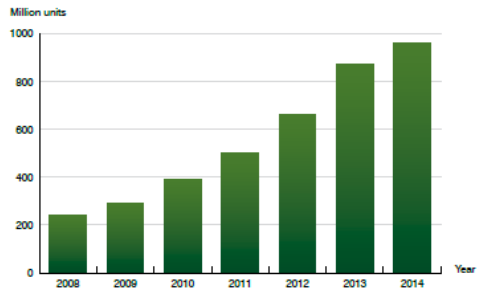
Proposing a Data Model for the Representation of Real Time Road Traffic Flow

Alex Alexandru SIROMASCENKO
Academy Of Economic Studies, Bucharest, Romania
Ingenios Soft Construct
alex_siro@yahoo.com, alex@ingenios.ro

Given recent developments in the fields of GIS data modelling, spatial data representation and storage in spatial databases, together with wireless Internet communications, it is becoming more obvious that the requirements for developing a real time road traffic information system are being met. This paper focuses on building a data model for traffic representation with support from the current free GIS resources, open source technologies and spatial databases. Community-created GIS maps can be used for easily populating an infrastructure model with accurate data; the spatial search features of relational databases can be used to map a given GPS position to the previously created network; open source ORM packages can be employed in mediating live traffic feeds into the model.
A testing mechanism will be devised in order to verify the feasibility of the solution, considering performance
Keywords: *data model, GIS, spatial database, open source technologies*

1 Introduction

Public interest in Web–enabled GIS applications has seen a rise in recent years, mainly due the decreasing cost of portable GPS devices, the increasingly sophisticated mobile devices and applications and the wider accessibility of wireless Internet services.
According to recent market research [1], GPS functions will be included in all but entry level handsets by 2014, after having grown from 8 percent to 15 percent of shipped handsets from 2008 to 2009. Shipments of GPS-enabled handsets with high-speed internet access capabilities are expected to reach 960 milion units, or 60 percent of all handset shipments, by 2014.



Annual shipments of GPS-enabled handsets (Worldwide 2008–2014)
Fig. 1. GPS-handset shipment prediction

As car ownership requires an average or above level of income, the prevalence of GPS-enabled handset ownership is expected to be higher among road traffic participants. Given these forecasts, it becomes apparent that real time collection of the positions of vechicles involved in urban traffic becomes feasible.
Traditionally, traffic data collection for traffic management systems was achieved through partially centralised networks of sensors, which were location-based, recording the traffic information for vechicles passing through a certain point/segment. The main disatvantages of such an approach compared to collecting real time data from each individual vehicle are the setup and maintenance costs of the sensors and the system`s unability to track individual journeys throughout the city.
This latter aspect is the most important, from a functional point of view, because traffic optimisation can benefit from the grouping of per vehicle data in the form of individual itineraries. With such information, a global image of traffic flows can be created, allowing planners to visualise traffic as a

collection of paths and allowing for optimisation through the allocation of itineraries.
Given these requirements, this paper will describe a potential data model for the quick manipulation and storage of traffic information.

2. Analysis of required entities

Given the requirements for real-time data collection and processing, the model will be split into 2 sub-models:

2.1 The static model

This sub-model will be conceived to deal with the representation of the road network and of the vehicles monitored by the system.

The main component of the road network is the **road segment**. A road segment is a traversable portion of a street between two points, open to vehicular traffic. Representing a segment in the data model requires considering two aspects: first, all the factors which can affect the segment`s ability to support vehicle traffic must be taken into account; second, the granularity of the segments will have a major impact on the performance of the system. All vehicle data will be represented in relation to a certain segment, and the speed of segment identification for a given vehicle is important for the process of data collection and mapping. As traffic load transfer can be carried out in any **intersection**, any given street must be split in relation to **crossroads**. The **number of lanes** also affects traffic flow, so a given segment must be further split in relation to any change in lanes number. Furthermore, if a street has two lanes, the flows on each of these lanes are independent and are part of different larger-scale flows, so a segment must be used to represent only one way of traffic.
Several road segments are grouped into a **street**. The street will be mainly used for communicating its name to the user, as it does not offer additional information over the one offered by a segment.
As a segment can be connected with more than two segments (previous and next segment, in the direction of traffic flow), navigating through segments requires a representation of intersections. A **node** is an entity which represents the meeting point of two or more segments. All the information contained in a node is also available in the segment entity, in order to minimise the number of joins required in querying. The redundance of information, in this case, is controlled – the quantity of redundant data can be determined during the generation of the road model, and does not change over time. Also, as the information is updated unfrequently, and its updates are not the subject of many use cases, the redundancy can be accepted, as it offers improved performance with minimal maintenance costs. The node organises the adjacent segments into two groups: the incoming segments and the ongoing segments. Upon arriving at a node through a given segment, a vehicle can be directed through any of the outgoing nodes. In order to optimise performance, the segments representing opposite ways on the same street can be joined together, in order to avoid “turning back” at an intersection.

In the first phase of the project, the road network model does not change over time. In a latter version, the information about the road system will be versionable, by adding a changelog system and history information about the segments, streets and nodes. As entries in these tables are marked as outdated, they will be moved to “history” partitions in order to keep a low number of records for current use, without affecting performance.
A **vehicle** is also part of the static submodel, because, although the positions of vehicles change very frequently, the number of individual vehicles taking part in traffic does not change considerably, sometimes not even on the long run.

2.2 The dynamic model

This sub-model is required to organise the data supplied during the real time usage of the monitoring system. When analysing the performance of the system, the granularity of data supplied to the dynamic model is of primary importance. As each individual vehicle is important in representing an accurate image of the real flow of traffic through the virtual model, the granularity will be controlled solely through the time interval between to consecutive reports of a vehicle`s position. In order to avoid the overcrowding of the system with update reports at certain moments, each client vehicle can be assigned its own reference moment from which data sending cycles will be calculated.

Thus, it is necessary for the dynamic model to contain a representation of the system **clock**. This entity`s purpose is to store information about a reference moment for calculating **reporting intervals**, as well as the size of these intervals. As a basic rule, given a reference moment and an interval, each newly logged in client will be assigned a moment in the last calculated reference reporting interval, in order to uniformly load the system, from a temporal point of view. Taking optimisation one step further, an algorithm can be employed in order to decrease the length of reporting intervals for more circulated road segments. In this manner, processing power is saved for the areas of the road network which can benefit from more frequent data updates, in order to maintain an accurate representation of the traffic flow. Clients will listen to changes in interval lengths while passing through different segments and adjust the communication with the server accordingly.

The dynamic model will combine the existing traffic representation paradigms. In traffic flow theory, modelling can take three approaches:

a.Microscopic model[2] – this category of models takes into account the motion of individual vechicles, with data communicated to the server on a timeframe basis. These models provide an accurate view of traffic conditions, but simulation errors can accumulate and processing costs are very high when applying them for large scale computations and predictions.

b. Macroscopic model – this type of models has been subject of intense research. *"Macroscopic traffic models consist of equations for a few aggregate quantities like the spatial density ρ, the average velocity V , and (in some cases) additional velocity moments. These equations are similar to fluid-dynamic equations, but some fundamental differences with respect to the dynamics of ordinary fluids have recently been recognized"*[3]. Macroscopic models allow for a greater efficiency in using computational resources.

c. Mesoscopic model – is a combination of the previous two models. *"Mesoscopic (meso) models tend to model the individual vehicles, but describe their behaviour in a simplified manner, on an aggregate level. A common way is to describe the vehicles' speed by using a speed/density relation that assigns an average speed based on the density of traffic ahead of the vehicle"* [4]

The mesoscopic approach brings together individual vehicle representation from the microscopic model, along with data aggregation from the macroscopic one.

Individual vehicle itineraries are important, because they can also bring additional traffic information at the macroscopic scale, other than single location segment statistics: they can be used to describe traffic flows from some regions of a city to the other ones, allowing for an accurate representation of infrastructure demands in certain timespans and for the prediction of short and long term traffic trends.

From a **microscopic** point of view, the **vehicle position** is the basis of the dynamic model. The number of vehicle positions, as mentioned before, is given by the **granularity** of updates, which can vary

depending on the congestion in the vehicle`s area. Furthermore, the efficiency of the system can be improved if the **time granularity** is replaced by the **distance granularity** in case the flow speed is very low. In this manner, frequent updates will not be received from vehicles stopped in a column at regular interval, but when these vehicles will advance a certain distance.On the other hand, a problem in maintaining an accurate view of the traffic flow might arise in case vehicles move very fast and pass through several segments during an update interval. In this case, information regarding the presence of vehicles in certain segments might be skipped. To overcome this issue, the **time referrence update system** can be complemented by **location referrence system**, which requires the client to know the start and end points of a segment and automatically report exiting/entering segments. In this case, time updating is important if it takes more time to pass a segment than the duration of the time frame.

From a **macroscopic** perspective, traffic data is represented through **segment loads**. These entities store information regarding the traffic conditions on a certain segment, during a certain interval. Segment load information is updated depending on a parameter called **load granularity**. This parameter, like the position granularity, can vary depending on the congestion of the segment, in order to better describe the conditions in more intensely circulated segments. Data describing the segment load is obtained by aggregating the microscopic information (vehicle position). The **average load** (number of vehicles / percentage of used length) and **average speed** are of primary interest when describing the flow through a segment.

When looking from a **mesoscopic** perspective, the model aggregates traffic data on a **per vehicle basis**.

The **vehicle behaviour** entity stores statistics regarding the manner in which a certain vehicle responds to different stimuli, such as the average speed of the segment and its congestion (this information is obtained from the macroscopic model). It mainly describes a relation between the flow throughout the segment and the vehicle`s speed. It allows for a behaviour-based itinerary allocation scheme, separating slower vehicles (with slower average speed or slower acceleration) from faster ones, which can use high speed segments more effectively. It can also be used to more accurately predict the travel time for each vehicle and it`s impact upon traffic, taking into account the differences between driving behaviours.

The **trip** entity groups the vehicle`s positions and creates a map of preferred departing points and destinations. Further aggregating this information into the macroscopic model, it becomes possible to correlate sources and destinations of traffic flow with respect to hourly patterns, in order to more accurately predict network loads.

3. Building the entity schema

In defining the tables, three aspects were taken into consideration:

3.1 Temporal navigation – given a certain level of aggregation (macroscopic, mesoscopic or microscopic), there must be a link between preceeding and following information and the current information. The vehicle positions must be linked as to easily reconstruct a trip and analyze the vehicle`s experience. Segment loads must be linked in order to easily emphasize the changes in load.

3.2 Inter-level navigation – navigation must be possible between the aggregation levels – a vehicle`s position (microscopic) must be linked to a traffic load (macroscopic) and to a trip (mesoscopic).

3.3 Data redundancy – because the system must deal with very large ammounts of data, processing must be done with a small number of joins between tables, in order not to affect performance. For example, the querying in the segment table in order to

navigate through the network`s state at a certain moment is optimised by joining the simultaneous segment loads through the **NodeLoad** entity.

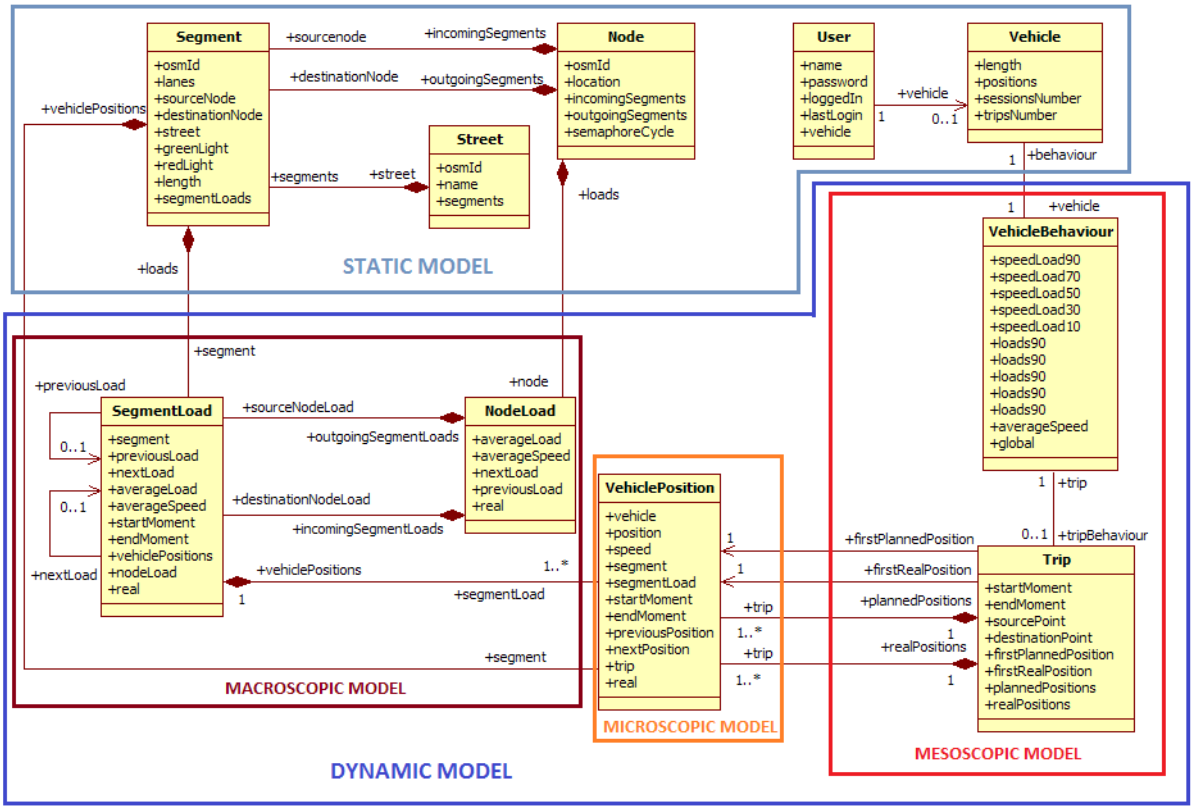


Fig. 2. Proposed data model

4. Spatial ORM

In order to allow object oriented manipulation of data during the model-view transformations and to facilitate the import of the street structure, the Hibernate Spatial ORM library was employed. This generic extension of Hibernate allows for Java representation of spatial structures defined in a spatial database (Oracle Spatial, PostgreSQL). Hibernate Spatial supports most of the functions from the OGC Simple Feature Specification.

Oracle Spatial features from the Oracle Database 11g Enterprise Edition were chosen for representing persistent data.

The Point and LineString implementations provided by Hibernate Spatial were used for mapping Java objects to SDO_GEOMETRY columns in the database[5]. Type matching was handled by the ORM, through the automatic generation of the table schema.

Example: a Java LineString attribute was mapped to an Oracle SDO_GEOMETRY column with the parameters:

SDO_GTYPE: 2002
SDO_SRID: 4326
SDO_ELEM_INFO[0]: 1
SDO_ELEM_INFO[0]: 2
SDO_ELEM_INFO[0]: 1

Additional spatial functionalities were employed through the spatial support offered by the Criteria API

The Hibernate Criteria API allows for the construction of queries from an objectual perspective. Support for spatial features is provided, through the SpatialRestrictions and OracleSpatialRestrictions objects. In this case SpatialRestrictions.within (for fetching all the segments intersecting a given area, represented by a rectangle). and OracleSpatialRestrictions.SDONN

(representation of the eponymous Oracle function, used for finding the nearest segment for a given point) were employed). These two functions make it possible to quickly map a driver on the road system, and to efficiently display real time information regarding the state of the traffic, querying only for the segments in concern.

5. Migrating data from a community project

In order to be functional, the static model must be populated with real-world data. In the context of growing interest in end-user GIS services, collaborative mapping projects have begun to appear. OpenStreetMap is an example of such a project, aiming to create a freely editable map of the world. Any type of source is accepted for map data, including GPS devices, aerial photography, local knowledge of the areas, other free sources or donations from former commercial applications.

The main component of the project is the OSM schema, running on a PostgreSQL database. The schema includes a representation of the submitted data in a proprietary form. Database information can be publicly downloaded or uploaded in the form of *.osm files, which have an XML structure.

Certain portions or the entire database can be downloaded using an online export tool

```
<way                          id="23125530"
user="me_my_self_and_I"     uid="18258"
visible="true"                version="1"
changeset="235251"  timestamp="2008-02-
29T13:36:13Z">
  <nd ref="248729665"/>
  <nd ref="129534986"/>
  <tag    k="created_by"    v="Potlatch
0.7b"/>
  <tag k="highway" v="residential"/>
  <tag  k="name"  v="Strada  Robescu  F.
Constantin"/>
 </way>
<node  id="248729665"  lat="44.4324625"
lon="26.111153"
user="me_my_self_and_I"     uid="18258"
visible="true"                version="2"
changeset="235251"  timestamp="2008-02-
29T13:35:37Z">
  <tag k="created_by" v="JOSM"/>
</node>
```

The main entities in the osm file structure are the way (representing a road, if the "highway" tag is present) and the node (representing GPS coordinates of points defining ways)[6]. In order to import the data, an XML parser was created. The goal was to transfer the way, node and way name to the entities in the proposed model: segment, node and street respectively. An efficient algorithm was devised, which prioritises "way" persistence, with all the available nodes pre-loaded into a transient map for quick fetching. Nodes referred by persisted ways are also persisted and marked as such. Thus, only required information, regarding road structure, is persisted. References to OSM ids are also persisted, in order to easily update the structure at a later point.

6. Testing the performance of algorithms employing the proposed data model

In order to test the performance of the import algorithm, and thus, the maintainability of the system, several tests were conducted, using differently sized *.osm files. The results are shown below:

Table 1. OSM file import performance

no	Segm	Nodes	Streets	Parse time	Persist time
1	5739	6127	712	1 sec.	4 sec.
2	2972	2477	284	<1 sec.	3 sec.
3	28951	52873	3523	1 sec.	16 sec.

The persistence of the entities also included spatial indexing of the nodes and segments.

Testing of the dynamic model was carried out through a random trip generation mechanism. The algorithm used a number of randomly generated pairs of departure and destination points. The points were generated within a user specified area and a trip was simulated for each pair of points. Given a source and a destination, a shortest path was found based on distance, using

the Dijkstra algorithm. After the shortest path was determined, the system simulated the actual trip. The trip involved position reports at each segment switch.

Simulation statistics:
trips: 1000
reports: 72902
total time: 2min 21sec
average report cost: 1.9 ms

The tests were conducted on a system featuring a dual-core Core i5-560M processor clocked at 2.67 GHZ and 4GB of RAM.

7. Conclusions

Live feed models for the representations of road traffic are becoming more feasible with the support of readily available resources.
In order for such a model to offer a balance between detailed data representation and quick processing of large volumes of information, the data must be split at three levels of abstraction: microscopic (vehicle positions) mesoscopic (vehicle behaviors and trips) and macroscopic (node and segment loads). The granularity of the data updates can be used to fine tune the model`s performance by increasing reports from more intensely used areas and by uniformly distributing reports, when time is considered.
Community projects such as OpenStreetMap offer an appropriate interface for getting access to an extensive database of spatial information. ORM tools with spatial features give the opportunity of bringing GIS operators to the end user.
When used in a simulation, the proposed model offers real-time performance in keeping track of the road network usage and can be quickly updated in order to respond to changes in the infrastructure.

8. Acknowledgment

This work was cofinanced from the European Social Fund through Sectorial Operational Programme Human Resources Development 2007-2013, project number POSDRU/107/1.5/S/77213 „Ph.D. for a career in interdisciplinary economic research at the European standards" (DOCCENT).

References

[1] Berg Insight AB Stockholm "GPS and Mobile Handsets – 4th edition, March 2010"
[2] Dirk Helbing "From microscopic to macroscopic traffic flow models", A Perspective Look at Nonlinear Media Lecture Notes in Physics, 1998, Volume 503/1998, 122-139, DOI: 10.1007/BFb0104959
[3] M. van den Berg, A. Hegyi, B. De Schutter, and J. Hellendoorn, "A macroscopic traffic flow model for integrated control of freeway and urban traffic networks," *Proceedings of the 42nd IEEE Conference on Decision and Control*, Maui, Hawaii, pp. 2774–2779, Dec. 2003
[4] W. Burghout, "Hybrid Microscopic–Mesoscopic Traffic Simulation", *Doctoral Dissertation,* Royal Institute of Technology, Stockholm, Sweden, 2004.
[5] Hibernate Spatial API - *http://www.hibernatespatial.org/hibernate-spatial-oracle/apidocs/index.html*
[6] OpenStreetMap Wiki – Data Primitives *http://wiki.openstreetmap.org/wiki/Data_Primitives*

Alex Alexandru SIROMASCENKO graduated from the Faculty of Cybernetics, Statistics and Economic Informatics of the Academy of Economic Studies in 2008 (Bachelor`s degree) and in 2010 (Master`s degree), specialising in Economic Informatics. He is currently a PhD candidate at the Academy of Economic Studies. His main domains of interest are road traffic optimisation, spatial databases and GIS technologies, data and business modelling, Web technologies and applications.