Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

STRATEGIC BEHAVIOR IN A LIVING ENVIRONMENT

Marco Lützenberger
Sebastian Ahrndt

DAI-Labor
Technische Universität Berlin
Ernst-Reuter-Platz 7
10587 Berlin, GERMANY

Benjamin Hirsch

EBTIC
Khalifa University
Abu Dhabi Campus
Abu Dhabi, UAE

Nils Masuch
Axel Heßler
Sahin Albayrak

DAI-Labor
Technische Universität Berlin
Ernst-Reuter-Platz 7
10587 Berlin, GERMANY

ABSTRACT

When it comes to road traffic, there seems to be no parameter more essential than the driver himself. His internal preferences, his attitude and his perception determine the traffic situation in a crucial way. Yet, as a matter of fact, only a small number of existing traffic simulation frameworks provide a model for the driver's behavior. These existing models usually focus on tactical aspects, and neglect decisions with a strategic touch. Recently, we have introduced a driver model which is able to incorporate strategic decisions. While this model already takes infrastructural installations into account, we refine our so far presented work in this paper and extend it to account for additional criteria, the so called Regional Conditions.

1 INTRODUCTION

Despite the wide range of available traffic simulation frameworks, most products share the fact that the vehicle simulation is done in a static fashion. As it is, reality is much more complex, because what actually happens on the road is not only determined by physics of motion, but also by the perception and attitudes of the drivers and external conditions. A driver with a high affinity for public transport for instance might change his means of transportation when confronted with extreme weather conditions near a subway station and available parking. This aspect does not affect the driving process per se, but influences the traffic situation a fortiori. Several approaches integrate stimuli–reaction principles and mimic individual driving styles by implementing cognitive abilities for the simulated drivers. Yet, a more comprehensive, "strategic" consideration is mostly missing.

In this paper we describe a model (see Section 3) which incorporates strategic reactions of the simulated drivers to influences of the surrounding infrastructure. Since we already provided several refinements for our drivers (Lützenberger et al. 2011, Lützenberger et al. 2011a, Lützenberger et al. 2011b), the focus of this work lies on the formalization of our infrastructure model. We start with a short and general consideration of influences on drivers (see Section 2) from which we derive a formal meta–model specification. Subsequently, we give a short introduction of our driver model and define an exemplary park–and–ride scenario (see

978-1-4577-2109-0/11/$26.00 ©2011 IEEE

Section 4). We demonstrate the functionality of our approach by presenting the simulation results of this example. We proceed by comparing our approach to related work (see Section 5) and show how our work differs from other approaches. Finally, we wrap up with a conclusion in Section 6.

2 CONCEPT

Based on a survey of traffic simulation frameworks (see Section 5), we can say that most frameworks work in a rather static way. Vehicles start at a dedicated position and proceed either on a way- or time-optimized route to a dedicated target position. We are not entirely satisfied with this principle, since there are many influences which are able to influence a driver in his initially selected strategy to reach a target location. As an example, consider the sudden occurrence of freezing rain in close distance to a subway station with parking capabilities. Although the driver has a clear objective, his strategy is pretty much subject to his internal state and his perception. While impassable road conditions and sufficient parking capabilities may even influence established drivers, a slight chill and a crowded car park may only convince careful individuals to take the train.

The bottom line is that in a realistic traffic situation, there are many stimuli able to influence a driver's decisions and traffic conditions, respectively. For this reason we seek to integrate those for traffic simulations. To do this, we first have to examine a stimulus' main characteristics. We use the example above to emphasize our approach.

2.1 Stimuli — A General Approach

The example above involves three different stimuli: The freezing rain, the subway station and a car park. The first element we can observe is a **location** attribute. While both, the subway station and the car park are immovably located at a distinct position, the freezing rain covers an area. In our model, we have to account for this aspect.

The next thing we can observe is that each stimulus has some kind of **effect** on the driver. While the freezing rain directly affects the driver in many aspects (velocity, distance to other vehicles), both, the subway station and the car park, influence the driver on another level. They can be considered as alternative options which are able to support a driver in achieving his goal. For these stimuli, we use the term *Infrastructural Feature*, which we defined as follows:

> *"An Infrastructural Feature can be everything which is able to fulfill a desire (or parts of it) of a person at a certain location of an infrastructure."* (Lützenberger et al. 2011a)

Despite the different nature of *Infrastructural Features* and the freezing rain, we can use the "effect-metaphor" in both cases. The only difference is that the freezing rain affects the driver without his agreement, while the effects of *Infrastructural Features* define an estimated target state. This estimation can be used by the drivers in order to determine if the alternative option is viable or not. In fact, we already described a model for *Infrastructural Features* (Lützenberger et al. 2011b), which we now refine and extend by so called *Regional Conditions*, such as the freezing rain. Before we proceed we provide a definition for the term: *Regional Condition*.

> *A Regional Condition can be everything which is able to affect or influence a person, its behavior, or its vehicle (physically) at a certain location of an infrastructure.*

The next thing we have to consider is a **precondition**. For the freezing rain, the situation is simple, since there is no precondition to sense uncomfortable weather. *Infrastructural Features* on the other hand require such specification. To provide its service, the car park has to feature at least one vacant parking lot. Further, the driver has to be in possession of a vehicle. The subway station however requires the driver to be on foot.

Finally, each stimulus is associated with some kind of duration which describes the required time for the stimulus' effects to affect the driver.

2.2 Stimuli — A Formal Approach

In our opinion, external influences on drivers can be described by the four characteristics we emphasized above. In the following, we introduce a formal model for their specification.

$\mathbb{L}$ocations: Traffic simulation frameworks move vehicles either on existing– or fictitious maps. For realistic results, we can assume that these maps are based on geographic coordinates and can be represented as *GPS Exchange Format* (*GPX*, http://www.topografix.com) data. To support not only distinct positions, but also "areas of influence", we extend *GPX* by an additional attribute which expresses the range of the respective stimulus. In the case of the freezing rain, the *GPX* coordinate can be interpreted as the rain's epicenter, while the range attribute can be understood as its radius. We define a stimulus' location l as follows:

$$l = (x,y), x \in GPX, y \in \mathbb{R}$$

We refer to the set of locations as $\mathbb{L}$.

$\mathbb{P}$reconditions: For the specification of preconditions use first order logic. A single precondition can thus be considered as predicate which either becomes true or false. Since most stimuli require more than one precondition, we allow for this as well and define the preconditions p of a stimulus as follows:

$$s.getPreconditions() := p = \{p_1, p_2, \ldots, p_n \mid p_i \subseteq domain(p_i) \times boolean\}$$

We refer to the set of preconditions as $\mathbb{P}$. We further define that a driver d is able to make use of the stimulus s if the conjunction of its preconditions is satisfied:

$$d.canUse(s) := p_1 \ \wedge \ p_2 \ \wedge \ldots \wedge p_n, \ p_i \in s.getPreconditions()$$

$\mathbb{E}$ffects: Effects define some kind of target state which applies after the stimulus has affected the driver. In case of the *Infrastructural Features*, these effects can be used by the agents to measure if it helps reaching the the potential target state and whether to make use of it or not. In case of the regional conditions, the stimulus affects the driver without his permission. Currently we provide an abbreviated model, since the effects of our *Regional Conditions* are focused on the drivers and disregard those of the vehicle (stopping distance, consumption, road grip, etc.). We aspire an extension for the near future (see Section 6.2) and currently define effects e as follows:

$$e = \{f \ \mid f : domain(x) \rightarrow domain(x), x \in d.getAttributes() \vee x \in s.getAttributes()\}$$

We refer to the set of effects as $\mathbb{E}$.

$\mathbb{D}$uration: Finally, we define a duration method for each stimulus, which is used by the simulation engine to determine the time t after which the effects of the stimulus occur. For *Regional Conditions*, this function is quite simple, since their effects occur immediately. For *Infrastructural Features*, this function can be highly complex and involve many parameters. For the duration *dur*, we assume a specification in compliance with:

$$dur(x_1, x_2, \ldots, x_n) : domain(x_1) \times domain(x_2) \times \ldots \times domain(x_n) \rightarrow t \in Time$$

We refer to the set of duration functions as $\mathbb{D}$.

So far, our definition suits well for many purposes, including the car park and the freezing rain. Yet, more complex systems can not be captured. The subway service for instance provides access at many distinct entrances, while the entire system is somehow interconnected. For this reason, we extend our model from single- occurrences to complex systems.

The $\mathbb{S}$timuli System: We define a *Stimuli System* as follows:

$$\mathbb{S} := \{(l,p,e,dur) \mid l \in \mathbb{L}, p \in \mathbb{P}, e \in \mathbb{E}, dur \in \mathbb{D}\} \tag{1}$$

Based on the given definition, it is possible to include environmental influences into simulation topologies. In the following, we will introduce a driver model which is able to deal with these influences.

2.3 The Driver Model

To react to influences as those defined above, our drivers have to constantly perceive their environment. They also have to possess knowledge about this environment and have a set of internal preferences and attributes. Further, each driver is committed to an ultimate goal to reach a certain target location. According to *Wooldridge* and *Jennings* (Wooldridge and Jennings 1995) we can understand the driver as an agent, and for our implementation we used that exact view. For the decision model, we made use of the *Belief-Desire-Intention* (*BDI*) paradigm (Rao and Georgeff 1995), which allows for a formal conceptualization of human behavior. The operation principle and behavior phases of our *BDI* agents are illustrated in Figure 1.

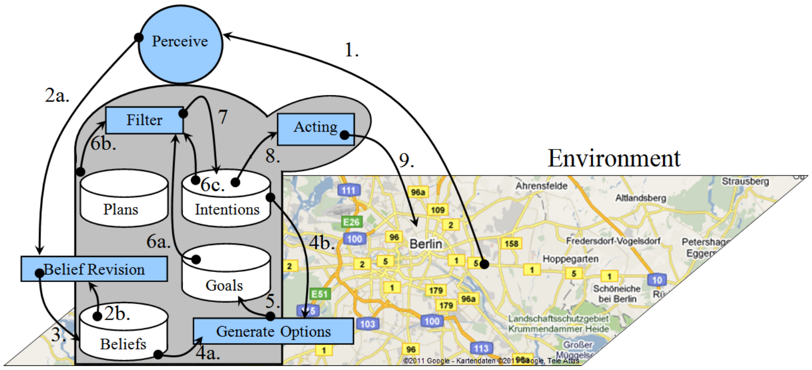


Figure 1: The architecture and actuation principle of our driver agents.

By default, an agent features *walk* and *drive* as basic capabilities (or, using the *BDI* terminology: *Plans*). Both plans are bundled into one so called *Plan Object* respectively, which contains additional information on the plan's preconditions, its effects and a function to calculate the duration for an intended trip. Further, each agent possesses an attribute which determines his current range of sight.

The agents' actuation comprises four phases. By accessing the drivers' positions and the *Regional Conditions*' location and range, the simulation engine determines if drivers are affected. In this case, the attributes of the drivers are altered according to the respective conditions' effects. *Regional Conditions* have to be evaluated first, since they directly affect a driver's state and attributes and may lead to different decisions in the subsequent phases. The freezing rain for instance may increase a driver's affinity for public transport and result in a lower cruising speed. Fog, as another example may reduce a driver's range of sight and result in *Infrastructural Features* to be recognized later than usual.

After the effects of *Regional Conditions* have been applied on each agent, the simulation engine uses the drivers' position and his range of sight to determine if *Infrastructural Features* are perceived (1.). The perception is then forwarded to the *Belief Revision*, where the agent updates his belief base (3.) by merging his current perception (2a.) with stored knowledge (2b.). Using his updated belief base (4a.) and his former intentions (4b.), the agent proceeds with the *Generate Options* phase, where preconditions of all

Infrastructural Features in the belief base are evaluated by means of the agent's `canUse(s)` method. For each positive evaluation, the sub-goal to utilize the *Infrastructural Features* is stored within the goal base of the agent (5.). In combination with the agent's basic plans (*walk* and *drive*) and his current intentions, the new set of goals constitutes the input (6a., 6b., 6c.) for the *Filter* phase. We distinguish between two types of goals. While the ultimate goal expresses the agent's overall objective to reach a certain location, only sub-goals can emerge dynamically indicating an agent's desire to utilize an *Infrastructural Feature*. By accessing the *Infrastructural Features*' effects and by making use of his basic plans, the agent computes alternative strategies to his target location involving any possible permutation of *Infrastructural Feature* utilizations. The resulting strategies are evaluated according to the agent's individual preferences (most easily by computing the required time) and finally, the favorite strategy is selected and inserted into the agent's intention repository (7.), from which his actuation is derived (8.) and his environment influenced (9.) once more.

For the reason of space, the driver model has only been outlined in this paper. A comprehensive introduction has been done in previous works (Lützenberger et al. 2011a, Lützenberger et al. 2011b).

By now, we presented an approach which comprehends strategic reactions of simulated drivers on influences of the surrounding infrastructure. In the following section, we advance from theory to practice.

3 YET ANOTHER THEORY?

With the described model for driver and topology, it is possible to include external influences for computer aided traffic simulations. The nature of these influences however is usually not focused on selected streets or intersections, but comprises at least urban dimensions. This aspect entails the problem of composing these scenarios, since, without a concrete approach for generating scenarios in urban dimension, we have just described "yet another" impractical theory.

In order to support and facilitate the automatic generation of simulation scenarios, we consciously developed a formal specification (1). Our approach supports large scale scenarios since for each type of influence, only one precondition– and one effect specification is required. The effort to include one single or one thousand car parks remains the same. Yet, we still have to deal with the distribution issue.

For large scale scenarios, the location attribute constitutes a serious problem. The effort to define preconditions and effects for one and for one thousand car parks may be the same, yet, the effort for their distribution grows linear with the amount. To solve this problem, we apply so called "semantic maps" for our simulation. In addition to *GPS* coordinates of traffic relevant objects, semantic maps feature an additional layer which enriches traffic objects by attributes. Semantic maps allow for a selective identification of places, such as restaurants or fuel stations or even special tracks, such as hike-, cycle- or even seaways.

The *OpenStreetmap Framework* (*OSM*) (Ramm et al. 2010) provides semantic features. *Open StreetMap* is an open project, which collects geographic data and makes them publicly available. On the lowest hierarchy level, the data consists of *GPS* coordinates. On top of that, the data is grouped to collections, which reflect complex entities such as streets, bike paths, subway stations, bus stops, and many more. These complex entities are extended by tags, which carry the semantic enrichment. Additional information such as a street name, amount of lanes, or a subway station's lines are stored here.

For our work, we have developed a graphical editor. This editor can import an *OSM* map and automatically locate *Infrastructural Features* for which preconditions and effects have been defined. To provide a high degree of flexibility, the initially loaded map can be manipulated by adding, removing or configuring selected *Infrastructural Feature* instances. Further, the map can be extended by previously defined *Regional Conditions*.

3.1 Infrastructure Features

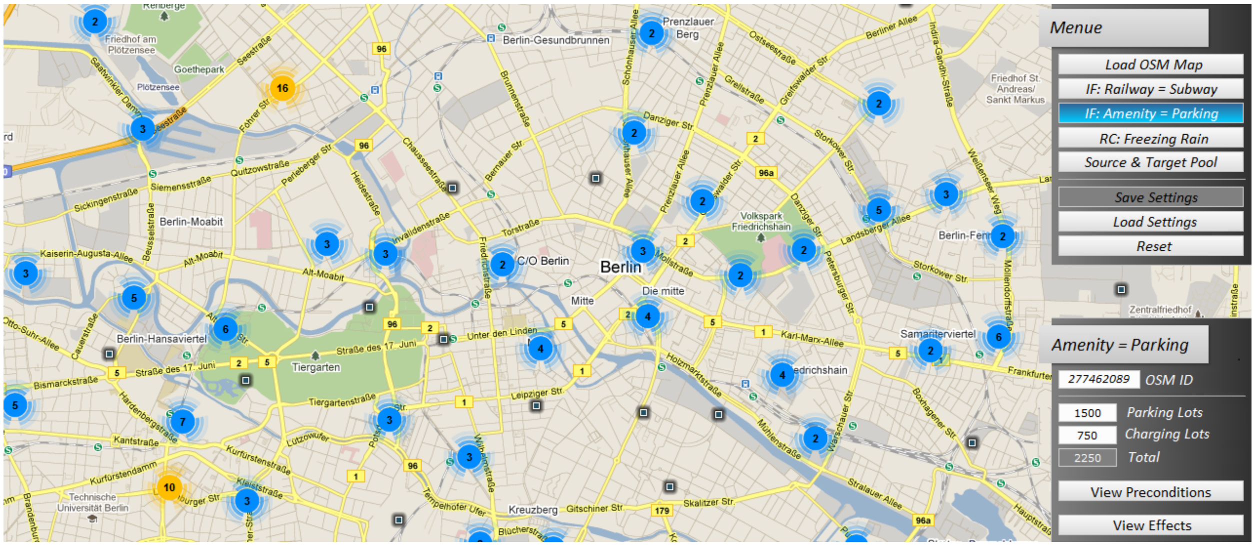
Our editor is able to import an *OSM* map and to retrieve particular points of interest from this map. The implementation of this mechanism was easy, since *OSM* simply extends *GPS* coordinates with key-value tags. Car parks for instance are tagged by the key-value combination: `amenity=parking`. Subway stations can be identified by the key-value combination: `railway=subway`. Currently, the editor does not support the entire key-value scope of *OSM*, but an extension can be done quickly by an addition to the editors' list of known key-value pairs.

Using our editor, the locations of *Infrastructural Features* can be derived automatically, but to comply with our model, we further have to provide specifications for preconditions and effects. We define those in separate files, which are loaded by the editor on startup. For the binding we use the *OSM* key-value pair as identifier.

After the *OSM* map and the provided preconditions and effects have been loaded, the user can choose which *Infrastructural Features* should be considered for his simulation scenario. The initial distribution of his selection is then presented on a map. Since neither the editor, nor the simulation engine require a particular map, we can perform simulations for any city of the world (as long as there is an *OSM* map).

The general principle works fine, although we often had problems with its flexibility to modify selected parameters. As an example consider the capacity of car parks. The capacity usually varies from instance to instance and can not be defined as global constant. *OSM* provides a large amount of additional information (in some cases even for parking capacities), yet, the framework is not all-powerful and for this reason we implemented the editor to present detailed information on the attributes of each single *Infrastructural Feature* and to allow for selective manipulations and extensions of those.

Figure 2 illustrates our editor, showing original car park occurrences for the capital region of *Berlin, Germany*. Spots with multiple occurrences are represented as colored circles, and labeled with the exact number of contained car parks. Zooming into these spots will cause the circles to collapse into black spots identifying single car parks.



277462089 OSM ID

1500 Parking Lots
750 Charging Lots
2250 Total

Figure 2: The *OSM* editor, showing original car park occurrences for the capital region of *Berlin, Germany*.

3.2 Regional Conditions

For the precondition and effects, we applied the same mechanism as with the *Infrastructural Features*. Since the volatile nature of *Regional Conditions* disagrees with persistent storage (at least within an *OSM* map), location and ranges can be manually defined through our editor. Nevertheless, our approach comprehends more sophisticated mechanisms. As an example, it is possible to use authentic weather data since those localize particular weather conditions by means of *GPS* coordinates. We consider an according feature for the near future (see Section 6.2).

To clarify our concept, and evaluate our work we used our editor to develop an expressive simulation scenario. We describe this scenario in the next section.

4 EVALUATION

In the following, we describe our implementation of the freezing rain scenario. We start by explaining the preconditions, effects and duration functions of the involved stimuli and proceed with required adjustments for the drivers. Subsequently, we explain how we assign our stimuli to locations, and finally we illustrate the collected results of our experiment.

4.1 The Subway Service

In order to make use of the subway service, we have to make sure that the driver d is currently not located in a vehicle. We define the following precondition:

$$d.getVehicle() = \emptyset$$

Although we are dealing with a complex and cross-linked system, we apply a rather simple implementation for its effects. Later we select a map with four subway stations and while the different instances are located at different positions, each station moves the driver to a universal target position $p \in GPX$. We define the effects as follows:

$$d.getPosition() = p$$

For the duration method, we apply *Newton's* law of motion and define *dur* to return the required time for the beeline distance between the instances' location and the universal target position. For this, we assume an average velocity of $34.52\frac{km}{h}$ (or $21.45\frac{m}{h}$), which we derived from the schedule of *Berliner Verkehrsbetriebe* (http://www.bvg.de), *Berlin*'s transportation company.

$$dur(x) := \frac{beeline(x,p)}{34.52\frac{km}{h}}$$

4.2 The Car Parks

For the car parks, we assume that each driver who wants to make use of a car park instance c, is riding a vehicle and that the instance provides enough capacity.

$$d.getVehicle() \neq \emptyset$$

$$c.getCurrentCapacity() > 0$$

For the effects, we define that the driver is no longer within his vehicle and that the capacity before the utilization *c_Before* is decreased by one.

$$d.getVehicle() = \emptyset$$

$$c.getCurrentCapacity() = c\_Before - 1$$

Finally, we define the duration method of the car park to constantly return one minute.

4.3 The Freezing Rain

Except for a location, the only thing *Regional Conditions* require is an effect specification. We design the effect to increase the driver's subway acceptance rate in dependency to his distance to the freezing rain's epicenter l.

$$e = \{f : domain(d.subwayAcceptance) \rightarrow domain(d.subwayAcceptance)\}$$

with:

$$f(x) := \begin{cases} 6 * x; beeline(d.getCurrentPosition(), l) < 100m \\ 4 * x; beeline(d.getCurrentPosition(), l) < 200m \\ 2 * x; else \end{cases}$$

4.4 The Driver

We extend the drivers by the attribute `d.subwayAcceptance` – their acceptance for the subway service. We initially assign an acceptance of 15% (14.58% to be accurate) which we derived from a comprehensive mobility study (Federal Ministry of Transport, Building and Urban Development 2010), which has been done by the *German Ministry of Transport, Building and Urban Development*. We also manipulate the filter module of our agents to perform an evaluation of the proposed strategies by accumulating the regular durations for any walk, drive and parking process and artificially lengthen the required time for using the subway service inversely proportional to the driver's acceptance towards the service. The higher a driver's acceptance for the subway, the lower the chance for artificially increased costs and the higher the probability for using the subway.

4.5 The Setup

We chose *Berlin* as simulation environment and extracted a fitting map fragment for our example (we selected a small map for illustrative reasons and not for any storage or computing limitations). We then automatically loaded car parks from the *OSM* map and respectively assigned a capacity of 3.000 parking lot. We also loaded subway stations from the underlying map and assigned the previously defined preconditions and effects. We specified a total amount of 10.000 vehicles for each simulation run and defined an area of potential start- and target locations. With the selected arrangement for these start- and target pools we made sure the vehicles pass the scope of our influences. The simulation setup is illustrated in Figure 3.

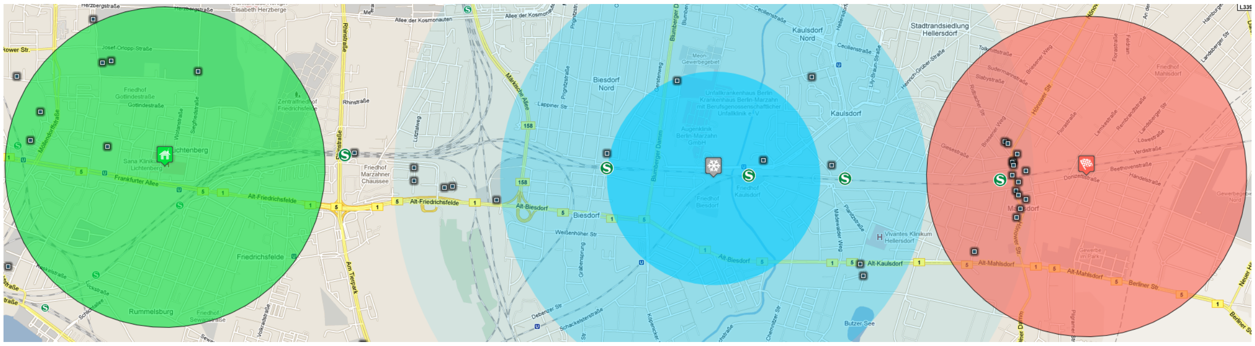


Figure 3: The evaluation scenario, including four subway stations, one car park for each subway station and several others in the target area, one zone with freezing rain (middle, light blue), as well as start- and target areas (left, green and right, red) of the vehicles.

In total, we performed two simulations. We performed the first simulation on the initial topology and for the second simulation, we additionally integrated freezing rain with a radius of 3000m between the start- and target area. At the beginning, the drivers are located within a parked vehicle at some random place within the start area. During runtime, the simulation engine randomly generates (superior) goals to reach a location from the target pool and places them in the goal base of a driver. The agent now starts his journey. Since no other option is available, the agent computes and executes a strategy involving his *drive* plan. Once a driver perceives a stimulus, further concurrent strategies will be proposed. In the case of a car park, the driver computes a strategy to park his car and walk to the target location. Usually this option will fail because we apply a strategy selection which is based on the required time to the target and walking strategies tend to be highly expensive. In the case of a subway station, any evaluation on making use of the service will fail, since we demand the driver to be on foot. Only when the driver sense both stimuli, the *Filter* module will be able to compute a valid strategy, involving a ride to the car park, a walk to the subway station and a walk to the target location. Depending on the driver's acceptance for the subway service –which possibly has been altered in case the driver is affected by the freezing rain– this optional strategy either replaces the original one or is rejected. We illustrate the results of both simulation runs in Figure 4.

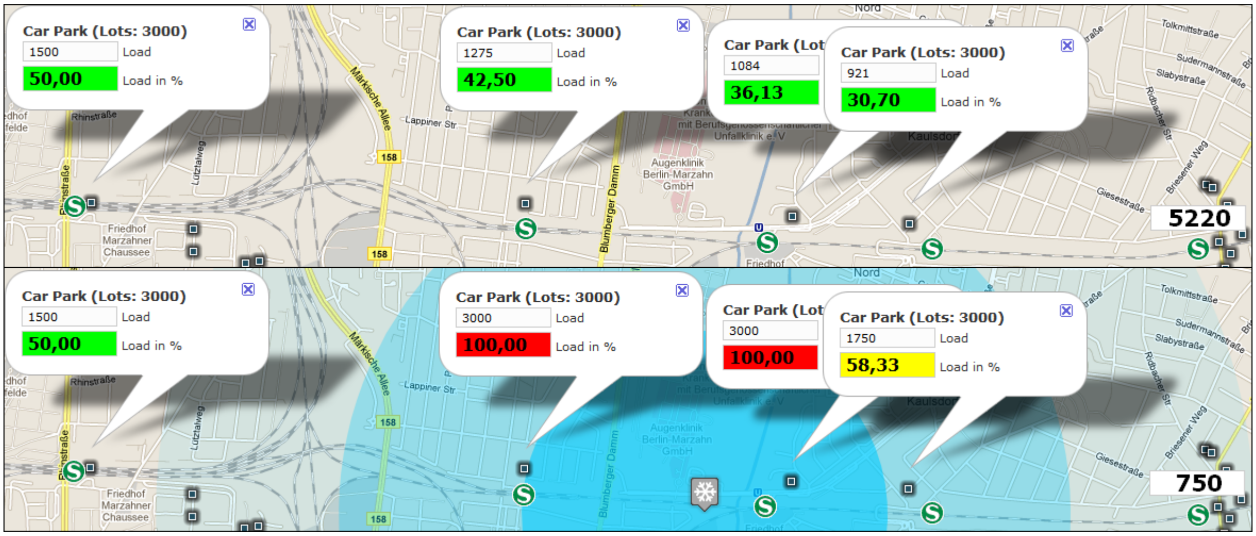


Figure 4: Results of the two simulations, showing the car parks' utilization in absolute and percentage values.

The upper illustration shows simulation results without an integration of freezing rain. The lower emphasizes the effects of that *Regional Condition*. One can clearly see that the parking situation in the target area is influenced by the freezing rain (5220 parked vehicles vs. 750 parked vehicles). According to these results, we can observe that different drivers and varying external conditions lead to different traffic situations. The consideration of these parameters is thus able to increase the quality of simulation results.

For the reason of comprehensibility, we used a rather simple example. In fact, we already performed a more comprehensive evaluation (Hoch et al. 2011).

5 RELATED WORK

Prior to our development, we performed a comprehensive survey on existing approaches in order to distinguish our work from others and to gain ideas for our own implementation. We started by examining popular and mostly commercial products. In particular, we tried to identify behavioral models for the simulated drivers which comprehend influences of the surrounding environment. We came to the conclusion, that commercial software provides no such feature and extended our survey to the academic domain. Here, we were able to identify many approaches with behavioral models for the simulated drivers. We recognized, that most works either implicitly or explicitly apply an agent-oriented view on the simulated vehicles. Yet, most conceptualizations were focused on the driving process per se and miss the more comprehensive consideration of how the superior goal of reaching a certain target location is actually realized. The vehicles of the examined approaches have been implemented to mimic different driving styles, such as aggressive or cautious. We were not able to identify individual strategic behavior, which derives alternative options and which is triggered by environmental influences.

In the following, we present selected approaches, which influenced our own work the most. For a more comprehensive survey on related work, the reader is referred to one of our previous works (Lützenberger et al. 2011, Lützenberger et al. 2011a, Lützenberger et al. 2011b)

Aimsun (Casas et al. 2010) is an integrated transport modeling software which is used to improve road infrastructure, reduce emissions, cut congestion and design urban environments for vehicles and pedestrians. Next to static and dynamic traffic assignment and mesoscopic simulations, the software supports further features, such as an introduction of public transport priority schemes, a definition of traffic management strategies with triggers and actions and many more.

VISSIM (Fellendorf and Vortisch 2010) is a commercial simulation program for multi-modal traffic flow modeling. With a high level of detail it accurately simulates urban and highway traffic, including pedestrians, cyclists and motorized vehicles. In comparison to most available simulation software, *VISSIM* allows for a modeling of the parking search process.

The *SUMO* (Krajzewicz 2010) simulation framework is an open–source, microscopic traffic simulation package, which has been developed as test bed for research matters. The software has been designed to handle large road networks and can be easily extended. A set extension modules is already provided, while custom implementations are supported as well.

Paruchuri et al. (Paruchuri et al. 2002) describe an approach for modeling unorganized traffic in a traffic simulation. The authors further present a way to use goals for the realization of different driving styles, such as aggressive, normal and cautious. Psychological traits, such as “confidence” and “rush” are implemented as well.

Rigolli et al. (Rigolli and Brady 2005) describe a mesoscopic traffic simulator, in which the simulated vehicles are implemented by using a behavioral multi-agent approach to model human drivers. The simulation software supports different environmental conditions, such as ice or fog.

6 CONCLUSION

In this paper we describe a model which can deal with strategic reactions of simulated drivers on stimuli of the surrounding infrastructure. While we already described the general idea (Lützenberger et al. 2011a) and explained a detailed concept (Lützenberger et al. 2011b), the present work focuses on the formal specification of external influences. We used an example to emphasize the characteristics of external influences and derived a formal specification thereof. Next, we gave a short introduction on the applied model for the drivers in order ensure the comprehension of a simple example. For this example, we created an example of an simulation topology, involving a subway network and parking capabilities. We ran two simulations, one on the original topology and one with an additional appliance of an extreme weather condition. We used the example to demonstrate the functionality of our approach and presented the simulation results. Based on a survey on related work, we can say that none of the examined state-of-the-art products or approaches has the capability to consider comprehensive strategic behavior, which evolves from infrastructural influences and the drivers’ attitudes, yet, our example and reality clarify the necessity for this kind of consideration.

Admittedly, in this paper we only gave little attention to the actual simulation engine, and instead put particular emphasis on the realization of the concept of the applied agents and the model of the infrastructure. The reason for that is that the simulation engine per se does not differ from contemporary approaches and that we see the contribution of our work elsewhere.

6.1 Experiences

In comparison to other traffic simulation frameworks, our concept allows for a specification of the driver's behavior. Most notably, this feature suits for the evaluation of any kind of driver guidance. Currently, we exploit this capability in an industrial founded project, in which we co-operate with the *German* car manufacturer *Volkswagen AG.*, under the objective to measure the effectiveness of Driver Assistance Systems (Hoch et al. 2011). Moreover, we apply our approach in a second nationally founded work, in which we co-operate with *BMW Group* and *Vattenfall Europe*, an international energy provider, under the objective to examine usage patterns and assistance systems for electric vehicles.

6.2 Future Work

We mentioned several improvements throughout the paper. For a start, we will extend the effects of *Regional Conditions* to the physical level. In itself, this is no brand-new feature, but regionally bounded effects (in contrast to a global one) have not been considered commonly. Especially in the context of electric vehicles, external conditions have a considerable impact on range and consumption characteristics and therefore on the behavior of the driver. We will account for an according consideration in the near future.

We also mentioned our idea to use authentic weather data as input for *Regional Conditions*. An according feature will increase the simulation's quality and facilitate the configuration process as well.

Further, we seek to extend our agents to recognize other agents. In doing so, the agents can negotiate with each other and check whether they share goals which are (almost) identical. This offers the possibility to co-operate and share resources like vehicles and to minimize the cost function of each agent (Assuming the cost function reflects monetary and ecological aspects, respectively). With an according feature, the simulation framework can be used for the simulation and optimization of car sharing projects.

ACKNOWLEDGMENTS

This work is partially funded by the *Federal Ministry for the Environment, Nature Conservation and Nuclear Safety* under the funding reference numbers 16EM0004 and 16EM0071.

REFERENCES

Barceló, J. (Ed.) 2010, October. *Fundamentals of Traffic Simulation*. 1st ed, Volume 145 of *International Series in Operations Research and Management Science*. Springer.

Casas, J., J. L. Ferrer, D. García, J. Perarnau, and A. Torday. 2010, October. "Traffic Simulation with Aimsun". See Barceló (2010), 173–232.

Federal Ministry of Transport, Building and Urban Development 2010. *Mobilität in Deutschland 2008*. Bonn and Berlin, Germany: Federal Ministry of Transport, Building and Urban Development.

Fellendorf, M., and P. Vortisch. 2010, October. "Microscopic Traffic Flow Simulator VISSIM". See Barceló (2010), 63–94.

Hoch, N., B. Werther, H.-P. Bensler, N. Masuch, M. Lützenberger, A. Heßler, S. Albayrak, and R. Y. Siegwart. 2011. "A User-Centric Approach for Efficient Daily Mobility Planning in E-Vehicle Infrastructure Networks". In *Advanced Microsystems for Automotive Applications 2011*, edited by G. Meyer and J. Valldorf, VDI-Buch, 185–198. Springer.

Krajzewicz, D. 2010, October. "Traffic Simulation with SUMO – Simulation of Urban Mobility". See Barceló (2010), 269–294.

Lützenberger, M., N. Masuch, B. Hirsch, S. Ahrndt, A. Heßler, and S. Albayrak. 2011a, May. "The BDI Driver in a Service City (Extended Abstract)". In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems, Taipei, Taiwan*, edited by K. Tumer, P. Yolum, L. Sonenberg, and P. Stone, 1257–1258.

Lützenberger, M., N. Masuch, B. Hirsch, S. Ahrndt, A. Heßler, and S. Albayrak. 2011b, June. "Strategic Behaviour in Dynamic Cities". In *Proceedings of the Summer Computer Simulation Conference, The Hague, The Netherlands*, edited by D. Weed, 148–155.

Lützenberger, M., N. Masuch, B. Hirsch, A. Heßler, and S. Albayrak. 2011, June. "Predicting Future(E-)Traffic". In *Proceedings of the 9th Industrial Simulation Conference, Venice, Italy*, edited by S. Balsamo and A. Marin, 169–176.

Paruchuri, P., A. R. Pullalarevu, and K. Karlapalem. 2002. "Multi Agent Simulation of Unorganized Traffic". In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems, Bologna, Italy*, edited by C. Castelfranchi and W. L. Johnson, 176–183.

Ramm, F., J. Topf, and S. Chilton. 2010, September. *OpenStreetMap — Enhancing the free map of the world*. 1st ed. UIT Campbridge Ltd.

Rao, A. S., and M. P. Georgeff. 1995, April. "BDI agents: From theory to practice". In *Proceedings of the 1st International Conference on Multiagent Systems, San Francisco, The United States of America*, edited by V. Lesser and L. Gasser, 312–319.

Rigolli, M., and M. Brady. 2005. "Towards a Behavioural Traffic Monitoring System". In *Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems, Utrecht, The Netherlands*, edited by F. Dignum, V. Dignum, S. König, S. Kraus, M. Pechoucek, M. Singh, D. Steiner, S. Thompson, and M. Wooldridge, 449–454.

Wooldridge, M., and N. R. Jennings. 1995, June. "Intelligent Agents: Theory and Practice". *Knowledge Engineering Review* 10 (2): 115–152.

AUTHOR BIOGRAPHIES

MARCO LÜTZENBERGER received a diploma in computer science in 2009 from the *Technische Universität Berlin*. From 2009 until now, he is part of the *Competence Center Agent Core Technologies* at the *DAI-Labor*, where he is working on his doctoral dissertation in the field of multi-agent based traffic simulations. The concept presented in this work is a central aspect of his thesis. His email address is marco.luetzenberger@dai-labor.de.

SEBASTIAN AHRNDT received his diploma in computer science in 2011 from the *Technische Universität Berlin*. He is currently preparing his doctoral thesis in the field of pervasive computing. His email address is sebastian.ahrndt@dai-labor.de.

BENJAMIN HIRSCH received a Master of Science in Artificial Intelligence from the *University of Amsterdam* in 2001 and a PhD in Computer Science from the *University of Liverpool* in 2005. From 2005 until 2011 he lead the *Competence Center Agent Core Technologies* at the *DAI-Labor*. Currently, Dr. Hirsch is employed at the *EBTIC*, were he works within the *iCampus* initiative and is leading the *iLearning* theme. His email address is benjamin.hirsch@kustar.ac.ae.

NILS MASUCH received his diploma in computer science in 2010 from the *Technische Universität Berlin*. He is currently preparing his doctoral thesis in the field of semantic service matching. His email address is nils.masuch@dai-labor.de.

AXEL HEẞLER Dipl.-Inform. Axel Heßler studied computer science at the Technische Universität Berlin. He currently leads the *Competence Center Agent Core Technologies* at the *DAI-Labor* and has worked on a number of projects within the laboratory, including the *JIAC* agent frameworks. Currently, he is finalizing his thesis about agent-oriented methodologies in the context of real-world projects. His email address is axel.hessler@dai-labor.de.

SAHIN ALBAYRAK, PROF. DR.-ING. HABIL. is the head of the chair Agent Technologies in Business Applications and Telecommunication. He is the founder and head of the *DAI-Labor*, currently employing about one hundred researchers and support staff. His email address is sahin.albayrak@dai-labor.de.