

On the Use of Orthogonal Context Uncertainty Models in the Engineering of Collaborative Embedded Systems

Torsten Bandyszak¹, Patrick Kuhs¹, Jasmin Kleinblotekamp¹, Marian Daun¹

Abstract: Collaborative embedded systems operate in highly dynamic and uncertain contexts. For instance, in vehicle platooning, several embedded systems form a collaborating group. Among others, they have to cope with unclear intentions of collaborators, uncertainties caused by imperfect technology, and the inherent indeterminacy in the physical surroundings. Hence, it is crucial to systematically consider and model context uncertainty during the development of these systems. Existing uncertainty modeling approaches mainly focus on one specific artifact type, and do not consider the need for supporting purpose-driven abstractions and corresponding perspectives on a system's operational context. In this paper, we build upon the SPES XT Context Modeling Framework, which employs three different perspectives, each using a specific diagram type. Based on state-of-the-art concepts related to uncertainty, we investigate the use of capturing uncertainty in separate, orthogonal models, and establish relations to the three different context perspectives. We illustrate our promising results using a vehicle platooning scenario.

Keywords: Collaborative embedded systems, Uncertainty, Orthogonal modeling, Context modeling, Requirements engineering

1 Introduction

Nowadays, embedded software systems are increasingly interconnected in order to realize more complex functionality and thus achieve higher goals that a single embedded system is not able to achieve on its own. For instance, car platooning is an application area where several systems form dynamic, collaborating groups aiming at avoiding traffic jams and increase safety as well as driving comfort (cf., e.g., [Ji16]). Cooperative Adaptive Cruise Control (CACC) systems [Fe16] are supposed to use inter-vehicle communication and exchange of information to enhance platooning functionalities and automate typical platooning maneuvers such as joining or leaving a platoon. Such collaborative embedded systems operate in highly dynamic contexts, and thus have to cope with various uncertainties related to the context and interactions between different CACC systems in a platoon. These uncertainties often indicate potential safety hazards, and play an important role in safety and risk assessments [Ax17]. Therefore, it is crucial to systematically consider and model context uncertainty during the development of these systems. In general, handling uncertainty is a major challenge in system development, and there are various definitions of uncertainty available in the literature (see, e.g., [RJC12] for a compilation of uncertainty sources).

¹ University of Duisburg-Essen, paluno – The Ruhr Institute for Software Technology, Gerlingstr. 16, 45127 Essen, {Torsten.Bandyszak; Patrick.Kuhs; Jasmin.Kleinblotekamp; Marian.Daun}@paluno.uni-due.de

Several approaches towards documenting uncertainty in a model-based manner have been proposed. Typically uncertainty information is documented in engineering artifacts, for instance, by annotating missing information in partial models [FaSa13], or capturing uncertainty factors in goal models (e.g., [Ch09]). Recently, Zhang et al. [Zh17b] proposed a comprehensive UML profile, which allows capturing a wide range of uncertainty information. However, these approaches do not explicitly account for consistently tracing uncertainty information through different views on the system to be developed, and its context. In requirements engineering, usually three different perspectives are distinguished: the *structural perspective*, the *functional perspective*, and the *behavioral perspective* [JD91]. These three perspectives are also suitable for systematically modelling the operational context of embedded systems, i.e., the context in which they operate at runtime [Da16]. Documenting relevant uncertainty information in each of these three context perspectives easily results in complex models that are hard to understand, quite similar to the management of variability information in the development of software product lines (see [Ba03]). This is particularly the case for collaborative embedded systems as models of these systems are already large on their own. Furthermore, uncertainty information is often documented redundantly, since there is no separate, dedicated artifact capturing uncertainties.

This paper investigates the use of orthogonal uncertainty modeling techniques in the engineering of collaborative embedded systems that need to operate in highly dynamic, uncertain contexts. To this end, we illustrate the possibilities of orthogonal uncertainty modelling, based on the SPES XT Context Modeling Framework [Da16] and a subset of uncertainty concepts taken from the comprehensive taxonomy of uncertainty proposed by Zhang et al. [Zh16]. The key solution idea is to handle uncertainty as a cross-cutting concern that manifests in different diagram types, similar to variability (cf., e.g., [Sc16]). For modeling uncertainties in an orthogonal manner, we use SysML Requirements Diagrams. Relationships are established to context models from the three perspectives mentioned above. We illustrate the orthogonal modeling approach using a case example focused on one particular platooning scenario, i.e., the joining maneuver, where a vehicle joins a platoon. First results show that our orthogonal approach aids the systematic documentation and analysis of context uncertainties.

In Section 2, an overview of fundamental approaches and existing works related to our approach is given. Section 3 introduces the case example and illustrates uncertainties within a specific platooning scenario. In Section 4, we describe our approach towards orthogonal context uncertainty modelling. Finally, Section 5 concludes the paper.

2 Fundamentals and Related Work

This section briefly introduces the SPES XT Context Modeling Framework, and provides an overview of existing approaches towards uncertainty modelling.

2.1 SPES XT Context Modeling Framework

The SPES XT Context Modeling Framework [Da16] allows for seamlessly integrating context models into the development of embedded systems. It separates the *context of knowledge*, which comprises sources of knowledge relevant for system development, from the *operational context*, i.e., the context in which a system is embedded at runtime. The latter is in focus of this paper, and briefly summarized here.

Based on a well-established classification in requirements engineering (cf. [JD91]), the operational context is structured into the following three distinct perspectives: The *structural operational context* focuses on physical context objects that interact with the system at runtime. Here, static structures and relationships are modeled, for instance, using SysML Block Definition Diagrams. In contrast, the *functional operational context* employs functional modelling languages, e.g., based on UML Activity Diagrams, in order to abstract from static properties of context objects, and focus on the functionality they provide. Dependencies between context functions and the system's functions can also be modeled. The third perspective, i.e., the *behavioral operational context*, allows for modeling the externally observable behavior of context objects. To this end, automata-based modeling languages such as state machines can be used.

2.2 Approaches for Modelling Uncertainty

Coping with uncertainty is a challenge that has increasingly attracted interest, most notably in the areas of self-adaptive systems (e.g., [Ch09, RJC12]), and in cyber-physical systems (c.f. [Zh16]). There are also some theories distinguishing fundamental kinds of uncertainties (e.g., [Wa03]). Several approaches towards formal modelling of uncertainties are available, the most prominent ones being probabilistic approaches, which can express the likelihood that some event will happen or some property holds, and fuzzy-logic approaches, where uncertainty is expressed as a degree of truth (see [RAC04]). For example, Markov chains are widely used in engineering to model state transitions triggered with a certain probability. Some authors focus on documenting contextual uncertainty in goal models. For instance, Cheng et al. [Ch09] use the obstacle notation to model uncertainties in KAOS models. The authors of [WS10] use the claim concept to document uncertain assumptions, their interrelations, as well as the effect on system goals. Other approaches deal with incorporating uncertainty in natural-language requirements [Wh09], incorporating the fuzziness of real-world data in UML Class Diagrams [MZY16], or annotating missing information in partial models [FaSa13].

Zhang et al. [Zh16] present a comprehensive conceptual model to structure different kinds of uncertainties. Their approach emphasizes the subjective notion of uncertainty, i.e., uncertainties are always related to statements about beliefs of some agent. In their taxonomy the core concepts to further characterize an uncertainty can be summarized as follows: Uncertainty has a specific type that captures about *what* there exists a lack of confidence, e.g., regarding the content of some other entity's subjective belief

statements. Uncertainty becomes visible in some position or artifact (i.e., *locality*), and occurs according to a specific *pattern* (e.g., it may be persistent, or occurring sporadically). Uncertainty also has an *effect* and a *risk* associated with it. Furthermore, uncertainty can be *measured*, e.g., using probabilistic approaches (see above). The detailed uncertainty-related concepts are beyond the scope of this paper; for details please refer to [Zh16]. In [Za17a], some uncertainty concepts are applied to enhance textual use case specifications with uncertainty information. There is also a UML profile available that comprises stereotypes reflecting the different uncertainty concepts [Zh17b]. However, to the best of our knowledge, there is no approach that explicitly addresses the need for tracing uncertainty information through several engineering artifacts, such as context models, or requirements and architecture models.

3 Uncertainties in the Vehicle Platooning Example

This section introduces the join maneuver as the specific scenario for illustrating our approach. Based on this scenario, it is shown how uncertainty can be characterized.

3.1 The Join Maneuver in Vehicle Platooning

We use the join protocol described in [Se14], and information about the context of a conventional ACC from [Da16] as a reference for identifying uncertainties as well as for creating the models presented in Section 4. The joining maneuver is based on the assumption that all the vehicles potentially engaging in a platoon are equipped with a CACC that is able to manage, among others, the joining procedure. A platoon is always led by one vehicle, which is denoted the platoon leader. The specific focus is on the case where a vehicle joins in the middle of the platoon. There are several possible variations as well as additional joining scenarios under certain conditions; these are, however, not considered here in order to keep the models simple. The basic joining maneuver consists of the following essential steps: First, the joining vehicle sends a join request after discovering the platoon. Then, the platoon leader computes a potential joining position in the platoon, and decides whether to accept or refuse the request. In case the request is accepted, the leader replies a positive acknowledge, as well as information about the position where to join in the platoon. The joining vehicle approaches the joining position, and notifies the platoon leader accordingly. Next, the platoon leader sends a command to the vehicle, which is supposed to open a gap for the joining vehicle. In order to open a gap, the vehicle that received the command for opening a gap notifies all the following vehicles that it temporarily forms a second platoon. Eventually this temporary platoon reduces its speed to open a gap, and notifies the leader that a gap has been formed. The leader then coordinates the joining vehicle to change the lane into the gap. When done, the successfully joined vehicle notifies the leader so that the second platoon can be commanded to close the gap and thereby integrate the temporary platoon back into the original one. More details on the join maneuver can be found in [Se14].

3.2 Identifying Uncertainty Information

Since the context of a CACC system is highly dynamic, and not only includes collaboration-related aspects (e.g., communication among different CACC embedded in different vehicles), but also the physical surroundings (e.g., road and weather conditions), it is characterized by several uncertainties that may occur during operation. In the joining maneuver, one important uncertainty concerns the potentially ambiguous information (cf. [RJC12]) from different sources, regarding a potential gap for joining the platoon. While the command and joining position received from the platoon leader indicates a free gap, the data obtained from a camera sensor might suggest that there actually is an object in the gap. For instance, a vehicle not part of the platoon may have joined in the gap (cf. [Se14, Ax17]). In this case there is a severe uncertainty whether the gap is open or not. This context uncertainty prevents the CACC to operate properly.

In Tab. 1, we describe this context uncertainty in detail. The uncertainty concepts are borrowed from [Zh16] (see also Section 2.2), while the way of documenting the information is inspired by the approach to specifying uncertainties in textual use case descriptions described in [Zh17a]. Please note that, in order to keep things simple, we do not use all the concepts related to uncertainty defined in [Zh16, Zh17b].

Uncertainty 1: Ambiguous information from different sources	
Kind	Environment, Content
Indeterminacy Source	Intentions of collaborators, Sensor imprecision, Communication failures
Locality	Gap information, Object recognition
Effect	Unable to change lane
Risk	High
Occurrence Pattern	Aperiodic

Tab. 1: Documentation of Uncertainty Information

The kind of uncertainty denotes that the uncertainty represents lack of confidence regarding the current state of the *environment* and the *content* of messages exchanged in a platoon. The locality, i.e., the artifact where the uncertainty becomes visible, is the *gap information* distributed by the platoon leader, and the *data coming from the vehicles own sensors*, which need to be compared by the CACC at runtime. The indeterminacy source that causes the ambiguity is rooted in unclear intentions of collaborators as well as the inherent imperfection of sensor and communication technology (cf. [RJC12]). The ambiguity, in turn, prevents the vehicle from safely *changing the lane*, if not properly handled. The risk associated with this uncertainty is *high* since some inappropriate action taken by the CACC in the presence of this uncertainty may pose a safety hazard. The occurrence pattern of the uncertainty is characterized as *aperiodic* as the uncertainty can only occur when a vehicle intends to join the platoon, i.e., it is event-triggered.

4 Orthogonal Uncertainty Modelling

In order to relate the uncertainties defined in Section 3.2 to context models of the CACC system, we suggest modelling the uncertainty information in a dedicated diagram that is orthogonal to the operational context models of the SPES XT Context Modeling Framework [Da16]. Since the scope of this paper is to motivate the use of orthogonal modelling techniques for documenting uncertainties, we do not propose a new modelling language. Instead, we use SysML Requirements Diagrams to depict uncertainty. In the following, we will introduce diagrams from the three context perspectives, and illustrate traceability relationships between them and the orthogonal uncertainty model.

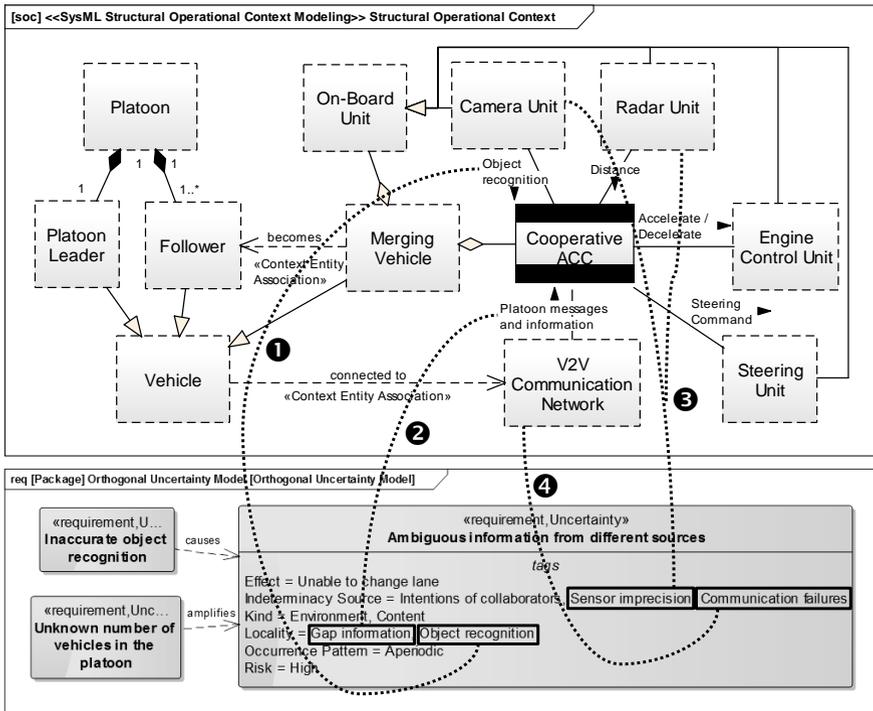


Fig. 1: Uncertainty Information in Structural Operational Context Models

The structural model of the CACC’s operational context shown in Fig. 1 depicts the CACC as the system under development (the context subject), and its relations to other control units (such as a *Camera Unit*, or the *Engine Control Unit*) onboard the vehicle that wants to join an existing platoon. The model also gives a structural perspective on the formation of a platoon, which consists of one leader and several followers.

The orthogonal uncertainty model comprising the ambiguity described in Section 3.2 is depicted at the bottom of Fig. 1. The relationships between the uncertainty concepts and

elements of the context model are depicted using dotted lines. This allows tracing uncertainty-related information to the model elements that are affected by or subject to uncertainties. For example, the two specific aspects of the uncertainty locality manifest in the interfaces between the CACC and the *Camera Unit*, and the *V2V Communication Network*, respectively (① and ②). Sensor imprecision as a reason for potential mismatches between the two information sources (i.e., the ambiguity) is related to the *Camera Unit* and the *Radar Unit* (③), while *Communication failures* relate to the *V2V Communication Network* used for realizing the exchange of messages in the platoon (④). The uncertainty model also allows specifying relationships between uncertainties (cf. [Zh17a]). For instance, another uncertainty “*Inaccurate object recognition*”, which is not specified in detail to ease readability, is typically one cause of ambiguity related to imperfect sensor technology. As another example, an *unknown number of vehicles in the platoon* can be seen as a factor that increases or amplifies the ambiguity uncertainty.

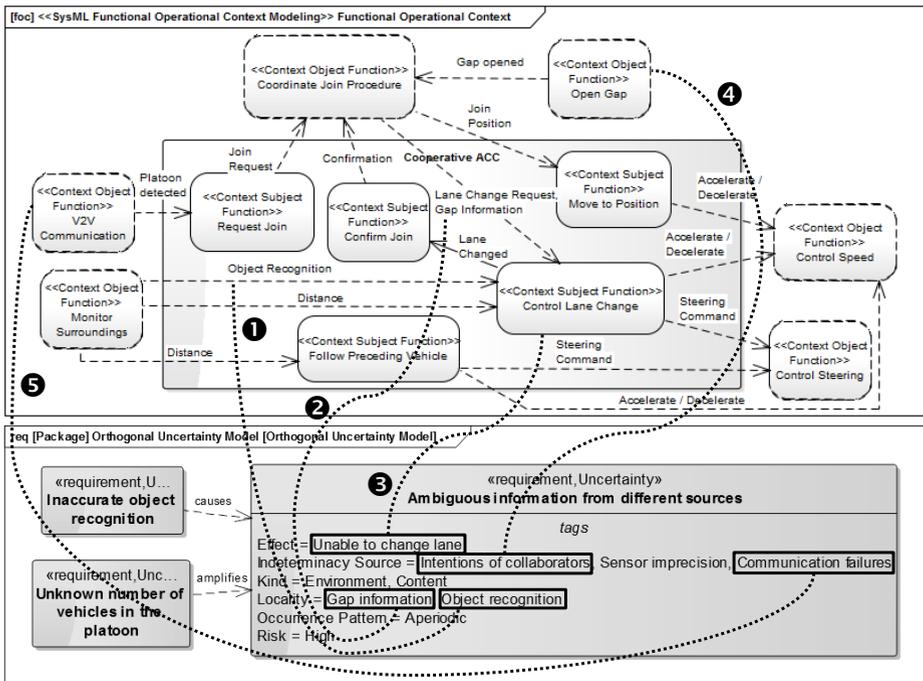


Fig. 2: Uncertainty Information in Functional Operational Context Models

Fig. 2 shows a functional operational context model and its relations to the same orthogonal uncertainty model. Here, relations ① and ② depict the same aspect as the respective relations in the structural operational context model (see Fig. 1), however, from a functional perspective that highlights the input/output data relations between the CACC’s functions and context object functions (depicted using dashed lines). The same applies to *communication failures* (⑤). Furthermore, the effect of the uncertainty, i.e.,

the inability to change the lane, will impact the CACC function “Control Lane Change” (❸). Unknown intentions of collaborators, which constitute another indeterminacy source, could manifest in a delayed gap opening (❹). Unclear intentions are specifically important if CACC implementations by different vendors are supported.

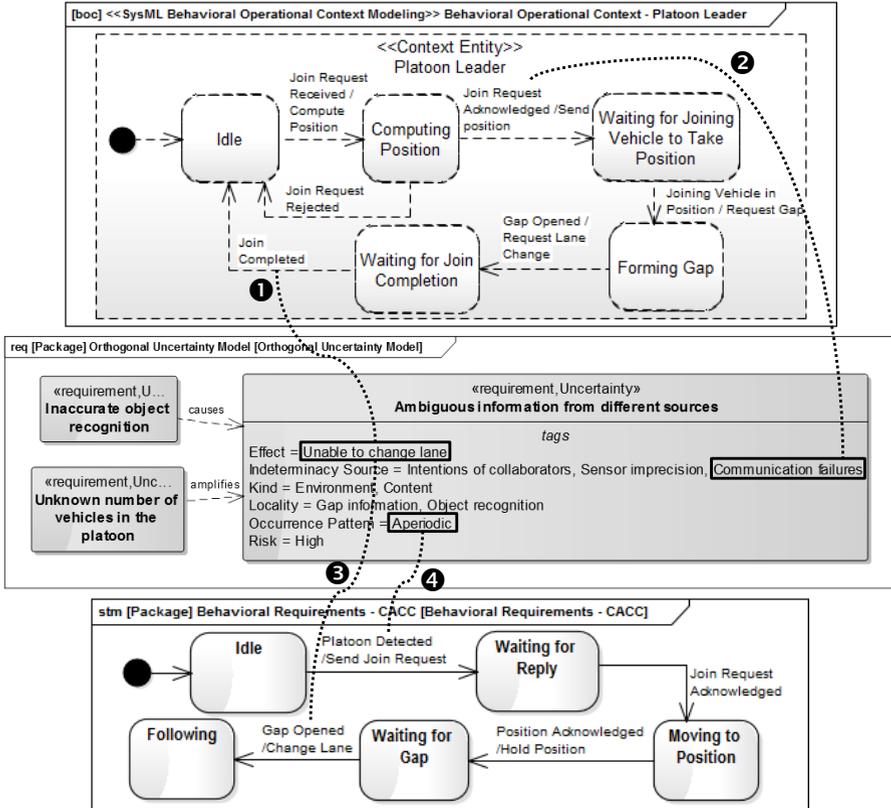


Fig. 3: Uncertainty Information in Behavioral Operational Context Models (based on [Se14])

The state machine diagrams in Fig. 3, which closely reflect the protocol automata presented in [Se14], take a behavioral view on the case example. The upper diagram presents the behavior of the context object “Platoon leader”, while the diagram at the bottom presents a behavioral requirements model describing the CACC. The inability to complete the join (i.e., to change the lane) in the presence of uncertainty may cause a deadlock in the behavior of the platoon leader (❶), which will require certain handling mechanisms, such as timeouts (see [Se14]). It also prevents the action “Change lane” to be carried out by the CACC (❸). Relation ❷ denotes that communication failures caused by imperfect technology may also negatively influence the exchange of request and acknowledgement messages. Relation ❹ specifies the occurrence trigger of the uncertainty, which is the detection of a platoon, accompanied by a join request.

Fig. 3 highlights an important aspect of our orthogonal uncertainty modeling approach, i.e., the fact that uncertainty does not only reflect in context models, but also affects requirements models (see relations ⑤ and ④). Similarly, architectural artifacts or source code might be associated with the same uncertainty using our orthogonal model. This also allows specifying additional requirements for mitigating potential hazardous effects of uncertainties, which is especially relevant for safety-critical systems.

5 Conclusion and Outlook

In this paper, we investigated the use of orthogonal models for documenting uncertainty. To do so, we exemplarily modelled orthogonal uncertainty information related to the context of an exemplary CACC system, with a particular focus on the procedure to coordinate the joining of a vehicle into a platoon. The information we capture in the orthogonal uncertainty models is based on state-of-the-art concepts relevant for uncertainty. We established relations between the orthogonal uncertainty model and context models from three different perspectives. The results show that orthogonal uncertainty modeling can reduce redundancies, and enhances the traceability of uncertainty information. The latter is essential for properly considering and mitigating potential runtime uncertainties, so that the system will satisfy its requirements in an uncertain operational context. Orthogonal uncertainty modeling is seen as advantageous in early development phases, and also has the potential to support safety analyses.

In future works, we aim at establishing an orthogonal uncertainty modeling language, including a suitable notation, as an extension of the SPES XT Context Modeling Framework. This extension will be based on a sound theoretical basis, i.e., an ontology that captures the specific characteristics of collaborative embedded systems as well as related uncertainties. To this end, we will investigate the use of more sophisticated uncertainty modeling concepts, e.g., from the comprehensive taxonomy presented in [Zh16]. The semantics of the relationships between the uncertainty model and context models also needs to be further elaborated. Moreover, methodological aspects, i.e., guidelines for coping with uncertainties, as well as the integration of orthogonal uncertainty models into hazard analyses and software process models, will be analyzed.

Acknowledgements. This research has been partly funded by the German Federal Ministry of Education and Research under grant no. 01IS16043V.

References

- [Ax17] Axelsson, J.: Safety in Vehicle Platooning: A Systematic Literature Review. In: IEEE Trans. on Intelligent Transportation Systems, Vol. 18 (5), pp. 1033–1045. IEEE, 2017.
- [Ba03] Bachmann, F. et al.: A Meta-model for Representing Variability in Product Family Development. In: PFE 2003, LNCS 3014, pp. 66–80. Springer, 2004.

- [Ch09] Cheng, B.H.C. et al.: A Goal-Based Modeling Approach to Develop Requirements of an Adaptive System with Environmental Uncertainty. In: MODELS 2009, LNCS 5795, pp. 468–483. Springer, 2009.
- [Da16] Daun, M. et al.: SPES XT Context Modeling Framework. In: Advanced Model-Based Engineering of Embedded Systems – Extensions of the SPES 2020 Methodology, pp. 43–57. Springer, 2016.
- [FaSa13] Famelis, M., Santosa, S.: MAV-Vis: A Notation for Model Uncertainty. In: 5th Int. Workshop on Modeling in Software Engineering (MiSE), pp. 7–12. IEEE, 2013.
- [Fe16] U.S. Federal Highway Administration: Cooperative Adaptive Cruise Control - Taking Cruise Control to the Next Level. Fact Sheet FHWA-HRT-16-044, Feb. 2016.
- [JD91] Jordan, K.A., Davis, A.M.: Requirements Engineering Metamodel - An Integrated View of Requirements. In: COMPSAC 1991, pp. 472–478. IEEE, 1991.
- [Ji16] Jia, D. et al.: A Survey on Platoon-Based Vehicular Cyber-Physical Systems. In: IEEE Comm. Surveys & Tutorials, Vol. 18, No. 1, 2016, pp. 263–284. IEEE, 2016.
- [MZY16] Ma, Z.M., Zhang, F., Yan, L.: Fuzzy Information Modeling in UML Class Diagram and Relational Database Models. In: Applied Soft Computing 11, pp. 4236–4245. Elsevier, 2016.
- [RAC04] Ranganathan, A., Al-Muhtadi, J., Campbell, R.H.: Reasoning about Uncertain Contexts in Pervasive Computing Environments. In: IEEE Pervasive Computing, Vol. 3 (2), pp. 62–70. IEEE, 2004.
- [RJC12] Ramirez, A.J., Jensen, A.C., Cheng, B.H.C.: A Taxonomy of Uncertainty for Dynamically Adaptive Systems. In: SEAMS 2012, pp. 99–108. IEEE, 2012.
- [Se14] Segata, M. et al.: Supporting Platooning Maneuvers through IVC: An Initial Protocol Analysis for the Join Maneuver. In: WONS 2014, pp. 130–137. IEEE, 2014.
- [Sc16] Schaefer, I. et al.: Variant Management and Reuse. In: Advanced Model-Based Engineering of Embedded Systems – Extensions of the SPES 2020 Methodology, pp. 197–222. Springer, 2016.
- [Wa03] Walker, W.E. et al.: Defining Uncertainty: A Conceptual Basis for Uncertainty Management in Model-Based Decision Support. In: Integrated Assessment, Vol. 4 (1), pp. 5–17. Swets & Zeitlinger, 2003.
- [WS10] Welsh, K., Sawyer, P.: Understanding the Scope of Uncertainty in Dynamically Adaptive Systems. In: REFSQ 2010, LNCS 6182, pp. 2–16. Springer, 2010.
- [Wh09] Whittle, J. et al.: RELAX: Incorporating Uncertainty into the Specification of Self-Adaptive Systems. In: 17th Int. RE Conference, pp. 79–88. IEEE, 2009.
- [Zh16] Zhang, M. et al.: Understanding Uncertainty in Cyber-Physical Systems: A Conceptual Model. In: ECMFA 2016, LNCS 9764, pp. 247–264. Springer, 2016.
- [Zh17a] Zhang, M. et al.: Specifying Uncertainty in Use Case Models in Industrial Settings. Simula Research Laboratory, Technical Report 2016-15, February 2017.
- [Zh17b] Zhang, M. et al.: Uncertainty-Wise Cyber-Physical System Test Modeling. In: Software & Systems Modeling, Published Online. Springer, 2017.