

Towards Cooperativeness in Geographic Databases

Andrea S. Hemerly^{1,2}, Antonio L. Furtado² and Marco A. Casanova¹

¹ Centro Científico Rio, IBM Brasil, Av. Presidente Vargas, 824/844,
20071 Rio de Janeiro RJ - Brasil

² Departamento de Informática, PUC-RJ, R. Marquês de São Vicente, 225,
22453 Rio de Janeiro RJ - Brasil

Abstract. This work addresses cooperative environments for geographic databases that help users achieve their goals. The nature of geographic data, as well as the requirements for an interface to geographic databases, are first discussed. Then, the components of a rule-based cooperative environment for conventional databases are described. The core of the paper analyzes cooperativeness with respect to geographic data, separately considering query processing and the graphical display of query results.

1 Introduction

We analyze in this paper the construction of cooperative environments that help users access geographic databases. We concentrate on the novel opportunities for cooperativeness that spatial data types and their operators provide, separating the problems of query processing from those related to the graphical display of query results.

We phrase our discussion in terms of the rule-driven cooperative environment we proposed in [3]. The environment comes equipped with a set of generic rules to which the interface designer can add new application dependent rules. In either case, each rule is assigned a type that indicates when it should be invoked during the processing of a query. The central component of the environment is an algorithm that tries to apply to the query all rules that modify requests, generating a new query. Then, it executes the modified query against the database. After a successful execution, it can perform new actions as the result of the cumulative processing of the rules designed to act when queries succeed. In case of a failed execution, the algorithm tries to apply, one at a time, all rules that generate alternative queries, until one succeeds. If all alternatives fail, it also cumulatively tries rules that compensate for failures. Finally, the algorithm uses rules designed to edit query results.

Cooperative query processing in general purpose database systems is explored, for example, in [1]. But, to the best of our knowledge, there are no published papers that explore geographic data and spatial operators strictly to obtain cooperative behavior. For a fuller account of the concepts described in this paper, the reader is referred to [4].

2 Cooperative Query Formulation over Geographic Databases

The automatic correction of ill-formulated queries perhaps provides the most natural examples of cooperative query processing. But this approach can also be extended to providing alternatives or explanations to a (failed) query and to complementing a query, in the sense of retrieving more information than the user explicitly asked for. In this section we show how the cooperative approach may help in the treatment of spatial queries with ambiguous semantics, especially those formulated with the help of linguistic spatial operators whose semantics depends on context information.

All examples in this section follow a simple pattern: the semantics of an operator *op* is defined with the help of parameters whose values are fixed either in the application domain model, or in the user domain model. Moreover, the definition of *op* is expressed as a rule that reduces *op* to other known operators. This approach is naturally adequate only when the semantics of *op* is ambiguous in so far as two users may have different parameter sets and, hence, may get different answers for the same query.

Let us first consider the imprecise metric operator, *near*, considered fundamental for the definition of spatial relationships. When a user formulates a spatial query, the notion of nearness is naturally influenced by the context of the query [5]. For example, cities in a state map may appear near if they are less than one hundred kilometers apart, depending on the scale, whereas streets may appear near in a city map only if they are less than one kilometer apart. Robinson [6] pointed out that, besides depending on the context of the query, the notion of nearness may also depend on the user.

Thus, we define that *P* is *near Q* iff the distance from *P* to *Q* is less than $n * length_unit$, where *length_unit* is a context dependent scalar parameter, and *n* is a scalar parameter, that can be either unique for the application and thus defined in domain model, or specific for each class of users and therefore defined in the user models. This translational definition of *near* is easily captured by a rule to be applied before query execution.

For example, to process the query 'Retrieve all hospitals *near* school S1', the system will first apply the rule that defines *near*, which results in the final query 'Retrieve all hospitals whose distance to school S1 is less than $n * length_unit$ '.

We next analyze the imprecise topological operator *between*. For simplicity, we assume in what follows that all objects involved are points. When $m > 2$, we define that the point *P* is *between* the points O_1, \dots, O_m iff *P* lies in the interior of the polygon created by connecting O_1, \dots, O_m , in this order. When $m = 2$, we may give a better definition by considering that *P* is *between* the points O_1 and O_2 iff *P* lies in the ellipsis such that one of the axis is the line from O_1 to O_2 and the other axis is $n * length_unit$ long. Indeed, this definition tries to capture the imprecise idea that a point between two others may not be exactly aligned with them.

We now turn to the spatial order operators *before* and *after*, whose semantics we define with the help of the concept of observation point. For example, a school

may be before or after a hospital with respect to a given observation point. The system's interface may represent an observation point by a special symbol and provide specific commands to move it around the user's visual space. It may also be interesting to define multiple named points of observation in the same visual space. A possible treatment for *before* and *after* would be to reduce them to *between* as follows: *P* is *before* *R* from *O* iff *P* is *between* *R* and *O*, and *P* is *after* *R* from *O* iff *R* is *between* *O* and *P*, where *O* is the observation point.

3 Cooperative Presentation of Geographic Data

In this section, we focus on the cooperative presentation of geographic data. We first observe that the design of an interface for geographic databases must carefully determine a context for the presentation of the query results. To help address this question, we define the concept of a *visual space* as consisting of a *background*, which is a collection of geo-referenced geometries or pictorial data, a *length unit* and an *observation point*. The last two concepts were explained in the previous section, whereas the role of the concept of background is to help users interpret the spatial position of the geometries retrieved by queries.

We assume in what follows that the interface will exhibit the geometries of the query results superposed with those of the background. Hence, the geometries of the background must be distinguished from those explicitly requested, and the background must not be dense to avoid shifting the focus of attention from the query results. However, instead of always superposing the query results onto a background, the interface may also exhibit the query results and the background on separate windows, or present a selection menu to the user and let him decide which background he wants, if any. For example, a reasonable visual space for queries about a city might consist of just the major avenues in the city map (and not all streets) as the background, one kilometer as the length unit and a given point in the city map, as an observation point.

We assume that each geographic database includes a collection of predefined visual spaces, some of which may be specific to a given class of users. We allow visual spaces to be constructed out of virtual objects, defined over the other objects stored in the database. For example, in the above visual space, we may define the background as a (database) view over the streets of the city map, which implies that any new major avenue added to the city map will be included in the background automatically.

The interface may follow several strategies to automatically select a visual space from those predefined. For example, it may choose the visual space whose background contains all geometries of the query results, or one whose background covers the geographic area of all the geometries of the query results. If these criteria fail, the interface may try to partition the query results by geographic area before re-applying them. On the other hand, if more than one visual space meets these criteria, the interface may resort to heuristics to guide the choice, such as to select the visual space whose background has the largest scale.

The above suggestions for the automatic selection of visual spaces are sources of rules that capture cooperative behavior. For example, consider the query 'Show all schools in district D'. Assume that the database in question contains a family of visual spaces, one for each district, whose backgrounds contain the perimeter and the major avenues of the districts. The strict result of the above query naturally contains just the geometries of the schools in district D, which is not very informative. The system will then exhibit these geometries along with the visual space for district D, which is clearly suitable for the query, since its qualification indicates that all retrieved geometries belong to the geographic area of district D.

The full paper [4] also discusses the use of rules to solve presentation ambiguities and conflicts and to edit answers.

4 Conclusions and Directions for Future Research

We explored some of the new opportunities, not present in the context of conventional databases, that the spatial data types and their operators provide for cooperative query processing, including the presentation of query results. We showed, among other features, that a cooperative environment may adequately capture the semantics of spatial linguistic operators, as well as provide visual contexts for exhibiting query results.

References

1. F. Cuppens and R. Demolombe. Cooperative answering: a methodology to provide intelligent access to databases. In L. Kerschberg, editor, *2nd International Conference on Expert Database Systems*, pages 621–643. Benjamin/Cummings, 1989.
2. Max J. Egenhofer. *Spatial Query Languages*. PhD thesis, University of Maine, Orono, ME, May 1989.
3. A.S. Hemerly, M.A. Casanova, and A.L. Furtado. Cooperative behaviour through request modification. In *10th International Conference on the Entity-Relationship Approach*, pages 607–621, San Mateo, CA, USA, 1991.
4. A.S. Hemerly, A.L. Furtado and M.A. Casanova, Towards Cooperativeness in Geographic Databases. Technical Report CCR-XXX, Rio Scientific Center, IBM Brasil, Rio de Janeiro, Brasil (Jan. 1993).
5. D.M. Mark and A.U. Frank. Concepts of space and spatial language. In *9th International Symposium on Computer-Assisted Cartography*, pages 538–556, Baltimore, MD, USA, 1989.
6. V.B. Robinson, M. Blaze, and D. Thongs. Representation and acquisition of a natural language relation for spatial information retrieval. In *2nd International Symposium on Spatial Data Handling*, pages 427–487, Seattle, Washington, USA, 1986.