Unifying Control and Verification of Cyber-Physical Systems (UnCoVerCPS)

WP3 Online verification for control (Task T3.5)

D3.4 Assessment of the certifiability for the relevant standards

---

Project funded by the European Commission under Horizon 2020, the Framework Programme for Research and Innovation (2014-2020). Grant No.: 643921.

WP3	D 3.4 Assessment of the certifiability of the UnCoverCPS approach
Authors	Jean-Louis CAMUS
Short description	Assessment of the certifiability of UnCoverCPS approach
Deliverable type	R
Dissemination level	PU
Delivery date	2019-01-21
Internally accepted by	Matthias Althoff, Xavier Fornari, Goran Frehse, Jens Oehlerking
Date of acceptance	2018-01-20
Keywords	Safety, Autonomy, Certification, ISO 26262, SOTIF

Document history:

Version	Date	Author/ Reviewer	Description
1.0	2018-05-25	Jean-Louis CAMUS	Under construction, first outline
1.1	2018-01-20	Jean-Louis CAMUS	First draft
1.2	2019-01-21	Jean-Louis Camus	Final version

Contents

1 Introduction

1.1 Purpose

This report addresses task 3.5 of the UnCoVerCPS (UCPS) proposal: assessment of the certifiability of system/software that is developed according to the UnCoVerCPS approach, for the relevant standards.

T 3.5 is described as follows in the proposal:

Safety is a key issue for cyber-physical systems addressed by various standards (ISO 26262, IEC 61508, EN 50126 or DO-178C), which use frameworks that typically provide:

- a domain-specific, risk-based approach to determine the safety level;
- applicable requirements or clauses to each safety level so as to avoid unreasonable residual risk;
- a domain specific safety lifecycle (management, development, production, operation) and support tailored for each corresponding phase.

A first objective of this task is to identify the impact of these safety standards with respect to the UnCoVerCPS approach at the application level. UnCoVerCPS can be considered as a true break through as it introduces verification and code generation of hybrid controllers as well as on-the-fly verification. **The task therefore investigates the objectives of the standards with respect to this new process and will propose means of conformance**.

A **second objective** of this task is the identification of impacts on the **tools** developed in UnCoVerCPS, since safety standards have or may have an impact on development processes. This objective will be addressed by **Task 3.5.2**.

The certifiability analysis is done with respect to safety objectives, which concern not only the development of application code, but also the complete process.

Although UnCoVerCPS addresses various classes of autonomous systems, such as autonomous vehicles (AV) or robots, we perform the analysis for one category of these systems, which are the autonomous vehicles and more precisely the autonomous cars. The main reason is that this is the domain which is currently best addressed by applicable or emerging standards. This report is more specifically defined in the proposal as a Report on UnCoVerCPS methodology with respect to ISO **26262** objectives (contribution to deliverable 3.3). As an extension, we will also address the emerging ISO 21442 standard addressing the Safety Of the Intended Functionality (**SOTIF**).

1.2 Report Outline

This report is structured as follows:

- It identifies the general issues of autonomy for traditional certification frameworks;
- It briefly introduces emerging approaches for autonomous vehicles, in particular the SOTIF (Safety Of the Intended Functionality);
- It analyzes certifiability of autonomous vehicles developed with unconstrained use of UnCoverCPS techniques;
- It proposes some tracks for making UCPS based autonomous vehicles certifiable;
- The main conclusion are drawn about current and possible certifiability of UCPS vehicles.

1.3 Glossary and acronyms

1.3.1 Acronyms

AI	Artificial Intelligence
ARP	Aerospace Recommended Practice
ASIL	Automotive Safety Integrity Level
AV	Autonomous Vehicle
ASIL	Automotive Safety Integrity Level
CNN	Convolutional Neural Network
ECSS	European Cooperation for Space Standardization
ESA	European Space Agency
FMEA	Failure Modes and Effects Analysis
HARA	Hazard analysis and Risk assessment
HAZOP	HAZard and Operability studies
I&C	Instrumentation and Control
IDAL	Item Development Assurance Level
IE	Initiating Event
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
MIL	Model In The Loop
PIL	Processor In the Loop
QM	Quality Management
RTCA	Radio Technical Committee for Aeronautics
SAE	Society of Automotive Engineers
SIL	Safety Integrity Level or Software In the Loop
SOTIF	Safety Of the Intended Functionality
SSIL	Software Safety Integrity Level
STAMP	Systems Theoretic Accident Model
STPA	Systems Theoretic Process Analysis
TD	Tool error Detection
TCL	Tool Confidence Leve
TI	Tool Impactl

UCPS UnCoVerCPS

1.3.2 Glossary

MIL Model In The Loop

Refers to the kind of testing done to verify the accuracy / acceptability of a plant model or a control system. MIL testing means that the model and its environment are simulated in the modeling framework without any physical hardware components. MIL allows testing at early stages of the development cycle.

SIL Software In the Loop

Refers to the kind of testing done to validate the behavior of the auto generated code used in the controller. The embedded software is tested within a simulated environment model but without any hardware.

PIL Processor In the Loop

Refers to the kind of testing done to validate the referenced model by generating production code using the model reference target. The code is cross-compiled for and executed on a target processor or an equivalent instruction set simulator. PIL level of testing can reveal faults that are caused by the target compiler or by the processor architecture.

2 General Issues of Autonomy for Traditional Certification

2.1 Paradigm of Traditional Safety Standards

Application domains where a system may harm or kill people, such as aerospace, rail, nuclear, chemical industry and more recently car industry have accumulated experience over the past decades regarding risks analysis and means to mitigate those risk. International experts' groups have defined guidelines with objectives, processes, methods and techniques for systematic means of risk analysis and mitigation, for instance:

- Civil aviation: ED79A/ARP4754A [1] , ED135/ARP4761 [2], ED12C/DO178C [3], ED80/DO254 [4]
- Space: ECSS-Q30 [5], ECSS-Q40 [6], ECSS-Q80 [7]
- Railway : EN 50126 [8], EN 50128 [9], EN 50129 [10]
- Nuclear: IEC 60880 [11], IEC 61226 [12], IEC 61513 [13]
- General purpose industry standard: IEC 61508 [14]
- Process industry: IEC 61511 [15]
- Automotive: ISO 26262 [16]

A comparison of these standards is provided in [17].

These industries have achieved a very high degree of safety; for instance, there has been no passenger killed in any large aircraft in 2017 all over the world.

The underlying paradigm of traditional safety standards is generally based on the following processes (with variations of terminology and/or number of steps):

- Identification of hazards;
- Definition of safety goals and functional requirements for mitigating those hazards;
- Definition of technical requirements;
- Design and implementation of a solution complying with the requirements;
- Verification and validation.

These standards differentiate:

- Random hardware faults.
- Systematic faults (due to requirements, design or implementation errors).

Criticality or integrity levels are usually attributed to functions and or architecture elements, for instance, SIL (Safety Integrity Level in rail and general industry), DAL (Development Assurance Level in civil aviation) or ASIL (Automotive Safety Integrity in automotive). For elements having systematic faults, such as software, the Safety Integrity Level or Development Assurance Level determines the degree of rigor for developing elements.

Important characteristics of this classical paradigm are the following:

- There are explicit requirements;
- These requirements define a deterministic behavior;
- Based on the requirements, one can determine for every possible input (or input sequence) the expected response of the system (one usually groups behaviors into equivalence classes for making verification and validation feasible);
- The most important verification activities are requirements-based (testing, review and possibly formal verification).

2.2 General Characteristics of Autonomous Vehicles

Although UnCoVerCPS addresses various classes of autonomous systems, such as autonomous vehicles (AV) or robots, we perform the analysis for one category of these systems, which are the autonomous vehicles and more precisely the autonomous cars. The main reason is that this is the domain which is currently best addressed by applicable or emerging standards. This report is more specifically defined in the proposal as a Report on UnCoVerCPS methodology with respect to ISO **26262** objectives (contribution to deliverable 3.3). As an extension, we will also address the emerging ISO 21442 standard addressing the Safety Of the Intended Functionality (**SOTIF**).

In this section we address the general case of AVs. The specific case of AVs based on the UnCoVerCPS principles are analyzed in sections 4,5 and 6 in this document.

2.2.1 Overall Architecture

The main components of an autonomous vehicles are:

- Perception;
- Trajectory planning;
- Actuation.

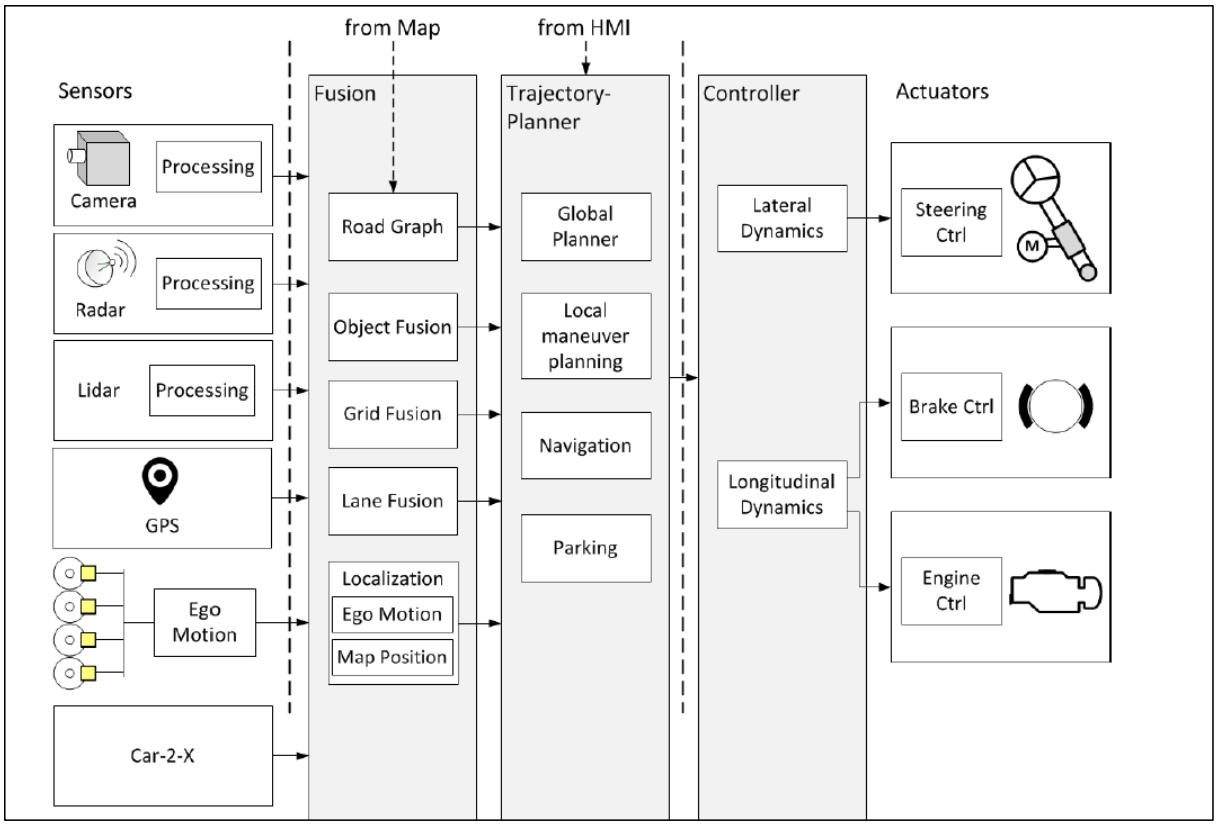

Figure 1 Simplified Example of Automated Driving Control System Architecture

2.2.2 Perception

Perception is a very complex functionality. It relies on advanced technologies such as smart sensors, sensor fusion, object recognition and tracking. Furthermore, it is nearly impossible to not use Artificial Intelligence techniques such a Convolutional Neural Networks (CNN) for object recognition.

There are several issues that cumulate:

- The quality of the physical sensor's signals, in case of strong or weak light, rain, snow, dust;
- Challenges and uncertainties in pattern recognition algorithms. Figure 2 (source [18]) shows an example of surprising lack of robustness of pattern recognition algorithms in case of minor perturbations.

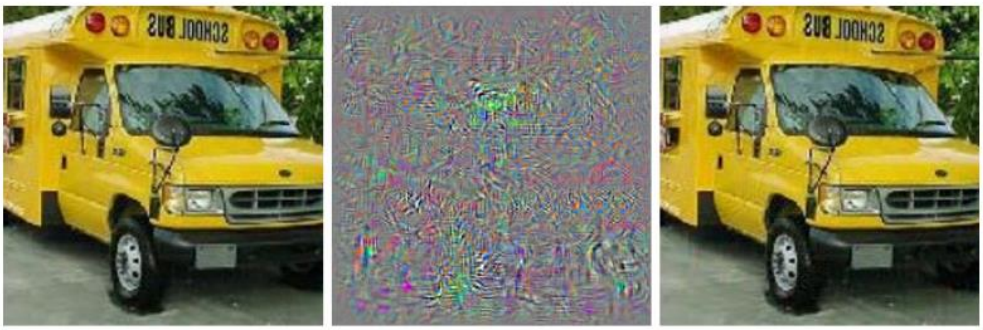

Figure 2 Sensitivity of automated perception to minor image degradation

Perception based on CNN is a functionality for which traditional safety standards are inappropriate, since it is difficult/impossible to write complete and deterministic requirements and verify the system against these.

Perception is not addressed in UnCoVerCPS but remains probably the weakest link in the control loop from a safety perspective.

2.2.3 Trajectory Planning

Trajectory planning is obviously highly critical since an inappropriate trajectory may result in collisions of the ego car with other cars and/or other obstacles.

It has conflicting objectives:

- Safety;
- Efficiency/optimality in the execution of the route. Just driving at 5 km/h for safety all the time would not be accepted by customers.

Optimality requires complex decision algorithms, possibly Artificial Intelligence (AI). Systems solely based on AI cannot be addressed by traditional safety standards for high degrees of criticality. UnCoVerCPS allows to develop systems that include AI and to verify those functions, as addressed in section 5.

2.2.4 Actuation

Actuation is typically realized by:

- Classical control algorithms;
- Simple physical sensors (speed, acceleration);
- Electrical/Hydraulic/Mechanical actuators

These aspects are well addressed by traditional safety standards, including ISO 26262.

Note that adaptive control (which is at the border of trajectory planning and actuation) remains manageable under classical safety standards as mentioned in [19].

2.2.5 Verification and Validation Issues

The space of situations that can be met by an AV is at best huge or even uncountable. According to [20]: “Autonomous vehicles would have to be driven hundreds of millions of miles and sometimes hundreds of billions of miles to demonstrate their reliability in terms of fatalities and injuries. Under even aggressive testing assumptions, existing fleets would take tens and sometimes hundreds of years to drive these miles—an impossible proposition if the aim is to demonstrate their performance prior to releasing them on the roads for consumer use. Therefore, at least for fatalities and injuries, test-driving alone cannot provide sufficient evidence for demonstrating autonomous vehicle safety. Developers of this technology and third-party testers will need to develop innovative methods of demonstrating safety and reliability”.

We address in section 5.3 how to combine physical test with simulation for validation of AVs.

2.3 Autonomous Vehicles and Traditional Safety Standards Summary

To summarize, AVs generally do not fit traditional safety standards for the following reasons:

- There are no explicit requirements for functionalities involving AI (most notably perception);
- Therefore, there is no possibility for requirements-based review and testing;

- The huge number of diverse situations is unmanageable;
- The complexity of the functionality is even such that it is hard to ensure that the specified functionality would ensure safety, even in the absence of error or failure of the implementation.

New approaches such as the SOTIF approach and STPA have emerged to (partly) address such issues (see section 3).

3 Emerging Approaches for Safety of Autonomous Vehicles

3.1 SOTIF

3.1.1 What is the SOTIF?

The SOTIF (Safety Of The Intended Functionality) is a concept that has emerged for addressing complex car systems. ISO 26262 addresses the safety as the absence of unreasonable risks that arise from malfunctions of the electric/electronic system in vehicles. However, for some systems that rely on sensing the environment and/or make complex decisions, there can be safety violations by limitations in the intended function of a system that is free from the faults defined in ISO26262. Examples of such safety violations include:

- The inability of the function to correctly comprehend the situation and operate safely, including functions using machine learning algorithms;
- Insufficient robustness of the function with respect to variations with sensor input or diverse environmental conditions.

The absence of this class of safety violations is defined as the Safety Of The Intended Functionality (SOTIF). A Publicly Available Specification (PAS) ISO/PAS 21448 :2018 is going to be released. Then it will serve as a basis for developing an ISO standard (ISO 21448).

This document provides guidance on the design, verification and validation measures applicable to avoid an unintended behavior in the system in the absence of the faults covered by ISO26262, resulting from technological and system shortcomings and/or reasonably foreseeable misuse.

3.1.2 SOTIF Concepts

We provide here a summary of the concepts as they are defined in the SOTIF PAS 21448. Please note that the following text reflects PAS 21448, and that there may be insufficiencies that we do not resolve here. The ISO working group developing the ISO standard 21448 has just started working on improvements.

A **scenario** is description of the temporal development between several scenes in a sequence of scenes. This is shown as a dashed path in the tree below.

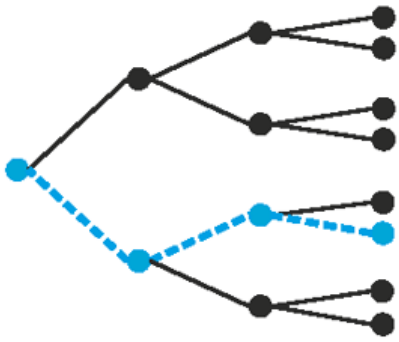

Figure 3 Scenario Concept in SOTIF

A **scene** is a snapshot of the environment including the scenery, dynamic elements, and all actor and observer self-representations, and the relationships between those entities.

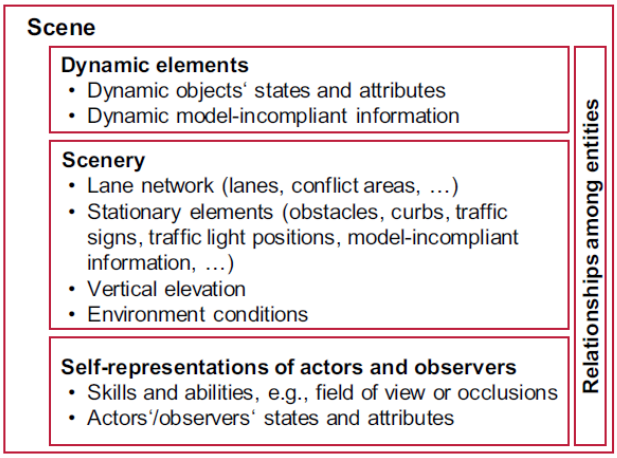

Figure 4 Scene Concept in SOTIF

A **situation** is a selection of an appropriate behavior pattern at a particular point of time. Please not that the situation contains what could be considered as a relevant scene.

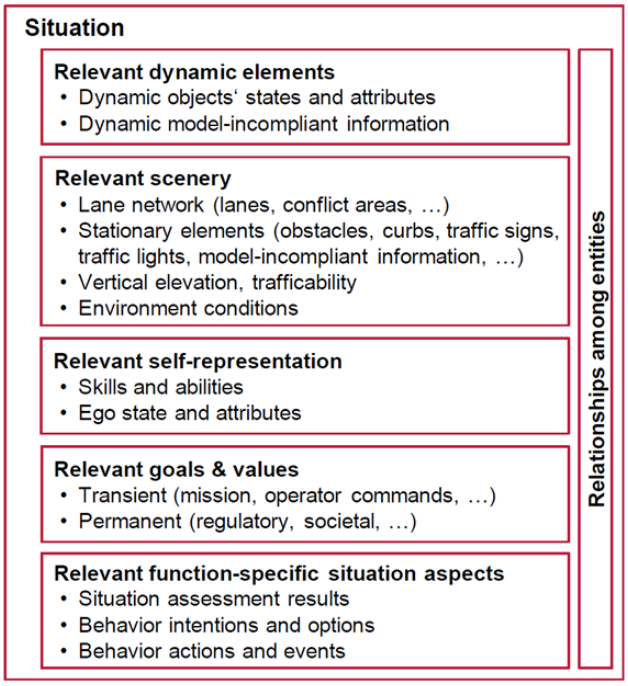

Figure 5 Situation Concept in SOTIF

A **use case** is the specification of a generalized field of application, possibly entailing the following information about the system:

- One or several scenarios;
- The functional range;
- The desired behavior;
- The system boundaries.

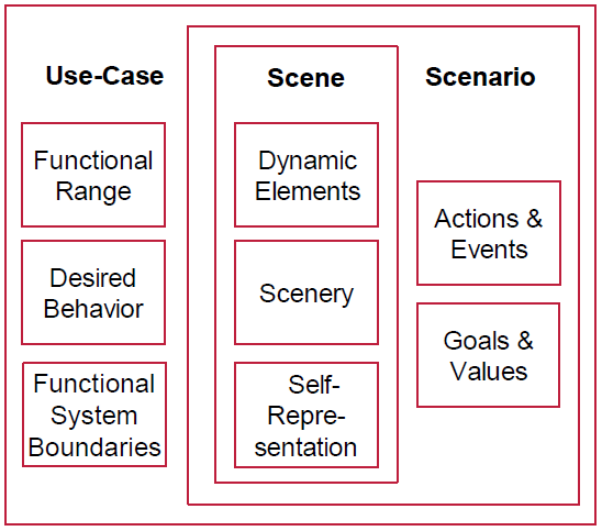

Figure 6 Use Case Concept in SOTIF

A **triggering event** is a specific condition of a driving scenario that serves as an initiator for a subsequent system reaction, possibly leading to a hazardous event.

3.1.3 SOTIF Goal

The scenarios which are part of the relevant use cases are classified into four areas.

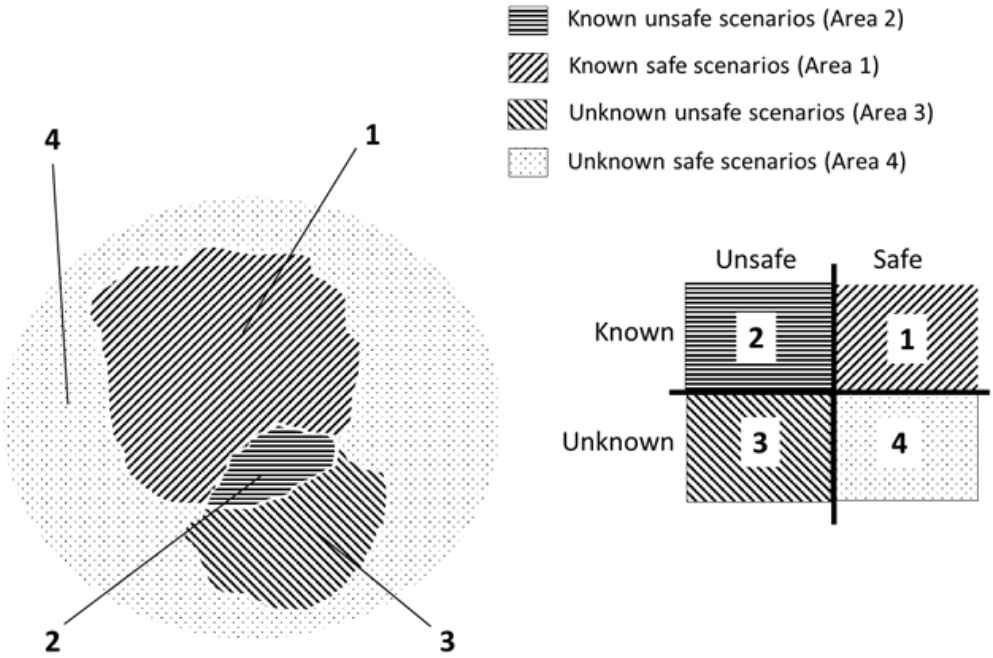

Figure 7 Visualisation of the Known/Unknown Use Case categories in SOTIF.

The **goals** of the SOTIF process with respect to Areas 1, 2, and 3 and relevant scenarios are:

- **Area 1**: Maximize or maintain area, while minimizing areas 2 & 3. This retains or improves safe functionality;
- **Area 2**: Minimize area with technical measures to an acceptable level; evaluate the potential risk; and, if necessary, shift hazardous scenarios to area 1 by improving the function or by restricting the use/performance;
- **Area 3**: Minimize area (the risk of the unknown) as much as possible with an accepted level of effort (every detected hazardous scenario will be shifted to area 2).

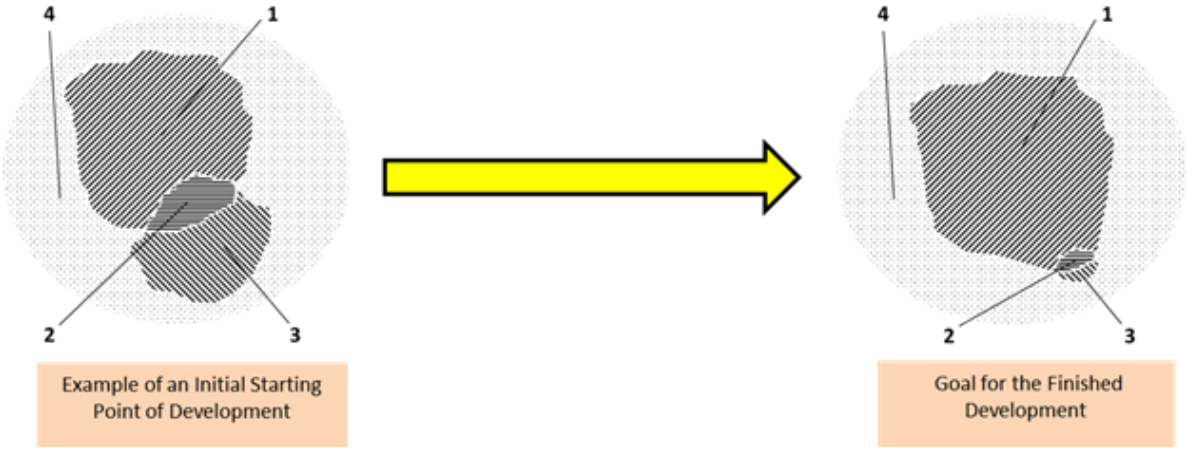

Figure 8 – Evolution of the use case categories with the SOTIF activities

3.1.4 SOTIF Lifecycle

Figure 9 describes a possible phasing between the SOTIF and ISO 26262 activities. The concept phase will typically require several iterations (not shown) to produce a final item definition.

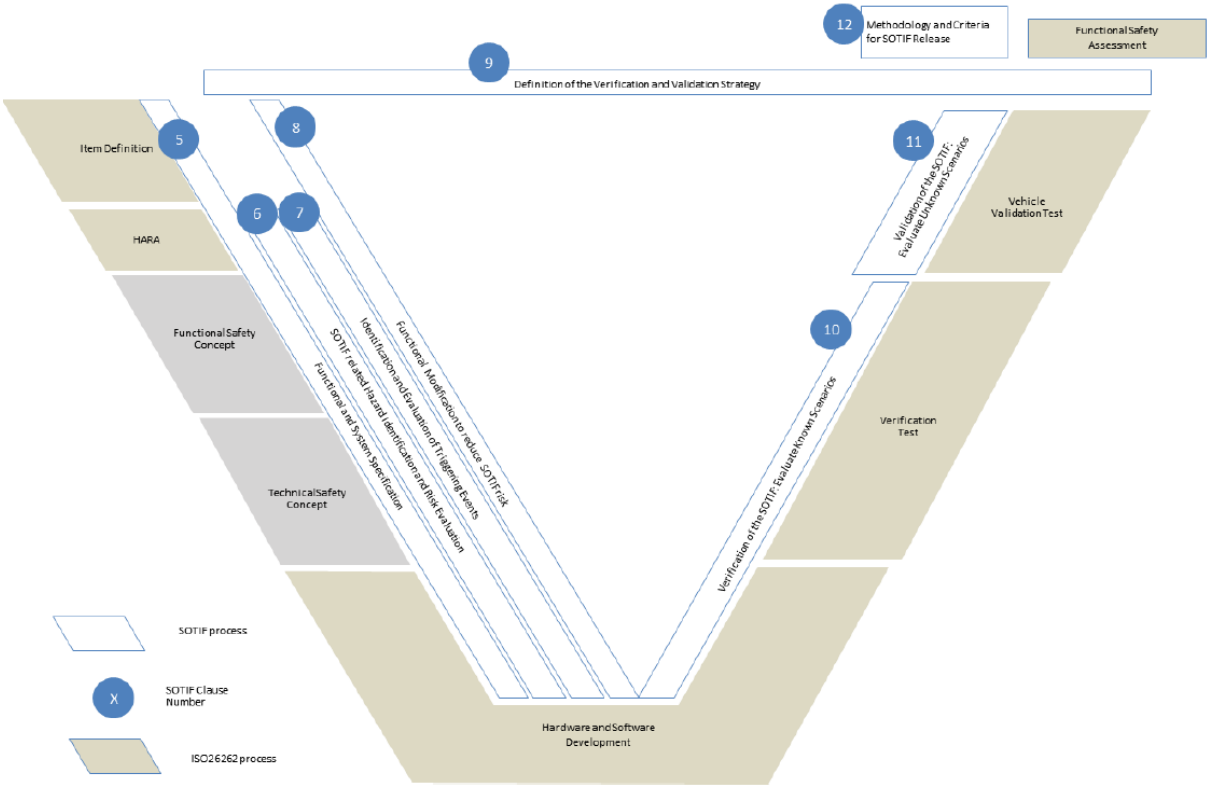

Figure 9 – - Possible Phasing of Product Development activities between SOTIF and ISO26262

The SOTIF methods are:

- Identify and evaluate the SOTIF risks associated with the intended functionality (Clause 6);
- Identify and evaluate hazardous use cases (Clause 7);
- Improve the system design as necessary through functional improvement or use case restriction to reduce SOTIF risk (Clause 8);
- Verify and validate the appropriateness of the design with respect to the SOTIF (Clause 9-11).

The SOTIF is used as a framework for assessing UnCoVerCPS in section 4. SOTIF mentions STPA as one of the possible means for analyzing the architecture (see section 3.2).

3.2 STPA Overview

3.2.1 STAMP/STPA Rationale

Traditional system safety approaches are being challenged by the introduction of new technology and the increasing complexity of the systems we are attempting to build. STAMP is a new systems thinking approach to engineering safer systems described in Nancy Leveson's book “Engineering a Safer World”

(MIT Press, January 2012). While relatively new, it is already being used in space, aviation, medical, defense, nuclear, automotive, and other sectors.

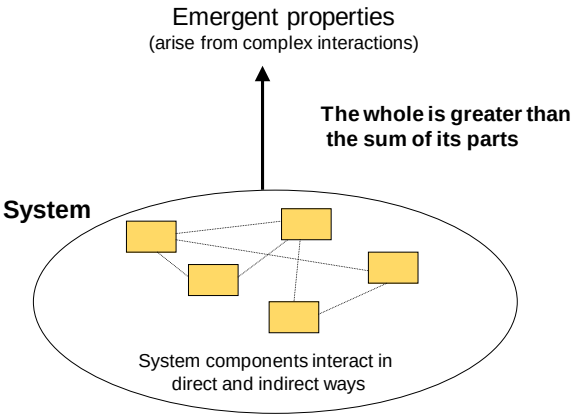

Figure 10 Safety and security are emergent properties.

STPA (Systems-Theoretic Process Analysis) is a powerful hazard analysis technique based on STAMP. These tools are increasingly used across diverse industry sectors. Application areas have included aviation, air traffic control, space, defense, the automotive industry, railways, chemicals, oil and gas, medical devices, health-care, and workplace safety, with a growing interest coming from new areas such as the pharmaceutical industry and the finance and insurance sectors. Ongoing developments aim at extending the application field of STPA to include security.

3.2.2 STPA Overview

The main paradigm of STAMP and STPA is that accidents are caused by inadequate control. Inadequate control is not limited to consequences of failures of technical components. This approach can capture software errors, sensors errors, human errors, organizational issues.

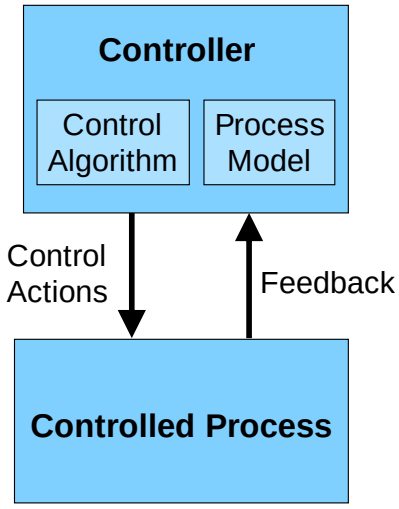

Figure 11 STAMP basic control loop.

Primarily, the system to analyze is described as a hierarchy of controllers and controlled processes.

- Control actions are provided to affect a controlled process;
- Feedback may be used to monitor the process;
- The process model (beliefs) is formed based on feedback and other information;
- The control algorithm determines appropriate control actions given current beliefs.

Four types of hazardous control actions are usually identified:

1. Control commands required for safety are not given;
2. Unsafe ones are given;
3. Potentially safe commands that are given too early or too late;
4. Control action stops too soon or is applied too long.

The four steps of an STPA analysis are:

1. Identify system accidents, hazards;
2. Draw the functional control structure;
3. Identify unsafe control actions;
4. Identify accident scenarios.

4 Primary Certifiability Analysis of UnCoVerCPS-Based Systems

4.1 Objective

This section analyses the certifiability of systems that would be developed with **unconstrainted** use of technologies from the UnCoVerCPS proposal. Section 5 proposes tracks for making certifiability more likely to be achievable when using UnCoverCPS principles.

4.2 Overview of UnCoVerCPS Principles

4.2.1 Paradigm Shit

As stated in the proposal, the overall goal in UnCoVerCPS is to develop holistic model-based design methods of future cyber-physical systems with a special focus on researching essentially new methods to guarantee safety and reliability in (partially) unknown environments. This is realized by a cross-domain approach for synthesizing and verifying controllers on-the-fly, i.e. during operation. In order to quickly react to situations that become critical, a tight integration between the control software and the verification software is realized.

The industry standard today is to test/verify the controller, while only testing the closed-loop dynamics as shown in Figure 12 on the left side. A further improvement for the industry standard would be to verify the closed-loop dynamics of the complete system (middle), which typically reveals further shortcomings of the system design. In UnCoVerCPS, we are going even one step further by continuously verifying the system on-the-fly during its operation in a changing environment (right side).

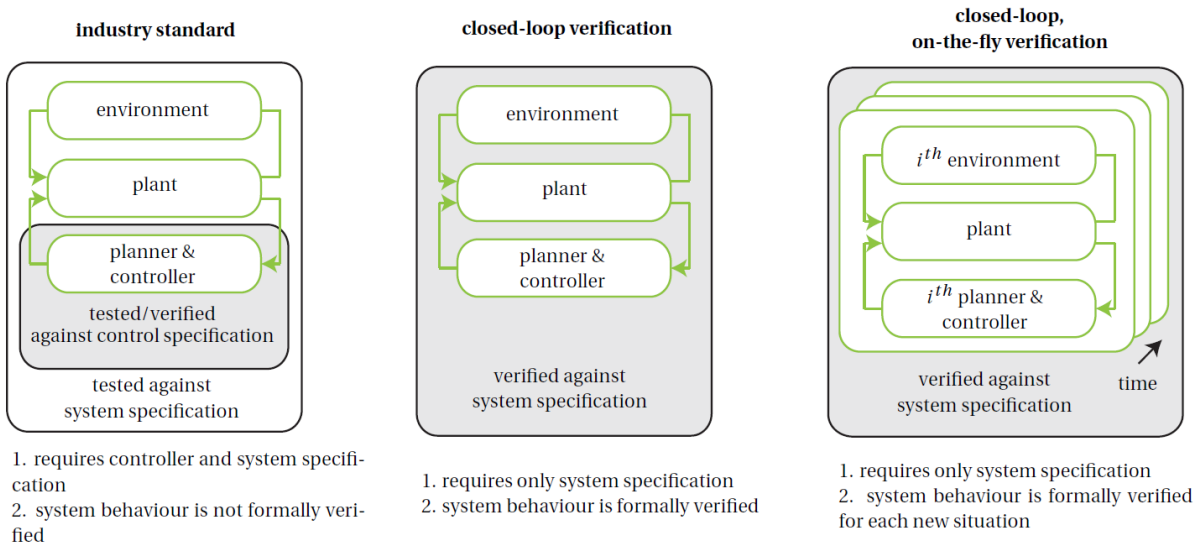

Figure 12 – UnCoVerCPS paradigm shift.

4.2.2 Modelling

First of all, a general description of the hybrid dynamics has to be realized of both the system and the environment to the start the workflow shown in Figure 13. We first derive specialized models for controller design. These are used to synthesize a controller and subsequently generate its implementation (left side). Similarly, we derive models for verification, where we abstract the system to enable automatic formal verification against the formal specification derived from the requirements (right side). Additionally, we leverage these verified abstract models for conformance testing of the generated controller implementations. In particular, we generate behaviors from the abstract models in order to verify or falsify the generated controller implementation. This ensures that with a small amount of tests, one can gain maximum confidence in whether the verification models include all real behaviors or not. This approach is novel and absolutely required to ensure safety in critical environments. Moreover, the figure shows the online extensions that we indicate with the grey on-the-fly annotations. We use information on the environment to create models for verification and control on-the-fly. The results from online verification and the (on-the-fly) controller models are used for online control.

UnCoVerCPS modelling activities are based on classical modelling for continuous dynamics using ordinary differential equations (ODEs) or differential-algebraic equations (DAEs). Next, we add uncertainty in the continuous dynamics, by either adding set-based uncertainties or stochastic uncertainties. This leads in the first case to differential inclusions and in the second case to stochastic-differential equations. The added uncertainty makes it possible to model behaviors of other entities in uncertain environments. In a further step, the continuous dynamics of different discrete modes of a system are aggregated to a hybrid system. In order to properly model discrete mode changes, modularity and hierarchy are supported by the tool SCADE of Esterel Technologies.

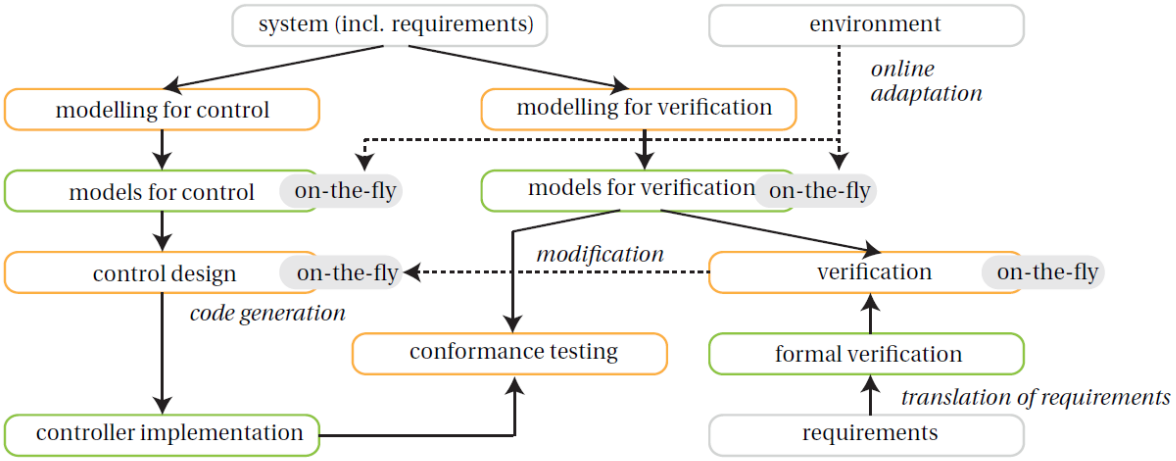

Figure 13 – UnCoVerCPS workflow.

4.2.3 Online verification

After each update from the decision making modules, the newly planned action is verified on-the-fly. We use reachability analysis to rigorously check if the reach-avoid problem formulated above can be satisfied for uncertain initial states, disturbances, sensor noise, and parameters. In order to realize on-the-fly verification, we use several strategies.

First, we improve offline techniques for computing reachable sets. This is realized by exploring new techniques that do not abstract the nonlinear continuous dynamics to a linear one, but to a polynomial one. This poses huge challenges since convex set representations are mapped to non-convex ones when not considering linear dynamics. The benefit, however, is that the abstraction errors in the computation can be drastically reduced, which avoids splitting of reachable sets. Splitting of reachable sets causes exponential complexity, so that its avoidance drastically reduces computation time.

Second, the improved offline techniques are used to pre-compute reachable sets for partial reference trajectories. Those partial reference trajectories can be combined during the decision making. If the final reachable set of one partial reference trajectory is within the initial set of the connecting one, it is guaranteed that all solutions stay within the reachable sets of the combined reference trajectories. Thus, one can obtain reachable sets without explicitly computing them during online operation.

Third, we explore compositional techniques to verify systems faster. Since the complexity of formally verifying systems is superlinear with respect to the number of continuous state variables, composition by parts drastically reduces the computation time. We explore assume-guarantee approaches as well as approximate simulation approaches.

In assume-guarantee reasoning, we assume sets of uncertain inputs to a system and guarantee certain specifications under this assumptions. If the output uncertainties are within the input uncertainties, the verification of all subsystems concludes that the full system is also verified.

4.2.4 Tool support

The following tools, shown on Figure 14, are developed (or complemented) and experimented in the UnCoVerCPS project:

- Modelling tools: SCADE and Simplorer (developed at Esterel Technologies);
- Verification tools: SpaceEx (developed at Université Joseph Fourier Grenoble 1), which verifies hybrid systems with linear continuous dynamics and CORA (developed at Technische Universität München) also verifies hybrid dynamics, but is less mature than SpaceEx, while handling nonlinear dynamics. We will newly develop versions for on-the-fly verification named SpaceExonl and CORAonl;
- Controller synthesis tools: DMPC-HS (newly developed at Universität Kassel) is a tool for model predictive control of non-stochastic systems, while ScenarioMPC (newly developed at Politecnico di Milano) is a tool for model predictive control of stochastic systems;
- Automatic code generation: SCADE (developed at Esterel Technologies) has the capability to generate code with a certified code generator, thus guaranteeing that the code is a correct implementation of the model;
- Conformance testing: ConfTest is newly developed at Robert Bosch GmbH;
- Specification formalization: formalSpec is newly developed at GE Global Research Europe.

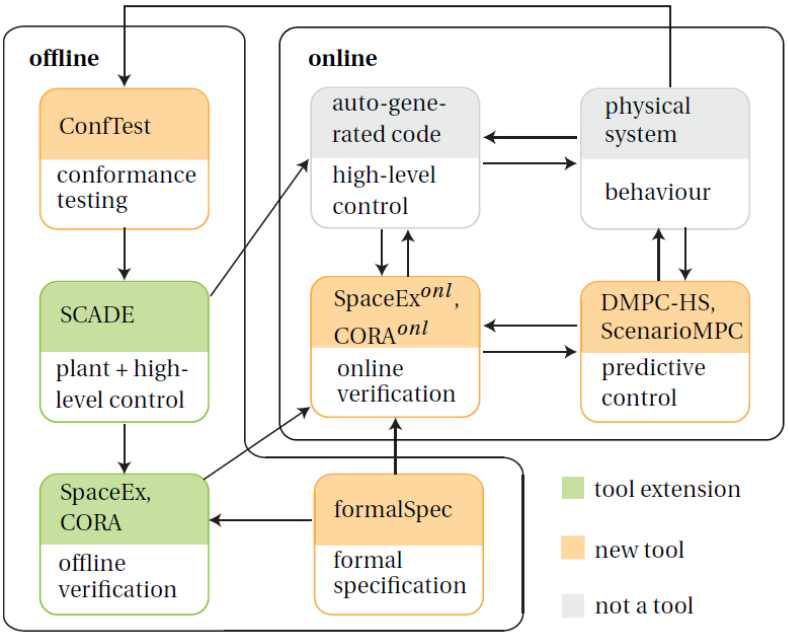

Figure 14 – UnCoVerCPS tools.

Section 5.4 addresses tool qualification requirements.

4.2.5 UnCoVerCPS Automated driving scenario example

The approach is illustrated on an automated driving scenario example.

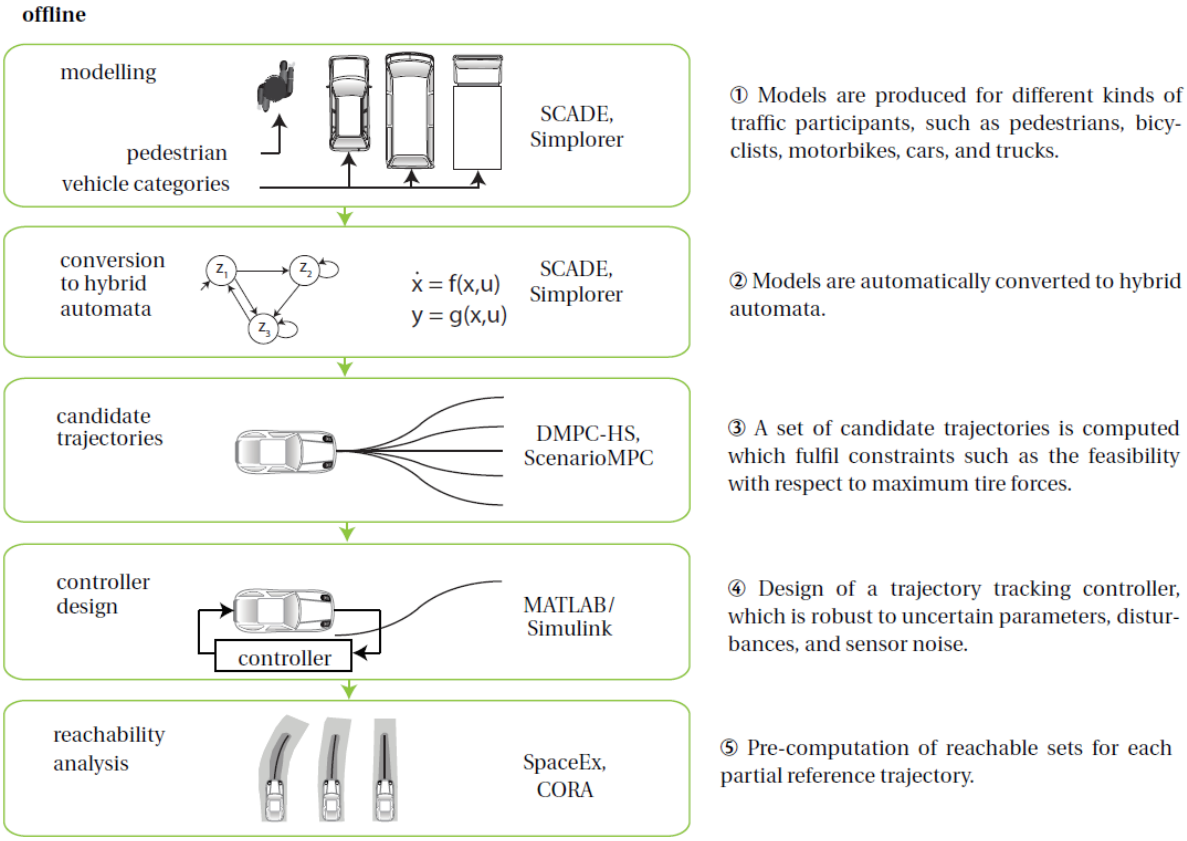

Figure 15 – UnCoVerCPS offline activities.

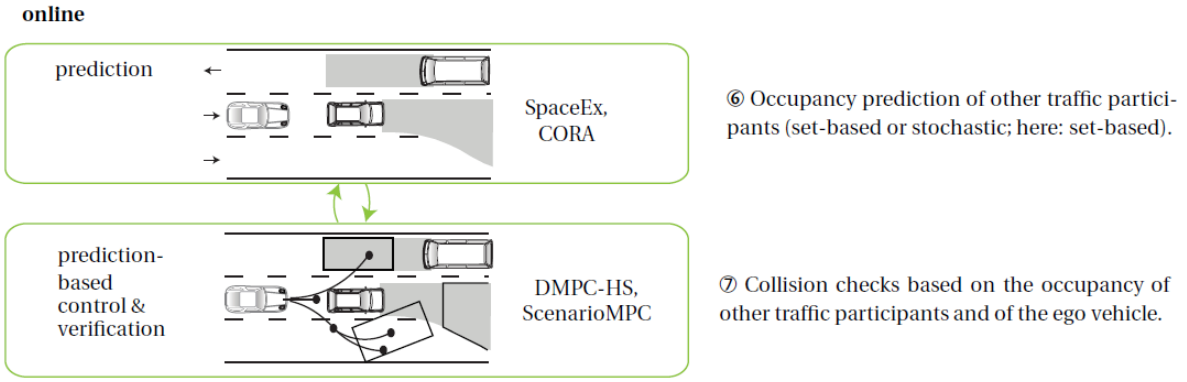

Figure 16 – AV Driving Scenario with on-the-fly verification.

We start by modelling typical traffic participants, such as pedestrians, bicyclists, motorbikes, cars, and trucks (see Figure 16 - ①). The dynamics of each traffic participant is modelled using simple point-mass models since most parameters of vehicles are unknown. We assign to each category a unique set of uncertain parameters, such as maximum acceleration and maximum velocity, which are chosen conservatively, (i.e. larger than they actually are). To further restrict the possible behaviors, we consider traffic rules. In case those rules are not respected, we skip certain assumptions on other traffic participants. Besides surrounding vehicles, we model the dynamics of the own vehicle plus its trajectory tracking controller including set-based and stochastic uncertainties, such as variations in tire-road friction and mass. Next, all models are automatically converted to a hybrid automaton by the planned extension of SCADE, which is used for the subsequent control and verification (②). In order to

quickly plan maneuvers online, a set of parameterized reference trajectories is pre-computed (③). Next, a trajectory tracking controller is designed to follow the reference trajectories, see (④). Given the closed-loop dynamics of the vehicle and the set of reference trajectories, we compute the set of reachable states for each reference trajectory and store the results in a database, (⑤). After all reachable sets are computed, we check during offline computation which reference trajectories can be safely connected, i.e., for which connections the final reachable set is enclosed in the initial reachable set of the connecting reference trajectory. A maneuver automaton is used to store the connectivity information. During online execution, the area occupied by other traffic participants is predicted using set-based and stochastic methods (⑥). For simplicity, we only assume set-based prediction from now on. Based on the predicted occupancy, possible reference trajectories are connected using the maneuver automaton. Since we considered all possible deviations beforehand using reachability analysis, we can prove online that a collision cannot occur by checking if the overapproximated occupancy intersects with the overapproximated occupancies of other traffic participants, see (⑦). In order to ensure that the maneuvers are verified for all times, we only allow reference trajectories to which a braking trajectory is attached, which is only executed when no new safe plan is found. New reference trajectories are only followed after they have been fully verified to ensure safety for all times. The proposed control/verification approach ensures that the ego vehicle will not cause a crash

4.3 Preliminary Screening of UnCoVerCPS Technologies

4.3.1 Technology Readiness Level

First, we analyze methods and tools according to their level of maturity.

Table 1: UnCoVerCPS Technology Readiness Level

Table 1: Technological readiness level of various technologies in UnCoVerCPS (TRL 1: basic principles observed; TRL 2: technology concept formulated; TRL 3: experimental proof of concept; TRL 4: technology validated in laboratory environment; TRL 5: technology validated in relevant environment).

technology	current TRL level	expected TRL level	explanation
on-the-fly synthesis and verification	TRL 1	TRL 2	completely new; no previous work known
conformance testing with set-based and stochastic techniques	TRL 1	TRL 2	completely new; no previous work known
automatic model transformation (e.g. to hybrid automata)	TRL 2	TRL 3	exist in principle; only academic prototypes so far
tool chain for critical cyber-physical systems	TRL 3	TRL 4	prototypes of individual solutions exist, but are disconnected
blade control of wind turbines	TRL 5	TRL 5	first control concepts are in use
verified automated driving	TRL 2	TRL 3	concept for vehicle on-the-fly verification has been formulated but not tested
verified smart grid control	TRL 2	TRL 3	basic ideas have been realised in simulations
human-robot collaborative manufacturing	TRL 3	TRL 4	prototypical robot exists

Most technologies are very far from mature, even at the end of the project, except for qualified code generation with SCADE and blade control of wind turbines.

Below, we use the term “true hybrid” in the sense of a combination of continuous time model (differential equations) with discrete time model (difference equations), as opposed to the case where everything is based on discretized equations.

Concerning true hybrid models and related techniques, in the remainder of this document:

- The use of true hybrid techniques for offline simulation, analysis and optimization is recommended;
- The use of true hybrid techniques embedded in real time safety-critical systems is considered as premature for cases where the hybrid part is on the safety critical path.

4.3.2 Modeling Considerations

The accuracy and simulation time of the models used in the methods are as critical as the technologies themselves.

For the sake of simplicity, we make the following assumptions:

- The models used for offline simulation can be made sufficiently accurate by accumulation of experience, comparison with real data, using powerful computation means (e.g. farms of GPUs) and sufficient time;
- The (implicit or explicit) models used for online computations of the most critical parts (e.g. reachability computations) :
    - Are based on provable properties with respect to explicit assumptions about the ego car and the environment;
    - Are computed with algorithms that can guarantee appropriate Worst Case Execution Time (WCET).

4.4 UnCoVerCPS Potential for SOTIF and ISO 26262 Objectives

4.4.1 Approach

This section analyses for each phase of ISO/PAS 21448 (SOTIF) and ISO 26262 the potential of the UnCoVerCPS approach and techniques for satisfying requirements of those standards concerning activities or lifecycle products. The potential of UnCoVerCPS (Cap) for supporting a given activity is classified as follows:

- **H** : high potential for satisfying the requirements of the standards;
- **M** : medium potential for satisfying the requirements of the standards;
- **N** : neutral with respect to these requirements;
- **P** : problematic; UnCoVerCPS principles or techniques may compromise certification (e.g. by lack of maturity).

4.4.2 SOTIF Potential

ISO/PAS 21448 (SOTIF) is not a standard (it is a Publicly Available Specification) and does not address level 5 (autonomy); this will be achieved by the upcoming standard ISO 21448 (planned for 2021). But since it represents the current state of the art for the automotive community, considered as normative, we use it as a framework for assessing certifiability. As stated in PAS 21448 4 "*The objective clauses of ISO 21448 (Clauses 5.1, 6.1, 7.1, 8.1, 9.1, 10.1, 11.1 and 12.1) are normative. All other content is informative. Compliance to this document can be claimed by listing the objectives and providing an argument that the objectives have been achieved.*"

ISO/PAS 21448 sections are identified in the first columns in the form SOTIF-<sectionnumber>, where <sectionnumber> represents the section number in the SOTIF PAS document.

Table 2: Potential of UnCoVerCPS with respect to SOTIF

Activity	Cap	UnCoVerCPS Capacity Synopsis	Comment
SOTIF-5 Functional and System specification			
SOTIF-5.2 Functional description	H	Hybrid modeling supports dynamic behavior specification.	No comment
SOTIF-5.3 Consideration on system design and architecture	H	Modeling of limitations and of counter measures with analysis of their impact.	No comment
SOTIF- 6 SOTIF Hazard Identification and Evaluation			
SOTIF-6.2 Hazard identification	N	N/A	See HARA support by dedicated techniques
SOTF-6.3 Hazard analysis	M	Hybrid simulation may support detailed dynamic analysis of scenarios and their effect.	There may be some support of simulation for assessing an engineering intuition about hazards. Qualitative Hazard analysis to be done with classical hazard analysis techniques.
SOTIF-6.4 Risk evaluation of the intended function	N	N/A	To be managed with risk analysis techniques
SOTIF-6.5 Specification of a validation target	N	N/A	To be managed with risk analysis techniques
SOTIF- 7 Identification and Evaluation of triggering events			
SOTIF-7.2 Analysis of triggering events	M	Hybrid simulation may support detailed dynamic analysis of scenarios and their effect.	No comment
SOTIF-7.3 Acceptability of the triggering events	M	Monte Carlo simulation may be used for quantitative aspects estimation of behavior that is not analyzable with Fault Tree or Markov process techniques.	To be managed with safety analysis techniques (e.g. FTA, Markov process analysis) where possible.
SOTIF- 8 Functional modifications to reduce SOTIF risk			

Activity	Cap	UnCoVerCPS Capacity Synopsis	Comment
SOTIF-8.3 Measures to improve the SOTIF	**H/P**	**a) On-the-fly verification is a major means of improving the Safety Of The Intended Functionality.** **b) Preliminary exploration of system improvements by simulation.**	**a) On-the-fly verification is the most important added value of UnCoVerCPS.** **b) Advanced hybrid simulation techniques of UnCoverCPS will need more time to mature but will also contribute to SOTIF.**
SOTIF-8.4 Updating the system specification	H/P	Same as above	Same as above
SOTIF- 9 Definition of the Verification and Validation strategy			
SOTIF-9.2 Planning and specification of integration and testing	N	N/A	The plan may include use of simulation techniques.
SOTIF-10 Verification of the SOTIF (Area 2)			
SOTIF-10.2 Sensor verification	M	Hybrid MIL/SIL Simulation.	PAS 21448 Table 5 C Injection of system inputs that trigger the potentially hazardous behavior. D In the loop testing (e.g. SIL / MIL) on selected SOTIF relevant use cases and scenarios. E Vehicle level testing on selected SOTIF relevant use cases and scenarios. F Sensor test under different environmental conditions (e.g. cold, damp, light,visibility conditions)
SOTIF-10.3 Decision algorithm verification	M	Hybrid MIL/SIL Simulation	PAS 21448 Table 6 D In the loop testing (e.g. SIL / MIL) on selected SOTIF relevant use cases and scenarios. E Vehicle level testing on selected SOTIF relevant use cases and scenarios. F Sensor test under different environmental conditions (e.g. cold, damp, light,visibility conditions)

Activity	Cap	UnCoVerCPS Capacity Synopsis	Comment
SOTIF-10.4 Actuation verification	M	Hybrid MIL/SIL Simulation	PAS 21448 Table 7 F In the loop testing (e.g. SIL / MIL) on selected SOTIF relevant use cases and scenarios.
SOTIF-10.5 Robustness and Controllability verification	M	Hybrid MIL/SIL Simulation	PAS 21448 Table 8 C In the loop testing (e.g. SIL / MIL) on selected SOTIF relevant use cases and scenarios.
SOTIF- 11 Validation of the SOTIF(Area 3)			
SOTIF- 11.2 Evaluation of residual risk	M	Hybrid SIL Simulation	PAS 21448 Table 9 C In the loop testing on randomized test cases (derived from a technical analysis and by error guessing). K Simulation of selected scenarios
SOTIF- 11.3 Validation test parameters	M	Hybrid Simulation	PAS 21448 Annex A For SOTIF, validation can consist of testing the vehicle under a wide range of operating conditions. It 1057 can be a mixture of SIL, HIL and real-world operation conditions. It may contain some structured testing, dedicated analysis and simulation but the key aspect, especially for area 3, is to have sufficient testing under sufficiently random operating conditions to expose unknown unsafe scenarios.
SOTIF- 12 Methodology and criteria for SOTIF release			
12.2 Methodology for evaluating SOTIF for release	N	N/A	No comment
SOTIF- 12.3 Criteria for SOTIF release	N	N/A	No comment

The main outcome of this SOTIF support potential is the following: SOTIF section 8.3 addresses the measures to improve the SOTIF. This is where UnCoVerCPS has the highest potential contribution:

- On-the-fly verification has the potential to keep the car in safe states, with robustness to the huge/infinite number of possible situations and limitations of the optimizing planning algorithms;
- Advanced hybrid simulation techniques of UnCoverCPS allow preliminary exploration of system improvements by simulation. During system design. Online hybrid simulation may be used provided online critical functions are partitioned.

4.4.3 ISO 26262:2018 Potential

This section analyses for each phase of ISO 26262 the potential of the UnCoVerCPS approach and techniques for satisfying requirements of those standards concerning activities or lifecycle products. *Note 1: Since edition 2 of ISO 26262 is going to be published soon, the analysis is based on the FDIS of ISO 26262:2018.*

Note 2: for readability, only analysis of relevant sections is provided.

Table 3: Potential of UnCoVerCPS with respect to ISO 26262

Activity	Cap	UnCoVerCPS Capacity Synopsis	Comment
ISO 26262-1 Vocabulary			
All	N	N/A	No comment
ISO 26262-Management of functional safety			
All	N	N/A	No comment
ISO 26262-3 Concept phase			
5 Item definition	N	N/A	
6 Hazard analysis and risk assessment	M	Hybrid simulation may support detailed dynamic analysis of scenarios and their effect.	Qualitative analysis to be done with classical hazard analysis techniques.
7 Functional safety concept	N	N/A	No comment
ISO 26262-4 Product development at the system level			
6 Technical safety concept	H/P	6.4.2 Safety Mechanisms: on-the-fly verification.	Partitioning/protection of on-the-fly verification is critical.
7 System and item integration and testing	N	N/A	No comment
8 Safety validation	N	N/A	No comment

Activity	Cap	UnCoVerCPS Capacity Synopsis	Comment
ISO 26262-5 Product development at the hardware level			
All	N	N/A	No comment
ISO 26262-6 Product development at the software level			
6 Specification of software safety requirements	H/P	Introduction of on-the-fly verification: this is the most important added value of UnCoVerCPS.	
7 Software architectural design	H/P	Introduction of on-the-fly verification: this is the most important added value of UnCoVerCPS.	
8 Software unit design and implementation	N	N/A	No comment
9 Software unit verification	N	N/A	No comment
10 Software integration and testing	H/P	Hybrid simulation support.	See ISO 26262-6 10.4.8 NOTE 3 Software integration testing can be executed in different environments, for example: -model-in-the-loop tests; - software-in-the-loop tests; - processor-in-the-loop tests; and - hardware-in-the-loop tests. But hybrid simulation may be difficult to test due to discontinuities.
11 Testing of the embedded software	N	N/A	No comment
ISO 26262-7			
All	N	N/A	No comment
ISO 26262-8 Supporting processes			
All	N	N/A	No comment
ISO 26262-9 Automotive Safety Integrity Level (ASIL)-oriented and safety-oriented analyses			
All	N	N/A	No comment
ISO 26262-10 Guideline on ISO 26262			
All	N	N/A	No comment
ISO 26262-11Guidelines on application of ISO 26262 to semiconductors			
All	N	N/A	No comment

Activity	Cap	UnCoVerCPS Capacity Synopsis	Comment
ISO 26262-12 Adaptation of ISO 26262 for Motorcycles			
All		Not analyzed	No comment

As part of ISO 26262:2018- 4 section 6.4.2, on-the-fly verification introduced by UnCoVerCPS brings an additional means in the technical safety concept for detecting/preventing the effect of failures in the electrical/electronic system.

4.5 Conclusions of Preliminary Certifiability of UnCoVerCPS Application

A preliminary analysis of the certifiability of unconstrained application of UnCoVerCPS proposal elements leads to the following conclusions (refined in the following sections):

- The scope is too generic to perform an accurate safety analysis. One needs to be more specific about the vehicle architecture;
- Several technologies are far from mature, even at the end of the project.

The next section re-considers certification under appropriate assumptions.

5 Proposals for Certifiability

In this section we introduce suggestions for making certification feasible and manageable.

5.1 Making the Architecture Manageable and Verifiable for Safety

In the very general case, without appropriate architecture, AVs cannot be proven to be safe.

Industrial experience and research on safe architectures usually recommend architectures based on decomposition into primary and safing channels:

- The **primary channels** can be devoted to optimality, performance and comfort. They can perform complex computations, with algorithms that may not have classical specification, for instance this can use artificial intelligence techniques;
- The **safing channels** are devoted to safety. Their primary objective is to minimize the risk of hazard, not on optimality. They rely on simple, well established principles.

This a generalization of the so-called simplex architecture [21] where the primary channel is called the complex controller and the safing channel is the safe controller. The primary/safing channels scheme can be chained and/or nested at several levels.

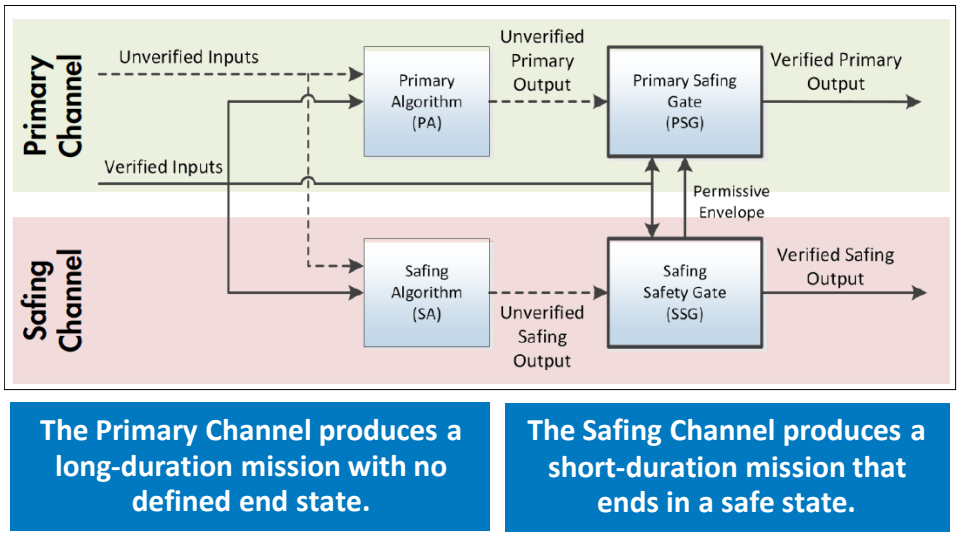

Figure 17 – Example of primary/safing architecture

Several aspects of UnCoVerCPS could **perfectly fit** that decomposition:

- Continuous online occupancy prediction;
- Continuous updated collision check principle.

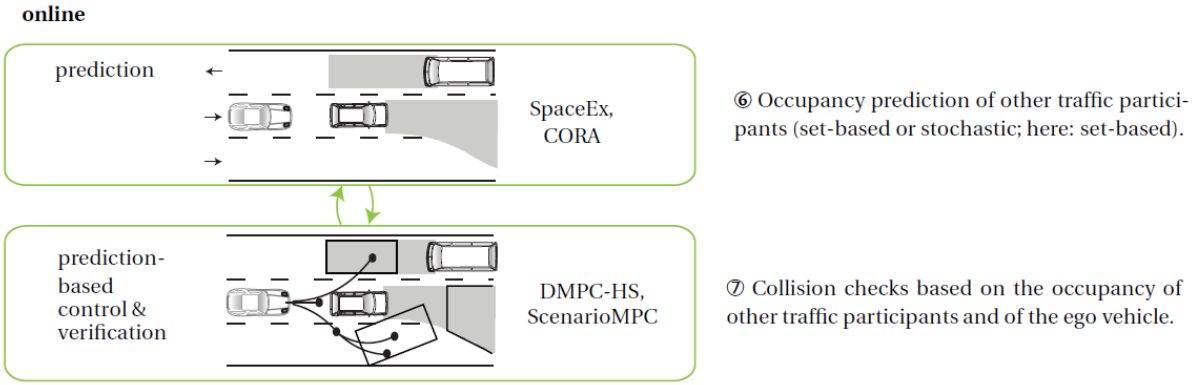

Figure 18 – UnCoverCPS continuous prediction and collision check principle

The architecture of the control system should explicitly exhibit the safing channel(s), and the most provable algorithms developed in UnCoVerCPS should be allocated to the safing channel. For instance [22] proposes an approach for fail-safe motion planning of autonomous vehicles, based on simple conservative laws of physics, and only uses simple sets of equations , with formalized context properties [23], and [20] provides good basis for mobile robots).

5.2 Implementation of the most critical parts

For the critical parts, there shall be a process, methods and techniques that ensure the required integrity level. This aspect is more traditional and is well addressed by classical safety standards (e.g. DO-178C, ISO 26262).

For Model-Based Development, SCADE Suite has a record track of almost 20 years in aerospace, rail, nuclear and automotive. Its characteristics include:

- Formal semantics;
- Full determinism;
- Simulation with model coverage analysis;
- Code generator qualified for the most demanding safety levels (DO-178C TQL 1, ISO 26262 TCL3) and was developed according to DO-330 TQL-1 [24];
- Automated Testing;
- Static memory and bounded Worst Case Execution Time (WCET).

So, it is a perfect fit for the design and implementation of discrete-time embedded control laws. Note that for mastering properties of the algorithms, in particular WCET, one should design the embedded control law in discrete-time form.

5.3 Verification and Validation

5.3.1 Verification

Verification answers the question: did we implement the system correctly with respect to its specification?

This question is essentially addressed by the classical safety standards, for instance several parts of ISO 26262. It is based on reviews and testing against the specification.

For the parts developed from explicit requirements, this is a classical, well mastered issue. This has to be the case at least for the safing channel.

For the primary channel, there may be a grey zone for functionality such as pattern recognition.

5.3.2 Validation

Validation attempts to answer the non-formal question: did we implement the right system with respect to the (usually non-formalized) expected behavior?

It is never easy to answer such a type of question, but for autonomous vehicles it is especially difficult because:

- The number of situations that can be met is huge;
- More fundamentally, there is a lack of expression of what is expected.

If one cannot express completely the expected behavior, but is able to express safety properties, then there is the possibility to perform tests against safety properties used as test oracles as in [25] for the ASTAA project (see Figure 19).

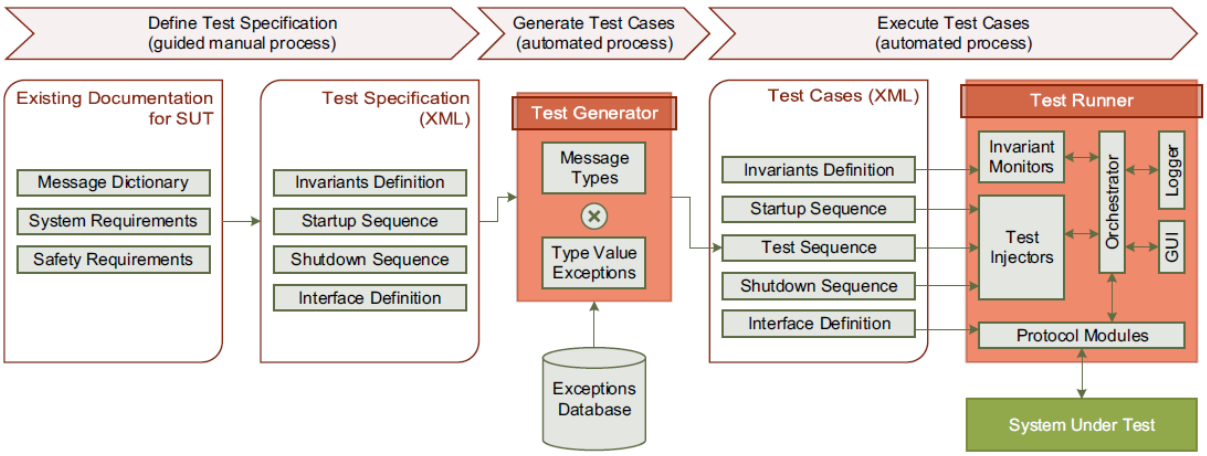

Figure 19 – ASTAA architecture diagram

Intensive simulation can be used as complement (but not a replacement) to physical road testing. It does not only bring speed in the testing activity; it also allows exploring situations that would be difficult/impossible to reach by road testing including situations that would be dangerous for other people). One can introduce a large number of variations into every element of the loop (Figure 20):

- Sensors;
- Perception;
- Vehicle components;
- Vehicle dynamics;
- Driving scenarios;

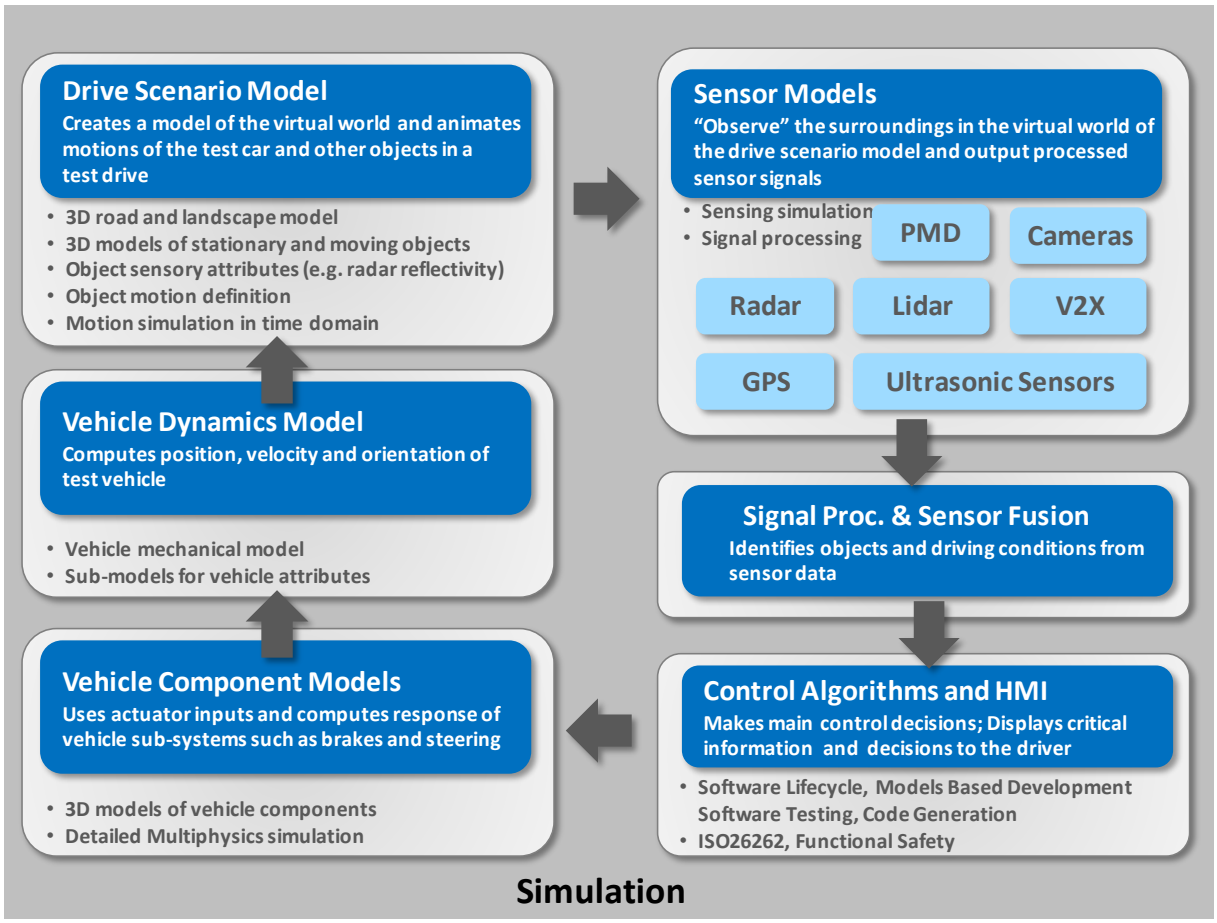

Figure 20 – Integrated simulation of the car elements and the environment

5.4 Tool Qualification

This section addresses the tool qualification requirements for the tools used in the UnCoverCPS approach. All safety standards differentiate offline tools and online tools.

5.4.1 Offline Tools

5.4.1.1 ISO 26262-8 Requirements for Tool Confidence

ISO 26262-8:2018 11.4.5.2 defines requirements for achieving appropriate confidence in offline tools as follows:

The intended usage of the software tool shall be analysed and evaluated to determine:

a) the possibility that a malfunction of a particular software tool can introduce or fail to detect errors in a safety-related item or element being developed. This is expressed by the classes of Tool Impact (TI):

— TI1 shall be selected when there is an argument that there is no such possibility;

— TI2 shall be selected in all other cases.

b) the confidence in measures that prevent the software tool from malfunctioning and producing corresponding erroneous output, or in measures that detect that the software tool has malfunctioned and has produced corresponding erroneous output. This is expressed by the classes of Tool error Detection (TD):

— TD1 shall be selected if there is a high degree of confidence that a malfunction and its corresponding erroneous output will be prevented or detected;

— TD2 shall be selected if there is a medium degree of confidence that a malfunction and its corresponding erroneous output will be prevented or detected;

— TD3 shall be selected in all other cases.

Based on the values determined for the classes of TI and TD (in accordance with 11.4.5.2, or 11.4.5.3), the required software Tool Confidence Level shall be determined according to Table 3 (of ISO 26262-8).

Table 4: ISO 26262-8 Table 3 TCL Determination

		Tool error detection		
		TD1	**TD2**	**TD3**
Tool impact	**TI1**	TCL1	TCL1	TCL1
	TI2	TCL1	TCL2	TCL3

ISO 26262-8 11.4.6.1 defines the following requirements concerning qualification

For the qualification of software tools classified at TCL3, the methods listed in Table 4 shall be applied. For the qualification of software tools classified at TCL2, the methods listed in Table 5 shall be applied. A software tool classified at TCL1 needs no qualification methods.

Table 5: ISO 26262-8 Table 4 Qualification of software tools classified TCL3

Methods		ASIL			
		A	**B**	**C**	**D**
1a	Increased confidence from use in accordance with 11.4.7	++	++	+	+
1b	Evaluation of the tool development process in accordance with 11.4.8	++	++	+	+
1c	Validation of the software tool in accordance with 11.4.9	+	+	++	++
1d	Development in accordance with a safety standard[a]	+	+	++	++
[a] No safety standard is fully applicable to the development of software tools. Instead, a relevant subset of requirements of the safety standard can be selected. EXAMPLE Development of the software tool in accordance with ISO 26262, IEC 61508, EN 50128 or RTCA DO-178C.					

Table 6: ISO 26262-8 Table 5 Qualification of software tools classified TCL2

Methods		ASIL			
		A	B	C	D
1a	Increased confidence from use in accordance with 11.4.7	++	++	++	+
1b	Evaluation of the tool development process in accordance with 11.4.8	++	++	++	+
1c	Validation of the software tool in accordance with 11.4.9	+	+	+	++
1d	Development in accordance with a safety standard[a]	+	+	+	+

[a] No safety standard is fully applicable to the development of software tools. Instead, a relevant subset of requirements of the safety standard can be selected.

EXAMPLE Development of the software tool in accordance with ISO 26262, IEC 61508, EN 50128 or RTCA DO-178C.

5.4.1.2 Analysis of UnCoVerCPS Offline Tools Qualification Requirements

Table 7 provides an classification of UnCoVerCPS Offline Tools.

Table 7: Classification of UnCoverCPS Offline Tools

Tool	Function	TI	TD	TCL
SCADE modeler (ANSYS)	Develop Scade models	2	1	1
SCADE Test (ANSYS)	Dynamic verification and verification of verification of Scade models.	2	2	2
Simplorer modeler (ANSYS)	Develop hybrid Simplorer models	2	1	1
Simplorer simulator (ANSYS)	Simulate hybrid models (can support also simulation from non-Simplorer models if provided in FMU form).	1	1	1
SpaceEx (UJF)	Verify hybrid systems with linear continuous dynamics	2	2	2
CORA (TUM)	Verify hybrid dynamics with nonlinear dynamics.	2	3	3
DMPC-HS (Universität Kassel)	Controller synthesis for model predictive control of non-stochastic systems.	2	3	3
ScenarioMPC (Politecnico di Milano)	Controller synthesis of model predictive control of stochastic systems.	2	3	3
SCADE automatic code generator (ANSYS)	Code generation guaranteeing that the code is a correct implementation of the model	2	3	3
ConfTest (Robert Bosch GmbH)	Conformance testing	2	2	2
FormalSpec (GE Global Research Europe)	Specification formalization	2	2	2

5.4.2 Online tool

Online tools have to be handled as embedded code. This would be the case for SpaceExonl and CORAonl.

Assuming that these elements would be in the planning part, and that the online verification would prevent unsafe control actions from these elements, a safety analysis would typically allocate something such as ASIL B to these elements, but this needs to be instantiated in a real context.

6 Summary and Conclusions

6.1 General considerations

It is not possible to state in in general whether a system developed with the UnCoVerCPS is certifiable for the following reasons:

- Certification concerns a complete system, including for instance the sensors and actuators, which are not addressed in UnCoVerCPS and in this report. A dedicated safety analysis would be required to provide a fully accurate diagnostic;
- Traditional certification frameworks (the SOTIF approach is not part of these) are based on paradigms which rely on fully explicit requirements against which one can perform systematic verification (review, analysis, testing). Autonomous Vehicles usually do not fit such frameworks and so does UnCoVerCPS. The traditional frameworks such as ISO 26262 would essentially address the implementation means;
- UnCoVerCPS addresses a level which is more concerned by the Safety Of the Intended Function (SOTIF) than by the traditional ISO 26262 level. The SOTIF analysis is still something new, which is at the stage of Publicly Available Specification [26]; it is expected to become a standard by end 2020. So, it is too early to have a formal reference for certification of Autonomous Vehicles;
- UnCoVerCPS includes techniques that will not achieve maturity in short term such as true hybrid *online* verification.

6.2 Weak Points

The following issues would make certification of UnCoVerCPS-based systems problematic if these were needed in the safety critical functionalities:

- The low TRL (Technological Readiness Level) of most individual techniques;
- The difficulty of developing, identifying and validating hybrid models;
- The uncertain Worst-Case Execution Time issues of several of these techniques;
- The mixture of technologies with different TRLs and different run-time performance;
- The lack of experience for integration of those techniques.

Automated formal verification is likely to be faced with two types of problems, which will limit its practical usage:

- Cultural: experience shows that it is difficult for many engineers to write down formal properties to verify; they feel more comfortable in expressing the “how” than the formal “what”;
- Technical: even with pure discrete time models, formal verification is sometimes hard to automate; for hybrid models, this is likely to be significantly more difficult;

But since this formal verification is not used for fail-safe trajectories computation, this is not a critical issue.

6.3 Strong Points

There are several important positive points in the UnCoVerCPS approach.

6.3.1 Models and Conformance Checking

The use of models is useful for an explicit, verifiable representation of the system in its environment and of the controller. The fact that it is difficult to identify, validate and simulate efficiently complex hybrid models is inherent to the reality. Not using models would certainly not solve the problem, it would just correspond to poor, implicit and unverifiable models. If models are difficult to exploit, then it may be better to seek reasonable, but explicit and measurable simplifications (e.g. using discrete time approximation) than using no model.

Systematic test of conformance between the models and the behavior of the real system is also a major advance. This is addressed by developing methods that automatically generate critical test cases. In order to achieve conformance, set-based and stochastic uncertainty are included in the models, especially the models describing entities surrounding the considered system.

6.3.2 On-the-fly verification

Deep learning is fashionable. But from a safety perspective, it is good that the core of the UnCoVerCPS principles does not rely on artificial intelligence, in particular deep learning. This is not to say that neural networks should not be used in AVs (we probably cannot live without them for perception), but it is good that the core of trajectory planning and checking relies on more verifiable principles.

Explicit handling of uncertainties with **reachable sets** is a very strong point of this approach. Pre-computing candidate trajectories and use of reachable sets computation contribute to make online WCET acceptable (this needs of course to be confirmed).

6.4 Recommendations

Whilst some of the techniques that are used or developed in UnCoverCPS project will not achieve sufficient maturity in the near term (e.g. true hybrid online simulation, formal verification of non-linear systems), some others may be used in mid-term in particular:

- Online verification based on reachability analysis;
- Qualified auto-coding.

From a safety and certification perspective, the following approach is recommended, to make certification possible:

- Design a control architecture clearly separating performance/comfort functions from safeguarding functions;
- Apply the UnCoVerCPS online verification approach to safeguarding;
- Whilst it is good to define a generic approach, one should not hesitate to use domain-specific models and algorithms to achieve the appropriate degree of performance and provability. For instance, papers on autonomous vehicles and robots trajectories that rely on simple and verifiable physical equations with conservative reachability and decision computations seem to be good candidates for the safeguarding parts [23] [22] [27];
- At software level, use fully mature design and implementation means for the safeguarding and control functions.

7 References

[1] SAE, "Certification considerations for highly integrated or complex aircraft systems, EUROCAE ED-79 and SAE Aerospace Recommended Practice ARP 4754," SAE, 2010.

[2] SAE, "EUROCAE ED135 and SAE ARP 4761, Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment," SAE, 1996.

[3] RTCA, "Software Considerations in Airborne Systems and Equipment Certification DO-178C," RTCA Inc, 2011.

[4] RTCA, "DO-254, Design Assurance Guidance for Airborne Electronic Hardware," RTCA, 2000.

[5] European Cooperation for Space Standardisation, "Space product assurance – Dependability, ECSS-Q-ST-30C," European Cooperation for Space Standardisation, 2009.

[6] European Cooperation for Space Standardisation, "Space product assurance – Safety ECSS-Q-ST-40C," European Cooperation for Space Standardisation, 2009.

[7] European Cooperation for Space Standardisation, "Space product assurance – Software product assurance, ECSS-Q-ST-80C," European Cooperation for Space Standardisation, 2009.

[8] CENELEC, EN 50126 - Railway applications – The specification and demonstration of reliability, availability, maintainability and safety (RAMS), CENELEC, 1999.

[9] CENELEC, EN 50128 Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems, 2011.

[10] CENELEC, "Railway applications – Communications, signalling and processing systems – Safety related electronic systems for signalling EN 50129," 2003.

[11] IEC, IEC 60880 Nuclear power plants Instrumentation and control systems important to safety Software aspects for computer-based systems performing category A functions, IEC, 2006.

[12] IEC, "Nuclear power plants – Instrumentation and control important to safety – Classification of instrumentation and control functions", edition 3.0," 2009.

[13] IEC, "IEC 61513 Nuclear power plants – Instrumentation and control for systems important to safety – General requirements for systems, edition 1.0," IEC, 2001.

[14] IEC, IEC 61508 Functional safety of E/E/PE safety-related systems, IEC, 2010.

[15] "IEC 61511 Functional safety – Safety instrumented systems for the process industry sector," IEC, 2016.

[16] ISO, ISO 26262:2018 Road vehicles – Functional safety, IEC, 2018.

[17] P. Baufreton, J. Blanquart, J. Boulanger, H. Delseny, J. Derrien, J. Gassino, G. Ladier, E. Ledinot, M. Leeman, J. Machrouh, P. Quéré and B. Ricque, "Multi-domain comparison of safety standards," in *ERTS*, Toulouse, 2010.

[18] Z. Pezzementi, T. Tabor, S. Yim, J. K. Chang, B. Drozd, D. Guttendorf, M. Wagner and P. Koopman, "Putting Image Manipulations in Context: Robustness Testing for Safe Perception," in *Robustness Inside-Out Testing (RIOT). NAVAIR 2018-165*, 2018.

[19] NASA, "Certification considerations for adaptive systems, NASA/CR-2015-218702," NASA, 2015.

[20] N. Kalra and S. M. Paddock, "Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?," RAND Corporation, 2016.

[21] S. BAK, D. Chivukula, O. Adekunle, M. Sun, M. Caccamo, M. Caccamo and L. Sha, "The system-level simplex architecture for improved real-time embedded system safety," in *RTAS '09: Proceedings of the 2009 15th IEEE Real-Time and Embedded Technology and Applications Symposium*, Washington, 2009.

[22] S. Magdici and M. Althoff, "Fail-Safe Motion Planning of Autonomous Vehicles," in *ITSC*, 2016.

[23] A. Rizaldi, J. Keinholz, M. Huber, J. Feldle, F. Immler, M. Althoff, E. Hilgendorf and T. Nipkow, "Formalising and Monitoring Traffic Rules for Autonomous Vehicles in Isabelle/HOL," in *IFM*, 2017.

[24] RTCA, "Software Tool Qualification Considerations DO-330," RTCA Inc., 2011.

[25] C. Hutchison, M. Zizyte, P. Lanigan, D. Guttendorf, M. Wagner, C. Le Goues and P. Koopman, "Robustness Testing of Autonomy Software," in *ICSE-SEIP '18: 40th International Conference on Software Engineering*, Gothenburg, Sweden, 2018.

[26] ISO TC 22, "Road vehicles - Safety Of The Intended Functionality - Draft Publicly Available Specification, PAS 21448," ISO (not yet released), 2018.

[27] S. B. Liu, H. Roehm, C. Heinzemann, I. Lutkebohle, J. Oehlerking and M. Althoff, "Provably Safe Motion of Mobile Robots in Human Environments".