25th ITS World Congress, Copenhagen, Denmark, 17-21 September 2018

Paper ID EU-TP1425

Modelling the C-ITS architectures: C-MobILE case study

Raul Ferrandez[*1], Yanja Dajsuren[2], Priyanka Karkhanis[2], Manuel Fünfrocken[3], Marcos Pillado[1]

1. Applus+ IDIADA, Spain
2. Eindhoven University of Technology, The Netherlands
3. htwsaar, Germany

*Email<raul.ferrandez@idiada.com>

Abstract

C-MobILE (Accelerating C-ITS Mobility Innovation and deployment in Europe) is a project co-founded by the European Commission that aims to stimulate large-scale C-ITS deployments interoperable across Europe. The main aim is to have well-defined operational procedures leading to decentralized and dynamic coupling of systems, services and stakeholders across national and organizational borders in an open, but secure C-ITS ecosystem, based on different access technologies across different transport modes, environments and countries. C-MobILE defines the C-ITS architecture framework conforming to the ISO/IEC/IEEE 42010 international standard and specifies stakeholders, concerns, model kinds and six core viewpoints: Context, Functional, Information, Implementation, Physical, and Communication. This paper describes the C-MobILE architectures conforming to the C-ITS architecture framework and modelling approach.

Keywords:

C-ITS, reference architecture, SysML, architecture modelling

Introduction

C-MobILE main objective is to define a Cooperative-ITS (C-ITS) reference architecture satisfying stakeholders from public and private parties. This means that the defined infrastructure must support various technologies, e.g., cellular and 802.11p, a wide range of users while maintaining sustainability and seamless operation within the deployment cities. Organisations can use the reference architecture to guide their internal development process as it reflects a common understanding of how the (future) ITS landscape will evolve. The C-MobILE reference architecture consolidates the CONVERGE, MOBiNET, and Dutch C-ITS Reference Architecture (DITCM) into one reference architecture and compares it with other architectures developed (e.g. C-The-Difference, SCOOP@F, InterCor, C-Roads, CITRUS, VRUITS, iKoPa, Talking Traffic, C-ITS Corridor, Eco-AT, CODECS, AUTOCITS, CITRUS, PROSPECT, XCYCLE, and SOLRED).

The C-MobILE reference architecture shall combine the essence of ITS/C-ITS architectures already created in previous projects. However, none of those architectures covers all; therefore, a combination of those architectures is necessary. The reference architecture is described according to C-ITS architecture framework conforming to the ISO/IEC/IEEE 42010 international standard for the systems and software engineering – architecture description [1].

The architecture process has been split into three parts: 1) the reference architecture for defining high-level architecture, 2) the concrete architecture for refining the high-level architecture into the medium-level architecture enhancing with functional interface descriptions, state diagrams and sequence diagrams to describe the interaction of high-level components, and 3) the implementation architecture for revising further the concrete architecture into a more detailed low-level architecture allowing cities to adapt their implementation to the C-MobILE architecture.

Problem Statement

The existing C-ITS projects have their own multi-disciplinary approaches in defining the reference architectures. These architectures were defined using various informal notations. The different categorizations, various modelling approaches, and ad-hoc notations lead to inconsistency in architecture and stakeholder communication. This demands a common modelling approach which can be used among different stakeholders during the development lifecycle.

Architecture Modelling

To meet C-MobILE goals of defining an architecture, a good architecting is required to manage the complexity faced by stakeholders of systems. We extended the conceptual model of an architecture framework of ISO42010 as shown in Figure 1. A good way to describe or express an architecture is to define an architecture description language (ADL) which can capture the essence of project. According to ISO42010 [2], ADL is any form of expression for use in architecture descriptions.

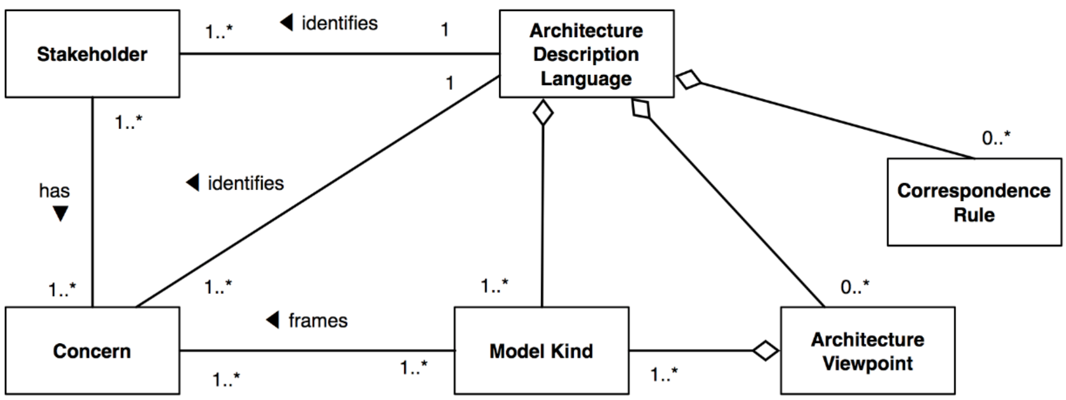

Figure 1 – Conceptual model of an Architecture Description Language (ADL) [2]

An architecture description documents architecture for stakeholders using a set of architecture views and models. An architecture view conforms to an architecture viewpoint which addresses stakeholder concerns. The C-ITS architecture viewpoints are defined in the scope of C-ITS architecture framework [4] which was proposed to establish a common practice for creating, interpreting, analysing and using architecture descriptions within the C-ITS domain. In the context of C-MobILE project, a set of six architecture viewpoints as part of the C-ITS architecture framework are defined. These are Context, Functional, Information, Implementation, Physical, and Communication viewpoints.

Regarding ADLs, different ADLs are proposed for various domains [5]. For example, EAST-ADL (Embedded Architectures and Software Technologies-Architecture Description Language) is an architecture description language for automotive domain. EAST-ADL [7]. AADL (Architecture Analysis and Design language) was developed to model software, hardware, and system architecture of real-time embedded systems such as aircraft, motorized vehicles and medical devices [8]. UML (Unified Modelling Language) is a general-purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system [9]. SysML (Systems Modelling language) of OMG is a general purpose graphical modelling language to support specification, analysis, design and verification of complex systems [10].

For C-MobILE C-ITS architectures, we propose to use SysML modelling language for architectural notations of the C-ITS architectures. SysML consists of structure, requirement, and behaviour diagram types as illustrated in Figure 2. We found SysML as a mature language known to majority of system architects and designers. It is supported by many mature tools such as Enterprise Architect (Sparx Systems) [11], Rational Rhapsody (IBM) [12], and Magic Draw (No Magic) [13].

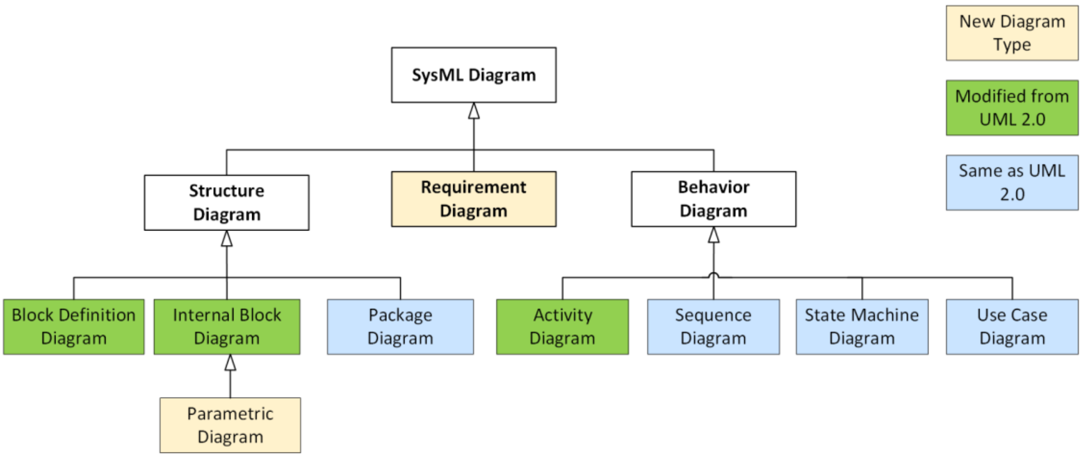

Figure 2- SysML Diagram Types

SysML diagram types are explained below:

- Structure Diagram provides relations between systems, subsystems and their components and interfaces in form of Block Definition Diagram (BDD), Internal Block Diagram (IBD), package and parametric diagrams.
- Requirement Diagram provides cross cutting relationships between requirement and system models.
- Behaviour Diagram-covers use case, state machine, activity and sequence diagrams.

Stakeholders and Concerns

All stakeholders (particularly system architects, developers and integrators) use the functional viewpoint as it is usually easy to understand and describes the system's functional structure. The functional viewpoint addresses the following concerns of the stakeholders.

- Functional capabilities describe the functionality that the system is required to provide. In the Architecture Viewpoint for the reference architecture, it defines what the system will explicitly do. The set of functional and quality requirements will be used to define the functionality.
- External interfaces address the interactions between systems e.g. based on data/control flow or events. A system can send or receive data either because of an internal state change or a state change in another system. A system can send or receive a request to perform a task or notify that something has been occurred. Block Definition Diagrams are identified as suitable modelling notations for capturing the external interfaces.
- Internal structure: A system can be further decomposed into sub-systems or components to meet its requirements. Its decomposition has an impact on the quality attributes e.g. on security, scalability, reliability, and availability. Therefore, the definition of the internal structure should be taken both functional and quality requirements into account. Block Definition Diagrams, Internal Block Definition Diagrams, Component Diagrams are identified as suitable modelling notations for capturing the internal structure.

As illustrated in Figure 3, we define *functional structure model* to address the stakeholder concerns for functional capabilities and external interfaces and *functional internal structure model* to address the internal structure of the element. SysML BDD and IBD can be both used to model the functional structure and the internal structure.

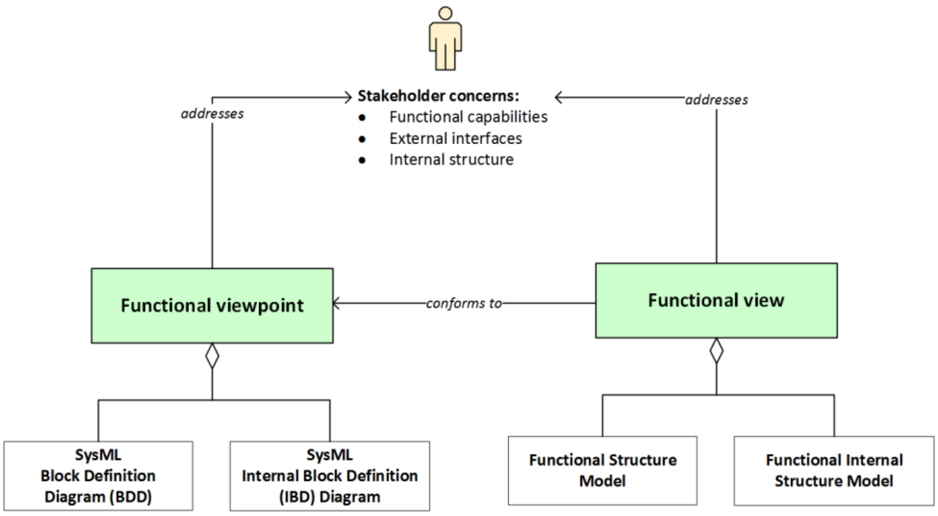

Figure 3. SysML diagram types for Functional viewpoint

The structure of a system is captured in functional structure models using SysML BDD by categorizing into systems and decomposing a system into sub-systems. A system defines the functionality and functional data flow interfaces between systems that are required to support a particular ITS application.

Functional Structure Modelling

For representing the functional structure model, blocks, their contents and relationships are shown using the BDD. The following BDD concepts are used for modelling the functional structure of the block/system:

- Block is a modular unit of system structure that encapsulates its contents which we use to represent a system in the context of C-ITS reference architecture. A system is further decomposed into sub-systems or functional components. A block can both provide and require Interfaces for both information and physical flows.
- A relationship between blocks can be represented by a composition ("has a" relationship) with a solid diamond or a reference with an open diamond.
- The flow port is a new definition from SysML. It represents what can go through a block in and/or out (e.g. data, energy), which will be of use in concrete and implementation architectures.

Functional viewpoint describes the system's runtime functional elements, their responsibilities, interfaces, and primary interactions. The functional view conforms to the functional viewpoint, helps the system's stakeholders understand the system structures, and has an impact on the system's quality properties. The system structure is captured in functional models using SysML block definition diagram (BDD) and by categorizing into systems and decomposing a system into sub-systems. A system defines the functionality and functional data flow interfaces between systems that are required to support a particular ITS application. Information flows depict the exchange of information between sub-systems.

Functional Internal Structure Modelling

The SysML Internal Block Definition Diagram provides an internal or white box representation of a system block. UML 2.0 composite structure diagram, the SysML IBD redefines the composite structure diagram by supporting blocks and flow ports. BDD can be used in combination with IBD i.e. functional structure of the system is represented as trees of modular systems/sub-systems, which further refined into the representation of final assembly of all blocks/systems. Blocks that can be further decomposed into IBD, are called composite blocks.

The following IBD concepts are used for modelling the internal structure of a block/system:

- BDD composite blocks are further decomposed into sub-systems or functional components (which are usually called in IBD as *parts*). Block or system that is further decomposed into sub-systems or functional components are connected via standard ports with exposed interfaces and/or flow ports.

- The item flow is a new definition from SysML as well. It represents the things that flow between blocks/systems and/or sub-systems across their connectors, which will be of use in implementation architecture.

C-MobILE Case Study

This section presents the case study of modelling C-ITS architectures in the context of the C-MobILE project.

Functional Structure Modelling

In Figure 4, the highest level functional model of C-ITS systems is illustrated in terms of systems, sub-systems and their inter-connections at an abstract level, which will be refined further during Concrete Architecture phase. High-level functional models for each System: Central System, Roadside System, Vehicle System, VRU/Traveller System are modelled in SysML Block Definition Diagram.

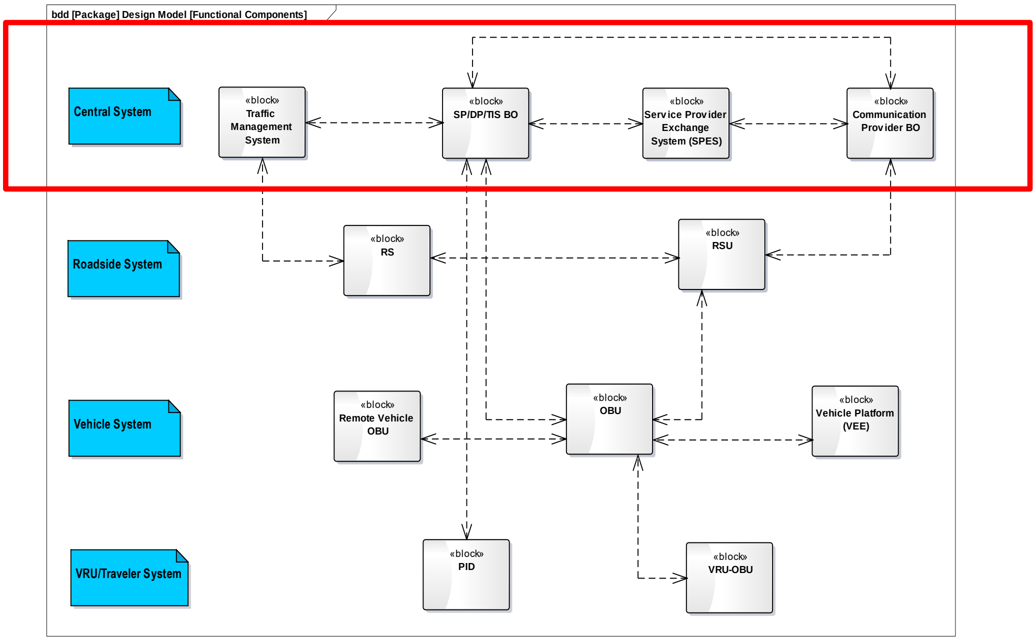

Figure 4 - High-level Functional model of the C-ITS systems for C-MobILE [6]

Functional Internal Structure Modelling

The Central System as highlighted in Figure 4 is decomposed further and captured in IBD in Figure 5. The Central System comprised all the sub-systems which aim to support connected vehicles, field and mobile devices and to perform management and administration functions. The sub-systems in this layer are typically virtual systems that can be aggregated together or geographical or functions distributed. The descriptions of sub-system are described in Table 1.

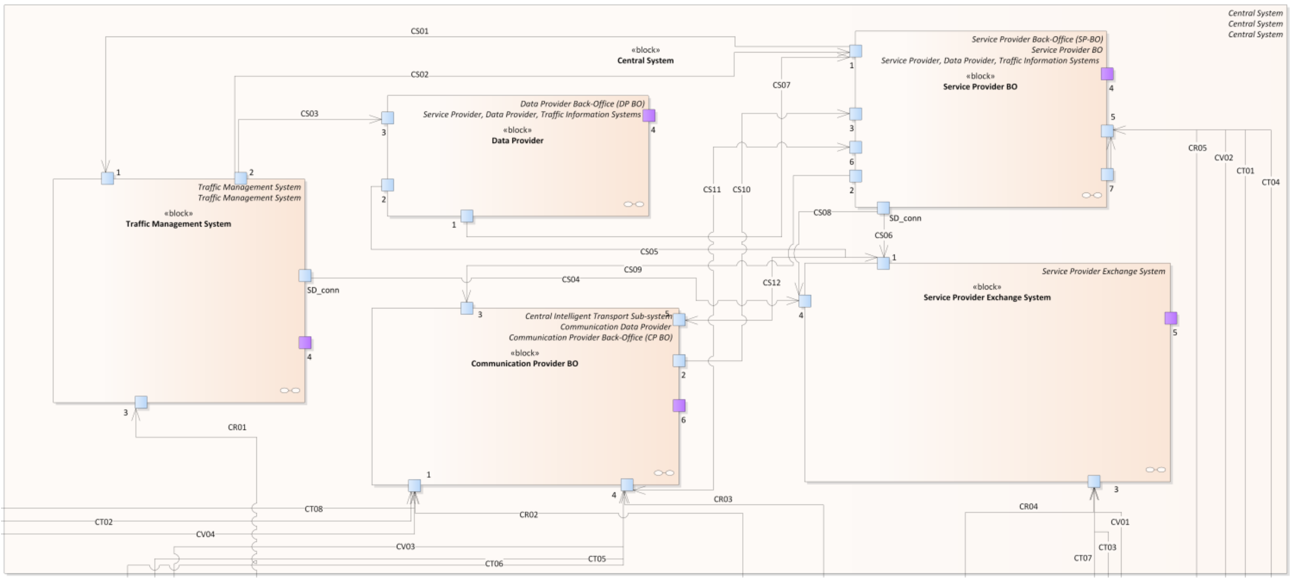

Figure 5 - IBD functional diagram of the Central System

Table 1 - Central Sub-Systems Descriptions

Internal Block	**Description**
Time Management System (TMS)	A TMS is the functional back-office system of the responsible road operator to enforce legal actions on urban or high-way road sections or intersections based on real-time traffic data from loops, cameras, speed sensors, etc. or actions by traffic controllers. The real-time data that flows from the Traffic Info System is integrated and processed by the TMS (e.g. for incident detection), and may result in traffic measures (e.g. traffic routing, dynamic speed limits) with the goal of improving safety and traffic flow.
Data Provider (DP)	A Data Provider BO system is a data system that collects and fuses floating car data and real-time traffic data from roadside sensor systems to increase insight in actual and expected traffic state (e.g. on traffic jams). The DS also distributes enriched (aggregated) information on traffic state (speed, flow and travel times) to service providers.
Communication Provider Back-Office (CP BO)	A generic back-office system of a communication provider used for access at several communication layers from other BO systems (like SP BO, TMS, TIS etc.) to send and receive ITS information to/from vehicles or other users.
Service Provider Exchange System (SPES)	It is an e-Market ('broker') system for discovery and exchange of ITS (end-user) services and ITS communication services; the SPES can support functions like service discovery, service authentication, authorization, accounting (AAA) and billing.
Service Provider Back-Office (SP BO)	A generic back-office system of a service provider used for the specific services of the SP to connected drivers or end-users to inform end users or other SP BO systems from providers. A SP BO can be used to support personal information services for, e.g.

	navigation or traffic information applications on OBU/PID. A SP BO can also be used to gather floating car data from OBU/PID;

Modeling for real-world deployments

In real-world deployments, the functional systems at one of the four layers (i.e. VRU, vehicle, roadside or central layer), can be deployed in one physical box. However, in case the involved functional systems are deployed in separate physical elements a communication network is needed to interconnect the functional systems at different levels. In Figure 6 the involved components and the connections between them are listed for all use cases of the service. The communication architecture for C-MobILE conforms to the general communications reference architecture defined in ETSI EN 302 665.

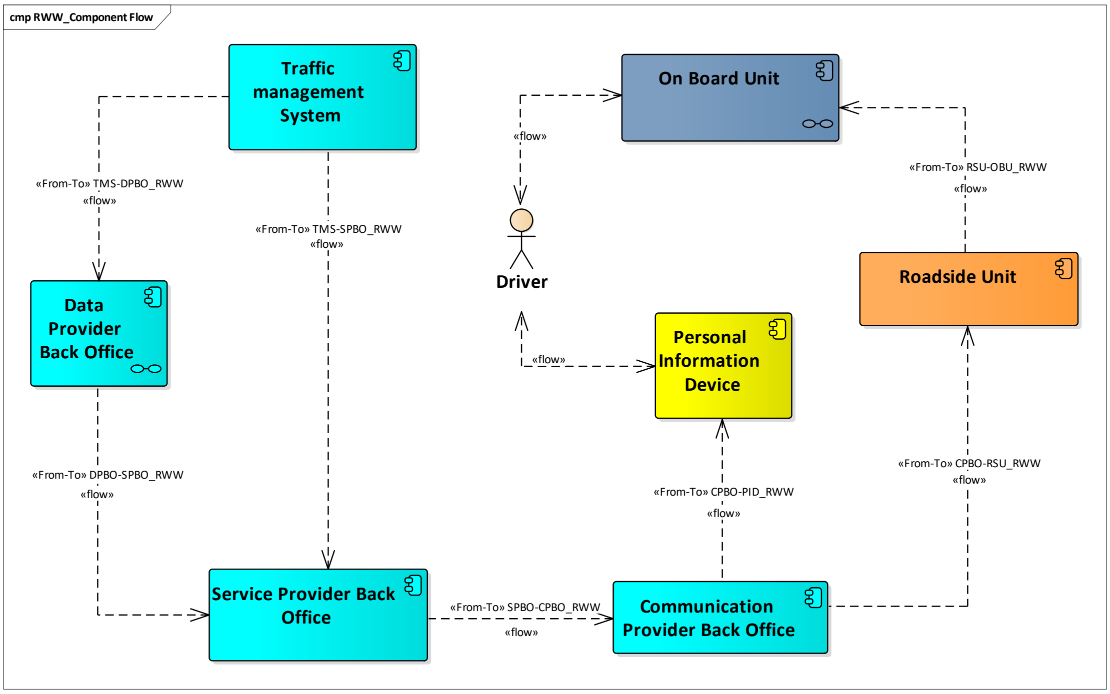

Figure 6 - Road Works Warning - Components and communication flows

The most important data structures and flows need to be considered during the architecture definition. For this purpose, information viewpoint and its respective view are of use. The information view consists of Information Structure Model, Data Flow Model, and Data Lifecycle Model, which can be represented using SysML Block Definition Diagram (BDD), Activity Diagram, and State Diagram respectively. In Figure 7, the sequence diagram for Road Works Warning for the cooperative case is illustrated.

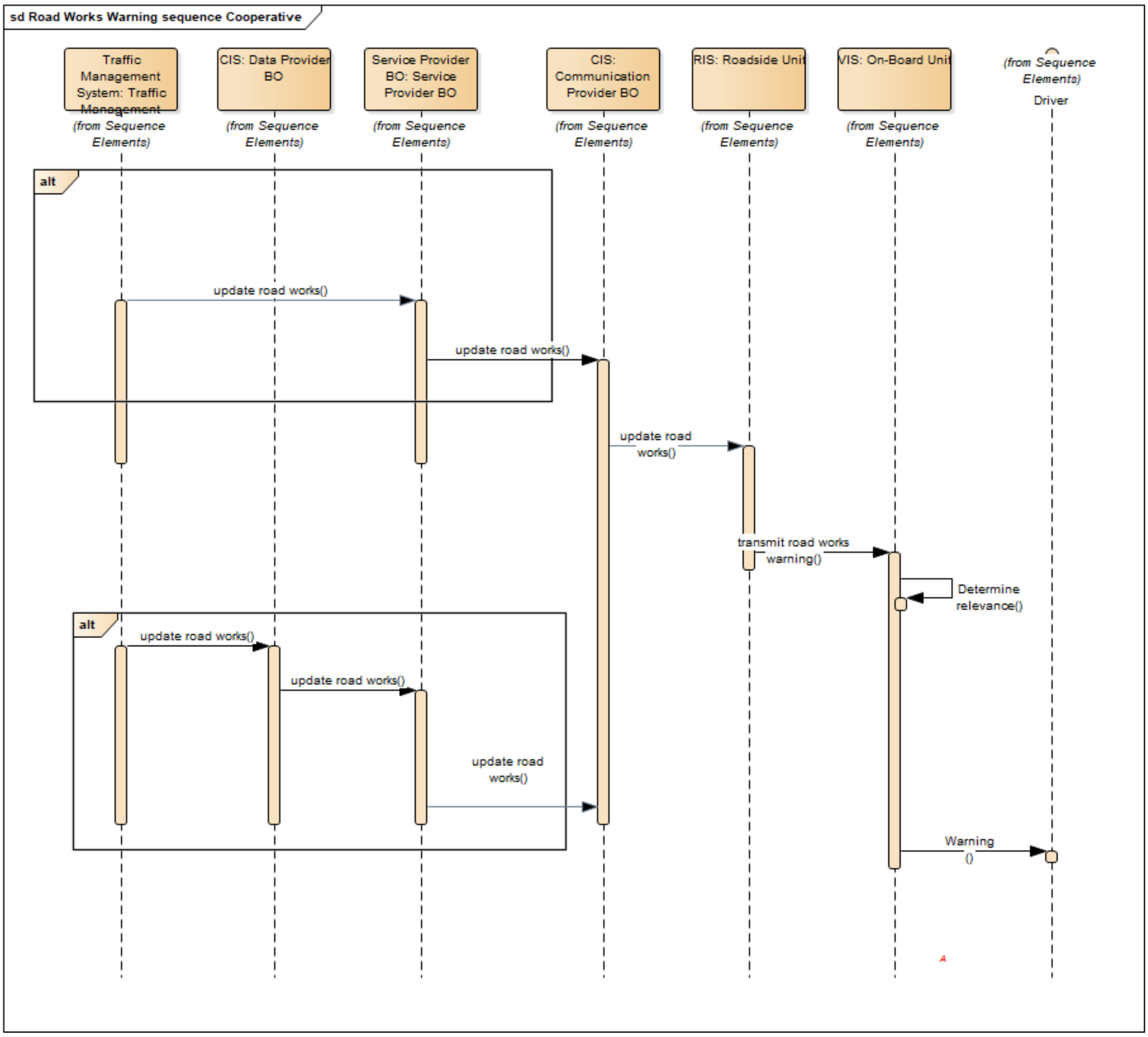

Figure 7 - Road Works Warning Sequence Cooperative

Conclusions and Future work

The C-ITS architecture framework has been developed by defining Architecture Viewpoints and their respective views addressing respective stakeholder concerns [3]. This framework is used for the definition of C-MobILE architectures using SysML as a modelling language facilitating the communication between different stakeholders.

In this paper, we discuss different modelling languages for other domains such as automotive and aerospace, and illustrated a case study of using SysML for modelling the C-ITS architectures for different viewpoints.

Acknowledgement

The C-MobILE project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 723311.

References

1. ISO/IEC/IEEE 42010:2011. Systems and software engineering – Architecture description. http://www.iso-architecture.org/

2. Emery, D., and Rich H. Every architecture description needs a framework: Expressing architecture frameworks using ISO/IEC 42010. Software Architecture, 2009 & European Conference on Software Architecture. WICSA/ECSA 2009. Joint Working IEEE/IFIP Conference on. IEEE, 2009.

3. Rozanski, N. and Woods, E. Software systems architecture: working with stakeholders using viewpoints and perspectives. Addison-Wesley, 2011.

4. Y. Dajsuren, P. Karkhanis, M.G.J van den Brand, M. Funfrocken "C-ITS Architecture Framework and Perspective", in IEEE ITSC 2018, in submission.

5. Y. Dajsuren, "On the design of an architecture framework and quality evaluation for automotive software systems," PhD thesis, 2015.

6. P. Karkhanis, M.G.J van den Brand, S. Rajkarnikar "Defining the C-ITS Architecture", in WASA 2018, in press.

7. Debruyne, V., Simonot-Lion, F., & Trinquet, Y. (2005). EAST-ADL—An architecture description language. In Architecture Description Languages (pp. 181-195). Springer, Boston, MA.

8. Feiler, Peter H., and David P. Gluch. Model-based engineering with AADL: an introduction to the SAE architecture analysis & design language. Addison-Wesley, 2012.

9. D'souza, Desmond F., and Alan Cameron Wills. Objects, components, and frameworks with UML: the catalysis approach. Vol. 1. Reading: addison-Wesley, 1998.

10. Friedenthal, Sanford, Alan Moore, and Rick Steiner. "OMG systems modeling language (OMG SysML™) tutorial." INCOSE international symposium. Vol. 18. No. 1. 2008.

11. Architect, Enterprise. "Sparx systems." (2010).

12. Douglass, B. P. "Analyze system safety using UML within the IBM Rational Rhapsody environment." IBM Rational White Paper, IBM Software Group (2009).

13. Šilingas, Darius, and Rimantas Butleris. "Towards implementing a framework for modeling software requirements in MagicDraw UML." Information Technology and Control 38.2 (2009).