Australasian Transport Research Forum 2011 Proceedings
28 - 30 September 2011, Adelaide, Australia
Publication website: http://www.patrec.org/atrf.aspx

A Traffic Simulation Standard based on Data Marts

Marc Miska[1], Kugamoorthy Gajananan[2], Edward Chung[1], Helmut Prendinger[2]

[1]Smart Transport Research Centre, Faculty of Build Environment and Engineering, Queensland University of Technology, 2 George St GPO Box 2434, Brisbane QLD 4001, Australia

[2]National Institute of Informatics, The Graduate University for Advanced Studies, 2-1-2 Hitosubashi, Chiyoda-ku, Tokyo 101-8430, Japan

marc.miska@qut.edu.au

Abstract

Traffic Simulation models tend to have their own data input and output formats. In an effort to standardise the input for traffic simulations, we introduce in this paper a set of data marts that aim to serve as a common interface between the necessary data, stored in dedicated databases, and the software packages, that require the input in a certain format. The data marts are developed based on real world objects (e.g. roads, traffic lights, controllers) rather than abstract models and hence contain all necessary information that can be transformed by the importing software package to their needs. The paper contains a full description of the data marts for network coding, simulation results, and scenario management, which have been discussed with industry partners to ensure sustainability.

1. Introduction

Traffic Simulation models tend to have their own data input and output formats. Some can be read with other software (e.g. text files), while others are written in a binary format, only to be accessed from a certain software package. While this is fine for the software providers, it constrains the user and requires a huge workload of data preparation for any simulation study. Traffic operations and simulation is based on a lot of data that usually is warehoused by the road authorities. The amount of data in total is too big to be handled at any given time, but subsets of the data are needed. Such a subset of data from a data warehouse is called a data mart and represents the access layer to a data warehouse for users.

While it is unfeasible to design a data warehouse for road authorities, due to the differences in company and agency structures, it is possible to define data marts, from a simulation user perspective, as guideline for road authorities and operators to which data is necessary to perform cutting edge traffic simulations. Further, it is a first step towards a standardisation of inputs for traffic simulations.

In this paper we introduce a set of data marts that aim to serve as a common interface between the necessary data, stored in dedicated databases, and the software packages, that require the input in a certain format. The data marts are developed based on real world objects (e.g. roads, traffic lights, controllers) rather than abstract models and hence contain all necessary information that can be transformed by the importing software package to their needs.

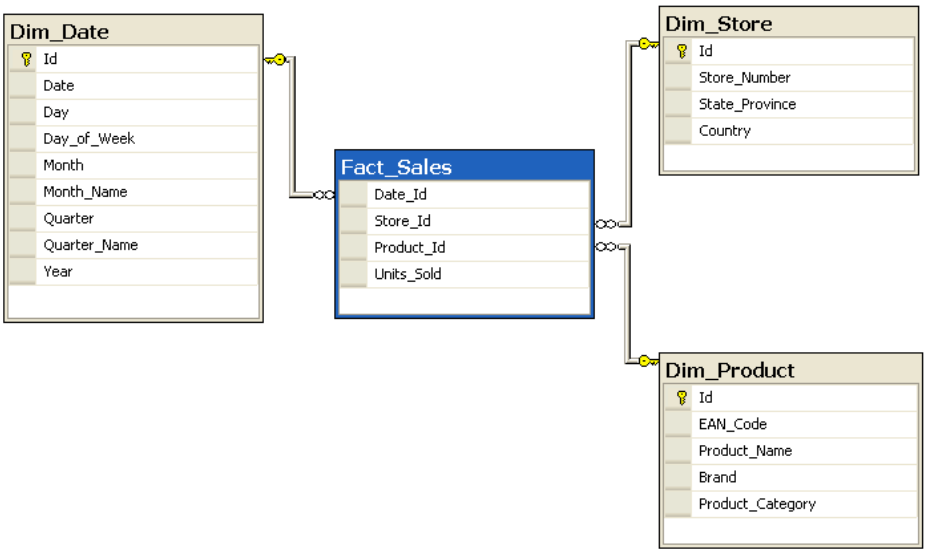
The paper contains a full description of the data marts for network coding, simulation results, and scenario management, which have been discussed with industry partners to ensure sustainability. Further we outline future activities on how those data marts can lead to an efficient and sustainable storage and exchange of traffic related information.

2. Data Warehouses and Data Marts

In order to standardize data analysis and enable simplified usage patterns, data warehouses are normally organized as problem-driven, small units, called “data marts”; each data mart is dedicated to the study of a specific problem. The data organization of a data mart, called a star schema, is very simple: the data being analysed, or facts, constitute the star’s centre; around the centre, other data describe the dimensions along which data analysis can be performed. In the archetypical case, facts are the sales of an organization, and dimensions enable the analysis by product, customer, point of sale, time of sale, and so on. In simple warehouses, data marts may extract their content directly from operational databases; in complex situations, the data warehouse architecture may be multilevel, and the data mart content may be loaded from intermediate repositories, often denoted as “operational data stores” (Bonifati, 2001).

To identify and build data marts the design should be driven by the purpose that each data mart is expected to address. As a consequence, the data mart design process must be based on a deep understanding of the expected usage. In a first step, user requirements are collected, and then translated into a star schema.

Figure 1: Example of a database star schema. A central fact table links records in dimension tables using unique identifiers to key related records.



The star schema consists of one or more fact tables referencing any number of dimension tables. In the following, we will gather the requirements for network description, simulation post processing and scenario management to define a data mart for each of them.

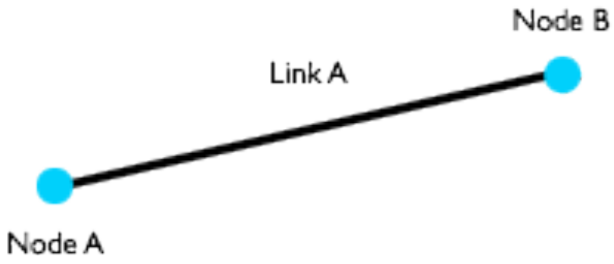
The proposed data marts aim to be guidelines for traffic and transport operation data warehouses, which are being developed around the world. However, warehousing methodologies are rapidly evolving but vary widely because the field of data warehousing is not very mature (Sen, 2005). These guidelines and the further development could prevent the creation of further incompatible data sources that burden international and national collaborations in research and practice.

3. Network coding

3.1 Links and Nodes

The transport infrastructure (i.e., roads, rail tracks) is described by a set of links and nodes. Nodes represent the start and end of the link that represents the transport infrastructure between (see Figure 2).

Figure 2: A Link, representing a road or track, defined by two Nodes

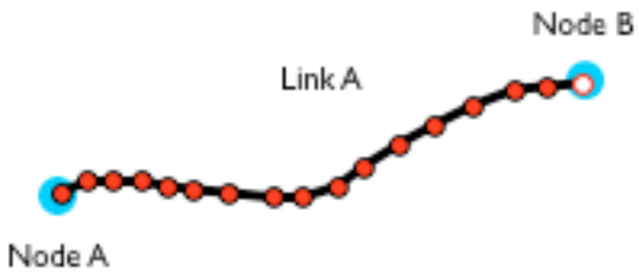


Nodes are the start or end of one Link, or connect two or more Links with each other. Diverging or merging (node connecting three or more links), as known from other network descriptions, requires the addition of an Area, which will be described later in the document. Nodes are described in 3-dimensional space by their coordinates (i.e., x, y, z) based on the World Geodetic System WGS84. Links are described by their start Node, end Node, and radius, and are in their basic form bidirectional.

3.2 WayPoints

Since links do not follow the same radius over a long stretch, WayPoints can be used to describe the curvature of the Link's centreline more detailed (see Figure 3).

Figure 3: A non-straight Link, defined by two Nodes and additional WayPoints

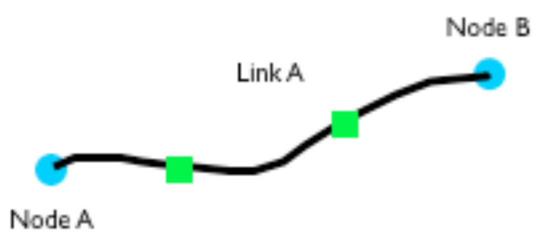


Links represent the centreline of the road's CrossSection (introduced in Level 2), so the geometry described in Level 1 has to represent the centreline of the full cross-section – not the road only. WayPoints, are describes, such as Nodes, in 3-dimensional space by their coordinates (i.e., x, y, z) based on the World Geodetic System WGS84, and the new radius of the link.

3.3 Markers

Along long Links, the CrossSection might change (e.g., additional lanes, bridges) at various points. To mark these locations, Markers are used (see Figure 4).

Figure 4: Markers, indicating changes in the cross-section of a link

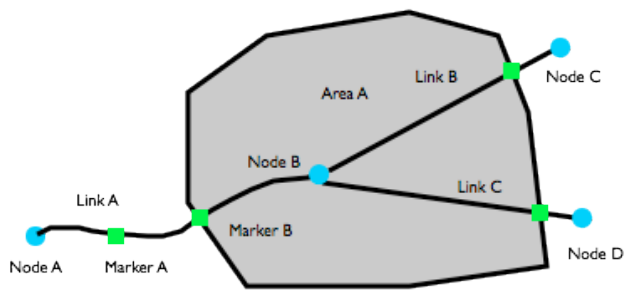


Markers are such as WayPoints an overlay of the link, and hence described in 3-dimensional space by their coordinates (i.e., x, y, z) based on the World Geodetic System WGS84, carrying the details of the new CrossSection. Additionally, Markers are used to indicate the boundary of an Area (see following section).

3.4 Areas

To describe intersections, merging and diverging more detailed, Areas are introduced. While Links are described as a cross-section along a centreline, Areas are described in all three dimensions. Areas consist of a Boundary that describes the border of the Area in space (see Figure 5).

Figure 5: An Area, used to connect three Links with each other

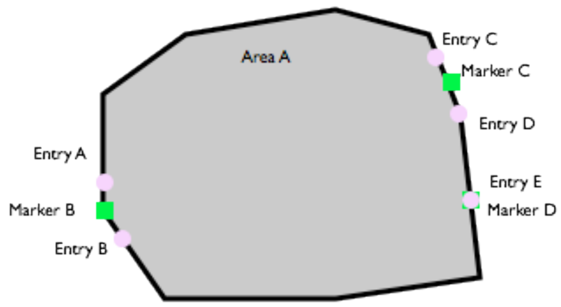


While the information would be sufficient for a macroscopic model, more detail is necessary to allow for microscopic modelling of the Area.

3.5 Entries

To allow for a detailed (i.e., microscopic) modelling of the Area, the first addition are Entries. Entries represent the access points to an Area in a unidirectional manner, hence Entries have to be defined for each direction of all Links entering or leaving the Area (see Figure 6).

Figure 6: An Area with Entries on the Boundary

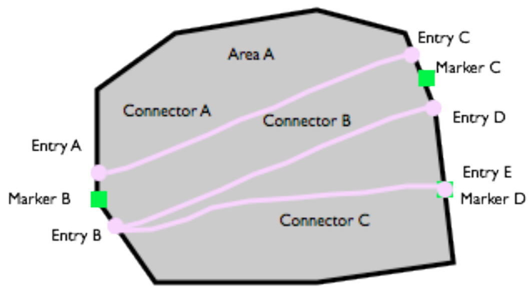


With the access point set, one can now define the movements inside the Area, if they are bounded, as they are usually for motor vehicles.

3.6 Connectors

To describe trajectories inside an area, Connectors are introduced. Connectors are described similar to Links, as centreline using WayPoints, but are unidirectional (see Figure 7).

Figure 7: Connectors describe the path vehicles follow in the Area

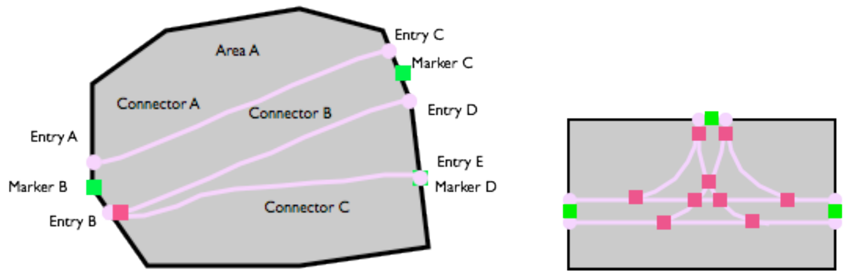


Connectors do not hold information on traffic rules or behaviour, but plainly describe the movement of a vehicle from an entry to an exit.

3.7 ConflictPoints

To allow for an easy identification of conflicts in an Area, ConflictPoints are used. ConflictPoints ensure the one-to-one matching of conflicting traffic streams without room for interpretation.

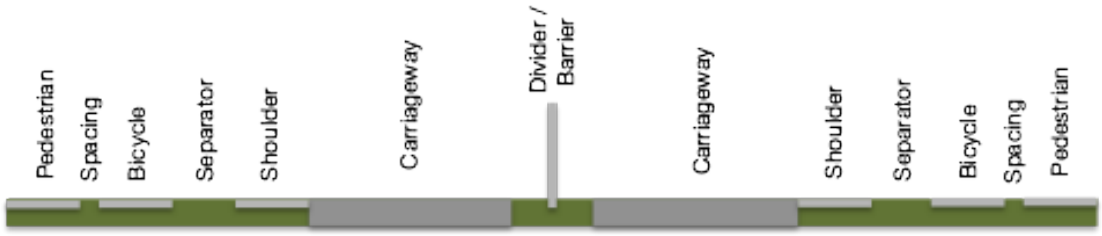
Figure 8: ConflictPoints at an on-ramp and T-junction



3.8 Cross Section

Links, described by their centreline already are described by their cross-section. The cross-section includes not only the carriageways, but also the whole traffic area, as shown in Figure 9.

Figure 9: ***Generalised cross-section of a roadway***

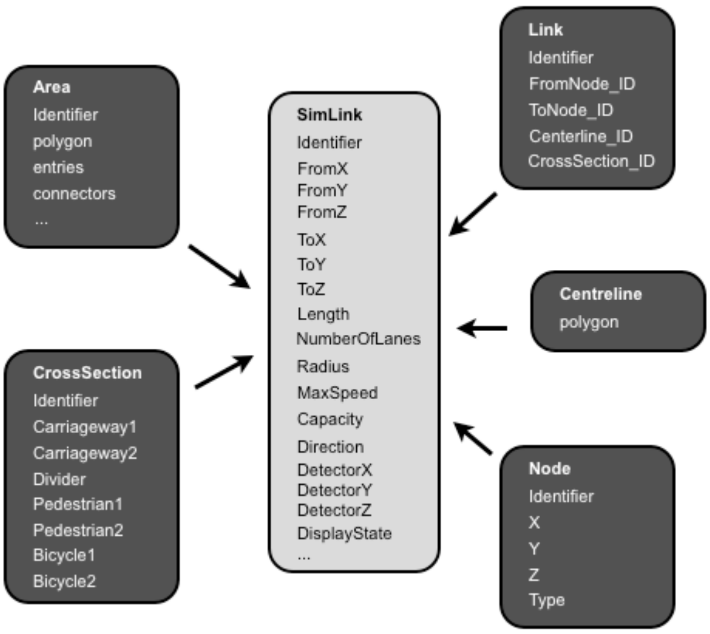


The benefit, of such a detailed description is that it contains enough information for various evaluations. The cross-section can change over time to allow for hard shoulder usage, or to integrate parking regulations into the simulation. Further, with more advances in driving psychology, this description contains sufficient information to feed perception modules of driver models.

3.9 Network Data Mart

Based on the specifications and requirements, a data mart for network processing should consist of simulation links that divide the network into harmonised links, with identical characteristics, such as their cross-section and driving rules. Figure 10 illustrates such a data mart.

Figure 10: ***Data mart that describes a transport networks through simulation links***



While the amount of data transferred with the data mart is bigger, the processing of the data mart is much quicker and unified.

4. Simulation Results

Simulation results describe traffic states along a timeline for further evaluation. Simulation packages allow the automated output of files containing specified information for post processing or to be used as feedback to control mechanisms of the simulation. Similar to the network coding, there is no standardised way of data output for simulation models. This limits the development of generic evaluation or visualisation tools, since the basic information of the input is not standardised.

4.1 Simulation elements

As an attempt to overcome this drawback, we propose a data mart that stores all dynamic information of the simulation, such as vehicle positions, vehicle dynamics, and detector and control states. A mart, used at the Smart Transport Research Centre, contains in its current version the following elements and categories:

- **Vehicle**
  - Position
  - Velocity
  - Acceleration
  - Vehicle Id
  - Link
  - Section
  - Lane
  - Emission
  - Lead Vehicle Id
  - Following Vehicle Id
  - Head Light Status
  - Brake Light Status

- **Traffic Light**
  - Position
  - Status
  - Link/ Node
  - Section
  - Lane

- **Display**
  - Position
  - Message
  - Link
  - Section
  - Lane

- **Sensor**
  - Position
  - Collection info
  - Current Reading
  - Link/Node

  - Section
  - Lane

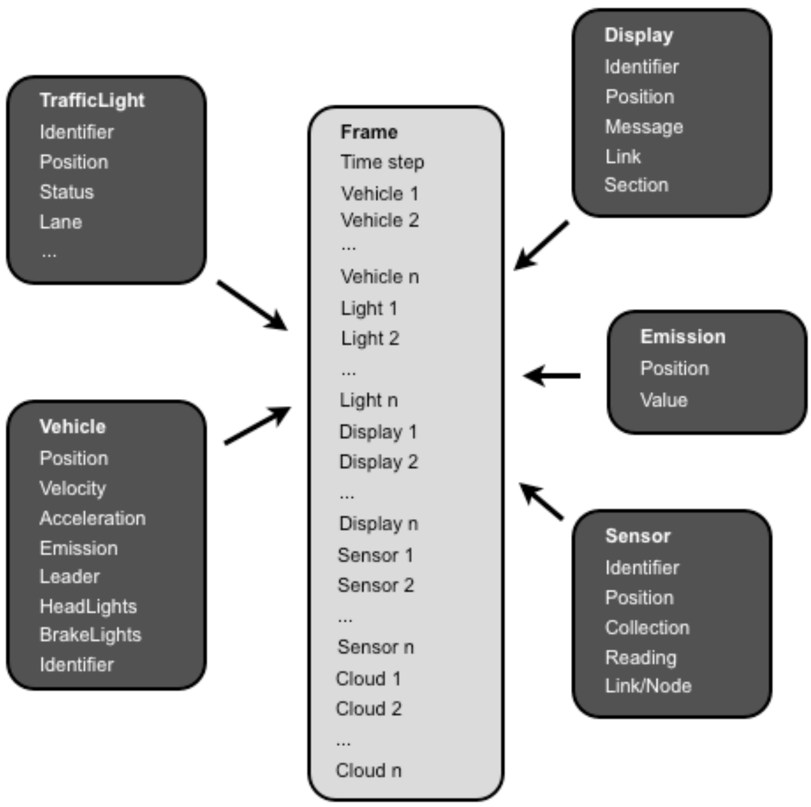
- **Emission Cloud**
  - Position
  - Connected sensors
  - Current Reading

This information, stored by time step, is the complete information needed to replay the whole simulation. While this information might be too much for some evaluation, it is necessary for others. By sending the output into a pipeline that can be read by evaluation tool according to their specific needs, the process allows for the independent development of simulation post processing tools.

4.2 Simulation Data Mart

Since the frame is constructed as a data mart, the form can be easily extracted. Figure 11 shows the frame with its feeding database tables.

Figure 11: ***Data mart that describes a simulation time step in detail***



The described mart is implemented in the OpenTraffic simulation suite, developed by the Smart Transport Research Centre in collaboration with the National Institute of Informatics in Tokyo, Japan.

5. Scenario Management

The word scenario in this paper is used for description of a driving task that can be forced on a particular car in the simulation with a given timeline (Gajananan, 2011). This allows the generation of driving situations to be used in driving simulator applications. While generating trajectories of vehicles for this is a very time consuming job, and simulation not always allows for the necessary mechanisms, the scenario management described here acts as automated commands given to a driver, rather than describing the exact trajectory of the car. The driving task is complex, and to achieve a desired behaviour, various control mechanisms are needed.

5.1 Obeying / violating traffic rules

Traffic scenarios for safety evaluation need to take into account that drivers do not obey all traffic rules. Usually, simulation allows for variations through driver characteristics such as aggressiveness. To generate a scenario for driving simulation, it is hardly enough to define the driver a driver as aggressive. One wants to define specific violations, to evaluate the effect. The identified actions are:

- Obeying/violating speed limits
- Obeying/violating traffic controls
- Obeying/violating lane boundaries

These actions used in a timeline description, allow describing specific driving rule violations.

5.2 Steering & Speed control

If more detailed control of a vehicle is needed, a script needs to dictate lateral and longitudinal movements of the script controlled vehicle. But instead of using a trajectory that describes a vehicles position over time by its x,y,z coordinates, the proposed controls are driving commands, which allow for more flexibility and generic scripts. The control commands are:

- Steer left/right
- Accelerate/decelerate
- Lane change left/right
- Turn left/right
- Merge left/right
- Exit on next ramp
- Overtake vehicle
- Pass vehicle
- Cut in between vehicles
- Emergency brake
- Engine failure
- Maintain given lateral and longitudinal gaps to obstacles and vehicles

These basic commands allow steering a driver according to a desired scenario and allow creating various situations, such as risky overtaking, incidents, drunk driving, and others.

5.3 Follow a vehicle (platooning)

Sometimes the necessary control of a scenario-controlled car depends on the actual movement of another vehicle. For instance, when simulating highway traffic with "truck-trains" (a platoon of trucks in which the driver controls the first truck and the others follow through computer control), one wants a driver controlling the first vehicle and the other to follow automatically. While one could achieve this by designing a new vehicle type, the straightforward way is to generate vehicles that automatically follow another vehicle.

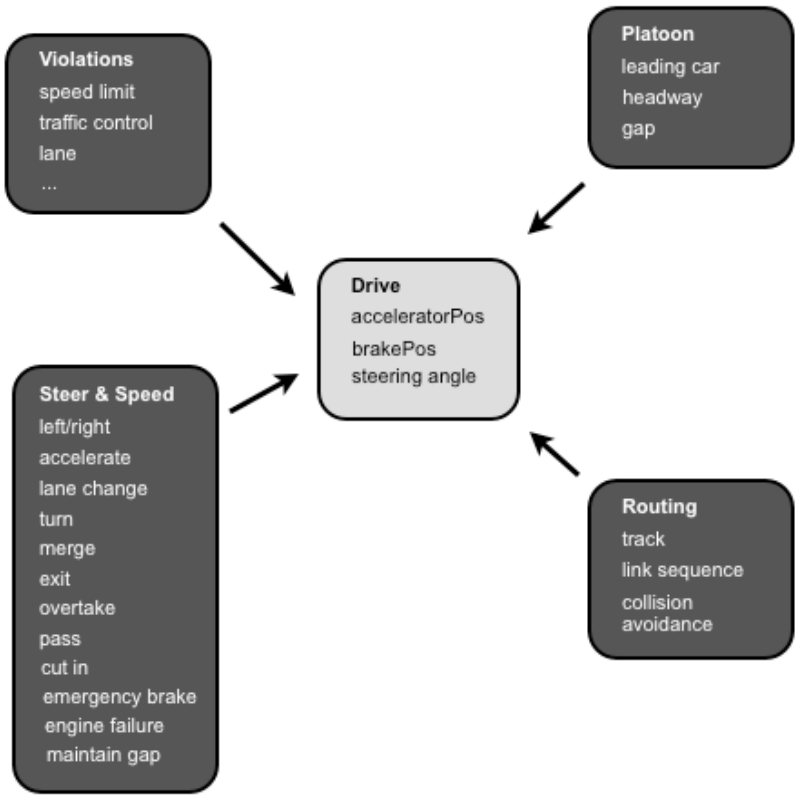
5.4 Follow defined route

Last but not least, the command to follow a route to a given destination. This route can be described as a trajectory, a sequence of links, or combination of links and lanes. Additionally, one can decide if the track is followed with respect to obstacles or if the route is followed blindly. The latter is still used in some driving simulators, which are logging a drivers action in simulated traffic, but without feedback of the drivers action to the simulator itself.

5.5 Scenario Data Mart

A scenario that consists of a timeline of commands to a driver can be described by a data mart that gives specific driving instructions to the driver in form of accelerator and brake position, as well as the steering angle.

Figure 12: ***Data mart that send commands to a driver***



The conversion of the commands to a accelerator position and steering angle requires an additional computation step, but that means that the data from the mart can seamlessly be integrated into the simulator.

6. Conclusion

In this paper contains we gave a description of data marts for network coding, simulation results, and scenario management, as an attempt to trigger standardisation of simulation model in and outputs. By following the ideas and concepts of business applications, transport could reach standards for data warehousing and data exchange via data marts, to ensure better collaboration and exchange of models and research data.

With data warehouse being in creation in many places around the world, but no effective standard for them, data marts could lead the way, and help users to convey their data needs to road authorities in an efficient way. This will, in the long run, lead to a more efficient data storage, and allows the research community to perform fast analysis through standardised data repositories.

References

Bonifati, A., Cattaneo, F., Fuggetta, A., Paraboschi, S., Designing Data Marts for Data Warehouses, ACM Transactions on Software Engineering and Methodology, Vol. 10, No. 4, October 2001

Sen, A. and Sinha, A.P., A Comparison of Data Warehousing Methodologies, Communications of the ACM March 2005/Vol. 48, No. 3

Gajananan, K., Prendinger, H., Nakasone, A., Doirado, E., Cuba, P., and Miska, M., Creating Interactive Driver Experiences with the Scenario Markup Language, International Conference on Entertainment Computing, 2011

Gajananan, K., Nakasone, A., Santos, E., Prendinger, H., and Miska, M., Scenario Authoring for Driver Behavioral Data Collection in 3D Virtual Environments, IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 2011