Designing a Reference Architecture for the C-ITS Services

Designing a Reference Architecture for the C-ITS Services

Priyanka Karkhanis(✉), Yanja Dajsuren, and Mark van den Brand

Department of Mathematics and Computer Science,
The Eindhoven University of Technology, Eindhoven, The Netherlands
{p.d.karkhanis,y.dajsuren,M.G.J.v.d.Brand}@tue.nl

Abstract. Cooperative-Intelligent Transport Systems (C-ITS) services aim to improve road transportation through enhanced cooperative, connected, and automated mobility services. The current reference architectures for C-ITS have different abstractions, with some focusing on technical aspects, while others center on specific traffic issues of particular countries. As a result, they have varying technical details and use ad-hoc notations, which limit their adaptability for cross-border deployment of C-ITS services. This paper presents a method for developing a C-ITS reference architecture, involving three essential parts: abstracting common C-ITS systems, describing them using an architecture framework, and reviewing the reference architecture. Using our method, we developed a C-ITS reference architecture for the C-ITS services for deployment sites across European countries. Our reference architecture offers a clear and easily understandable design of C-ITS systems, eliminates the use of ad-hoc notations, and promotes adaptability. This makes it easier for stakeholders to communicate and collaborate across countries, facilitating the deployment of C-ITS services.

Keywords: C-ITS reference architecture · Cooperative Intelligent transport systems · C-ITS

1 Introduction

Cooperative Intelligent Transport Systems (C-ITS), also known as connected vehicle technology in the United States, is a group of technologies and applications that enable efficient communication between Vehicle and X, also known as V2X, where X can be any vehicle, piece of infrastructure, or pedestrian [7,24]. The advantages of C-ITS include increased road safety, decreased traffic, improved transportation efficiency, increased mobility, reduced environmental impact, and support for economic development [2].

Since 2008, and even earlier, C-ITS has been at the center of several initiatives [7]. There have been many C-ITS deployment projects in Europe and USA since that time [15]. The C-ITS services are an innovative array of technologies facilitated by digital connectivity between vehicles and between vehicles

B. Tekinerdogan et al. (Eds.): ECSA 2023, LNCS 14212, pp. 117–132, 2023.
https://doi.org/10.1007/978-3-031-42592-9_8

and transport infrastructure to enhance road safety, traffic efficiency, and driving comfort [20]. The primary functions of C-ITS services are to display regulatory boundaries via signs informing road users of specific obligations, restrictions, or prohibitions; to warn road users of impending incidents and the exact nature of those incidents, and to provide information to road users to improve road safety and comfort during a journey [7].

To promote the deployment of C-ITS services, the development of a reference architecture is seen as a crucial first step. In general, reference architecture provides a shared language, reusable structures, and established practices. The ISO 14813-5:2020 Intelligent Transport Systems standard [12] defines a reference architecture as a high-level overview of the systems and connections relevant to C-ITS services and stakeholders. Reference architectures developed in various projects such as CONVERGE [10], DITCM [21], and NordicWay [26] address several challenges, including technical and operational issues. However, their respective reference architectural designs encompass distinct aspects. For example, NordicWay emphasizes communication to gain a deeper technical understanding of interoperability, while DITCM consists of a system architecture and a description of the business aspects within a Dutch context. From these diverse reference architecture designs, three limitations were readily apparent: First, the information contained within the reference architectures is excessively technical, making it difficult for stakeholders in other nations to comprehend and utilize them. Second, using ad-hoc notations to model reference architectures can lead to fragmentation, with each country making its reference architecture that might not work with designs from other countries. Lastly, the majority of reference architectures are designed for specific objectives or countries, which may not be compatible with diverse traffic patterns and geographical conditions. These problems may cause disconnected and disjointed designs, which limit the usefulness of C-ITS as a whole. The widespread implementation of C-ITS services across European Union nations, along with the resolution of road transport issues, necessitates the adoption of a generic reference architecture. By utilizing the generic architecture, the extensive deployment of C-ITS services across nations becomes achievable [16].

Furthermore, it is important to acknowledge that the comprehensive understanding of C-ITS services, regardless of their extent, is limited due to the lack of publications by the academic and research communities. Studies such as [4,18] highlight a lack of emphasis on sharing information that could be crucial and particularly valuable in advancing cooperative transportation. According to our current knowledge, there is no explicitly defined methodology or procedure for developing the C-ITS service reference architecture. Therefore, we define the following research question to develop a reference architecture that can be applied across deployment sites with different C-ITS services.

RQ: How to develop a reference architecture that can be used to deploy diverse C-ITS services across different deployment sites?

Our goal is to develop a reference architecture that different deployment sites in the European Union (EU) can use as a model for rolling out C-ITS services. To accomplish this goal, we developed an organized method that aided us in producing an efficient reference architecture. We describe our method, which consists of three phases: the abstraction phase, which involves identifying and separating common C-ITS systems; the description phase, which applies architecture standards; and the architecture review phase, which incorporates the feedback of C-ITS stakeholders to verify the reference architecture. These three phases address the three issues previously mentioned. The abstraction phase eliminates unnecessary technical details and maintains a high-level understanding of the systems that can be readily comprehended by stakeholders from different deployment sites across countries. The challenge of addressing ad-hoc descriptions and constructing a more systematic representation of C-ITS systems is addressed during the description phase. In the final phase of the architecture review, stakeholders from various countries verify the reference architecture's applicability at diverse deployment sites. We demonstrate the usage of the proposed method in the C-MobILE project and developed a C-ITS reference architecture that is applied in eight deployment sites across EU [28]. The C-MobILE (Accelerating C-ITS Mobility Innovation and deployment)[1] is a large-scale C-ITS deployment project with around 37 project partners and took place between 2017 and 2021. The project's objective is to demonstrate C-ITS solutions on a large scale in urban and motorway environments by providing C-ITS services to end-users.

This paper is organized as follows: The Sect. 2 describes the method, which is divided into three phases, i.e., the abstraction, description, and review phases. We present the reference architectural models developed using the proposed method in Sect. 3. We share the lessons learned in Sect. 4. Finally, we conclude our paper and discuss future directions in Sect. 5.

2 Method

In this section, we describe the method comprised of three phases-abstraction, description, and architecture review, that were used to create the reference architecture for C-ITS services. The three phases as shown in Fig. 1 address the limitations of excessive technicality of information; ad-hoc notations causing fragmentation and incompatibility between countries' reference architectures; and a lack of adaptability to diverse traffic patterns and geographical conditions.

2.1 Abstraction Phase

The process of abstraction in reference architecture includes making decisions concerning the technologies, applications, and vendors involved [1]. It entails extracting important information from a wider amount of data that may include systems, algorithms, implementation methodologies, multiple approaches, or specific domain expertise [11]. The difficulty lies in determining what kinds of high-level information are sufficiently broad to be included in our C-ITS reference architecture. According to Cloutier et al. [5], a reference architecture should

[1] https://c-mobile-project.eu/.

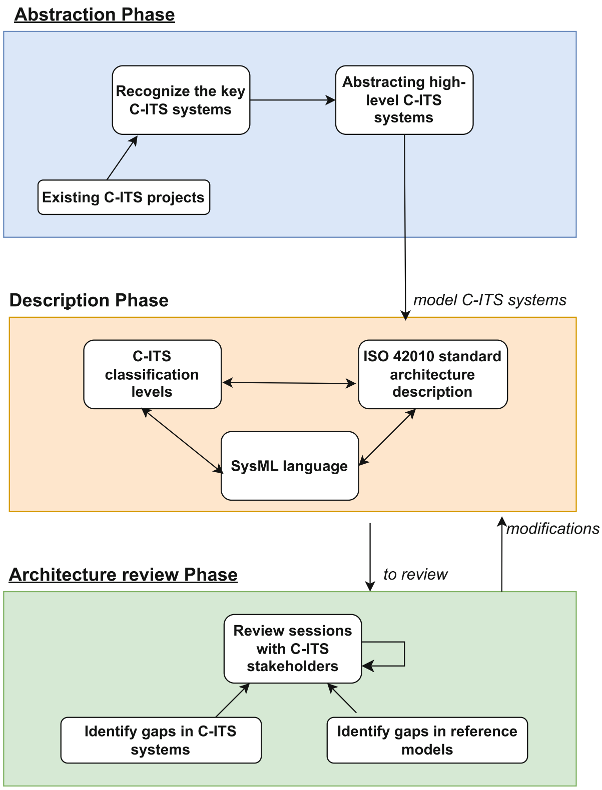

Fig. 1. Reference architecture process - three phases

include "business architecture", "customer context", and "technical architecture" in order to demonstrate its completeness and maturity. However, when systems evolve and get more complex, the scope of what should be included in the reference architecture needs to be adjusted [27]. Therefore, it is essential to define the abstraction phase to demarcate what should be included in our reference architecture. In line with the objectives of the C-MobILE project, our reference architecture restricts the incorporation of excessive technical complexities and focuses on eliciting high-level systems of the C-ITS services. This enables subsequent phases to focus on the dedicated attention required to design the technical architecture and business models specific to each individual C-ITS service. By striking this balance, we ensure that our reference architecture provides a solid framework for the development and deployment of C-ITS services while allowing for necessary customization and adaptability in later stages. As shown in Fig. 1, we do the abstraction in two steps: recognizing the key systems of C-ITS and defining the boundary of the abstraction.

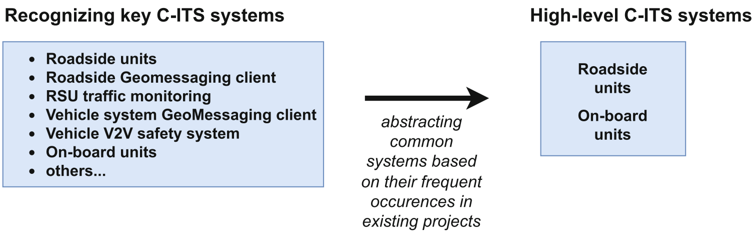

Fig. 2. Abstraction phase

1. Recognizing Key Systems of C-ITS Services. Understanding the fundamental concepts of systems, interfaces, and their relationships is essential for the efficient operation of C-ITS services. The most reliable method to confirm our understanding of current and ongoing C-ITS services is through an analysis of already-completed C-ITS projects. We considered the projects such as DITCM [21], CONVERGE [10], COMPASS4D [19], NordicWay [26], and US-ITS (ARC-IT) [25] that provide the most recent C-ITS services and relevant technologies, as well as areas for improvement. The projects had already achieved TRL5 and TRL7 milestones. The Technology Readiness Level (TRL) scale, originally developed by NASA in the 1990s s and adopted by EU-funded projects as part of the Horizon 2020 framework program in 2014, is used to measure the maturity of a technology, with levels ranging from TRL 1 to TRL 9[2]. The usefulness of a particular TRL depends on the specific objectives and requirements of the project. TRL 5 is better for analyzing basic technology functionality and feasibility. TRL 7 is better for assessing system performance and readiness for operational deployment. We recognized it is essential to include the existing knowledge from TRL 5 to TRL 7, which provided us with a comprehensive perspective extending from basic functionality to deployment details. From these projects, we gathered information including systems, interfaces, protocols, networks, technology, and terminology, as well as their corresponding C-ITS services, regions, and countries. We maintain such information in an internal repository in a structured way that allows us to organize them based on country- or project-specific details. The repository can be updated to reflect any new information, e.g., country or project. With the aid of this step, we were able to identify C-ITS systems that are widely utilized across all EU countries, as well as any regional systems, and define a glossary for the reference architecture. The next step is to identify which data would be suitable for the reference architecture.

2. Defining the Abstraction Boundary. In order to decide what should be a part of a reference architecture, we establish the abstraction phase. Our main objective revolves around the selection of C-ITS systems that exhibit a sufficiently high-level nature. These systems should also be generic, meaning they

[2] https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf.

can be applied to a wide range of scenarios and circumstances. The key criteria we emphasize are flexibility and adaptability, ensuring that the chosen C-ITS systems can be easily adjusted and tailored to suit the specific requirements and characteristics of different deployment sites across various regions. We illustrate our abstraction criterion with roadside and vehicle systems, as shown in Fig. 2. From the collection of key C-ITS systems, we abstracted vehicle systems, such as on-board units (OBUs), as well as their interconnections with other high-level systems, such as roadside units. The C-ITS service uses these terms, "on-board units" and "roadside units", to describe two types of devices. Whatever the specifics of the underlying technology, these systems are always a part of any C-ITS service. The inclusion of this information has been done with the intention of providing a comprehensive overview of C-ITS systems that are common and have been utilized in the past in C-ITS initiatives. Next, we represent them in a structured manner to be understood by interdisciplinary C-ITS stakeholders.

2.2 Description Phase

Inconsistent representations of C-ITS systems in the past have been a significant issue, as they can contribute to confusion and misunderstandings among stakeholders [14]. Ad-hoc diagrams, made without a standardized method or notation, can generate confusion since different people will interpret the same system in different ways. In order to address this issue, we introduce a description phase that incorporates well-established standards and practices for describing C-ITS systems. The ISO 14813-5:2020 Intelligent Transport Systems standard [12] recommends using an established method for defining reference architectures within ITS International Standards to ensure seamless integration for deploying C-ITS services internationally. Incorporating well-established standards and practices facilitates the development of a universal and generic reference architecture that can be applied in multiple countries. As shown in Fig. 3 we used the C-ITS architecture framework [28] compatible with the ISO 42010 international stan-

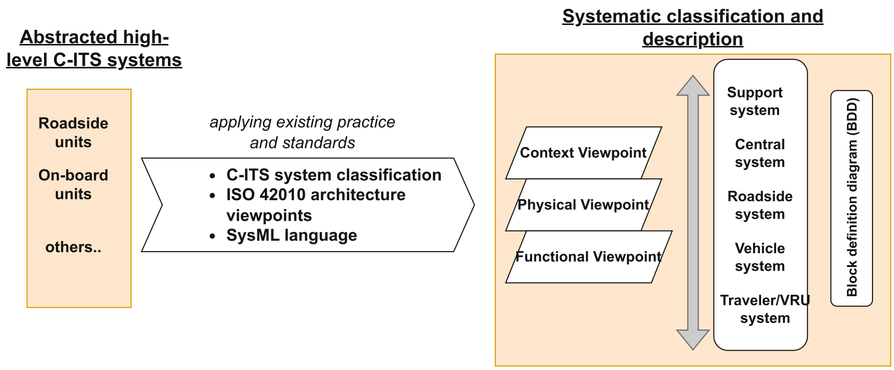

Fig. 3. Description phase

dard [13] for describing the architecture. The ISO/IEC/IEEE 42010 architecture framework facilitates a common practice for creating, interpreting, analyzing, and employing architecture descriptions within a specific application domain or stakeholder community [28]. We used C-ITS system classification, which is a common practice utilized in previous projects [10,25] to group C-ITS systems. To model the systems, we used SysML language which is a general-purpose modeling language to represent complex systems, structures, behaviors, and requirements [9,22]. According to a survey conducted by [22], SysML has been adopted in multidisciplinary domains, including automotive, information technology, and transportation. This indicates that SysML has gained recognition as a modeling language appropriate for various domains and system types, supported by a considerable availability of mature tools that facilitate SysML usage [6,22]. Leveraging established practices in the description of C-ITS systems promotes a comprehensive understanding of the system, enhances communication among stakeholders, and provides a structured and unambiguous representation. Next, we discuss the C-ITS system classification and the architecture descriptions.

C-ITS System Classification is a well-established practice for categorizing C-ITS systems utilized by numerous initiatives, including DITCM [21], CONVERGE [10], and US-ITS [25]. C-ITS classification facilitates the organization and categorization of various C-ITS system types based on their functionality, making it simpler for various stakeholders, such as vehicle manufacturers, infrastructure operators, and service providers, to communicate and collaborate effectively. Based on the analysis of existing C-ITS projects, we propose a system classification for C-ITS consisting of five levels that categorize various C-ITS system types, and those levels are support system, central system, vehicle system, roadside system, and traveler/VRU system. These levels are consistent with the definitions that were applied during the earlier C-ITS studies, e.g., DITCM [21]. These five levels were used to categorize the high-level C-ITS systems that were analyzed during the abstraction phase.

- **Support System** is comprised of sub-systems performing various tasks like governance, test and certification management, security, and credentials management.
- **Central System** is comprised of subsystems such as traffic monitoring and traffic control, which monitor surveillance cameras to capture and process traffic data, which is then disseminated to drivers through traffic control.
- **Roadside System** is comprised of sub-systems that cover the ITS infrastructure on or along physical road infrastructure, like roadside units or signal control.
- **Vehicle System** is comprised of sub-systems that are integrated within vehicles, such as onboard systems, advanced driver assistance, safety systems, navigation, and remote data collection.
- **Traveler/VRU System** is comprised of personal devices, like mobile devices or navigation devices, and specific systems connected to vehicles of VRUs, like tags.

C-ITS Architecture Description is essential for designing, deploying, and managing complex, modern systems that enhance communication and collaboration through coordinated and integrated operations. We used the C-ITS architecture framework that complies with the conceptual model of the international standard ISO/IEC/IEEE 42010 [13]. The international standard provides a comprehensive framework for describing the architecture of a system, ensuring that all stakeholders have a common understanding of the system under development. C-ITS architecture framework consists of five viewpoints that represent the C-ITS systems and their relationships and capture the concerns of various stakeholders [28]. Each viewpoint represents a unique aspect of the system, such as its functionality, structure, behavior, or context. These viewpoints facilitate developers', architects, and end-user's comprehension of the system's architecture. These viewpoints were also utilized in a few C-ITS initiatives, such as COMPASS4D [19] and CVRIA [25]. In this paper, we discuss the context, functional, and physical viewpoints.

- **Context viewpoint** describes how the system interacts with its surroundings (people, systems, and external entities). A context view helps stakeholders comprehend the system's roles and organization.
- **Functional viewpoint** describes the system's runtime functional elements' responsibilities, interfaces, and main interactions. A functional approach helps stakeholders comprehend system structures and affects quality.
- **Physical viewpoint** describes the system's physical deployment and dependencies. A physical view captures the hardware environment, technical environment requirements for each element, and software elements to the runtime environment that will execute them, helping stakeholders deploy, test, and maintain the system.

To effectively model the C-ITS systems in various viewpoints, we employed the SysML Block Definition Diagram (BDD). The BDD allowed us to depict the high-level C-ITS systems, providing stakeholders with essential base-level information about the structure, behavior, and relationships of the C-ITS systems [9,14].

2.3 Architecture Review

To determine whether the reference architecture is efficient, an architecture review must be conducted while the project is still in the planning or development phase [17]. The entire project team, including architects, developers, project managers, and business analysts, reviews the system's progress during the development of the reference architecture. We reviewed our reference models, including the structure, components, interfaces, and interactions with the interdisciplinary C-ITS stakeholders. The review phase helped us identify any architectural flaws early on and ensure that the essential C-ITS systems were accurately captured and readily understood by all parties. Overall, the architecture review served as a quality control safeguard to ensure that the reference architecture for C-ITS systems met the project's objectives [3].

The architecture review process is shown in Fig. 4. Main C-ITS systems were initially identified during the abstraction phase. The C-ITS reference architecture consists of a Glossary that details all the concepts and their respective definitions, Reference Architectural Models that are outlined in SysML with the help of the Enterprise Architect tool, and system descriptions that provide more information regarding the C-ITS systems. These were developed during the description phase. The Glossary and System Descriptions are outside the scope of this paper, but can be found in the reference architecture deliverable of C-MobILE project [28]. During the development process, we validated the identified C-ITS systems and models through focused group sessions consisting of 10–15 (depending upon session's goals) architecture experts, use-case experts, deployment site leaders from different countries and regions, and C-ITS application professionals. The C-ITS architects of the C-MobILE project led these focus group sessions, during which the results such as a reference architecture model, a glossary, and a list of open issues and challenges were presented for immediate feedback and discussion. After analyzing the feedback from the focus group, we refined the C-ITS reference architecture. As a result, we updated the repository and models for each session accordingly. This iterative cycle of weekly revisions lasted for about 4–5 months before the final reference architecture was approved by the C-ITS experts of all C-MobILE deployment sites. The outcomes of our C-MobILE reference architecture models are presented below.

3 Results

In this section, we present the reference architecture models that were developed using the proposed method for the C-MobILE project. In our deliverable [28], a complete and detailed explanation of outcomes can be found.

3.1 C-ITS System Classification

We categorize C-ITS systems into five groups: Support, Central, Roadside, Vehicle, and Traveller/VRU System. We analyzed and reverse architect the systems and subsystems following the steps described in the Sect. 2.1, Abstraction phase. We identified that the DITCM reference architecture covers most of the C-MobILE pre-selected systems. The Fig. 5 shows the high-level systems identified from the abstraction phase.

3.2 C-ITS Architecture Description

We describe our C-ITS reference architecture using the C-ITS architecture framework explained in the Description phase in Sect. 2.2. In this paper, we focus on context, functional, and physical viewpoints.

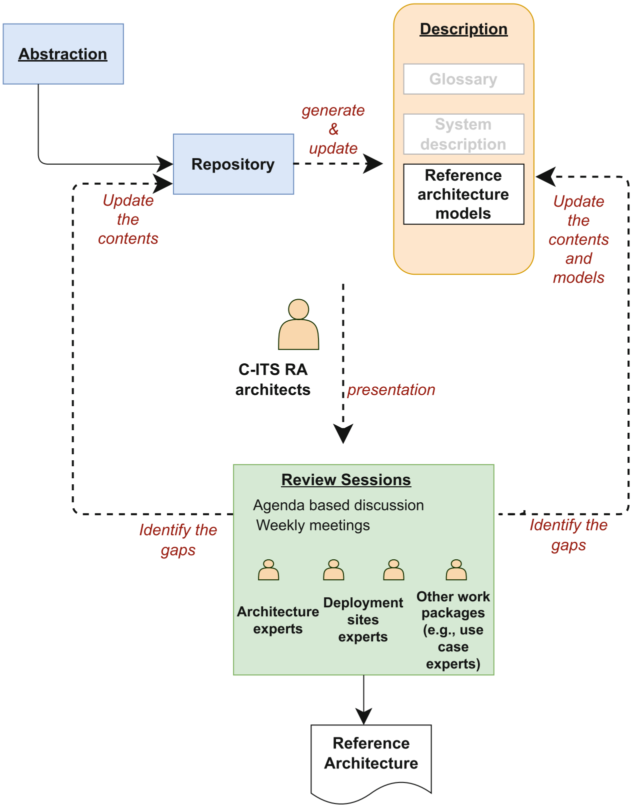

Fig. 4. Architecture review phase

1. Context Viewpoint defines the system's dependencies and interactions with its surroundings (people, systems, and external things). The context perspective helps stakeholders, such as system/software architects, designers, developers, and users to understand the system context. This viewpoint addresses stakeholders' major concerns by describing the high-level system scope and responsibilities by identifying external entities and their relationships. The notations that we commonly see used for context diagrams are SysML Block Definition Diagram. In Fig. 6, the main C-ITS systems are depicted as black boxes and corresponding actors' connections with those systems [28].

– Vehicle Driver: A actor driving motorized vehicles such as cars, buses, and trucks. The actors in this category interact with the vehicle system using the On-Board unit (OBU), Human Machine Interface (HMI), and other vehicle interfaces.
– Vulnerable Road User (VRU): VRUs are people like pedestrians, cyclists, and motorcyclists. These actors interact with Traveler/VRU systems using HMI, tablet, and mobile.

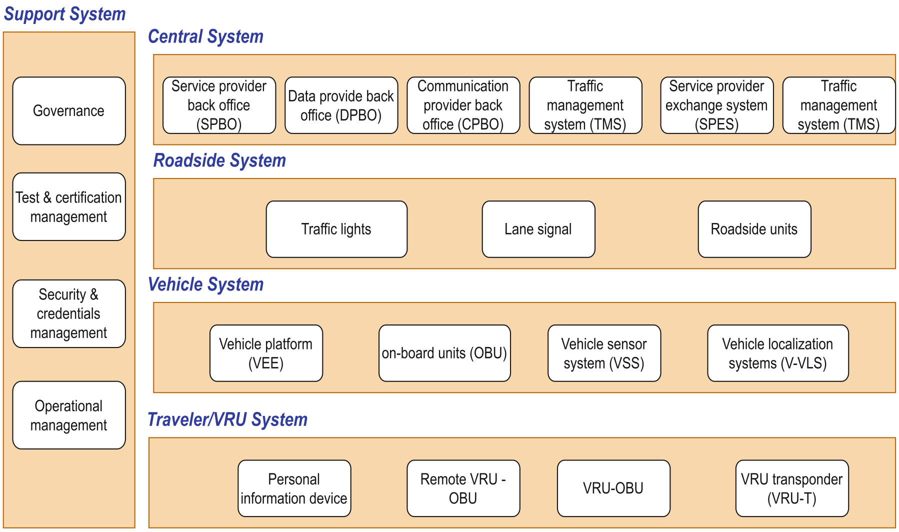

Fig. 5. C-ITS system classification [28]

- Road Operator: This actor collects and analyzes roadside data through multiple communication routes.
- Service Provider: This actor directly supports the Central System, which provides numerous functions such as a navigation provider that provides navigation services and a traffic information provider that provides road traffic information such as traffic jams, incidents, and road work warnings to end users or organizations.
- Governance: This actor directly supports the Support System, which includes legal authorities, test and certification management, security, and credentials management.

2. Functional Viewpoint describes the system's run-time functional elements' responsibilities, interfaces, and main interactions. The functional view helps stakeholders comprehend system structures and affects quality properties. The functional viewpoint is straightforward to understand and describes the system's functional structure, so stakeholders, especially system architects, developers, and integrators use it. Functional capabilities and external interfaces displaying system interaction are stakeholders' major concerns. In Fig. 7, the dependencies between the C-ITS systems across different levels are shown.

3. Physical Viewpoint considers the topology of subsystems and their physical linkages at each domain of interest. Subsystems contain functional components that provide ITS application functionality and interfaces. Users, architects, system maintainers, and OEMs are the major stakeholders in the Physical

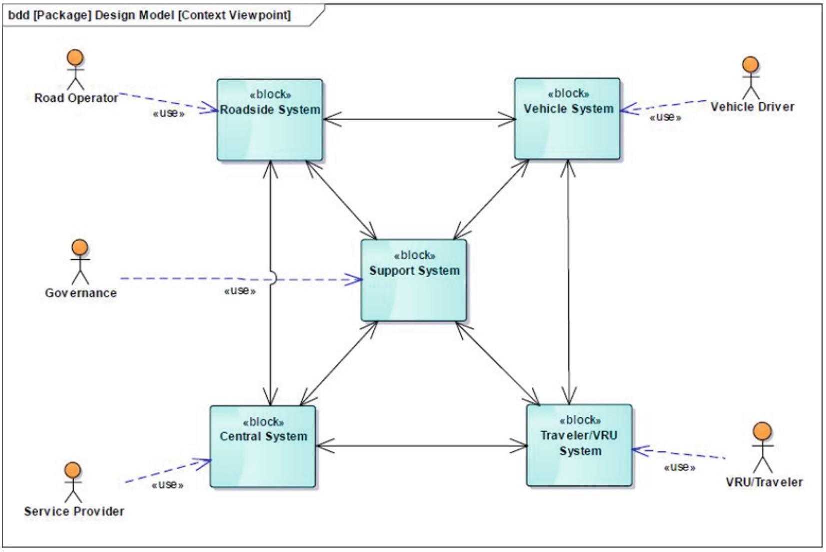

Fig. 6. Context model [8, 14, 28]

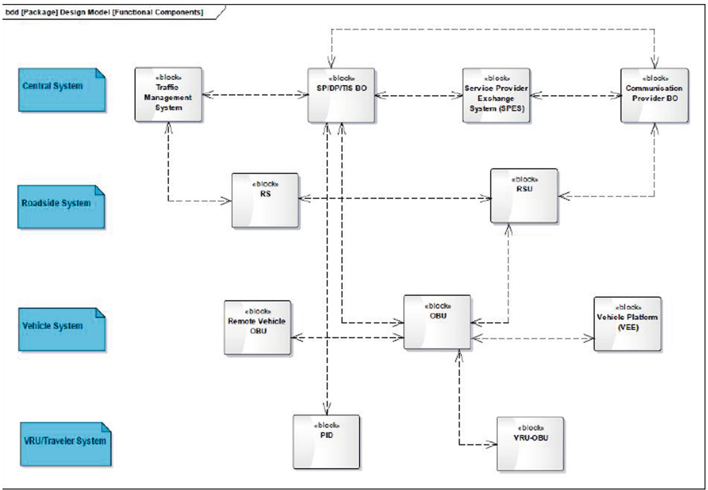

Fig. 7. Functional models [8, 14, 28]

Viewpoint. This viewpoint addresses stakeholders' concerns about physical component decomposition and specification. In Fig. 8, the C-ITS systems for each C-ITS system classification level are highlighted.

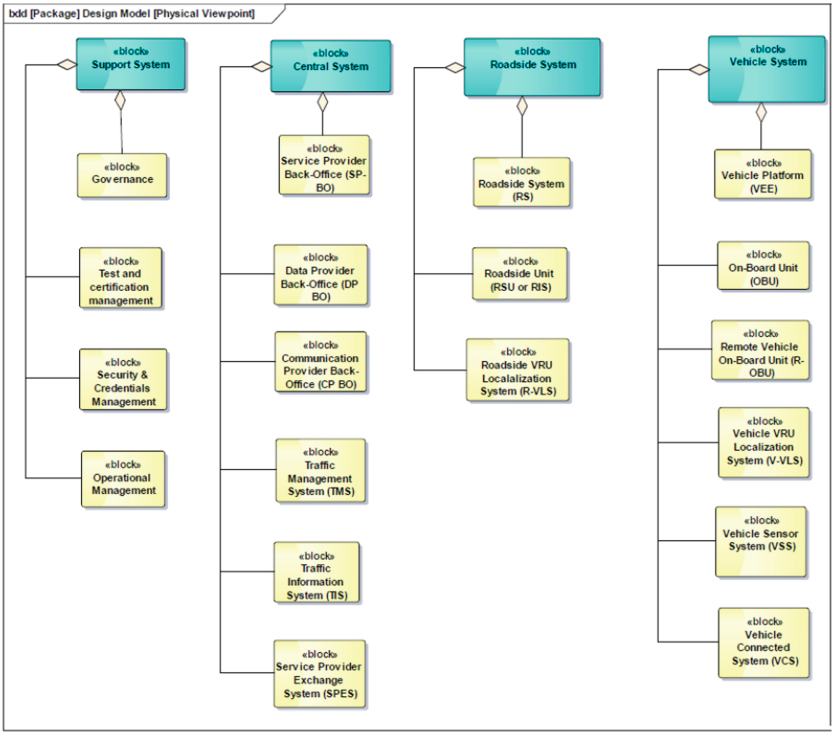

Fig. 8. Physical models [8,14,28]

4 Lessons Learnt

We described the method for developing a reference architecture for C-ITS services and demonstrated its usage in creating a C-ITS reference architecture in the context of C-MobILE project. We share below the lessons learned from both method and their usage in creating a C-ITS reference architecture:

- Dividing the reference architecture design process into distinct phases such as abstraction, description, and review can help in streamlining the process and making it more manageable among interdisciplinary stakeholders from different countries.
- All phases require stakeholders, including architecture specialists from deployment sites and related projects. These phases clarify and improve collaboration. For example, deciding on how to abstract, describe, and assess early on can save time by eliminating redundant discussion on concept definition and description language. It showed that effective participation with clear phases can offer useful expertise and opinions, align concepts, and ensure that the reference design reflects real-world use cases.
- While it was a lengthy process to capture the information in the repository during the abstraction phase of identifying the key C-ITS systems, doing so

served the aim of collecting all data in one place. However, a more efficient technique could be implemented if time and resources are not a constraint.
- As discussed earlier, reference designs are frequently described without a generic language or notations, causing confusion. By selecting the widely utilized modeling language SysML, supported by robust tools like Enterprise Architect and IBM Rational Rhapsody, we successfully minimized unnecessary communication challenges and errors in architecture modeling [22].
- Organizing a meeting with numerous stakeholders can be challenging, which is why we opt for iterative meetings with clear and pre-defined goals during the architecture review phase that concentrates on specific aspects. For example, we prioritize discussions centered around vehicle-to-vehicle C-ITS services and the associated C-ITS systems.
- Given that none of the related projects applied an architecture framework conforming to the ISO 42010 international standard, utilizing an architecture framework was useful in defining the views and perspectives that addressed the concerns of the C-MobILE project's stakeholders [14].

5 Conclusion and Future Work

This paper describes the design process for a C-ITS reference architecture. It seeks to address the lack of a method to develop a comprehensive C-ITS reference architecture across different deployment sites. Existing C-ITS reference architectures lack adaptability and are region-specific. We propose therefore a method involving abstraction, description, and architecture review phases. In the abstraction phase, we analyzed existing C-ITS initiatives to identify essential systems and categorize them. In the description phase, these systems were modeled in SysML following the C-ITS architecture framework proposed in [28]. In the review phase, the reference architecture models were evaluated by diverse C-ITS architects and experts, ensuring a high-quality architecture design satisfying the needs of the different deployment sites. The reference architecture models for the C-MobILE project were created using the proposed method. The reference architecture was instantiated by the 8 deployment sites and further helped realize C-ITS services defined in the C-MobILE project.

There are several opportunities for future research in the field of reference architecture for C-ITS. While the method has been developed in the context of European countries, extending its applicability to other countries requires careful consideration of unique factors such as technological infrastructure, industry collaborations, and public acceptance. These factors determine C-ITS service implementation and readiness [23]. Therefore, it is crucial to acknowledge these variations and adapt the method accordingly. Although the reference architecture created using the proposed method was used in the eight deployment sites, there was no feedback after the implementation of the C-ITS services because of the nature of the project plan. Existing evaluation methods for software architecture e.g., ATAM, SAAM may not be suitable for evaluating the reference architecture of C-ITS services. Therefore, further study is needed to develop a more nuanced evaluation approach specifically for reference architectures in this field. Additionally, exploring methods for updating the reference architecture to accommodate future modifications would be valuable.

Acknowledgment. The C-MobILE project is funded by the European Union's "Horizon 2020 research and innovation programme" under grant agreement No 723311.

References

1. Angelov, S., Grefen, P., Greefhorst, D.: A classification of software reference architectures: analyzing their success and effectiveness. In: 2009 Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture, pp. 141–150. IEEE (2009)
2. Asselin-Miller, N., et al.: Study on the deployment of C-ITS in Europe: Final report. Report for DG MOVE MOVE/C 3, 2014-794 (2016)
3. Babar, M.A., Gorton, I.: Software architecture review: the state of practice. Computer **42**(7), 26–32 (2009)
4. Botte, M., Pariota, L., D'Acierno, L., Bifulco, G.N.: An overview of cooperative driving in the European union: policies and practices. Electronics **8**(6), 616 (2019)
5. Cloutier, R., Muller, G., Verma, D., Nilchiani, R., Hole, E., Bone, M.: The concept of reference architectures. Syst. Eng. **13**(1), 14–27 (2010)
6. Dajsuren, Y., van den Brand, M., Serebrenik, A., Huisman, R.: Automotive ADLS: a study on enforcing consistency through multiple architectural levels. In: Proceedings of the 8th international ACM SIGSOFT conference on Quality of Software Architectures, pp. 71–80 (2012)
7. European Union: The c-roads platform an overview of harmonized C-ITS deployment in europe (2020). https://www.c-roads.eu/fileadmin/user_upload/media/Dokumente/C-Roads_Brochure_2021_final_2.pdf
8. Ferrandez, R., Dajsuren, Y., Karkhanis, P., Fünfrocken, M., Pillado, M.: Modeling the C-ITS architectures: C-mobile case study. In: ITS World Congress (2018)
9. Friedenthal, S., Moore, A., Steiner, R.: Omg systems modeling language (omg sysml) tutorial. In: INCOSE International Symposium, vol. 9, pp. 65–67. Citeseer (2006)
10. Fünfrocken, M., Vogt, J., Wieker, H.: The converge project - a systems network for its (2021)
11. Houngbo, H., Mercer, R.E.: Method mention extraction from scientific research papers. In: Proceedings of COLING 2012, pp. 1211–1222 (2012)
12. ISO: Iso 14813–5:2020(en) intelligent transport systems - reference model architecture(s) for the its sector - part 5: Requirements for architecture description in its standards (2020)
13. ISO/IEC/IEEE: Systems and software engineering - architecture description. ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471–2000), pp. 1–46 (2011). https://doi.org/10.1109/IEEESTD.2011.6129467
14. Karkhanis, P., van den Brand, M.G., Rajkarnikar, S.: Defining the C-ITS reference architecture. In: 2018 IEEE International Conference on Software Architecture Companion (ICSA-C), pp. 148–151. IEEE (2018)
15. Kotsi, A., Mitsakis, E., Tzanis, D.: Overview of C-ITS deployment projects in Europe and USA. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), pp. 1–6. IEEE (2020)

16. Lu, M., et al.: Cooperative and connected intelligent transport systems for sustainable European road transport. In: Proceedings 7th Transport Research Arena, Vienna, Austria (2018)
17. Maranzano, J.F., Rozsypal, S.A., Zimmerman, G.H., Warnken, G.W., Wirth, P.E., Weiss, D.M.: Architecture reviews: practice and experience. IEEE Softw. **22**(2), 34–43 (2005)
18. Marilisa, B., Luigi, P., Nicola, B.G., et al.: C-ITS communication: an insight on the current research activities in the European union. Int. J. Transport. Syst. **3** (2018)
19. Mitsakis, E., et al.: Large scale deployment of cooperative mobility systems in Europe: Compass4d. In: 2014 International Conference on Connected Vehicles and Expo (ICCVE), pp. 469–476. IEEE (2014)
20. Mitsakis, E., Kotsi, A.: Cooperative intelligent transport systems as a policy tool for mitigating the impacts of climate change on road transport. In: Nathanail, E.G., Karakikes, I.D. (eds.) CSUM 2018. AISC, vol. 879, pp. 418–425. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-02305-8_51
21. van Sambeek, M., et al.: Towards an architecture for cooperative-intelligent transport system (C-ITS) applications in the Netherlands. Eindhoven University of Technology, Technical Report Beta Working Paper series (485) (2015)
22. Santos, T.L., Soares, M.S.: A survey on what users think about sysml. Syst. Eng. (2023)
23. Shaaban, K., Elamin, M., Alsoub, M.: Intelligent transportation systems in a developing country: benefits and challenges of implementation. Transport. Res. Procedia **55**, 1373–1380 (2021)
24. Sjoberg, K., Andres, P., Buburuzan, T., Brakemeier, A.: Cooperative intelligent transport systems in Europe: current deployment status and outlook. IEEE Veh. Technol. Mag. **12**(2), 89–97 (2017)
25. USD of Transportation: Architecture reference for cooperative and intelligent transportation. https://www.arc-it.net/
26. Europe Union: Nordicway - architecture, services and interoperability (2017). https://www.nordicway.net/
27. Wang, Q., von Tunzelmann, N.: Complexity and the functions of the firm: breadth and depth. Res. Policy **29**(7–8), 805–818 (2000)
28. Yanja Dajsuren, P.K., Kadiogullari, D., Fünfrocken, M.: C-MobILE d3.1 reference architecture (2017). https://c-mobile-project.eu/wp-content/uploads/sites/19/2018/06/C-MobILE-D3.1-ReferenceArchitecture-v1.1.pdf