HAL
open science

A System of Systems Framework: Cooperative Maneuvers Manager for Autonomous Vehicles

Mohamad Ali Assaad, Reine Talj, Ali Charara

To cite this version:

Mohamad Ali Assaad, Reine Talj, Ali Charara. A System of Systems Framework: Cooperative Maneuvers Manager for Autonomous Vehicles. 13th Annual International Conference on System of Systems Engineering (SoSE 2018), Jun 2018, Paris, France. pp.92-97, 10.1109/SYSOSE.2018.8428759 . hal-01900496

HAL Id: hal-01900496

https://hal.science/hal-01900496v1

Submitted on 6 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A system of systems framework: Cooperative Maneuvers Manager for Autonomous Vehicles

Mohamad Ali Assaad*, Reine Talj*, and Ali Charara*
*Sorbonne Universités, Université de technologie de Compiègne, CNRS, Heudiasyc UMR 7253.
57 avenue de Landshut, 60203 Compiègne, France.
Email: mohamad-ali.assaad, reine.talj, ali.charara@hds.utc.fr

***Abstract*—Systems of Systems (SoS) are complex systems, in which constituent systems are by themselves independent systems, that cooperate to achieve a common goal. Intelligent Transportation Systems (ITS) are SoS that seek to achieve better transportation experience for users, by using communication between different users of such systems. In this paper, we present the Cooperative Maneuvers Manager for Autonomous Vehicles (CMMAV), a SoS framework that contributes to the ITS by enabling autonomous vehicles to perform complex maneuvers on highways in a cooperative way. We used Systems of Systems Approach to Context-based Requirements Engineering (SoS-ACRE) to model the system.**

***Index Terms*—Systems of systems, Intelligent transportation systems, autonomous vehicles.**

I. Introduction

Systems of Systems (SoS) are systems that emerge from the collaboration between multiple independent systems, called constituent systems, that seek to achieve better performance, or share common objectives. Systems of Systems offer capabilities that are not achievable in a single system, they have no single point failure, and continue to work even in changing environments. The United States Department of Defense (DoD), in their framework [1], define a system as follows: an assemblage of constituent systems - machine, human - that perform activities [...] and are interacting or interdependent. Therefore, a System of systems, which is a system, is the result of multiple different component systems, working together. Intelligent Transportation Systems (ITS) are an example of SoS, in which different government and independent organizations, that share the same objectives, work together to improve the state of transportation around the world. ITS's applications range from providing information to its users, to assisting them in taking different decisions. Advanced vehicle control and safety systems (AVCSS), is the ITS category that contains the different systems that integrates in connected (autonomous) vehicles, to assist the driver or take control over the vehicle in different situations. As a contribution to this category, we introduce in this paper the cooperative maneuvers manager for autonomous vehicles (CMMAV), a SoS that manages complex maneuvers on highways in a cooperative way. This paper is organized as follows: Section II introduces SoS, with their definitions and types. It then shed the light on their engineering challenges, and introduces frameworks as a solution to build working SoS. Section III contains an example of SoS, the ITS. It justifies why ITS is an SoS, and introduce its different categories and applications. Later on we introduce our contribution to the ITS in subsection III-B. Section IV describes our framework, and shows its different views. Later on, we show how we face the SoS challenges such as stable intermediate forms. Finally, we finish with a conclusion in section V.

II. Systems of Systems

A. Definition

The multidisciplinary of SoS, and their presence in a wide range of applications, produced several definitions of SoS ([2], [3]). These definitions mainly are based on the specific domains of application in which a particular SoS exists ([4], [5]). Another approach to define SoS, was to propose characteristics, shared by SoS regardless of their domain of applications. These definitions describe key characteristics of SoS, and their component systems. Maier [6] lists 5 characteristics: Operational Independence, Managerial Independence, Geographic Distribution, Evolutionary Development, and Emergent Behavior. Other definitions identify other characteristics such as autonomy and belonging of the constituent systems in [7], and continuous evolution of the SoS in [8].

B. SoS Types

All SoS share the same characteristics, the independence of constituent systems, the evolutionary nature of the overall system, and the geographical distribution of the SoS. But not all SoS are the same, in the sense that the degree to which these characteristics are applied differ from one SoS to another. To account for these differences, Maier [6] proposed 3 types of SoS: directed, collaborative and virtual. Later on Dahmann *et al.* [9] proposed a fourth type: acknowledged. Collins *et al.* [10] proposed 3 key factors that could be used to identify a SoS's type: the nature of its objectives, its governance, and the inter-relationship of its constituent systems (Figure 1).

C. SoS Architectural Frameworks

Comparisons between traditional systems and SoS of ([11], [12]), show that the current system engineering (SE) process is not enough to build and manage SoS. In a traditional system, parts are chosen as to achieve one objective (or set of objectives). These objectives are specific to this system, and all parts of the system will work to achieve these objectives. Whereas in a SoS, parts have their proper objectives, they

SoS Objectives: Emergent					**SoS Objectives: Recognized**
SoS Governance: Community	*Virtual*	*Collaborative*	*Acknowledged*	*Directed*	**SoS Governance: Centrally Managed**
Inter-relationship: Independent					**Inter-relationship: Subordinate**

Fig. 1. The four types of SoS [10].

belong to the SoS because they share a common goal, or that joining the SoS offers them some gain. Nevertheless, whenever the conditions of belonging to the SoS conflict with their interests, they can leave the SoS. An SoS is an environment where systems reside and systems can join, operate and interact within it [13]. Therefore, to build a SoS, is to create a system that can join an environment, and be an environment to its constituent systems, in such a way as to achieve the SoS goal, as well as its constituent systems'. Both enterprises and network could be used as an inspiration in developing SoS, in that they are systems that combine different independent constituent systems to achieve a goal. This leads to the introduction of SoS architectural frameworks ([13], [6], [12]). An Architecture Framework is "a tool $[\cdots]$ that should describe a method for designing an information system in terms of a set of building blocks, and for showing how the building blocks fit together" [14]. It contains all the rules, processes, methods, and constituent systems that a SoS will use during its lifetime.

III. Intelligent Transportation Systems as SoS

Intelligent Transportation Systems (ITS) are any system or technology that aims to solve a problem, and/or to improve performance in a traffic system [15]. ITS by itself is not a system, but it is the placeholder of myriad of applications, infrastructures, standards, and technologies, that seek to make more intelligent roads, vehicles, and users [16]. The USA, Europe, and Japan are the three main contributors to the ITS. ITS America[1], ITS Japan[2], and ERTICO in Europe[3], are all government organizations that share a common goal: to be a placeholder for any effort that contributes to the ITS [18]. Moreover, car manufacturers with their driving assistance modules, standardization organizations with their standards, and a lot more independent parties that work in any related domain, are all constituents in the ITS [19]. The previously mentioned parties are all independent systems, which have their own objectives, operations, and resources, but they all share the desire to improve the current state of transportation systems. Applying the characteristics in sections II-A and II-B, we could conclude that ITS is a collaborative SoS. Figure 2 shows an example of an intelligent transportation system, with its different component systems.

[1]https://www.itsa.org/
[2]http://www.its-jp.org/
[3]http://ertico.com/

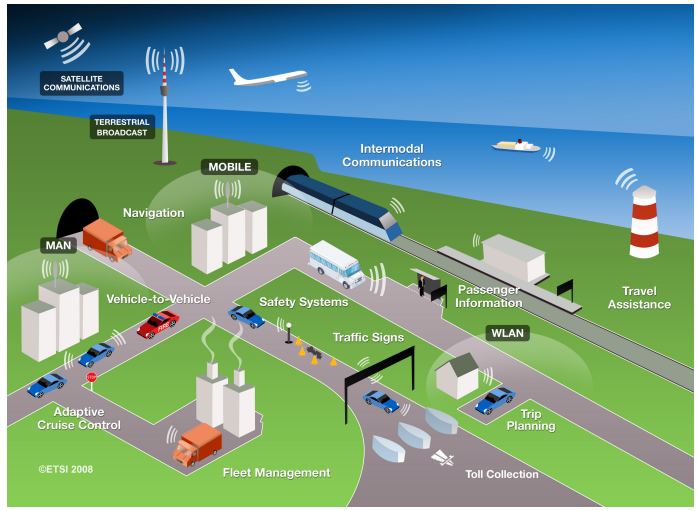

Fig. 2. Intelligent Transportation Systems [20]

A. ITS Categories

The scope of ITS applications varies from gathering information, to providing assistance to transportation users. Based on the different applications and target users, [18] lists major ITS categories: Advanced Traffic Management Systems (ATMS), Advanced Travelers Information Systems (ATIS), Commercial Vehicles Operation (CVO), Advanced Public Transportation Systems (APTS), Advanced Vehicles Control and Safety Systems (AVCSS), Emergency Management Systems (EMS) and more. CVO targets logistics related to commercial vehicles, with applications such as automatic vehicle identification, and automatic vehicle location. ATIS does not target a specific type of users, but rather a specific type of applications, broadcasting real-time information to all users, from vehicles, to infrastructure. Optimal routing guidance and highway advisory radio are examples of ATIS applications.

B. Cooperative Maneuvers Manager for Autonomous Vehicles

The Advanced Vehicles Control and Safety Systems (AVCSS) are all systems that use in-vehicle sensors and communications, in order to provide assistance to the driver, or take control over the vehicle to perform a certain action [16]. Automated Parking, collision warning system, and in-vehicle display system, are examples of AVCSS. To contribute to this category, we propose the Cooperative Maneuvers Manager for autonomous vehicles (CMMAV). CMMAV is a SoS that enables connected autonomous vehicles (CAV) to perform complex lateral maneuvers such as overtaking on highways, in a cooperative way. In an autonomous vehicle, CMMAV takes control over the vehicle in such maneuvers, while in a connected human-driven vehicle, it displays indicators to the driver, to inform them about the best time to execute the maneuver, or otherwise, if it is not safe to do so. The architecture of CMMAV is shown in Figure3: CMMAV is composed of one or more CAV, where each CAV has its

owners and/or passengers. Assuming an internal architecture similar to the one in [17], each CAV that belongs to the CMMAV, integrates the Cooperative Lateral Maneuvers Manager (CLMM) module, which is responsible for lateral maneuvers in CMMAV. CMMAV operates in a subject vehicle as the

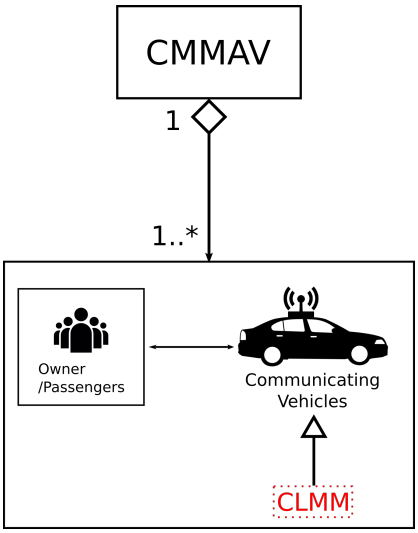

Fig. 3. The architecture of CMMAV: CLMM - Cooperative Lateral Maneuvers Manager.

following: it evaluates some conditions to choose the best maneuver based on the situation. To perform the chosen maneuver, the subject vehicle negotiates with its neighbors through CLMM. After that, CLMM decides on an action (actions) based on the negotiation results, and sends the needed information to the vehicle's modules to perform the chosen actions. The objective of CMMAV is to enable equipped CAVs to perform complex maneuvers on highways in a cooperative way, which leads to reducing travel time, increase on road security, and reducing congestion, amongst others.

C. CMMAV Classification

Taking Maier's characteristics [6] of SoS and applying it to CMMAV, we could say that CMMAV is a SoS for the following reasons:

- CMMAV's constituent systems, the CAVs, are operationally independent since they can operate outside the CMMAV.
- CMMAV's constituent systems are also managerially independent, since they have each their owner, which is not the CMMAV.
- CMMAV's evolution depends on the technical state of its constituent systems, as well as its ecosystem, which are in constant development, which leads to the constant evolution of the CMMAV.
- CMMAV is distributed geographically since its constituent systems are distributed in its place of deployment.
- CMMAV's emergent behaviors are inherited from traffic behaviors (traffic shock wave, accordion effect).

After recognizing that CMMAV is a SoS, we should define its type, using the 3 characteristics from [10]:

- CMMAV's objectives are clear and pre-defined.
- CMMAV is governed by its creators, which are not its constituent systems.
- The interrelationships of its constituent systems are independent from the CMMAV, however, in the context of the maneuvers, interrelationships are defined by CMMAV.

Using Figure 1, we deduce that CMMAV is a directed SoS, in which constituent systems join because they share the same objectives as the CMMAV, even if by doing so, they accept to lose part of their independence.

IV. The CMMAV Framework

A. SoS-ACRE

Systems of Systems Approach to Context-based Requirements Engineering (SoS-ACRE) [21], is a modeling method that uses the needs of all constituent systems and stakeholders in the SoS in order to build a multi-view framework that maps the capabilities, goals, and requirements in a SoS to their respective sources. In this context, needs refer to all capabilities, goals, and requirements that may exist in the system. SoS-ACRE contains 9 views: Source Element View, Requirement Description View, Context Definition View, Requirement Context View, Rule Set Definition View, Validation View, Traceability View, Context interaction view, and validation interaction view. As in [21], we will use Systems Modeling Language (SysML)[4], a general purpose modeling language for engineering systems, to build our framework.

B. Source Element View

The first step in building the framework, is to define its ecosystem, and the actors in this ecosystem, that affect or get affected by the CMMAV. CMMAV's constituent systems'

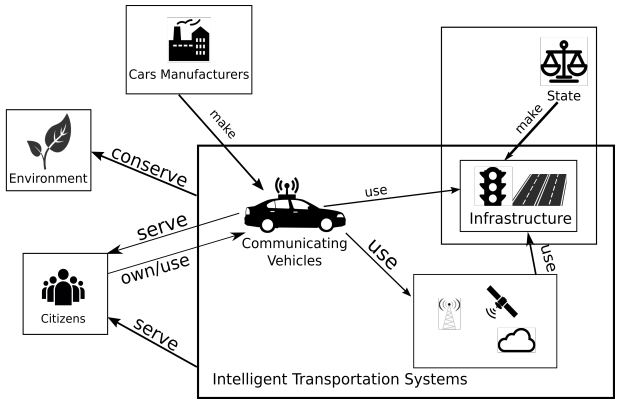

Fig. 4. CAV's Ecosystem

ecosystem is shown in Figure 4. CAVs are part of the ITS SoS, which uses communication technologies, and infrastructures, to serve citizens that use CAV, and to conserve the environment. At the same time, car manufacturers have a

[4]http://www.omgsysml.org/

strong relation to the CAV, since they make them. Each relation of the previous relations should translate into a requirement in the framework. This view contains the sources of all the requirements in the SoS, and by listing the actors in this ecosystem, we could populate this view. Figure 5 shows a part of the source element view of the CMMAV.

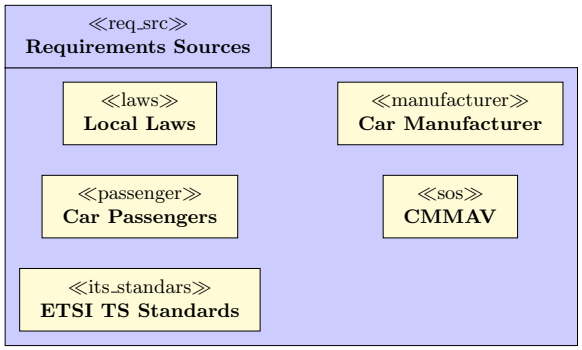

Fig. 5. Source Element View

C. Context Definition View

After recognizing the different requirement sources, it is important to put these sources in context. Any actor related to the SoS could be either a stakeholder, or a component. A stakeholder is any entity related to the SoS, but does not belong to it. While constituent systems are the entities that compose the SoS. Following with our example, the "state" is a stakeholder, since it is responsible for the infrastructure that the CAVs use. And a CAV equipped with the CLMM is a component of the SoS. This distinction is important, because the SoS should treat requirements from the stakeholders, differently than it does with the requirements of constituent systems. For example, SoS must ensure that the requirements that comes from the traffic laws will be met, otherwise, it will not be able to function in its ecosystem. However, if the SoS management decides not to include a certain requirement of the constituent systems in its requirements, it is up to the constituent systems to decide on joining the SoS or not. Figure 6 shows a part of the stakeholders context view in

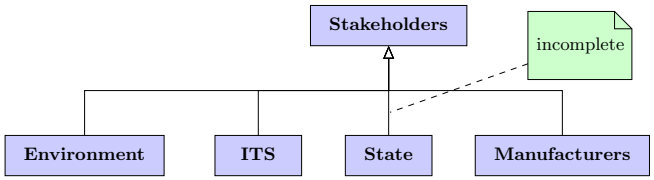

Fig. 6. Stakeholders Context View

CMMAV. The inheritance connector of SySML indicates that the environment, the state, ITS and the manufacturers are all stakeholders in the CMMAV, while the note "incomplete" indicates that this view is incomplete, and more stakeholders might be added.

D. Requirements Description

Each stakeholder, or actor in the SoS environment, has needs, requirements, or goals associated to the SoS. Together, they form the source elements to the SoS requirements. The requirements of these sources must be clear to both the SoS and the elements, so a description of each requirement must be formulated. Let's take the CMMAV example, and the requirements of the "Local Traffic Laws" source element from Figure 7. Each requirement has its own unique identifier, a name, and a text description.

Using this approach, a complete and clear requirements description is available to the SoS management, as well as any interested actor. Each requirement is traced to its source from the source elements view. The requirements description view ensures that the SoS is using the right requirements from the source elements, and it is the source to the next view: The rule set definition view.

E. Rule definition View

After describing all the requirements in the SoS, each requirement must have a rule (or set of rules) that describe how this requirement will be met in the SoS. Each rule is traced as well to its requirement source, which in turn is used to trace the rule to the source element from the source element view. A requirement might have several rules to ensure that it is met. Continuing with the previous example, the requirement "Respect Speed Limits" requires two rules to ensure it is met (Figure 8): where ES refers to the ego-vehicle's speed, $MinLS$ and $MaxLS$ are the minimum and maximum allowed speeds on a given lane respectively. These two rules ensure that the CLMM does not send a speed reference to the subject vehicle that is not allowed by the law requirements, even if that vehicle contains another module that controls its speed based on the speed limits. These two rules, applied in the CMMAV, ensure that the SoS respects the requirements of the local traffic laws.

F. Requirement Context View

Each requirement in the SoS traces back to a source, and each requirement involves different actors, processes, and resources to be met. To trace behaviors to their sources in the system, or analyze a specific action in the system, it is important to show the context of each requirement, on what other requirements it depends, and who is responsible for meeting it. Continuing with our previous example, we take the "Respect Speed Limit" requirement. Figure 9 shows one of the contexts in which this requirement is used in the system. This use-case diagram reads as follows: The requirement "Overtake When Possible", from the CMMAV source, extends the "Reduce Travel Time" requirement, from the Owners/Passengers source, which in turn is composed from multiple requirements, one of which is "Use Maximum Speed", which comes from owners/passengers as well. Finally, "Respect Speed Limits" constrains the "Use maximum speed" requirement. This context allows us to see how requirements

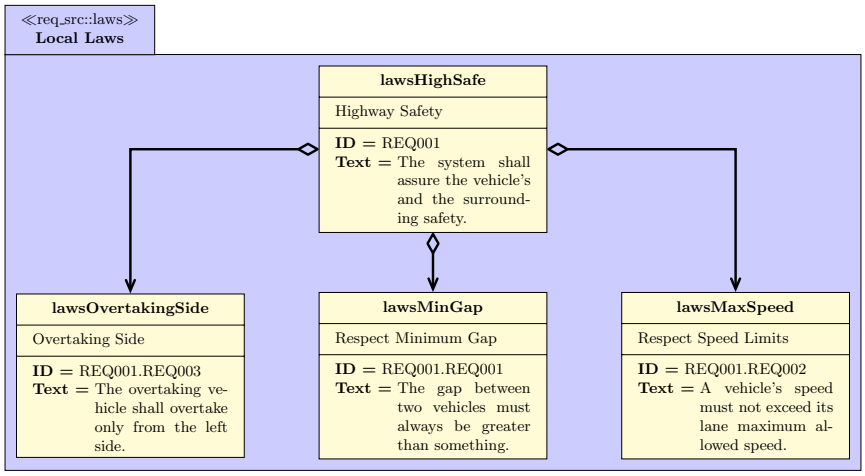

Fig. 7. Local traffic laws' requirements

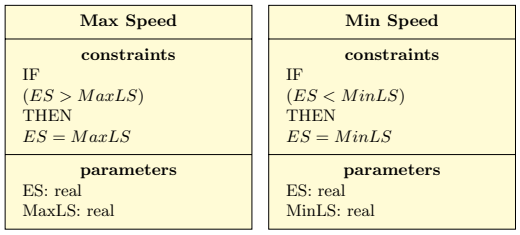

Fig. 8. Speed Limits Rules

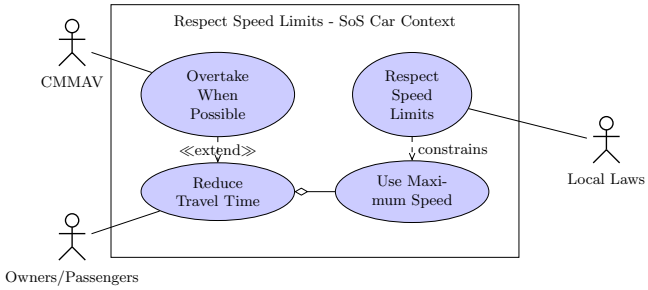

Fig. 9. Respect Speed Limits Context

relate to each other, and enables us to see the relation between two seemingly unrelated requirements.

G. Traceability View

So far we have identified all sources in the CMMAV, then from these sources, we extracted requirements that need to be met in the system. Afterwards we defined the rules that constrain, or define the way requirements will be integrated in the system, followed by use cases that describe each requirement in the context of the different associated stakeholders. The next step is to trace all the elements of the framework to their sources: requirements to source elements, rules to requirements, and so on. Some of the relations in this view already exist implicitly in different views. For example, in Figure 7, two types of relations exist: the requirement-source relation, expressed by separating the requirements in different packages that belong each to a unique source. And the requirement-requirement relation, by using a composed identifier in child requirements that contains their parent's identifier. The objective of the traceability view is to show all the relations explicitly in a dedicated view. Different types of relations exist: rule-requirement, use case-requirement, etc. These relations may be represented by tables, SysML graphs, or any tool that helps with this objective. Table I shows the requirement-rule relations for the requirement "Respect Speed Limits". This view could be used later on whenever a new element is added to the framework, to determine what it affects, and what needs to be modified.

TABLE I
RULE-REQUIREMENT TRACEABILITY TABLE.

Requirement ID	Requirement Name	Refined by
REQ001.REQ002	Respect Speed Limits	Rule::Max Speed
REQ001.REQ002	Respect Speed Limits	Rule::Min Speed

H. Validation View

The objective of this view is to ensure that all requirements are really met in the system, by visualizing the different use-cases in different scenarios. The realization of this view is part of future work, where we use simulations to validate that our system respects its requirements.

I. CMMAV and SoS Challenges

Table II shows some of the challenges facing SoS, and how they are handled in the CMMAV.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented systems of systems, their definitions, and types. We then introduced architecture frameworks,

TABLE II
SoS CHALLENGES AND PROPOSED SOLUTIONS.

Challenge	Proposed Solution
Stable Intermediate Forms	In the stages that SoS-vehicles share the roads with other non-SoS-vehicles, SoS-vehicles use their on board sensors to detect non-SoS-vehicles, and show their intentions by using external indicators such as signal lights, just as any normal vehicle would do, to assure that the system could be integrated in its environment in every stage of development.
Emergent Behavior	To deal with shock wave and accordion effect, SoS vehicles do not request cooperation from their neighbors when they are answering to a request from another vehicle.
Policy Triage	Every vehicle that joins the system must meet its requirements (requirements that come from the CMMAV in the requirement definition view).
Evolutionary Development	Any new objective for the CMMAV, could be easily integrated by adding modules such as CLMM. And any new requirement that comes from its ecosystem, could be integrated into the CMMAV and broadcasted to its constituent systems using the cloud.
Heterogeneity of constituent systems	Different manufacturers means different internal architectures, to account for this, the CLMM contains an internal communication block that has the responsibility of communicating with the vehicle's different modules. This means that for each new vehicle's architecture (different manufacturer) that enters the SoS, it is enough to adapt the communication block for this specific vehicle's architecture to be able to communicate with other SoS vehicles.

a method that could be used to describe an SoS from its different aspects. We then introduced intelligent transportation systems, a SoS composed from different component systems that are by themselves SoS. We then proposed the CMMAV, a cooperative maneuvers manager for autonomous vehicles, as our contribution to ITS. CMMAV is a SoS that manages the different maneuvers of autonomous vehicles on highways in a cooperative way. It is integrated in autonomous vehicles via a module called cooperative lateral maneuvers manager (CLMM). CMMAV takes into account the presence of non-autonomous vehicles in its environment. To build it, we used SoS-ACRE, a context-based requirement engineering modeling method, that uses requirements from all of the system's stakeholders and constituent systems, to build a SoS that could be integrated into its ecosystem. In future work, the system will be tested using autonomous vehicles of the Heudiasyc laboratory[5] [22], to validate the framework on a local level between a couple of vehicles. To analyze the effect of the system on a global level, *i.e.* on a highway between multiple vehicles, we will simulate the different scenarios using an adapted simulation software.

ACKNOWLEDGMENT

This work was carried out and funded in the framework of the Labex MS2T. It was supported by the French Government, through the program Investments for the future managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02).

[5]www.hds.utc.fr

REFERENCES

[1] US Department of Defense, "The DoDAF Architecture Framework," Version 2.02, September 2010.
[2] M. Jamshidi. "Systems of Systems Engineering: Principles and Applications," CRC Press, Ch. 1, Nov. 2008.
[3] D. Luzeaux. "Systems of Systems," John Wiley & Sons, Inc. 2013.
[4] P. G. Carlock, and R. E. Fenton, "Systems of Systems (SoS) enterprise systems for information-intensive organizations," Systems Engineering 4(4):242-261, 2001.
[5] R. S. Pei, "Systems of Systems Integration (SoSI) - a smart way of acquiring Army C4I2WS systems," Proceedings of the Summer Computer Simulation Conference, Monterey Bay, CA, 2000.
[6] M. W. Maier,"Architecting Principles for Systems-of-Systems," INCOSE Sixth annual International Symposium of the International Council on Systems Engineering. Volume 6, pp. 565573, Boston, MA, 1996.
[7] J. Boardman and B. Sauser, "System of Systems - the meaning of of," IEEE/SMC International Conference on System of Systems Engineering, Los Angeles, CA, 2006.
[8] R. Abbott, "Open at the Top; Open at the Bottom; and Continually (but slowly) Evolving," IEEE/SMC International Conference on Systems of Systems Engineering, Los Angeles, CA, 2006.
[9] J. S. Dahmann and K. J. Baldwin, "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering," IEEE Systems Conference, Montreal, Quebec, pp. 1-7, 2008.
[10] B. Collins, S. Doskey, J. Moreland, "Relative Comparison of the Rate of Convergence of collaborative Systems of Systems: A Quantified Case Study," Presented to the System of Systems Engineering Collaborators Information Exchange, George Washington University, 2016.
[11] G. Rebovich, "Systems of Systems Engineering: Principles and Applications," CRC Press, Ch. 6, Nov. 2008
[12] A. Gorod, B. Sauser, and J. Boardman, "System-of-Systems Engineering Management: A Review of Modern History and a Path Forward," IEEE Systems Journal, vol. 2, no. 4, December, 2008.
[13] C. Nielsen, P. Larsen, J. Fitzgerald, J. Woodcock, and J. Peleska, "Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions," ACM Comput. Surv. 48, 2, Article 18, September, 2015.
[14] The Open Group Architecture Framework, http://pubs.opengroup.org/architecture/togaf8-doc/arch/.
[15] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," Communications Surveys Tutorials, IEEE, vol. 13, no. 4, pp. 584-616, 2011.
[16] L. Figueiredo, I. Jesus, J. A. T. Machado, J. R. Ferreira and J. L. Martins de Carvalho, "Towards the development of intelligent transportation systems," ITSC 2001. IEEE Intelligent Transportation Systems, Oakland, CA, pp. 1206-1211, 2001.
[17] J. Straub et al., "An internetworked self-driving car system-of-systems," 12th System of Systems Engineering Conference (SoSE), Waikoloa, HI, pp. 1-6, Jun. 2017.
[18] Yen-Chun Jim Wu, Pi-Ju Lee, "The use of patent analysis in assessing ITS innovations: US, Europe and Japan," Transportation Research Part A: Policy and Practice, vol. 41, no. 6, pp. 568-586, 2007.
[19] K. Dellios, C. Patsakis and D. Polemi, "Automobile 2.0: Reformulating the Automotive Platform as an IT System," in IT Professional, vol. 18, no. 5, pp. 48-56, 2016.
[20] ETSI ITS, "Standards on the move", 7th European Congress on ITS, Geneva, 2008.
[21] J. Holt, S. Perry, R. Payne, J. Bryans, S. Hallerstede and F. O. Hansen, "A Model-Based Approach for Requirements Engineering for Systems of Systems," in IEEE Systems Journal, vol. 9, no. 1, pp. 252-262, March 2015.
[22] P. Xu, G. Dherbomez, E. Hery, A. Abidli and P. Bonnifait, "System Architecture of a Driverless Electric Car in the Grand Cooperative Driving Challenge," in IEEE Intelligent Transportation Systems Magazine, vol. 10, no. 1, pp. 47-59, 2018.