Scalable HLA Co-Simulations of Connected and Automated Vehicles using Aggregation of Virtual Federates

Himanshu Neema[†], Harmon Nine[†], Thomas Roth[‡], Wenqi Guo[‡]
[†]Vanderbilt University, Nashville, TN, USA
[‡]National Institute of Standards and Technology, Gaithersburg, MD, USA

***Abstract*—Automated Vehicles (AVs) aim to transfer driving responsibility from humans to the vehicle itself with the objective of enhancing vehicle safety by increasing awareness and reducing response time. At present, automation in vehicles is limited to advanced driver-assistance systems such as adaptive cruise control, which require constant human supervision. A key step toward full automation is realizing communication between vehicles and road infrastructure (Vehicle-to-Infrastructure, V2I) and communication between different vehicles (Vehicle-to-Vehicle, V2V). Connected and Automated Vehicles (CAVs) that leverage V2V and V2I technologies (together called Vehicle-to-Everything, V2X) have information beyond the capabilities of their local sensors which can lead to better informed, and therefore safer, decisions. However, CAVs are not yet available to the general public and additional research is required to understand how these vehicles could perform during the various stages of their potential deployment. One approach to study the performance of CAVs at scale is co-simulation, where a road network could be modeled together with a V2X communications network to evaluate the impact of CAV deployments on the transportation system. The IEEE High-Level Architecture (HLA) is a co-simulation standard that is well-suited for evaluating such complex scenarios but has scalability issues with respect to the number of processes required to run a co-simulation and the amount of data those processes may need to exchange at runtime. This paper presents a novel approach to create scalable co-simulations of road transportation networks containing CAVs and vehicle communication networks. The CAVs are aggregated into a single process while retaining distinct virtual identities in the HLA co-simulation, reducing the number of processes required to control the vehicles and the number of messages required to communicate between simulations. A case study is provided to demonstrate the approach using a co-simulation between the Simulation of Urban Mobility (SUMO) traffic simulator and the OMNeT++ network simulator.**

***Index Terms*—networked co-simulation, cyber-physical systems, system-of-systems, modeling and simulation, road traffic simulation**

I. Introduction

Connected and Automated Vehicles (CAVs) have the potential to significantly transform road transportation systems. Their proliferation can lead to safer and more efficient driving as well as free human passengers to do other productive tasks [1]. Operation of CAVs is enabled through communication with other vehicles and digital infrastructure in the driving environment, including vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication technologies. The information obtained through communication, referred to in general as vehicle-to-everything (V2X), can improve the ability of CAVs to make safe and efficient decisions. Despite this potential, CAVs are currently not available for purchase to the average driver and existing road infrastructure is unprepared to support a full transition to automated driving.

Any future deployment would be staged, with CAVs intermixed with human drivers and varying degrees of digital infrastructure capable of communicating with vehicles. To support this deployment, there is a need to understand how CAVs will affect existing road traffic and the surrounding driving environment, including potential impacts to vulnerable road users (VRU) and humans outside the transportation network. This requires a comprehensive evaluation of the road transportation system under different conditions, including percent penetration of CAVs and V2X communications technologies. Such an evaluation could only be performed where both road transportation network models and V2X communication network models are well integrated.

Co-simulation is a modeling and simulation technique that integrates multiple simulators, such as traffic simulators and network simulators, into a middleware that facilitates coordination and data exchange at runtime. The IEEE High Level Architecture (HLA) defines a standardized set of services that simulators (called federates) can use to interact with each other in a co-simulation (called a federation) [2]. The co-simulation of CAV deployment requires at least three federates to model the road network, the V2X communications network, and the vehicle intelligence. While integration into a co-simulation can be a challenge, re-use of available and maintained simulation software can be more desirable than the development of entirely new capabilities that replicate the combined functionality of existing simulators.

Combining different simulators to model a road transportation network containing both vehicle and V2X network traffic poses significant scalability challenges. In general, the integration of disparate simulates requires synchronization and coordination mechanisms to ensure consistency across different simulation domains. For road transportation, the computational burden of the simulation further increases with the complexity of interactions between vehicles, infrastructure, and communication networks. This paper presents a novel approach

979-8-3503-7594-7/24/$31.00 ©2024 IEEE
DOI 10.1109/DESTION62938.2024.00007

to developing scalable HLA co-simulations of CAVs. The Simulation of Urban Mobility (SUMO) traffic simulator [3] and the OMNeT++ network simulator [4] were integrated into an HLA federation that contains multiple CAVs on a shared road network with V2V communications. An aggregation technique is described that allows one federate to manage all the vehicle controllers while maintaining direct addressability to individual vehicles within the HLA federation. This technique improves the scalability of the co-simulation with respect to the number of CAVs in the transportation system and the amount of data they exchange, which results in improved performance for large vehicle networks. A road network that consists of nine city blocks in a grid pattern containing a fixed number of CAVs driving random routes is used to demonstrate the approach. The co-simulation was implemented using a model-based framework intended to simplify the development of HLA co-simulations, called the Cyber-Physical Systems Wind Tunnel (CPSWT) [5]. The framework uses an open-source Run-Time Infrastructure (RTI), called Portico [6], that implements the HLA standard.

The rest of the paper is organized as follows. Section 2 lists the core requirements of large-scale networked co-simulation of CAVs and describes where the proposed approach of HLA federate aggregation could be useful. Section 3 reviews the work related to this problem. Section 4 provides a comprehensive overview of the system architecture and the implementation details of federate aggregation. Section 5 presents a case study that demonstrates the approach using V2V communications between a fixed number of CAVs on a closed road network. Finally, Section 6 concludes and provides directions for future work.

II. Scalability of HLA Co-Simulations

Simulation-based studies that model both road traffic and vehicle communication networks aim to measure the potential impact of large-scale deployment of CAVs from both a safety and efficiency perspective. These studies must run many simulations that consider parameters including the number of vehicles on the road, the percentage of vehicles that can communicate, the location and types of digital infrastructure available to vehicles, the communication methods used to exchange information, and more. A significant challenge in these simulations is the need to model a large number of independent, intelligent agents that represent the automated vehicles and their interactions with the driving environment. These vehicles are individual cyber-physical systems that capture local environment data, analyze the data to plan for their next course of action, and actuate that plan as a maneuver through steering, throttle, and brake commands.

In HLA terminology, a federate represents an active participant (a process) in the distributed simulation, while an object represents an entity (shared memory) in the simulation environment. A CAV requires active decision making which more closely resembles an HLA federate (an intelligent agent) than an HLA object (data). Additionally, each CAV under simulation will execute different control strategies due to differences in vehicle model, vehicle make, local road conditions, and strategic objective. One vehicle may be running adaptive cruise control on a highway at relatively constant speed, while another vehicle may be executing a complicated merge maneuver in dense traffic. Simulating thousands of these agents requires significant computing, networking, and storage resources. Furthermore, even if these resource requirements are met, the large amount of coordination and data exchange required to simulate these agents on a shared road network will reduce simulation runtime performance.

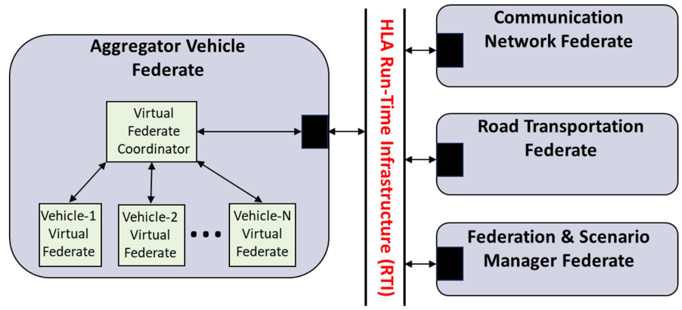

Fig. 1: Federation Aggregation Concept Overview

An important observation is that despite the specific variations among CAVs, there are a lot of commonalities that can be exploited for efficient simulations. Aggregation of the CAV controllers into a single component that controls each individual vehicle under simulation is a potentially useful solution to explore. Figure 1 illustrates this concept where an *Aggregator* federate contains *N* vehicle controllers that rely on the aggregator to coordinate with the federation. From an HLA perspective, the other federates have no visibility into these vehicle controllers, which are the equivalent to private memory hidden within the aggregator, beyond the information the aggregator elects to publish on their behalf.

This paper presents the novel concept of *virtual federates* which are intelligent agents with similar functionalities that are combined into a single aggregator federate to improve the co-simulation scalability. Each of the virtual federates can make independent control decisions which are marshalled through a single aggregated interface to the HLA federation. This allows the communication network federate, the road transportation federate, and even other virtual federates to communicate with the virtual vehicle federates as if the vehicles were implemented as separate HLA federates. The key advantage of this approach is that by reducing the number of federates required for a high-fidelity co-simulation of many intelligent agents such as CAVs, a significant improvement can be achieved in system resource utilization (e.g., system memory, CPU usage, network bandwidth) as well as overall simulation runtime.

III. Related Work

The performance and scalability of road traffic simulations is an important research topic due to the need to simulate large transportation networks containing tens of thousands of vehicles. One approach for improved performance is to partition the road network into different regions, based on geographic location, and run each partition as a separate simulation [7]. This reduces both the number of vehicles per simulation, and the number of messages that must be processed per network simulation. However, vehicles can cross the boundaries into neighboring partitions, and additional synchronization mechanisms are needed to transfer vehicles from one partition (simulation) to another. One approach to implementing cross-partition synchronization is to define overlapping regions at the edges of neighboring partitions which can act as buffers for vehicles [8]. These buffers exchange contents with neighboring partitions only once every 300-time steps (where each time step is equivalent to 0.1 s), with buffer sizes determined based on this time interval and a maximum vehicle speed of 12 m/s. A more sophisticated variation of this method is presented in [9]. Here, the partitions feature extended regions on their borders, termed *extended layers*, which serve as "images" of neighboring border regions. Vehicles from neighboring partitions are represented within these extended layers, reducing the need for frequent synchronizations for vehicle transfers. The partitions can also be defined based on criteria other than geographic location. An alternative approach to partitioning is to create clusters of vehicles based on the degree of interaction between each other, or proximity, to reduce the amount of inter-partition communication [10]. The initial allocation of vehicles to partitions can be determined based on planned route [11]. While the above-mentioned proximity-based partitioning approaches minimize inter-partition communication, they do not directly support a comprehensive evaluation of CAVs that interact with the digital infrastructure.

Vehicular Ad hoc Network (VANET) is a type of Mobile Ad Hoc Network where vehicles communicate with other vehicles or the digital infrastructure using standardized protocols [12]. However, road traffic simulation must also address the increasing complexity of V2X communication technologies, mixed CAV traffic, and modeling vehicle intelligence. The research trend to address this challenge has been to integrate traffic simulators with communication network simulators into co-simulations [13]. The Simulation of Urban Mobility (SUMO) [3] excels at microscopic traffic simulation, modeling individual vehicles and their interactions on a detailed road network. OMNeT++ [4] is a widely used communication network simulator with a large library of models such as for network protocols and devices, routing protocols, and mobile hosts. The combination of these simulators using a co-simulation environment such as VEINS [14] can produce higher fidelity simulations that are better suited to modeling the complex behavior of CAVs. However, these approaches lack capability to model vehicle intelligence in detail, their interactions with other vehicles and infrastructure, and scenario-based experimentation. The HLA-federate aggregation solution presented in the paper along with the experimentation features in CPSWT provides a solution for these problems.

The implementation of time synchronization and data exchange in the co-simulation runtime environment also has significant impact on simulation scalability. The U.S. Department of Energy invested significant effort to develop its open-source Hierarchical Engine for Large-scale Infrastructure Co-Simulation (HELICS) environment to simulate energy systems that contain hundreds of thousands of intelligent agents [15]. One of the main reasons behind the development of HELICS was the poor scalability of HLA to the large federation sizes that would be necessary to model distributed energy resources at scale [16]. The scalability of HLA federations can be improved using the same partitioning techniques discussed for vehicle traffic, through separating federates into different clusters based on their degree of interaction [17]. The number of messages exchanged between federates can also be reduced through more efficient data encoding and publish/subscribe mechanisms implemented in the underlying runtime infrastructure [18]. This paper aims to further improve HLA scalability for these situations that require large federation sizes by developing the concept of virtual federates.

IV. Integration Architecture

To model mixed CAV traffic with V2X communications, a SUMO traffic model was combined with an OMNeT++ network simulation using the HLA co-simulation standard. Both simulators participate in the co-simulation as a federate, i.e. the SumoFederate and the OmnetFederate. The SumoFederate models a closed road network that contains a fixed number of vehicles that are continuously driving. The OmnetFederate represents these same vehicles as wireless hosts in a network model using its open-source INET Framework library [19]. As the vehicles travel through the SUMO road network, vehicle positions are relayed to the INET model using HLA to update the position of each vehicle's corresponding wireless host. V2V communications are simulated within the INET model, subject to signal distance constraints on the wireless hosts.

Each CAV is represented by a separate vehicle federate to model the control logic of individual vehicles. This allows each vehicle to make independent control decisions in the road network according to local circumstances in the simulation. That is, the vehicle federates can send interactions to the SumoFederate to modify their behavior in the SUMO model. The vehicle federates interact with each other using V2V messages that pass through the OmnetFederate network simulation. Whether two vehicle federates are able to interact is dependent on whether their corresponding wireless hosts in the INET model are within broadcast range.

The fidelity of the simulation depends on how closely the positions of the wireless hosts in the OmnetFederate track their corresponding vehicle positions in the SumoFederate. The accuracy of the position tracking affects which vehicles are able to communicate with each other due to their limited communication range and constantly changing positions. The vehicle position information is exchanged between federates using HLA interactions at fixed simulation time steps. This implementation has two major scalability challenges: both the number of vehicle federates and the number of HLA

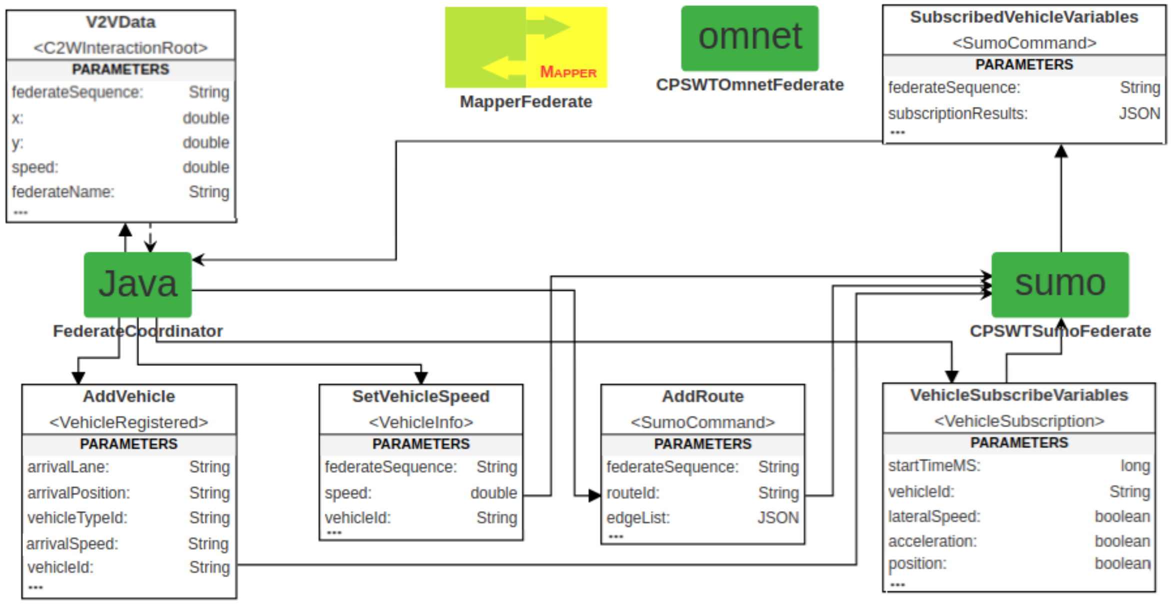

Fig. 2: Simpified Federation Model in WebGME

interactions exchanged per time step for position updates scale linearly with the number of vehicles in the road network.

The concept of an *aggregator* federate containing multiple *virtual federates* can be used to improve the scalability of this implementation. Rather than using separate federates for each vehicle controller, a single aggregator federate can represent all the vehicles in the federation. This aggregator, called the FederateCoordinator, contains N virtual federates where N is the number of vehicles under control in the road network. A virtual federate has access to the same HLA service set, and can both publish interactions (position updates to SUMO; V2V messages) and subscribe to interactions (V2V messages from other virtual vehicle federates). However, the aggregator can send a single vehicle position update that aggregates the updates from each virtual federate to reduce the number of HLA interactions sent each time step. This reduces both the number of non-virtual federates and the number of HLA interactions per time step for position updates from N to *1*. This implementation does introduce a new scalability concern: the performance of the aggregation function. However, as shown in Section V, a virtual federate is expected to require less computational resources than a real federate and the aggregation function is expected to have improved performance over processing separate HLA interactions.

The simplified federation model for this implementation, with some parameters in the HLA interactions omitted for brevity, is shown in Figure 2. The full federation model is given in Appendix A for reference.

The FederateCoordinator receives the interactions destined for any of the virtual vehicle federates. Each virtual vehicle federate can inform the FederateCoordinator of its subscription interests, which the FederateCoordinator will declare in the HLA federation on its behalf. When the FederateCoordinator receives an HLA interaction, it forwards the data to all virtual vehicle federates that have expressed interest. The virtual federates are sub-processes of the FederateCoordinator, and can use the forwarded data to execute any implemented vehicle-specific control logic. If the execution of this control logic needs to send an interaction to the HLA federation, such as updated position information based on vehicle maneuvers, the necessary data to construct the interaction is shared with the FederateCoordinator to send on the virtual federate's behalf.

Dependent on the data model, some interactions may be aggregated while others are not. In Figure 2, only the *SubscribedVehicleVariables* interaction contains aggregated information. The aggregated interactions pass through an aggregation function (or disaggregation function for subscribed interactions) implemented by the FederateCoordinator before being passed between virtual federates and the HLA federation. The non-aggregated interactions, instead, have their sender modified (the *federateSequence* string) to reflect the identity of the virtual federate that provided the data for the interaction.

In the example, the FederateCoordinator publishes an interaction called *VehicleSubscribeVariables*, which directs the SumoFederate to publish certain properties (e.g. speed, position) of a vehicle (identified by vehicleId) every simulation time step of the SUMO model. The SumoFederate publishes the *SubscribedVehicleVariables* interaction in response which contains all of the subscribed properties encoded in JSON. This JSON is encoded as an array where each element contains the data associated with a single vehicle. The FederateCoordinator can pass the individual elements of this array to the relevant virtual federate.

The CPSWTOmnetFederate provides an HLA interface for the OMNeT++ model, and the CPSWTSumoFederate provides an HLA interface for the SUMO model. To update the posi-

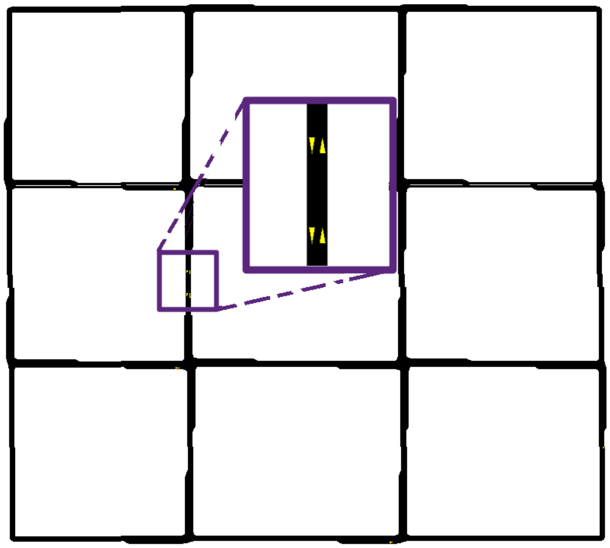

Fig. 3: SUMO Simulation of a 3-by-3 block grid containing 12 vehicles (four vehicles shown magnified as yellow triangles)

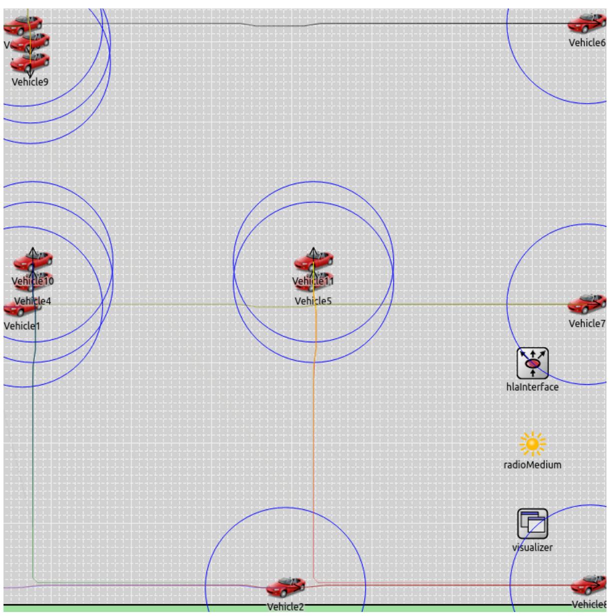

Fig. 4: Wireless hosts in INET for the SUMO vehicles

tions of the wireless hosts representing the vehicles in the OmnetFederate, *VehicleSubscribeVariables* interactions must be sent to the SumoFederate (here, from the FederateCoordinator) to subscribe to the 2-dimensional position coordinates of the vehicles. Each *SubscribedVehicleVariables* consequently sent by the SumoFederate contains the position information for all of the vehicles encoded as JSON in the 'subscriptionResults' parameter. The MapperFederate is configured to translate this interaction into a *BulkSetPosition* interaction (not shown in Figure 2 as it is contained inside the MapperFederate model), to which the OmnetFederate implicitly subscribes. Using this interaction, the OmnetFederate updates the positions of the wireless hosts. The MapperFederate is likewise used to translate the *V2VData* interactions sent from the FederateCoordinator into interactions subscribed by the OmnetFederate to allow that data to propagate through the simulated V2X communications network. The implementation of this mapping function, specific to the OmnetFederate, is discussed in [20].

V. Case Study

This section presents a case study to demonstrate that the concept of virtual federates can lead to improved performance in the co-simulation of road transportation networks. The case study compares the performance of two functionally identical federations: one federation that contains separate federates for each vehicle under simulation, and one federation that contains a single aggregator federate, called *FederateCoordinator*, that controls all of the vehicles. To ensure comparable results:

- both federations were run in a virtual machine with 4 CPUs and 32 GB memory on the same physical machine with comparable background loads;
- both federations were implemented using the same HLA runtime infrastructure (Portico [6]); and
- the vehicle federates and the aggregator federate were implemented in the same programming language (Java).

The federations were implemented based on the HLA model shown in Figure 2. Six separate federations, each using a different number of CAVs, were run for both federation types, i.e. one type with each CAV represented by its own federate (*type A*), and one with a single aggregator federate controlling all of the CAVs (*type B*). The number of CAVs in these federations were 6, 12, 18, 24, 30, and 36. Besides the CAVs, both types of federation contain 3 other federates (shown in Figure 2) that are identical between the federation types A and B. The *SumoFederate* models a 3-by-3 block grid shown in Figure 3, and each vehicle in SUMO (whether represented by a real or virtual federate) has an associated wireless host modeled in the INET Framework shown in Figure 4. There is also a *Mapper* federate which takes interactions containing position updates for the CAVs from the SumoFederate and maps them to interactions containing position updates for the corresponding wireless hosts in the *OmnetFederate*.

The simulations use the techniques of dynamic messaging and embedded interactions as described in [20], which are central for general-purpose, reusable communication network simulation within HLA co-simulations. The time taken for the federations to complete 500 simulation time steps is shown in Table I. From the results, we can see that with 6 CAVs, the simulation took longer for type B federations than for type A. However, for a larger number of federates (12 through 36) the time savings incurred by type B federations over type A range from 13.8 % to 19.9 %, which are significant.

Table II shows the amount of memory taken by the federates representing the CAVs in both types of federation. The memory taken by each federate was determined using the `top` utility, and did not vary significantly over a simulation run. The results shown are a snapshot of the memory taken after 200 simulation time steps. From the table, we can see that the memory savings in the federations containing only the

No. CAVs	Actual Federates (type A)	FederateCoordinator (type B)	% Difference
6	122.863	139.183	-13.3 %
12	167.286	144.204	13.8 %
18	242.916	200.968	17.3 %
24	337.083	295.188	12.4 %
30	517.810	414.879	19.9 %
36	692.281	580.088	16.2 %

TABLE I: Execution times (s) for 500 simulation time steps of federations containing actual federates for the CAVs and a single aggregator federate representing all CAVs.

No. CAVs	Actual Federates (type A)	FederateCoordinator (type B)	Multiplicative Factor
6	1,049,928	179,866	5.8
12	2,161,727	327,536	6.6
18	3,343,710	191,244	17.5
24	4,536,408	190,222	23.8
30	5,479,762	193,618	28.3
36	6,436,844	201,398	32.0

TABLE II: Amount of resident memory (KB) taken by federates representing CAVs in both types of federation.

No. CAVs	Actual Federates (type A)	FederateCoordinator (type B)	Multiplicative Factor
6	0.248	0.042	5.9
12	0.376	0.037	10.2
18	0.392	0.032	12.2
24	0.414	0.024	17.3
30	0.405	0.019	21.1
36	0.377	0.015	25.1

TABLE III: Amount of CPU seconds taken per wall clock second by federates representing CAVs in both types of federation. CPU time was calculated by taking the difference of cumulative CPU time between two samples taken 5 wall clock minutes apart.

aggregator federate are significant: the memory taken by type A federations ranges from 5.8 to 32 times the amount taken by type B federations, increasing with the number of CAVs. NOTE: We could not simulate a higher number of CAVs in the type A federations due to limited available memory in computer (32 GB).

Table III shows the amount of CPU time (s) taken per second of wall-clock time for the two types of federation. Again, there is a significant savings in CPU usage for type B federations: the CPU usage in the type A federations ranges from 5.9 to 25.1 times higher than the CPU using for type B federations. As with memory, the savings in CPU time increase with the number of CAVs.

VI. Conclusion & Future Work

This paper presented a novel solution to improve the scalability of HLA co-simulations of connected and automated vehicles. It introduced the concept of virtual and aggregate federates, which can be used to reduce the number of federates in a federation and the number of interactions exchanged at runtime. A simple case study was demonstrated that showed an implementation of virtual federates can lead to significant performance improvements, both in terms of execution time and computational resources required, compared to functionally equivalent federations that implement each federate separately. This will allow for the study of larger road transportation networks, containing more communicating vehicles, using the same amount of compute resources. These studies are necessary to support large-scale evaluations of the potential impact of CAVs on existing vehicle networks during the initial stages of deployment, which will be required to accelerate the adoption of V2V and V2X technologies and move towards realization of fully autonomous vehicle operations.

Future extension of this work includes dynamic creation of new virtual federates using parameterization and automation (to allow vehicles to enter and leave the road network), development and demonstration of aggregation functions that can handle a more diverse set of virtual federates (to model different types of automated vehicles including significant differences in sensors or perception), and extending the model to include other intelligent actors such as digital infrastructure along the roads. In addition, the techniques developed for aggregation of virtual federates are generic and could be applied in other application domains such as the co-simulation of a transactive distribution grid involving a large number of grid consumers (and prosumers) that require simulating individualized behavior that must coordinate through communication with grid operators.

VII. Acknowledgements

This work at Vanderbilt University is supported in part by award #70NANB22H216 from the U.S. National Institute of Standards and Technology (NIST). Official contribution of the NIST is not subject to copyright in the United States. Certain commercial products are identified in order to adequately specify the procedure; this does not imply endorsement or recommendation by NIST, nor does it imply that such products are necessarily the best available for the purpose.

References

[1] Daniel Watzenig and Martin Horn. *Automated driving: safer and more efficient future driving*. Springer, 2016.

[2] 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)*, pages 1–38, 2010. doi: 10.1109/IEEESTD.2010.5553440.

[3] M. Behrisch D. Krajzewicz, J. Erdmann and L. Bieker. Recent development and applications of SUMO-simulation of urban mobility. *International journal on advances in systems and measurements*, 5(3&4), 2012.

[4] A Varga. The OMNeT++ discrete event simulation system. *In: Proceedings of the European Simulation Multiconference (ESM'2001)*, 2001.

[5] H. Neema, J. Sztipanovits, T. Raub C. Steinbrink, B. Cornelsen, and S. Lehnhoff. Simulation integration platforms for cyber-physical systems. In *Proceedings of the Workshop on Design Automation for CPS and IoT*, pages 10–19, 2019. doi: 10.1145/3313151.3313169.

[6] Portico. Available: https://github.com/openlvc/portico (visited on March 21, 2023).

[7] A. Ventresque, Q. Bragard, E. Liu, D. Nowak, L. Murphy, G. Theodoropoulos, and Q. Liu. SParTSim: A space partitioning guided by road network for distributed traffic simulations. In *2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real-Time Applications*, pages 202–209, 2012. doi: 10.1109/DS-RT.2012.37.

[8] B. Jiang and H. Zhang. Realization of distributed traffic simulation system with SCA and SDO. In *2009 Second International Conference on Future Information Technology and Management Engineering*, pages 222–225, 2009. doi: 10.1109/FITME.2009.61.

[9] Y. Xu, V. Viswanathan, and W. Cai. Reducing synchronization overhead with computation replication in parallel agent-based road traffic simulation. *IEEE Transactions on Parallel & Distributed Systems*, 28(11):3286–3297, nov 2017. doi: 10.1109/TPDS.2017.2714165.

[10] H. Aydt Y. Xu and M. Lees. SEMSim: A distributed architecture for multi-scale traffic simulation. In *2012 ACM/IEEE/SCS 26th Workshop on Principles of Advanced and Distributed Simulation*, pages 178–180, 2012. doi: 10.1109/PADS.2012.40.

[11] J. Yu Z. Fu and M. Sarwat. Demonstrating GeoSparkSim: A scalable microscopic road network traffic simulator based on Apache Spark. In *Proceedings of the 16th International Symposium on Spatial and Temporal Databases*, SSTD '19, page 186–189, New York, NY, USA, 2019. Association for Computing Machinery. doi: 10.1145/3340964.3340984.

[12] M. R. Ghori, K. Zamli, N. Quosthoni, M. Hisyam, and M. Montaser. Vehicular ad-hoc network (VANET): Review. In *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, pages 1–6, 2018. doi: 10.1109/ICIRD.2018.8376311.

[13] J. S. Weber, M. Neves, and T. Ferreto. VANET simulators: an updated review. *Journal of the Brazilian Computer Society*, 27(1):8, 2021. doi: 10.1186/s13173-021-00113-x.

[14] C. Sommer, R. German, and F. Dressler. Bidirectionally coupled network and road traffic simulation for improved IVC analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15, 2011. doi: 10.1109/TMC.2010.133.

[15] T. D. Hardy, B. Palmintier, P. L. Top, D. Krishnamurthy, and J. C. Fuller. HELICS: A co-simulation framework for scalable multi-domain modeling and analysis. *IEEE Access*, 12:24325–24347, 2024. doi: 10.1109/ACCESS.2024.3363615.

[16] B. Palmintier, D. Krishnamurthy, S. Smith P. Top, J. Daily, and J. Fuller. Design of the HELICS high-performance transmission-distribution-communication-market co-simulation framework. In *2017 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, pages 1–6, 2017. doi: 10.1109/MSCPES.2017.8064542.

[17] H. Neema, T. Roth, C. Wang, W. Guo, and A. Bhattacharjee. Integrating multiple HLA federations for effective simulation-based evaluations of CPS. In *2022 IEEE Workshop on Design Automation for CPS and IoT (DESTION)*, pages 19–26, 2022. doi: 10.1109/DESTION56136.2022.00010.

[18] Z. Wang, X. Jiang, and J. Shi. Efficient communication approach in HLA-based distributed simulation for VR application. 2:593–601, 06 2006.

[19] Levente Mészáros, Andras Varga, and Michael Kirsche. Inet framework. *Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem*, pages 55–106, 2019.

[20] H. Neema, H. Nine, and T. Roth. Reusable network simulation for CPS co-simulations. In *Proceedings of Cyber-Physical Systems and Internet of Things Week 2023*, CPS-IoT Week '23, page 122–129, New York, NY, USA, 2023. Association for Computing Machinery. doi: 10.1145/3576914.3587531.

APPENDIX

Figure 5 shows the complete federation model used for the case study which includes the complete list of parameters for each HLA interaction.

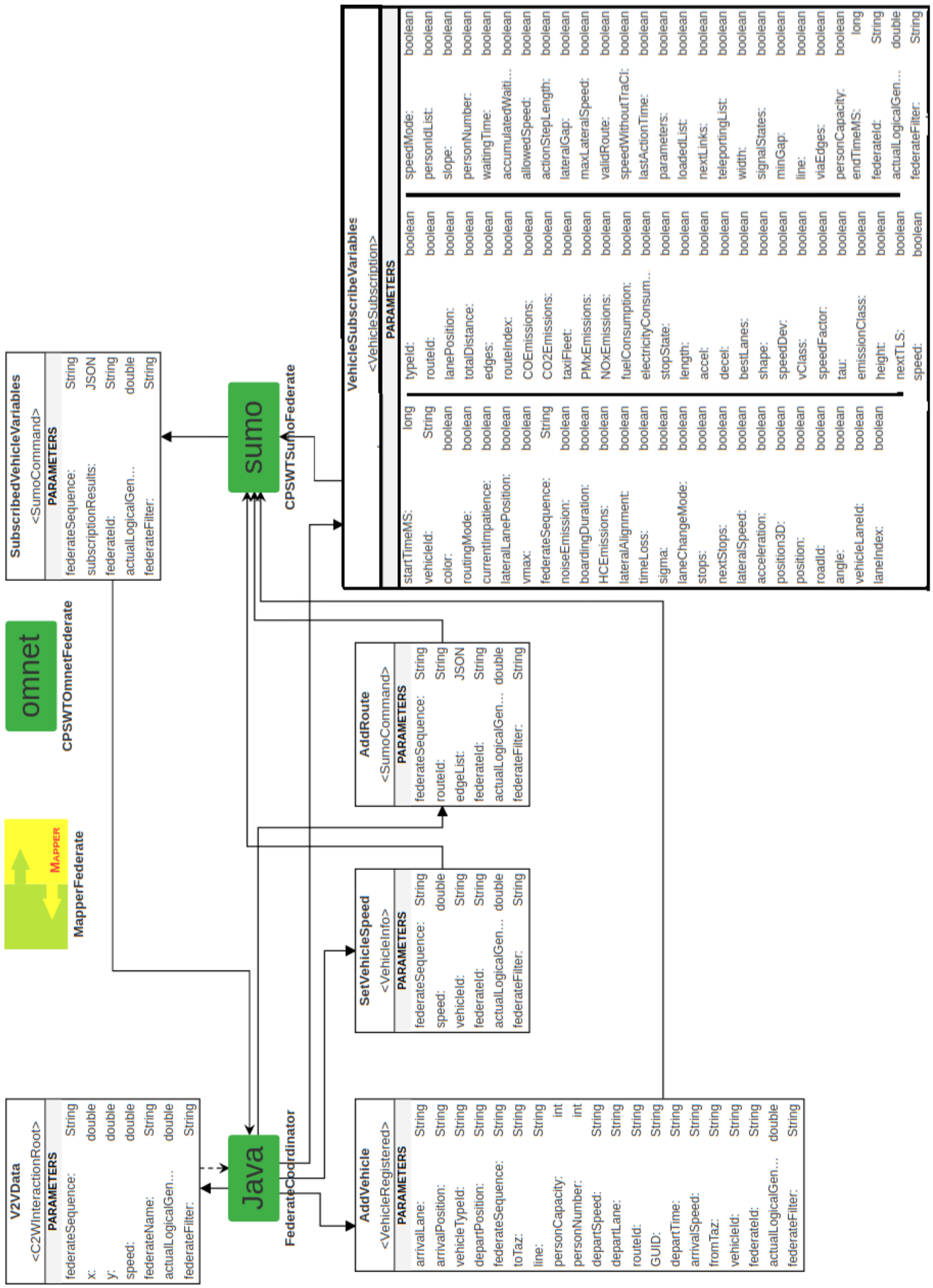

Fig. 5: Complete Federation Model in WebGME