Query Languages

IDENTIFICATION OF A DEFINITION FORMALISM FOR A SPATIAL VIEW

C. CLARAMUNT [1], M. MAINGUENAUD [2]

[1] Swiss Federal Institute of Technology,
Rural Engineering Department,
1015 Lausanne, Switzerland
E-mail: Christophe.Claramunt@dgr.epfl.ch
Telefax: 41-21-693-37-39

[2] France Télécom,
Institut National des Télécommunications,
9 rue Charles Fourier, 91011 Evry, France
E-mail: Michel.Mainguenaud@int-evry.fr
Telefax: 33-1-60-76-47-80

ABSTRACT. Today spatial database users need dynamic and flexible query and visualization languages. The objective of this paper is to propose an extension of the classic view concept as found in database models to meet these requirements. Our research relies on a spatial data model and on a visualization query language which could provide a basis for the identification of a spatial view formalism. We will propose and illustrate the spatial view definition as a dynamic tool and as a function of users' points of view.

1. INTRODUCTION

The proposed research attempts to resolve visualization needs by defining a spatial view concept for qualitative and personalized viewing of geographic information in function of its meaning. To reach this end, our approach proposes to extend the view concept, as it is defined for database models, to spatial information characteristics.

A view was first defined in the relational DBMS world as a virtual relation derived from one or several relation (e.g. Ceri 1991, Bertino 1992). An object-oriented view is similarly defined as a virtual class derived by an object-oriented query (Kim 1989) or as a virtual, possibly restructured, subschema graph of the global schema (Tanaka 1988). Object-oriented views allow the expression of external schemas derived from the logical level, while integrating relative dynamics within their definition (operators) (Bertino 1992). Generally, a view allows the representation of data according to a point of view and in function of objectives which are different from those of the schema. Its role may be seen as a flexible and evolutive representation; flexible in that it allows the user to choose the appropriate representation, and evolutive because it contributes to the correct cohabitation of the database schema and applications over time. Views are thus a favored means for dynamic expression of adaptable and evolutive forms of a given data set.

Unfortunately, no view definition has been offered for spatial data. This paper will propose an extension of the view concept to spatial data. We will define a spatial view as the necessary integration, for spatial data, of spatial and visualization operators in a classic view model. We will develop this approach by a preliminary identification of a general model for spatial data.

We begin this study with an overview of visualization query languages and by examining the state of the art to evaluate current proposals related to the spatial view concept. We will attempt to identify the characteristics of a spatial database, a reference framework among the spatial database models will be identified in order to facilitate further research. We will analyze the objectives and mechanisms which lead to the representation of a set of spatial information; identify the semantics of this heterogeneous information and their visualization operations which are the constituent elements of a spatial view. Interpretation factors for spatial views will be addressed, which will lead us to describe the diversity of information components to be represented and their interrelations, the elaboration of a graphical interface and the presentation constraints which interfere with spatial information visualization processes. Having identified these elements, we may then propose a language for spatial view definition.

2. RELATED WORK

The deficiencies of the SQL query language for spatial applications are today well known (Egenhofer 1992). One of the weaknesses of this language is the lack of visualization functions for spatial information (Frank 1982, Ingram 1987, Raper 1991, Egenhofer 1991). Graphic display of geographic features involves two issues: specifying the information to be displayed and the format in which to display it (Ingram 1987). Some attempts have been made to introduce requirements for graphical presentation in a spatial SQL (Frank 1982, Egenhofer 1991). Egenhofer proposes a Graphic Presentation Language which contains commands to direct, among others, the display of objects, spatial context, the query window, map scale and, taken as a whole, how to display query results.

The concept of a spatial view, or the idea associated with it, is indirectly addressed in graphical applications and hardly addressed in the area of spatial database research. Beard identifies the applicational, structural and graphical constraints which are associated with generalization processes and which underscore the need for a generalized view of a spatial database (Beard 1991). For graphical applications, the need for flexible visualizations to manipulate graphical data has been identified. Egenhofer points out that the choice of visualization must be controllable so that users may choose the visualization style which is appropriate for their application needs; this dynamic flexibility provides an essential advantage with respect to traditional maps (Egenhofer 1991). The term multiple view is proposed to designate a graphical layer for controling the visualization of arbitrary objects (Haarslev 1990); the view notion is also used to designate the semantic regrouping of multiple numerical maps (Tanaka 1988). Abel identifies the interest of a form of spatial view for geographic applications (Abel 1992). He notes that even if modeling tools such as the entity-relationship formalism, which allows the integration of different points of view, are useful, they cannot resolve all the problems of a large organization which represents a dynamic context concerning data representation, and notably conflicts between highly varied points of view. All these research efforts have indirectly addressed the spatial view concept. We will pursue similar objectives but with a database approach which will lead to a more formal definition.

3. A SPATIAL MODEL FRAMEWORK

3.1. Spatial data semantic

We will develop the spatial view proposal as an extension of the relational view, considering that the relational model is well known and provides a vehicle to present our ideas. A semantic approach to relations and attribute specifications may help the management of the results of spatial visualization operations. The quality of a spatial data model successively conditions the expressiveness of query and visualization languages. In particular, the characteristics of a spatial entity require the definition of relation and attribute consistency:

- *spatial overlap* of two instances of a same spatial entity (e.g. two overlapping flood zones) equivalent or not to a spatial partition (Mainguenaud 1993);

- *classic attribute domains* (e.g. integer, real, character) and *value modes* (ordinal, nominal, cardinal) defined by Stevens (1946). A spatial entity attribute has additional validity conditions. The *granule* defines the attribute validity for the *integrality* or for the *sub-set* of the spatial entity (e.g. an average altitude attribute is defined for the integrality and not for a subset of a spatial entity). The *topology* defines the topological validity (*interior*, *boundary*, *global*) of an attribute (e.g. a crop attribute is defined for the interior of a parcel).

These conditions may be specified as in the following example :

> **Create Table non_overlap** City
> (<u>Name</u>: string, (sub-set, global), primary key;
> Population: string, (integral, interior))
> Spatial-Representation: Spatial-Rep)

In this example, a city is a non overlapping spatial entity (one city at one geographic location). The city name is available for the globality (interior and boundary) and is still valid for a subset of the spatial entity. Conversely, the city population is available for the integrality (not for a subset) and for the interior of the spatial entity. We specify the spatial representation of the city by a special type (Spatial-Rep) which will be defined in the next sub-section. A similar framework may be used for query and visualization operators to propose a semantic control of the resulting alphanumeric data.

3.2. A complex object approach for spatial information description

We propose to define the spatial representation of an entity by a *"complex object"* definition (Adiba 1987). A complex object allows the extension of basic data types (e.g. integer, real, character). A complex object attribute may be decomposed into a set of component attributes. Attribute decomposition can be iterated as long as needed to ensure a complete description of an object (Parent 1989). This structure has some similarities with the encapsulation mechanism defined by the abstract data type. The latter is at the basis of the encapsulation mechanism defined within object-oriented databases (Stonebraker 1983). Complex objects allow the representation of static properties of spatial entities. We will use complex object notation, presented by the following notation, for the definition of the spatial representation of an entity:

> An atomic type is noted A: T where T is a type (e.g. integer, real, spatial domain). Types are recursively defined as follows:
>
> if T1, ..., Tn are distinct types and A an attribute then A: [A1: T1, ..., An:Tn] is a tuple of name A;
> if T1 is a type and A an attribute then A: {T1} is a set type of name A;
> if T1 is a type and A an attribute then A: <A1: T1, ..., An: Tn> is a list type of name A;
> if T1 is a type and A an attribute then A: (value(T1), ..., value(T1)) is an enumerate type of name A.

We characterize the elements of each spatial entity by a *spatial representation* (Spatial-Rep). Each spatial representation groups the cartographic primitives and their properties for each spatial entity :

- *cartographic primitive* (Cartographic-Prim): cartographic primitive defined on a spatial domain; it allows the spatial expression of a spatial entity. Several authors propose a spatial domain definition with different semantic complexities (e.g. Egenhofer 1991, Worboys 1993). We will consider, for our demonstration, a spatial domain specified by a complex object called geom without considering its structure;

- *observation scale* (Observation-Scale): property which specifies the spatial referencing scheme or the perception level of the phenomenon (e.g. global, regional, local) within which the cartographic primitive is defined (adapted from Golay 1992);

- *validity scale* (Validity-Scale): property which defines the scale validity of the cartographic primitive;

To complete these definitions, the *significant scale* (Significant-Scale) specifies and references the observation scale for which the spatial entity attributes are semantically interpreted.

The complex object of the spatial representation is then defined by:

Spatial-Rep:
[
<
[Cartographic-Prim: geom;
Observation-Scale: string;
Validity-Scale:[Limit-Min: real; Scale-Capture: real; Limit-Max: real]

]
>
Significant-Scale: integer
]

If the complex object of a spatial representation has at least two cartographic primitives, then the corresponding spatial entity has a multi-valued spatial representation or, in other words, a multiple spatial representation. A spatial entity may have many cartographic primitives for a considered scale if their validity scale intervals overlap. In this case, the spatial entity has multiple interpretations for a same scale (e.g. two different user views for the same spatial entity).

For each spatial entity, the spatial representation defines and manages the cartographic primitives and their properties. This specification of the properties at the spatial entity level allows data representation in a heterogeneous application situation (i.e. many users and semantics for same spatial entities).

3.3. Real or virtual representation

A spatial entity visualization may be realized from two types of cartographic primitive (adapted from Sacks-Davis 1987):

- *"real"* cartographic primitives which correspond to a geometric space reality and are significant for spatial entity attributes (e.g. a city polygon defining an urban area limit which is geometrically accurate);

- *"virtual"* cartographic primitives related to a spatial entity but not geometrically significant (e.g. a city polygon represented by a schematic limit which is not geometrically accurate).

A spatial operation is generally efficient if it is performed from real cartographic primitives of spatial entities. For instance, a spatial intersection operation has to be computed from a real cartographic primitive and not from a virtual representation, a function of a real cartographic primitive. Conversely, virtual representations do not guarantee the quality of the result of a spatial operation. However, a virtual representation may be used for approximate spatial operations in the case of complex or large databases, where it provides a preliminary evaluation result. Operator validity rules are inferred from real or virtual types of cartographic primitives and thus independently of spatial entity semantics. Therefore, the complex object definition of a spatial entity may be extended by a the definition of a mode, which specifies the real or the virtual type of a cartographic primitive:

Spatial-Rep:
[
<
[Cartographic-Prim: geom;
Observation-Scale: string;
Validity-Scale: [Limit-Min: real; Scale-Capture: real; Limit-Max: real]

Mode: (real, virtual)
]
>
Significant-Scale: integer
]

3.4. Visualization variables and functions

We have developed a static representation model for spatial information. This model allows the identification of the entities implicated in an SQL query (from clause) and a relative evaluation of operation applications (where clause). We will now extend this reference model by a description of the visualization parameters and functions which will lead to a management of the graphic presentation of query results (within the select clause).

Visualization processes are a function of spatial representation and particularly of the different cartographic primitive levels. The spatial representation definition permits the encapsulation of identified and stable visualization parameters. As we consider the spatial view as a query interface and not as a cartographic tool, this definition level is sufficient. Therefore, the system is told how the values (spatial entities in this case) are to be displayed (Osborn 1986). These parameter visualizations are classified and referred to as Bertin variables (Bertin 1983). These variables allow the integration of visualization parameters in a spatial query language (Egenhofer 1991). Combination of these variables provides a visualization operation with sequential parameters such as color, value (gray degree or intensity), pattern (graphic symbol), size (e.g. of a graphic symbol), texture (e.g. pattern resolution) and orientation (e.g. of a graphic symbol). These Bertin variables are applied to cartographic primitives (e.g. a polygon may have an exterior visualization color and an interior symbol pattern). For our study, we will define the Bertin variables as a complex visualization object (Bertin-Variable) which is integrated in the spatial representation:

Spatial-Rep:
[
<
[Cartographic-Prim: geom;
Observation-Scale: string;
Validity-Scale: [Limit-Inf: real; Scale-Capture: real; Limit-Sup: real]
Mode: (real, virtual);
Visualization: Bertin-Variable
]
>
Significant-Scale: integer
]

In order to extend visualization capabilities, visualization functions complete the graphic presentation language (Ingram 1987). Many authors propose visualization functions for spatial information which may be logically inserted in the select clause of a query:

- generalization operators (Brassel 1987, Beard 1991) such as simplification, smoothing, aggregation, amalgamation, merging, collapse, refinement, exaggeration, enhancement and displacement (McMaster 1991);

- thematic classification operators in accordance with cartographic methods (Huang 1993);

- display and position operators for textual information managed by specialized algorithms (Freeman 1987);

- context operators such as legend display (Egenhofer 1991).

This overview of visualization functions aims to illustrate our purpose. This shows the interest of visualization functions within a spatial query language. However, visualization functions for query languages have to be limited to single operations so as to avoid complex manipulations which may change the nature of query objectives. A visualization process creates new entities defined as the projection of spatial entities by query and visualization operations or, in other words, as the visualization domain of the visualization operation. This visualization process analysis may be continued by an evaluation of the management of constraint presentations defined as the association of windowing parameters and human visualization capacity (Robinson 1984). This allows the user to interact with the graphic interface and improve the quality level of the application (Voisard 1991). The specification within this model of the spatial and visualization properties gives the user an aid-oriented approach which facilitates visualization operations compared to a free solution where the user has to choose all visualization parameters.

4. A SPATIAL VIEW CONCEPT PROPOSAL

We have defined a spatial representation model and analyzed parameter and function visualizations. We propose an approach to spatial view definition from these static and dynamic elements. A spatial view is designed as the application of spatial and non spatial queries (O) combined with spatial visualizations (OV) of one or many spatial and non spatial relations. Together, the query and visualization operations form a manipulation language. A collection (C) is defined as a set of spatial and non spatial instances used within a spatial view creation. This definition is related to the horizontal view proposed by Hegner (Hegner 1991) in that the spatial view is designed as a dynamic grouping of different complementary entities.

Each triplet [{C}, {O}, {VO}] defines a spatial view atom. A spatial view atom is a member of at least one spatial view. A spatial view atom is a visualized spatial relation and is derived from a collection set. A spatial view atom may be defined as the application of spatial and non spatial operations combined with visualization operations on a collection set. Spatial view atoms are closed under their definition. A basic spatial relation is defined as a spatial view atom with empty spatial and non spatial operations and an empty set of visualization operations. A spatial view groups together spatial view atoms. It is defined as a relation which is both spatial and non spatial and may be composed of relations and derived relations. Unlike the relational view definition, a spatial view is not a relation but a set of relations.

Each spatial or non spatial entity projected in the realization process of the spatial view will be a member of a spatial view atom decomposed into (1) a collection set of spatial and non spatial entities (2) a query and visualization set and returning (3) a visualization entity set. Figure 1 shows a comparison of view and spatial view realization processes:

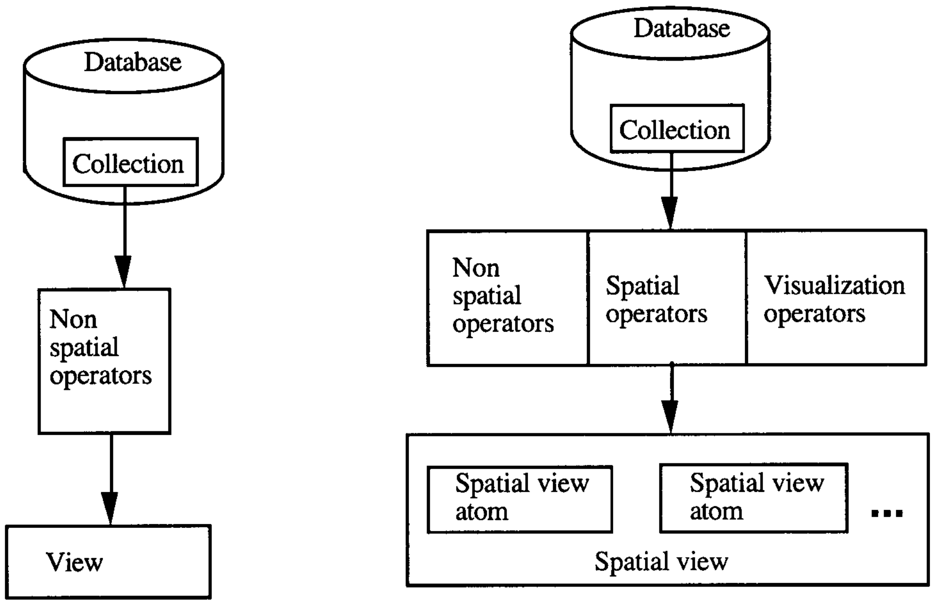

Figure 1: View and spatial view realization

A same spatial or non spatial entity may be member of different collections, different spatial view atoms and many spatial view realization processes. To each virtual or real visualization entity, correspond one-to-many entity references from a collection used by the spatial view process.

Human perception of graphic information is limited by visual acuteness and size limitations of a graphic interface, especially in the case of complex queries. Moreover, when the scale is reduced, linear entities are overestimated with respect to polygonal entities, or visualized classes may be reduced and the visual density may increase. These progressive and interrelated phenomena have to be controlled to ensure the quality of visualization results. The different visualization collections are semantically and cartographically interdependent. Visualization entities may be ordered by relative semantic importance and relative query roles. This can be seen as an answer to both application objectives and the progressive saturation of a display space query. The respective contribution of each spatial view atom may be consequently analyzed depending on the query semantics with respect to user and application objectives. We can distinguish three level of collection importance for spatial view atoms:

- collections directly involved in the query result (essential);

- collections used within the query (important);

- collections used as background maps (useful) (Ingram 1987).

We will represent, within the spatial view definition, this classification by an order attribute which manages the relative importance of the different spatial view atoms (essential, important, useful). Each spatial view will be additionally defined by a visualization scale (bounded and referenced) which will be specified by a visualization scale attribute. Therefore a spatial view is defined as:

```
Spatial-View:
[
Name;
Visualization-Scale: {[Limit-min, Scale-Display, Limit-Max]};
Order: [essential: {Spatial-View-Atom};
        important: {Spatial-View-Atom};
        useful: {Spatial-View-Atom}]
]
```

To explain this concept, we propose a basic example of a spatial view design. In particular we will try to show the relative importance of each spatial view atom. A first classification of spatial view atoms displays the spatial representation of Portuguese cities which have more than 100 000 inhabitants (essential order):

C1, a spatial entity collection:
C1: {City, Country}

Create Spatial-View-Atom Classification
select City.Spatial-Rep
display text City.Name
from City, Country
where City.Population > 100 000
and Country.Name = 'Portugal'
and City.Spatial-Rep **inside** Country.Spatial-Rep

Two additional spatial view atoms, Country (important order) and River (useful order), complete the first spatial view atom and provide a complement for the spatial view:

C2 a spatial entity collection:
C2: {Country}
C3, a spatial entity collection:
C3: {River}

Create Spatial-View-Atom Country
select Country.Spatial-Rep
from Country
where Country.Name = 'Portugal'

Create Spatial-View-Atom River
select River.Spatial-Rep
from River

where River.Length > 300
and Country.Name = 'Portugal'
and River.Spatial-Rep **intersect** Country.Spatial-Rep

The explicit designation of the spatial view atoms identifies the different related queries of the spatial view. The spatial view example is finally defined as :

Create Spatial View
[
Name: example;
Visualization-Scale: [1/20 000, 1/10 000, 1/5 000]
Order: essential: [Classification]
important: [Country]
useful: [River]
]

This spatial view example shows a visualization composition which groups together the related spatial view atoms. The query meaning is explained by the Classification spatial view atom (essential order). The query result is dependent on the Country spatial view atom (important order) and the final visualization result is graphically completed by the River spatial view atom (useful order). This spatial view gives, among others, an idea of Bertin variable utility in the query visualization process for the different spatial entities visualized (pattern, size, symbol).

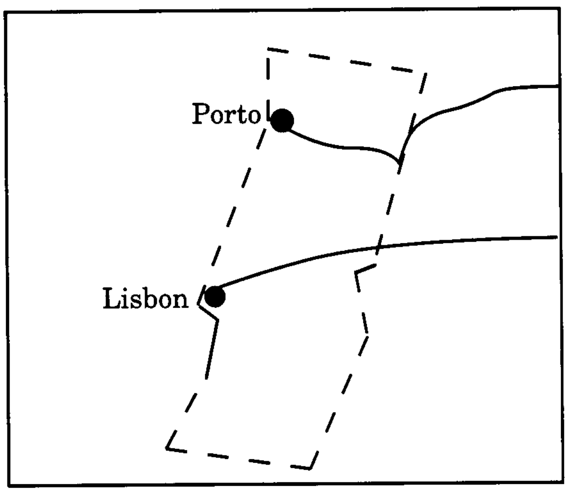

Figure 2: spatial view example

The semantic order defined within the complex attribute definition may be completed, at the interface level, by a logical order which controls the display priorities (e.g. polygons are generally displayed before lines).

5. CONCLUSION

The spatial view concept gives users flexibility to compose a graphic database view adapted to spatial application requirements. The spatial view may be suited for spatial database queries or even updates. From an external point of view, the spatial view concept allows different, independent external interpretations of database schemas. This provides a support for schema evolutions and a partial solution for application portability with respect to user needs.

We have proposed and justified a spatial view concept as an extension of the view developed within classic database models. We have identified static and dynamic constituents of spatial entities by a complex object specification. The spatial view is realized as a sequence of selection and visualization operations. This process is structured by spatial view atoms which are the incremental, basic elements of a spatial view definition. Spatial view atoms provide an ordered relation to characterize the relative importance of the visualized data.

At present we have several directions for our future research. The spatial view concept has to be defined exhaustively and formally. In particular, spatial view properties and manipulation capabilities are interesting development issues. Materialized or derived views provide two different implementation solutions which have to be analyzed with respect to application conditions for the spatial view case. For instance, a materialized spatial view has to be maintained through database evolution which leads to the study of recomposition rules and mechanisms. Integration of the spatial view concept towards a semantic network for hypertext navigation may give a user-oriented application interface.

Many research activities are related to an improvement of the spatial view. New extensions for the classic view concept, current developments in the area of visual query languages using metaphors or near natural languages are beneficial factors. Additional developments in constraint presentation issues are also a related development to this work.

ACKNOWLEDGEMENTS

We would like to thank Christine Parent and Marius Thériault for their helpful comments on the subject of this paper. We are also grateful to Laura Vidale for her attentive reading of preliminary versions.

REFERENCES

Abel, D. J., Yap, S. K., Ackland, R. G., Cameron, M. A., Smith, D. F. and Walker, F. G. (1992), Environmental Decision Support System Project: An Exploration of Alternative Architectures for Geographic Information Systems, in *International Journal of GIS,* vol. 6, n° 3, pp. 193-204

Adiba, M. E. (1987), Modeling Complex Objects for Multimedia Databases, in *Entity-Relationship Approach : Ten Years of Experience in Information Modeling,* eds. Spaccapietra, S., pp. 89-117

Beard, K. (1991), Constraints on Rule Formation, in *Map Generalization,* eds. Buttenfield, B. P. and McMaster, R. B., pp. 121-135

Bertin, J. (1983), *Semiology of Graphics,* University of Wisconsin Press

Bertino, E. (1992), A View Mechanism for Object-Oriented Databases, in *Proc. of Advances in Database Technology EDBT '92*, Vienna, March 23-27, pp. 136-151

Brassel, K. E. and Weibel, R. (1987), A Review and Conceptual Framework of Automated Map Generalization, in *International Journal of GIS*, vol. 2, n° 3, pp. 229-244

Ceri, S. and Widom, J., (1991), Deriving Production Rules for Incremental View Maintenance, in *Proc.* of *17th Int. Conf. on Very Large Data Bases*, Barcelona, September 3-6, pp. 577-589

Egenhofer, M. J. (1991), Extending SQL for Graphic Display, in *Cartography and GIS*, vol. 18, no. 4, pp. 230-245

Egenhofer, M. J. (1992), Why not SQL, in *Int. J. of GIS,* vol. 6, no. 2, pp. 71-85

Frank, A. U. (1982), MAPQUERY: Data Base Query Language for Retrieval of Geometric data and their Graphical Presentation, in *Computer Graphics*, vol. 16, n° 3, pp. 199-207

Freeman, H. and Ahn, J. (1987), On the Problem of Placing Names in a Geographical Map, in *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 1, n° 1, pp. 121-140

Golay, F. (1992), *Modélisation des Systèmes d'Information à Référence Spatiale et de leurs Domaines d'Utilisation Spécialisés; Aspects Méthodologiques, Organisationnels et Technologiques*, PhD Dissertation, EPFL, Lausanne

Haarslev, V. and Möller, R. (1990), A Declarative Formalism for Specifying Graphical Layout, in *Proc.* of *IEEE Workshop on Visual Languages*, Skokie, Illinois, October 4-6, pp. 54-59

Hegner, S. J. (1991), Pairwise-Definable Subdirect Decompositions of General Database Schemata, in *Proc.* of *Symp. on Mathematical Fundamentals of Database Systems*, Rostock

Huang, Z. (1993), Designing a Spatial Query Language for Spatial Data Query and Analysis, in *GIS Technology and Applications*, eds. Lu, H. and Ooi, B. C., Singapore, June 21-22, pp. 209-223

Ingram, K. J. and Phillips, W. W. (1987), Geographic Information Processing Using a SQL-Based Query Language, in *Proc.* of *Auto-Carto Conf. 8*, Baltimore, March-April, pp. 326-335

Kim, W. (1989), A model of Queries in Object-Oriented Databases, in *Proc.* of *International Conf. on Very Large Data Bases*, Amsterdam, August, pp. 423-432

McMaster, R. and Mark, D. M. (1991), The Design of a Graphical Interface for Knowledge Acquisition in Cartographic Generalization, in *Proc.* of *GIS/LIS'91*, Atlanta, October 28 - November 1, pp. 311-320

Mainguenaud, M. (1993), The Results of Geographical Information System Queries, in *Proc.* of *IEEE/CS Visual Languages '93*, Bergen, Norway, August 25-27, pp. 362-364

Osborn, S.L. and Heaven, T. E. (1986), The Design of a Relational Database System with Abstract Data Types for Domains, in *Proc.* of *ACM Transactions on Database Systems*, vol. 11, n° 3, pp. 357-373

Parent, C. and Spaccapietra, S. (1989), Complex Objects Modeling: An Entity-Relationship Approach, in *Nested Relations and Complex Objects in Databases*, Springer-Verlag, n° 361, pp. 272-296

Raper, J. F. and Bundock, M. S. (1991), UGIX: A Layer Based Model for a GIS Interface, in *Cognitive and Linguistic Aspects of Geographic Space*, eds. Mark, D. M. and Frank, A. U., pp. 449-475

Robinson, A. H., Sale, R. D. and Morrison, J. L. (1984), *Elements of Cartography*, ed. John Wiley, New York

Sacks-Davis, R., McDonell, K. J. and Ooi, B. C. (1987), *GEOQL - A Query Language for Geographic Information Systems*, Internal Report no. 87/2, Dept of Computer Science, Royal Melbourne Institute of Technology, Australia

Stevens, S., (1946), On the Theory of Scales of Measurement, in *Sciences Magazine*, 103, pp. 677-680

Stonebraker, M., Rubenstein, B. and Guttman, A. (1983), Application of Abstract Data Types and Abstract Indices to CAD Data Bases, in *Proc.* of *Engineering Design Applications of ACM IEEE Database Week*, Los Angeles, pp. 107-113

Tanaka, M and Ichikawa, T. (1988), A Visual User Interface for Map Information Retrieval Based on Semantic Significance, in *IEEE Transaction on Software Engineering*, vol. 14, n° 6, pp. 666-670

Voisard, A. (1991), Towards a Toolbox for Geographical User Interfaces, in *Advances in Spatial Databases*, eds. Günther, O. and Schek, H.-J., Zurich, August 28-30, pp. 75-98

Worboys, M. F. and Bofakos, P. (1993), A Canonical Model of Areal Spatial Objects, in *Advances in Spatial Databases*, eds. Abel, D. J. and Ooi, B. C., Singapore, June 23-25, pp. 36-52