar to the concept of the MSE. When using an absolute value, the user is establishing a scaling factor in the temporal domain. For example, if the user wants information to use in a cadastre:

$$MTE (\text{Building}) = 1 \text{ month}$$

If the user wants a more detailed description, to detect the progress of a building's construction, then the time interval should be smaller:

$$MTE (\text{Building}) = 1 \text{ day}$$

The relative value implies a categorization of time in classes. The temporal relationship in the class level is a part/whole one. Within our model, time is considered to be a linear function, so no relation at the object level is necessary, since that relation is inherited from the class

level. In order to use this temporal hierarchy, we try to associate the temporal extension of each query to a specific temporal class. The rule that is used is that the class that is assigned, is the most detailed one. So the temporal class of the query "How has Orono changed from 1988 to 1996?" is addressed at the year class. If the query was "How has Orono changed from 11/06/1988 to 09/07/1996?", then the associated class would be the most detailed, in this case the class day. Issues that describe incompatibility between different time systems or systems with different granularity are not the focus of this paper.

In the temporal domain, queries can be separated into point-based and interval-based [2,3,11]. The point-based ones assign a timestamp during the query process (e.g. "Does the car exist at 21:30, 11/09/99?"). Queries of this type can address change at the level of existence of an entity (Query Level 1). This can be reasonably assumed, since single-source type information can provide an answer. But for the other two levels of queries, existence of object's change and characteristics of object's change, we claim that no timestamp can be assigned since the format of the query is a differential one, meaning at least two sources are required. Even if sometimes a timestamp seems to be given, like "Did the car move at 21:30, 11/09/99?", a comparison through a duration of time is implied; for instance since it was last observed or was first detected, etc. So the queries of the second and the third level are treated exclusively as interval-based, meaning that a duration of time has to be examined for a valid answer. Queries of level one can be treated either as point-based or interval-based queries, depending on their formulation.

Another problem is that there is no continuous information, so an interval has to be projected in a discrete model. The flowchart (Fig. 7) describes the

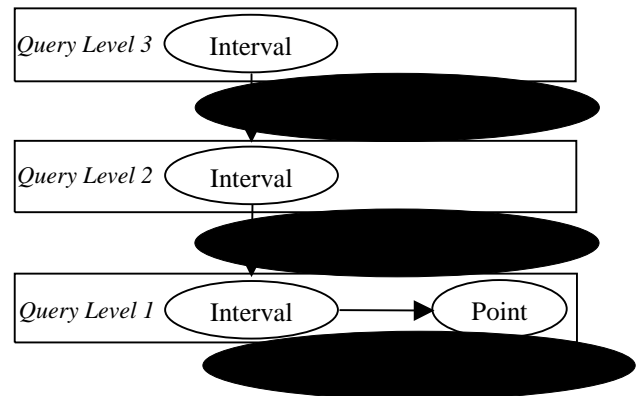


Figure 7. Interval-based vs. point-based queries

operations that have to be done in the temporal domain. First the same propagation rule applied in the spatial

domain, is used. So if we have the two MTE functions as

Class_Month = MTE (Class_Year),
 Class_Day = MTE (Class_Month)

and the query is “How has Orono changed in 1984?”, the query is transformed into “Did Orono change from January,1984 until December,1984?”. Note that through the MTE function the interval is transformed to the units of the MTE. After reapplying the rule, the

query is transformed to “Did Orono exist from 1/31/84 until 12/31/84 ?” (Query Level 1). After the two transformations at the query levels, one more is necessary to project the interval (1/31/84-12/31/84) to our snapshot model. Based on the temporal class that the temporal extension belongs to (in our case it is Class_Day), a segmentation is applied by using the class as an MTE for that interval. So finally our query has the form “Did Orono exist every day from 1/31/84 until 12/31/84?”, which is a level one, point-based query.

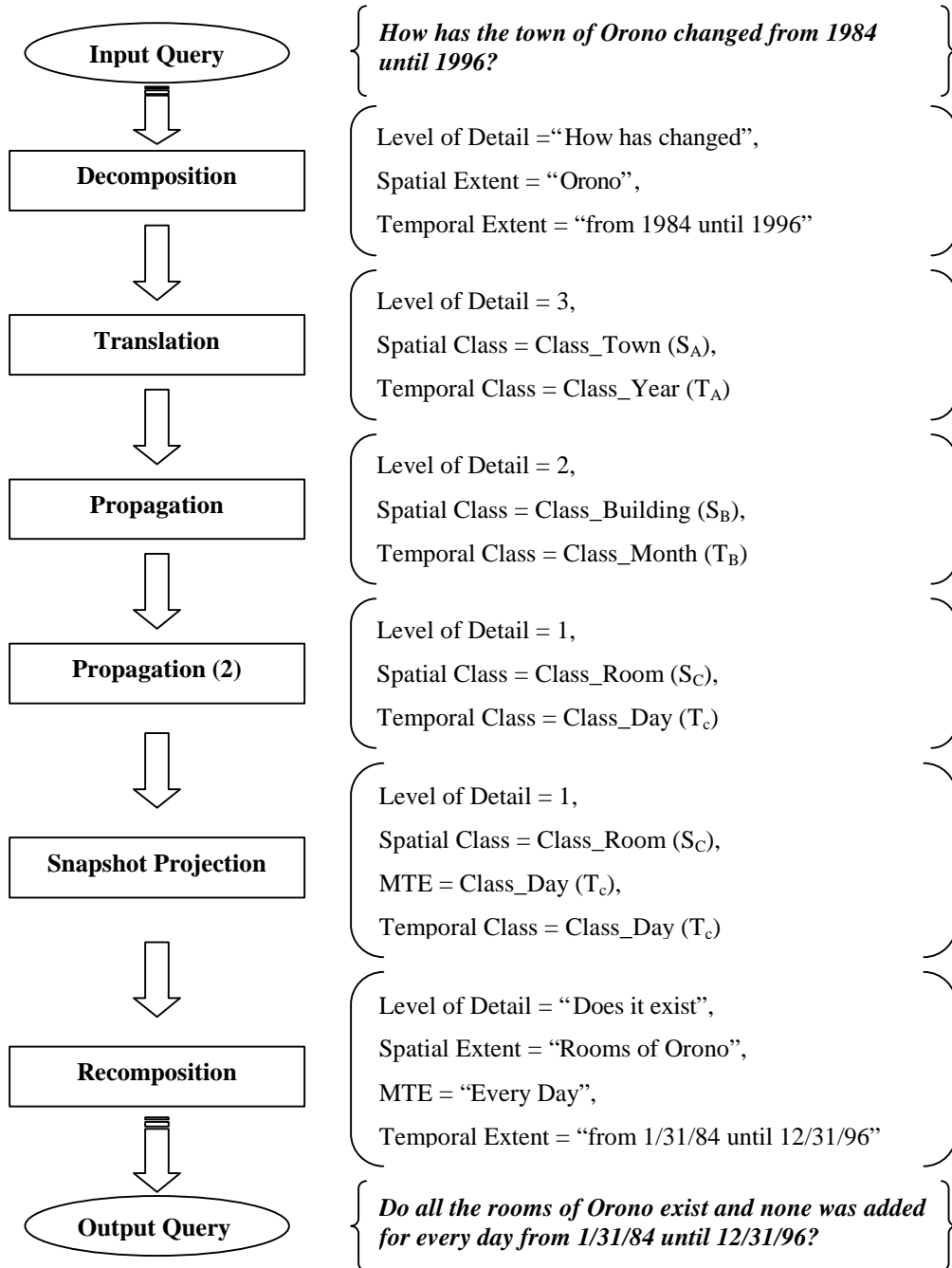


Figure 8. Workflow of the reconstruction query process

6. Spatio-Temporal Domain

If we combine the MSE and the MTE functions, queries at the spatiotemporal domain can be introduced and addressed within our system. Let's assume that we have the following hierarchy, after the two minimum element functions are applied:

Level of Detail	MSE	MTE
1) Existence of Entity	S _A) Class_Town	T _A) Class_Year
2) Existence of Object's Change	S _B) Class_Building	T _B) Class_Month
3) Characteristics of Object's Change	S _C) Class_Room	T _C) Class_Day

An example is presented (Fig. 8) to explain the whole process that the system follows, based on functions previously introduced.

The initial parameters of the sliding rule, are set to:

$$S_0 = \text{Spatial Extent's corresponding class,} \\ a_s = 1, b_s = 1$$

$$T_0 = \text{Temporal Extent's corresponding class,} \\ a_t = 1, b_t = 1$$

If the user is not satisfied with the results' resolution, a new set of parameters can be given. This dynamic navigation process can be reapplied, until the desired area of interest is established in the resolution domain.

Another way to approach the query resolution in the spatiotemporal domain is to construct a 2D graph, with axes representing the resolution in each domain. Depending on the level of detail, a point-based or an interval-based area of interest in the resolution domain is assigned (Fig. 9a.). In some cases temporal resolution is undefined, for example "How much has Boardman Hall changed?" (Fig. 9b.). If the spatial resolution is undefined, for example "How much has the year 1999 changed?", then the graph takes the form of figure 9c.

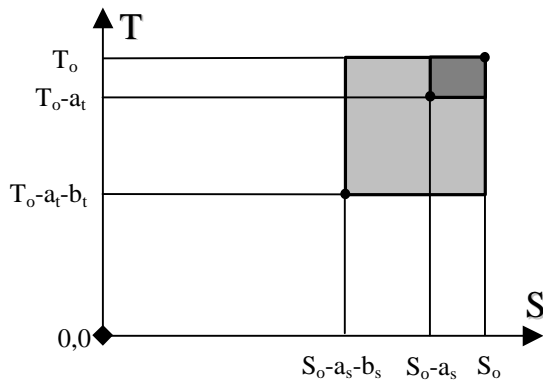


Figure 9a. Spatial and temporal resolution defined

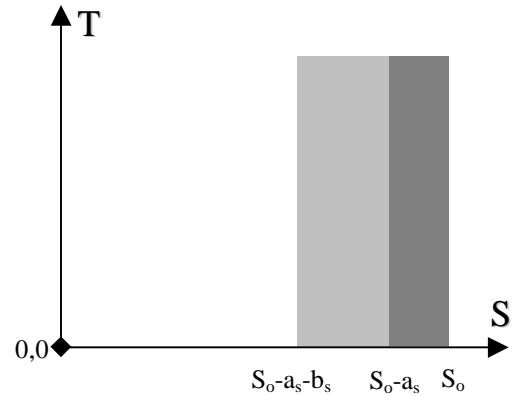


Figure 9b. Temporal resolution undefined

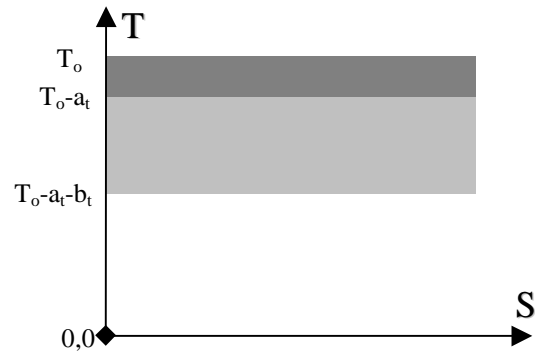


Figure 9c. Spatial resolution undefined

7. Conclusions and future work

In this paper, we presented the use of a dynamic classification scheme to describe change and its propagation in spatiotemporal queries. Two minimum elements were introduced in the spatial and temporal domains, acting as thresholds to avoid the return of redundant information. Three levels of change were identified: Existence of Entity, Existence of Object's Change and Characteristics of Object's Change. We discussed the dependence of our model architecture with these levels of detail in a query, and addressed its function under the presence of a dynamic hierarchical classification.

By using a sliding rule approach we can apply the two minimum element functions to hierarchically structured data. This defines a propagation function, whereby the initial position of the rule relates the query to specific classes of data hierarchy. Two zoom-in parameters are introduced, expressing the distance in number of nodes between the classes in the minimum element functions. This enables a user to navigate

through different representations in the resolution domain, defining his/her own semantic hierarchies. Thus, no prior knowledge of the classification scheme in our datasets is necessary. Instead, we enable the user to actually navigate in different resolution spaces that may be unknown to him/her.

Examples of change propagation in the spatiotemporal domain are presented and a 2D graph, with axes representing the resolution in each domain is described. By using the 2D graph, we can define a reference frame for change resolution. Our future work includes a comparison of queries based on their resolution. Topological relationships [4,6] (e.g. overlap, intersect, etc.) can be applied to establish relationships in the spatiotemporal domain, that can lead to a comparison of different queries on different systems, as long as the hierarchical trees are the same. Furthermore, by comparing the topological relationship of different systems, a global indexing mechanism for spatiotemporal resolution can be created, that will be based on the granularity of each system. With this, specific databases can be accessed, depending on the resolution of the query, both at the spatial and the temporal domain.

Acknowledgements

We wish to gratefully acknowledge NIMA for support of this research under grant number NMA202-98-1-1113. The research of Peggy Agouris is further supported by the National Science Foundation through CAREER grant number IIS-9702233, and the National Aeronautics and Space Administration under grant COE 97-0015.

References

- [1] A. Artale, E. Franconi, N. Guarino and L. Pazzi, "Part-Whole Relations in Object-Centered Systems: An Overview", *Data and Knowledge Engineering (DKE) Journal*, North-Holland, Elsevier, 1996, pp. 347-383.
- [2] C. Bettini, X. Wang and S. Jajodia, "Temporal Semantic Assumptions and Their Use in Databases", *IEEE Transactions on Knowledge and Data Engineering*, To appear.
- [3] C. Bettini, X. Wang and S. Jajodia, "An Architecture for Supporting Interoperability among Temporal Databases" *Temporal Databases*, Dagstuhl, 1997, pp. 36-55.
- [4] H. Bruns and M. Egenhofer, "Similarity of Spatial Scenes", *Spatial Data Handling '96*, Delft, Netherlands, 1996, pp. 4A31-42.
- [5] J. Clifford and A. Croker, "The historical relational data model (HRDM) and algebra based on lifespans", *In Proceedings of the 3rd IEEE International Conference on Data Engineering*, 1987, pp. 528-537.
- [6] M. Egenhofer and R. Franzosa, "On the Equivalency of Topological relations", *Intl Journal of geographical Information Systems*, Vol. 9(2), 1995, pp. 133-152.
- [7] N.W.J. Hazelton, "Extending GIS to include dynamic modeling", *Proceedings of the Third Colloquium of the Spatial Information Research Center*, University of Otago, Dunedin, New Zealand, 1991, pp. 73-82.
- [8] K. Hornsby and M. Egenhofer, "Identity-based change: a foundation for spatio-temporal knowledge representation", *International Journal of Geographical Information Science*.14(3). To appear.
- [9] G. Langran, "Time in Geographic Information Systems", London, Taylor & Francis, 1992.
- [10] A. Montanari, E. Maim, E. Ciapessoni, and E. Ratto, "Dealing with Time Granularity in the Event calculus", *In Proceedings of the International Conference on Fifth Generation Computer Systems 1992*, volume 2, Tokyo, Japan, 1992, pp. 702-712.
- [11] Y. Shahar and C. Cheng, "Model-Based Visualization of Temporal Abstraction", *Fifth International Workshop on Temporal Representation and Reasoning (TIME '98)*, Sanibel Island, Florida, 1998, pp. 11-20.
- [12] J. Szego, "Human Cartography: Mapping the World of Man", Stockholm: Swedish Council for Building Research, 1987.
- [13] V. Tsotras and N. Kangelaris, "The Snapshot Index: An I/O-optimal access method for timeslice queries", *Information Systems* 20(3), 1995, pp. 237-260.
- [14] X. Wang, S. Jajodia, and S. V. Subrahmanian, "Temporal Modules: An Approach Toward Federated Temporal Databases", *SIGMOD Conference*, 1993, pp. 227-236.