Queensland University of Technology
Brisbane Australia

This is the author's version of a work that was submitted/accepted for publication in the following source:

Miska, Marc, Gajananan, Kugamoorthy, Chung, Edward, & Predinger, Helmut (2011) A traffic simulation standard based on data marts. In Tisato, Peter, Oxlad, Lindsay, & Taylor, Michael (Eds.) *Proceedings of the Australasian Transport Research Forum 2011*, PATREC, Adelaide, S. Aust. , pp. 1-11.

This file was downloaded from: http://eprints.qut.edu.au/46314/

© Copyright 2011 University of South Australia and the South Australian Department for Transport, Energy and Infrastructure.

Notice: *Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source:*

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 framework based on ontology and data marts that aims 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 ontology describing 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 gives a background on the used technologies, describes simulation ontology, and gives a full description of the data marts.

1. Introduction

Mobility is and will remain a major issue for traffic engineers, road authorities and governments for the coming year. To test and evaluate new developments in dynamic traffic management, traffic simulation is used to refine and assess new constructions and control measures before applying them in the real world. Even for evaluating Social Problems, simulation can be a useful tool (Kuwahara, 2008).

The problem with simulation and simulation models from a research perspective is, that new developments, new theories, and new ideas often require a fundamental adjustment in parts of the simulation software. This need can for various reasons not being satisfied from commercial software providers, and so more and more simulation models are created to test specific ideas. The problem is, that while focusing on a specific area, other areas of the simulation tool are treated with less accuracy, which could compromise the results and lead to biased conclusions. Miska (2007) introduced a framework for online traffic simulation that tried to overcome such an issue by providing a stable and modular framework, in which each module could be updated or exchanged without affecting the simulation software as a whole. However, solution algorithms for specific behavioural models or control strategies are growing more and more complicated and require broad information throughout the network, so that the border between modules of a simulator grow more and more fuzzy.

Taking this into account we want to introduce a different approach for a flexible and easily adaptable simulation tool. This approach is based on ontology for traffic simulation, defining the objects and relationships between real-world objects to be used in modelling. The goal is to divide the physical representation and physical state change of objects in the model from the intelligence that is build to control and optimize these movements. The result we will create a toolbox that can represent any transportation network, based on today's technology, with basic simulation functionality. Movements and controls will follow a basic rule-set to ensure feasible and collision free traffic simulation. Such rules can then be overridden through outside algorithms that can have access to each individual object of the model and so can gather any necessary information. In this way, modellers can easily create networks out of standard objects, collect and reuse algorithms from the traffic community for traffic flow and traffic control, and at the same time test their own methods in a fully functional traffic simulation environment.

This paper illustrates the path chosen by the Smart Transport Research Centre, its partners and collaborators, to standardise simulation inputs, based on ontology and data mart oriented data storage.

2. Technology Background

2.2 Ontology

The term ontology comes from the field of philosophy that is concerned with the study of being or existence. In philosophy, one can talk about ontology as a theory of the nature of existence (e.g., Aristotle's ontology offers primitive categories, such as substance and quality, which were presumed to account for All That Is). In computer and information science, ontology is a technical term denoting an artefact that is designed for a purpose, which is to enable the modelling of knowledge about some domain, real or imagined (Gruber, 2009). The term had been adopted by early Artificial Intelligence (AI) researchers, who recognized the applicability of the work from mathematical logic (McCarthy, 1980) and argued that AI researchers could create new ontologies as computational models that enable certain kinds of automated reasoning (Hayes, 1985). In the 1980's the AI community came to use the term ontology to refer to both a theory of a modelled world and a component of knowledge systems. Some researchers, drawing inspiration from philosophical ontologies, viewed computational ontology as a kind of applied philosophy (Sowa, 1984). In the early 1990's, an effort to create interoperability standards identified a technology stack that called out the ontology layer as a standard component of knowledge systems (Neches et. al., 1991). Gruber (1995) defines ontology as an "explicit specification of a conceptualization," which is, in turn, "the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold among them." While the terms specification and conceptualization have caused much debate, the essential points of this definition of ontology are:

- Ontology defines (specifies) the concepts, relationships, and other distinctions that are relevant for modelling a domain.
- The specification takes the form of the definitions of representational vocabulary (classes, relations, and so forth), which provide meanings for the vocabulary and formal constraints on its coherent use.

One objection to this definition is that it is overly broad, allowing for a range of specifications from simple glossaries to logical theories couched in predicate calculus (Smith et. al., 2001). But this holds true for data models of any complexity; for example, a relational database of a single table and column is still an instance of the relational data model. Taking a more pragmatic view, one can say that ontology is a tool and product of engineering and thereby defined by its use. From this perspective, what matters is the use of ontologies to provide the representational machinery with which to instantiate domain models in knowledge bases, make queries to knowledge-based services, and represent the results of calling such services. For example, an API to a search service might offer no more than a textual glossary of terms with which to formulate queries, and this would act as ontology.

Ontology is often discussed in the applied context of software and database engineering, yet it has a theoretical grounding as well. Ontology specifies a vocabulary with which to make assertions, which may be inputs or outputs of knowledge agents (such as a software program). As an interface specification, the ontology provides a language for communicating with the agent. An agent supporting this interface is not required to use the terms of the ontology as an internal encoding of its knowledge. Nonetheless, the definitions and formal constraints of the ontology do put restrictions on what can be meaningfully stated in this language. In essence, committing to ontology (e.g. supporting an interface using the ontology's vocabulary) requires that statements that are asserted on inputs and outputs be logically consistent with the definitions and constraints of the ontology (Gruber, 1995). This is analogous to the requirement that rows of a database table (or insert statements in SQL) must be consistent with integrity constraints, which are stated declaratively and independently of internal data formats.

Similarly, while ontology must be formulated in some representation language, it is intended to be a semantic level specification - that is, it is independent of data modelling strategy or implementation. For instance, a conventional database model may represent the identity of individuals using a primary key that assigns a unique identifier to each individual. However, the primary key identifier is an artefact of the modelling process and does not denote something in the domain. Ontologies are typically formulated in languages, which are closer in expressive power to logical formalisms such as the predicate calculus. This allows the ontology designer to be able to state semantic constraints without forcing a particular encoding strategy. For example, in typical ontology formalisms one would be able to say that an individual was a member of class or has some attribute value without referring to any implementation patterns such as the use of primary key identifiers. Similarly, in ontology one might represent constraints that hold across relations in a simple declaration (A is a subclass of B), which might be encoded as a join on foreign keys in the relational model.

The heritage of computational ontology in philosophical ontology is a rich body of theory about how to make ontological distinctions in a systematic and coherent manner. For example, many of the insights of "formal ontology" motivated by understanding "the real world" can be applied when building computational ontologies for worlds of data (Guarino, 1995). When ontologies are encoded in standard formalisms, it is also possible to reuse large, previously designed ontologies motivated by systematic accounts of human knowledge or language (http://suo.ieee.org/). In this context, ontologies embody the results of academic research, and offer an operational method to put theory to practice.

2.2 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).

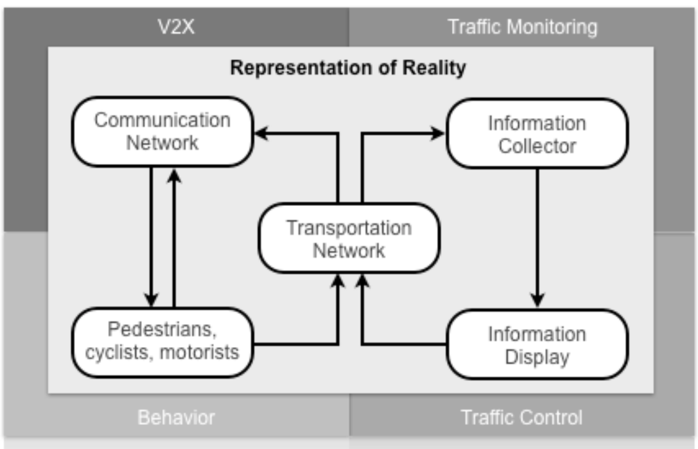
To identify and build data marts the design should be driven by the purpose that each data mart is expected to address. As a consequence, the data mart design process must be based on a deep understanding of the expected usage. In a first step, user requirements are collected, and then translated into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

Understanding the real world is what traffic simulation aims for, and looking for standard formalisms and to create reusable parts for further developments would be highly effective. Therefore, it would be beneficial to create an ontology for traffic simulation modeling. An ontology that might be not complete from the start, due to future developments, but that can be reused and extended for future developments.
If we take a look at traffic simulation modeling, than we can divide it in several areas. First we have the representation of reality, which includes the physical components of the real world. These components include the transportation network, including all information and control installations (e.g. traffic lights, traffic detection, road signs), the people using the transportation network, either by foot, some kind of vehicle or public transport, and as a rather new development, the communication network, that enables us to use the convenience of navigation systems, improved travel information from probe vehicle data, or the improved safety components from vehicle to infrastructure and vehicle to vehicle communication.
Underlying this representation of the real world, we have the mathematical models that aim to describe the behavior of network users, control the traffic by means of guidance and restrictions, and models that emulate the traffic monitoring system and inter-vehicle or vehicle to infrastructure communication to study its effect. Figure 1 gives an overview of the areas.

Figure 1: Areas of traffic simulation modelling



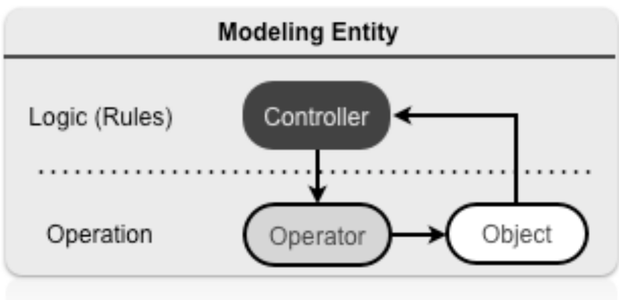
While the models keep changing and being extended according to new findings, the changes of the real world elements is less drastically. If we look at specific objects, such as pedestrians, cars, or traffic lights, we can find a hierarchy in their operation (see Figure 2).

Figure 2: Hierarchy of simulation tasks for pedestrians, vehicles and traffic signals

DTM	Supervision	Pedestrian simulation	Traffic simulation	Network control
DTM	Intelligence	route choice collision avoidance	route choice car following lane changing	vehicle actuation co-ordination
Basics	Logic (Rule based)	set speed set direction	set speed set direction	signal program
Basics	Operation	walk / stop	accelerate	set color
Basics	Object			

On the lowest level, pedestrians can just walk or stop walking or vehicles can accelerate or decelerate, while traffic lights just show one or a combination of colors. This fact has not changed since their invention. Changing the state of those objects, such as the color the traffic light is shown; early models use rule-based operations (e.g. fixed time control). This we classify as the basics of each object that we consider fixed. Every object can have certain states and the change of states is governed by simple rules (e.g. if there is a wall in front the pedestrian will stop or change directions).
Now, not all changes can be expressed in simple rules, since he circumstances are constantly changing and more and more factors have to be taken into account. This level of reaction we define as the intelligence level. An outside method is determining what an object is going to do based on different sources of information and calculations and not based on pure instinct. This is the area where the ontology stops and just provides and interfaces for users to attach their model to the simulation itself. Just to be complete, we have introduced a supervision layer that deals with synchronizing simulations of more than one object.
We will now continue to define the vocabulary to describe the basic layers of the real world representation. Therefore, the basic layers are combined to modeling entities, a bundle of the object with an operator that simply can change the states of an object, and a controller that will trigger such state change. Figure 3 shows the concept of a modeling entity.

Figure 3: Proposed concept of a modelling entity used for traffic simulation



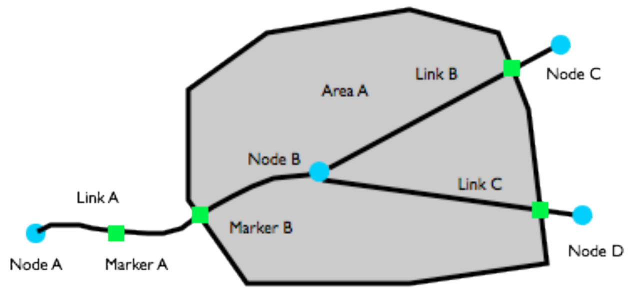
With the framework set, the ontology for all simulation elements needs to be developed to ensure that various simulation tools can run with the same inputs and produce transferable results. This standardization is the topic of the following sections.

3. Standardised Network Description

3.1 Network ontology

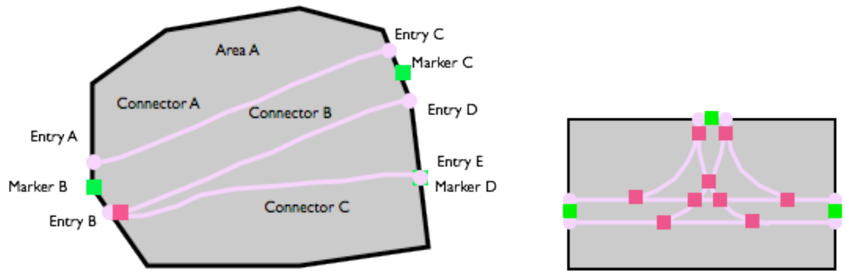
The transport infrastructure (i.e., roads, rail tracks) consists traditionally of nodes and links that carry parameters to be used by the simulation models to determine driving behaviour. Depending on the level of detail, from microscopic to macroscopic, the number of parameters required changes drastically. In our approach, we use ontology to describe a network, rather than building an abstract version of it. In a first layer, we split network elements in nodes, links, markers, and areas to replicate the geometry of the network (see Figure 4).

Figure 4: An Area, used to connect three Links with each other



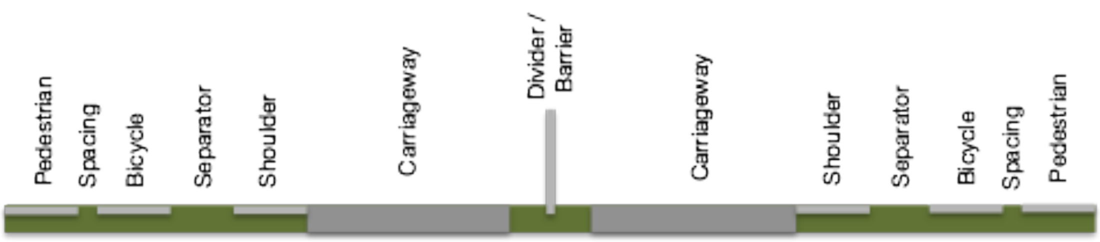
While such information would be sufficient for a macroscopic model, more detail is necessary to allow for microscopic modelling. Hence, we describe traffic areas in more detail to capture vehicle trajectories and conflict points as shown in Figure 5.

Figure 5: Additional behavioural data for traffic areas



Finally, links are being connected to a cross-section profile. The cross-section includes not only the carriageways, but also the whole traffic area, as shown in Figure 6.

Figure 6: ***Generalised cross-section of a roadway***

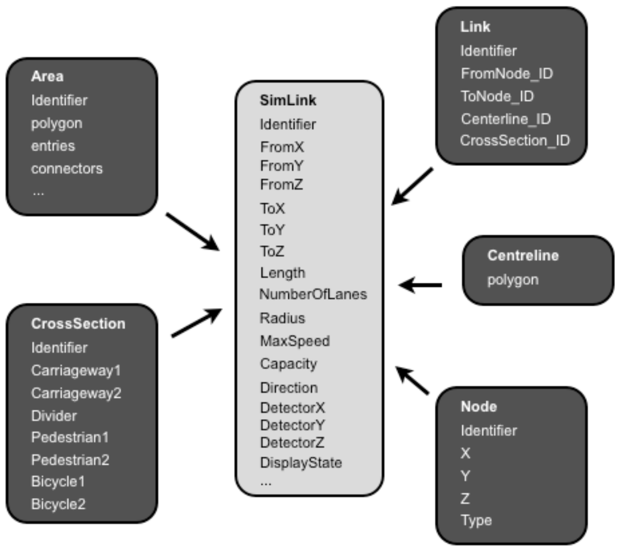


The benefit, of such a detailed description is that it contains enough information for various evaluations. The cross-section can change over time to allow for hard shoulder usage, or to integrate parking regulations into the simulation. Further, with more advances in driving psychology, this description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

3.2 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 7 illustrates such a data mart.

Figure 7: ***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. Standardised 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 ontology

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

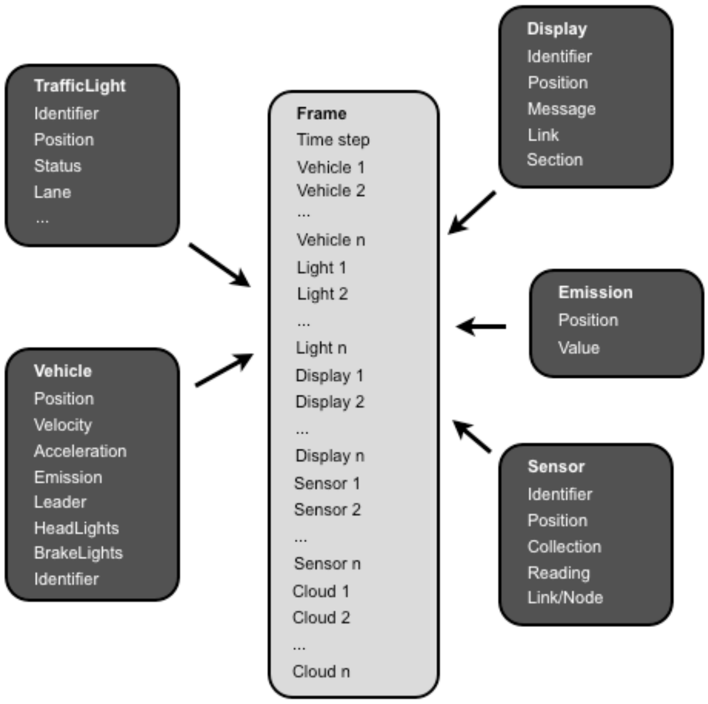
- **Emission Cloud**
  - Position
  - Connected sensors
  - Current Reading

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 8 shows the frame with its feeding database tables.

Figure 8: ***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. Standardised 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 Scenario Ontology

5.1.1 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.

5.1.2 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.

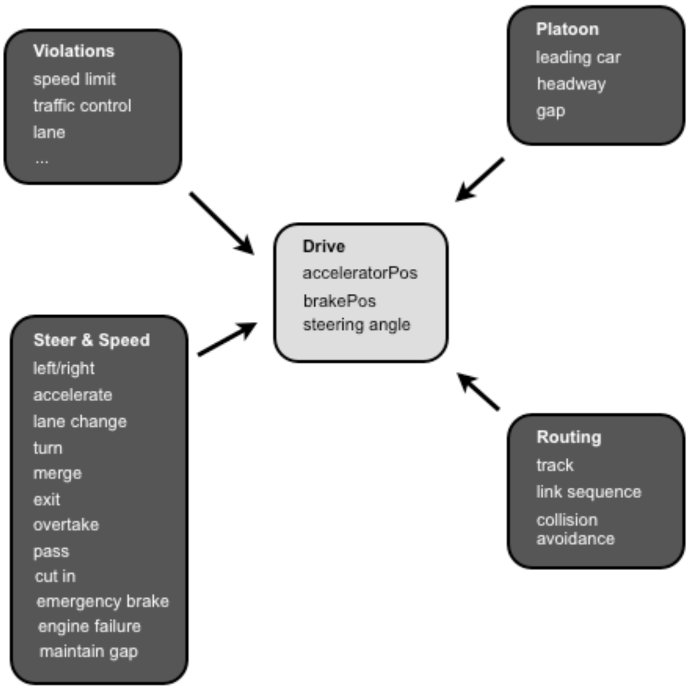
5.1.3 Following a vehicle (platooning)
Sometimes the necessary control of a scenario-controlled car depends on the actual movement of another vehicle. For instance, when simulating highway traffic with “truck-trains” (a platoon of trucks in which the driver controls the first truck and the others follow 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.

5.1.4 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.

5.2 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 9: ***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 we gave a description of framework of simulation ontology and resulting data marts that allows for a more efficient development of simulation tools due to standardisation and reuse of existing model, implemented within the framework. Data marts for network coding, simulation results, and scenario management, have been proposed 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 aroun

d 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

Chung, E., Ohtani, O., Warita, H., Kuwahara, M., Morita, H., (2006), Does Weather affect Highway Capacity?, 5th TRB International Symposium on Highway Capacity and Quality of Service

Chung, E., Ohtani, O., Kuwahara, M., (2005), Effect of Rainfall on Travel Time and Travel Demand ・5th ITS European Congress

Gruber, T. (2009), Ontology, in the Encyclopedia of Database Systems, Ling Liu and M. Tamer Özsu (Eds.), Springer

Gruber, T. R. (1995), Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal Human-Computer Studies, 43(5-6):907-928

Guarino, N. (1995), Formal Ontology, Conceptual Analysis and Knowledge Representation, International Journal of Human-Computer Studies, 43(5-6):625–640

Hayes, P. J. (1985), The Second Naive Physics Manifesto, in Hobbs and Moore (eds.), Formal Theories of the Common-Sense World, Norwood: Ablex

Kuwahara, M., (2008), Evaluating Social Problems using Transport Simulation, Workshop on Transportation Researches for Urban Safety, Asian Institute of Technology (AIT), Bangkok

McCarthy, J. (1980), Circumscription -- A Form of Non-Monotonic Reasoning, Artificial Intelligence, 5(13): 27-39

Miska, M., Real time Traffic Management by Microscopic Online Simulation, T2007/1, January 2007, TRAIL Thesis Series, The Netherlands

Neches, R., Fikes, R. E., Finin, T., Gruber, T. R., Patil, R., Senator, T., & Swartout, W. R. (1991), Enabling technology for knowledge sharing. AI Magazine, 12(3):16-36

Papageorgiou, M., Kosmatopoulos, E., Papamichail, I., Yibing Wang, (2008), A Misapplication of the Local Ramp Metering Strategy ALINEA, IEEE Transactions on Intelligent Transportation Systems, Volume 9, Issue 2, pages 360 - 365

Pushpakumara, T.D.C, Chung, E., El Faouzi, N.E., Kuwahara, M:, (2007), Effect of Rainfall on Travel Time and Accuracy of Travel Time Prediction with Rainfall・The 11th World Conference on Transport Research

Smith, B. and Welty, C. (2001), Ontology - towards a new synthesis. Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS2001)

Sowa, J. F. (1984), Conceptual Structures. Information Processing in Mind and Machine, Reading, MA: Addison Wesley