Mapping Network Relationships from Spatial Database Schemas to GML Documents

André C. Hora, Clodoveu A. Davis Jr., Mirella M. Moro

Universidade Federal de Minas Gerais, Brazil
{andrech,clodoveu,mirella}@dcc.ufmg.br

Abstract. Spatial data encoded in GML documents are used in various applications and are especially suited to storing, manipulating and exchanging geographic information. However, a large share of currently available spatial data is stored in spatial databases. This article presents a method to map arcs and nodes, organized in a network using spatial relationships, from a spatial database to a GML document. Specifically, a geographical conceptual schema and the corresponding GML schema are used as guide to retrieve and reorganize networking information found in the spatial database, thus generating a GML document. The proposed methodology is verified in a case study, in which networking relationships from real-world databases are mapped to GML documents that can be queried using standard XML languages such as XPath and XQuery.

Categories and Subject Descriptors: H. Information Systems [**H.m. Miscellaneous**]: Databases

Keywords: Spatial database, spatial modeling, GML data

1. INTRODUCTION

Geographical conceptual modeling is a crucial step for the successful development of spatial applications. Geographic conceptual models, such as OMT-G [Borges et al. 2001], include special primitives and spatial relationships to represent geometrical and topological characteristics of spatial data. Several types of spatial relationships can be modeled, including topological relationships, spatial aggregations, and network relationships. Network relationships occur between objects that participate in an arc-node structure, such as highway networks, electricity networks or water supply systems. For instance, in a highway network, nodes represent crossroads, while arcs represent highway segments. After the conceptual modeling phase, the resulting schema is usually mapped to a physical schema [Hora et al. 2010a] and implemented in a geographic database management system. Information is then collected and organized in a database, as determined by its design [Borges et al. 2001].

On a different perspective, XML is widely adopted as a standard language for representing, exchanging and storing structured and semi-structured data. Its auto-descriptive structure and its textual and non-proprietary format are among the reasons behind this widespread adoption. The existence of several additional languages for supporting and manipulating XML documents, such as XPath, XQuery, XML Schema and DTD, makes its use even more attractive for data management. The most important XML variation dedicated to geographic data and applications, the Geography Markup Language (GML), is a standard developed and promoted by the Open Geospatial Consortium (OGC). Data stored in spatial databases, such as Oracle Spatial or PostGIS, can be exported to or imported from XML or GML, but without semantic interpretation of the data. GML documents are also used in the specification of OGC Web services such as Web Map Service (WMS), Web Feature Service (WFS) and others.

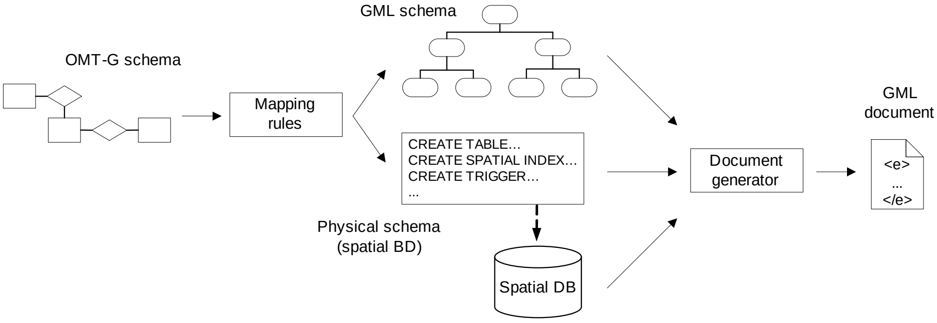


Fig. 1. Steps to generate the GML document

Considering the various applications of GML, and the need to exchange data between systems, this work presents a methodology for mapping geographical data, specifically arc-node network relationships, from the contents of a spatial database to GML documents, without incurring in any loss of information. Naturally, in order to avoid semantic losses, such a mapping must consider the definitions provided by the original conceptual schema (not just by looking at the stored tables). Therefore, the process sstarts with conceptual modeling, then the conceptual schema is transformed into both a GML schema and a spatial database physical schema. Using the GML schema and the geographic information loaded into tables defined by the physical schema, a proper GML document can be built. Query languages such as XPath and XQuery can then be used to query the resulting GML document. Figure 1 shows a general view of these steps.

This article is organized as follows. Section 2 discusses related work. Section 3 presents the geographical conceptual modeling step, based on OMT-G. Section 4 presents a formalization of arc-node networks, the generation of GML documents and the mapping algorithms. Section 5 shows a case study on how to obtain GML documents from data stored in a real spatial database. Finally, Section 6 presents conclusions and future work.

2. RELATED WORK

There are different techniques for mapping conceptual schemas to XML schemas. Pigozzo and Quintarelli [2005] present an algorithm to generate XML schemas from extended entity-relationship (EER) schemas. The resulting XML schema contains all the characteristics of the original one: entities, relationships, attributes, cardinality and specializations. Franceschet et al. [2007] map primitives from the ChronoGeoGraph spatio-temporal model to XML and GML schemas, but the mapping algorithm is not detailed. Liu and Li [2006] propose creating XML schemas from entity-relationship (ER) schemas with some quality criteria. Some design goals are sought by all of these approaches: preservation of modeling information and integrity constraints, absence of redundancy, highly connected structure, and reversible result. Few details on the schema mapping algorithm and the criteria for selecting first-level classes (direct descendants of the root node in an XML schema) are given.

Park et al. [2003] propose a system that converts geographic information stored in a spatial database into GML documents for interoperable use by various GIS applications. Their approach, however, does not perform semantic interpretation of the data stored in the database. On a different perspective, the generation of synthetic XML documents is studied in detail by Barbosa et al. [2002].

In this article, we use the OMT-G conceptual model as a starting point, since it has been developed by our group, is widely used in Brazil, and includes all the necessary primitives for sophisticated spatial data modeling. Of course, the ideas and techniques presented here can be adapted to other spatial data models, provided they have primitives for arc-node network modeling. The mapping methodology, the transformation algorithm and the general rules for mapping between schemas have been introduced in Hora et al. [2010a; 2010b], but the coverage of network relationships is expanded in this article.

3. GEOGRAPHIC DATA MODELING: OMT-G AND GML SCHEMA

The OMT-G spatial data model extends Unified Modeling Language (UML) concepts and diagrams to include geographic representations and special kinds of relationships. OMT-G uses the primitives defined for the UML class diagram and introduces geographic characteristics in order to enhance its capacity of representing the semantics of spatial data. Classes and relationships are the basic primitives for the creation of OMT-G schemas. Classes can be either conventional or spatial. Conventional classes do not have geographic properties and behave exactly as UML classes. Spatial classes include a geographic representation, which can be individualized (geo-object) or space-covering (geo-field). Geo-objects can be represented using points, lines, polygons or network elements (nodes, unidirectional arcs and bidirectional arcs). Geo-fields represent continuously varying phenomena, usually seen as surfaces, and their geometric representation includes isolines, tessellation, planar subdivision, triangulation, or sampling. Relationships can also be conventional (equivalent to UML relationships) or spatial. Spatial relationships include topological relationships, network relationships and spatial aggregations. Generalizations and specializations are specified as in UML, and they can be total/partial or disjoint/overlapping. Generalizations and specializations require participating classes to have the same type of representation. Another primitive, called conceptual generalization, allows the modeling of objects with multiple geographic representations. Refer to Borges et al. [2001; 2002] for more information on OMT-G.

GML Schema is a language for describing geographic objects, which are categorized into particular types. GML objects (also called features) can be concrete, such as streets or buildings; or conceptual, such as regions or city blocks. Objects are described in terms of their properties (which include geometric shape and location) and descriptive properties (as in database attributes). A geometry property can usually be defined as a Polygon, a LineString, or a Point, but there are other geometric representations in GML Schema. The GML framework does not define geographic representations for specific applications. To define these objects and their purpose, the designer needs to create a GML application schema, which is based on GML core schemas [Lake et al. 2004].

GML Schema, as opposed to conceptual models such as OMT-G, focuses on a geometric representations that are closer to the physical implementation. Hence, GML Schema represents only the basic vector geometry types and a coverage (raster) type, not including properties that allow high-level spatial concepts (e.g., isolines or planar subdivisions) to be seen as more than collections of LineStrings or Polygons. It also does not include special relationships, and as a result there is the need to create algorithms to allow mapping from conceptual schemas to GML schemas with no conceptual loss.

4. MAPPING NETWORK RELATIONSHIPS TO GML DOCUMENTS

This section presents the proposed method to map arc-node network structures from spatial databases to GML documents. Section 4.1 introduces a brief formalization of network structures. Section 4.2 presents a modeling of the problem based on the contents of an OMT-G schema and the process of generating the corresponding GML schema. Section 4.3 describes the step in which geographic data are validated and GML documents are generated, including the mapping algorithms.

4.1 Network Relationships

Conceptually, an arc-node network relationship requires the association of each arc to exactly two nodes, while each node can be related to one or more arcs, as defined as follows.

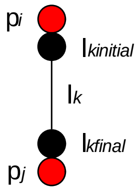


Fig. 2. Arc-node structure

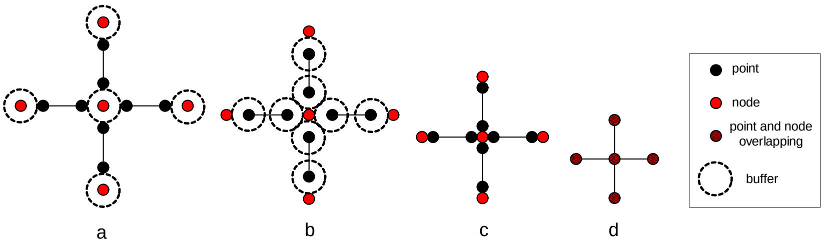


Fig. 3. Topological relationships between arcs and nodes: (a) buffer in nodes, (b) buffer in arcs, (c) arc and node touching, and (d) arc and node coinciding

Definition 1. Let $G = \{N, A\}$ be an arc-node network structure, composed of a set of nodes $N = \{n_0, n_1, ..., n_p\}$ and a set of arcs $A = \{a_0, a_1, ..., a_q\}$. Members of N and members of A are related according to the following constraints: (1) for every node $n_i \in N$ there must be at least one arc $a_k \in A$; (2) for every arc $a_k \in A$ there must be exactly two nodes $n_i, n_j \in N$.

Adapting Definition 1 to the context of spatial databases, nodes are represented by points, and arcs are represented by lines. Each line is related to exactly two points. The line's initial vertex must coincide with one of the points, and its final vertex must coincide with the other point. The required correspondence between nodes and arcs is enforced using database constraints. Thus, Definition 2 formalizes the arc-node structure from the standpoint of a spatial database.

Definition 2. Let $G = \{P, L\}$ be an arc-node network structure stored in a spatial database, composed of a set of points (nodes) $P = \{p_0, p_1, ..., p_p\}$ and a set of lines (arcs) $L = \{l_0, l_1, ..., l_q\}$. Let every line l_k be composed of a set of vertices, such that $l_k = \{v_{kinitial}, v_{k2}, ..., v_{kfinal}\}$. Members of P and members of L are related according to the following constraints: (1) every point $p_i \in P$ must be *related* to at least one vertex $v_{kinitial}$ or $v_{kfinal} \in l_k$; (2) each vertex $v_{kinitial}$ and v_{kfinal} of every line $l_k \in L$ must be *related* to exactly one point $p_i \in P$.

Figure 2 shows a generic relationship between a line (arc) l_k and the points (nodes) p_i and p_j. Relationships between points and line vertices are illustrated in Figure 3 using structures with four arcs and five nodes. Points $p_i \in P$ and vertices $v_{kinitial}$ and $v_{kfinal} \in l_k$ can be related through either the *disjoint*, *touch*, *overlap* or *coincide* topological relationships. When they are disjoint, a buffer zone at a given distance d must be created over the points, so that the arcs that are geometrically connected to the node can be identified (Figures 3a and 3b). When one of the other relationships (*touch*, *overlap* or *coincide*) occurs, the connection between node and arc can be directly identified (Figures 3c and 3d).

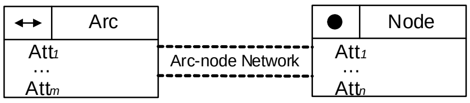
4.2 Modeling the GML Schema from OMT-G Schema

Given an OMT-G class diagram containing a network relationship as the one presented in Figure 4a, there are two alternatives for mapping the relationship to GML Schema: (1) using the basic geometric types from GML version 2, such as Point and LineString, or (2) using the GML Topology elements from GML version 3. In the first approach:

(1) XML elements are created to represent the node and arc classes along with their respective attributes;

(2) Two sub-elements $r1$ and $r2$ are created in the arc element to represent the relationship between the arc and each of the connected nodes;

(3) A sub-element GML Point and a GML LineString are respectively created in the node and arc elements, in order to represent the geometry of the elements;



(a) OMT-G network relationship

```
<element name="no">
  <complexType>
    <sequence>
      <element name="att1" type="type of att1"/>
      . . .
      <element name="attn" type="type of attn"/>
      <element ref="gml:Point"/>
    </sequence>
  </complexType>
</element>
<key name="nokey">
  <selector xpath=".//no"/>
  <field xpath="key of node"/>
</key>
```

(b) GML Schema representing a node

```
<element name="arco">
  <complexType>
    <sequence>
      <element name="att1" type="type of att1"/>
      . . .
      <element name="attm" type="type of attm"/>
      <element name="r1"/>
      <element name="r2"/>
      <element ref="gml:LineString"/>
    </sequence>
  </complexType>
</element>
<key name="arcokey">
  <selector xpath=".//arco"/>
  <field xpath="key of arc"/>
</key>
<keyref name="arcokeyref1" refer="nokey">
  <selector xpath=".//arco"/>
  <field xpath="r1"/>
</keyref>
<keyref name="arcokeyref2" refer="nokey">
  <selector xpath=".//arco"/>
  <field xpath="r2"/>
</keyref>
```

(c) GML Schema representing an arc

Fig. 4. OMT-G network relationship and its schemas in GML 2

Algorithm 1 Validating the network structure

```
Input: an arc-node network structure $G$
Output: true if the structure is consistent, false otherwise
1: Let $G = \{P, L\}$ be the arc-node network structure stored in a spatial database, composed of a set of points (nodes)
   $P = \{p_0, p_1, ..., p_p\}$ and a set of lines (arcs) $L = \{l_0, l_1, ..., l_q\}$
2: Let every line $l_k$ be composed of a set of vertices such that $l_k = \{v_{kinitial}, v_{k2}, ..., v_{kfinal}\}$
3: for each point $p_i \in P$ do
4:     Check if $p_i$ is related to at least one vertex $v_{kinitial}$ or $v_{kfinal} \in l_k$
5: end for
6: for each line $l_k \in L$ do
7:     Check if each vertex $v_{kinitial}$ and $v_{kfinal}$ of $l_k$ are related to exactly one point $p_i \in P$
8: end for
9: return the consistence of $G$
```

(4) Primary keys (XML key) are defined for arcs and nodes;

(5) Two foreign keys (XML keyref) are defined for the arcs, one for $r1$ and the other for $r2$, referencing the primary key of each related node.

Figures 4b and 4c show a GML Schema fragment created to represent the node and arc classes.

The second mapping approach is slightly different from the first one. The main difference regards the use of GML Node and GML Edge elements instead of GML Point and GML LineString. Furthermore, neither is necessary to create sub-elements for $r1$ and $r2$, nor to define foreign keys in the arcs, since standard elements from GML Topology are able to manage network relationships, as shown in the next section.

4.3 Validating the Spatial Data and Generating the GML Document

Before generating the GML document, the consistency of the network structure must be verified in the spatial database. This is done in order to avoid the creation of GML documents that are inconsistent with the conceptual constraints from the definition of the network. Algorithm 1 presents the validation. Lines 3 to 5 verify whether there is at least one arc for each node, and lines 6 to 8 verify if each arc corresponds to exactly two nodes.

After the validation, the GML document is created based on the GML Schema and on the information that is available in the spatial database. Considering the first mapping approach (GML version

Algorithm 2 Generating the GML document

```
Input: an arc-node network structure
Output: the GML document
 1: Let $G = \{P, L\}$ be the arc-node network structure stored in a spatial database, composed of a set of points (nodes)
    $P = \{p_0, p_1, ..., p_p\}$ and a set of lines (arcs) $L = \{l_0, l_1, ..., l_q\}$
 2: Let every line $l_k$ be composed of a set of vertices such that $l_k = \{v_{kinitial}, v_{k2}, ..., v_{kfinal}\}$
 3: Let every line and point be composed of a set attributes $Att_i = \{att_{i1}, att_{i2}, ..., att_{im}\}$
 4: for each point $p_i \in P$ do
 5:     Create an element $e$ in the GML document
 6:     Create each attribute of $Att_i$ of $p_i$ as a sub-element of $e$
 7:     Create the Point representing $p_i$ as sub-element of $e$
 8: end for
 9: for each line $l_k \in L$ do
10:     Create an element $e$ in the GML document
11:     Create each attribute of $Att_k$ of $l_k$ as a sub-element of $e$
12:     Let $p_i$ and $p_j$ be the points related to the vertices $v_{kinitial}$ and $v_{kfinal} \in l_k$
13:     Create two references $r1$ and $r2$ to $p_i$ and $p_j$ as a sub-element of $e$
14:     Create the LineString representing $l_k$ as sub-element of $e$
15: end for
```

2), for each node p_i and arc l_k found in the spatial database, a new element is created in the GML document. For each node or arc attribute a corresponding sub-element e is created. Initial and final vertices of each arc are related to their respective nodes, and two XML sub-elements $r1$ (origin of the arc) and $r2$ (destination of the arc) are created, including references to their nodes, p_i and p_j. A sub-element GML Point or GML LineString is then created in e, containing the geometric shape of the node or of the arc. The creation of the GML document is detailed in Algorithm 2. In lines 4 to 8, elements corresponding to the nodes are mapped, and in lines 9 to 15 arcs are mapped. The generic structure of the elements created in the GML document to represent nodes and arcs is shown in Figures 5a and 5b.

Figures 5c and 5d present the node and arc elements, considering the GML version 3 approach. Comparing with the GML version 2 alternative, one can see that the contents are the same, although using the specific arc and node elements from GML version 3. The node element (Figure 5c) is a GML Node element that contains a GML Point as its geometric shape. The arc element (Figure 5d) is a GML Edge element that contains a GML LineString and two GML directedNode elements The first element contains the `orientation="-"` attribute, and therefore corresponds to the $r1$ node; the second element has a `orientation="+"` attribute, and therefore corresponds to $r2$. The `xlink:href` attribute indicates the nodes that are related to the arc.

In spatial databases, the relationships between nodes and arcs are enforced using spatial functions. These relationships are important, since they allow navigating through the network. Since the standard XML query languages do not support the use of such spatial functions, sub-elements $r1$ and $r2$ (GML 2) and GML directedNode (GML 3) fulfill that need, since they materialize the spatial relationship at the time the GML document was created. Therefore, query languages such as XPath and XQuery can be used to query GML documents as to navigating through the network.

5. CASE STUDY

This section presents a case study to exemplify the proposed method. An OMT-G schema similar to the one represented in Figure 4a was mapped to two GML Schemas. The arc-node network represents a *street network* and a *water distribution network* in which arcs are the street segments and water pipeline segments, and nodes are the street nodes and water plumbing connections. Street segments are described by the attributes *id*, *id_thoroughfare*, *name_thoroughfare*, *type*, *r1*, *r2* and *geometry*, while street nodes are described by attributes *id*, *region* and *geometry*. Water pipeline segments include attributes *id*, *depth*, *length*, *network_type*, *diameter*, *r1*, *r2* and *geometry*, while water plumbing connections have attributes *id*, *type*, *diameter* and *geometry*.

A database instance based on the OMT-G schema was manually created and populated in Oracle Spatial, using data retrieved from the spatial database of the city of Belo Horizonte, Brazil. This

```
<nop_i>
  <att_1>key of p_i</att_1>
  . . .
  <att_n>key of att_in</att_n>
  <gml:Point>
    <gml:coordinates>
      x,y
    </gml:coordinates>
  </gml:Point>
</nop_i>
```

(a) Node element - GML 2

```
<arcol_k>
  <att_1>key of l_k</att_1>
  . . .
  <att_m>key of att_km</att_m>
  <r1>ref key of p_i</r1>
  <r2>ref key of p_j</r2>
  <gml:LineString>
    <gml:coordinates>
      x1,y1 ... xn,yn
    </gml:coordinates>
  </gml:LineString>
</arcol_k>
```

(b) Arc element - GML 2

```
<nop_i>
  <att_1>key of p_i</att_1>
    . . .
  <att_n>value of att_in</att_n>
  <gml:Node gml:id="key of p_i">
    <gml:PointProperty>
      <gml:Point>
        <gml:coordinates>
          x,y
        </gml:coordinates>
      </gml:Point>
    </gml:PointProperty>
  </gml:Node>
</nop_i>
```

(c) Node element - GML 3

```
<arcol_k>
  <att_1>key of l_k</att_1>
    . . .
  <att_m>value of att_km</att_m>
  <gml:Edge gml:id="key of l_k">
    <gml:directedNode orientation="-" xlink:href="ref key of p_i">
    <gml:directedNode orientation="+" xlink:href="ref key of p_j">
    <gml:CurveProperty>
      <gml:LineString>
        <gml:coordinates>
          x1,y1 ... xn,yn
        </gml:coordinates>
      </gml:LineString>
    </gml:CurveProperty>
  </gml:Edge>
</arcol_k>
```

(d) Arc element - GML 3

Fig. 5. Elements based on GML 2 and 3

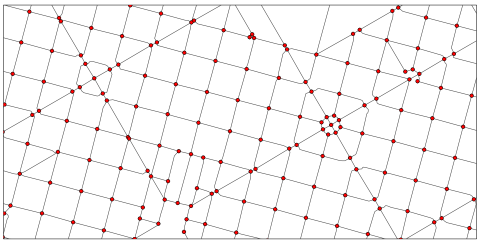


Fig. 6. Street network of Belo Horizonte

```
(a) root/street_node[id=//street_segment[id="1"]/r1
    or id=//street_segment[id="2"]/r2]

(b) for $n in root/street_node[id="2"],
        $s in root/street_segment
    where $n/id = $s/r1 or $n/id = $s/r2
    return $s
```

Fig. 7. (a) XPath and (b) XQuery queries to the street network

database contains 47,889 street segments, 31,469 street nodes, 90,742 water plumbing segments and 85,367 water plumbing nodes. In both cases, the relation between nodes and arcs occurs in a similar fashion to Figure 3d, in which nodes and arcs overlap. Figure 6 shows a graphical representation of a fragment of the street network stored in the spatial database, in which we can see the street segments and street nodes.

GML documents were then created through a prototype implemented in PL/SQL based on the proposed approach. The documents were generated using Oracle XML DB and Spatial functions, such as *xmlelement*, *xmlforest*, *xmltype* and *to_gmlgeomety*. The size of the generated documents is 24MB to the street network and 60MB to the water plumbing network, and they contain all of the arcs and nodes stored in the spatial database. The spatial data used to load the database, the generated schemas and the generated documents are available at `http://www.dcc.ufmg.br/~andrech/omtg2xml`.

As previously mentioned, network traversal can be implemented using standard XML query languages using the $r1$ and $r2$ sub-elements in each arc. Figure 7a presents an example of a XPath query that retrieves the two nodes related to the arc whose *id* is 1, and Figure 7b shows a query that retrieves all arcs related to the node whose *id* is 2 in the street network.

6. CONCLUSION

This article presented a method for mapping network relationships from spatial databases to GML documents, using the underlying conceptual schema as a guide to avoid semantic loss. Our goal is to facilitate data interoperability for GIS applications of networking. The problem has been formalized and algorithms for the generation of the target GML document were presented, considering alternatives for GML versions 2 and 3. The method was verified using a case study with real-world data from a municipal spatial database. This work is part of a project for automating the mapping between OMT-G conceptual schemas and both GML schemas and physical object-relational schemas for spatial database management systems, and includes the generation of GML documents from the contents of such databases. Future work includes improving the mapping of other OMT-G primitives, such as generalizations, specializations, conceptual generalizations, spatial aggregations, and topological relationships, to allow the generation of GML documents.

Acknowledgments

The authors acknowledge the support from FAPEMIG and CNPq for research grants and scholarships.

REFERENCES

Barbosa, D., Mendelzon, A. O., Keenleyside, J., and Lyons, K. A. ToXgene: a template-based data generator for XML. In *Proceedings of ACM SIGMOD International Conference on Management of Data*. Madison, USA, pp. 49–54, 2002.

Borges, K. A. V., Davis, C. A., and Laender, A. H. F. OMT-G: An Object-Oriented Data Model for Geographic Applications. *Geoinformatica* 5 (3): 221–260, 2001.

Borges, K. A. V., Davis, C. A., and Laender, A. H. F. Integrity Constraints in Spatial Databases. In *Proceedings of Database Integrity: Challenges and Solutions*. pp. 144–171, 2002.

Franceschet, M., Montanari, A., and Gubiani, D. Modeling and Validating Spatio-Temporal Conceptual Schemas in XML Schema. In *Proceedings of International Conference on Database and Expert Systems Applications*. Regensburg, Germany, pp. 25–29, 2007.

Hora, A. C., Davis Jr., C. A., and Moro, M. M. Generating XML/GML Schemas from Geographic Conceptual Schemas. In *Proceedings of Alberto Mendelzon International Workshop on Foundations of Data Management*. Buenos Aires, Argentina, 2010a.

Hora, A. C., Davis Jr., C. A., and Moro, M. M. Mapeamento de Relacionamentos em Rede Armazenados em Bancos de Dados Espaciais para Documentos GML. In *Proceedings of Brazilian Symposium on Databases (Short Papers)*. Belo Horizonte, Brazil, 2010b.

Lake, R., Burggraf, D., Trninic, M., and Rae, L. *Geography Mark-Up Language: Foundation for the Geo-Web.* John Wiley & Sons, 2004.

Liu, C. and Li, J. Designing Quality XML Schemas from E-R Diagrams. In *Proceedings of Conference on Web-Age Information*. Hong Kong, China, pp. 508–519, 2006.

Park, S.-Y., Lee, J.-D., and Bae, H.-Y. Easily Accessible GML-based Geographic Information System for Multiple Data Server over the Web. In *Proceedings of Information Systems Technology and its Applications*, 2003.

Pigozzo, P. and Quintarelli, E. An Algorithm for Generating XML Schemas from ER Schemas. In *Proceedings of Italian Symposium on Database Systems*. Bressanone, Italy, pp. 192–199, 2005.