Design of a Knowledge-Base Strategy for Capability-Aware Treatment of Uncertainties of Automated Driving Systems

DeJiu Chen[1], Kenneth Östberg[2],
Matthias Becker[1], Håkan Sivencrona[3], Fredrik Warg[2]

[1]KTH Royal Institute of Technology, SE-10044 Stockholm, Sweden
{chendj, mabecker}@kth.se
[2]RISE - Research Institutes of Sweden Box 857, SE-50115 Borås, Sweden
{kenneth.ostberg, fredrik.warg}@ri.se
[3]Zenuity AB, Lindholmspiren 2, SE-41756 Göteborg, Sweden
hakan.sivencrona@zenuity.com

Published in:

Proceedings of SAFECOMP 2018 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Västerås, Sweden, September 18, 2018, pp 446-457, Springer International Publishing.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-99229-7_38

Design of a Knowledge-Base Strategy for Capability-Aware Treatment of Uncertainties of Automated Driving Systems

DeJiu Chen[1] [0000-0001-7048-0108](✉), Kenneth Östberg[2] [0000-0001-7672-6826], Matthias Becker[1] [0000-0002-1276-3609], Håkan Sivencrona[3] [0000-0002-5371-5048], Fredrik Warg[2] [0000-0003-4069-6252]

[1] KTH Royal Institute of Technology, SE-10044 Stockholm, Sweden
{chendj, mabecker}@kth.se

[2] RISE - Research Institutes of Sweden Box 857, SE-50115 Borås, Sweden
{kenneth.ostberg, fredrik.warg}@ri.se

[3]Zenuity AB, Lindholmspiren 2, SE-41756 Göteborg, Sweden
hakan.sivencrona@zenuity.com

Abstract. Automated Driving Systems (ADS) represent a key technological advancement in the area of Cyber-physical systems (CPS) and Embedded Control Systems (ECS) with the aim of promoting traffic safety and environmental sustainability. The operation of ADS however exhibits several uncertainties that if improperly treated in development and operation would lead to safety and performance related problems. This paper presents the design of a knowledge-base (KB) strategy for a systematic treatment of such uncertainties and their system-wide implications on design-space and state-space. In the context of this approach, we use the term Knowledge-Base (KB) to refer to the model that stipulates the fundamental facts of a CPS in regard to the overall system operational states, action sequences, as well as the related costs or constraint factors. The model constitutes a formal basis for describing, communicating and inferring particular operational truths as well as the belief and knowledge representing the awareness or comprehension of such truths. For the reasoning of ADS behaviors and safety risks, each system operational state is explicitly formulated as a conjunction of environmental state and some collective states showing the ADS capabilities for perception, control and actuations. Uncertainty Models (UM) are associated as attributes to such state definitions for describing and quantifying the corresponding belief or knowledge status due to the presences of evidences about system performance and deficiencies, etc. On a broader perspective, the approach is part of our research on bridging the gaps among intelligent functions, system capability and dependability for mission-&safety-critical CPS, through a combination of development- and run-time measures.

Keywords: Automated Driving System (ADS), Cyber-Physical System (CPS), Embedded Control System (ECS), Knowledge-Base (KB), Uncertainty Models (UM), Safety.

1 Introduction

Cyber-Physical Systems (CPS) and the underlying Embedded Control Systems (ECS) are the key enabling technologies behind autonomous vehicles, smart production systems, medical equipment and many other intelligent products. Many of these products are inherently safety- or mission-critical as the physical aspect, represented by the dynamics or energy flows under control, implies that a system failure could lead to unreasonable risks. This calls for, on the one hand, advanced formalisms, methods and tools for verification and validation, correct-by-construction and fault avoidance; and on the other hand, the deployment of specific safety functions and technologies for fault tolerance and fault treatment. Currently, the cyber aspect, characterized by information treatment and control logics for the operation perception, control of behaviors [1], is on an increasing degree based on Artificial Intelligence (AI), particularly Machine Learning (ML) and Artificial Neural Networks (ANN). The implementation relies on the provision of embedded resources for the sensing, communication, computation and actuation with an increasing degree of heterogeneity (e.g. a mixture of generic microcontrollers and specific AI accelerators). In this paper, we refer to the actual ability of a CPS to conduct specific tasks or actions regarding the system operations as *CPS Capability*.

Automated Driving System (ADS) [2] is a type of advanced CPS that can support self-governed driving behaviors in complex operational environments (e.g. public streets), with many potential economic, social and environmental benefits. However, being inherently safety critical, ADS is currently facing some fundamental challenges in risk management that necessitate a holistic strategy for fault avoidance, fault tolerance and fault treatment. One key factor behind the challenges is that the operation of ADS exhibits several types of uncertainty that make conventional quality assurance through formal verification and validation inadequate. In particular, in regard to the operational environment of ADS, there is an inherent uncertainty due to the emergent properties of traffic environment where heterogeneous traffic objects are composed randomly. Meanwhile, uncertainty can also show up in the perception of operational situations due to the design and performance issues of sensors (e.g. radar and camera) and services, such as delimited knowledge about the environment, unoptimized sensor position in vehicle, insufficient communication bandwidth. In general, a system could exhibit nondeterministic behaviors due to emergent behaviors and faults in the implementation because of partial specification, data inconsistency, imperfect synchronization and hardware reliability, etc. For ADS with AI functions, nondeterministic behaviors can also arise due to the gap between training set and real operational conditions, the inherent stochasticity in algorithms, and the complex interplay with actual CPS capability regarding perception, communication, computation and actuation.

This paper presents the design of a Knowledge-Base (KB) strategy [3] for a systematic treatment of such uncertainties of CPS and their system-wide implications on design-space and state-space. The approach is part of our research on bridging the gaps among intelligent functions, system capability and dependability for mission-&safety-

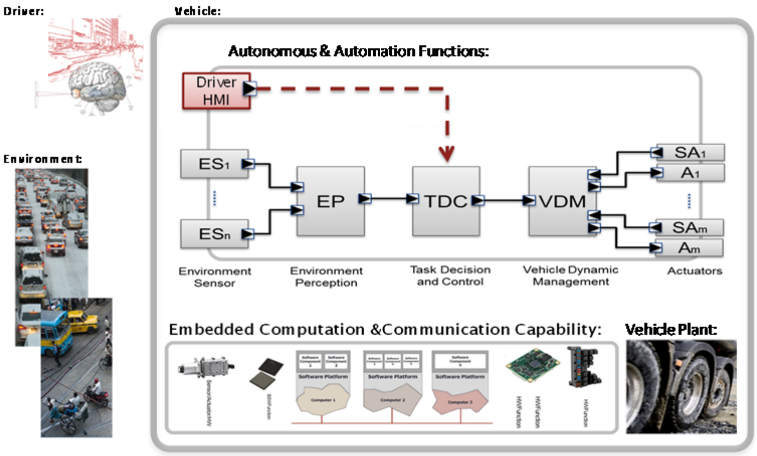

Driver:
Vehicle:
Autonomous & Automation Functions:
Driver HMI
Environment:
ES1
ESn
EP
TDC
VDM
SA1
A1
SAm
Am
Environment Sensor
Environment Perception
Task Decision and Control
Vehicle Dynamic Management
Actuators
Embedded Computation &Communication Capability:
Vehicle Plant:

Fig. 1. A schematic overview of key aspects of ADS.

critical CPS, through a combination of development-time and run-time measures [4]. In particular, the run-time measures are related to the design of embedded services for the awareness of operational situations and capabilities including the uncertainties, and then the assessment of operational risks. The development-time measures are centered on the enrichment of existing system ontology and frameworks (e.g. EAST-ADL [5]), by addressing the composition of heterogeneous functions and components, including those based on AI technologies.

The rest of this chapter is structured into the following sections: Section 2 provides an overview of related concepts in regard to the ADS architecture and uncertainty. Section 3 presents the proposed KB strategy, including the supported key modeling concepts and uncertainty descriptions. An overview of related technologies is given in Section 4. Section 5 summarizes our conclusions.

2 ADS System and Uncertainty

Fig. 1 provides a schematic overview of ADS where the top-level system aspects are given by *Environment*, *Driver* and *Vehicle*. This conceptual view refines the generic architectural pattern introduced in [6]. While the *Environment* and *Driver* together constitute the operational context of ADS, the *Vehicle* refers to the product content of ADS given by a mixed composition of cyber and physical units. In particular, the *Environment* denotes the context in which ADS operates. It is defined by some external static and dynamic situations, including the road situation (e.g. lanes and road geometry) and the traffic situation (e.g. adjacent vehicles and pedestrians). The *Driver* refers to the person in the vehicle who interacts with the ADS [7]. The *Vehicle*, representing the product content of ADS, corresponds to a complete car or truck system. It is composed of some functional contents, shown as *Autonomous&Automation Func-*

tions in Fig. 1, and some technical contents, shown as *Embedded Computation and Communication Capability* and *Vehicle Plant* in Fig. 1. Such contents collectively determine *ADS Capability*, referring to the ability of *ADS* to conduct specific driving tasks. *ADS Capability* is a specialized form of *CPS Capability* mentioned previously.

The functional contents consist of the control logics, organized into a decision hierarchy [8, 9] as shown in Fig. 2**.** The lower two layers, *Operation Control* and *Operation Decision*, implements a supervisory control strategy for the dynamics of vehicle plant. We refer to the control functions of these two layers collectively as *VDM* (Vehicle Dynamic Management). For automated driving, the tactic decision functions in the layer above decide the tactic actions, relating to the choice of target points on the road, as well as the preferred sequence of moves to reach the target points, such as accelerating and veering. The top layer contains strategic decisions for achieving a mission (e.g., the choice of routes from city A to city B). We refer to the functions in the tactic and strategic layers collectively as *TDC* (Task Decision and Control), shown in Fig. 1. Along with the decision hierarchy, there are also functions for operation perception. We refer the functions for the sensing of ADS *Environment* (defined previously) as *Environment Sensor* (*ES*); and the functions for the transformation of monitored environment data into a consolidated world-model as *Environment Perception (EP)*, also shown in Fig. 1. We refer the plant sensing and actuation functions for VDM as *SA* (Sensor for Actuation of plant) and *A* (Actuator of plant), respectively.

Across the decision hierarchy from VDM to TDC, an increasing degree of autonomy can be observed. Normally, VDM functions in the lower two operational layers are dominated by reactive feedback control. The design relies on *prior* knowledge about

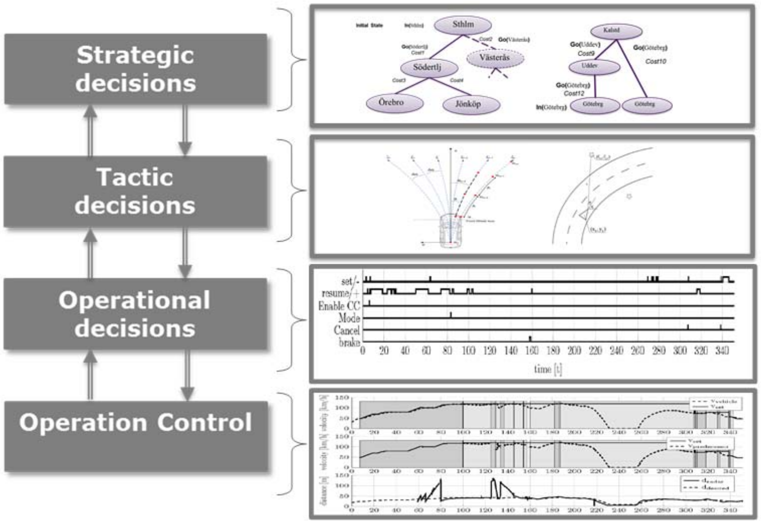


Fig. 2. The hierarchy of decision and control in ADS and example contents.

the plant (*Vehicle Plant*) in the form of models for the discrete and continuous dynamics, with the goal of ensuring highly deterministic behaviors. That is, given a particular sequence of inputs, such a function will always produce the same sequence of outputs while passing through the same sequence of states. In general, such a functional determinism, together with effective use of well-formulated *prior* knowledge, facilitates the verification and validation. For example, test generation using FSM or model-checking is a well-known approach [10]. Compared to VDM, such a prior knowledge centric and determinism based approach would be far from being sufficient for the design of TDC as well as the ES and EP functions, due to the associated inherent functional uncertainties of ADS. The design has to provide sufficient flexibility of such functions to allow for appropriate compensation of partial or incorrect *prior* knowledge, as well as for effective exploration and management of possible outcomes of actions. This implies not only a higher level of autonomy, but also an increased non-determinism that makes the verification and validation challenging.

For ADS, we distinguish two types of inherent functional uncertainty as for other autonomous systems [11, 12]: 1. *Aleatory Uncertainty*; and 2. *Epistemic Uncertainty*. The aleatory uncertainty is related to the contextual complexity of ADS. It is caused by the emergent properties from random interactions of heterogeneous traffic objects in the physical operational environment (i.e. *Environment*), under different conditions of weather, road, and physical locations. The presence of high aleatory uncertainty makes it difficult for the TDC to predict the dynamic trajectories of traffic objects, and thereby the upcoming traffic situations and the effects of its actions on the environment. Aleatory uncertainty is also known as statistical uncertainty and is representative of unknowns that differ in each particular operation scenario. The epistemic uncertainty is related to the design and performance of perception functions (i.e. *ES* and *EP*). It is caused by the effects of probabilistic algorithms, restricted observability, physical limitation, hidden variables, under-specification or semantic ignorance when monitoring and processing the environment situations. Epistemic uncertainty is also known as systematic uncertainty. In CPS, these functional uncertainties are further affected by the actual capability of system providing the implementation (i.e. *CPS Capability*). For ADS, any anomaly regarding the assumed *ADS Capability*, i.e. the faults or errors exhibited by the computation and communication resources and vehicle plant, could result in additional nondeterminism of the corresponding control functions. One related constraint is *functional safety*, referring to the freedom from unacceptable risk of hazards as specified by ISO 26262. It requires a set of measures for fault avoidance, fault tolerance and fault treatment. One key aspect is the support for a formal specification of such uncertainties and thereby for a qualified anomaly detection and risk mitigation. We present in the follow-up sections our strategy to an enriched ADS description, emphasizing the knowledge and uncertainty modeling.

3 Design of Knowledge-Base (KB) Strategy

The strategy introduced here aims to allow the above-mentioned types of uncertainty

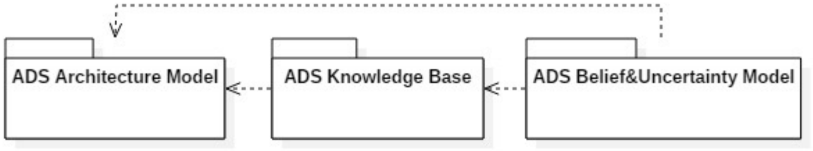


Fig. 3. The modeling packages and their dependencies.

to be treated systematically together with a well-defined ADS system ontology. The modeling packages, shown in Fig. 3, include *ADS Architecture Model*, *ADS Knowledge Base*, and *ADS Belief&Uncertainty Model*. The key aspect is related to an integrated formal specification of system commitments for automated driving in various operational environments, and then the exploitation of such information for the systems engineering as well as the design of embedded services for anomaly detection and risk mitigation. As the base technologies, the following two existing modeling frameworks are adopted and extended for ADS: 1. EAST-ADL (Electronics Architecture and Software Technology - Architecture Description Language) [5] for the development of *ADS Architecture Model*; and 2. U-Model (Uncertainty conceptual model for CPS) [13] for the development of *ADS Belief&Uncertainty Model*. The *ADS Knowledge Base* presented below provides the support for a formal specification of the operational properties across the ADS *Environment*, *Driver* and *Vehicle*, effectively merging any semantic gaps between the system description and beliefs.

The package diagram in Fig. 3 shows dependencies between the modeling packages. The *ADS Architecture Model* contains all the functional and technical design commitments regarding the *Environment*, *Driver* and *Vehicle*. The *ADS Knowledge Base* depends on the *ADS Architecture Model* as the corresponding description of operational properties relies on the design. These operational properties in turn constitute the basis for the uncertainty and belief statements by the *ADS Belief&Uncertainty Model*. Such statements can then be used to refine requirements, design solutions, verification and validation cases in the *ADS Architecture Model* when necessary.

3.1 ADS Knowledge-Base (KB)

Here, the term ADS Knowledge-Base (KB) refers to the models that stipulate the fundamental facts in regard to the overall system operational states, action sequences, as well as the related costs or constraint factors. The model constitutes a formal basis for describing and inferring particular operational truths as well as the belief and knowledge representing the awareness or comprehension of such truths.

For the reasoning of ADS behaviors and operational risks, we formulate each system operational state as a conjunctive state:

$$S_{\mathrm{k}} = \left(S_{Env_{\mathrm{k}}},\ S_{Dri_{\mathrm{k}}},\ S_{Veh_{\mathrm{k}}}\right); S_{\mathrm{k}} \in \mathrm{S}, \mathrm{S} \subseteq S_{Env} \times S_{Dri} \times S_{Veh} \tag{1}$$

Here, the state variable S_{k} refers to the operational condition of an ADS system at a particular time point *k*. The discretized behavior description assumes a discretization of time *t* with $t \in \mathbb{R}_+$ (i.e. the set of non-negative real numbers) and a fixed time interval dt > 0. Every time of *k* corresponds to a discrete time step with $\text{k} = \lfloor t/dt \rfloor$ and $k \in \mathbb{Z}_+$ (i.e. the set of non-negative integers). The state S_k is given by the conjunction (logical AND) of S_{Env_k}, S_{Dri_k}, S_{Veh_k}, referring to the respective operational conditions of *Environment*, *Driver* and *Vehicle* at the same time point. These state variables collectively define how the ADS react to the actions by the environment, the driver and the vehicle. We use *S* to denote the overall state space of an ADS, i.e. all possible state conditions, which is the subset of all possible combinations of operational conditions of environment, driver and vehicle ($\text{i.e.}\, S_{Env} \times S_{Dri} \times S_{Veh}$). A particular *behavior* of ADS is then a sequence of chosen operational conditions:

$$S^H = (S_0\ , S_1\ , \dots, S_{\text{H}-1}) \tag{2}$$

where the variable *H* denotes the length of this sequence in terms of a time horizon value given by max *k*. Furthermore, we define complete *trajectory* or operational trace of ADS as:

$$\xi^H = \big((S_0\ , A_0\ ), (S_1\ , A_1\ ), \dots, (S_{\text{H}-1}, A_{\text{H}-1})\big);\ \xi^H \in \Xi\ ,\ \Xi \subseteq 2^{S\times A} \tag{3}$$

Here, we use Ξ to denote the overall trajectories of an ADS, which is the subset of all possible combinations of all state and action pairs ($2^{S\times A}$) with *A* for all possible actions. Each segment of the trajectory consists of a pair of operational condition S_k and operational action A_k at the time instance *k*. The operational action A_k can be given by an action of environmental object (e.g. braking of preceding vehicle), a driver action (e.g. starting ADS), an action of ADS vehicle (e.g. steering to the right), or a combination of multiple actions at the same time instance. For an ADS, the choice of its action at any given instant is given by its *Autonomous&Automation Functions* according to the current as well as the past operation perceptions.

ADS operational performance is measured by the cost function associated to a trajectory $J(\xi^H)$. Accordingly, each requirement or constraint on the system is a proposition φ_i that can be satisfied or not satisfied:

$$\xi^H \vDash \varphi_i \ \text{ or } \ J(\xi^H) \vDash \varphi_j;\ \ \varphi_i\ ,\ \varphi_j \in \Phi \tag{4}$$

The variable Φ denotes all requirements. Fig. 4 illustrates the overall state space (S) of an ADS system and some of two possible trajectories (ξ_1^{H1}, ξ_2^{H2}) and (ξ_1^{H1}, ξ_3^{H3}). The first one represents a fail-safe scenario, where hazards are successfully detected at S_3 with ξ_2^{H2} as the safety measure for returning to the safe state S_5 . The second one represents crash scenario with final state given by S_6 . In system development, safety requirements are used to specify such trajectories.

In ADS, the VDM functions implement the driving actions selected by the associated TDC functions (see also Fig. 1). We have used $S_{Veh_\text{k}}$to denote the vehicle state

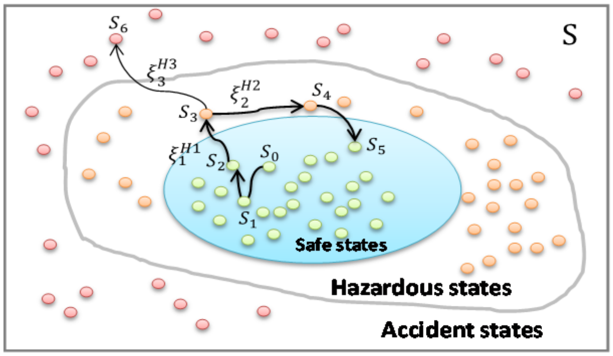


Fig. 4. Overall state space (S) of an ADS and some of the possible trajectories. Each point represents a state S_{k}, which is by a conjunction (logical AND) of states of the environment ($\boldsymbol{S_{Env_{\mathrm{k}}}}$), the driver ($\boldsymbol{S_{Dri_{\mathrm{k}}}}$), and the vehicle ($\boldsymbol{S_{Veh_{\mathrm{k}}}}$).

S_{Veh} at time k. The state is defined as follows:

$$S_{Veh} = \left[P_x \ , P_y \ , \theta, \nu, \omega, a\ \right]^T \qquad (5)$$

We let $(P_x\ ,\ P_y) \in \mathbb{R}^2$ be the vehicle position relative to some fixed coordinate frame and $\theta \in [-\pi,\ \pi]$ be the yaw angle of the vehicle, and ω be the angular speed. The vehicle moves forward with speed $\nu \in [0,\ \nu_{\mathrm{Max}}]$, where ν_{Max} is the maximum speed. The acceleration is given by a. The basic motion of the vehicle is then given by:

$$\begin{array}{ll} \text{Continuous time:} & \dot{S}_{Veh} = f\left(S_{Veh},\ u\right); \\ \text{Discrete time:} & S_{Veh_{\mathrm{k+1}}} = f_d\left(S_{Veh_{\mathrm{k}}}, u_{\mathrm{k}}\right) \end{array} \qquad (6)$$

Here, u denotes the request of tactic action from the TDC to the VDM as the choice of θ and ν , i.e. $u = [\ \theta, \mathrm{v}]^T$. For example, at a particular time instant, the motion of the vehicle is determined by: $\dot{P}_x = \nu \cos\theta;\ \dot{P}_y = \nu \sin\theta;$ and $\dot{\theta} = \omega$. For a tactic decision *Turn_left*, we have $\dot{P}_x = \nu \cos\theta;\ \dot{P}_y = \nu \sin\theta;\ \dot{\theta} = \pi;$ For a tactic decision *Stop*, we have $P_x \overset{\cdot}{=} 0; P_y = \nu;$ and $\dot{\theta} = 0$. Formally, with 2^A possible choices of actions A by TDC, we have:

$$S_{Veh_{\mathrm{k+1}}} \subseteq (\mathbb{R}^6 \times 2^A) \qquad (5)$$

3.2 Integration of Uncertainty Modeling and System Description

For ADS, the introduction of Uncertainty Models (UM) aims to constitute a formal basis for describing and inferring particular operational truths on the basis of the Knowledge-Base (KB). By describing and quantifying the corresponding belief or knowledge status, such models describe the degree of awareness or comprehension of some truths. The models can be used by system developers for the reasoning of functional and technical commitments at design-time or by embedded services for anomaly detection and risk mitigation at run-time. To this end, one key base technology adopted in our approach is the U-Model (Uncertainty conceptual model for CPS) [13], which aims to constitute a reference framework and standard. The core of the U-Model is a *Belief Model*, with the key meta-model concepts shown in Table 1.

Table 1. Key meta-model concepts of U-Model [13].

Key Concepts

- ❖ **BeliefAgent**: a physical entity owning one or more Beliefs about phenomena/notion.
- ❖ **Uncertainty**: a state whereby a BeliefAgent does not have full confidence in a Belief it holds.
- ❖ **Belief**: an implicit subjective explanation of some phenomena or notions by a BeliefAgent.
- ❖ **BeliefStatement:** a concrete and explicit specification of some Belief held by a BeliefAgent about possible phenomena or notions belonging to a given subject area.
- ❖ **Evidence:** an observation or a record of a real-world event occurrence, or, alternatively, the conclusion of some formalized chain of logical inference for determining the truthfulness.
- ❖ **EvidenceKnowledge**: an objective relationship between a BeliefStatement and relevant Evidence. It identifies if the corresponding BeliefAgent is aware of the appropriate Evidence.
- ❖ **Indeterminacy**: a situation whereby the full knowledge necessary to determine the required factual state of some phenomena/notions is unavailable.
- ❖ **IndeterminacySource**: factors that lead to Uncertainty.
- ❖ **IndeterminacyNature**: the specific kind of indeterminacy that can be InsufficientResolution MissingInfo, Nondeterminism, and a combination of more than one kinds of indeterminacy.
- ❖ **IndeterminacyKnowledge:** an objective relationship between an IndeterminacySource and the awareness that the BeliefAgent has of that source.
- ❖ **KnowledgeType**: an enumeration) of four values:

1. **KnownKnown** – BeliefAgent consciously aware of some relevant aspect.
2. **KnownUnknown** (Conscious Ignorance) – BeliefAgent aware of the ignorant of some aspect.
3. **UnknownKnown** (Tacit Knowledge) – BeliefAgent not explicitly aware of some relevant aspect that it may be able to exploit in some way.
4. **UnknownUnknown** (Meta Ignorance) – BeliefAgent unaware of some relevant aspect.

- ❖ **Measurement:** the optional quantification (or qualification) that specifies the degree of indeterminacy of the IndeterminacySource.
- ❖ Measure: an objective concept specifying method of measuring uncertainty.

We also adopt EAST-ADL [5] as the base technology for the description of the functional and technical commitments in the system design. The key concepts of integrating KB, U-Model, and system description in EAST-ADL for ADS are shown in Fig. 5. With the integration, *Evidence* in uncertainty description can have its semantics given by some associated operational behavior, operation trajectory, or operation performance, which is defined by KB (see Equation 2, 3, 4). Such operations are conducted by system objects given as *EAPrototype,* which is an abstract class in EAST-ADL for the target vehicle or its environment objects and operator. The factors that lead to uncertainty are declared by the associations from *IndeterminacySource* to the EAST-ADL abstract classes for system environment, system functions, hardware components, and system anomaly. With such associations, the sources of non-determinism or indeterminacy are systematically distinguished, including the aleatory uncertainty, epistemic uncertainty, and the deficiency of ADS capability, as defined in Section 2.

The *EAST-ADL FunctionPrototype* and *HardwareComponentPrototype* refer to the application functions in an automotive vehicle and the related hardware components

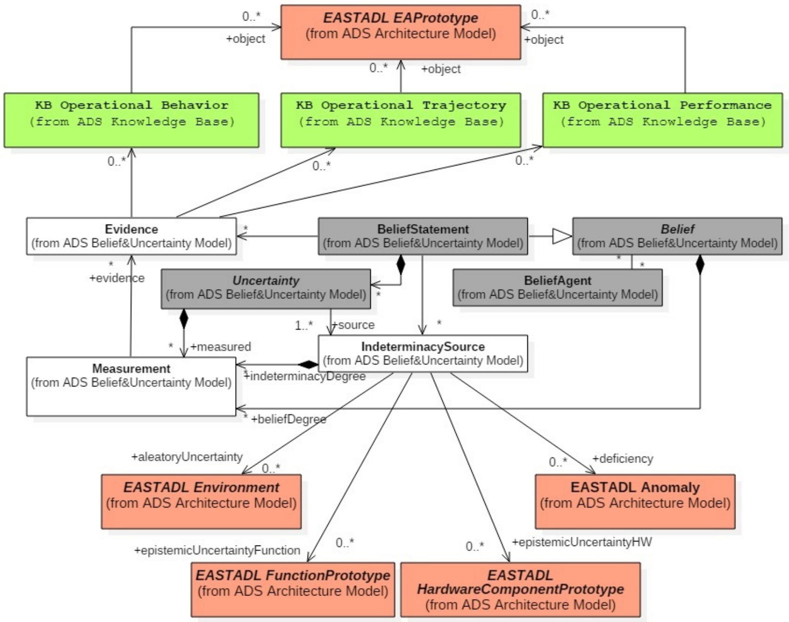
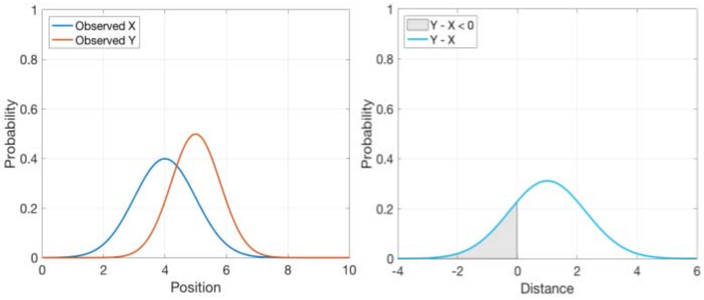

EASTADL EAPrototype
(from ADS Architecture Model)
0..*
+object
0..*
+object
0..*
+object
KB Operational Behavior
(from ADS Knowledge Base)
KB Operational Trajectory
(from ADS Knowledge Base)
KB Operational Performance
(from ADS Knowledge Base)
0..*
0..*
0..*
Evidence
(from ADS Belief&Uncertainty Model)
*
BeliefStatement
(from ADS Belief&Uncertainty Model)
Belief
(from ADS Belief&Uncertainty Model)
*
*
BeliefAgent
(from ADS Belief&Uncertainty Model)
*
+evidence
Uncertainty
(from ADS Belief&Uncertainty Model)
*
1..*
+source
*
*
+measured
IndeterminacySource
(from ADS Belief&Uncertainty Model)
Measurement
(from ADS Belief&Uncertainty Model)
+indeterminacyDegree
*
+beliefDegree
+aleatoryUncertainty
0..*
0..*
+deficiency
EASTADL Environment
(from ADS Architecture Model)
EASTADL Anomaly
(from ADS Architecture Model)
+epistemicUncertaintyFunction
0..*
0..*
+epistemicUncertaintyHW
EASTADL FunctionPrototype
(from ADS Architecture Model)
EASTADL
HardwareComponentPrototype
(from ADS Architecture Model)

Fig. 5. Key design concepts of integrating UM, KB, and EAST-ADL for ADS description.

providing the computation and communication capability. As mentioned earlier, ADS has an increased heterogeneity in regard to the composition of functions and components through the inclusion of AI functions and specific hardware components. The modeling support presented constitutes a formal basis for clarifying and managing the related uncertainties. For the AI related artefacts, the U-Model provides support for the declaration of uncertainty patterns (e.g. periodic and random) and uncertainty measurement with probability, ambiguity and fuzziness. See [13] for further details.

The modeling support constitutes a formal basis for the reasoning of ADS behaviors and safety risks. As example, let's assume a vehicle system is subject to a behavior requirement on its state given as the relation of its position to the positon of other vehicles: $X < Y$. Due to uncertainties in the system, both variables cannot be determined exactly. The uncertainty of monitoring function (i.e. *IndeterminacySource*) is defined with an additional belief statement (i.e. *BeliefStatement*), where observed X and Y are given as independent random variables (i.e. *Evidence*) with the uncertainties quantified with normally distribution ($X \sim N(\mu_X, \sigma_X^2)$ and $Y \sim N(\mu_Y, \sigma_Y^2)$). The measurement is shown in Fig. 6 (a) where the monitoring of variable X delivers: $\mu_X = 4$ and $\sigma_X = 0.8$. Similarly, the monitoring of variable Y results in $\mu_Y = 4.5$ and $\sigma_X = 1$. To validate the stipulated constraint $X < Y$, we can evaluate $(Y - X) > 0$. As both X and Y are normal distributed, the result of $Y - X$ is normally distributed itself, with $(Y - X) \sim N(\mu_Y - \mu_X, \sigma_Y^2 + \sigma_X^2)$. The distribution of this new variable is shown in Fig. 6 (b). The area under the distribution at a distance below 0 then represents the probability that the initial constraint is violated and leads to a hazardous state.



(a) (b)

Fig. 6. Probabilistic uncertainty measurements for two monitored variables.

4 Related Work

Domain specific modeling frameworks have been developed for safety and mission-critical cyber-physical systems, such as AADL [15] and EAST-ADL [5]. Compared to the more generic systems modeling framework, SysML [14], these technologies provide dedicated support for the domain concepts regarding both methodology and technology. For the specification of uncertainties, all these modeling technologies need to be enhanced with additional modeling support for the descriptions of related patterns and metrics. The work presented in this paper has it primary goal of consolidating the architectural and operational concepts of ADS so that the descriptions of uncertainty can be semantically justified. Uncertainty as a condition of information quality has been one key concept of information theory [16]. The measure of *entropy* has been used for the quantification the information disorder or uncertainty. For CPS, the studies of uncertainties presented in [11, 12, 13, 17] constitutes the basis of our work. For CPS, explicit uncertainty modeling also constitutes the basis for effective diagnostics, dynamic anomaly detection and quality-of-service adaption [18, 19, 20].

5 Conclusion

CPS as an engineering field cuts across a number of application domains and technical areas beyond the conventional domains of control engineering and embedded systems. This work aims to support the integration of separately developed heterogeneous functions and components (including AI functions and components) by proposing a Knowledge-Base (KB) strategy for a systematic treatment of uncertainties. Through an integration with U-Model and EAST-ADL, the approach makes it possible for each uncertainty description to have well-defined semantics and architectural targets. Future work will consider the enrichment of uncertainty modeling for the analysis of safety knowledge for ADS as well as the synthesis of safety rules.

Acknowledgements. The research has been supported by the Swedish government agency for innovation systems (VINNOVA) in the ESPLANADE project (ref 2016-04268).

References

1. EC. https://ec.europa.eu/transport/themes/its_en. Last accessed 2018/05/28.
2. SAE International: SAE J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for on-Road Motor Vehicle. 2016.
3. FREDERICK, H.-R., WATERMAN, D., LENAT, D.: Building Expert Systems. Addison-Wesley. ISBN 0-201-10686-8. (1983)
4. CHEN, D., LU, Z.: A Methodological Framework for Model-Based Self-management of Services and Components in Dependable Cyber-Physical Systems. In: Advances in Dependability Engineering of Complex Systems, pp. 97-105. Springer, Cham (2017).
5. KOLAGARI, R., et.al.: Model-Based Analysis and Engineering of Automotive Architectures with EAST-ADL: Revisited. In: Intl. Journal of Conceptual Structures and Smart Applications (IJCSSA). Vol 3, Iss 2, 2015. IGI Global Publishing, Hershey, USA. (2015)
6. JOHANSSON R. et al.: A Strategy for Assessing Safe Use of Sensors in Autonomous Road Vehicles. In: SAFECOMP 2017. LNCS, vol 10488. Springer, Cham. (2017)
7. MIYAJIMA, C., et.al.: Analyzing driver gaze behavior and consistency of decision making during automated driving. IEEE Intelligent Vehicles Symp., 2015-Augus(Iv).(2015)
8. Michon, J. A.: A Critical View of Driver Behavior Models: What Do We Know, What Should We Do? In: Human Behavior and Traffic Safety. Plenum, 1985.(1985)
9. ALBUS, J.S., PROCTOR, F.G.: A Reference Model Architecture for Intelligent Hybrid Control Systems. In: Proc. of the IFAC, San Francisco, CA, 1996.(1996)
10. CIMATTI, A., et al.: NUSMV a new symbolic model checker. In: International Journal on Software Tools for Technology Transfer. March 2000, Vol 2, Iss 4, pp 410–425. (2000)
11. DER KIUREGHIAN, A., DITLEVSEN, O.: Aleatory or epistemic? Does it matter?. Structural Safety 31.2, 105-112. (2009).
12. HERMANN G. M.: Quantifying uncertainty: modern computational representation of probability and applications, Extreme Man-Made and Natural Hazards in Dynamics of Structures. In: NATO Security through Science Series, 2007, 105-135. (2007)
13. ZHANG, M., et al.: Understanding uncertainty in cyber-physical systems: A conceptual model. In: Eu. Conf. on Modelling Foundations and Applications. Springer, Cham (2016).
14. SysML. OMG Systems Modeling Language (OMG SysML™), OMG.
15. FEILER, P. H., GLUCH, D. P.: Model-Based Engineering with AADL: An Introduction to the SAE Architecture Analysis & Design Language. Addison-Wesley.2012. (2012)
16. MACKAY, DAVID J. C.: Information Theory, Inference, and Learning Algorithms. Cambridge: Cambridge University Press, 2003. ISBN 0-521-64298-1. (2003).
17. AVEN, T., et al.: Uncertainty in risk assessment: the representation and treatment of uncertainties by probabilistic and non-probabilistic methods. Wiley. (2013).
18. MEEDENIYA, I. et al.: Evaluating probabilistic models with uncertain model parameters. Software&Systems Modeling 13.4, 1395-1415 (2014).
19. YING, J. et al.: A hidden Markov model-based algorithm for fault diagnosis with partial and imperfect tests. IEEE Trans. on Systems, Man, and Cybernetics, Part C. 30.4. (2000).
20. ZHANG, X., GU, C., LIN, J.: Support vector machines for anomaly detection. In: Intelligent Control and Automation, 2006. IEEE 6th World Congress on WCICA. (2006)