The Results of Geographical Information System Queries

M. Mainguenaud

FRANCE-TELECOM - Institut National des Télécommunications
9 rue Charles FOURIER F91011 EVRY - FRANCE
+ 33 1 60 76 47 14 + 33 1 60 76 47 80 (fax)
Email: Michel.Mainguenaud@int-evry.fr

Abstract :

In this paper,we study the definition of the results for GIS queries. They are based on a graphical representation. Such a representation may introduce some ambiguities depending on the class of queries. We study two different ambiguities due to the aggregate functions and the notion of homogeneity.

I. Introduction

Designing a user friendly query language is one of the main issues to deal with when building spatial information systems. In this paper, we focus on the management of Geographical Information Systems (GIS) queries. The definition of a query with a Query-By-Example philosophy may introduce some ambiguities [2]. The 2D representation of the results introduces two major problems studied here: the management of the aggregate functions and the homogeneity of the results.

II. The definition of GIS results

II.1. Some queries

Let us define a toy database as a sub-set of the database designed to evaluate the expressive power of the GIS query language presented in [1] (i.e., logical benchmark). Figure 1 presents the set of relations defining the toy database.

Town (<u>#town</u>,name,population,economical_activity, cost_hotel, spatial_representation)
Transport (<u>#transport</u>, name, transport_type, price, spatial_representation)
Network (<u>#line</u>, <u>origin,</u> <u>destination,</u> <u>departure_h,</u> <u>arrival_h</u>, #transport)

Figure 1 - The toy database

The relation Network corresponds to the first level of abstraction of the relation Transport. It allows the user to see the relation Transport as a logical graph. We adopt the three-step definition process of a GIS query defined in [3]: (1) the actual user query specifying the retrieval of data to be displayed; (2) additional queries, called display queries, necessary to separate query results into more detailed sets, each to be displayed in individual format, and (3) the actual display description specifying how to render the data. The large number of details about the cartographic display makes the query formulation complex and tedious. Therefore we do not consider steps (2) and (3) in this paper.
Let Q1 be the following query: "Which roads have an intersection with an urban part such that the length is less than 1 km long." Let Q2 be the following query: "Which are the air-routes from Paris to the French towns such as the economical activity is tourism and such that the cost of transportation is less than 1,000 units."

II.2. The basic components

A GIS query requires spatial visualization of the results. Two components are defined: the type of the results and the display mode.
The type of the results is based on the basic objects defining the results. Two classes can be distinguished. The results can be defined with the basic objects involved in the query (i.e., the town and the road in the query Q1). This class, named Basic-result-oriented type, corresponds to the extension of the SQL query language with predicates as a spatial data base interface query language. The results can be defined with the return of the spatial operators and the objects involved in the query. This class, named Total-oriented type, corresponds to the extension of the SQL query language with predicates and operators.
The display mode is based on the structurating of the result. This notion corresponds to the philosophy of defining the result. Two classes can be distinguished. The results can be defined as the union of the objects (basic objects or basic objects and the results of the spatial operators) defining the results (i.e., simultaneously displayed). This class, named Union of elementary results, corresponds to the set-oriented approach (like the relational data model does). The results can be defined as the presentation element by element of the elementary results. This class, named Elementary-result-at-a-time, corresponds to the record-at-a-time approach (like a classical programming language does).
Generally, the pair (Basic-result-oriented type, Union of elementary results) is the only available policy. Nevertheless, it may lead to wrong results or ambiguities. An alternative way is the definition of GIS results as a combination of these two components.

II.3. The possible management

In the context of a "classical" application (i.e., payroll), the definition of the result is clearly defined. The main difference here is the introduction of a "background" screen to localize the spatial representation of the results (i.e., the display of the context) and the introduction of new operators (i.e., spatial operators such as for example intersection, inclusion). To manage such results, we define three parts: the static part, the dynamic part and the representation part.
The static part corresponds to the definition of the "objects" involved in the results. These objects rely on the philosophy of the SQL extensions (i.e., with predicates, with predicates and operators, with composition of operators). Depending on the semantics of the operators, the data model associated to the "objects" involved in the result has to be defined. The management of the static part is DBMS data model dependent and does not introduce any ambiguity.
The dynamic part corresponds to the transformation of an analytical query to a result-based query. Once the results of a query are displayed, the user may ask for some information about the visualized data. The query may concern the data model of a component of the result (i.e., available information of an object that has not been displayed to simplify the presentation such that the population of a town). It is therefore immediately satisfied since the information is available. The demand may have been previously defined (i.e., the notion of hyper-media [1]). It is therefore equivalent to a new and (pre)defined query. The demand may involve a graphical object that does not belong to the results but belongs to the "background" screen. It is therefore equivalent to a point location operator. The management of the dynamic part does not introduce any ambiguity.
The representation part introduces some ambiguities in the result management process depending on the components of the results. These ambiguities are detailed in the following sections.

III. The aggregate functions

An aggregate function is defined as the inference of information from the data set or from the result of an operator (i.e., the sum of the transportation costs). To present the ambiguities due to the aggregate functions, we use the classical decomposition of GIS queries into network-oriented queries and thematic-oriented queries.

III.1. Network-oriented queries

Network operators (i.e., the path operator) imply a Total-oriented type since the user only defines the starting point and the end point. The standard choice of a pair (Basic-result-oriented type, Union of elementary results) is therefore impossible. The pair (Total-oriented type, Union of elementary results) leads to a wrong result as soon as the query involves an aggregate function (i.e., query Q2). The definition of the result as the union of several paths (i.e., several graphs) does not allow a distinction between the paths respecting the aggregate function and the paths who do not (out of common nodes and/or edges of two paths). Two solutions are possible: (1) The display mode must follow an Elementary-result-at-a-time philosophy or (2) the pairwise intersection must be applied to detect such a configuration. Unfortunately such configurations are very frequent and the pairwise intersection operator is a very costly operator.

III.2. Thematic-oriented queries

Figure 2 presents a toy data set. The road labels correspond to the length of the intersection with a town. The standard choice of the pair (Basic-result-oriented type, Union of elementary results) leads to some wrong answers. Figure 3 presents the result of query Q1. The result is defined by a set of pair (town, road) since the spatial representation of the intersection between the towns and the roads is not available.

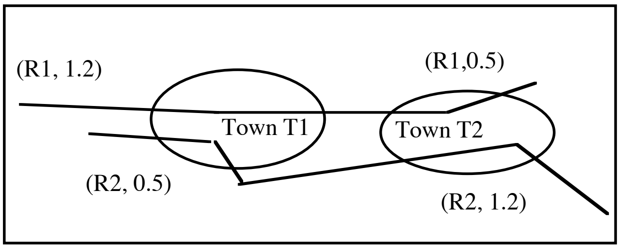

Figure 2 - The toy data set

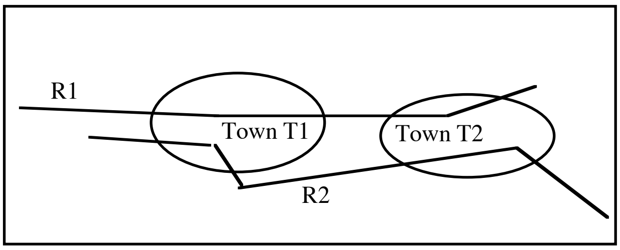

Figure 3 - The result of the query Q1

This option provides a wrong answer since the user may incorrectly infer that the pair (Town T1, Road R2) satisfies the selection criteria of the query Q1. Three solutions are possible: (1) The display mode must follow an Elementary-result-at-a-time philosophy; (2) the pairwise intersection must be detected to prevent such configurations but this operator is known as being costly or (3) the type of the result must follow a Total-oriented type.

III.3. Conclusion

The introduction of aggregate functions in the query definition process implies a modification of the current definition of the results. To avoid ambiguities, the result must follow a Total-oriented type. The display mode may be a mixture between the Elementary-result-at-a-time philosophy and the Union of elementary results philosophy depending on the network operators. The results of the network operator may be presented with a Union of elementary results philosophy but must be "enlighted" with an Elementary-result-at-a-time philosophy.

IV. Homogeneity

The homogeneity is defined as the graphical continuity of the results. The scale is one of the most significant example. The map display must provide a homogeneous scale. Nevertheless, the user may infer the scale if the distortion is continuous depending on the cartographical projection (i.e., the world at a very small scale such as 1:100,000,000).

IV.1. An example

France is a country with overseas lands. Several islands (i.e., Corsica) near the main part of France may be represented without altering the global representation for the results of query Q2. The introduction of Papeete (a French town in the Pacific Ocean) introduces a scale breaking. A worldwide display is now required. To maintain a coherent display since the major part of the results is located in continental part of France, a reduction of the non-relevant space will be introduced by providing several small lateral squares. This scale breaking leads to three ambiguities in the interpretation of the results. These ambiguities are: (1) The link Paris-Papeete does not reflect the scale or even a continuous evolution of the scale; (2) The scales may be different (i.e., the surface of the Tahiti island is 1000 km^2); (3) The relative orientation may be altered depending on the number of small squares.
The management of scale breaking may be: (1) application dependent (i.e., the SET SCALE of [3]); (2) automatically defined depending on the results of the query or (3) user defined.

IV.2 Dealing with the ambiguities

To prevent part of these ambiguities, two levels of action may be defined: (1) the definition and the manipulation of the spatial representation and (2) the definition of a display philosophy.

The Abstract Data Type (ADT) spatial_representation (see figure 1) must provide at least the definition of the scale breaking. The spatial representation of an object may be defined as the aggregation of two components: the main part and the set of dependent part. In our example, the main part is the continental representation of France and the dependent parts are the overseas lands not included in the continental representation. To handle such data, an "encrust" operator must be defined to provide the merge of the main part and the relevant dependent parts. Since dependent parts are the recursive application of the map definition, all data required for the main part are also available for each dependent part (i.e., the scale).
Let [ ] and { } be the database aggregation constructor and set constructor. Let MAP be the basic type describing a map. The spatial_representation attribute must be at least defined by:

```
spatial_representation ::
        [  Main_part : MAP,
           {Dependent_parts : MAP} ]
```

The signature of the encrust operator is defined by:

```
encrust :: MAP x MAP -> MAP
```

The results are defined as a Total-oriented type since the network operators may be involved in a query. The Union of elementary results may be applied as soon as the query does not involved an aggregate and no scale breaking appears. Otherwise, the result management process must provide: (1) an abstraction mechanism to model the operator involving an aggregate function and (2) an instance function to provide the elementary result-at-a-time philosophy. The result is therefore defined with real data (i.e., the background screen) and abstract data (i.e., a link defined between an origin and a destination).

V. Conclusion

Visual programming languages are very promising since they are more natural than textual query language (i.e., Extended SQL) and they offer a higher level of abstraction. The results of a query must be presented with a visual formalism since the textual formalism is not adapted to manage spatial data. This paper presents some of the ambiguities implied by the visual representation of the results: the management of the aggregate functions and the homogeneity of the result. Nevertheless, several classical cartographical problems are still open such as the name placement or the aggregation.

References

[1] Boursier P., Mainguenaud M.: Spatial Query Languages: Extended SQL vs. Visual Languages vs. Hypermaps, 5th Int. Symp. on Spatial Data Handling, Charleston USA, Aug. 1992

[2] Calcinelli D., Mainguenaud M.: The Management of the Ambiguities in a Graphical Query Language for GIS, 2nd Symp. on Large Spatial Databases, Zurich, Switzerland, Aug. 1991, Lecture Notes in Computer Science n° 525

[3] Egenhofer M.: Extending SQL for Graphical Display, Cartography and Geographic Information System, Vol 18, n°4, 1991