CIT2016 – XII Congreso de Ingeniería del Transporte
València, Universitat Politècnica de València, 2016.
DOI: http://dx.doi.org/10.4995/CIT2016.2016.3616

DATABASES FOR HIGHWAY INVENTORIES. PROPOSAL FOR A NEW MODEL

Santiago Higuera de Frutos
Assistant professor, Technical University of Madrid (UPM), Spain
María Castro Malpica
Associate Professor, Technical University of Madrid (UPM), Spain

SUMMARY

Database models for highway inventories are based on classical schemes for relational databases: many related tables, in which the database designer establishes, a priori, every detail that they consider relevant for inventory management. This kind of database presents several problems. First, adapting the model and its applications when new database features appear is difficult. In addition, the different needs of different sets of road inventory users are difficult to fulfil with these schemes. For example, maintenance management services, road authorities and emergency services have different needs. In addition, this kind of database cannot be adapted to new scenarios, such as other countries and regions (that may classify roads or name certain elements differently). The problem is more complex if the language used in these scenarios is not the same as that used in the database design. In addition, technicians need a long time to learn to use the database efficiently. This paper proposes a flexible, multilanguage and multipurpose database model, which gives an effective and simple solution to the aforementioned problems.

1. INTRODUCTION

A highway inventory is an ordered set of data about a road network, its elements and characteristics. It is a basic element for good highway management and maintenance. Different agents are interested in highway inventories and each one has different needs. Highway administration authorities or road design and construction companies do not have the same information needs as road maintenance companies or emergency and civil protection services.

Three levels of information could be considered in highway inventories:

1. *Geometry and topology:* information about the road itself, what roads there are, where the beginning and end of the road is, the identification and denomination of roads, their horizontal and vertical alignment, their nodes (intersections or interchanges).
2. *Roadway characteristics*: for example, number of lanes, lane width, kind of pavement and slopes.
3. *Highway elements*: auxiliary works and accessories (e.g. highway drainage, traffic signals, guard-rails, noise barriers) and structures (e.g. retaining walls, bridges,

tunnels).

There are two kinds of data models for highway inventories. On the one hand, there are data models whose purpose is information exchange between different software applications. These use standards such as LandXML (LandXML, 2016), GML (Geographic Markup Language) (OGC, 2004), and several European initiatives inside the INSPIRE directive (European Commission, 2016), EuroRoadS (Svärd, 2006; Euroroads Forum 2016) and INSPIRE Specification on Transport Networks (INSPIRE, 2014). In Spain, nowadays, the National Geographic Institute is redefining its road network data model in order to adapt it to the Data Specification on Transport Networks from the INSPIRE directive (IGN, 2015). The main problem with these standards is that it is necessary to set the complete list of features and specific characteristics needed beforehand, and this results in excessive complexity. Developing applications that implement these standards is a very complex task and most developers do not use them.

The second group is made up of data models independent of standards and developed ad hoc for a specific project or software. This is the case, for example, of the Spanish Highway Inventory (Ministerio de Fomento, 2009) and also of several types of commercial inventory software (Rebolj et al., 2008). They usually employ commercial relational database software and the information about the data model is usually limited. This kind of inventory and data model cannot be used for other purposes than those for which they were conceived. If any criteria or inventory element requirements change, these applications often become obsolete. The problem is more complex if the language used in these scenarios is not the same as that used in the database design. In addition, technicians need a long time to learn to use the database efficiently.

This paper proposes a flexible, multilanguage and multipurpose database model, which gives an effective and simple solution to the aforementioned problems.

2. MODEL PROPOSAL

The data model that will be proposed is based on a non-SQL database and it does not require a previous design or the previous listing of inventory elements and their characteristics. It starts with a straightforward model, and users build and adapt it to their needs in an incremental way. This data model provides a good tool for any schema of any kind of highway inventory and for solving the information needs of any user.

A highway inventory is, basically, a geographic information database. Existing real world highway features are depicted as geographic information features stored in a database. Olaya (2012) indicates that geographic information consists of two main components:

1. *Spatial component*: refers to position inside an established reference system. It provides location and geometric information on features.
2. *Thematic component*: establishes the nature and specific characteristics of features.

Besides the spatial and thematic components of geographic information, Olaya (2012) refers to time as a third component.

Geographic information dimensions must be considered. Registered features could be from single points (0D) up to three dimensional volumes. In the model that will be proposed, three dimensional terrain is represented with altitude as a thematic variable, not as a third spatial component. This data model is an adaptation of the data model used in OpenStreetMap. From the OpenStreetMap data model it takes the characteristics best adapted for use in the highway inventory domain (Bennet, 2010; Ramm et al., 2011; OpenStreetMap, 2016).

2.1 Basic data types

Three types of basic data are defined: *Nodes*, *Lines* and *Relations*. These three basic types are used to model all highway features. The default format for representing the data model is *XML* (World Wide Web Consortium, 2015). This does not prevent the software that exploits the information generated from using formats other than XML to represent the data.

There are two attributes common to every basic data type: (1) the numerical identifier *Id*, and (2) the creation or last modification date, *timestamp*. Each has a numerical *Id*, but these are only unique within each type, so there could be a *Node, Line* and *Relation* all with the same *Id* number. There is no confusion, they are different element types. If there is a need for additional metadata, it must be provided through a *tagging procedure* (explained below). Modifications to author or program version could be examples of this kind of metadata.

The unit of thematic information is the *Tag*, which is a key-value pair. Each inventory element has a set of tags describing it. For example, to describe an element that is a traffic signal located at kilometre point 24 ("PK" in Spanish) on the N-340 highway, the tags could be those shown in Fig. 1.

- 'Traffic-sign' = 'yes'
- 'highway' = 'N-340'
- 'PK' = '24'

Fig. 1 – Example with minimum tagging for a traffic sign feature

A more detailed explanation of nodes, lines and relations is included below.

2.1.1 Nodes

Nodes are spatial points. It is the only data type with information about position. The other data types, *Lines* and *Relations*, use *Node* references in order to provide spatial information. A *Node* could indicate an inventory element, for example a traffic sign or a milestone, but it could also represent each point in a *Line*, or the junction between two roads or any change of direction on a road.

Each *Node* is uniquely identified in the database table by a numeric identifier *Id.* It also has a *TimeStamp* attribute with the date of its creation or last modification. Each *Node* also has two attributes *longitude* and *latitude,* defining its spatial position. These coordinates use the WGS-84 reference system (NIMA, 2000). Seven significant decimal places are used to obtain 1 cm accuracy in latitude and between 1 cm at the Equator and 0.6 cm at Greenwich in longitude (Bennet, 2010).

Thematic information for *Nodes* is implemented with an undetermined number of *tags* that provide all the information needed by applications. Fig. 2 shows an example of one *Node* in XML format. It is an XML element with attributes for *Id, timestamp, longitude* and *latitude.* The only child elements of the *Node* are *tag* elements, each of them with their *key* (*k*) and *value* (*v*).

```
<node id=”3450987” timestamp=”2015-02-01”
          lon=”-3.4356725” lat=”40.9878652”>
          <tag k=”traffic-sign” v=”yes” />
          <tag k=”cod” v=”P-1” />
          <tag k=”name” v=”Side road junction” />
</node>
```

Fig. 2 – Example of XML format for a Node

2.1.2 Lines

A *Line* is an ordered collection of *Nodes.* It could describe lineal inventory elements, for example a road section or a road marking. In single carriageway roads, a *Line* could be used to define road alignment. In highways with more than one carriageway or with more than one lane in each carriageway, more than one *Line* could be used.

Each *Line* element has an identifier, *Id,* and a *timestamp* attribute. It has an ordered list with the component *Nodes, nd.* These nodes are identified by their Ids. Each *Line* element also has an undetermined number of *tag* elements. These *tags* are used to describe all the thematic information. Fig. 3 shows an example of *Line* in XML format. The only child elements of *Lines* are the collection of *Nodes* and *tags.* A *Line* must have at least two nodes. The maximum number of nodes depends on database client software. The ordering of *Nodes* in a *Line* must be preserved. This order entails an implicit direction in road path, but this could be changed with tag elements. *Lines* could also be used to describe closed areas. In this case, the *Line* is the outer edge of this area. If the first and last node of a *Line* are the same, the *Line* may define a closed area, which can be tagged to indicate what the area represents. This applies to many elements, such as a parking area or other inventory elements that represent an area. However, the first and last *Node* of a *Line* can be the same and not represent an area. Roundabouts are an example of this case.

```
<line id="435672" timestamp="2012-03-27" />
            <nd id="345672" />
            …. (here there is an undetermined number of node elements)
            <nd id="345987" />
            <tag k="highway" v="yes" />
            …. (here there is an undetermined number of tag elements)
            <tag k="radius" v="450" />
            <tag k="grade" v="0.03" />
</line>
```

Fig. 3 – XML example for a *Line* element

On the other hand, *Lines* can cross without being joined. *Lines* are physically connected if they share a node, or more than one node. Two *Lines* must only be joined with a *Node* if they actually physically connect. The best example of this is a bridge carrying one road over another. It is not possible to stop halfway along the bridge and turn onto the road below, so the data used to represent the bridge and the road below should reflect this.

2.1.3 Relations

The third kind of basic data type is *Relations*. *Relations* are lists of basic data types, including other *Relations*. *Relations* exist to allow model features that cannot be described using a single *node* or *line*, or where two of the same type of feature overlap. Examples include complex intersections of highways or the turn restrictions at junctions. Fig 4 shows the XML for a *relation* representing a turn restriction. The *Relation* element has the common attributes for all the basic types, and no others. Child elements are the list of tags and the list of relation members. Each member element has attributes, giving its type and *Id* number, along with a role. The role attribute is a simple string whose values and significance are defined by the type of relation itself. The role can be left blank for relation types that do not require one. In the example, the type is given as a restriction and the members show the route that it is illegal to take using the node representing the junction with the turn restriction, and the line representing the roads that it is illegal to access.

```
<relation id="113421" timestamp="2009-11-03T10:08:27Z">
            <member type="node" id="270186" role="via"/>
            <member type="line" id="4418767" role="from"/>
            <member type="line" id="4641665" role="to"/>
            <tag k="restriction" v="no_right_turn"/>
            <tag k="type" v="restriction"/>
</relation>
```

Fig. 4 – Example of Relation in XML format

2.2 Linear referencing

Milestones (also called stations or, in Spanish, PK) are frequently used in highways as a reference system. Milestones are an approximate distance indicator and an immutable reference on highways (they can be identified in aerial photos). Each point is located using its distance from the last milestone found while going along the road (Ministerio de Fomento, 2009).

GIS applications resolve Milestones with linear referencing. Tagging nodes and route relations are used in order to implement linear referencing in the proposed data model. An example is shown in Fig 5.

```
<relation id="113421" timestamp="2009-11-03T10:08:27Z">
            <member type=”line” id=”270186” role=”1”/>
            <member type=”line” id=”4418767” role=”2”/>
            <tag k=”route” v=”A-2”/>
</relation>
<node id=”3450987” timestamp=”2015-02-01”
            lon=”-3.4356725” lat=”40.9878652”>
            <tag k=”route” v=”A-2” />
            <tag k=”pk” v=”5+0090” />
</node>
```

Fig. 5 – Example of milestone referencing

2.3 Altitude and vertical profiles

Elements needing altitude data can use a tag to define it. This way of defining altitude is known as 2.5D. If all nodes in a Line have an altitude tag, it's possible to draw and analyse the vertical profile of the line. If the application is using a database manager system with 3D analysis algorithms and functions, it can do spatial analysis in 3D.

3. DATABASE SCHEMA IMPLEMENTATION

XML will be used as the default format to exchange information between different applications. Each application will use a database manager system internally to store and process the information. A simple database schema is shown in the following paragraphs. The tables needed are at least the following:

- *Node Table:* with information about the geometry of *Nodes*. It has fields for *Id*, *timestamp*, *longitude* and *latitude*.
- *Line Table:* with information about *Lines*. It has two columns with *Id* and *timestamp*.
- *Relation Table:* with information about *Relations*. It has columns to store the *Id* and *timestamp* of the *Relations*.
- *NodeLine Table:* it has a reference to *Nodes* that make up the *Lines*. Each register has a field for the *node id*, another for the *line id* and an integer number with the position order of the node in the line.
- *RelationMember Table:* it stores reference of relation members. It has columns for *member type*, *member id* and *member role.*
- *NodeTag, LineTag and RelationTag:* they have information with tags for each *Node*,

Line or *Relation*. Each table has columns for the element *Id* and the key and value of the tag.

Any relational database can store this schema. For straightforward inventories sqlite database is a good solution. Any programming language could be used to manage sqlite (SQLITE, 2016). Sqlite also allows the use of smartphones as devices to record data or to look up data from an existing inventory (Higuera de Frutos and Castro, 2014). More complex inventories can use a database manager system located in a server. If the database manager system allows spatial analysis, live visualization and complex studies could be carried out for any inventory model.

The XML format enables an easy interchange of information between application programs. It is also easy to make tools for exchanging data with common GIS software (Shapefile format and others). Developing simple tools, this model could use OpenStreetMap data and a wide range of high quality software to edit and render data, for example, the graphic editor JOSM (JOSM, 2016) or the advanced visualization tool Mapnik (MAPNIK, 2016).

4. CONCLUSIONS

The proposed data model offers a lot of advantages compared with previous inventory data models. It is an open and known scheme, it is easy to develop software tools in any programming language, customized objectively for any agent interested in exploiting inventory information. External tools would be used for (1) creating and editing the database, (2) querying and analysing data and (3) visualizing numerically or alphanumerically.

Information could be reused by different inventories. Databases generated by an agent could easily be reused by other inventories, by adding new tags to existing elements or by creating new elements, tags and relations. Every agent interested could develop their own tools for editing, analysing and visualizing their data.

Versatility and flexibility are other advantages of this data model. The incremental process enables straightforward or complex inventories, with or without previous planning. The incremental process steers users through the creation of tags and relations needed in each inventory. This way of working allows the creation of complex inventories for highway administration authorities or straightforward models for small town halls, forest services or emergency situations after natural disasters.

Finally, it is necessary to highlight the fact that the XML format enables an easy interchange of information between application programs and that the model allows different languages to be used.

REFERENCES

BENNET, J. (2010). *OpenStreetMap.* Packt publishing, Birmingham
EUROROADS FORUM (2016). *Memorandum of Undertanding for EuroRoadS Forum.* http://www.euroroads.org/php/forum.php
EUROPEAN COMISSION (2016) *INSPIRE: Infrastructure for Spatial Information in the European Community*. European Comission http://inspire.ec.europa.eu/
HIGUERA, S. and CASTRO, M. (2014). *Using smartphones as a very low-cost tool for road inventories*, Transportation Research Part C: Emerging Technologies, Volume 38, January 2014, Pages 136-145.
IGN (2015). Especificaciones del producto Redes e Infraestructuras del Transporte (RT) del IGN. *Grupo de trabajo de Redes e Infraestructuras del Transporte*, IGN
INSPIRE (2014). *D2.8.I.7Data Specification on Transport Networks – Technical Guidelines.* European Commission Joint Research Centre.
LANDXML (2016). *LandXML.org Industry Consortium.* http://www.landxml.org
MAPNIK (2016) http://mapnik.org/
MINISTERIO DE FOMENTO (2009). *Inventario de características geométricas y de equipamiento.* Ministerio de Fomento, Madrid.
NIMA (2000). *Department of Defense World Geodetic System 1984: Its definition and Relationship with Local Geodetic Systems.* NIMA TR8350.2, Third edition, amendment 1.
OGC (2004). *LandGML Interoperability Experiment.* Open Geospatial Consortium. http://www.opengeospatial.org/projects/initiatives/landgmlie
OLAYA, V. (2012). *Sistemas de Información Geográfica.* OSGeo, Madrid.
OPENSTREETMAP (2016) http://openstreetmap.org
RAMM, F., TOPF, J. and CHILTON, S. (2011). *OpenStreetMap: using and enhancing the free map of the world.* UIT Cambridge Ltd., Cambridge.
REBOLJ, D., TIBAUT, A., ČUŠ-BABIČ, N., MAGDIČ, A. and PODBREZNIK, P. (2008) Development and application of a road product model. *Automation in Construction 17 (2008) 719–728*
SQLITE (2016). SQLite Consortium https://www.sqlite.org/
SVÄRD, T (2006). Final specification of core European road data. *EuroRoadS Forum.* http://www.euroroads.org/php/documents.php
WORLD WIDE WEB CONSORTIUM (2015). *Extensible Markup Language.* *https://www.w3.org/XML/*