Extending SUMO to support tailored driving styles

Joel Gonçalves, Rosaldo J. F. Rossetti

Artificial Intelligence and Computer Science Laboratory (LIACC)
Department of Informatics Engineering (DEI)
Faculty of Engineering, University of Porto (FEUP)
Rua Dr. Roberto Frias, S/N 4200-465 Porto, Portugal
{pro12009, rossetti}@fe.up.pt

Abstract. Driving behaviour plays a fundamental role in transportation systems, where each single driver has a unique behaviour. In this paper we propose a methodology for eliciting driving behaviour from actual drivers, extract patterns, and generate populations of drivers based on the extracted driving styles. We show some preliminary results from driving behaviour speed. With this work we intent to provide the means for SUMO users to easily generate not only vehicle but also driving style custom populations.

Keywords: SUMO, Traffic Simulators, Driving Behaviour, Behaviour Elicitation.

1 Introduction

Nowadays, Transportation System research is concerned with several important issues related to rapid development of traffic network, especially in urban areas. This phenomenon has introduced dramatic changes in citizens’ mobility and quality of life. Furthermore, it has proved to be a difficult challenge to cope with by researchers, decision makers, and practitioners [1]. While an adequate transportation system enables a good experience for the users, the contrary may be the source of important economic, social, and environmental issues.

By its nature, a transport system can easily become far too complex to be modelled with traditional mathematical approaches [2]. The elements composing the system (e.g. pedestrians, vehicles, road network layout, signalling layout, control systems, and so forth), the various interactions between them, and the solution space for solving a specific problem can be overwhelming. Under such conditions, simulation emerges as a natural approach for handling this complexity. Using simulation, one can model the desired transport system, explore applicable actions to the system, and predict the overcoming results of each action [3, 4]. This approach gives the advantage of covering a vast space of solutions in short time and without disrupting the real system. However, we cannot ignore the significant challenges for modelling the system in a simulation project, especially when the problem to be solved may be influenced by multiple entities with their specific interactions and dependencies [2].
The use of Multi-Agent Systems (MASs) as a paradigm for modelling the Transportation Systems rapidly emerged [5–7]. Specifically, microscopic traffic simulators have the ability to represent each individual vehicle in the transportation system. Each of these vehicles represents a driver, with a pre-defined starting point and destination point. Depending on the network, the drivers may choose their own path according to some decision-making process; they also may change lanes to take over other slower vehicles. Albeit these simulators give a coherent solution for analysing some problems, most of the criticism to this approach is focused on the validation of the tools. In this paper, we intent to contribute with a methodology for evolving the rigid and predictable vehicle behaviour within traffic simulators with behaviours that mimic real driver intentions underlying their decision making.

Our main hypothesis is that if in our simulations we create a virtual population of drivers where each of them resembles a set of extracted driving behaviour patterns, then our simulations will inherit driving behaviour validation and our predictions will be more accurate than traditional driving behaviour approaches when the number of drivers is not very high. In this situation, normal distributions used to play the randomness of driving behaviour are not appropriate. Furthermore, with the driving behaviour extension it would open new possibilities of application for these simulators, e.g., compose a distributed simulation where traffic simulators would manage the non-player drivers in a driving simulation.

In Section 2 we present the methodology, and then in Section 3 we propose the architecture for implementing a SUMO-based solution. After that, we show our preliminary elicited behaviour results in Section 4. Finally, we discuss our contributions and draw conclusions from this work in the last section.

2 Methodology

Our proposed methodology is composed by four core features: (i) driving behaviour modelling, (ii) behaviour elicitation and extraction, (iii) virtual population generation, and (iv) validation and calibration. An overview of this methodology is presented in Figure 1.

The blue elements correspond to the driving behaviour modelling features. The main process behind this phase is to maximize the usage of current driving models already used (e.g., car-following models), and fine tune their parameters in order to resemble the desired driving style. An important step is the mapping between metrics to model parameters since driving performance measurements may not be easily mapped. In those cases the model may be discarded or suffer significant changes. After identifying the proper driving performance metrics we can then design experiments to force users provide the performance metrics in the relevant driving contexts. In this phase we believe that a validation of the overall experiment design, along with the identified metrics, should be reviewed by experts in driving behaviour modelling in order to ensure the mapped metrics were validated.
Next, the driving behaviour elicitation phase is where users participate in the designed experiment and data is collected. When data collection is concluded, multiple data mining related tasks can be performed in order to extract relevant information. It is expectable that tasks such as clustering, stereotype extraction, and even classification are useful to keep the information organized. The creation of clusters (groups with similar behaviour) can be very convenient: (i) they provide general overview of driving behaviour population, (ii) as multiples drivers can belong to a single group, the set of driving behaviour groups does not scale linearly with the number of participants, and (iii) the compression of elicited behaviour in a set of groups, properly characterized, eases the dissemination of that information. In this phase, a calibration procedure can be performed at the individual driver level by measuring the differences between the output of the extracted profile and the actual driving performance in a similar scenario. At group level, careful assessment should be made by analysing how efficient the clustering is by a mathematical perspective, while from a driving behaviour perspective the meaning of the identified group should be representative of a relevant class of driving style.

In the final phase, we proceed to generate the virtual population of drivers to our traffic simulations. However, it may be relevant the frequency with which a given behaviour should appear during the simulation. Thus, researchers should design the population behaviour distribution that is more appropriate to a given problem. This should be validated by taking a sample of drivers from the traffic system simulated in order to assess the assigned distribution. After the population generation, the parameters should be interpreted by the traffic simulator and the simulation can then begin. A final validation step should be made to compare the obtained results with the actual system (if possible).

### 3 Software Architecture

The software architecture proposed is composed of two subsystems in a distributed framework as presented in Figure 2.

In the Remote Domain (RD), there will be a remote server whose major tasks are serving as data repository and performing driving behaviour dissemination. So, this
defines the boundaries between researchers who provide driving behaviour data content and those who just need to obtain a set of driving behaviours. Also, there would be implementations of driving behaviours since more naïve models may not be adequate to represent complex patterns. Thus users could retrieve the driving behaviour from the latest identified clusters (along with their parameters values) and their respective implementations.

In the Local Domain (LD), there are two modules similar to the RD. In essence, these are subsets of the RD content, which were retrieved from the RD and only contain the user’s desired content. A major component in this architecture is the Behaviour Manager which is responsible for interacting with the SUMO microscopic simulator in order to read and update the simulation values.

In practice, the Behaviour Manager works as a standard application connected to SUMO through TraCI and acts as a broker between the simulation engine and the agents that implement real driving behaviours. These agents have a set of vehicles to manage, they perceive the simulation through the Behaviour Manager, assigns updated values to the vehicles in order to simulate the driving behaviour and the Behaviour Manager finally commits the values in the simulation.

With such an approach we may sacrifice a bit of the simulation performance so as to get a flexible solution that will not require major changes to the SUMO microscopic simulator. An obvious bottleneck is the Behaviour Manager since after broadcasting the simulation state to the hosted agents it must wait for the responses and apply the changes to the simulator before requesting the engine to continue to the next step. Also agent coordination and communication with the Behaviour Manager may cause significant overhead in communication channels, especially in case of thousands of vehicles.

The main advantage is that researchers are free to develop their own agent communities’ implementations along with their coordination techniques. This is valid as long as the Behaviour Manager is able to read the simulation state and update vehicle states in the SUMO engine. This flexibility gives the possibility to have a
generic interface for integrating agents based on the Peer-Agent-Design [10] concept, which require more research since they are not yet fully developed.

4 Preliminary velocity behaviour elicitation experiment

Despite the early stage of this project development, we already conducted some preliminary experiments to elicit behaviour from human drivers. More specifically, we collected data regarding the extraction of velocity clusters which will represent class of drivers with similar speed management while driving. Note that this experiment does not consider other vehicles; hence no interaction with headway vehicles or side vehicles in crossroads is considered. For this purpose we used a low-cost driving simulator we already developed in [8], as depicted in Figure 3. The hardware setup is composed by a Logitech G27 steering wheel, a 40-inch-screen TV, and the Serious Game software.

For this experiment we used a total of 9 participants that performed a 10-minute’s period training for adapting to the driving simulator, and then they had to complete three laps from the map presented in Figure 3b. The participants were instructed to respect traffic rules, in particular the road markings and a velocity limit of 120 Km/h.

We defined as dependant variables the vehicle’s position and its instant velocity; hence we can capture vehicle’s position and velocity. As a result, we obtain two time series, with particular importance given to the speed time series.

After performing some preliminary analysis we identified two major behaviour changes: straight lines and curves. Hence we proceed to differentiate such situations by segmenting data according to the vehicle’s position on the map so in the end we aim at obtaining the straight line velocity behaviour cluster (Svel) and the curve behaviour clusters (Cvel). Also, each driver’s velocity time series were merged in order to compensate some eventual errors that could occur during the experiment, leading to more balanced driving behaviour patterns.

![a) User taking experiments.](image1.png)  ![b) IC-DEEP map used.](image2.png)

**Fig. 3.** Using IC-DEEP as a behaviour elicitation tool.
Concerning the data mining tasks, we conducted a computation of a cost distance matrix between all time series using the Dynamic Time Warping algorithm [9]. Once the matrix was finished we then proceeded to group users using Matlab’s hierarchical clustering tool using the Ward method. In the end we obtain the logic groups based on their time series frequency pattern, meaning that even if they two series are desynchronized the method tolerates the difference as long as they have similar frequency pattern.

In Table 1 we present the identified clusters for the straight line velocity. As we can see, even for a small sample it was possible to identify four different groups. We can observe the “spectrum” from a top group that could be considered more aggressive driving due to their highest velocity and standard deviation values, while in the bottom we have a more presumably defensive approach to velocity in straight lines.

As for the curve approach and leaving, we identified three different clusters presented in Figure 4. We interpret the results similarly to the first one as a normal approach since that was already expected by simple observation of people driving.

**Table 1.** Descriptive statistics of each straight line velocity cluster (units in Km/h).

<table>
<thead>
<tr>
<th>Cluster</th>
<th>Max</th>
<th>Mean</th>
<th>Med</th>
<th>Min</th>
<th>Mode</th>
<th>STD</th>
<th>Var</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{vel_1}$</td>
<td>118.19</td>
<td>108.50</td>
<td>110.46</td>
<td>90.38</td>
<td>109.77</td>
<td>6.08</td>
<td>36.93</td>
<td>$U_1, U_5, U_9$</td>
</tr>
<tr>
<td>$S_{vel_2}$</td>
<td>116.62</td>
<td>103.60</td>
<td>105.64</td>
<td>80.52</td>
<td>107.52</td>
<td>8.18</td>
<td>66.97</td>
<td>$U_3, U_4, U_7, U_8$</td>
</tr>
<tr>
<td>$S_{vel_3}$</td>
<td>113.90</td>
<td>99.98</td>
<td>100.49</td>
<td>85.33</td>
<td>100.39</td>
<td>6.99</td>
<td>48.81</td>
<td>$U_6$</td>
</tr>
<tr>
<td>$S_{vel_4}$</td>
<td>91.34</td>
<td>80.88</td>
<td>81.22</td>
<td>69.33</td>
<td>80.84</td>
<td>4.98</td>
<td>24.82</td>
<td>$U_2$</td>
</tr>
</tbody>
</table>

*Fig. 4.* Graphical representation of velocity, as drivers handles curves, for each cluster.
Concerning the second, we observed an approach similar to the first one but the leaving phase is much smoother. And finally the third group can be interpreted as a very sportive driving style since the velocity is kept under high values.

5 Conclusions

In this paper we present a methodology to extend SUMO for enabling more complex driving behaviour models than those currently implemented in some microscopic simulators. Our aim is to ultimately equip SUMO with the necessary tools for representing individual vehicles not just as a set of vehicles performance parameters with a shared driving behaviour, but rather as a simulation population that has a set of vehicle types and a set of driving behaviour models assigned to each vehicle.

We propose that the software architecture support sharing and disseminating elicited data, since it is expectable the elicitation phases will be time consuming and computationally expensive. While on the one hand a remote domain system stores information from driving behaviour elicitation and disseminates the extracted clusters with other researchers, on the other hand we have a local domain with the subset that is interesting to a concrete SUMO experiment. In order to abstract concrete driving behaviour implementations we defined a broker element, the Behaviour Manager, to extend the SUMO simulation and to enable researchers to use their desired tools for creating driver communities.

Overall, our methodology is based on eliciting behaviour from actual drivers in order to generate a virtual society of drivers that represent real-world counterparts in transportation systems. We present results from elicited speed behaviour management using a low-cost simulator on a sample of drivers. As expected the results detected different speed behaviour patterns from different drivers under the same conditions. This allows us to identify generic behaviour patterns which can be used to characterize individual drivers in traffic network simulation settings. We believe this approach will improve the overall realism of traffic simulations by ensuring the behaviour of vehicles mimic real drivers at an individual scale.

References


