

Integrating C-ITS Architecture*

1stPriyanka Karkhanis

Mathematics and Computer Science Department
TU/e
City, Country
email address

2nd Mark van den Brand

dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

3rd Saurab Rajkarnikar

dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address

Abstract—C-ITS (Cooperative Intelligent Transport Systems) architecture is an initiative to facilitate cooperative, connected and automated mobility. It is based on the concept of System of Systems architecture and promotes a new way of thinking for solving grand challenges where the interactions of technology, policy and economics are the primary drivers. The C-ITS domain comprises widely spread systems (e.g. Traffic Management System, Traffic Light Controller, and Vehicle On-Board Units) with their independent usages, but demand for a strategy to facilitate the convergence of these complex and heterogeneous systems. The main objective of C-ITS architecture is to define an integrated architecture based on a number of existing C-ITS projects. The architecture provides a way to standardize and a unifying modeling approach by means of a common language that can be reused by other organizations to guide their internal development processes. The architecture reflects a common understanding of how the future ITS landscape will evolve. The architecture and its concepts are based on the conceptual model of the ISO/IEC/IEEE 42010 international standard for architecture descriptions of systems, system of systems and software. It defines architecture viewpoints for C-ITS systems and uses the concept of architecture perspective for shaping these architecture viewpoints. The architecture process has been split in three phases. The first phase is a high level reference architecture identifying common systems and components at a conceptual level. The second phase is the concrete architecture refining the first phase by focusing on general functionalities and interfaces required between C-ITS systems. The third phase is the implementation architecture defining functionalities more in detail and provides deployment across borders. We demonstrate this by means of C-MoBILE reference architecture which allows large scale demonstrations of integrations of C-ITS systems across Europe.

Index Terms—component, formatting, style, styling, insert

I. INTRODUCTION

The C-MoBILE (Accelerating C-ITS Mobility Innovation and deployment in Europe) project is a European Union (EU) project that spans across eight C-ITS equipped deployment sites and regions with more than 36 participating institutes and companies. It aims to make road transport safe and efficient while decreasing casualties and serious injuries on European roads. The eight C-ITS equipped deployment sites already have some C-ITS services through various projects that took place in the past. However, many of these are not interoperable. Thus, C-MoBILE plans to become a common approach that ensures interoperability with seamless service and become a basis for large scale deployment in Europe. It

is working with various public and private stakeholders while carrying out and developing cost effective business models particularly from the end users perspective.

To help reach the project goals, a good architecture, capable of being deployed to whole Europe is needed. The architecture definition process in C-MoBILE has been defined to support the following sub-goals:

- To analyse existing C-ITS architectures to provide common concepts and vocabulary.
- To identify a set of patterns that have been detected (or applied implicitly) during the analysis of existing C-ITS architectures and their implementations.
- To create a C-ITS reference architecture that enables pan-European interoperability of C-ITS (concrete/implementation) architectures based on the generalization of existing C-ITS architectures.
- To define an implementation architecture specifying components and their relationships (interfaces) guided and constrained by the C-ITS reference architecture.
- To identify service-relevant parts of the architecture and define services based on the business analysis.

Some C-ITS projects that are/were being deployed and are considered for C-MoBILE are: The Dutch C-ITS Reference Architecture[3], Spookfiles A58[4], Praktijkproef Amsterdam[5], C-ITS Corridor[6], MOBiNET[7], VRUITS[8], CONVERGE[9], COMPASS4D[10], NordicWay[11] and ARC-IT[12].

II. PROBLEM STATEMENT

C-MoBILE architecture aims for large scale demonstration project across various pilot sites. These various deployment sites either have their own defined ITS architecture. These various pilot sites have their own multidisciplinary approach towards their specific deployed strategy. These architectures comprise of different design patterns in the form of informal notations. There is no standard notation to help in merging these architectures into a silo-based architecture. An architecture that should not change the existing technology but to harmonize with other existing architectures. The C-ITS domain covers not only software/system engineering field, but also traffic engineering, civil engineering, information technology etc., which require a unified definition of architecture for the C-ITS domain. Without the common C-ITS architecture framework, different categorizations and ad-hoc notations have

been used in the existing C-ITS architectures. However, it's a challenge to consolidate existing architectures ensuring that the concerns such as interoperability, security, availability, and maintainability shall be addressed.

III. METHODOLOGY

We considered various C-ITS projects as mentioned in Introduction section. Apart from these C-ITS architectures, we considered current ongoing ITS implementation for the cities involved in C-MoBILE. We applied bottom-top approach or reverse architect approach by extracting the systems, protocols, networks etc. from all of these architectures manually. These extracted information was required at given architecture phases based on their requirements. A repository has been constructed consolidating all the necessary and required information from existing architectures. A thorough analysis was done to extract the commonalities, but not leaving behind their specific implementation details. As a well-defined architecture framework is considered to be an important part of any architecture description [13], we define an architecture framework for the C-ITS domain. To put architecture framework and architecture description concepts in context, we extend the conceptual model of the ISO/IEC/IEEE 42010 architecture framework. The C-ITS architecture framework specifies stakeholders, their concerns, viewpoints, model kinds, and correspondence rules. In addition, architecture perspectives related to C-MoBILE project will be addressed. C-ITS architects can use an architecture framework to represent the C-ITS reference architecture, concrete, implementation and pilot site architectures. An architecture framework is one of the widely-applied approaches used in software/system architecting of complex systems. The architecture frameworks facilitate communication and cooperation between different stakeholders during architecting and building complex systems such as C-ITS. Many different stakeholders with their interweaving concerns require a systematic approach for addressing complexity and full lifecycle of the system.

To design C-MoBILE architecture framework strategy, the architecture process has been split into three parts: the reference architecture for defining high-level architecture, the concrete architecture for refining the high-level architecture into the medium-level architecture, and the implementation architecture for revising further the concrete architecture into more detailed low-level architecture. In the first design phase, the reference architecture has been created by analysing existing architectures, described in next section. In addition, the current architectures in the various pilot sites have been taken into account. In parallel, use-cases, business-cases, and requirements to the C-MoBILE system have been collected. The C-ITS architecture framework is also defined in the scope of this design phase which will be used to describe the C-MoBILE concrete, implementation, and pilot site architectures. During the second design phase, the reference architecture will be used to create the medium-level concrete architecture. Furthermore, services, interfaces, and concepts will be described to provide a guideline for the final stage. In the third design

phase, interfaces and concepts will be described in detail to create a low-level implementation architecture.

We propose to use Systems Modelling Language (SysML) diagram types for architectural notations of the C-ITS architectures. The SysML is a general purpose modelling language for engineering systems and consists of structure diagram, requirement diagram, and behavior diagram. The architecture framework and the same modelling approach can enable common language and will be reused for the next deployment projects. Furthermore, organizations can use the architecture to guide their internal development process as it reflects a common understanding of how the (future) ITS landscape will evolve.

IV. C-MOBILE REFERENCE ARCHITECTURE

As a result of the architecture analysis and reverse architecting process, we have extracted the reference architecture from existing reference architectures, which was consistent with the DITCM reference architecture. Therefore, the C-MoBILE reference architecture is adopted from the DITCM reference architecture. We defined our reference architecture and result was written in ITS World Congress 2018 which is currently in submission.

C-MoBILE Reference Architecture focuses more on an abstract level and use black box approach wherever possible. It describes various systems at a high level in the form of models using SysML, a general purpose modeling language for engineering systems. The SysML Block Definition Diagram and Internal Block Diagram are used to represent the models for the reference architecture at an abstract level providing base level information to architects. However, due to the heterogeneous nature of such interfaces this will not be possible for all interfaces of the architecture. For example, there exists several competing standards for roadside infrastructure to communicate with traffic management centers. C-MoBILE reference architecture does neither have the resource nor the intention to change/redefine all those standards. Instead, at high level we highlight the common systems, their interfaces and protocols by considering various existing projects to ensure interoperability.

The architecture framework and reference architecture should be used as a basis for developing concrete, implementation and C-MoBILE pilot site architectures. This will enable C-MoBILE deployment at EU defined eight pilot sites and beyond.

V. CONCLUSIONS AND FURTHER WORK

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

REFERENCES

- [1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529–551, April 1955.
- [2] J. Clerk Maxwell, *A Treatise on Electricity and Magnetism*, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in *Magnetism*, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
- [4] K. Elissa, “Title of paper if known,” unpublished.
- [5] R. Nicole, “Title of paper with only first word capitalized,” *J. Name Stand. Abbrev.*, in press.
- [6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” *IEEE Transl. J. Magn. Japan*, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].
- [7] M. Young, *The Technical Writer’s Handbook*. Mill Valley, CA: University Science, 1989.