Towards Similarity-based Topological Query Languages

Alberto Belussi[1], Omar Boucelma[2], Barbara Catania[3], Yassine Lassoued[2], and Paola Podestà[3]

[1] DI, University of Verona, Italy
[2] LSIS-CNRS, Université Paul Cézanne Aix-Marseille III, France
[3] DISI, University of Genova, Italy

Abstract. In recent times, the proliferation of spatial data on the Internet is beginning to allow a much larger audience to access and share data currently available in various Geographic Information Systems (GISs). Unfortunately, even if the user can potentially access a huge amount of data, often, she has not enough knowledge about the spatial domain she wants to query, resulting in a reduction of the quality of the query results. This aspect is even more relevant in integration architectures, where the user often specifies a global query over a global schema, without having knowledge about the specific local schemas over which the query has to be executed. In order to overcome such problem, a possible solution is to introduce some mechanism of query relaxation, by which approximated answers are returned to the user. In this paper, we consider the relaxation problem for spatial topological queries. In particular, we present some relaxed topological predicates and we show in which application contexts they can be significantly used. In order to make such predicates effectively usable, we discuss how GQuery, an XML-based spatial query language, can be extended to support similarity-based queries through the proposed operators.

1 Introduction

The proliferation of spatial data on the Internet is beginning to allow a much larger audience to access and share data currently available in various Geographic Information Systems (GISs). As spatial data increase in importance, many public and private organizations need to disseminate and have access to the latest data at a minimum (right) cost and as fast as possible. One of the main problems in making this objective feasible is due to the gap existing between the data made available on the Web and the user's knowledge of such data during query specification. Indeed, the user may not exactly know the spatial domain she wants to query, in terms of properties, available features, and geometric types used to represent such features. This aspect is even more relevant in integration architectures, where a global query is expressed over a global schema, without having knowledge about the specific local schemas over which the query has to be executed. Differences in data sources may depend on how each single data

source models spatial objects in terms of their descriptive attributes (length of a river, population in a town), their type (region, line, point), their geometric type, and their topology. For example, one dataset $M1$ may represent roads and bridges as regions, another dataset $M2$ may represent roads as regions and bridges as lines, a third dataset $M3$ may represent both as lines.

The gap between stored data and user knowledge may impact the quality of the results obtained by a query execution, reducing user satisfaction in using a given application. The main cause of this unsatisfaction relies on the usage of equality-based queries, by which the user specifies in an exact way the constraints that data to be retrieved must satisfy. In order to overcome such problems, similarly to what has been done in the multimedia context, a possible solution is to introduce some mechanism of query relaxation, by which approximated answers are returned to the user, possibly introducing some false hits, but at the same time making query answers more satisfactory from the user point of view.

In this paper, we consider a specific sub-problem of the one cited above, concerning the relaxation problem for spatial topological queries, representing one of the most important classes of queries in spatial applications. In particular: (i) we present some relaxed topological predicates, that we call *weak*; (ii) we show in which application contexts they can be significantly used; (iii) we extend an existing spatial query language to cope with weak topological operations, discussing implementation issues.

Weak topological predicates are obtained from the usual one, that we call *strong*, by specifying an error threshold. Such threshold is used by the query processor to relax the topological predicate into a set of predicates, whose semantic distance from the given one is lower than or equal to the specified threshold. The definition of weak topological predicates thus relies on the usage of a similarity function between topological predicates. To this purpose, in this paper we consider the function presented in [1]. Such function extends other previously defined functions by considering pairs of topological predicates applied over pairs of objects with possibly different dimension. We then show how weak topological predicates can be used in the Web and other integration contexts and, since XML is becoming the de-facto standard for data representation and processing in such environments, we discuss how weak topological predicates can be represented using XML-like standards. In the GIS context, the OpenGIS consortium (OGC) has adopted GML (Geography Markup Language) for the XML representation and transport of geographic data [14]. GML data can be manipulated through Web Feature Services (WFSs), by which it is possible to describe or get features from a spatial data source on the Web. However, WFS is not a real query language and cannot be used to join data from different sources or to perform spatial analysis. Based on these limitations and the large diffusion of XQuery as query language for XML data, GQuery has been recently defined to overcome some of these limitations, by extending XQuery with the ability of using GML geometric types and specifying functions manipulating such types [7]. Due to its characteristics, in this paper we show how GQuery can be extended to deal with weak topological predicates, from a syntax and implementation point of view.

We remark that, even if several similarity functions for topological predicates have been defined (see for example [4, 8, 10]), the only work we are aware of dealing with similarity-based processing for spatial data is presented in [11], addressing spatial similarity for queries with multiple constraints. A methodology is proposed for spatial similarity retrieval in response to complex queries formed by combinations of logical or relational operators, in presence of null values. Spatial similarity is however considered from a conceptual rather than implementation point of view. On the other hand, here we consider a specific sub-problem of what considered in [11] and we provide concrete and easily implementable solutions. The approximation concept we consider in this paper is also different from that presented in [5], where uncertainty on object representation, due to broad boundaries, leads to the definition of approximated topological relationships.

The remainder of the paper is organized as follows. The reference model and topological distance are introduced in Section 2. Section 3 presents some scenarios of possible usage of weak topological predicates and formally introduce them. The proposed similarity-based language is then presented in Section 4, together with the discussion of some implementation issues. Finally, Section 5 presents some conclusions and outlines future work.

2 The Reference Spatial Data Model

The spatial model. We define a *map schema* as a set of feature types, object classes representing real word entities (such as lakes, rivers, etc.). Each feature type has some descriptive attributes, including a feature identifier and a *spatial attribute*, having a given dimension. We assume that values for the spatial attribute are modeled according to the OGC (Open GeoSpatial Consortium) *simple feature* geometric model [13]. In such a model, the geometry of an object can be of type: point, describing a single location in the coordinate space (dimension 0, also denoted with P); line, representing a linear interpolation of an ordered sequence of points (dimension 1, also denoted with L); polygon - more generally called region -, defined as an ordered sequence of closed lines defining the exterior and interior boundaries (holes) of an area(dimension 2, also denoted by R); recursively, a collection of disjoint geometries. We assume that the same feature type may belong to one or more map schemas, possibly with different dimensions. The instance of a map schema is called *map* and is a set of features, instances of the feature types belonging to the map schema. The same feature may belong to one or more maps, associated with possibly different geometries and dimensions according to the map schemas.

Topological relationships. Features inside a map are related by topological relationships. Topological relationships can be formally defined by using the 9-intersection model [9]. In the 9-intersection model, each spatial object A is represented by 3 point-sets: its interior A°, its exterior A^-, and its boundary ∂A. A topological relation can be represented as a 3x3-matrix, called *9-intersection matrix*, defined as follows:

Name	Definition	Object type
Disjoint (d)	$f_1 \cap f_2 = \emptyset$	All
Touch (t)	$(f_1^\circ \cap f_2^\circ = \emptyset) \wedge (f_1 \cap f_2) \neq \emptyset)$	R/R, R/L, R/P, L/L, L/P
In (i)	$(f_1 \cap f_2 = f_1) \wedge (f_1^\circ \cap f_2^\circ) \neq \emptyset)$	R/R, L/L, L/R, P/R, P/L
Contains (c)	$(f_1 \cap f_2 = f_2) \wedge (f_1^\circ \cap f_2^\circ) \neq \emptyset)$	R/R, R/L, R/P, L/L, L/P
Equal (e)	$f_1 = f_2$	R/R, L/L, P/P
Cross (r)	$dim(f_1^\circ \cap f_2^\circ) = (max(dim(f_1^\circ), dim(f_2^\circ)) - 1) \wedge$ $(f_1 \cap f_2) \neq f_1 \wedge (f_1 \cap f_2) \neq f_2$	L/R L/L
Overlap (o)	$dim(f_1^\circ) = dim(f_2^\circ) = dim(f_1^\circ \cap f_2^\circ) \wedge$ $(f_1 \cap f_2) \neq f_1 \wedge (f_1 \cap f_2) \neq f_2$	R/R L/L
Covers (v)	$(f_2 \cap f_1 = f_2) \wedge (f_2^\circ \cap f_1^\circ) \neq \emptyset) \wedge (f_1 - f_1^\circ) \cap (f_2 - f_2^\circ) \neq \emptyset$	R/R, R/L, R/P, L/L, L/P
CoveredBy (vb)	$(f_1 \cap f_2 = f_1) \wedge (f_1^\circ \cap f_2^\circ) \neq \emptyset) \wedge (f_1 - f_1^\circ) \cap (f_2 - f_2^\circ) \neq \emptyset$	R/R, L/L, L/R, P/R, P/L

Table 1. Definition of the reference set of topological relationships

$$R(A,B) = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{pmatrix}$$

The obtained relations are mutually exclusive and represent a complete coverage. In [6], this model has been extended by considering for each 9 intersection its dimension, obtaining the *extended 9-intersection model*. Since the number of such relationships is quite high, a partition of extended 9-intersection matrices has been proposed, grouping together similar matrices and assigning a name to each group. The result is the definition of the following set of binary, mutually exclusive topological relationships, refining those presented in [6]: $TREL = \{Disjoint, Touch, In, Contains, Equal, Cross, Overlap, Covers, CoveredBy\}$.[1] The semantics of such topological relationships is presented in Table 1. It is easy to show that not all relationships can be defined for any pair of dimensions. In the following, we use the notation θ_{d_1,d_2} to denote the topological relation θ applied to pairs of objects having dimension d_1 and d_2 and $REL(d_1, d_2)$ to denote the set of topological relationships defined over pairs of objects having dimension d_1 and d_2.

Topological distance. In this paper, we consider the topological distance presented in [1], defined over topological relationships represented according to the 9-intersection model. Since each topological relationship in $TREL$ corresponds to a set of 9-intersection matrices, topological distance is a total function defined in two steps: first a distance function between two 9-intersection matrices is defined, then such function is used in computing the final result.

[1] $Covers$ and $CoveredBy$ have been defined as refinements of relations $Contains$ and In and are not considered in [6].

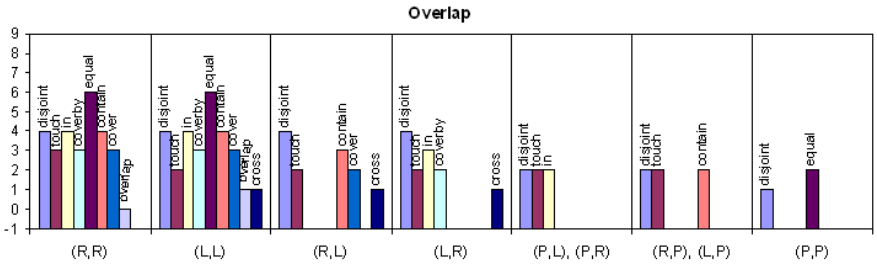

Fig. 1. Distance values (times 9) for the $Overlap_{R,R}$ topological relationship

The distance between two 9-intersection matrices ψ_1 and ψ_2 has been first defined in [8] as the number of different cells in the two matrices. Two cells are considered different if one corresponds to a non-empty intersection (whatever is its dimension) and the other to an empty intersection. Here, we normalize such distance by dividing it by the total number of cells (9).

Since each relationship in $TREL$ corresponds to a set of 9-intersection matrices, we can then compute the distance between two topological relationships $\theta^1_{d_1,d_2}$ and $\theta^2_{d_3,d_4}$ as the minimum distance between any 9-intersection matrix defining $\theta^1_{d_1,d_2}$ and any 9-intersection matrix defining $\theta^2_{d_3,d_4}$. We denote this distance by $d(\theta^1_{d_1,d_2}, \theta^2_{d_3,d_4})$.

Based on the topological distance, given a topological relationship $\theta^1_{d_1,d_2}$, all topological relationships $\theta^2_{d_3,d_4}$ can be ordered with respect to $\theta^1_{d_1,d_2}$ depending on the distance value. All values for $d(\theta^1_{d_1,d_2}, \theta^2_{d_3,d_4})$ can be found in [1]. Figure 1 just presents distances $d(Overlap_{R,R}, \theta^2_{d_3,d_4})$.

3 Weak Topological Predicates

In the following, we present two contexts in which similarity-based topological predicates can be useful. The first scenario concerns query specification in a Web context, the second scenario concerns query execution under a mediator architecture. Then, we formally introduce weak topological predicates.[2]

In the following scenarios, we use three distinct maps M_1, M_2, and M_3, sketched in Figure 2. They represent roads (identified by r_i) and bridges (identified by b_i) with different dimensions: $(2,2)$ in M_1, $(2,1)$ in M_2, and $(1,1)$ in M_3. We also assume that the following topological relationships holds:[3] (i) $Overlap(r1,b1)$, $Overlap(r2,b2)$, $Cover(r6,b6)$ in M_1; (ii) $Cross(r1,b1)$, $Cross(r2,b2)$, $Cross(r3,b3)$, $Cover(r6,b6)$ in M_2; (iii) $Overlap(r1,b1)$, $Cross(r2,b2)$, $Cross(r7,b7)$, $Overlap(r5,b5)$ in M_3.

[2] In the following, the term 'topological predicate' is used to denote the predicate induced by a topological relation and both notations $a\theta b$ and $\theta(a,b)$ $\theta \in TREL$ are used.

[3] For the sake of simplicity, we do not list relationships based on $Disjoint$.

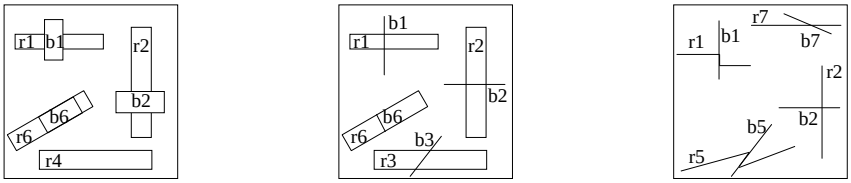

Fig. 2. Sketch of the content of the map examples

3.1 Scenario 1

Consider a user that wants to query some spatial data available on the Web, without having a detailed knowledge about such data. When the user specifies the query, she may not know the resolution of the underlying database, therefore she may not be able to specify the query in an exact way since topological predicates are not always defined when changing object dimensions. As a consequence, the quality of the obtained result may be reduced since interesting pairs may not be returned.

For example, suppose she wants to determine which pairs of roads and bridges *Overlap*, i.e., intersect and the intersection has the same type of the input objects. This query can be specified as follows: $GQ = \{(r, b) |\ r$ is a road, b is a bridge, r *Overlap* $b\}$. If roads and bridges are represented as regions, as in map M_1, the correct predicate would be *Overlap*. However, if roads and bridges are represented as lines, as in map M_3, besides *Overlap*, also predicate $Cross$, checking for intersections having dimension lower than those of input spatial objects, could be relevant for the user. If roads are represented as regions and bridges as lines, as in map M_2, *Overlap* is not defined and, based on the topological distance, $Cross$, which is the most similar predicate to *Overlap*, could be used.

In this context, a similarity-based approach could be very useful. The user could specify the query by: (i) assuming data have the maximal dimension, i.e., all polygons (in order to made available to the user the larger set of available topological predicates); (ii) providing a threshold value. Such value can be used to increase the quality of the generated result, e.g., to return more information even if not necessarily significant for the user.

For example, suppose the user wants to execute query GQ up to an error ϵ. Actually, this error depends on the user's application and needs. Let us suppose, for instance, that $\epsilon = 22\%$. If the dimension of roads and bridges in the map where the query has to be executed are d_3 and d_4, the query processor can use the topological distance introduced in Section 2 to rewrite the topological predicate *Overlap* into a set of topological predicates $\theta^1_{d_3,d_4}, ..., \theta^n_{d_3,d_4}$ such that $d(Overlap_{R,R}, \theta^i_{d_3,d_4}) \leq 0.22, i = 1, ..., n$. The union of the result sets is then returned to the user. According to Figure 1, we have that:

$d(Overlap_{R,R}, \theta_{R,R}) \leq 0.22$ for $\theta \in \{Overlap\}$

$d(Overlap_{R,R}, \theta_{R,L}) \leq 0.22$ for $\theta \in \{Cross, Cover, Touch\}$

$d(Overlap_{R,R}, \theta_{L,L}) \leq 0.22$ for $\theta \in \{Overlap, Cross, Touch\}$

Thus, the query processor rewrites GQ as follows:

- M_1: $GQ_1 = \{(r, b) |\ r$ is a road, b is a bridge, $r\ Overlap\ b\ \}$
- M_2: $GQ_2 = \{(r, b) |\ r$ is a road, b is a bridge, $r\theta b$, $\theta \in \{Cross, Cover, Touch\}\}$
- M_3: $GQ_3 = \{(r, b) |\ r$ is a road, b is a bridge, $r\theta b$, $\theta \in \{Overlap, Cross, Touch\}\}$

We notice that the user may initially not know what is the right threshold to be used in the query. However, as usual in similarity-based approaches, she may refine the threshold value, depending on the results obtained in previously executed queries, in the context of the same querying session.

3.2 Scenario 2

The second scenario deals with mediation systems. Mediation systems provide users with a uniform access to a multitude of data sources, without duplicating such data, via a common model. The user poses her query against a virtual global schema and the query is in turn rewritten into queries against the real local sources, taking into account differences in the models and query languages. The basic architecture of a mediation system is based on two main components: the mediator and the wrappers. The mediator allows "semantic translations" by rewriting the user's query into queries over data sources expressed in a common query language, which is specific to the mediator. Each data source is accessed through a wrapper. When a query is posed against a data source, the corresponding wrapper translates it according to the data source query language.

In the context of GIS data, VirGIS is a mediation system based on OpenGIS standards that addresses the issue of integrating GIS data and tools [2, 3]. In the VirGIS system, adding a new data source is easy thanks to two main things: (i) wrappers are replaced by WFS servers and there is no need to define new ones when adding a new source; (ii) VirGIS uses a mediation approach in which adding a new data source consists only in declaring its capabilities to the mediator and describing its schema (mappings) according to the global one. VirGIS supports topological operators, which are executed at the mediator level.

In general, mediator systems, including VirGIS, take into account differences concerning feature representation in local sources. However, mediators usually do not usually consider the impact of topological information on query rewriting. The problem here is that different topological predicates should be considered for execution at the local level, in order to return results that are consistent with the global request.

As an example, assume that the maps in Figure 2 represent three local sources to be integrated. Suppose that at the global level features are represented with the maximum dimension by which they appear in the local sources, in order to made available to the user the larger set of available topological predicates. In our example, this means that at the global level, *road* and *bridges* will be both represented as regions. Actually, in more general cases, the features representation, in terms of dimensions, depend on users and their applications. Specific interfaces to the users' applications can be used and may impose their own features representations. That is, for each application, we can assume that such an interface generates queries according to predefined features dimensions that are

suitable for the application. Assume now that the user, at the global level, wants to execute the query $GQ = \{(r, b)|\ r$ is a road, b is a bridge, $r\ Overlap\ b\}$.

Under this scenario, a reasonable approach for query execution at the local level would be that of rewriting the global predicate into the most similar ones (i.e., into those having the minimum distance from the global predicate) in each local source. According to Figure 1, GQ will be rewritten in the following three queries and the obtained results integrated using ad hoc merge operators:

- M_1: $GQ_1 = \{(r, b)|\ r$ is a road, b is a bridge, $r\ Overlap\ b\}$.
- M_2: $GQ_2 = \{(r, b)|\ r$ is a road, b is a bridge, $r\ Cross\ b\}$.
- M_3: $GQ_3 = \{(r, b)|\ r$ is a road, b is a bridge, $r\ \theta\ b$, $\theta \in \{Overlap, Cross\}\}$

We notice that in M_3 two predicates are considered since, according to the distance function, they have the same (minimum) distance with respect to the global predicate.

3.3 Weak Topological Predicates

In order to formally support the queries introduced above, spatial query languages should be extended with the ability of specifying similarity-based topological predicates. Such predicates relax the usual ones by allowing a certain distance between the specified predicate and those really executed. For this reason, we call them *weak topological predicates*, to distinguish them from the usual predicates, that we call *strong*. Strong predicates correspond to partial functions, on the other hand weak predicates are always defined. Given a topological relation θ_{d_3,d_4}, we also define its Nearest Neighbor relations in $REL(d_1, d_2)$ as the topological relations in $REL(d_1, d_2)$ at the minimum distance from θ_{d_3,d_4}.

Definition 1 (Strong and Weak topological predicates). *Let SO be the set of spatial objects. Let dim be a function that, given an object $o \in SO$, returns its dimension (i.e., R, L, or P). Let $d_1, d_2, d_3, d_4 \in \{R, L, P\}$. Let $\theta \in TREL$.*

- *The strong topological predicate for θ is defined as $\theta : SO \times SO \rightarrow Bool$ and $\theta(o_1, o_2) = true$ if and only if $\theta \in REL(dim(o_1), dim(o_2))$ and the conditions pointed out in Table 1 are true for o_1 and o_2. If $\theta \notin REL(dim(o_1), dim(o_2))$, $\theta(o_1, o_2)$ is undefined.*
- *The weak topological predicate for θ with respect to d_3 and d_4 is defined as $\theta^{w:d_3,d_4} : SO \times SO \times [0...1] \rightarrow Bool$ and $\theta^{w:d_3,d_4}(o_1, o_2, \rho) = true$ if there exists $\overline{\theta} \in \{\psi|\psi \in REL(dim(o_1), dim(o_2)), d(\psi, \theta_{d_3,d_4}) \leq \rho\}$ such that $\overline{\theta}(o_1, o_2)$ is true.*
- *A Nearest Neighbor topological relation in $REL(d_1, d_2)$ for θ_{d_3,d_4} is a topological relation $\overline{\theta} \in REL(d_1, d_2)$ such that $d(\overline{\theta}, \theta_{d_3,d_4}) = min\{d(\psi, \theta_{d_3,d_4})|\psi \in REL(d_1, d_2)\}$. This set of relations is denoted by $NN^{d_3,d_4}_{d_1,d_2}(\theta)$.* □

Example 1. Consider Scenario 1. If the user queries are specified over objects with the maximum resolution, GQ can be specified as follows: $GQ = \{(r, b)|\ r$ is a road, b is a bridge, $Overlap^{w:R,R}(r, b, 0.22)\}$. In Scenario 2, the global query $GQ = \{(r, b)|\ r$ is a road, b is a bridge, $Overlap(r, b)\}$ can be locally re-written as follows:

- M_1: $GQ_1 = \{(r, b) |\ r$ is a road, b is a bridge, $\theta \in NN^{R,R}_{R,R}(Overlap)$, $r\ \theta\ b\}$. In this case, $NN^{R,R}_{R,R}(Overlap) = \{Overlap\}$.
- M_2: $GQ_2 = \{(r, b) |\ r$ is a road, b is a bridge, $\theta \in NN^{R,R}_{R,L}(Overlap)$, $r\ \theta\ b\}$. In this case, $NN^{R,R}_{R,L}(Overlap) = \{Cross\}$.
- M_3: $GQ_3 = \{(r, b) |\ r$ is a road, b is a bridge, $\theta \in NN^{R,R}_{L,L}(Overlap)$, $r\ \theta\ b\}$. In this case, $NN^{R,R}_{L,L}(Overlap) = \{Overlap, Cross\}$. □

4 GQuery^s: a Similarity-Based Spatial Query Language

Weak topological predicates can be used to extend existing spatial query languages, in order to directly support similarity-based computations. Since motivations for the usage of weak topological predicates come from distributed architectures where XML is becoming the de-facto standard for data representation and processing, we discuss how weak topological predicates can be represented using XML-like standards. To this purpose, we consider GQuery [7, 2], an XML-like spatial data query language based on XQuery, for query specification, and GML, for data representation.

GML is an XML-like language for representing spatial data, proposed by the OpenGIS consortium. The basic concept is the Feature, i.e., an (object) abstraction of the real world phenomena, with spatial and non-spatial attributes. Spatial attributes may be points, lines, or polygons, as defined in Section 2. Figure 3 reports an example of GML representation for a road feature, represented as a polygon, and a bridge feature, represented as a line. Note that a polygon is defined as a (set of) LineRing, i.e., lines where the first and the last point coincide. In the following, we first present the proposed extension of GQuery, called GQuery^s, and then we discuss a possible approach for its implementation.

4.1 GQuery^s: the Syntax

A GQuery query is composed of expressions. Each expression is made up of built-in or user-defined functions. An expression is either a function call, a value, or generates an error. The result of an expression can be the input of a new one. A value is an ordered sequence of items. An item is a node or an atomic value. There is no distinction between an item and a sequence containing one value. Nodes are those defined for XQuery: document, element, attribute, text, comment, processing-instruction and namespace nodes. Writing a query consists in combining simple expression (like atomic values), path expressions (from XPath [18]), FLOWER expression (For-Let-Where-Return), test expressions (if-then-return-else-return), or (pre- or user defined) functions. Non spatial operators are arithmetic operators (+,-,×,/,mod), operators over sequences (concatenation, union, difference), comparison operators (between atomic values, nodes, and sequences), and boolean operators.

Spatial operators are applied to sequences. We have three types of spatial operators. The first two categories perform spatial analysis, the third implements

```
<Road name = 'A12'>
  <geometry>
    <gml:Polygon gid='98217'
                 srsName='http://www.opengis.net/gml/srs/epsg.xml#4326'>
      <gml:LinearRing>
        <gml:coordinates> ... </gml:coordinates>
      </gml:LinearRing>
    </gml:Polygon>
  </geometry>
</Road>

<Bridge name = 'main_bridge'>
  <geometry>
    <gml:LineString gid='45234'
                    srsName='http://www.opengis.net/gml/srs/epsg.xml#4326'>
      <gml:coordinates>....</gml:coordinates>
    </gml:LineString>
  </geometry>
</Bridge>
```

Fig. 3. An example of GML data representation

strong topological predicates (in the following *node* is a GML data node having a geometric type):

- operators which perform spatial analysis and return numeric values:
  $area, length : (node) \rightarrow numeric\,value$
  $distance : (node, node) \rightarrow numeric\,value$
- operators which perform spatial analysis and return GML values:
  $convexhull, centroid : (node) \rightarrow node$
- strong topological operators:
  $\theta : (node, node) \rightarrow boolean$ where $\theta \in TREL$.

GQuery^s is obtained from GQuery by introducing weak topological operators and a Nearest Neighbor operator is_NN, checking the Nearest Neighbor relation between two topological predicates, according to Definition 1:

- Weak topological operators are defined as follows:
  $\theta^w : (node, node, dim, dim, numeric\,value) \rightarrow\ boolean$
  where $\theta \in TREL$, $dim \in \{R, L, P\}$, $numeric\,value = [0, 1]$.
  $\theta^w(n_1, n_2, d_3, d_4, \epsilon)$ returns true if and only if $\theta^{w:d_3,d_4}(o_1, o_2, \epsilon) = true$ and o_i is the spatial object corresponding to n_i.
- The is_NN operator is defined as follows:
  $is\_NN : (TREL, dim, dim, TREL, dim, dim) \rightarrow\ boolean$
  where $dim \in \{R, L, P\}$.
  $is\_NN(r_1, d_1, d_2, r_2, d_3, d_4)$ returns true if and only if $r_1 \in NN^{d_3,d_4}_{d_1,d_2}(r_2)$.

The result of a GQuery expression is another GML document, thus GQuery is closed. Errors are raised when input parameters have not the right geometric

Determine all roads overlapping some bridge.

```
for $x in document(bridge.xml), $y in document(road.xml)
where overlap($x/geometry, $y/geometry) = true
return $x
```

Determine all roads overlapping some bridge, up to a 22% error.

```
for $x in document(bridge.xml), $y in document(road.xml)
where overlap^w($x/geometry, $y/geometry,R,L,0.22) = true
return $x
```

Fig. 4. GQuery^s examples

type. For example, the function call $overlap(node_1, node_2)$ returns a boolean value if and only if $node_1$ and $node_2$ are both polygons or lines, otherwise it raises an error. Figure 4 presents some examples of GQuery^s queries.

4.2 GQuery^s Query Processing

The GQuery^s model extends the XQuery model to deal with spatial and topological operators. This means that the GQuery^s implementation must rely on the usage of external functions. The main steps to process a query that requires a spatial processing are the following:

1. translate GML documents representing the input of the GQuery query into the right format of the input of external functions involved in the spatial computation;
2. use external spatial functions to perform the spatial computation;
3. translate the result into GML format.

GQuery^s uses as external functions the Java Topology Suite (JTS) [12], an Open Source API providing spatial object model and fundamental geometry function and strong topological relations. However, such API does not support weak topological and Nearest Neighbor operators and do not provide methods for converting JTS results into GML format. As a first step, JTS has therefore been extended in two ways, obtaining the JTS^s API:

- a new method $ConvertToGML$ is added to JTS, converting JTS Geometry Objects into GML;
- one new method is added for any weak topological predicates and one for computing the is_NN predicate. Such methods rely on Definition 1 and on the JTS implementation of strong topological predicates.

5 Conclusions and Future Work

In this paper we have presented an approach for similarity-based specification and execution of topological queries. The proposed solution relies on the definition of weak topological predicates, relaxing the traditional ones with the

specification of the maximal error allowed in executing such predicates. Topological distance between topological predicates is computed according to the function defined in [1]. In order to show the usability of the proposed concepts, we have also presented some reference application scenarios. We have finally discussed how such operators can be implemented in the context of GQuery, an XQuery-based spatial query language that can be effectively used in the identified applications. We are currently extending the VirGIS architecture [3] to deal with weak topological predicates. Future works include the extension of the proposed approach to other spatial relations, such as directional ones, the definition of a weak algebra and the analysis of its properties, the definition of query processing strategies for weak topological predicates, and an exhaustive experimentation, based on real and synthetic data.

References

1. A. Belussi, B. Catania, and P. Podestà. Towards Topological Consistency and Similarity of Multiresolution Geographical Maps. In *Proc. of ACM GIS*, pages 220–229, 2005.
2. O. Boucelma, M. Essid, and Z. Lacroix. A WFS-based Mediation System for GIS interoperability. In *Proc. of ACM GIS*, pages 23–28, 2002.
3. O. Boucelma, M. Essid, Z. Lacroix, J. Vinel, J-Y. Garinet, and A. Betari. VirGIS: Mediation for Geographical Information Systems. In *Proc. of ICDE*, pages 855–856, 2004.
4. H.T. Burns and M. J. Egenhofer. Similarity of Spatial Scenes. In *Proc. of SDH*, pages 31–42, 1996.
5. E. Clementini and P. Di Felice. A Spatial Model for Complex Objects with a Broad Boundary Supporting Queries on Uncertain Data. *Data & Knowledge Engineering*, 37(3): 285-305, 2001.
6. E. Clementini, P. Di Felice, and P. van Oosterom. A Small Set of Formal Topological Relationships Suitable for End-User Interaction. In *LNCS 692: Proc. of SSD*, pages 277–295, 1993.
7. F-M. Colonna and O. Boucelma. Querying GML Data. In *Proc. of CoPSTIC*, pages 11–13, 2003.
8. M. J. Egenhofer and K. Al-Taha. Reasoning about Gradual Changes of Topological Relationships. In *LNCS 639: Theory and Methods of Spatio-Temporal Reasoning in Geographic Space*, pages 196–219, 1992.
9. M. J. Egenhofer and J. Herring. Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases. *Tech. Rep., Dep. of Surveying Engineering, University of Maine*, 1990.
10. M. J. Egenhofer and D. Mark. Modeling Conceptual Neighborhoods of Topological Line-Region Relations. *Int. Journal of Geographical Information Systems*, 9(5):555–565, 1995.
11. K. Nedas and M. Egenhofer. Spatial Similarity Queries with Logical Operators. In *LNCS 2750: Proc. of SSTD*, pages 430-448, 2003.
12. JTS Topology Suite. http://www.vividsolutions.com/jts/jtshome.htm
13. OpenGeoSpatial Consortium. OpenGIS Simple Features Specification for SQL. *Tec. Rep., OGC 99-049*, 1999.
14. OpenGIS. Geography Markup Language (GML) 3.0. http://www.opengeospatial.org.