

## **Interoperable architecture between simulation and pilots for cooperative and automated driving**

**Robbin Blokpoel<sup>1\*</sup>, Meng Lu<sup>2</sup>, Ondrej Pribyl<sup>3</sup>, Reza Dariani<sup>4</sup>**

1. Product Manager Research, Dynniq, The Netherlands, [robbin.blokpoel@dynniq.com](mailto:robbin.blokpoel@dynniq.com)

2. Strategic Innovation Manager, Dynniq, The Netherlands

3. Associate professor at Czech Technical University

4. Researcher, German Aerospace Centre, Germany

### **Abstract**

In recent years many C-ITS innovation projects have been implemented and early commercial deployment is starting. Simultaneously, automated driving is gaining momentum. Both developments can benefit greatly from simulations for impact assessment. Previous projects struggled with using the same systems from the real-world in simulation and suffered from unrealistic driver behaviour with respect to C-ITS information supplied. The paper first presents a high-level system decomposition to determine the system boundaries and scope. This is followed by an intersection and vehicle specific hardware oriented architecture, which emphasizes where functionality specific for C-ITS and automated driving is implemented. Using that knowledge a simulation architecture was derived, explaining which components were not required or replaced by simulation specific ones. Lastly, different interfacing methods are discussed comparing them with respect to ease of use, compatibility and computational speed. This was all done keeping maximum compatibility between real-world and simulation systems while optimizing simulation speed.

### **Keywords:**

Traffic simulation, automated driving, C-ITS

### **Introduction**

Over the last years many projects have been carried out with cooperative and automated technology, like Contrast[1], FREILOT[2] and AutoNet2030[3]. Gradually trials have developed into early commercial deployments as Compass4D announced in [4]. These projects are mostly following the available ETSI and IEEE standards for communication and architecture and extend these where necessary for the innovative nature of a project, like in i-GAME[5]. In parallel projects were executed to assess the impact of cooperative technology through simulations. The project iTetris[6] combined communication simulation with traffic simulation and eCoMove had specific subprojects focussing on impact assessment resulting in service specific impact assessments [7,8].

These projects used a different implementation of the cooperative applications meant specifically for simulation. Sometimes even simulation specific control algorithms were used. In COLOMBO a generic method to connect real-world controllers to a simulation environment was developed [9] to overcome this problem. Similarly, eCoMove used a Vissim interface for the same purpose. However, to be able to fully integrate real-world deployment and simulations, the systems supplying the cooperative information should be identical as well.

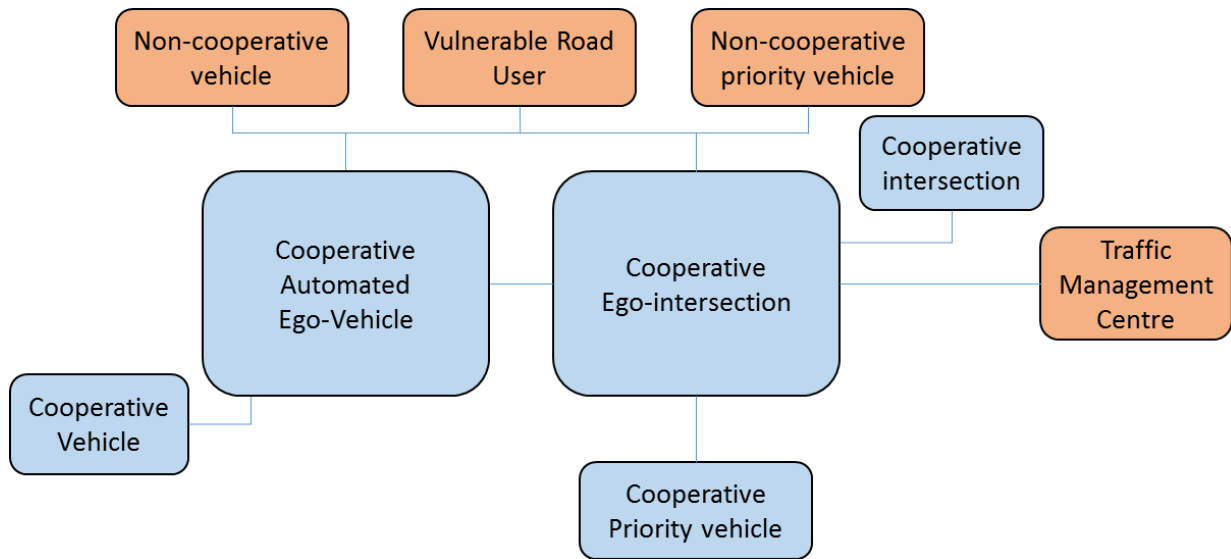
Another major problem identified in the work of [7] was the driver behaviour. Green Light Optimal Speed Advice (GLOSA) trials had better results than simulations. This was caused by the inability of traditional car-following models to make minor adjustments to a speed advice close to an intersection. This is when a human driver or an Advanced Driver Assistance Systems (ADAS) system assesses the gap and speed difference of the vehicle ahead to prevent stopping, while the traditional car following model obeys the speed advice until the vehicle has to stop to prevent a collision. Therefore, having realistic ADAS or automation logic in the loop of a simulation is imperative for realistic impact assessment with simulations.

The MAVEN project aims to have fully interoperable simulation and demonstration implementations. This allows for having the actual systems supplying cooperative information and realistic automation logic in the loop of the simulation. For this a clear reference architecture is required that supports the same building blocks in both situations. A key challenge for this is the differing requirements. In the real-world the systems have to be robust against measurement errors and communication failures, while in simulation a fast execution speed of a scenario, preferably 10x or more than real-time speed, is very important. Another new element is the vehicle emulation, which is a technique to simulate one or more vehicles in a real environment. This can be used to test large platoon behaviour without requiring a large fleet of autonomous vehicles and therefore keeping investment costs acceptable.

The paper will first discuss the high-level system decomposition and hardware architecture. This gives a good introduction on how the systems interact with each other and what is inside the scope of the paper. This is followed by software architecture for simulations. Lastly, the interfaces are described, which is applied both the real-world system and the simulation system.

### System decomposition and hardware architecture

A first step for the architecture is the high-level hardware oriented system decomposition:

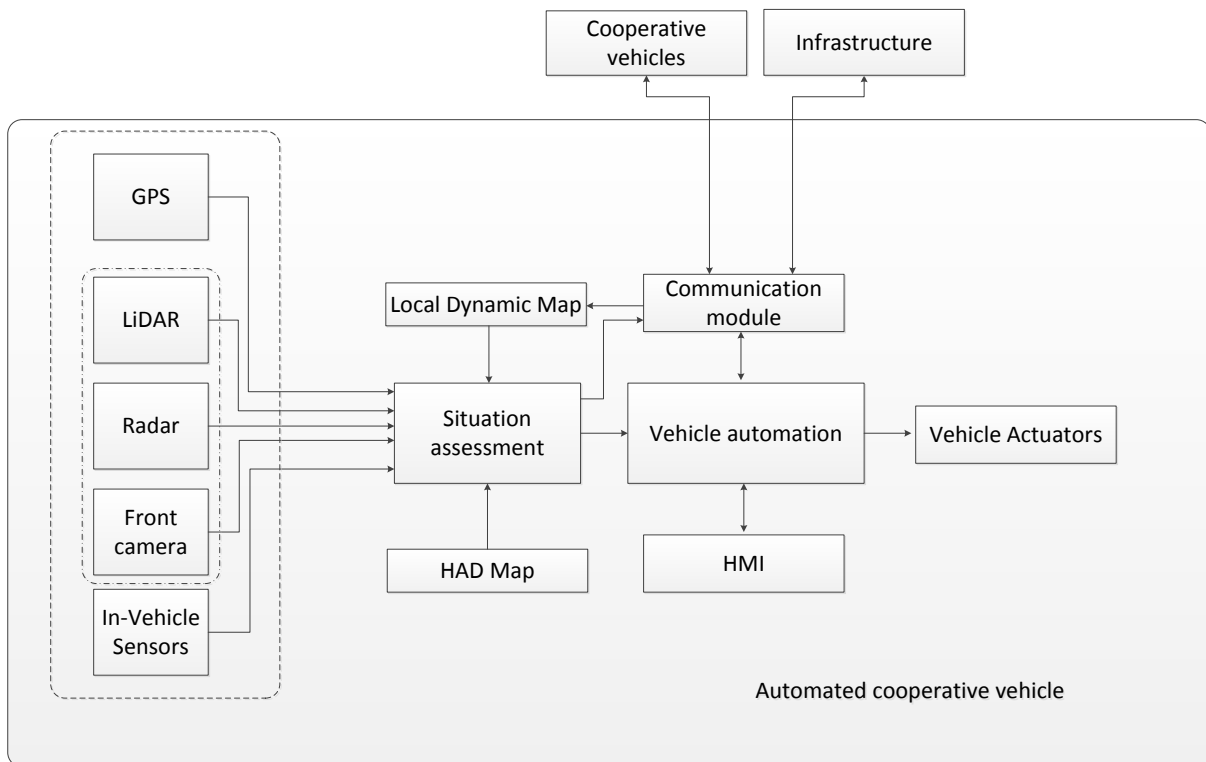


**Figure 1 –High-level system decomposition**

The main actors involved are the Cooperative vehicle and cooperative intersection, which will be described in further detail in Figure 2 and 3. The figure also shows the actors outside the boundaries of the architecture who interact with the system. These are non-cooperative vehicles and Vulnerable Road Users (VRU) as actors that only interact in traditional ways with the system, e.g. respecting the traffic lights and safe interactions with other traffic participants. The non-cooperative priority vehicles have a traditional method of requesting priority; this can be short-range radio (KAR, a Dutch standard using Short Range Device for priority), VEHICLE COMPACT identification and priority (VECOM) or another local technology. Cooperative priority vehicles are vehicles who request priority in a traditional way from a functional perspective (using check-in and check-out points), but use new cooperative technology as communication channel for this. Lastly, the Traffic Management Centre (TMC) is an external actor that may change policy parameters in the intersections and coordinates green waves over multiple intersections. The road authority or traffic management software can trigger this.

### Vehicle Architecture

The cooperative vehicle can also be an automated vehicle and should have full cooperative functionality, including for instance GLOSA and vehicle automation. These vehicles also interact with each other, especially for platooning applications and safety related interactions, in Figure 1 this shows with multiple instances of the cooperative vehicle. The same holds for cooperative intersections, as exchange of data between them enables more optimal traffic control plans. The overall in-vehicle architecture for a cooperative automated vehicle in the scope of MAVEN can be described by vehicle sensors, situation assessment, vehicle automation, vehicle communication module, Human-Machine Interface (HMI) and vehicle actuators. Figure 2 illustrates the overall concept of the in-vehicle architecture of an automated cooperative vehicle



**Figure 2 – Vehicle architecture**

### *Vehicle sensors*

The vehicle has many sensors. The GPS receiver is important for cooperative functionality. It can be enhanced by correction data (Differential GPS) if needed and provides the vehicle position in a global coordinate system. The perception sensors are grouped with a dashed line in the figure and consist of Light Detection And Ranging (LiDAR), Radar, front camera and etc.. These sensors monitor the environment in which the vehicle drives. In-vehicle sensors are also required. These include sensors for velocity, acceleration, yaw angle and yaw rate.

### *Situation assessment*

To plan trajectories a sensor data fusion is necessary; detected objects must be classified, their uncertainty and their standard deviation must be considered and future behaviour must be predicted, the road must be characterized and any relevant data from other cooperative vehicles and infrastructure must be used in the situation assessment level. These data are delivered from the Local Dynamic Map (LDM). The output of this level, the environment model, is the input for the vehicle automation function.

### *Vehicle automation*

In order to drive highly automated while platooning either as platoon leader or follower, different vehicle automation functions are needed. The functions developed in MAVEN are more advanced than existing ADAS or even than a combination of different ADAS. For example, following the leader of a platoon can be seen as ACC combined with a lane keeping system, but MAVEN designed use-cases are

Interoperable architecture for simulation and pilots for cooperative and automated driving

designed for more complicated, critical and realistic scenarios. For example, each follower must be able to be leader at any time and the vehicle automation must always consider trajectory planning and all necessary low level controllers for the entire cooperative automated movements in MAVEN. Furthermore, the vehicle automation has to include interactions with the infrastructure and other cooperative vehicles. For example when the trip of the leader vehicle ends at the next intersection, another vehicle in the platoon has to be able to take the role of leader.

#### *Communication Module*

The envisioned MAVEN use cases are based on the capability of performing platooning, cooperative manoeuvring, negotiation sessions between cooperative automated vehicles and with the cooperative intersection, as well as cooperative sensing. To support these functionalities, V2X communications are needed. V2X communications are possible thanks to the use of ETSI ITS G5 communication modules compatible with the latest stable versions of the ETSI ITS standards [1-8]. Available communication modules are capable to support Unaligned Packet Encoding Rules (UPER) coding and decoding of ETSI ITS Cooperative Awareness Message (CAM) and Decentralized Environmental Notification Message (DENM) messages as well as SAE Map Data (MAP) and Signal Phase and Timing (SPaT). Extended message sets suitable to MAVEN use cases are built on the top or complementing the available ones thanks to the extensibility properties offered by such communication modules. In general, the software of a communication module also provides an LDM with functionalities for managing dynamic information related to other cooperative vehicles or infrastructure detected via V2X. In the context of MAVEN this database can be extended with information achieved with the V2X cooperative sensing functionality and integrated with the environment model.

#### *Human Machine Interface (HMI)*

Automated driving technology which must be implemented in MAVEN informs the driver about forehead situation and infrastructure messages especially in the case of platooning. In this case the driver has the chance to accept or reject to form a platoon and can be informed about possibility of potential platooning at a given time in a given position. As a result, the HMI offers the possibility for the vehicle to communicate with driver and also transfer driver decisions about platooning to other vehicles and infrastructure via the communication module.

Nevertheless, HMI is not a central research topic in MAVEN. Therefore, the developed HMI will be just experimental and only for debugging and demonstration purposes.

#### *Vehicle actuators*

As MAVEN aims at highly automated driving, the whole actuator part (longitudinal and lateral) is controlled by the automation system. Each vehicle will individually take care of the actuator control (e.g. no remote control by the road operator) including probability checks (longitudinal/lateral control commands) before manoeuvring. Even though MAVEN is aiming for a higher automation level, the driver will always be able to take over the vehicle control. Challenging in this context is also a

Interoperable architecture for simulation and pilots for cooperative and automated driving

comfortable and safe vehicle control which doesn't lead to driver confusion or fear.

*HAD Maps*

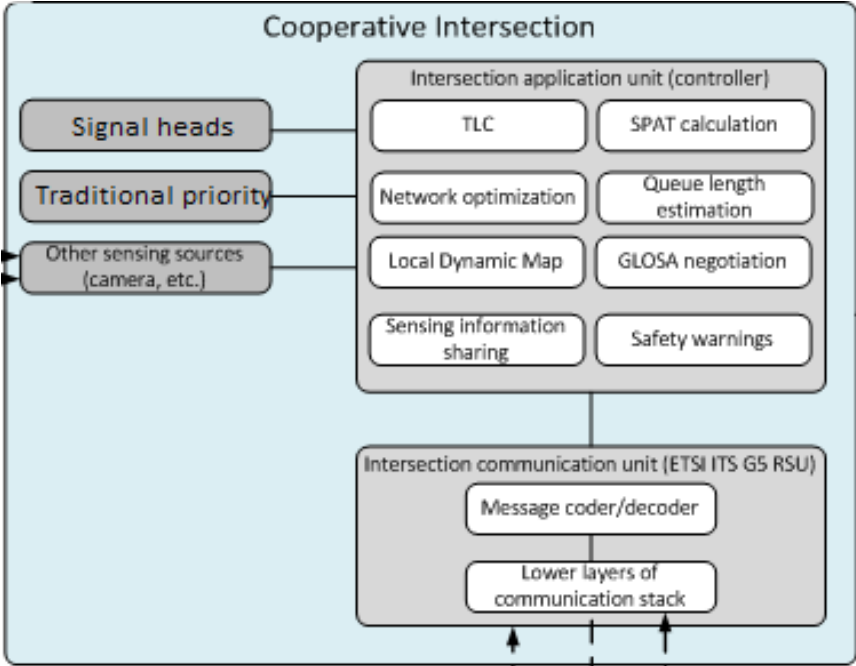
Highly or fully automated driving is not possible without detailed knowledge of the close vicinity of the vehicle. That means that a high accuracy map needs to be made available to the driving logic that provides a reliable and fresh list of static map features, like number of lanes, lane dividers, curbs, traffic lights and information on bicycle or pedestrian paths. Additionally, these different attributes have to be attached to some logical and semantic information that is volatile and dynamic, like traffic light state. Based on this the sensor fusion is able to execute a precise positioning and getting a proper picture of the surrounding enable it to make decisions. And there is a dependency between the availability of required features and the ability to run different scenarios especially in urban areas, which needs to be investigated.

*Local Dynamic Map (LDM)*

Local Dynamic Map provides dynamic information such as traffic light state and it future state as well as other vehicle states, received from V2X and prediction of their future states.

**Infrastructure Architecture**

In Figure 3, the hardware architecture of the intersection is shown. It has a block very similar to the vehicle for intersection communication. Often the communication units are even interchangeable, but just have different form factors and antennae adapted to their specific environment. For a vehicle, the antenna is usually smaller, while for the infrastructure horizontal gain is most important, resulting in designs of over 30cm. From a software point of view, the units are identical, both encode and decode messages and implement the lower level communication protocols of the standards.



**Figure 3 – Hardware oriented intersection system decomposition**

The intersection application unit is very different from the vehicle side. Some components exist on non-cooperative intersections as well. The Traffic Light Controller (TLC) and signal heads are part of any controlled intersection and sensors are necessary for actuated control. For adaptive control network optimization and queue length estimation are required as well, while sensing information is often shared with surrounding intersections or the Traffic Management Centre (TMC). Traditional priority was already introduced with Figure 1 and is an important input parameter for the intersection logic. For cooperative functionality, the traditional traffic control systems often have to extend their functionality and make status information and predictions available to external systems. The SPaT calculation for instance functions better when more information is supplied to it. Safety warnings often rely on local cooperative data or inputs from the TMC.

The LDM is again a central point for all cooperative functionality, storing dynamic data from incoming messages but also from other components, for example to be able to create SPaT messages. Interesting here is that the Highly Automated Driving (HAD) map data is not present here. The infrastructure does not need very detailed information about the road topology, but a more functional view is sufficient. The amount of lanes, their curvature, a mapping of signal groups to lanes and the position of the stop line are sufficient static data. This data is read as a configuration file for the LDM and used to create an ETSI standardized MAP message [10]. This message is required for vehicles to interpret the SPaT as that one only refers to signal group numbers for the dynamic data and the MAP has the geographic mapping of these signal groups. Vehicle positions from Cooperative Awareness Message (CAM) are matched to the lanes defined in the MAP and based on this position cooperative priority can be supplied to them.

With automated vehicles, more opportunities arise and this is represented by the GLOSA negotiation component. These vehicles can also indicate their destination and despite the traffic controller trying to find the most optimal solution for all traffic, the destination of vehicles has to be estimated based on historical averages. When a platoon of automated vehicles is approaching on a road, an early indication of their destination can result in a different optimal traffic control plan. This results in a negotiation process, initially the traffic light control is assuming certain average percentages and calculates a plan based on that. When this results in a passage without stopping for the entire platoon, the plan will not change. However, if this is not the case, then the destination information of the platoon is likely to result in a different optimal plan. After the information is processed a new GLOSA advice is calculated and disseminated via SPaT and the platoon proceeds following this advice. It should be noted that this is not a priority mechanism and therefore it is not guaranteed the plan will change, especially when an emergency or public transport priority was also requested. Similarly, the sensing information from an automated vehicle can assist the traffic control optimization in green wave optimization. When a queue is longer than estimated or vehicles ahead of a platoon have a lower speed, green times may need to be adjusted to be able to let the intended vehicles pass through a green phase without stopping. This is then indicated by the platoon by passing their

sensor information. When a difference with the queue estimations is detected, the plan can be adjusted which is again passed to the vehicles through SPaT messages.

Lastly, the platoons themselves are different from regular vehicles. Due to smaller car following distances the throughput is higher and shorter green phases may be required. Due to increased fuel efficiency with reduced aerodynamic drag, it is already likely automated vehicles will form platoons by themselves. However, when the traffic light controller notices several non-platooning vehicles that have this capability, it may instruct these vehicles to form a platoon if this improves overall traffic efficiency. Contrary, the traffic controller may also advice a platoon break-up when the platoon cannot pass one intersection as a whole, or when safety related circumstances require this.

### Simulation architecture

As explained in the introduction it is important to keep the simulation as accurate as possible, but also interoperable with the real-world systems for ease of use. However, simulation has another requirement, important for impact assessment, which is the simulation speed. Fast simulation allows for more extensive evaluation of scenario's and in general traffic engineers expect at least 10x real time speed for a network with 5 complex intersections on a contemporary desktop pc. For this reason, it will not be feasible to simulate each vehicle with several separate processes as in a real vehicle. This would result in over a thousand individual processes communicating with each other and running in parallel. Additionally, the messages for communication are ASN.1 UPER encoded. This means values can start and end halfway a byte. Encoding and decoding are therefore quite computationally intensive.

Figure 4 shows the simulation architecture, based on selecting required components from the hardware architecture and adding new ones specifically required for simulation.

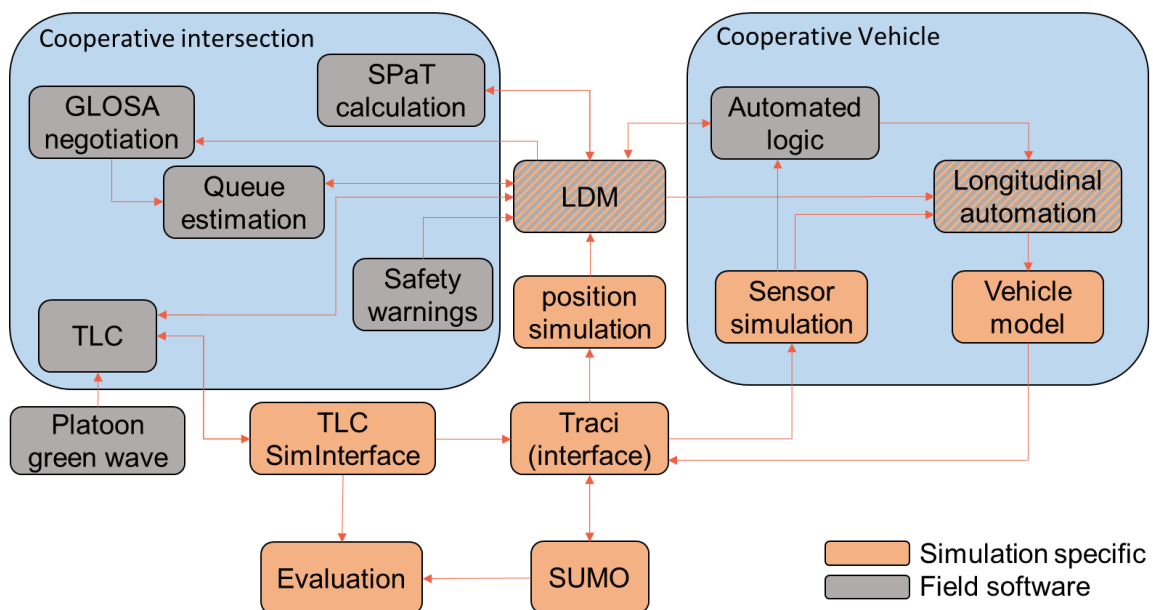


Figure 4 – High level simulation architecture



Comparing this figure with the cooperative vehicle of Figure 2 and 3 reveals some differences. Components that are identical to the real-world implementation are grey, simulation specific components orange and adapted elements are grey/orange striped. Important for the striped element is that the interfaces to the grey elements should stay the same. Both the vehicle and the intersection have a shared LDM now. This is because the communication units have been removed, saving a lot of computational time for encoding and decoding messages. Systems connected to this LDM will not notice a difference, the same data is still present in the same format.

The simulation architecture contains several new components shared between the vehicle and the intersection. Most importantly, SUMO, this is a traffic simulation software package [11], but in theory another simulation package could be used as well. The interface towards SUMO is called Traci and can be used to retrieve data about anything happening in the simulation, e.g. vehicle speed, position, route, detector status, vehicle ID on detector, etc. The interface can also be used to change data in the simulation, e.g. signal head status, vehicle speed, vehicle route, etc. Positioning simulation replaces the actual positioning sensor of the vehicle. As Traci offers a 100% accurate position, it is not realistic to use this for simulations. The results of this simulated position are used to replace the regular CAM messages received by the LDM. A separate TLC interface for simulation is moved out of the cooperative intersection to emphasize that for simulation several additional functionalities had to be added. Signal heads, infrastructure sensors and traditional priority have been integrated with this interface as the actual hardware is not present and it has to connect to Traci for this. Evaluation is a new component required for impact assessment, TLC and SUMO logging is used for this. Lastly, the platoon green wave component from the TMC connects directly with the TLC using the same interface as in the real world (just like the LDM and queue estimation components).

When looking just at the cooperative vehicle it can be noticed that many components are removed. The positioning was already discussed, but other sensors also have to be replaced by sensor simulation, which acquires data from Traci. The HAD map is also removed as the simulation is not detailed enough to include details in such a map. The simulation software mostly simulates one-dimensional movement and lateral lane positions and lane changes are simulated using discrete sub-lanes. The decision whether to change lanes is evaluated based on vehicle positions on the other lane. This means the automation functionality also has to focus on the longitudinal dimension, while modelling the lateral speed of a lane change to determine the sub-lane correctly. Lastly, the automation functionality has no actuators to interact with, so a vehicle model is required to translate the outputs of the real-world automation into speed, lane and route information for Traci.

Looking at the cooperative intersection the functionality is the same as on the street, the simulation specific functionality is all in the TLC SimInterface. An important requirement is that this interface should be in charge of the timing and the controller should be able to run with variable clock speed so the simulation can run as fast as the computations allow.

## Interfaces

For the interfaces some challenges arise related to compatibility. Contemporary V2X messages standardized by ETSI and SAE standards use ASN.1 UPER encoding [10]. Therefore, new V2X messages should also follow this in order to be easily adopted for standardization. Inside the vehicle or the roadside unit this is different. Ideally, in a simulation environment the data can be reached directly in memory without the need of encoding and decoding. However, different components can be written in different programming languages or even run on different machines. In that case there are different options as shown in the table below:

**Table 1 – Interfacing options**

Interface method	Speed	Ease of use	Description
UPER ASN.1	Slow	Medium	This interface method is difficult to implement but very compatible because it is already used in ETSI standards.
PER ASN.1	Fast	Medium	Almost the same as previous, but a computationally much faster method of encoding and decoding at the cost of slightly larger messages.
JSON/XML	Medium	Easy	Basically converts information into a large string. Since the encoding is also sent with each message this is slower.
DLL	Fast	Difficult	DLLs are fast as they basically enable to share the memory between different processes. However, importing can be cumbersome and difficult between different programming languages.
Direct object sharing	Fastest	Difficult	This is the fastest solution but only possible when the same programming language is used.

When comparing different possible interfaces, the PER ASN.1 is most suitable as a good trade-off between compatibility and speed. Whenever possible, direct object sharing could be used as well.

## Conclusion

This paper presented an architecture for cooperative and automated driving technologies extending ETSI standards. Important was the step to go from real-world hardware architecture to a simulation architecture. This was done keeping maximal compatibility and re-use of real-world systems, while enabling retrieving sensor information from the simulation environment and changing states of traffic lights and vehicles according to the actuator outputs. Most importantly, the inclusion of real-world automation should solve problems that occurred in previous projects leading to unrealistic simulation results. Lastly, specific adjustments were made to increase the simulation speed.

## Acknowledgements

The paper presents some preliminary results of the EU-funded project MAVEN (Managing Automated Vehicles Enhances Network), which is funded by the European Commission Horizon 2020 Research and Innovation Framework Programme, under Grant Agreement No. 690727. The authors thank the MAVEN consortium partners for their contributions and support.

## References

1. Passchier, I. et al., Influencing driver behavior via in-car speed advice in a field operational test, In Proceedings 9<sup>th</sup> ITS European Congress, Dublin, Ireland, 2013.
2. Koenders, E., Blom, G., Blanco, R., Energy efficient intersection control: Pilot evaluation results, In Proceedings 19<sup>th</sup> ITS World Congress, Vienna, Austria, 2012.
3. de La Fortelle, A., et. al., Network of automated vehicles: The AutoNet2030 vision, In Proceedings 21<sup>st</sup> ITS World Congress, Detroit, USA, 2014.
4. Compass4d consortium, Announcement of Compass4d continuation in 2016, [http://www.compass4d.eu/en/media\\_room/press\\_releases/announcement-of-compass4d-continuation-in-2016.htm](http://www.compass4d.eu/en/media_room/press_releases/announcement-of-compass4d-continuation-in-2016.htm) (accessed 10-01-2017).
5. Van de Sluis, J., et al. Proposal for extended message set for supervised automated driving, i-GAME project deliverable 3.2, 2015.
6. Rondinone, M., et. al., ITETRIS: a modular simulation platform for the large scale evaluation of cooperative ITS applications, Simulation Modelling Practice and Theory 34, pp99-125, May 2013.
7. Blokpoel, R.J, Islam, Md.F., Vreeswijk, J., Impact analysis of the ecoApproach advice application, In Proceedings 10<sup>th</sup> ITS European congress, Helsinki, Finland, 2014.
8. Islam, Md.F., Vreeswijk, J., Blokpoel, R.J., Impact analysis of the ecoBalanced priority and ecoGreen wave applications, In Proceedings 10<sup>th</sup> ITS European congress, Helsinki, Finland, 2014.
9. Blokpoel, R.J., Interface between proprietary controllers and SUMO, In proceedings 2<sup>nd</sup> SUMO conference, Berlin, 2014.
10. SAE J2735: "Dedicated Short Range Communications (DSRC) Message Set Dictionary". 2009-11-19
11. Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L., Recent Development and Applications of SUMO - Simulation of Urban Mobility, International Journal On Advances in Systems and Measurements, 5 (3&4)pp128-138, December 2012.