A REVISITED DATABASE PROJECTION OPERATOR FOR NETWORK FACILITIES IN A GEOGRAPHICAL INFORMATION SYSTEM

Christophe CLARAMUNT (1), Michel MAINGUENAUD (2)

(1) The Nottingham Trent University,
Department of Computing, Burton Street, Nottingham, NG1 4BU, UK
Email: clac@doc.ntu.ac.uk

(2) Institut National des Télécommunications,
9 rue Charles Fourier, F91011 Evry, France
Email: Michel.Mainguenaud@int-evry.fr

Abstract.

Within GIS, networks have been mainly represented at the geometrical level. This paper argues that the definition of additional levels of abstraction extend the semantics of GIS networks. Networks are analysed and modelled at the conceptual and logical levels, independently of the underlying geometrical representation. The management of this two-levels representation leads to the development of a new interpretation of the database projection operator. This new operator is oriented to a semantic representation of a network. Network component properties are identified at the node and link component levels. We define four semantic constants: Global, Subset N, Subset L and Subset N,L. The semantic of alphanumeric properties are identified and their propagation with the application of network operators is characterised in function of the semantic constants. The closure of network manipulations on the database schema supports the definition of a data model that integrates the result of a network operator. This data model is built as a subset of the data models involved in network operators.

Keywords:

Geographical Information Systems, Network operators, Database model

I. INTRODUCTION

A lot of efforts are under progress to elaborate innovative solutions for the representation and exploration of complex database applications. In the context of geographical databases, several spatial data models have been identified through the integration of geometrical and topological principles (e.g., David 1993, Guting 1989, 1993 and 1995, Haas 1991, Kolovson 1993, Larue 1993, Scholl 1989, Van Oosterom 1993). Similarly, many proposals have been defined to formalise spatial query languages within databases (e.g., Brossier-Wansek 1995, Egenhofer 1990, Frank 1982, Orenstein 1988). The common and accepted model representation of discrete entities in space integrates the underlying structural constraints that organise the geographical representation. For instance, cartographic models use topological relationships to structure entities in space (Peuquet 1984). Similarly, graph structures are introduced into database models to model networks (Mainguenaud 1995). Network structures are particularly useful to represent physical or influence relationships in space. Physical networks include the distribution of electricity, gas, water or telecommunication resources. Influence networks describe economical or social patterns in space. Networks can be also used to represent spatial navigation processes in space and time, i.e., a movement, generally a human one, between several locations in space. Such processes are represented throughout cognitive representations of space that integrate complementary levels of abstraction (i.e., large and local scales according to (Kuiper 1978). In this case, network nodes represent symbolic and discrete location in space, edges displacements between these places (Claramunt 1995).

If several spatial database models and query languages have been proposed to represent the properties of networks in space (Claramunt 1996, Guting 1989, Haas 1991), the definition of a data manipulation language that operates, organises and presents a set of network queries within their geographical context is still an important research challenge to be addressed. Surprisingly, the definition of operators that address the presentation of query results has been hardly investigated within classic databases. Database query languages are mainly based on a logic of predicates that restrict a query result to a set of tuples or objects. The projection operator restricts a query result to some of the relation attributes[1]. To extend the semantics of the projection operator, aggregate functions have been proposed (e.g., sum, average, maximum, minimum, count). These functions can be integrated within the projection operator in order to extend the semantics delivered by the query result. In a GIS context, a query result combines network, spatial and alphanumerical properties. A first extension proposed to handle spatial data in query languages is the introduction of spatial predicates (e.g., in the where clause of an extended SQL-like query language). However, the semantics of these languages is not adapted to the complexity of geographical applications in which query operations are often oriented toward the spatial and logical manipulation of entities. The introduction of spatial operators, (e.g., in the select clause), improves the benefit of spatial

[1] Note that a projection operator is also defined within object-oriented database query languages as an operator which restricts resulting attributes, hides and eventually renames or redefines object properties.

operators as a new spatial semantics may be derived from their application (Guting 1989 and 1993, Haas 1991, Larue 1993, Orenstein 1988).

For alphanumeric attributes, current spatial query languages assume that the semantics of the operand components is still valid for the alphanumeric attributes delivered by a projection operator. This assumption is correct for queries based on the application of predicates in which the resulting semantics is not changed (i.e., no new attribute nor spatial representation are created). However the introduction of spatial operators within database queries change the resulting semantics as the spatial component of the query result may be derived (e.g., application of a spatial intersection operator). The propagation of alphanumeric properties within spatial operators have been studied in a previous work (Mainguenaud 1994). A classification defines the semantics of spatial entities at the topological level (i.e., an alphanumeric attribute is valid at either the interior, boundary or global spatial representation) and at a partial or global spatial representation (i.e., an alphanumeric attribute is valid, or not, for a subset of the spatial representation of this entity). The entity level defines the possible overlap, or not, in space for two instances of a same entity.

Accordingly, a data definition language must identify (1) an appropriate semantics of alphanumeric attributes and (2) rules for the propagation of these attributes within network operators. We can make a distinction between network operators that generate or not a new semantics (i.e., creation of new entities or not). That is a mandatory requirement to avoid inconsistencies or false interpretations in the analysis of network query results. This paper proposes an analysis of the semantics of GIS network operators and the definition of propagation rules that monitor the propagation of network component attributes. This analysis will support the redefinition of a projection operator suited for the presentation of query results that involve the manipulation of spatial data networks. From the database point of view, this application context represents a case study which may act as a reference and can be generalised with some minor adaptations to any network representations. From a GIS point of view, the context is of particular interest for applications oriented toward the representation of physical and semantic networks. The remainder of this paper is organised as follows. Section II presents the data model we use to explain the manipulations of networks. Section III discusses network operators and their semantics. Section IV develops the identification and application of a revisited projection operator. Section V draws the conclusion.

II. DATA MODEL SUPPORT

The definition of a network data model implies the integration of logical, geometrical and alphanumeric properties into a user-defined network representation. Therefore, modelling a network application requires the identification of the following elements:

1) *logical connections* between network entities. The concept of graph is well adapted to model such connections.
2) *geometrical properties* of the network and its component entities. In the context of our model, the geometrical level is defined by the concept of Abstract Data Type that allows an independent representation of the physical data model (Stemple 1986).
3) alphanumerical properties of the network described within a so-called *data structural model*. We use a complex object data model for the description of alphanumerical properties with three constructors: [ ] for tuple, { } for set and < > for enumerated type (similar models with a matching semantics could be used without loss of generality).
4) logical, geometrical and alphanumeric levels represented by an homogeneous *user-defined network* model.

We model a network with the concept of graph. A graph is a pair (V, E) where V is a set of vertices and E is a sub-set of V x V. An incident function maps an edge to a couple of vertices. This basic definition is completed by labelling functions. Two labelling functions allow the introduction of alphanumeric properties for vertices and edges. A node models a vertex with its labelling function. A link models an edge with its labelling function. Let us consider a graph as defined in (Cruz 1987) (Table 1).

G (V, E, ν, ε, Ψ)
V is a set of vertices, ν is a labelling function for vertices
E is a set of edges, ε is a labelling function for edges
Ψ is an incident function E -> V x V

Table 1 - Definition of a graph

In order to simplify the notation, a network built on a graph G (V, E, ν, ε, Ψ) with labelled vertices and labelled edges is denoted N (N, L). For example a node is used to model an airport, a link to model a connection between two airports. Networks can be defined from one to many level of abstractions depending on the application semantic. The integration of networks defined at complementary scales can be realised by the application of logical operators that allow to compose and decompose network subsets (Claramunt 1996, Frank 1982, Mainguenaud 1996). For the purposes of this research, we restrict the domain of study to a set of networks defined at a same abstraction level (the same principles can be easily extended to multi-level representations).

II.1. Logical and geometrical levels

A network is structured through two complementary logical and geometrical levels which are defined as follows:

1) **logical level**: A network is logically defined by a set of nodes and by a set of links. This level is independent of the underlying geographical representation. The logical level supports the application of network operators (e.g., path, paths)
2) **geometrical level**: This level describes the geometrical properties of the network entities. It permits the application of metric operators (e.g., distance, area). Spatial constraints and operations are generally derived and defined from the geometrical level. They allow to manipulate the geometrical and topological properties of network entities (e.g., adjacency, inclusion).

This two-levels structure leads to the manipulation of spatial entities at two distinct levels (1) the logical level and the application of logical operators (e.g., logical connection) (2) the geometrical level and the application of spatial operators (e.g., spatial intersection). For instance, two distinct flight routes that intersect in space may be not logically connected. The distinction between these two structural levels are generally not represented and integrated within current database models and applications. However, a distinct representation of the geometrical and logical levels is more adapted to a finer representation of the semantics of spatial networks, particularly for applications which are oriented to the modelling of logical connections (e.g., maritime or aerial navigation).

A spatial network includes two orthogonal logical and geometrical levels. The logical level is mandatory by definition. The geometrical level is optional. We propose a definition of a network in function of the completeness of its geometrical level, with the definition provided in (Guptill 1995), that is, depending on the availability or not of the spatial representation instances. Let us define two Boolean functions to handle the existence or not of a spatial representation for a node and for a link, respectively:

IsSpatialNode: Node_type -> Boolean
IsSpatialLink: Link_type -> Boolean

Definition: A network is **spatially complete** whenever a spatial representation is available for all nodes and all links.

N (N, L) build on a graph G (V, E, ν, ε, Ψ) is spatially complete <=>

$\forall n_i \in N, \forall l_i \in L$ / IsSpatialNode (n_i) = True ∧ IsSpatialLink (l_i) = True

Definition: A network is **spatially incomplete** whenever at least one spatial representation is not available for a node or a link and at least a spatial representation is available.

N (N, L) build on a graph G (V, E, ν, ε, Ψ) is spatially incomplete <=>

($\exists n_i \in N$ / IsSpatialNode (n_i) = False ∨ $\exists l_i \in L$ / IsSpatialLink (l_i) = False) ∧
($\exists n_i \in N$ / IsSpatialNode (n_i) = True ∨ $\exists l_i \in L$ / IsSpatialLink (l_i) = True)

Definition: A network is **a-spatial** whenever no spatial representation is available for all nodes and all links.

N (N, L) build on a graph G (V, E, ν, ε, Ψ) is a-spatial <=>

$\forall n_i \in N, \forall l_i \in L$ / IsSpatialNode (n_i) = False $\wedge$ IsSpatialLink (l_i) = False

From an application point of view, spatially complete networks allow to represent networks which are described by a complete geometry in space at both the node and link levels. On the other hand, spatially incomplete networks are oriented toward the representation of schematic or immaterial networks in space (i.e., spatial networks that describes influence or gravitational forces with no link geometry) or networks that are constituted by some incomplete data (i.e., some spatial representations are missing).

II.2. User defined level

Network applications require to handle the logical representation of a network (e.g., in order to define a query that delivers the flight connections or the paths between two airports), the alphanumeric properties (e.g., a query that delivers Air France flights) and the visualisation of a geometrical representation (e.g., a query that displays a map that presents the flight connections between two airports). The representation of a network must handle these complementary model levels and their semantic relationships in order to process and combine the semantics of these queries.

Let us consider an abstract data type modelling an object identifier, named Oid_type (e.g., a name, a number). Using the notations of complex objects, a graph and a network are defined with Abstract Data Types as follows:

```
Vertex_type     =   [ Oid: Oid_type]
Vertices_type   =   { Vertex_type }
Node_type       =   [ Oid: Oid_type, Representation: Spatial_Representation_type]
Nodes_type      =   { Node_type }
Edge_type       =   [ Oid: Oid_type, Origin: Vertex_type, Destination: Vertex_type ]
Edges_type      =   { Edge_type }
Link_type       =   [ Oid: Oid_type, Origin: Node_type, Destination: Node_type,
                      Representation: Spatial_Representation_type]
Links_type      =   { Link_type }
Graph_type      =   [ Vertices: Vertices_type, Edges: Edges_type ]
Graphs_type     =   { Graph_type }
Network_type    =   [ Nodes: Nodes_type, Links: Links_type ]
Networks_type   =   { Network_type}
```

Node_type, Link_type and Network_type can be respectively considered as sub-types of Vertex_type, Edge_type and Graph_type. To simplify the presentation, the set of integrity constraints defined on such a schema is not presented.

Using the previous network definitions, we define an example database that describes an international flight network between some European airports: London, Brussels, Amsterdam and Paris (Figure 1). To explain by example the different logical and geometrical configurations of a network, let us consider that this database may be a-spatial (Figure 1A), spatially complete (Figure 1B) or spatially incomplete (Figure 1C). Using the notations of complex objects, the network and its components are defined with Abstract Data Types as follows (note that the logical representation of this network and its components is still valid when the spatial attributes Representation are not included in the network and component types):

Airport_type = [Oid: Oid_type, Representation: Spatial_Representation_type,
Category: string]

Flight_type = [Oid: Oid_type, Representation: Spatial_Representation_type,
Origin: Airport_type, Destination: Airport_type,
Duration: float, Price: float]

EU-Network_type = [Airports : {Airport_type},
Flights : {Flight_type}]

Airport_type represents the nodes of the netowrks. Flight_type the links of the network. Alphanumerical and spatial attributes for Airport_type (e.g., Category) and for Flight_type (e.g., Duration) correspond to the introduction of the functions ν and ε in the definition of a graph, respectively. They constitute a network modelled by the EU-Network_type. In particular, the a-spatial network may represent the point of view of a user which is only interested in the logical connections of the flights. The Paris - Brussels and Paris - Amsterdam links overlap in the geometrical level although they are logically distinct (Figure 1B/1A). Symbolic representations in Figure 1C such as Paris or the link Paris-Amsterdam have no particular spatial location. A link such as the London-Brussels one has a spatial location which is only relevant for its origin and destination (e.g., London and Brussels).

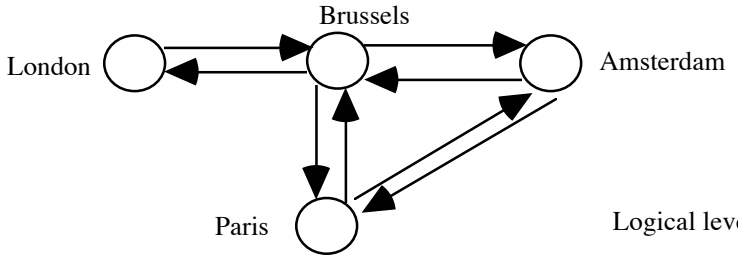

Figure 1A

Logical level of a graph

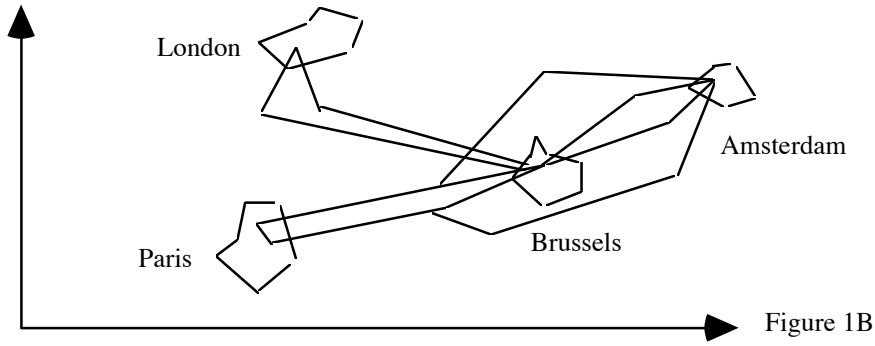

Figure 1B
Geometrical (spatially complete) level of a graph

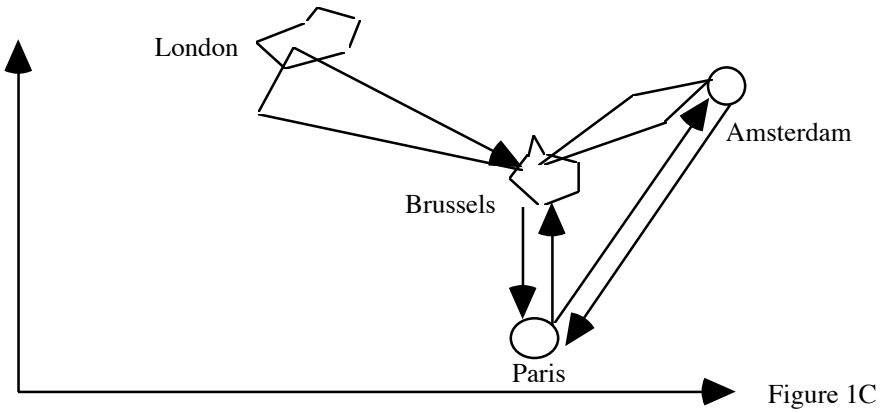

Figure 1C
Geometrical (spatially incomplete) level of a graph

Figure 1 - Example database

III. NETWORK OPERATORS

Several database query languages have been proposed for the manipulation of networks (e.g., Brossier-Wansek 1995, Cruz 1987, Guting 1989). We analyse query operators that manipulate and derive the semantics of spatial networks. The definition of a minimal set of network operators is far away the scope of this paper. We propose a classification of network operators based on a distinction of the query results in order to identify different semantic levels in the re-analysis of the projection operator. That leads to a distinction between operators that manipulate networks and network components (i.e., nodes and/or links). We analyse these network operators from a *structural*, *topological*, *semantic* or *set-oriented* points of view:

- Structural operators correspond to the manipulation of graph components - network to node and link or node and link to network (i.e., decomposition or composition operations) -. They are applied on networks or network components.
- Topological operators manipulate graph properties (e.g., paths operator), they manipulate networks or a network and some of its components and deliver a network (or a set of networks).
- Semantic operators manipulate the semantics of alphanumerical attributes for networks or network components. They correspond to classic database operators.

- Set-oriented operators are conventional set theory operators (e.g., intersection). They are applied on networks or network components.

III.1. Network-oriented operators

The main objective of network-oriented operators is to provide one (or several) network(s) as a result. We hereafter detail the classification (structural, topological, semantic and set-oriented). The Build_In operator is an example of a structural network-oriented operator applied on a set of links and on a set of nodes to define a network, i.e., composition. The Build_In operator is the equivalent of the Insert operator in a conventional database. Nodes and Links may exist in the database without being regrouped in a network (similar to an association relationship in a Entity-Relationship model). The signature of the Build_In operator is as follows:

Build_In: Links_type x Nodes_type -> Network_type

The path operator is one of the most frequent topological network-oriented operator used in network applications. However, evaluating all possible paths in a network between an origin and a destination is not a relevant query in a GIS context (i.e., from a user point of view). A limitation must therefore be introduced (Mainguenaud 1996). Two levels of requirements can be defined:

1) The first level operates at the link level (e.g., a query that delivers a path whose flight links are restricted to a specific company: Air France flights). Let us define an abstract data type Criteria_type to model such requirements.
2) The second level operates at the path level (e.g., the global duration is less than 4 hours). Let us define an abstract data type Constraints_type to model such requirements.

A path is evaluated on an underlying network. Defining the most efficient algorithm to compute the transitive closure for the evaluation of a path operator is not relevant in our context. Let us define a generic path operator. In the context of a GIS query, several origins or destinations may be defined. A constrained definition is introduced on the path operator. The evaluation must be restricted to insure a realistic time of computation. The path evaluation can be required between a single origin and a single destination, between a single origin and a set of destinations or between a set of origins and a single destination. Conventional path operators, such as defined in Gral (Guting 1989) or Starburst (Haas 1991), have the following signature:

Path: Node_type x Node_type x
Constraints_type x Criteria_type x
Network_type -> Graph_type
the result is a unique path (e.g., shortest path)

In order to provide a realistic solution for GIS applications, a relevant signature is:

Paths: Nodes_type x Nodes_type x
Constraints_type x Criteria_type x
Network_type -> Graphs_type
the result is a set of possible paths

The result of the second signature is a set of graphs. Each graph from this set is a path without a cycle (an origin, a destination and the set of links between the origin and the destination). The application of a constraint such as «the path duration is less than four hours» may be verified but the real duration is not provided. However, we argue that a valuable decision must be taken depending on several selection criteria. A shortest path may not be the most convenient criteria as additional thematic parameters may be applied. Such a query is difficult to formulate from a computational point of view. Therefore, a human interaction with the query result is highly recommendable. In the case of our example database, a query delivering a set of paths between Paris and London is the most appropriate solution as a relevant choice implies a human decision process that combines additional criteria (e.g., company preferences, timetables). However, this objective implies the introduction of additional information that represent the network semantic in order to support such an interactive decision process. Then we introduce a labelling function in the definition of a network as follows:

N (N, L, α)
where α is a labelling function that complements the network semantic.

Using the function α, the limitation of the context in which a network is embedded (Figure 1) disappears. Our example database can now be extended. Alphanumerical and spatial attributes for EU_Network_type (e.g., Condition) correspond to the introduction of the function α in the definition of a network.

EU-Network_type = [Oid: Oid_type,
Airports : {Airport_type}, Flights : {Flight_type}
Condition: string, Cumulated_Length: float,
Companies: (string), Length_Air_Traffic_Corridor: float,
Air_Traffic_Noise: float,
Context : Spatial_Representation_type]

Accordingly the signature of the path operator is as follows:

Paths: Nodes_type x Nodes_type x
Constraints_type x Criteria_type x Network_type -> Networks_type

The Largest_Common_Subgraph operator is a second example of a topological network-oriented operator. This operator is applied on two networks, and returns a network. Its signature is as follows:

Largest_Common_Subgraph: Network_type x Network_type -> Network_type

Such operators are often NP-complete. In the case of a GIS query, a network mostly represents a path. A path (without cycle in our case) is a Directed Acyclic Graph (DAG). In a path a node has at most one predecessor and one successor. This topology allows the application of operators that are in theory NP-complete but with a realistic complexity in the context of our network model (even with a large graph).

The Networks_Selection operator is an example of a semantic network-oriented operator applied on a set of networks. This operator manipulates the semantics of a set of networks verifying some selection criteria. It is similar to the relational selection operator. Its signature is as follows:

Networks_Selection: Networks_type x Criteria_type -> Networks_type

The Networks_Intersection operator is an example of a set-oriented network-oriented operator applied on two sets of networks. The resulting networks belong to the two sets of networks from the left part of the signature (e.g., same Oid). Its signature is as follows:

Networks_Intersection: Networks_type x Networks_type -> Networks_type

Table 2 summarises relevant configurations for network-oriented operators with an operator example for each identified class (structural, topological, semantic and set-oriented). The left part of a signature is presented as a regular expression on the different types. This table represents the possible operator configurations. No other left part of the signature can be involved since the pair (Links_type, Network_type) implies the definition of Nodes_type to validate Links_type, and therefore corresponds to (Network_type)+. By definition, a structural operator can only be defined from its network components in order to compose a network (i.e., applies for nodes and links). Structural operators correspond to the manipulation of graph components from a level of abstraction to another one. In the context of our model, only nodes or links (i.e., composition: Build_In operator) can be involved in the left part of the signature to provide a network. This configuration delivers for example a network from a sequence of nodes and links. By definition, topological operators involve at least a network in the signature. No topological operator can be defined with a signature that does not involve at least one network. Semantic and set-oriented operators provide a set of networks and therefore can only be defined with (Networks_type)+ in their left part of the signature.

Left part of the signature	structural	topological	semantic	set-oriented
(Nodes_type)+ (Links_type)+	Build_In			
(Nodes_type)+ Network_type		Paths		
(Network_type)+		Largest_Common_ Subgraph	Networks_ Selection	Networks_ Intersection

Table 2 - Network-oriented operators

III.2. Node-oriented operators

The main objective of node-oriented operators is the extraction of nodes. They can be applied on a network or on one (or several) set(s) of nodes or links. The choice of relevant nodes is based on structural, semantic, or set-oriented purposes. The Nodes_Projection operator is an example of a structural node-oriented operator applied on a network, i.e., decomposition (e.g., a query that delivers the cities of a network). This operator transforms a network into its unique set of nodes. Its signature is as follows:

Nodes_Projection: Network_type -> Nodes_type

The Nodes_Projection operator is applied on a Network_type (i.e., a unique network). The Origins (resp. Destinations) operator is a second example of a structural node-oriented operator. It is applied on a set of links, i.e., decomposition. This operator manipulates the structural properties and returns a set of nodes verifying that resulting nodes are the origin (resp. destination) of a link (e.g., a query that delivers the airport which is the departure of a flight). Its signature is as follows:

Origins: Links_type -> Nodes_type

The Nodes_Selection operator is an example of a semantic node-oriented operator applied on a set of nodes. This operator manipulates the semantics of nodes and returns a set of nodes that verify some selection criteria from a set of nodes (e.g., a query that delivers a set of airports of a particular category). Its signature is as follows:

Nodes_Selection: Nodes_type x Criteria_type -> Nodes_type

The Nodes_Difference operator is an example of a set-oriented node-oriented operator. It is applied on two sets of nodes and delivers a set of nodes that verify that these nodes belong exclusively to the first set of nodes in the left part of the signature. Its signature is as follows:

Nodes_Difference: Nodes_type x Nodes_type -> Nodes_type

Table 3 summarises the relevant configurations for node-oriented operators with an operator example for each class (structural, semantic and set-oriented). This table represents the possible configurations. As a node can be derived from a network or a link, structural operators are defined from Network_type and Links_type. By definition no structural operator can be defined with Nodes_type in its signature. Topological operators are not relevant for node-oriented operators since a node is an atomic component of a network. Semantic and set-oriented operators provide a set of nodes and therefore can only be defined with Nodes_type in their left part of the signature.

Left part of the signature	structural	semantic	set-oriented
Network_type	Nodes_Projection		
(Nodes_type)+		Nodes_Selection	Nodes_Difference
(Links_type)+	Origins		

Table 3 - Node-oriented operators

III.3. Link-oriented operators

The main objective of link-oriented operators is the extraction of links from a network. They deliver a set of links. They can be applied on a network or on one (or several) set(s) of links or nodes. The choice of relevant links may be based on structural, semantic or set-oriented purposes. The Links_Projection operator is an example of a structural link-oriented operator applied on a network, i.e., decomposition. This operator transforms a network into a set of links (e.g., a query that delivers the links of a network). Its signature is as follows:

Links_Projection: Network_type -> Links_type

The Build_In_Link operator is a second example of a structural link-oriented operator applied on two sets of nodes, i.e., composition. This operator derives a set of links from two sets of nodes. The semantics of the Build_In_Link operator is similar to the Build_In operator but the level of interaction is the link instead of being the network. Its signature is as follows:

Build_In_Link: Nodes_type x Nodes_type -> Links_type.

The Links_Selection operator is an example of a semantic link-oriented operator applied on a set of links. This operator manipulates the semantics of a set of links verifying some selection criteria from a set of links (e.g., a query that delivers the flights that take less than two hours). Its signature is as follows:

Links_Selection: Links_type x Criteria_type -> Links_type

The Links_Intersection operator is an example of a set-oriented link-oriented operator applied on two sets of links. This operator delivers a set of links that belongs to the two sets of links of the left part of the signature. Its signature is as follows:

Links_Intersection: Links_type x Links_type -> Links_type

Table 4 summarises the relevant configurations for link-oriented operators with an operator example for each class (structural, semantic and set-oriented). As a link may be derived from a higher abstraction level, i.e., a network, or from a lower abstraction level, i.e., a set of nodes, structural operators are defined from Nodes_type or Network_type. By definition, no structural operator can be defined with Links_type in its signature. Topological operators are not relevant for link-oriented operators since a link is an element of the Cartesian product of atomic components, i.e., nodes. Semantic and set-oriented operators are applied on sets of links and therefore can only be defined with Links_type in their left part of the signature.

Left part of the signature	Structural	Semantic	Set-Oriented
Network_type	Links_Projection		
(Links_type)+		Links_Selection	Links_Intersection
Nodes_type (Nodes_type)+	Build_In_Link		

Table 4 -Link-oriented operators

III.4. Derived operators

The efficiency of a network query language can be improved by the definition of derived operators that combine the semantics of network, nodes and links operators. The signature of these operators may involve types such as Criteria_type, Constraints_type, Network(s)_type, Nodes_type or Links_type in the left part (i.e., operands) and types Nodes_type, Links_types or Network(s)_type in the right part (i.e., result). As an example, let us define the operator Starting_Places with the following signature:

Starting_Places: Network_type -> Nodes_type

The Starting_Places operator returns the set of nodes which have no predecessor in a given network. This function does not increase the expressive power of the query language since it can be expressed by a composition of operators, on a network, Ne, using a functional notation by:

Nodes_Difference (Origins (Links_Projection (Ne)), Destinations (Links_Projection (Ne)))

Furthermore, it does not change the problematic of managing network operators as the right part of the signature is still based on the same concept: a node. However it provides a user-oriented and user-friendly operator in the context of network queries.

IV. PROJECTION OPERATOR

IV.1. Network semantic

A network is a composite entity (nodes and links) whose alphanumerical properties are constrained at the internal (within the network components) or at the external levels (e.g., by another network logically connected to one or many nodes of this network). These properties convey some analogies with the spatial domain whereas the alphanumeric properties of entities may be constrained either at the internal (e.g., interior, boundary) or external levels (e.g., merge of independent spatial maps that share some common entities).

Within network representations, the semantics of alphanumeric properties have to be analysed with respect to their node and link components. This leads to separate attributes defined (and restricted to) at the composite (i.e., network or link levels) and component levels (i.e., node, link, node and link levels). From the following three sorts, Network, Node and Link, we define the notion of derived attribute and inherited attribute. Network operators allow transformations between these three sorts: Network to Node or Link, Link to Node, and Link and Node to Network. The data model associated with the result of these operators is built from the original data model (i.e., functions α, ν, ε) and attributes that can be provided by the operand(s). An attribute is said to be derived whenever its definition is provided by the application of an aggregate function on operand(s) (i.e., creation of a new semantics). An attribute is said to be inherited whenever its definition is provided by the conventional projection operator or by a composition or decomposition operation (i.e., propagation of an existing semantics).

We introduce a set of semantic constants associated with network and network component attributes:

- A "Global" network attribute is an attribute whose semantics is restricted to the composite network as a whole, i.e., no inheritance of such an attribute is authorised at the node and link component levels from the network level. A «Global» network attribute is relevant for the representation of qualitative data (e.g., an attribute that represents the general condition of a network) as well as for a network attribute derived from the application of an aggregated function on nodes and links (e.g., a cumulated length attribute of a network). Similarly a "Global" link attribute is an attribute whose validity is relevant for the link as an element of the Cartesian product N x N, i.e., this attribute is not relevant for any node component (e.g., a duration attribute for a flight link). By default node attributes are classified as "Global".

• A "Subset $_{N,L}$" network attribute is an attribute whose semantic is also relevant at the node and link composite levels, i.e., inheritance of such an attribute is authorised at the node and link component levels from the network (e.g., an attribute that represents the list of companies that use a network).

• A "Subset $_{N}$" network attribute is an attribute whose semantic is also relevant at the node composite level, i.e., inheritance of such an attribute is authorised at the node component level from the network level (e.g., an air traffic noise attribute). A "Subset $_{N}$" link attribute is an attribute whose validity is relevant for a subset of the element of the Cartesian product N x N (i.e., based on origin and/or destination).

• A "Subset $_{L}$" network attribute is an attribute whose semantic is also relevant at the link composite level, i.e., inheritance of such an attribute is authorised at the link component level from the network level (e.g., length of an air traffic corridor attribute).

A semantic constant "Subset $_{L}$" link attribute is not directly defined at the link level as it needs an origin and a destination to exist (i.e., equivalent to a global semantics). The only way to provide a "Subset $_{L}$" link attribute is an inheritance from the network level. No "Subset $_{N}$" semantic constant is defined for a node attribute from the static point of view as a node represents an atomic component for network operators. The only way to provide a "Subset $_{N}$" node attribute is an inheritance from the network or link levels. Figure 2 describes the static definition of semantic constants for network component attributes.

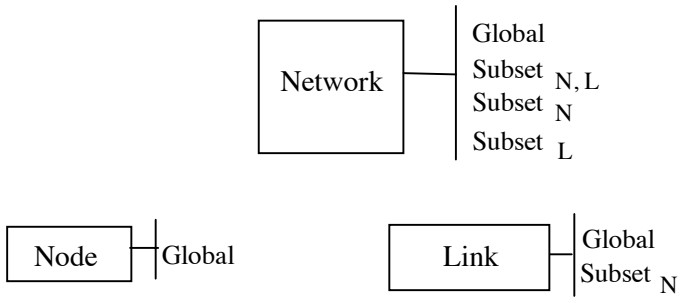

Figure 2 - Semantic constants, static definitions

We introduce the semantic properties of the network components, from the static point of view, as follows:

Let N (N, L, α) represents a network with its labelling function
Let A $_{N, L}$ be the set of Network attributes provided by the labelling function α
Let A $_{L}$ be the set of Link attributes
Let A $_{N}$ be the set of Node attributes
Let a_i be an attribute of A $_{N, L}$ we qualify the characterisation of a_i as either a Global, Subset $_{N,L}$, Subset $_{N}$ or Subset $_{L}$ attribute depending on its semantics.

Let a_i be an attribute of A_L we qualify the characterisation of a_i as either a Global or Subset $_N$ attribute depending on its semantics.
Let a_i be an attribute of A_N, we qualify the characterisation of a_i as a Global attribute.
Let C be a function that provides the classification of an attribute

Sort_type = <Network, Node, Link>

C: string x Sort_type -> <Global, Subset $_{N,L}$, Subset $_L$, Subset $_N$>

IV.2. Network operators

The left part of a signature of an operator is built with one to several sorts (Network, Link, Node). A basic right part of a query (i.e., a result) is defined as a sort. The data model associated with the result is built from the different data models involved in the left part of the signature. Semantic constants are associated with each attribute involved in the data model (i.e., in the left part of the signature). A transfer rule is applied from the left part to the right part of the signature in order to define the result data model (i.e., a set of data model attributes and their semantic constants).

The data model of the result is built with a sub-set of the respective data models involved in the left part and the derived attributes involved in the query. Conventional problems of database schema integration such as homonyms (i.e., two attributes with the same name but a different application semantics) are not relevant in the context of our model since the operators identified are closed on the same database schema. The unique source of conflict may be between two attributes with a same application semantic but different semantic constants. Therefore, we define an order for the semantic constants: from the lower level of abstraction to the higher: Global, $Subset_N$, $Subset_L$, $Subset_{N,L}$. Whenever a conflict arises, the priority is given to the higher level of abstraction (manipulations are closed on the database schema).

We classify network operators into two groups: intra-component and heterogeneous operators. Intra-component operators involve the same sort in the left part and in the right parts of the signature. Intra-component operators regroup semantic and set-oriented operators. They do not provide a transfer of data models from the left part to the right part of the signature as the semantics of the data models is similar. The data model result is defined as the union as some attributes may appear in the data model of the sub-set of the operands, nevertheless the semantics is similar since the sorts are identical. Heterogeneous operators regroup structural and topological operators and therefore involve different sorts in the left part. The definition of transfer rules is relevant to define the data model of the result. Figure 3 and Figure 4 illustrate the different levels of abstraction involved in network operators. The decomposition reflects the different levels of abstraction from networks to nodes (Figure 3). The composition reflects the different levels of abstraction from node to link - to build a link - and node and link to network - to define a network - (Figure 4).

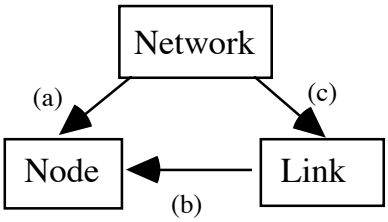

Figure 3 – Decomposition

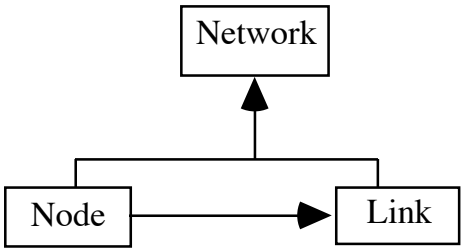

Figure 4 - Composition

IV.3. Heterogeneous operators

These operators provide a transfer since they manipulate network components rather than instances. Therefore transfer rules are defined from the conceptual point of view and are independent of the operator semantic (the approach is far different from spatial oriented operators as no creation of spatial representation is involved).

Node-oriented operators classified as structural deal with a decomposition and lead to a transfer from the network level or the link level. An attribute for a node is derived (arrow (a) Figure 3) from a network attribute classified as Subset $_{N,L}$ or Subset $_{N}$

$$A_N = A_N \cup \{ a_i \in A_{N,L} / C(a_i, \text{Network}) = \text{Subset}_{N,L} \vee C(a_i, \text{Network}) = \text{Subset}_N \}$$

An attribute for a node is derived (arrow (b) Figure 3) from a link attribute classified as Subset $_{N,L}$ or Subset $_{N}$

$$A_N = A_N \cup \{ a_i \in A_L / C(a_i, \text{Link}) = \text{Subset}_{N,L} \vee C(a_i, \text{Link}) = \text{Subset}_N \}$$

Link-oriented operators classified as structural deal with a decomposition and lead to a transfer from the network level. An attribute for a link is derived (arrow (c) Figure 3) from a network attribute classified as Subset $_{N,L}$, Subset $_{N}$ or Subset $_{L}$

$$A_L = A_L \cup \{ a_i \in A_{N,L} / C(a_i, \text{Network}) = \text{Subset}_{N,L} \vee C(a_i, \text{Network}) = \text{Subset}_N \vee C(a_i, \text{Network}) = \text{Subset}_L \}$$

Network-oriented operators classified as structural deal with a composition and lead to a transfer from the left part to the right part of a signature. Functions ν, ε and α are involved. A signature defined with at least a Node(s)_type sort (resp. Link(s)_type) in the left part implies a transfer of attributes from the Node(s)_type (resp. Link(s)_type), i.e., the function ν (resp. ε), to the result. The

extension of the data model - from the network on which the operator is applied or from the query definition process - is provided by aggregate constraints (i.e., function α). A network attribute is inherited from a node attribute classified as Subset $_N$. A network attribute is inherited from a link attribute classified as Subset $_{N, L}$, Subset $_N$ or Subset $_L$. A network attribute is derived from the query when it is required (e.g., calculated from an aggregate function) and is classified as Global.

$$A_{N,L} = A_{N,L} \cup \{ a_i \in A_N / C(a_i, \text{Node}) = \text{Subset}_N\} \cup \{a_i \in A_L / C(a_i, \text{Link}) = \text{Subset}_{N,L} \vee C(a_i, \text{Link}) = \text{Subset}_N \vee C(a_i, \text{Link}) = \text{Subset}_L \} \cup \{ a_i / a_i \text{ is obtained by an aggregate function on N and/or L}\}$$

Network-oriented operators classified as topological with a more complex signature than (Network_type)+ (e.g., path operator) can be considered as a special case of network-oriented operators providing a network from a network. The nodes or the links involved in the signature must already belong to the networks involved in the left part of the signature (e.g., the origin and the destination of a path operator). Table 5 summarises the transfer function for heterogeneous operators which involve a decomposition (i.e., from a network to a link or from a link to a node).

	Network -> Link	Link -> Node
Global	-	-
Subset $_N$	Subset $_N$	Subset $_N$
Subset $_L$	Subset $_L$	-
Subset $_{N,L}$	Subset $_{N,L}$	Subset $_N$

Table 5 - Transfer data model for heterogeneous operators

The semantic constant associated with an attribute in the result data model may be different from its initial one. Subset$_{N,L}$ classification is concerned by this change as the link properties is not valid anymore when the result is a node sort. The new semantic constant is Subset $_N$ in this case. We extend the definition of derived attribute as follows: an attribute is said to be derived when its classification changes when its sort changes.

IV.4. Dynamic of network operators

A sequence of network operators is realised at either a same (e.g., a sequence of Node_Difference operators applied on several node sets) or different abstraction level (e.g., a sequence of operations that successively decompose a network to links and these links to nodes - arrow (c) then (b) Figure 3). We qualify these sequences as horizontal and vertical, respectively.

The sequence of operators may be interpreted as the evaluation of a path in a graph G (V, E, ν, ε, Ψ) (Figure 5). Vertices are the different sorts involved in a signature. Edges model the transition

from the left part of a signature to the right part. The node labelling function models the sorts involved in a signature. The edge labelling function models the classification of attributes involved in the transfer function.

V = {Network, Node, Link}
E = { (Network, Network), (Network, Link), (Network, Node),
(Link, Link), (Link, Node), (Node, Node) }
ν = sorts
ε: semantic constants concerned with the transfer function from the left part to the right part

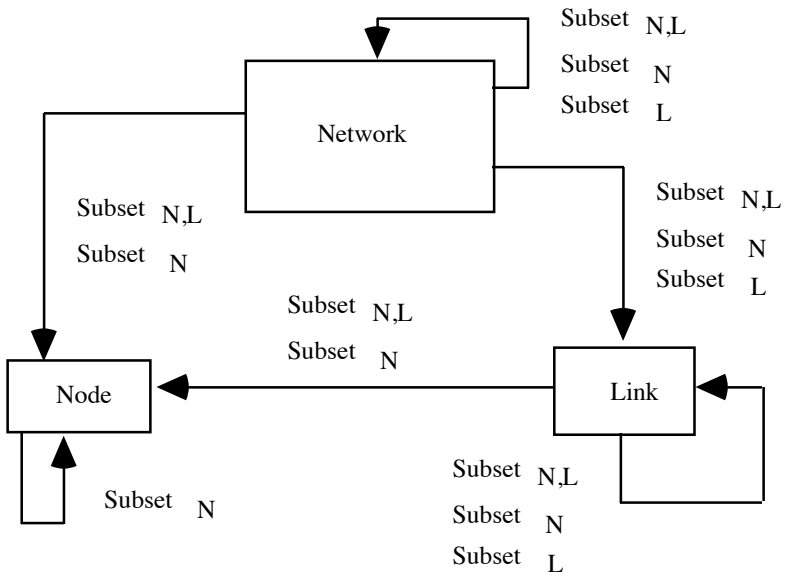

Figure 5 - Dynamic of operators

IV.5. Application

The semantic of derived results is provided by the successive propagation of semantic constants. Let us consider as an example, the Starting_Places operator. Its functional expression is:

Nodes_Difference (Origins (Links_Projection (Ne)), Destinations (Links_Projection (Ne)))

The successive propagation of attributes throughout the application of network operators (Table 5) within this Starting_Places composed operator is as follows:

Links_Projection: (Global, Subset $_N$, Subset $_L$, Subset $_{N,L}$) -> (-, Subset $_N$, Subset $_L$, Subset $_{N,L}$)
Origins: (-, Subset $_N$, Subset $_L$, Subset $_{N,L}$) -> (-, Subset $_N$, -, Subset $_N$)
Destinations (-, Subset $_N$, Subset $_L$, Subset $_{N,L}$) -> (-, Subset $_N$, -, Subset $_N$)
Nodes_Difference (-, Subset $_N$, -, -) -> (-, Subset $_N$, - , -)

Resulting node attributes are original network attributes which have Subset $_N$ or Subset $_{N,L}$ semantic constants (Subset $_{N,L}$ is transferred as a Subset $_N$ semantic constant by the Origin and Destination operators at the links level). Let us illustrate the semantic constants and the propagation rules for network operators with our example database. The application of the Starting_Places operator on a network requires the application of the Links_Projection operator. Figure 6 presents the data model transfer.

Attributes	Semantic constant	Links_Projection
Oid	Global	-
Condition	Global	-
Cumulated_Length	Global	-
Companies	Subset $_{N,L}$	Subset $_{N,L}$
Length_Air_Traffic_Corridor	Subset $_L$	Subset $_L$
Air_Traffic_Noise	Subset $_N$	Subset $_N$
Context	Subset $_{N,L}$	Subset $_{N,L}$

Figure 6 - Data model transfer for the Links_projection operator

Once the Links_Projection operator has been applied, the Origins and Destination operators are performed to transform a set of links into two sets of nodes. Figure 7 presents the second data model transfer.

Attributes	Semantic constant	Origins/Destinations
Companies	Subset $_{N,L}$	Subset $_N$
Length_Air_Traffic_Corridor	Subset $_L$	-
Air_Traffic_Noise	Subset $_N$	Subset $_N$
Context	Subset $_{N,L}$	Subset $_N$

Figure 7 - Data model transfer for the Origins/Destination operators

The Nodes_Difference operator is applied. The data model of a node is not changed as the Nodes_Difference operator belongs to the intra-component operators. Figure 8 presents the final data model associated with the nodes. This data model is enriched with a subset of the Network_type (i.e., Companies, Air_Traffic_Noise and Context attributes).

Attributes	Semantic constant
Oid	Global
Representation	Global
Category	Global
Companies	Subset $_N$
Air_Traffic_Noise	Subset $_N$
Context	Subset $_N$

Figure 8 - Final data model

V. CONCLUSION

The large diffusion of spatial database applications in scientific, planning and business domains implies the emergence of new requirements in terms of data representation and derivation. Particularly, current spatial data models and query language operations have to be extended in order to integrate a more complete semantic representation of complex domains. The integration and representation of graph structures within spatial data models is of particular interest for many application areas involved in the management of physical or immaterial networks.

This paper proposes the study of spatial network properties and their dynamic integration (i.e., using network operators) within the database projection operator. In order to represent the semantics of spatial network, we propose a classification that distinguishes the logical, spatially complete and spatially incomplete networks. The semantic of network components are studied at the network, link and node levels. The proposed model qualifies the semantics of alphanumeric attributes by a set of constants (i.e., Global, Subset $_{N}$ and Subset $_{L}$, Subset $_{N,L}$). These constants allows, or not, their propagation upward at a higher abstraction level (i.e., from nodes to link or network), downward at a lower abstraction level (i.e., from network to links / links to nodes / network to nodes) or at the same abstraction level. The propagation of these semantic properties are characterised and illustrated throughout the application of network operators that manipulate network components. Further work concerns the identification of an integrated semantic model that combines geometrical and network properties.

REFERENCES

Brossier-Wansek A. and Mainguenaud M. (1995) Manipulation of graphs with a visual query language: application to a Geographical Information System. In *Proceedings of the 3rd IFIP Conference on Visual Database Systems*, Spaccapietra, S. and Jain, R. eds., Chapman and Hall, London, pp. 227-246.

Claramunt C. and Mainguenaud, M. (1995) Dynamic and flexible vision of a spatial database. In *Proceedings of the DEXA'95 International Workshop on Database and Expert System Applications*, Revell, N. and Min Tjoa, A. eds., Omnipress, London, pp. 483-492.

Claramunt C. and Mainguenaud M. (1996) A spatial data model for navigation knowledge. In *Advances in GIS Research II*, Kraak, M-J. and Molenaar, M. eds., Taylor and Francis, Delft, pp. 345-357.

Cruz I. F. and Mendelzon A. O. and Wood P. T. (1987) A graphical query language supporting recursion. In *Proceedings of the ACM SIG Management Of Data -SIGMOD- Conference*, San-Fransisco, pp. 323-330.

David B., Raynal L., Schorter G. and Mansart V. (1993) GeO2: Why objects in a geographical DBMS. In *Advances in Spatial Databases*, Abel, D. J. and Ooi, B. C. eds. Springer-Verlag, Singapore, Lecture Notes in Computer Science, n. 692, pp. 264-276.

Egenhofer M. (1990) Manipulating the graphical representation of query results in geographic information systems. In *Proceedings of the IEEE Workshop on Visual Languages*, Skokie, Illinois, pp. 119-124.

Frank A. U. (1982) Mapquery - Database query language for retrieval of geometric data and its graphical representation, *ACM Computer Graphics*, 16 (3), pp. 199-207.

Guptill S.C. and Morrison J.L. (1995), *Elements of Spatial Data Quality*, Elsevier Science and the International Cartographic Association, Oxford, UK.

Guting R. H. (1989) GRAL: An extensible relational database system for geometric applications. In *Proceedings of the 15th International Conference on Very Large Data Bases*, VLDB, Amsterdam, p. 33-44.

Guting R. H. and Schneider M. (1993) Realms: A foundation for spatial data types in database systems. In *Advances in Spatial Databases*, Abel, D. and Ooi, B. C. eds., Springer-Verlag, Singapore, Lecture Notes in Computer Science, n. 692, pp. 14-35.

Guting R. H., de Ridder T. and Schneider M. (1995) Implementation of the ROSE algebra: Efficient algorithms for Realm-based spatial data types. In *Advances in Spatial Databases*, Egenhofer, M. J. and Herring, J. R. eds., Portland, Springer-Verlag, pp. 216-239.

Haas L. and Cody W. F. (1991) Exploiting extensible DBMS in integrated GIS. In *Proceedings of the 2nd International Symposium on Large Spatial Database,* Gunther, O. and Schek, H.-J. eds., Springer-Verlag, Zurich, Lecture Notes in Computer Science, n. 525, pp. 423-450.

Kolovson P.C., Neimat M. and Potamianos S. (1993) Interoperability of spatial and attribute data managers: A case study. In *Advances in Spatial Databases*, Abel, D. J. and Ooi, B. C. Eds. Springer-Verlag, Singapore, Lecture Notes in Computer Science, n. 692, pp. 239-263.

Kuiper B. (1978) Modelling spatial knowledge. *Cognitive Science*, 2, pp. 129-153.

Larue T., Pastre D. and Viemont Y. (1993) Strong integration of spatial domains and operators in a relational database system. In *Advances in Spatial Databases*, Abel, D. J. and Ooi, B. C. eds., Springer-Verlag, Singapore, Lecture Notes in Computer Science, n. 692, pp. 53-71.

Mainguenaud M. (1994) Consistency of geographical information system results. *Computers, Environment and Urban Systems*, Pergamon Press, Vol. 18, pp. 333-342.

Mainguenaud M. (1995) The modelling of the geographic information system network component. *International Journal of Geographical Information Systems,* 593, Taylor and Francis, 9 (6), pp. 573-595.

Mainguenaud M. (1996) Constraint-based queries in a geographical database for network facilities. *Computers, Environment and Urban Systems*, Pergamon Press, Vol. 20, pp. 139-151.

Orenstein J. A., Manola F.A. (1988) PROBE spatial data modeling and query processing in an image database application. *IEEE Transactions on Software Engineering*, 14 (6), pp. 611-629.

Peuquet D. J. (1984) A conceptual framework and comparison of spatial data models. *Cartographica*, 21 (4), pp. 66-113.

Scholl M. and Voisard A. (1989) Thematic map modelling. In *Proceedings of the 1st International Symposium on Large Spatial Databases*, Santa-Barbara, Springer-Verlag, Lectures Notes in Computer Science, n. 409, pp. 167-190.

Stemple D., Sheard T. and Bunker R. (1986) Abstract data types in databases: Specification, Manipulation and Access. In *Proceedings of the 2nd Int. Conference on Data Engineering*, Los Angeles, pp. 590-597.

Van Oosterom P. J. M. (1993) *Reactive Data Structures for Geographic Information Systems*, Oxford: Oxford University Press.