2015 IEEE 18th International Conference on Intelligent Transportation Systems

Multi-Resolution-Modeling for Testing and Evaluation of VANET applications

Manuel Schiller, Thomas Behrens, Alois Knoll
Robotics and Embedded Systems
Technische Universität München
Germany
manuel.schiller@in.tum.de

***Abstract*—The evaluation and testing of future driving assistance systems based on Vehicular Ad-hoc Networks (VANETs) in real testbeds is difficult due to the need for repeatable scenarios and large-scale experiments. Therefore, a novel framework based on multi-resolution modeling is presented to test automotive software both accurately and efficiently in large-scale scenarios using virtual test drives in a simulated environment. The approach enables the precise and large-scale evaluation of real-world implementations. This is done through the synchronized execution of simulation models of multiple resolutions representing the vehicle and network domain, as well as the applications encapsulated in virtual Electronic Control Units. This paper provides a detailed and formal description of the applied multi-resolution methodology and explains the developed generic testing platform and the components it is comprised of.**

I. INTRODUCTION

Future vehicle generations will be equipped not only with the conventional onboard sensors, but also with wireless communication hardware to allow the information exchange based on VANETs. A high variety of applications, commonly referred to as Advanced Driver Assistance Systems (ADAS) such as cooperative driving and subsequently automated driving, are enabled through this infrastructure-less communication between the vehicles on the road. Since the achievable range of wireless communication is greater than that of conventional sensors, even vehicles which are farther away and beyond the range of vision can act as relevant information sinks and sources, and therefore affect the ADAS as well as the road traffic system as a whole. The evaluation of these complex applications poses challenges for traditional testing methods due to the increased test complexity caused by the higher number of influences which need to be taken into account. Real-world test drives result in a high effort, even for small-scale scenarios [1], which renders their use rather infeasible for large-scale scenarios.

For these reasons, simulation will be a key methodology to evaluate real-world implementations of VANET-based driving assistance systems in virtual test drives. In the automotive industry the use of simulation is well established in the development process of traditional driver assistance. However, the current emphasis is primarily on the simulation of individual vehicles and their components at a very high level of detail [2]. This approach does not scale well in terms of computational effort for the required type of scenarios which comprise a large number of vehicles. In contrast, the common approach for investigating VANETs in large-scale scenarios is to model the entities in low detail in order to handle the computational complexity. While this level of detail is adequate to capture statistical effects, it is insufficient when real implementations of ADAS need to be evaluated.

Each vehicle is modeled equivalently in the simulation space in both approaches. For the simulative evaluation of applications based on Vehicle-to-Vehicle (V2V) communication a new modeling methodology is required. In this paper we present a holistic approach based on Multi-Resolution Modeling (MRM), which combines the benefits of highly detailed as well as less complex models to achieve an efficient yet accurate simulation of large-scale scenarios.

The remainder of this paper is organized as follows: The testing and evaluation of real-world implementations of ADAS imposes a certain set of additional requirements, which are discussed in section II before giving an overview of the related work. In section III we describe the concept of the multi-resolution model and present a formal specification. In section IV the realization of the concept is detailed. Section V explains the implementation of the framework and in section VI the simulation cost is analyzed for the presented framework. Section VII concludes the paper and gives an outlook for future work.

II. BACKGROUND AND RELATED WORK

Before we proceed to the discussion of related work, it is essential to illustrate our scope and area of application. In order to evaluate and test real implementations of ADAS in a simulated, virtual environment, a holistic view of the entities, effects and dynamics which influence the behavior of the system "VANET", is necessary. This overall system is therefore decomposed into three relevant domains, i.e. the physical domain of each vehicle, the logical domain embodied by the applications as well as the communication network connecting the vehicles through the wireless channel. The domains are coupled either uni- or bidirectionally as shown in figure 1. In contrast to existing approaches, we aim not only to cover the network characteristics but also the behavior of the vehicles and the network-aware applications in high fidelity where necessary, in order to achieve accurate results in the virtual test drives.

In the following section we give a brief overview of existing approaches for VANET simulation and Multi-Resolution Modeling.

978-1-4673-6596-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ITSC.2015.64

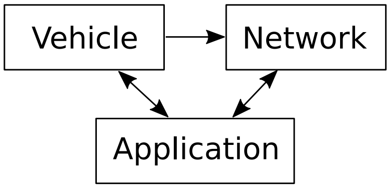

Fig. 1. Interactions and feedback coupling of the relevant domains comprising a VANET

A. *VANET Simulation*

The usual strategy to simulate VANETs found in literature is to bidirectionally couple a network simulator and a microscopic traffic simulation. Following this approach, the interactions between road traffic and network protocols are represented, and the mutual impact can be explored [3], [4]. A number of VANET research simulation frameworks which employ this coupling strategy have been developed [3], [5], [6]. They allow researchers to focus on their specific area of interest, i.e. low-level networking such as medium access, or high-level concepts of applications such as lowering CO_2-emissions or reducing traffic jams. However, they provide insufficient support for the application domain, which prevents the integration of real-world implementations of automotive embedded software into the simulation context. Since large-scale simulations are usually conducted to perform a statistical analysis of the simulation results, efficient but rather simplistic microscopic traffic simulators are used to generate realistic mobility models. When testing and evaluating real ADAS implementations, a more detailed representation of a vehicle's state including its sensors and actors in the simulation is absolutely vital.

B. *Multi-Resolution Modeling*

MRM is defined as the combination of different models of the same phenomenon at multiple levels of resolution which are then executed together [7]. This methodology allows us to find a good balance between simulation accuracy and computing resources, and thus enables the simulations of complex and large-scale systems. This is achieved by employing combinations of high-resolution models, which provide accurate simulation results at the cost of high computational efforts, and of less precise but also less resource consuming low-resolution models. The basic operations in MRM simulations are defined as *aggregation*, the transformation from a high-resolution model to a lower resolution model, and the inverse process as *disaggregation*. These changes in resolution need to be performed to allow entities to interact on the same level of resolution. Alternatively, a concurrent representation-based model has been proposed in [8]. Here, the different levels of resolution coexist simultaneously for each entity.

III. MULTI-RESOLUTION MODELING FOR VANET SIMULATION

A. *Conceptual Overview*

Our aim is to provide a generic platform for the evaluation of real-world implementations of VANET-based ADAS using test drives in a virtual environment. We therefore aim at coupling these implementations with models of different resolutions to represent the aforementioned three domains in a multi-resolution simulation. Contrary to conventional VANET simulation we are not interested in investigating a large number of vehicles from a bird's perspective but rather focus on a single vehicle or a limited number of vehicles which are used to conduct these virtual test drives. In the following we will refer to this kind of vehicle as the EGO car. Without loss of generality, we will describe the concept based on the existence of a single EGO car. However, multiple EGO cars can be simulated in the same overall simulation if applicable. The ADAS under investigation is imagined to be on board of such an EGO car. The simulated measurements, sensor values and network messages are fed into the ADAS. Depending on the nature of the ADAS, it directly or indirectly influences the vehicle's state and behavior. As we target cooperative and autonomous driving applications which inherently rely on bidirectional network communication, the vehicles surrounding the EGO car also need to exhibit the features of the ADAS, i.e. they are also assumed to have the ADAS on board.

As previously mentioned, potentially every simulated vehicle equipped with wireless communication technology can affect the ADAS due to the high transmission range as well as through multi-hop communication which can lead to even larger coverage. Additionally, simulation models of conventional sensors such as radar or synthetic video cameras must be fed with a high-resolution representation of the vehicle's surroundings. To avoid having to simulate every vehicle and its components in high detail, we distinguish between highly and less significant vehicles with respect to the EGO car. This distinction is based on the respective distance between the surrounding vehicles and the EGO car. Nearby vehicles are inherently of more relevance because they pose a higher danger in terms of possible collisions and because their messages transmitted on the vehicular network are of greater importance due to the vicinity of their origin.

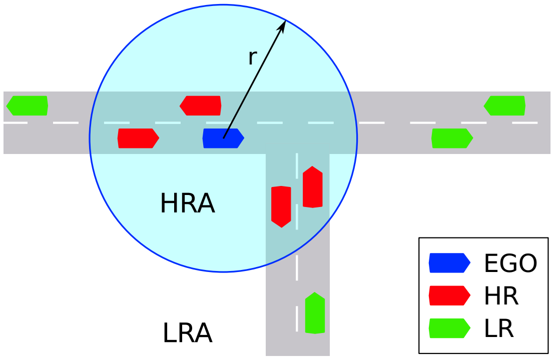

Fig. 2. Dynamic partitioning of the simulated area

Based on this distance criterion we dynamically partition the simulated area into a High Resolution Area (HRA) and a Low Resolution Area (LRA). A schematic view of such a region of interest is shown in figure 2. Here, the HRA is defined as a circle with radius r centered around the EGO car. Every other vehicle which is located within the HRA is modeled in High Resolution (HR), whereas the vehicles outside the HRA are modeled in Low Resolution (LR). This approach is also known as a playbox [8], however, due to the dynamic nature of road traffic, the partitioning of the simulation area is time-dependent. This multi-resolution modeling approach allows to capture both macroscopic and microscopic features of the overall system at the respective level of detail.

Each vehicle in the simulation is modeled in one of the three resolutions EGO, HR or LR. Since the HRA is defined to be centered around the EGO car, the region of interest moves along with the movements of the EGO car. The classification of the surrounding vehicles into HR and LR therefore needs to be performed continuously to reflect the changes in modeling resolution when the vehicles move outside or inside the HRA.

Despite being modeled in different resolutions, the vehicles must be able to interact with each other to capture the mutual influences of the three domains as shown in figure 1 in the overall system. An interaction between vehicles can happen on different channels and possibly cause a change in the state and behavior of one or more submodels of which an entity is comprised. Table I lists the types of inter- and cross-resolution interactions that are considered in our multi-resolution modeling approach.

TABLE I. CROSS- AND INTER-RESOLUTION INTERACTIONS

from to	EGO	HR	LR
EGO	– driver – sensors – network	– driver – sensors – network	– network
HR	– driver – sensors – network	– driver – sensors – network	– network
LR	– driver	– driver	– driver – network

Each vehicle, unless controlled by an autonomous ADAS, is steered by a *driver* model, which is responsible for the driving decisions such as route choice, velocity, lane changing, etc. Typically, microscopic traffic flow is represented using car-following models which take into account the motion of the preceding vehicle. In our multi-resolution model the driver models of EGO and HR vehicles interact bi-directionally, whereas the LR vehicles can be influenced by the other resolutions but do not affect those. This results in the asymmetric interaction matrix shown in table I.

As already stated above, it is necessary to model the vehicles' *sensors* to feed their synthetic measurements into the ADAS under investigation. We define a sensor interaction as the ability of a conventional sensor to perceive another vehicle and its characteristics, such as speed, position, etc. Depending on the complexity of the sensor model, this requires the other vehicle to be simulated in high detail (e.g. 3D rendering). As the detection range of such sensor technology is rather limited (which is one of the main motivations to introduce VANETs in the first place), sensor interactions are only modeled within the HRA and therefore between EGO-HR and HR-HR vehicles.

The third interaction which is covered across different resolutions and within vehicles of the same resolution is the simulated communication *network* itself. Due to the higher transmission range of wireless communication, it is possible for a network packet to be received at a distance which is larger than the extent of the HRA. Additionally, a network packet might be re-transmitted through multi-hop routing so information can travel from far away areas. Therefore, network communication between vehicles of all resolutions is considered to be able to capture the characteristics of the information flow as well as network properties such as medium contention.

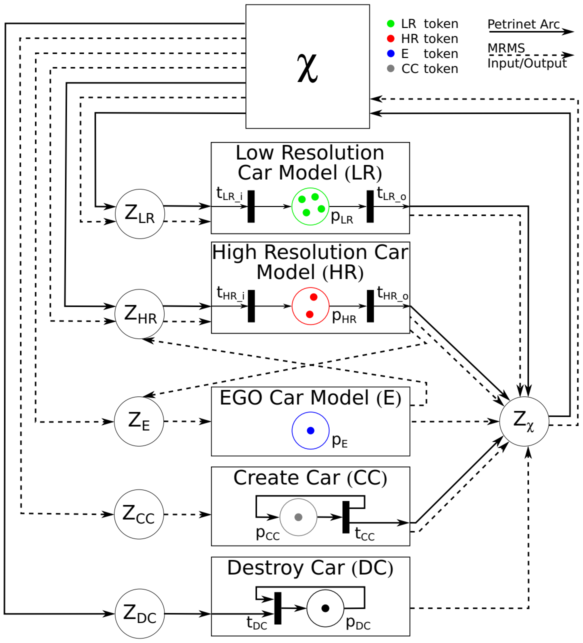

Fig. 3. Hybrid system model consisting of a MRMS and a colored petri net

B. Formal Description

In order to clarify the previously described concept and the relationships between the models of different resolution, we formally define the underlying multi-resolution methodology in the following. This formal description follows a hybrid approach based on the Multi-Resolution Model System Specification (MRMS) [9] and the Colored Petri Net (CPN) [10] formalisms. Figure 3 illustrates the overall system consisting of the two descriptions which are explained in detail in the following. To distinguish between the two formalisms, the petri net arcs are represented by solid lines whereas the input-output-mapping of the MRMS is visualized using dotted lines.

1) Multi-Resolution Model System Specification: The MRMS is an extension of the Discrete Event System Specification (DEVS) [11, pp.138-150] and defines a concept for the description of multi-resolution modeling systems based on the definition of a Multi-Resolution Model Family (MRMF). A MRMF is defined as the set of models which represent the same entity at different levels of resolution in the overall simulation. Using these resolution models the simulation is adapted to the desired varying level of detail. In our current system specification the only MRMF is *Car*, while the other models are single resolution DEVS models. We describe how the domain models are mapped to each resolution of MRMF *Car* in section IV.

The decision which resolution is applied for a given vehicle is taken by the model resolution controller χ. As shown in figure 3, inputs and outputs of every model are connected to χ. Based on the simulation state, χ performs resolution switches if specific criteria are met, such as the distance criterion introduced in the previous section. The system structure also demonstrates its easy extensibility. Additional resolutions for MRMF *Car* as well as for another MRMF, e.g. smart infrastructure, can be added without affecting the existing models and only χ has to be adjusted.

The MRMS formalism is used to specify the multi-resolution system. The detailed explanation of the symbols used in the following can be found in [9]. Based on the definition in [9] the MRMS is given by:

$$\text{MRMS} = \ < X_S, Y_S, \kappa, \{\mathcal{M}_k\}, \chi, M_\chi >$$

where:

$X_S = \emptyset$: system inputs; $Y_S = \emptyset$: system outputs

$\kappa = \{\text{Car, CC, DC}\}$: entities in the simulation

$\{\mathcal{M}_k\} = \{\mathcal{M}_{Car}, \mathcal{M}_{CC}, \mathcal{M}_{DC}\}$: set of MRMFs in the simulation

$\mathcal{M}_{CC}$, $\mathcal{M}_{DC}$: represent single resolution models of creator and destroyer of cars

$\mathcal{M}_{Car} =< \gamma_{Car}, \{M_{HR}, M_{LR}, M_E\} >$: MRMF *Car*

$\gamma_{Car} = \{r_{LR}, r_{HR}, r_E\}$: resolutions of MRMF *Car*

$M_\chi =< X_\chi, s_{0,\chi}, S_\chi, Y_\chi, \pi, \psi, M_\varphi, \delta_\chi, \lambda_\chi, \tau_\chi >$: model of resolution controller χ

$\varphi = \pi(x_\chi, s_\chi) = \{\{r_{LR}, r_{HR}, r_E\}, r_{CC}, r_{DC}\}$: current resolution configuration

$M_\varphi =< D_\varphi, \{I_{\varphi,d}\}, \{Z_{\varphi,d}\} >$: current model of the system

$D_\varphi = \{HR, LR, E, CC, DC\}$: set of modules in the current system

$I_{\varphi,*}$: set of influencer of module *, see table II

$Z_{\varphi,*}$: relations of module *, see table II

TABLE II. INFLUENCERS $I_\varphi, *$ AND MODULE RELATIONS $Z_\varphi, *$ OF THE MRMS

symbol / *	$I_{\varphi,*}$	$Z_{\varphi,*}$
HR	$\{\chi, E\}$	$\{Y_E \times Y_\chi \rightarrow X_{HR}\}$
LR	$\{\chi\}$	$\{Y_\chi \rightarrow X_{LR}\}$
E	$\{\chi, HR\}$	$\{Y_{HR} \times Y_\chi \rightarrow X_E\}$
CC	$\{\chi\}$	$\{Y_\chi \rightarrow \emptyset\}$
DC	$\{\}$	$\{\}$
χ	$\{LR, HR, E\}$	$\{Y_E \times Y_{HR} \times Y_{LR} \rightarrow X_\chi\}$

2) *Colored Petri Net:* The MRMS is superimposed by a CPN which expresses the dynamic multiplicity of the system by representing the vehicles as tokens, which are contained in places denoting the different resolutions. As it is shown in figure 3, each model of the MRMF *Car* contains a place. In contrast to the HR and LR models the EGO model is not connected to the petri net since the number of EGO cars is considered static and an EGO car cannot change its resolution during the simulation. The model resolution controller χ acts as another nested petri net which redirects the tokens to the respective resolution place depending on the current simulation state.

Table III lists the description of the CPN based on the formalism given in [10]:

$$CPN = (S, T, W, C, M_0)$$

TABLE III. FORMALIZATION OF THE COLORED PETRI NET

P / T	χ	p_{LR}	p_{HR}	p_{CC}	p_{DC}	p_E	
t_{LR_i}	-1_χ	$+1_{LR}$					W
t_{LR_o}	$+1_{LR}$	-1_{LR}					
t_{HR_i}	-1_χ		$+1_{HR}$				
t_{HR_o}	$+1_{HR}$		-1_{HR}				
t_{CC}	$+1_{CC}$			$(1-1)_{CC}$			
t_{DC}	-1_χ				$(1-1)_{DC}$		
C	0	∞_{LR}	∞_{HR}	1_{CC}	1_{DC}	∞_E	
M_0	0	0	0	1_{CC}	1_{DC}	n_E	
M_t	0	n_{LR}	n_{HR}	1_{CC}	1_{DC}	n_E	

W is the multiset of arcs between transitions T and places P. C is the capacity of the places, M_0 holds the initial marking of the places and M_t is the current configuration at simulation time t. The colors of the token are $\{LR, HR, CC, E\}$. W indicates how many tokens are taken from or, respectively, added to a place by a transition. The index of each number shows the color of the token. In p_{CC} and p_{DC} one token is removed from and added to the place at the same time, hence it is denoted as "(1-1)". χ is a proxy for a sub-petri net which forwards tokens of the set LR, HR, CC depending on the simulation state.

IV. REALIZATION

Following the definition of [12] we distinguish between the Multi-Resolution Space and the Simulation Space. In figure 4 the MRMF *Car* is shown as the set of the three Resolution Object Models (ROMs) EGO, HR and LR. Each ROM integrates multiple Simulation Object Models (SOMs) to simulate the three relevant domains of each vehicle at the respective level of detail. The SOMs are coupled logically in an uni- or bi-directional manner with each other. In the following sections we describe how the different SOMs represent the three domains in the overall system.

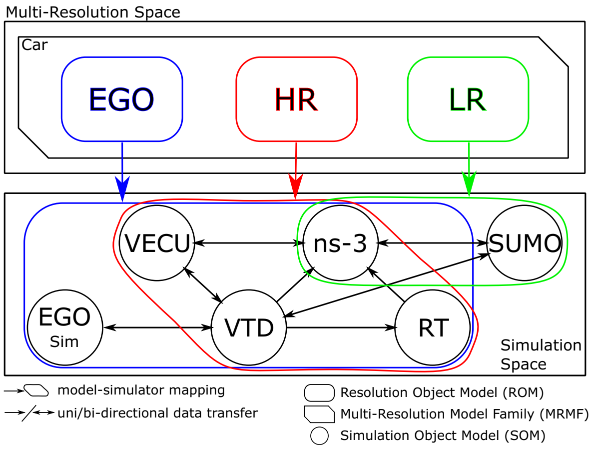

Fig. 4. Mapping between Multi-Resolution and Simulation Space

A. Vehicle Domain

To substitute the real vehicle by its simulated counterpart, the virtual vehicle must provide its state variables in a *sufficient range*, in *sufficient precision* and in a *sufficient temporal resolution*. The concrete manifestations of these three requirements depend on the respective use case. However, as stated in section III-A, this only applies for the EGO car and the high-resolution vehicles as only these vehicles are assumed to have the ADAS on board. The physical domain of the vehicles residing in the LRA can therefore be represented in much lower detail since the simulated state variables and sensor values do not need to be fed into the ADAS implementation under investigation.

The vehicle domain is consequently represented by two different traffic and vehicle simulators of different resolutions. The low-resolution vehicles are simulated using the traffic simulator Simulation of Urban MObility (SUMO) [13]. SUMO is a microscopic, space-continuous and time-discrete simulator. It is well known for its high execution speed and can handle 200,000 vehicles in real-time when using time steps of 1 second [13]. Due to its efficiency, which is partly achieved through its simplified driver model, SUMO is ideally suited to simulate a high number of vehicles residing in the LRA. SUMO only provides a limited set of simulation state variables such as velocity and position, which is sufficient for the LR vehicles.

The EGO and the high-resolution vehicles are simulated by the nanoscopic traffic and vehicle simulator VIRES Virtual Test Drive (VTD). VTD has been developed for the automotive industry as a virtual test environment used for the development of ADAS [14]. Its focus lies on interactive high-realism simulation of driver behavior, vehicle dynamics and sensors. Each simulated vehicle can be equipped with arbitrary simulated sensors, e.g. RADAR. VTD is highly modular, so any standard component may be exchanged by a custom and potentially more detailed implementation. Its standard driver model is based on the intelligent driver model [15], however an external driver model may be applied if necessary. The same concept applies to the vehicle dynamics simulation, for which the standard single-track model can be substituted by an arbitrarily complex vehicle dynamics model adapted for specific vehicles. This corresponds to *EGO Sim* in figure 4, a SOM which simulates selected components of the EGO car in a higher resolution through additional simulation models.

B. Application Domain

Our approach is designed to provide a generic testing platform for real implementations of ADAS which are based on vehicular network communication. One main goal is to support the evaluation of *unmodified* applications in an execution environment which is as close as possible to the real system on which the applications will be deployed in series production. Since the behavior of these network-aware ADAS implementations heavily depends on underlying software layers such as network stacks, the implementations of these lower layers also need to be included in the evaluation. These requirements could be fulfilled by executing the software prototypes on real hardware Electronic Control Units (ECUs) and coupling those with the other domain representations in a hardware-in-the-loop simulation. However, this approach is infeasible for the following reasons:

In the automotive industry the development process of both ECU hard- and software is carried out in parallel, which results in only relatively late availability of the hardware and would thus delay testing of the software prototypes. Additionally, conducting typical VANETs scenarios would require a large number of ECUs as well as a high logistic effort for setting up and performing the actual experiments. Another important aspect to consider is that despite the attempt to reduce the computational effort by employing the multi-resolution modeling approach, performing the overall simulation in real time is often not possible due to the model complexity. The resulting simulator overload causes the simulators to lag behind the real time execution of the software prototypes and thus invalidates the results [16].

For these reasons we integrate *virtual* ECUs (VECUs) as the representation of the application domain into the overall system. This approach solves the dependency on hardware availability and the scalability issues. As the run-time behavior of such a virtualized system is under full control, the time perception of the software prototype can be decoupled from the wall clock time and its execution can be synchronized with the execution speed of the other simulators. While there are various approaches available for virtualizing such embedded systems, we have chosen a hardware-abstract approach for two reasons: The final hardware design is usually determined rather late in the development cycle and the high computational effort required by detailed hardware models is too high. The VECUs are created using ETAS Virtual ECU[1] based on a formal AUTOSAR architectural model [17] and on the hardware independent C code of the implementation. The resulting VECUs can be executed on a traditional desktop PC on top of the host operating system rather than interacting directly with the actual hardware. The VECU execution is stimulated by an internal clock or through virtual interrupts. The internal clock can either progress with respect to the wall clock when running in real-time mode or clock ticks can be injected from the outside, which allows full control over the execution of the VECU. Each VECU is equipped with virtual devices which enable the communication with the outside world.

This virtualization approach allows the execution of unmodified implementations of embedded automotive software in the overall system. A separate VECU is instantiated for each EGO and HR car, whereas the application domain for the LR cars is realized through simplified application models which are executed within the network simulator.

C. Network Domain

In order to evaluate real implementations of VANETs applications in virtual test drives, the network domain is represented using two SOMs to model the wireless communication network connecting the vehicles and their applications.

The transmission and reception of network packets on the wireless communication network is simulated through a discrete event network simulation. For each car in the simulation there exists a corresponding node within the network

---

[1] http://www.etas.com/en/products/isolar_eve.php

simulation. The full network stack as well as simplified models of the applications are executed within the network simulator for the LR cars. Since the ADAS under investigation is executed in separate VECU instances for the EGO and HR vehicles, the VECUs need to be coupled with the network simulation. Therefore, their corresponding nodes within the network simulation act as proxies. The network stack for the EGO and HR vehicles is separated into two parts with the upper layers (including the application layer) belonging to the VECUs, and the lower layers being realized by the network simulator. The network stack is split at the medium access (MAC) layer to allow evaluation of arbitrary routing and transport layers as well as the application functionality, all of which are typically implemented in software and executed in the VECU. The proxy nodes then handle all lower layer functionality that is usually performed by hardware. Since all vehicles exist in the network simulation, direct and indirect communication between vehicles across all resolutions is possible. In order to enable communication between LR vehicles and VECUs above the MAC layer, the LR vehicles need to have compliant implementations of the relevant VANETs protocols (e.g. routing protocols such as GeoNetworking) and, if necessary and applicable, also application models which can act as traffic sources, e.g. transmitting periodic beacons. We employ the packet-level network simulator ns-3 [18] for the simulation of the network domain. To allow synchronization of the network simulator with the other domain representations we implemented a custom event scheduler which can be controlled from the outside.

Accurate modeling of the physical communication channel, which is heavily influenced by traffic dynamics and road surroundings, is crucial to obtain valid simulation results when investigating upper layer protocols and applications [19]. The wireless communication channel therefore needs to be modeled in high fidelity. This is particularly important for critical situations when non-line-of-sight conditions exist, which are caused by static, e.g. buildings, and dynamic obstacles such as large vehicles [20]. We therefore employ a radio propagation simulation based on ray tracing (RT) in order to model the underlying physical channel in high accuracy. Since such ray-optical models require a high computational effort we exploit the processing power of Graphics Processing Units (GPUs) to achieve a high performance, deterministic model of the communication channel. In order to be able to perform large-scale evaluations, this detailed model is only applied for the vehicles within the HRA. Radio propagation for communication happening between EGO-LR, HR-LR and LR-LR vehicles is determined using a simplified model of much lower computational cost.

V. SYNCHRONIZATION AND IMPLEMENTATION

The three described domain representations are either time-driven (vehicle simulator), event-driven (network simulator) or both (VECUs). In order to achieve a deterministic co-simulation comprised of all three domains, the domain representations must be synchronized. The Simulation Synchronizer & Scheduler (SSS) ensures that the execution of the subsystems is synchronous so that no time drifts can occur and causality errors, i.e. executing events from the past, are avoided. Since none of the system representations allows the execution of rollbacks, a conservative synchronization algorithm based on a global event list is applied. The vehicle domain is represented by two different simulators which are synchronized directly with each other based on the master-slave principle, which allows to perform the simulation within the HRA with a high time resolution while the LRA is simulated with coarser time steps.

In our implementation, the domain representations do not communicate directly with each other but through the SSS. The underlying federation concept is derived from the High Level Architecture (HLA), a generic framework for distributed simulations [21]. Each system representation is connected to the SSS by means of a specific ambassador software component which handles the message exchange as shown in figure 5. These messages involve both the synchronization and the exchange of simulation state data. The ambassadors translate the messages from SSS to the respective subsystem and vice versa. This enables the replacement of any given subsystem by either another software implementation or even by real hardware by modifying the corresponding ambassador only. The flexibility of the architecture also makes it possible to add more simulators to the overall simulation and to distribute the system representations on multiple machines.

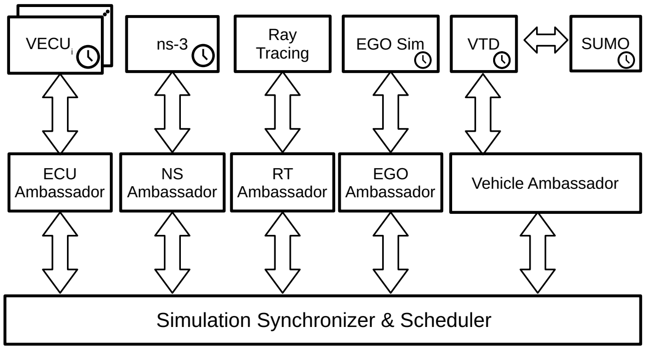

Fig. 5. Implementational overview of the multi-resolution framework

VI. COST ANALYSIS

The presented multi-resolution methodology provides means to solve the trade-off between accuracy and scalability. Scalability aims at maximizing the total number of vehicles while their distribution among the resolutions is affecting the achievable accuracy. In the following we give an abstract cost formulation which illustrates the influencing factors that can be adjusted.

$$\begin{aligned} T_t = & \sum_{i=0}^{n_V} C(\text{ns-3}) + \sum_{k=0}^{n_{LR}} C(\text{SUMO}) + \\ & \sum_{l=0}^{n_{HR}+n_{EGO}} [C(\text{VECU}) + C(\text{VTD}) + C(\text{RT})] + \\ & \sum_{m=0}^{n_{EGO}} C(\text{EGO}) + \sum^{m_{\text{VECU}}} \text{SyncOverhead} \end{aligned} \tag{1}$$

T_t in equation 1 denotes the total simulation cost for a given configuration M_t. M_t is the marking of the CPN at time

t as listed in table III. The overall count of vehicles is n_V, which is the sum of the vehicles in the respective resolution models: $n_V = n_{LR} + n_{HR} + n_{EGO}$
T_t is the sum of the cost functions $C(x)$ for each simulation model, which depend on the amount of vehicles in the respective resolution. Additionally, *SyncOverhead* is the cost of serially executing the VECUs when network messages are received at their corresponding proxy node, which ensures a deterministic execution. m_{VECU} is the number of network messages that leave the network domain and need to be processed at the VECUs. It depends on the number of simulated messages sent and the number of VECUs that are instantiated for a given configuration.

The definition and extent of the HRA implicitly defines the resolution distribution of the vehicles. Depending on the respective test scenario, the model resolution controller χ could be implemented to take the cost function into account and for example keep simulation cost constant at simulation runtime by adapting the definition HRA dynamically.

VII. CONCLUSION

In this paper we proposed a novel approach for modeling vehicular ad-hoc networks at multiple resolutions to allow evaluation and testing of network-aware automotive embedded systems in virtual test drives. The presented methodology aims at solving the trade-off between accuracy and scalability by dynamically partitioning the simulated area into a region of interest and its surrounding area. Depending on this spatial distinction, models of different resolutions for the vehicle, application and network domain are applied to achieve a highly detailed simulation where necessary while maintaining a high overall performance. The interactions between entities of different resolutions are carefully modeled to allow meaningful cross-resolution interactions. In order to clarify the system dynamics and the complex correlations between the models of variable resolutions, we define the proposed system based on formal descriptions. This definition allows the addition of further entities and models to the generic system to realize future testing scenarios. The actual automotive software implementations which need to be tested in the simulated environment are integrated into the overall system by means of virtualization. This enables the detailed analysis of network protocol and application implementations in the context of a realistic runtime execution. The presented approach facilitates detailed and large-scale evaluations early in the product development cycle without being dependent on the availability of real hardware.

As our next steps we plan to tackle hardware-in-the-loop simulation by combining both real and virtual ECUs for later stages in the development cycle as well as reducing the synchronization overhead.

REFERENCES

[1] M. Sepulcre, J. Gozalvez, and J. Hernandez, "Cooperative vehicle-to-vehicle active safety testing under challenging conditions," *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 233 – 255, 2013.

[2] O. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, "Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations," *Vehicle System Dynamics*, vol. 44, no. 7, pp. 569–590, 2006.

[3] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.

[4] F. J. Ros, J. A. Martinez, and P. M. Ruiz, "A survey on modeling and simulation of vehicular networks: Communications, mobility, and tools," *Computer Communications*, vol. 43, pp. 1–15, 2014.

[5] M. Rondinone, J. Maneros, D. Krajzewicz, R. Bauza, P. Cataldi, F. Hrizi, J. Gozalvez, V. Kumar, M. Röckl, L. Lin *et al.*, "iTETRIS: a modular simulation platform for the large scale evaluation of cooperative ITS applications," *Simulation Modelling Practice and Theory*, vol. 34, pp. 99–125, 2013.

[6] B. Schünemann, "V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems," *Computer Networks*, vol. 55, no. 14, pp. 3189 – 3198, 2011.

[7] P. K. Davis and R. Hillestad, "Families of models that cross levels of resolution: Issues for design, calibration and management," in *Proceedings of the 25th conference on Winter simulation*. ACM, 1993, pp. 1003–1012.

[8] P. Reynolds, S. Srinivasan, and A. Natrajan, "Consistency maintenance in multi-resolution simulations," *ACM Transactions on Modeling and Computer Simulation*, vol. 7, pp. 368–392, 1997.

[9] L. Baohong and H. Kedi, "A Formal Description Specification for Multi-resolution Modeling (MRM) Based on DEVS Formalism," in *Artificial Intelligence and Simulation*. Springer, 2005, vol. 3397, pp. 285–294.

[10] K. Jensen, "Coloured petri nets and the invariant-method," *Theoretical computer science*, vol. 14, no. 3, pp. 317–336, 1981.

[11] B. Zeigler, H. Praehofer, and T. Kim, *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, 2000.

[12] S.-Y. Hong and T. G. Kim, "Specification of multi-resolution modeling space for multi-resolution system simulation," *Simulation*, vol. 89, no. 1, pp. 28–40, 2013.

[13] D. Krajzewicz, M. Bonert, and P. Wagner, "The open source traffic simulation package sumo," in *RoboCup 2006*, June 2006.

[14] K. von Neumann-Cosel, M. Dupuis, and C. Weiss, "Virtual test drive - provision of a consistent tool-set for [D,H,S,V]-in-the-loop," in *Proceedings of the Driving Simulation Conference*, Monaco, 2009.

[15] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, p. 1805, 2000.

[16] E. Weingärtner, F. Schmidt, H. Vom Lehn, T. Heer, and K. Wehrle, "Slicetime: A platform for scalable and accurate network emulation." in *Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation (NSDI '11)*, Boston, USA, March 2011.

[17] AUTOSAR. (2015) official webpage. [Online]. Available: http://www.autosar.org

[18] The network simulator ns-3. (2015). [Online]. Available: http://www.nsnam.org/

[19] F. Martinez, M. Fogue, M. Coll, J.-C. Cano, C. Calafate, and P. Manzoni, "Assessing the Impact of a Realistic Radio Propagation Model on VANET Scenarios Using Real Maps," in *9th IEEE International Symposium on Network Computing and Applications (NCA)*, July 2010, pp. 132–139.

[20] M. Boban, T. Vinhoza, M. Ferreira, J. Barros, and O. Tonguz, "Impact of Vehicles as Obstacles in Vehicular Ad Hoc Networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 15–28, 2011.

[21] IEEE, "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)– Framework and Rules," *IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000)*, pp. 1–38, 2010.