IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 23, 2019, accepted August 12, 2019, date of publication August 26, 2019, date of current version September 16, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937453

A Methodology for the Design of Safety-Compliant and Secure Communication of Autonomous Vehicles

ROBERTO PASSERONE[1], (Member, IEEE), DANIELA CANCILA[2], MICHELE ALBANO[3], (Senior Member, IEEE), SEBTI MOUELHI[4], SANDOR PLOSZ[5], ERKKI JANTUNEN[6], ANNA RYABOKON[7], EMINE LAAROUCHI[2], CSABA HEGEDŰS[8], AND PAL VARGA[5], (Member, IEEE)

[1]Dipartimento di Ingegneria e Scienza dell'Informazione, University of Trento, 38123 Trento, Italy
[2]CEA, LIST, CEA Saclay, F91191 Gif-sur-Yvette, France
[3]Department of Computer Science, Aalborg University, 9220 Aalborg, Denmark
[4]ECE Paris.Lyon—École d'ingénieurs, INSEEC U., F75015 Paris, France
[5]Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, 1111 Budapest, Hungary
[6]VTT Technical Research Centre of Finland Ltd., 02044 Espoo, Finland
[7]TTTech Computertechnik AG, 1040 Vienna, Austria
[8]AITIA International Inc., 1039 Budapest, Hungary

Corresponding author: Daniela Cancila (daniela.cancila@cea.fr)

This work was supported in part by the EU ECSEL JU through the H2020 Framework Programme, within project JU (PRODUCTIVE 4.0, www.productive40.eu)—and its National co-funding schemes, under Grant 737459, and in part by the Higher Education Excellence Program of the Ministry of Human Capacities, Hungary, in the frame of Artificial Intelligence Research Area of Budapest University of Technology and Economics (BME FIKP-MI/SC).

ABSTRACT The automotive industry is increasing its effort towards scientific and technological innovations regarding autonomous vehicles. The expectation is a reduction of road accidents, which are too often caused by human errors. Moreover, technological solutions, such as connected autonomous vehicle platoons, are expected to help humans in emergency situations. In this context, safety and security issues do not yet have a satisfactory answer. In this paper, we address the domain of secure communication among vehicles – especially the issues related to authentication and authorization of inter-vehicular signals and services carrying safety commands. We propose a novel design methodology, where we take a contract-based approach for specifying safety, and combine it in the design flow with the use of the Arrowhead Framework to support security. Furthermore, we present the results through a demo, which employs model-based design for software implementation and the physical realization on autonomous model cars.

INDEX TERMS Contract-based approach, arrowhead framework, security and safety co-design, autonomous vehicles, heterogeneous design.

I. INTRODUCTION

The trend towards the adoption of autonomous vehicles has been tremendously increasing in the last few years, with the expectation of a significant reduction of road accidents, increased fuel economy and an overall higher traffic throughput [1]–[3]. Today, road tests of autonomous vehicles are in place in several countries, such as those from Uber in Pittsburgh [4] and from Valeo in France [5]. For example, in the Valeo test, an autonomous car drives on the Paris beltway, automatically adapting its speed to traffic conditions. During the road tests, the French authorities demand a human driver supervisor (in the car), whose duty is to act in emergency situations. Their presence also reassures other drivers, who do not realize that the car is an autonomous vehicle. The objective, however, is to eventually reach full autonomy, increasing both safety and average throughput, using appropriate distributed control techniques.

A careful examination of road tests reveals several open problems both at the scientific and technical level, as well as at the economic and social level. Clearly, traffic control is best tackled through a cooperative approach, where vehicles

The associate editor coordinating the review of this article and approving it for publication was Lei Shu.

exchange information to jointly build a detailed picture of the current situation [6], [7]. This, in turn, requires an efficient and *secure* vehicle-to-vehicle (V2V) communication mechanism to exchange information (e.g., an obstacle on the road) [8], [9]. At the same time, the heterogeneous nature of the problem, and its fast evolution, demand a standardized infrastructure and a design methodology by which designers can unambiguously formulate the requirements and the properties of the system, to deliver functional as well as non-functional *safety* guarantees [10]. This is essential to drive the adoption of the technology, and lower the risk perceived by the user. For this reason, we are witnessing an increasing demand in considering both safety and security properties together [11], [12]. Nonetheless, this trend is hampered because those properties are specified, analyzed and developed by different teams having different backgrounds and tools. Moreover, in many critical applications, safety properties shall be compliant to specific safety standards for system certification.

The main novelty and contribution of our work is to provide a methodology that starting from natural language requirements reaches the prototyping stage of a platooning autonomous vehicle system, with an additional focus on safety and security requirements. Moreover, the proposed methodology is compliant with the ISO26262 [13] safety norm. A newer norm called ISO/PAS 21448:2019 *Road vehicles – Safety Of The Intended Functionality* (SOTIF), addressing the different levels of autonomy especially in emergency intervention systems, was recently released; SOTIF is not discussed in this paper but, being an extension of ISO26262, our work can be made compliant with it as future work. We show that a secure communication protocol between autonomous vehicles can raise conflicts with the safety requirements and decrease the overall safety level of the system. Thanks to the Arrowhead Framework [14] we are able to select a suitable security algorithm compatible with the safety requirements. In this work we consider vehicles pertaining to a single stakeholder; an extension to multiple stakeholders could use blockchains technology at platoon creation time, and then progress as per this paper.

In this context, our research work focuses on delivering and combining these two aspects.

First, we address the problem of specifying and verifying the correct and *safe* behavior of the system by means of a contract-based approach [15], which separates assumptions from guarantees, partitioning the properties across the various components and teams. This method is particularly well suited to the distributed nature of the application, where several vehicles contribute to ensure a positive final outcome. We integrate contracts with well known standards, such as SysML, to facilitate their adoption in traditional design flows, and with three dimensional animated scenarios, to offer a better understating of the cyber-physical system being analyzed.

Second, we combine this technique with a Service Oriented Architecture (SOA) based on the Arrowhead Framework [14] to manage all V2V communication activities, ensuring *security*. This approach provides all the benefits of SOAs (e.g., service reuse and composition, loose coupling of functionalities) [16] enhanced by the capabilities of the Arrowhead Framework (e.g., service discovery, orchestration, security).

Moreover, similarly to the contract-based approach, streamlining all interactions by means of formally-defined service interfaces allows easier application to distributed systems. This methodology is supported by a dedicated tool, called CAT (Contract Analysis Tool), which integrates the different parts and drives the analysis engine. We present our results using a physical demonstration that employs model cars [17].

The paper is structured as follows. First, we review related work and discuss background material in Section II. Then, Section III provides the core of the promoted design methodology and discusses its compliance with safety standards. Section IV constitutes the bulk of our study, and deals with the technical details of the methodology. This includes surveying heterogeneous tools and languages, to be used for a platooning vehicle system, from the early specification to the physical realization on autonomous model cars. The analysis of the results is introduced in Section V. Finally, Section VI provides an extended outlook of the social and economical impact of the proposed methodologies, and summarizes our conclusions.

II. RELATED WORK AND BACKGROUND

Developing safe and secure communication systems for autonomous vehicles is a challenging problem that has been extensively studied in the literature. Willke et al. present the main techniques and their applications [18], while more recently Dressler et al. discuss research directions for new generation protocols [19]. Notably, the ETSI TC ITS working group leads the European standardization activities to a secure V2V communication [20]. Among the main concerns, the group addresses *trust* of the communication, exchange formats for messages and protocols, as well as the impact of technological design to guarantee privacy issues. To guarantee *trust*, vehicles receive a certificate by a common authority. To guarantee *privacy*, vehicles receive a pseudonym, which is regularly changed to reduce the risk of full traceability.

To manage safety, some protocols use periodic messages to provide general information (such as the position of the vehicle and velocity) and event-trigger messages to communicate safety information (e.g., accident in position XYZ) [21]. Most of the existing approaches address safety issues by considering the impact of security protocol communication and the guarantee of privacy. Indeed, the time needed to manage both privacy and security policies could impact the safety commands execution [8], [21], [22]. This decreases the safety level of a vehicle – thus generating a conflict with the ISO26262 standard [13]. In particular, collision avoidance can suffer from the adopted communication protocol [8] as well the adopted pseudonym exchange policy (silent period) [23], [24]. Considering these existing works, we directly address safety and security issues at the start of the design phase. Safety properties, including collision avoidance, are specified by contracts, while security is ensured by adopting the Arrowhead Framework, which supports a Service Oriented Architecture (SOA) and promotes a separation of concerns between services and communication protocols. Like the Arrowhead Framework, SEROSA [25] is based on SOA principles and supports security and privacy solutions for vehicular communication. SEROSA introduces two different protocols: a protocol for the service acquisition (authentication) and a protocol for pseudonym resolution and revocation. The main advantage is that SEROSA does not need to initialize the system if an authentication command is sent. On the other hand, the Arrowhead Framework is becoming a *de facto* European standard platform for several applications domains.

The aim of our work is to extend the support of Arrowhead towards safety issues handled with *contracts*. Aligned with current practices [9], [26], [27], Arrowhead makes use of certificates to implement secure communication. One of the key problems, also raised by the literature described above, has to do with the delay introduced by the security key exchange process, which may have adverse safety effects. In the contract-based extension, we use specific constructs in the safety assertions to formally define the temporal constraints, and then take them into account during simulation and verification to validate the design. The service oriented architecture implemented using Arrowhead services uses certificates to generate security tokens [28], which are subsequently used to provide lightweight security to the actual communication. This approach is particularly effective in our vehicle platooning case study, since we perform the more time consuming actions early during platoon creation, to benefit later of a secure communication channel with low overhead. Moreover, being the security tokens temporary, they can also limit the danger of security key leakage: after being detected as malicious, a system would still be able to authenticate itself by means of its certificate, but then it would not be authorized by the Arrowhead Framework and its security token would not get renewed.

A. CONTRACT-BASED DESIGN

To reduce the complexity of system design, relevant standards, such as AUTOSAR [29], define a multi-layered abstraction framework resulting in a *complete decoupling* of automotive software functions starting from the underlying hardware controllers. However, to ensure safety requirements, a failure hypothesis in these frameworks must often be propagated over multiple layers, breaking the principle of separation of concerns.

To solve this dilemma, we adopt a contract-based approach, which abstracts from the particular implementation of software and hardware components of the designed communication system by defining *safety assumptions* and *promises* of these components [30]. By replacing real implementations with contracts, developers can achieve a significant reduction of the system design and implementation time. In particular, we use contracts in this paper to construct a specification that crosses the boundaries between vehicles, while retaining their separation through the use of assumptions and guarantees. The application of the contract-based approach can be facilitated by the adoption of formalized languages enabling modeling and execution of failure propagation models [31], [32]. In this work, we adopt the pattern-based language BCL [33], which is oriented to *semi-formal* simulation-based methods that proved to be able to handle real-world systems efficiently, in contrast to approaches based solely on formal methods [34]. In addition, BCL can be integrated with Matlab/Simulink, and has been shown to provide an efficient path to dependability assessment for service oriented specifications (SOA), which is at the basis of this work [35]. The formal semantics of BCL is defined by mapping the patterns into Linear Temporal Logic (LTL) [33]. This mapping also serves as the contract implementation that we use when we want to analyze the specification during simulation. While we could conceivably use LTL directly for contract specification, BCL offers a simpler and more constrained formalism, which helps avoid mistakes and is more easily mastered by designers not familiar with formal assertions.

B. ARROWHEAD FRAMEWORK

We adopt the Arrowhead Framework [14] to specify, analyze and ensure the *security* properties of the system, such as the authentication and authorization of the control-command signal. In Arrowhead, all interactions are mediated by means of *services*, produced and consumed by *systems*, and executed on *devices*. The latter can manifest themselves as any kind of provider of computational capabilities, e.g., an embedded system, a laptop, or a virtual machine in the cloud. A distributed application is implemented on top of a *System of Systems*, which is a set of systems that interact in an Arrowhead-compliant manner. Services are offered as a set of functions that can be invoked remotely, and communication can be mediated by different communication technologies, as Arrowhead is agnostic to the underlying protocols. This comprises the format encoding the messages (e.g., JSON, XML, MQTT), the communication protocol (e.g., HTTP, HTTPS, XMPP) and the communication paradigm (e.g., REST, publish/subscribe).

Arrowhead normalizes all interactions by means of SOA, and thus enables interoperability between systems that are natively based on different technologies. This approach simplifies software development and maintenance activities, thus reducing dramatically time-to-market, supporting the deployment, and improving maintainability of interconnected cooperative applications. Finally, being service-oriented, applications enjoy advantages of loose coupling in space, time and synchronization [36].

Arrowhead services are divided into *application services* and *core services*. Application services implement the functional requirements of the use cases, while core services manage the application of non-functional requirements and the platform itself. The core services provide tangible added value for the users of the Arrowhead Framework. Some services are "mandatory" and need to be present in any system of systems design, others are "supporting", helping designers and integrators focus on the main functionality of their system, and leave other utilities to the framework.

The mandatory core services are *ServiceRegistry*, *Orchestration*, and *Authorization* (see Figure 1). In order to support deeper understanding of the technical background regarding our solution, the following paragraphs provide an overview of these core services.

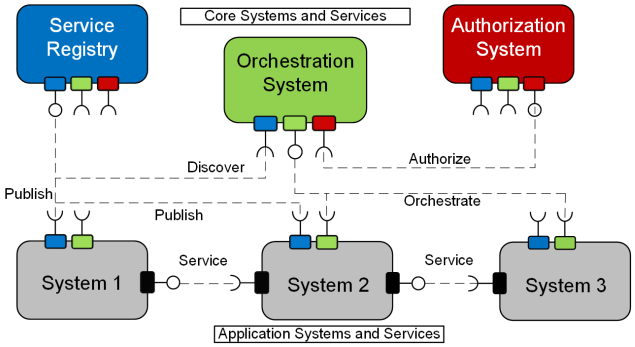


FIGURE 1. **A generic System of Systems in the Arrowhead Framework. The colors of the connectors represent the service producer system.**

ServiceRegistry is used to keep track of all services and systems active in the system of systems. In particular, each system publishes the offered services through the ServiceRegistry, acting as the entry point which is then interrogated by other systems to perform the discovery of services and systems.

Orchestration, in the context of SOA [16], is the process that defines how systems get interconnected and provide their services to other systems. This operation is executed by systems wanting to consume a service, and it allows the re-use of existing services and systems to create new services and functionalities [37]. The Orchestration services makes use of ServiceRegistry to acquire a list of systems and services. It then computes a match between the systems requiring services, the service requests, and the systems that can provide them.

In an Arrowhead-compliant system of systems, a service can only be accessed by an authorized consumer. Authentication is managed by means of X.509 certificates [38], and authorization is performed through the Authorization service. Systems contact the service with their credentials and use it to authenticate themselves, and to consequently be able to access other systems for service fruition. The implementation used in this work is token-based authorization provided by the Arrowhead Framework: during orchestration, the service consumer receives an authorization token that validates it and allows for further interactions with the service provider.

III. OVERVIEW AND METHODOLOGY

In this section, we discuss the steps that we use in our methodology to make the analysis systematic, so that it can be implemented in a usable and structured design process. The case study is intended to validate and evaluate the different phases of the methodology. Because of the strong dependencies of our case study on physical factors, such as communication latency and sensors, and the degree of autonomy afforded by the software control, we operate in the domain of Cyber-Physical Systems. For this reason, we propose an inclusive approach based on heterogeneous languages and tools, including different functionalities and teams. The main advantage is an open and modular design flow that affords flexibility and adaptability to internal languages, tools and processes, adopted by the industry.

Our approach, shown in Figure 3, is composed of a series of steps supported by CAT (Contract Analysis Tool) and by the Arrowhead Framework. In the figure, the main steps of the methodology are identified by blocks, to which we associate the input and output information (shown on the left and on the right, respectively). As expected, in some cases the output of one block is the input for the next phase, as is often the case in the validation steps. The steps shown in the figure also refer to where the design phase applies with respect to the feasibility study (Section IV) and the experimental analysis on the safety and security integration (Section V). Figure 2 shows the underlying tool architecture designed to support the logical steps of the methodology. The designer input is managed by CAT for design and verification, which in turn relies on Arrowhead for the runtime environment. In the rest of this section, we describe each step.

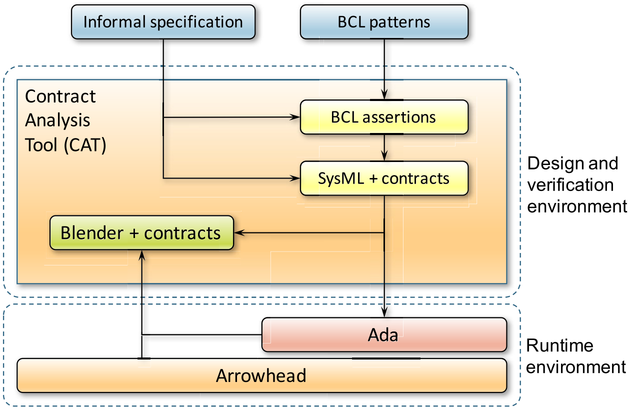


FIGURE 2. **Tool architecture, showing the information flow between design and verification in CAT and the runtime environment.**

A. CONTRACT SPECIFICATION IN BCL

The activities begin with an informal identification of the user needs, expressed mostly in natural language, focusing on the safety and security aspects.

To make them usable for analysis, these requirements must be formalized, together with an initial specification of the system architecture. For the first part, we adopt a contract-based

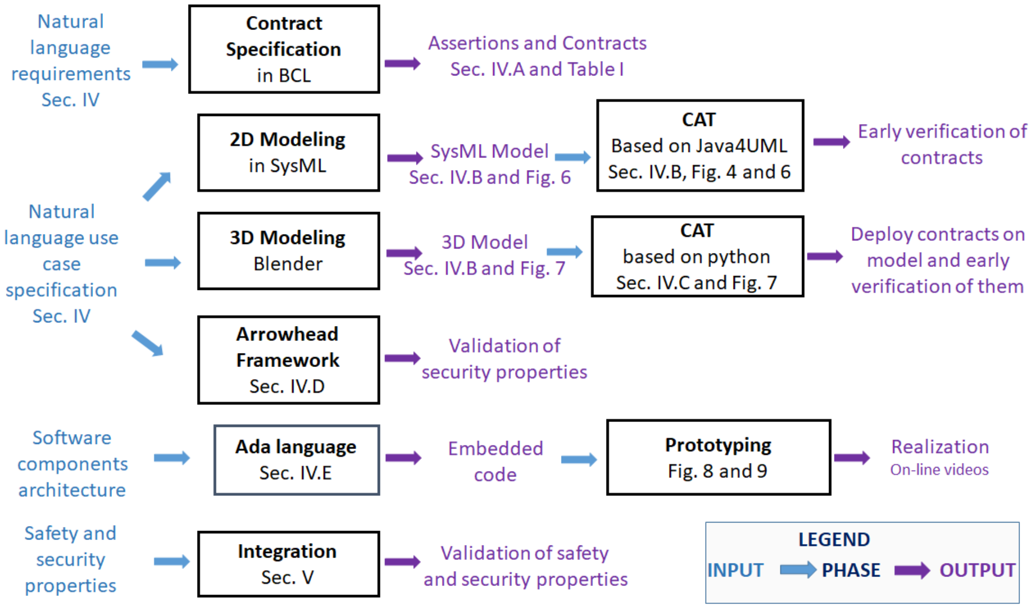

Natural language requirements Sec. IV
Contract Specification in BCL
Assertions and Contracts Sec. IV.A and Table I
2D Modeling in SysML
SysML Model Sec. IV.B and Fig. 6
CAT Based on Java4UML Sec. IV.B, Fig. 4 and 6
Early verification of contracts
Natural language use case specification Sec. IV
3D Modeling Blender
3D Model Sec. IV.B and Fig. 7
CAT based on python Sec. IV.C and Fig. 7
Deploy contracts on model and early verification of them
Arrowhead Framework Sec. IV.D
Validation of security properties
Software components architecture
Ada language Sec. IV.E
Embedded code
Prototyping Fig. 8 and 9
Realization On-line videos
Safety and security properties
Integration Sec. V
Validation of safety and security properties
LEGEND
INPUT
PHASE
OUTPUT

FIGURE 3. The steps of the design flow, from specification to prototyping.

approach, which can neatly distinguish between the responsibilities of different components in the form of assumptions and guarantees, expressed as *assertions* in the pattern-based BCL language [33]. BCL is convenient, because the natural language requirements map logically to the patterns, which add structure and help avoid the most common mistakes. The contracts expressed by the assertions are traced, measured and validated along the design flow, to help address cross-layer issues. The architecture and functionality of the design are instead specified in SysML, an OMG standard which is broadly adopted in the industry, using the Eclipse IDE.

B. 2D MODELING

The use case, generally initially expressed in natural language, is modeled using a component-based formalism. In this work, we adopt SysML, because it is a widely used OMG standard, supported by industrial tools (often, but not always) available via open-tools (e.g., IBM, PTC/Artisan Studio, Softeam, Papyrus). In Figure 3, the SysML model is the output of the modeling phase and the input for CAT.

C. CAT - IMPLEMENTATION VERSION FOR ECPLISE

The contracts and the system architecture are brought together in a combined SysML specification, where the individual assertions are associated to the ports of the components, while contracts are associated to the components themselves. This makes assertions reusable across different entities in the design, an essential feature to make programming more efficient and to prevent errors. The integration and annotation of the SysML specification with contracts can be accomplished in several ways. In our methodology, CAT defines a meta-model and a domain-specific language extending SysML with the appropriate notions of `contract`, `guarantee` and `assumption` as UML Metaclass constraints (see Figure 4). This choice is based on industrial requirements, provided by early feasibility tests in the railway application domain [39], where the properties were expressed as OCL constraints. CAT functions as an add-on Eclipse plug-in and is based on Java4UML. Because assumptions and guarantees may belong to several different contracts, they are represented as `stereotype`.

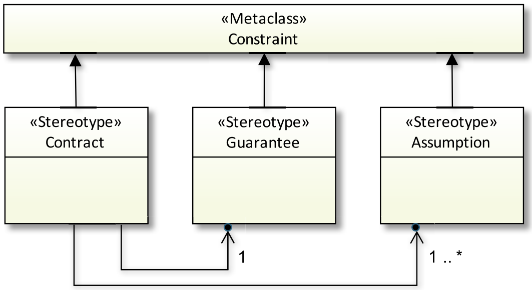


FIGURE 4. Domain-Specific Language interface between BCL and SysML/UML.

D. 3D MODELING

One distinctive feature of our methodology, which is particularly useful in distributed and dynamic scenarios such as autonomous vehicles, is the ability to simulate the system through a three-dimensional animated model. A 3D representation can graphically simplify complex problems and offers a better understanding of the issues being analyzed. We choose Blender as modeling tool because it is free and open-source, and provides flexible interfaces for integrating external tools. Moreover, Blender contains many features that are characteristic of high-end 3D software. The feature that most interests us is the Python scripting tool for creation and prototyping, task automation and custom tools. The built-in Python console and text editor makes it simple to integrate scripts and customize the software at will. The 3D model is the output of this phase and the input for CAT.

E. CAT - IMPLEMENTATION IN PYTHON

CAT is used through the Python interface of Blender to extend the 3D model with contracts where the assertions are expressed in LTL, the underlying formal semantics of BCL [33]. CAT then verifies the validity of the contracts for each step of the 3D simulation, providing evidence of the correct and safe behavior of the system or, in case of a negative result, detecting the contract violations.

F. ARROWHEAD FRAMEWORK

The specification of security aspects follows a parallel bottom-up path, and is grounded on and assisted by the Arrowhead Framework. The design paradigm is to separate the application protocol from the underlying infrastructure services that implement the communication primitives. We consider that communication is articulated into three steps (discovery of recipient, authorization, data exchange) and that each communication primitive must take care of all of them, either in advance (e.g., authentication during handshake) or on demand (e.g., authentication done when sending a message). SOA allows the use of services for all of these operations. In the particular case of the Arrowhead Framework, different implementations can be provided for the communication primitives, for instance to compare different approaches and scenarios. For example, both the discovery and the authentication steps can be done in advance when targeting the same recipient, for performance sake. Moreover, different protocols can be used for each step, for example to grant different degrees of security to each message.

Security imposes certain requirements on the system behavior, such as employing defined processes for data exchange or an appropriate communication set up, which may lead to an unexpected violation of a contract. The communication primitives taken into account by the safety contracts are considered abstract operations, and their interpretation as `guarantees` is as lazy as possible to delay their resolution to actual delays. The timing analysis is thus articulated into two steps. The first step considers the application protocol and builds an expression that expresses the timing `guarantees` of the protocol. Later on, the implementation of the primitives are taken into account, to associate delays to their executions. The values used for the delay associated to each implementation of a primitive are computed by either formal proofs or by means of benchmarking.

G. INTEGRATION

In our methodology, the model in Blender is therefore annotated with the additional performance information considered for each primitive, which are later on resolved to time delay gathered from the system prototype. The final step is to perform an integrated verification. This can be accomplished by checking the consistency of each component implementation with the corresponding contract. If satisfied, and if no violation was detected in the previous CAT analysis, then the contract theory guarantees correctness by compositional rules [30]. While potentially conservative, the advantage of this strategy is that contract implementation can be checked separately on each component. Otherwise, the CAT analysis can be repeated with refined contracts and performance values, to provide an overall integrated verification. In particular, in our case study, we evaluate both the abstract behavior and the measured performance to tackle the interaction between safety contracts and security.

H. ADA

The following step consists in the development of the actual embedded software. While this step is still manual in our design flow, the experience and results obtained through the early model evaluation are instrumental to implementing code that is well structured and satisfies the constraints. A model transformation tool is planned for our future work. We target the Ada language, because of the guarantees that it can provide in the domain of critical systems and its widespread adoption by the industry.

I. COMPLIANCE WITH ISO26262

The combined use of the Arrowhead Framework and the contract-based approach is consistent with the directives of safety international standards, such as ISO26262 [13] and CENELEC [40]–[42] in the automotive and railway domains, respectively. Compliance with the standards can be determined by satisfying one or more objectives in the norms and by non-interfering with pre-existing industrial processes. Our methodology is to some extent independent of the particular languages employed, letting companies integrate the approach in their internal processes, including the safety management policy (e.g., ref. [13], Part II, Clauses 5, 6 and 7) and (qualified) tools, and eventually supporting the arguments to be presented to the certification authority. Clause 5 aims "to develop a description of the item with regard to its functionality, interfaces, environmental conditions, legal requirements, known hazards, etc. The boundary of the item and its interfaces, as well as assumptions concerning other items, elements, systems and components are determined".

With no intention of settling the issue but only to suggest a possibly simplistic strategy for that compliance, we use contracts as a means to specify the safety and security properties that the systems should meet on the system's and components' interfaces. We model the system under examination in 2D (via SysML), where components, functionality and requirements are specified. Then we simulate the system in 3D (via Blender). This simulation provides an immediate visualization of the overall system and wished properties.

We use CAT (Contract Analysis Tool) to specify and verify safety-related properties on both 2D and 3D system modeling (technical details are provided in Section IV). Security properties are firstly specified via contracts, then modeled and tested using data extracted via experiments with the Arrowhead Framework. Finally, we use the Ada programming language to develop the embedded code. The promoted methodology does not constrain industries to adopt a predefined language, i.e., industries can adopt their modeling and embedded programming languages as well as their know-how. Instead, the promoted methodology helps industries provide a clear and structured definition of a component and of its dependencies and/or interactions with the environment and other components, and therefore it meets Objective 5 (“The objective of this clause is to define the requirements for the organizations that are responsible for the safety lifecycle, or that perform safety activities in the safety lifecycle”) and Clause 5 of ref. [13].

A complete study of compliance is on-going. Moreover, our future work aims to address ISO/PAS 21448:2019.

In our case study, the code as well as the adopted heterogeneous multi-sensor architecture are only introduced as an instance of the methodology and cannot be, in any case, used in an industrial application without validation. Nevertheless, the specification of safety and security requirements and their analysis facilitates this critical step, detecting errors as early as possible and therefore reducing the overall cost in the design and developments phases

IV. FEASIBILITY STUDY

In this section, we illustrate the steps of our methodology on a fully fledged case study, involving autonomous cars.

Natural Language Use Case Specification: The case study is based on the concept of *platoon* of vehicles [43], [44]. A system of systems is called a “platoon” if the systems are able to operate closely together and, moreover, one system is able to send control signals to the others. The concept of vehicle platooning aims to increase road capacity, traffic fluidity and could be useful in emergency scenarios.

We consider a platooning system composed of k connected vehicles $V_1 ... V_k$, as shown in Figure 5 for $k = 3$. In our use case, Vehicle V_1 is the leader.

Natural Language Requirements: In the rest of this paper, we address the following two properties. First, the V2V communication ought to be *safe*: if vehicle V_1 detects an obstacle on its route, then V_1 alerts V_2 through a *brake* signal. This V2V communication shall be guaranteed within a given elapsed time. Second, the V2V communication ought to be *secure*. This involves that the authorization and authentication phases should be guaranteed.

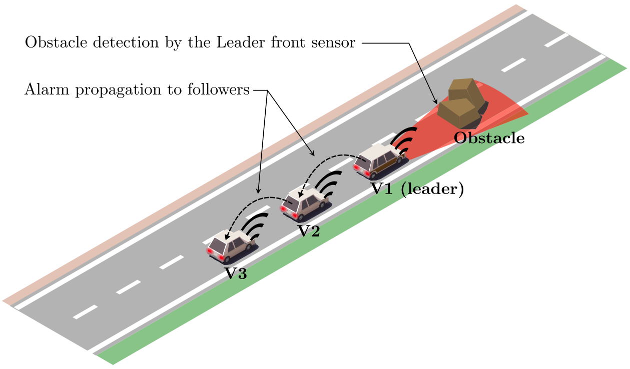

Obstacle detection by the Leader front sensor
Alarm propagation to followers
Obstacle
V1 (leader)
V2
V3

FIGURE 5. A platoon of connected vehicles.

TABLE 1. System Requirements, formalized as guarantees and assumptions.

Id	Assertion	Component
r_1	**Everytime** [OBSTACLE] **then** [BRAKE CMD] **within** [Y ms]	$V_i \mid \forall i = 1..k$
r_2	**Everytime** [OBSTACLE] **then** [SEND AlarmToV_{i+1}] **within** [X ms]	$V_i \mid \forall i = 1..k-1$
r_3	[WiFi connection] **always**	Infrastructure
r_4	**Everytime** [RECV Alarm] **then** [BRAKE CMD] **within** [Y ms]	$V_i \mid \forall i = 2..k$
r_5	**Everytime** [RECV Alarm] **then** [SEND AlarmToV_{i+1}] **within** [X ms]	$V_i \mid \forall i = 2..k-1$
r_6	**Everytime** [Brake CMD] **then** [STOP] **within** [W ms]	$V_i \mid \forall i = 1..k$
r_7	**Everytime** [OBSTACLE] **then** [OBSTACLE AVOIDED] **within** $[X * (\lvert \text{List}_V \rvert - 1) + Y + W \quad]$	The Platoon System
r_8	[Heterogeneous multi-sensors architecture able to measure the distance between two vehicles] **always**	$V_i \mid \forall i = 1..k$
r_9	**Everytime** $[d(\lvert V_{i-1} \rvert, \lvert V_i \rvert) < 25m]$ **then** [BRAKE CMD (V_i)] **within** [Y ms] **AND** [Transfer BrakeAlarmToV_{i+1}] **within** [X ms] **AND** [SEND NotifyToV_{i-1}] **within** [X ms]	$V_i \mid \forall i = 1..k-1$
r_{10}	**Everytime** $[d(\lvert V_{i-1} \rvert, \lvert V_i \rvert) > 50m]$ **then** [SEND BrakeNotifyToV_{i-1}] **within** [X ms]	$V_i \mid \forall i = 2..k$
r_{11}	[Arrowhead support] **always**	Infrastructure
r_{12}	**Everytime** [RequiredAuthentification $V_i - V_{i-1}$] **then** [Authentication Arrowhead.Established] **within** [Z ms]	$V_i \mid \forall i = 2..k$

A. CONTRACTS SPECIFICATION

We formalize safety properties using contracts. Broadly speaking, a contract is a pair (assumptions, guarantee) such that the guarantee is a service provided by the component and the assumptions are the requirements needed to accomplish the guarantee. Table 1 introduces the subset of assertions that we use to build contracts that deal with the brake signal control command. In BCL, contracts are expressed using patterns and expressions over the system variables. Patterns are built in layers to express invariants in terms of events and their time and logical relations. In particular, the keywords **Everytime** [E] **then** [C] assert that whenever event **E** occurs, the event **C** also occurs, while the keyword **within** delimits the constraint in the time domain. The keyword **always** specifies that a timing constraint must be satisfied at every instant. Other assertions, which we do not show for brevity, can be used to model other aspects of the system.

We denote by List_V the (finite) list of vehicles in the platooning system, where V_1 and V_k are the first and the last vehicle, respectively. Each vehicle $V_i \in \text{List}_V$ communicates with V_{i-1} and V_{i+1}. Each assertion is related to a component, as either an assumption or a guarantee. For instance, assertion $r5$ asserts that if vehicle V_i receives an alarm signal and V_i is not in the last position of the list (V_k), then it propagates the signal alarm further.

We build contracts by combining a set of assumptions with a guarantee. For instance,

Contract $\mathbf{V_i2V_{i+1}Latency}$ $= (r3, r5)$.

states that the maximum time of a V2V alarm message transmission (X ms in Guarantee $r5$) is guaranteed if the WiFi connection is always available (Assumption $r3$). This implies that engineers have to pay attention to the WiFi infrastructure, which should be resilient even in the cases of emergency and catastrophic scenarios.

Contract Stop $= ((r1, r2, r3, r4, r5, r6, r11), r7)$

asserts the maximum latency to stop all vehicles in the platooning system. Guarantee $r7$ represents the worst case execution time (WCET) for the whole chain of control. Parameter W depends on a set of variables, such as velocity and weight. The contract is based on the messaging latency between two vehicles (X), the time to send the Brake command internally (Y) and the time to fully stop the vehicle (W).

The two contracts

Contract Detection Min. Distance $= ((r1, r2, r3, r8), r9)$

Contract Detection Max. Distance $= ((r1, r2, r3, r8), r10)$

define the minimum and the maximum acceptable distance between vehicles, respectively. The contracts are primarily used to keep a safe distance between each pair of consecutive vehicles. For instance, when an animal crosses the road between the first and the second vehicle, the obstacle is detected by the second vehicle, but not by the first which cannot send the alarm brake command. To avoid an accident, the second vehicle becomes the leader of the platooning system: it brakes, sends the brake alarm to the next vehicle[1] and notifies the previous of the current operation (**Contract Detection Max. Distance**). The contract assumes $r8$, i.e., a multi-sensors heterogeneous architecture (a minimum of position, velocity and data from an ultrasonic sensor is demanded) able to detect obstacles. Naturally, **Contract Detection Min. Distance** is also in place between the second and the third vehicle.

Contract Detection Min. Distance is a means to reduce the threshold of severity that could occur in an accident [13], [45]. Indeed, in the case of a failure of the communication system, the embedded system is able to automatically brake, thanks to the exploitation of the multi-sensors heterogeneous architecture and the related embedded functionality. For the sake of completeness, our prototype realizes also the automatic reduction of velocity of the system [17].

Contract Secure Authentication $= (r11, r12)$

asserts the maximum latency needed to have authentication and authorization of a service. Finally,

Contract Safe&Secure Brake Command $=$ $((r3, r8, r11, r12), r4)$

guarantees that a vehicle executes the alarm brake command, which has been received from an authenticated vehicle. The assumptions are a WiFi infrastructure, resilient to catastrophic scenarios, an embedded multi-sensors heterogeneous architecture able to detect the distance between vehicles in real time and the Arrowhead support infrastructure.

During the design and development phases, engineers specify the variables with the expected values and perform some (static) analysis to verify that the system meets all deadlines. This analysis is currently adopted in the industry and aims to reduce, or avoid, the possibility to find an error late in the development phase [46]. In our use case, we let X and W be 100 ms and 120 ms, respectively.

B. 2D MODELING AND CAT

Figure 6 represents part of the SysML model of the scenario (ref. Figure 5), including **Contract $\mathbf{V_i2V_{i+1}Latency}$**.

In the figure, the three blocks represent the leader vehicle, a vehicle in the platoon system and the WiFi infrastructure. Both vehicles use the WiFi, represented via a vertical connector between the corresponding ports. The leader vehicle is able to send the alarm to the second vehicle, represented by two ports and the horizontal connector. Assumption $r3$ is allocated to the connector between WiFi and a vehicle, and by extension to its connected ports. Guarantee $r2$ is allocated to the connector between the two vehicles. **Contract $\mathbf{V_i2V_{i+1}Latency}$** is allocated to the vehicle, which receives the Brake command. CAT gives designers the freedom of selecting the best option to allocate assumptions, guarantees and contracts. For instance, in **Contract $\mathbf{V_i2V_{i+1}Latency}$**, we use connectors as target model elements to specify the assumption and guarantee, increasing readability of the model.

C. 3D MODELING AND CAT

In order to have a better view of the unfolding of the system, we construct a 3D model of our scenario in Blender (ref Figure 7). More specifically, we construct a 3D model of a platoon containing $k = 3$ vehicles on a highway. We then define an animated scenario where the platoon is going at a regular speed on the highway, and the leader of the platoon detects a stopped truck on the road. The first vehicle starts braking and sends an alert signal to the following vehicles that start braking as well, until the whole platoon is fully stopped. This scenario, while simple to describe, involves many safety

[1] Assertions $r1$ and $r2$. The related contracts are omitted

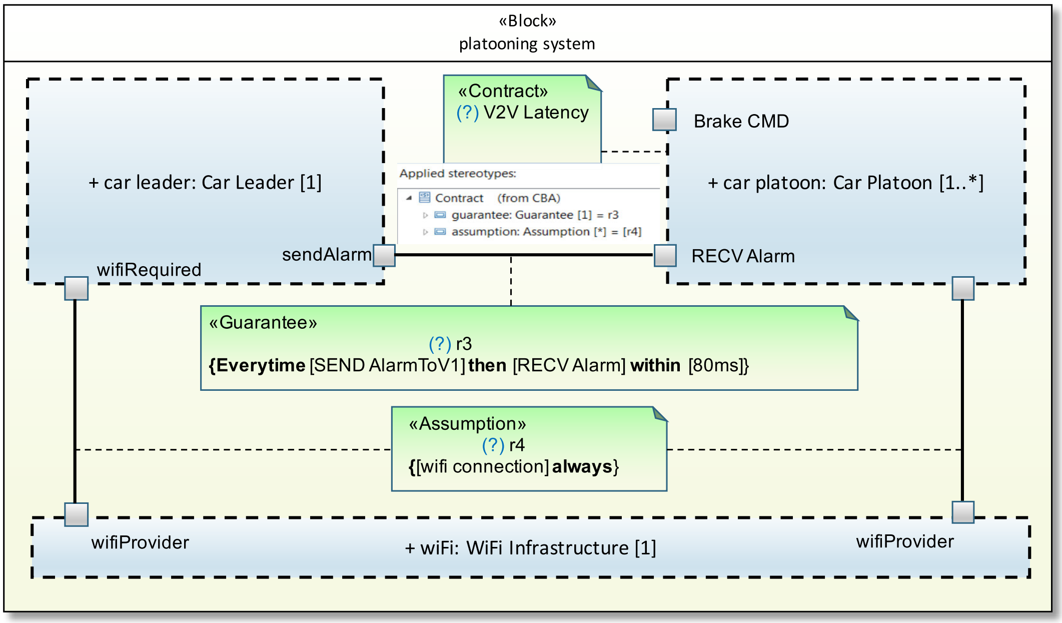

«Block»
platooning system
+ car leader: Car Leader [1]
«Contract»
(?) V2V Latency
Applied stereotypes:
Contract (from CBA)
guarantee: Guarantee [1] = r3
assumption: Assumption [*] = [r4]
Brake CMD
+ car platoon: Car Platoon [1..*]
sendAlarm
wifiRequired
RECV Alarm
«Guarantee»
(?) r3
{Everytime [SEND AlarmToV1] then [RECV Alarm] within [80ms]}
«Assumption»
(?) r4
{[wifi connection] always}
wifiProvider
+ wiFi: WiFi Infrastructure [1]
wifiProvider

FIGURE 6. Platooning System 2D Model with contracts.

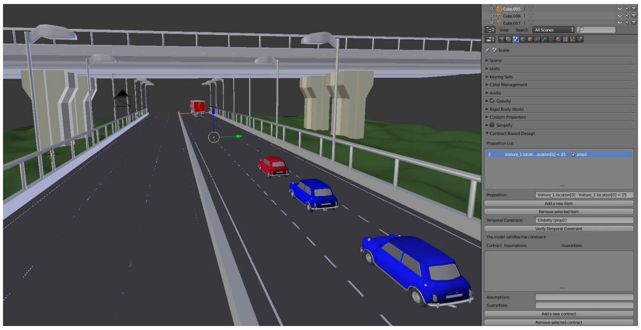


FIGURE 7. Platooning System 3D Model with contracts. Once contracts were validated, we generated the video, available on `https://youtu.be/jSvLUz4hURM`

issues that are not visible at first sight and that need to be investigated.

After constructing the 3D model and defining animated scenarios, we use CAT to verify compliance with the contracts. As discussed, the current implementation of CAT exploits Linear Temporal Logic (LTL) as a formal underlying language. In the literature, a first LTL tool, integrated with Blender and developed in Python 2, was proposed in [47]. CAT (based on Python 3) capitalizes on that result.

For an example, **Contract Detection Min. Distance** becomes the following set of LTL propositions:

Id	LTL Proposition
prop1	Distance(Vehicle1,Vehicle2) < 25
prop2	Vehicle2.Braking()
prop3	Vehicle2.SendBrakeToVehicle3()
prop4	Vehicle2.NotifyVehicle1()

Similarly, we specify Propositions 5 to 8 for each other vehicle in the platooning system (please note that for the last vehicle V_k we specify Propositions 5 to 7 only, since V_k does not transfer the brake command to other vehicles). We use CAT to introduce the above LTL propositions. Then, we specify the LTL formula

$$\text{Globally}[\text{prop1} \rightarrow [\text{prop2} \wedge \text{prop3} \wedge \text{prop4}]]$$

which stands for *At all states, the distance between two given vehicles of the platoon should not be less than 25 meters. If it is less, then the subsequent vehicle simultaneously sends three commands: braking, sending the brake command alarm to the next vehicle (if it exists) and notifying the previous one.* At the implementation level, their execution is delayed due to hardware constraints, multi-sensors architecture, and software implementation.

To capture the overall semantics of **Contract Detection Min. Distance**, we need a formula for each pair of vehicles (at the implementation level, we optimize the formulas being tested). Finally, we specify the propositions and the formula in the animated scenarios. CAT is able to automatically verify the formula, thanks to Python scripting.

Contract Secure Authentication becomes the following set of LTL propositions:

Id	LTL Proposition
prop1	RequireAuthentication(Vehicle2, Vehicle1)
prop2	ReponseAuthentication(Vehicle1, Vehicle2)
prop3	AuthenticationSuccessful(Vehicle2, Vehicle1) $\leq$ 120 ms

We then specify the following LTL proposition:

$$\text{Globally}[\text{prop1} \rightarrow [\text{prop2} \wedge \text{prop3}]]$$

which stands for *At all states, a secure authentication is established in less than 120 ms*. Technically, a 3D scenario is a set of *frames* combined together with *keys* representing the set of values of all parameters involved within the model at a specific frame: object locations, physical constraints, etc. The verification process of the temporal constraints is based on assigning states to *frames*. As a result, the notion of time is summed up into frames: we can choose which time unit to assign for each frame by manipulating the FPS (Frame Per Second) [48]. Although we can specify and verify timing constraints in CAT, we need specific tools to accomplish exact measures, such as the Arrowhead Framework support.

D. ARROWHEAD FRAMEWORK

In this work, we use only the mandatory core services of the Arrowhead Framework, to achieve trust between the members of the platoon. Automotive units are connected together via a multi-hop wireless protocol, for example ad-hoc IEEE 802.11g or IEEE 802.11p. One automotive unit (the leader) hosts the mandatory Arrowhead core services.

Communication is service-oriented, so, for instance, each automotive unit that offers an application service `BrakeSignal` registers it in the Service Registry system hosted by the leader. Later on, each automotive unit completes its setup by looking for the `BrakeSignal` service through the Orchestration service, and by authenticating against them. After that, the automotive units will start driving in a line, and each unit can exchange messages with the following and preceding unit. Refer to Figure 1 in Section II-B for a representation of the interactions between the involved systems, considering that the *application service* mediating the interaction is the `BrakeSignal` service.

We consider that the messages are sent by the preceding unit to the following one. The initiator of a brake signal chain of message exchanges is thus the leader.

To test our approach, two implementations were taken into consideration:

- **Simple Web Service (SWS)**, which considers that the unit receiving the message publishes a service, which is then contacted by the unit that intends to send a message;
- **Web Service with persistent HTTPS implementation (Server-Sent Events, or SSE)**, which considers that the unit available to send a message publishes a service, the unit that wants to receive it connects to the service, and the service is kept on a keepalive status. The service publisher will then send the message to the service consumer when time due.

Both protocols have been tested in a cleartext (HTTP) and in a secured (HTTPS with Arrowhead tokens) manner as well. Authentication for the latter cases are provided by the X.509 certificate hierarchy of Arrowhead [28].

E. PROTOTYPING AND IMPLEMENTATION

Video animations of the braking alarm propagation scenario of Figure 5 using platoons of two and three wheeled robots are available on YouTube under the links `http://y2u.be/2WHyy5Z7nv4` and `http://y2u.be/Cl-vGISxBe4`.

The communication architecture of the distributed application comprising our wheeled robots is given in Figure 8. The two robots are represented in the Arrowhead Framework by one Arrowhead system each, and can communicate in a secure manner through either a Simple Web Service published into the Arrowhead Framework, or using Service-Sent Events (see Section IV-D). The two systems can also communicate with the core systems of the Arrowhead Framework (see Section II-B), which are deployed on the first robot, to discover their communication recipients, for authentication, and for key management.

The hardware of the robot, whose picture is reported in Figure 9, is composed of: 1) four micro DC (Direct Current) geared motors used to rotate the wheels with a power supply of 7.5V, 2) two motor encoders with a resolution of 20 PPR (Pulses Per motor Revolution) which can be fixed on the front or rear motors, 3) an ultrasonic sensor (HC-SR04) positioned at the front of the robot, 4) a Romeo (DFRobot product) low-level slave robot controller used to efficiently interface (using Arduino functions) with the three first hardware components, and 5) a Raspberry Pi (RPi) master high-level control card on which the control software is deployed to command the Romeo board. Data exchange is wired between the two boards using the I^{2}C bus.

The control software is distributed and real-time and based on object-oriented component-based method of design. It is implemented using the annexes D and E of the Ada Reference Manual resp. of real-time and distributed systems. Annex E (commonly abbreviated DSA) makes the middleware layer completely transparent and the development much easier. The distribution in our software is managed by the middleware PolyORB [49] maintained by the company AdaCore. It supports different distribution models including CORBA and DSA but also the Ravenscar profile (a restricted tasking Ada subset used for hard real-time). Interoperability and service invocations between vehicles is simply implemented as method calls on references to remote objects [17].

The application is executed under fully preemptive versions of Linux kernel 4.4.21 (patch Preempt_RT). Prior tests were made to evaluate low latency, preemption and deadline respectfulness of the scheduler `SCHED_FIFO` under extremely stressful processing conditions using the tools `cyclictest` and `hackbench` [50], but also Ada concurrent programs. Results were positive arguing a high level of real-time determinism and low run-time overhead. Control run-time jobs of each robot are executed concurrently in deadline-sensitive periodic tasks. They have priority over system calls and scheduled using `SCHED_FIFO` enabled by the Ada dispatching policy `FIFO_Within_Priorities` [17].

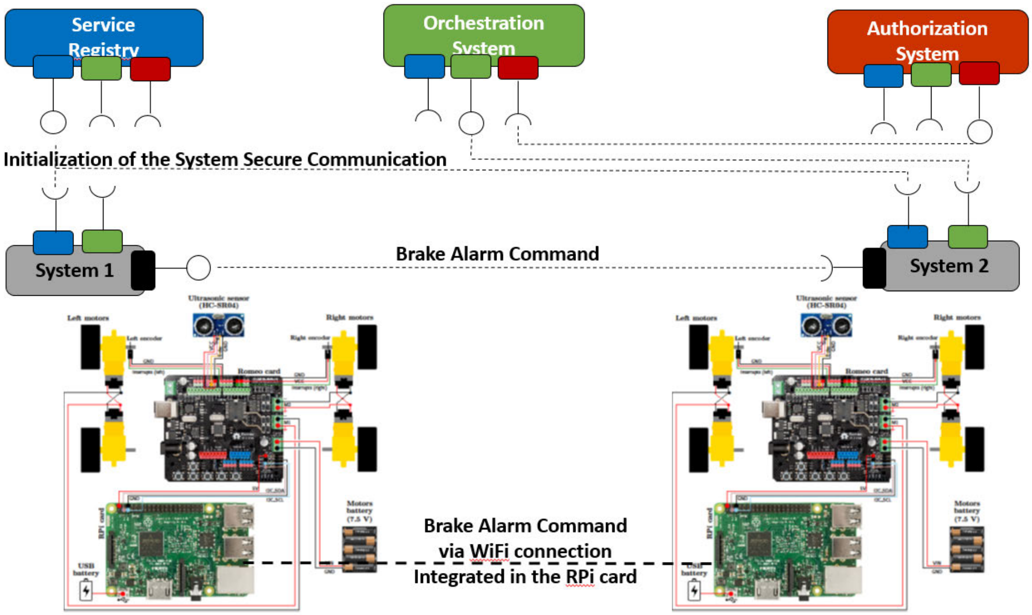


FIGURE 8. Wheeled robots communication architecture.

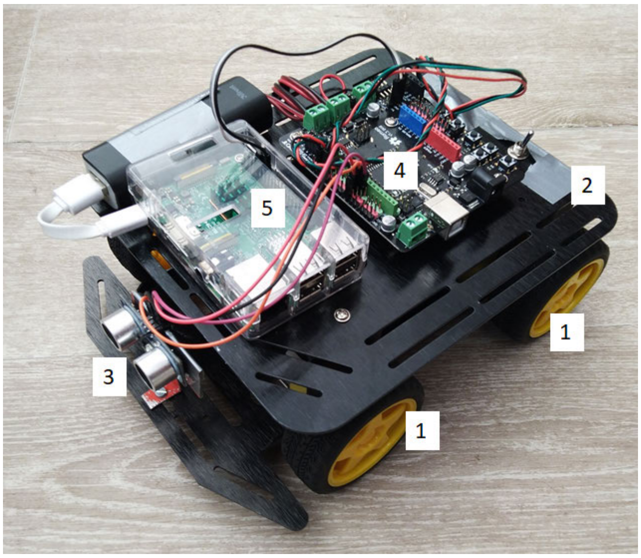


FIGURE 9. Hardware architecture of the wheeled robot.

The **Contract $\mathbf{V_i 2 V_{i+1}}$Latency** on networking latency between two robots in the platoon is natively supported by the code [2] (see [17] for details). The parameter X was defined to be the deadline of the driving job of robots. In order to ensure the contract, we simply check the deadline respectfulness when a braking alarm is propagated to the follower.

Vehicular networking technologies and the study of their underlying latency impact and security issues are mandatory to perform tests under real circumstances of car circulation. The main standards under consideration by the industry are the IEEE 802.11p for Wireless Access in Vehicular Environments (WAVE), based on Vehicular Ad-Hoc Network, and the promised Long Term Evolution (LTE)-based V2X conducted by the Third Generation Partnership Project (3GPP), based on cellular networks. The ITS-G5 European norm and the Dedicated Short-Range Communication (DSRC) equivalent standardization in USA and Japan are pushing forward IEEE 802.11p. They cover the specifications of the two lowest (physical and data link) protocol stack layers of vehicular networking systems at a frequency band ranging in [5.855, 5.925] GHz with V2V latency ranging between 10 ms for fully autonomous vehicles and 100 ms for semi-autonomous connected vehicles. Currently, these standards are mature, stable and implemented by industries on thousands of cars. LTE-Vehicle (LTE-V) has been deployed since a few years. This picture got more complicated with the arrival of 5G (fifth generation cellular network) technology shortly after the launch of the LTE-V standardization process. 5G devices embed a multitude of millimeter wave antennas allowing very low latency (up to 1 ms) while increasing throughput. Meanwhile, possible 5G solutions and the enhanced V2X (eV2X) services are being analyzed and considered by 3GPP in order to be up to date. A deep study of these issues at the functional, design, and implementation levels using the adapted hardware and software support would be a natural extension of this work.

V. EXPERIMENTAL ANALYSIS OF THE INTEGRATION

In this section, we analyze both the performance and the properties of the platooning system. As briefly discussed in Section IV, the sensors embedded in the leader of the platoon always check whether there are obstacles in front of

[2] https://github.com/mouelhis/adawrplatoon

the vehicle. If any obstacle is detected, the alarm is immediately activated. Once the alarm is activated, the leader of the platoon starts braking and the authentication process with the second vehicle of the platoon is triggered. This process ensures the mutual authentication of the communicating parties, and the certificates allow for strong security to protect the transmission of the alarm signal sent by the platoon over the V2V communication. Therefore, no external actor can interfere with the vehicular network, for instance to make the cars brake without the presence of an obstacle. If the authentication process succeeds, the alarm is sent to the second vehicle of the platoon. Once the second vehicle receives the alarm signal, it immediately starts braking and launches an authentication process with the third vehicle of the platoon. If this process is successful, the alarm signal is transferred to the third vehicle that forthwith starts braking until the whole platoon is stopped.

The example given above presents the theoretically required control flow, from detecting the obstacle until the car platoon is fully stopped. Trust concerns are kept out of the way by strong token-based security, but in practice things can get more complicated and a closer look reveals several potential problems.

The first problem is the potential successive failures of the authentication process between two vehicles. In this case, the alarm signal is not transmitted (or transmitted with a delay). Subsequently, the succeeding vehicle will not brake in time and will crash into the vehicle in front of it. In order to address this issue, we combine safety and security techniques, methods, and tools. The Arrowhead Framework secures the V2V channel *in advance* by providing security *tokens* to each vehicle. The impact of communication problems is therefore minimized by reducing the number of messages to be exchanged to support urgent events. As a result, this technique also reduces the probability of message retransmissions, which are the root causes of much longer communication delays. Moreover, to enhance safety, we equip each vehicle of the platoon, and not only the leader, with its own sensors. If the distance between the two vehicles decreases, the vehicle brakes, independently of the communication activities.

The second problem is related to timing constraints under good communication conditions (no message retransmissions on the underlying protocols). The experimental tests analyzed three different implementation for the communication primitives: unsecure webservice, token-based webservice, and SSL-based SSE. Figure 10 compares the communication latency of the three approaches. The approach based on a Web Service with persistent HTTPS (SSE) is much faster, and it still makes use of the strong authentication/authorization capabilities of the Arrowhead Framework to manage the security of the SSL channel. A summary of the results is reported in Table 2, which highlights that the (non-optimized) webservice implementations do not satisfy the contracts, while the SSE implementation is a viable method to establish trust in a car fleet.

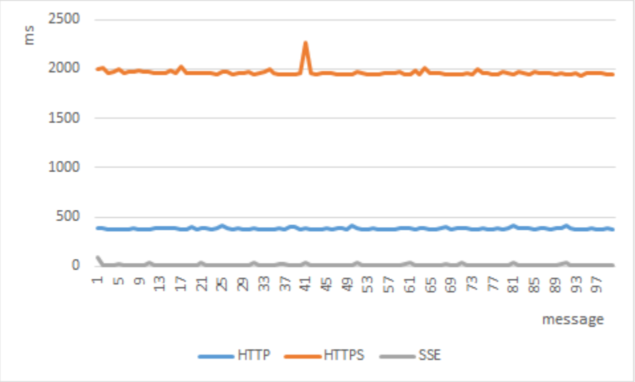


FIGURE 10. Comparison of implementations of primitives.

TABLE 2. Measures for a V2V secure communication.

Approach	mean delay	maximum
Unsecure HTTP	382.2	420.0
HTTPS	1963.1	2268.0
Vanilla SSE	11.9	88.0
SSE with keepalive	13.2	37.0

Higher performance can be granted to the SSE technique by exchanging periodic keepalive messages through the channel. Figure 11 analyzes a run with 10 messages sent through the SSE approach, and shows that the first message takes much longer than the following ones, since it has to wake up a dormant connection. Thus we propose to exchange periodic keepalive messages through the persistent Web Service, to speed up the communication of urgent alarms such as in the case at hand. Figure 12 presents the Cumulative Distribution Function for the message delays in the case of the SSE approach with keepalive messages, over a total of 100 messages.

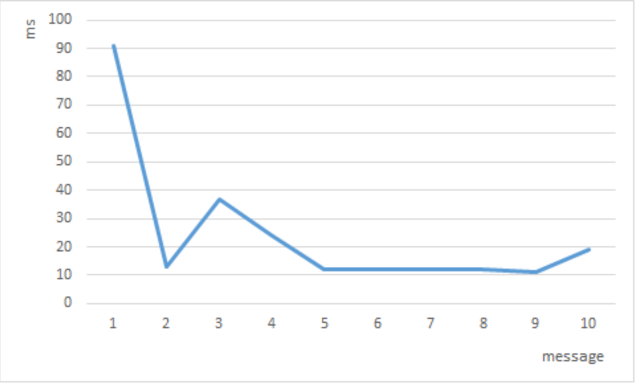


FIGURE 11. Delays for 10 messages.

The results show that the time needed for the signal to be transmitted from one vehicle to another is lower than 40 ms, if the best approach is used. This value represents the measure of Z in assertion $r12$. With this, we can validate our contracts. In particular, the correctness of **Contract Secure Authentication** is validated with respect to the requirement

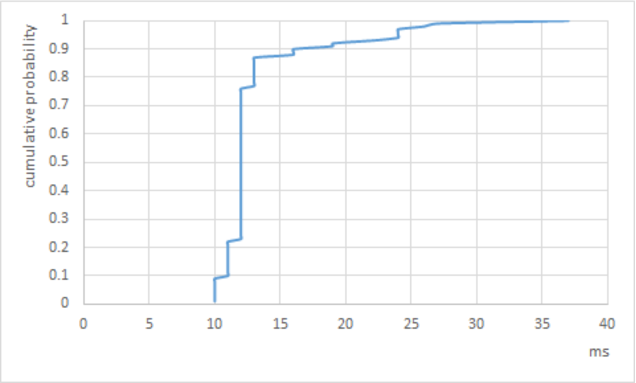


FIGURE 12. Cumulative Distribution Function for the SSE approach with keepalive.

of 120 ms latency. Clearly, if we select another approach, for example Unsecured HTTP, the contract is not satisfied, and correctness cannot be guaranteed according to our CAT analysis. Therefore, either designers change the specification on the model or another approach ought to be considered. Indeed, this latency value constitutes a constraint as the vehicles can be going at high speed (in a real scenario) and the timing needs to be optimized in order to ensure that the succeeding vehicle starts braking as soon as possible. To solve this issue and to guarantee that the succeeding vehicle starts braking before it is too late, we propose to not wait until the authentication process is successful to transfer the alarm signal. Therefore, once the alarm signal is received, the vehicle instantly starts braking, sends the signal to the succeeding vehicle, starts the authentication process with both vehicles and checks its own sensors. Consequently, the timing between the transmission of the alarm signal and the effective braking of the successive vehicle is optimized. This scenario is specified by **Contract Safe & Secure Brake Command**. Following our tests, the timing needed for a safe alarm command transfer is 80 ms (see Section IV-E), while the time needed for a secure communication is lower than 40 ms if we adopt SSE with keepalive. The contract is therefore verified: the implementation is consistent with respect to the initial specification. Similarly to **Contract Secure Authentication**, also in this case the choice of the approach plays a decisive role in the validation of the contract.

VI. OUTLOOK AND CONCLUDING REMARKS

There are multiple benefits from automated driving technology. Important examples of such advantages include optimized power consumption, reduced emissions, improved safety by decreased number of accidents caused by human errors, increased passenger comfort by allowing for other activities while driving, support of elderly or impaired users. Furthermore, a lively competition on the international level indicates that autonomous vehicles are widely accepted as means to increase the economic performance of involved countries. According to the Automotive World website, 75% of cars on the road will be autonomous by 2035.[3] For instance, the automotive industry is the largest private investor of R&D in Europe and by 2030 the expected economic impact of automated driving will be up to €71bn and the global market for automated vehicles will be 44 million vehicles respectively [51].

To realize these goals, international cooperation in R&D is essential and can be strengthened by means of an agreed approach and projects of common interest. Worldwide, various initiatives have been started to support competitiveness and growth in the automotive sector. For example, trilateral EU-US-Japan agreements simplify collaboration and sharing of information on vehicle and road automation.[4] National programs, such as the USA's "Intelligent Transportation Systems Strategic Plan", Japan's "Automated Driving for Universal Services", China's "Strategic Alliance for Intelligent and Connected Vehicles" or "Smart Car Council" of South Korea, invest significant funds into research, development, standardization, and promotion of autonomous driving technologies. In Europe, competitiveness and growth in the automotive sector is supported by various initiatives such as the European Road Transport Research Advisory Council (ERTRAC)[5] and GEAR 2030,[6] which put together automotive stakeholders and policymakers to ensure a coordinated approach in Europe. Moreover, the European guidelines such as ERTRAC Automated Driving Roadmap [51] are used by industry and academia in order to create national roadmaps reflecting local objectives and regulations, and generally to facilitate public acceptance within a country.

In this paper we pursue these objectives by addressing the potential conflicts between safety and security properties. We considered safety and security properties during the design phases of a platooning autonomous system: from textual specification, modeling, 3D simulation, embedded software to early prototyping. At the methodological level, we adopted the Arrowhead Framework and a contract-based approach, supported by CAT for simulation-based contract verification.

Although the involved supporting tools are not yet fully optimized, our study clearly shows that the issue can effectively be specified and analyzed at the design level. However, at the prototyping level, we are faced with the problem of ensuring safety with low latency. This technical constraint could be mitigated with the use of very high-performance processors and 5G. Nonetheless, promoting these issues at the design level allows engineers to find alternative solutions to guarantee the required safety level.

ACKNOWLEDGMENT

This work is supported by the *Arrowhead Makes You Productive* song, originally composed by Erkki Jantunen for this

[3] https://www.automotiveworld.com/megatrends-articles/ethernet-fast-track-connected-car/

[4] http://vra-net.eu

[5] http://www.ertrac.org/

[6] http://ec.europa.eu/growth/sectors/automotive/policy-strategy/

work. https://productive40.eu/2018/01/18/arrowhead-makes-you-productive.

The research has received funding from the EU ECSEL JU under the H2020 Framework Programme, JU grant nr. 737459 (Productive4.0 project, www.productive40.eu). This Joint Undertaking receives support from the European Union's Horizon 2020 research and the participating national funding organizations, including the Finnish Funding Agency for Innovation Tekes.

Part of this work was supported by the Higher Education Excellence Program of the Ministry of Human Capacities, Hungary, in the frame of Artificial Intelligence research area of Budapest University of Technology and Economics (BME FIKP-MI/SC).

REFERENCES

[1] A. Talebpour and H. S. Mahmassani, "Influence of connected and autonomous vehicles on traffic flow stability and throughput," *Transp. Res. C, Emerg. Technol.*, vol. 71, pp. 143–163, Oct. 2016.

[2] K. Ma and H. Wang, "Influence of exclusive lanes for connected and autonomous vehicles on freeway traffic flow," *IEEE Access*, vol. 7, pp. 50168–50178, 2019.

[3] K. Lee and D. Kum, "Collision avoidance/mitigation system: Motion planning of autonomous vehicle via predictive occupancy map," *IEEE Access*, vol. 7, pp. 52846–52857, 2019.

[4] G. Bensinger and J. Nicas, "Uber to put 100 autonomous volvo SUVs on road in Pittsburgh," *Wall Street J.*, Aug. 2016.

[5] C. Bounoux. *La Voiture Autonome Valeo Sur Le Peripherique Parisien*. Accessed: Sep. 4, 2019. [Online]. Available: https://www.youtube.com/watch?v=dd6pqIuhaHkr, TL television.

[6] J. Nie, J. Zhang, W. Ding, X. Wan, X. Chen, and B. Ran, "Decentralized cooperative lane-changing decision-making for connected autonomous vehicles," *IEEE Access*, vol. 4, pp. 9413–9420, 2016.

[7] C. Zhai, F. Luo, and Y. Liu, "Cooperative look-ahead control of vehicle platoon for maximizing fuel efficiency under system constraints," *IEEE Access*, vol. 6, pp. 37700–37714, 2018.

[8] S. Biswas, R. Tatchikou, and F. Dion, "Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety," *IEEE Commun. Mag.*, vol. 44, no. 1, pp. 74–82, Jan. 2006.

[9] E. Talavera, A. D. Álvarez, and J. E. Naranjo, "A review of security aspects in vehicular ad-hoc networks," *IEEE Access*, vol. 7, pp. 41981–41988, 2019.

[10] D. Cancila, R. Passerone, T. Vardanega, and M. Panunzio, "Toward correctness in the specification and handling of non-functional attributes of high-integrity real-time embedded systems," *IEEE Trans. Ind. Informat.*, vol. 6, no. 2, pp. 181–194, May 2010.

[11] S. Plósz, C. Schmittner, and P. Varga, "Combining safety and security analysis for industrial collaborative automation systems," in *Proc. Int. Conf. Comput. Saf., Rel., Secur.* Cham, Switzerland: Springer, 2017.

[12] S. Plósz and P. Varga, "Security and safety risk analysis of vision guided autonomous vehicles," in *Proc. IEEE Ind. Cyber-Phys. Syst. (ICPS)*. Cham, Switzerland: Springer, May 2018, pp. 193–198.

[13] *Road Vehicles—Functional Safety*, Standard ISO-26262-1, International Organization for Standardization, Geneva, Switzerland, 2018.

[14] J. Delsing, "The arrowhead framework architecture," in *IoT Automation: Arrowhead Framework*, J. Delsing, Ed. Boca Raton, FL, USA: CRC Press, 2017, ch. 3.

[15] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. Henzinger, and K. G. Larsen, *Contracts for System Design* (Foundations and Trends in Electronic Design Automation), vol. 12. Boston, MA, USA: NOW, 2018.

[16] T. Erl, *SOA Principles of Service Design*. Upper Saddle River, NJ, USA: Prentice-Hall, 2007.

[17] S. Mouelhi, D. Cancila, and A. Ramdane-Cherif, "Distributed object-oriented design of autonomous control systems for connected vehicle platoons," in *Proc. 22nd Int. Conf. Eng. Complex Comput. Syst. (ICECCS)*, Nov. 2017, pp. 40–49.

[18] T. L. Willke, P. Tientrakool, and N. F. Maxemchuk, "A survey of inter-vehicle communication protocols and their applications," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 2, pp. 3–20, 2nd Quart., 2009.

[19] F. Dressler, H. Hartenstein, O. Altintas, and O. Tonguz, "Inter-vehicle communication: Quo vadis," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 170–177, Jun. 2014.

[20] B. Lonc and P. Cincilla, "Cooperative ITS security framework: Standards and implementations progress in Europe," in *Proc. IEEE 17th Int. Symp. A, World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2016, pp. 1–6.

[21] M. Torrent-Moreno, J. Mittag, P. Santi, and H. Hartenstein, "Vehicle-to-vehicle communication: Fair transmit power control for safety-critical information," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3684–3703, Sep. 2009.

[22] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: Design and architecture," *IEEE Commun. Mag.*, vol. 46, no. 11, pp. 100–109, Nov. 2008.

[23] S. Lefevre, J. Petit, R. Bajcsy, C. Laugier, and F. Kargl, "Impact of V2X privacy strategies on intersection collision avoidance systems," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2013, pp. 71–78.

[24] PRESERVE FP7 Project165. *Preparing Secure Vehicle-to-X Communication Systems*. Accessed: Sep. 4, 2019. [Online]. Available: https://www.preserve-project.eu/

[25] S. Gisdakis, M. Laganá, T. Giannetsos, and P. Papadimitratos, "SEROSA: SERvice oriented security architecture for vehicular communications," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2013, pp. 111–118.

[26] J. Yoo and J. H. Yi, "Code-based authentication scheme for lightweight integrity checking of smart vehicles," *IEEE Access*, vol. 6, pp. 46731–46741, 2018.

[27] L. Wu, Q. Sun, X. Wang, J. Wang, S. Yu, Y. Zou, B. Liu, and Z. Zhu, "An efficient privacy-preserving mutual authentication scheme for secure V2V communication in vehicular ad hoc network," *IEEE Access*, vol. 7, pp. 55050–55063, 2019.

[28] S. Plósz, C. Hegedűs, and P. Varga, "Advanced security considerations in the arrowhead framework," in *Proc. Int. Conf. Comput. Saf., Rel., Secur.* Cham, Switzerland: Springer, 2016, pp. 234–245.

[29] *AUTOSAR Standard*. [Online]. Available: https://www.autosar.org

[30] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone, "Taming Dr. Frankenstein: Contract-based design for cyber-physical systems," *Eur. J. Control*, vol. 18, no. 3, pp. 217–238, 2012.

[31] A. Baumgart, P. Reinkemeier, A. Rettberg, I. Stierand, E. Thaden, and R. Weber, "A model–based design methodology with contracts to enhance the development process of safety–critical systems," in *Proc. IFIP Int. Workshop Softw. Technol. Embedded Ubiquitous Syst.*, 2010, pp. 59–70.

[32] M. Bozzano, A. Cimatti, C. Mattarei, and S. Tonetta, "Formal safety assessment via contract-based design," in *Proc. Int. Symp. Automated Technol. Verification Anal.*, Sydney, NSW, Australia, Nov. 2014, pp. 81–97.

[33] O. Ferrante, R. Passerone, A. Ferrari, L. Mangeruca, and C. Sofronis, "BCL: A compositional contract language for embedded systems," in *Proc. IEEE Emerg. Technol. Factory Automat. (ETFA)*, Barcelona, Spain, Sep. 2014, pp. 1–6.

[34] S. Quinton and S. Graf, "Contract-based verification of hierarchical systems of components," in *Proc. 6th IEEE Int. Conf. Softw. Eng. Formal Methods*, Nov. 2008, pp. 377–381.

[35] L. D. Lago, O. Ferrante, R. Passerone, and A. Ferrari, "Dependability assessment of SOA-based CPS with contracts and model-based fault injection," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 360–369, Jan. 2018.

[36] M. Albano, L. L. Ferreira, L. M. Pinho, and A. R. Alkhawaja, "Message-oriented middleware for smart grids," *Comput. Standards Interfaces*, vol. 38, pp. 133–143, Feb. 2015.

[37] K. Nagorny, R. Harrison, A. W. Colombo, and G. Kreutz, "A formal engineering approach for control and monitoring systems in a service-oriented environment," in *Proc. 11th IEEE Int. Conf. Ind. Inform. (INDIN)*, Jul. 2013, pp. 480–487.

[38] International Telecommunication Union. (2012). *X.509: Public-Key and Attribute Certificate Frameworks*. [Online]. Available: http://www.itu.int/rec/T-REC-X.509-201210-I/en

[39] D. Cancila, E. Soubiran, and R. Passerone, "Feasibility study in the use of contract-based approaches to deal with safety-related properties in CPS," *Ada User J.*, vol. 35, no. 4, pp. 272–277, Dec. 2014.

[40] *EN-50126: Application Ferroviaires—Spècification et Démonstration de Fiabilité, Disponibilité, Maintenabilité et Sécurité (FMDS)*, CENELEC, Brussels, Belgium, 1999.

[41] *EN-50128: Applications Ferroviaires—Système de Signalisation, de Télécommunication et de Traitement—Logiciels pour Systémes de Commande et de Protection Ferroviaire*, CENELEC, Brussels, Belgium, 2001.

[42] *EN-50129: Application Ferroviaires—Système de Signalisation, de Télécommunication et de Traitement—Systémes électroniques Relatifs à la Sécurité Pour la Signalisation*, CENELEC, Brussels, Belgium, 2001.

[43] S. Sheikholeslam and C. A. Desoer, "Longitudinal control of a platoon of vehicles," in *Proc. Amer. Control Conf.*, May 1990, pp. 291–296.

[44] SARTRE EU Project. *Safe Road Trains for the Environment*. Accessed: Sep. 4, 2019. [Online]. Available: https://www.roadtraffic-technology.com/projects/the-sartre-project/

[45] *Functional Safety of Electrical, Electronic and Programmable Electronic Systems*, Standard IEC 61508:1998, 2000.

[46] A. Hall, "Realising the benefits of formal methods," *J. Universal Comput. Sci.*, vol. 13, no. 5, pp. 669–678, 2007.

[47] H. Zaghaghi. *Animation Temporal Verification*. Accessed: Sep. 4, 2019. [Online]. Available: http://www.foro3d.com/f230/animation-temporal-verification-77370.html

[48] D. Cancila, H. Zaatiti, and R. Passerone, "Cyber-physical system and contract-based design—A three dimensional view," in *Proc. Workshop Embedded Cyber-Phys. Syst. Educ. (WESE)*, Amsterdam, The Netherlands, Oct. 2015, Art. no. 4.

[49] AdaCore. (2003). *PolyORB User Guide*. [Online]. Available: http://docs.adacore.com

[50] R. Russell *et al. Latency Testing Utilities RT-Tests*. [Online]. Available: https://www.kernel.org/pub/linux/utils/rt-tests/

[51] *"Connectivity and Automated Driving," "Automated Driving Roadmap," Version 5.0*, ERTRAC Working Group, Berkeley, CA, USA, 2015, pp. 1–48.

ROBERTO PASSERONE (S'96–M'05) received the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California at Berkeley, in 1997 and 2004, respectively. Before joining the University of Trento, he was a Research Scientist with Cadence Design Systems. He is currently an Associate Professor of electronics with the Department of Information Engineering and Computer Science, University of Trento, Italy. He has published numerous research articles on international conferences and journals in the area of design methods for systems and integrated circuits, formal models, and design methodologies for embedded systems, with particular attention to image processing, and wireless sensor networks. He has participated in several European projects on design methodologies, including SPEEDS, SPRINT, and DANSE, and he was the Local Coordinator for ArtistDesign, COMBEST, and CyPhERS. He has served as the Track Chair for the real-time and networked embedded systems with ETFA, from 2008 to 2010, and the General Chair and the Program Chair for various editions of SIES.

DANIELA CANCILA received the Laurea (M.S.) degree in philosophy from the University of Roma (La Sapienza), Rome, Italy, the Diplôme d'études Approfondies degree in discrete mathematics and theoretical computer science from the Institut de Mathématiques de Luminy, Marseille, France, and the Ph.D. degree in theoretical computer science from the University of Udine, Italy, in 2004. Since 2008, she has been a Research Engineer with the French Alternative Energies and Atomic Energy Commission, Gif-sur-Yvette, France. Since 2014, she teaches system and software dependability with the Master Nuclear Energy (Paris-Saclay University). She is currently a CEA Expert on safety for (autonomous) cyber-physical systems. Her research interests include model-based design and safety analysis of cyber-physical systems.

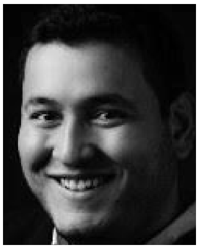
MICHELE ALBANO (M'11–SM'19) received the Ph.D. degree in computer science from the University of Pisa, Italy. He is currently a tenure-track Assistant Professor with the Department of Computer Science, Aalborg University, Denmark. His research interests include on distributed systems and embedded systems. He has been active in more than ten European research projects, and he acted as a technical manager for CELTIC project Green-T; a work package Leader for FP7 IP ROMEO, ITEA2 CarCoDe, and ECSEL MANTIS. He is currently an Editor of the Open Access book *"The MANTIS book: Cyber Physical System Based Proactive Maintenance"*, and he is also the Editor in Chief for the *Journal of Industrial Engineering and Management Science*, River Publishers.

SEBTI MOUELHI received the M.S. degree in computer science from the University of Lorraine, Nancy, France, in 2007, and the Ph.D. degree in computer science from the University of Franche-Comté, Besançon, France, in 2011. He was a Postdoctoral Researcher with INRIA, Grenoble, France, in 2011. He has been a Research and Development Engineer with SafeRiver, Montrouge, France, for about three years, since 2012. In 2015, he was an Engineer in safety assurance at ALSTOM Transport, Saint-Ouen, France. Since 2015, he has been an Associate Professor in embedded systems with the ECE Paris.Lyon, INSEEC U., Paris, France. His research interests include software engineering, formal methods, embedded systems, real time, and automatic control.

SANDOR PLOSZ is currently pursuing the Ph.D. degree with the Budapest University of Technology and Economics (BME). His research interests include safety and security management, network processes, network management, and measurement. He is also a member of the SmartCom Laboratory, BME. He has been involved in several industrial research and development projects in these topics.

ERKKI JANTUNEN was with the shipbuilding industry in structural, vibration, and hydrodynamic fields, from 1978 to 1990. Since 1990, he has been employed by VTT having various project responsibilities related to maintenance, condition monitoring, and diagnosis of rotating machinery. He is currently a Principal Scientist with the VTT Technical Research Center of Finland Ltd. He has been a Project Manager of many research projects. He is also the author and coauthor of several books and more than 150 research articles in the field of condition monitoring, diagnosis and prognosis, and e-maintenance. He has a position as a Visiting Professor at the University of Sunderland. He has been a member of the editorial board and acted as a Reviewer of a number of scientific journals.

ANNA RYABOKON received the doctorate degree in computer science from Alpen-Adria-Universität Klagenfurt, in 2015. Since, she has been joining TTTech Computertechnik AG, Austria, in 2017, she has been a Coordinator of Research and Development projects in the domain of safety-critical and autonomous systems. Her research activities mostly aimed at development and application of logic-based methods for artificial intelligence approaches to automated decision making.



EMINE LAAROUCHI received the engineering degree from the National Engineering School of Carthage, Tunisia, in 2015, and the M.S. degree in computer science (specialized in design of complex industrial systems) from the École Polytechnique, Palaiseau, France, in 2016. He is currently pursuing the Ph.D. degree on safety analysis of autonomous cyber-physical systems with the French Alternative Energies and Atomic Energy Commission, Gif-sur-Yvette, France, and Télécom Paris, Palaiseau, France. He is currently with Siemens Mobility, as a Research Engineer specialized in safety analysis of railway systems. His research interests include the safety analysis of autonomous cyber-physical systems, software engineering, and complex systems.

CSABA HEGEDŰS received the M.Sc. degree in electrical engineering from the Budapest University of Technology and Economics, BME, Hungary, specializing in telecommunications. He is currently a Research and a development Engineer with AITIA International Inc. At AITIA International Inc., he is also a main Contributor and a Developer on the ARTEMIS MANTIS, Arrowhead and Productive4.0 projects. He was the main contributor to the latest generation advances of the Arrowhead Framework. His research interests include the communication issues of cyber-physical systems and the Internet of Things architectures.

PAL VARGA (M'06) received the M.Sc. and Ph.D. degrees from the Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics (BME), Hungary, in 1997 and 2011, respectively, where he is currently an Associate Professor. Besides, he is also the Director with AITIA International Inc. Earlier, he was working for Ericsson Hungary and Tecnomen Ireland, as a Software Design Engineer and a System Architect, respectively. His main research interest include communication systems, network monitoring, network performance measurements, root cause analysis, fault localization, traffic classification, end to end QoS and SLA issues and hardware acceleration, and the Internet of Things. He has been involved in various industrial and European research and development projects in these topics.