PRISSMA Project
Plateforme de Recherche et d'Investissement pour la Sûreté
et la Sécurité de la Mobilité Autonome
04/2021 - 04/2024

[DELIVERABLE 2.6] DEFINITION OF PROCEDURES OF SCENARIO MANAGEMENT AND RESULTS ANALYSIS – INITIAL REPORT

FIRST DELIVERABLE FOR TASK 2.4: MANAGEMENT, SELECTION AND USE OF SCENARIOS, PARAMETERS AND EVALUATION VARIABLES - INITIAL REPORT

Main authors : K. Quintero (IRT SystemX), C. Bohn (IRT SystemX), M. El-Zeenny (IRT SystemX), E. Guasch (Airbus-Protect), J-S Barreiro (ESI), J. Petit (Ansys), D. Gruyer (Univ Eiffel), Sio-Song Ieng (Univ Eiffel), F. Vienne (Univ Eiffel), E. Chateauroux (Transpolis), L. Maisonobe (STRMTG), J-B Horel (INRIA), R. Mateescu (INRIA), G.Durand (AVSimulation), C. Gava (SPHEREA)

Keywords: scenario management, AV scenario-based validation, combinatorial management, AI-based certification.

Abstract. This deliverable provides an initial report on the activities of task 2.4 of the PRISSMA project which should address procedures for scenario management and results analysis with the objective of AI certification in automated vehicles. Current and future activities of the task are addressed as well as both the scientific and organisational challenges. First, the subject of scenario descriptions (from a singleton perspective) is addressed and different approaches are presented from a practical standpoint from the partners in the task. Subsequently, the subject of scenario management is addressed with general methodological recommendations and the current implementation in one possible scenario manager inspired by the input of french and international OEMs and regulatory bodies. The current challenges of transposing descriptive scenarios towards simulation scenarios is then addressed with different views from partners, proposals and points of attention as well as the challenge of the combinatorial management for scenarios. A future version of this deliverable will cover the following activities in the task.

Résumé. Ce livrable fournit un rapport initial sur les activités de la tâche 2.4 du projet PRISSMA qui doit adresser les procédures de gestion de scénarios et analyses de résultats avec l'objectif de certification des systèmes à base d'IA pour le véhicule automatisé. Les activités en cours et à venir sont adressées ainsi que les verrous scientifiques et organisationnels. Le sujet de la description de scénarios d'un point de vue unitaire est abordé selon les différentes perspectives pratiques des partenaires de la tâche. Également, le sujet de la gestion des scénarios est abordé avec des recommandations générales ainsi que l'implémentation actuelle sur un gestionnaire de scénarios inspiré par l'apport de constructeurs et d'organismes de réglementation français et internationaux. Les verrous pour le passage de scénarios descriptifs à scénarios de simulation est adressé avec des différentes visions des partenaires pour la mise en pratique dans PRISSMA dans la tâche, des propositions et des points d'attention ainsi que le challenge de la gestion de la combinatoire. Une future version de ce livrable détaillera les prochaines activités de la tâche.

Contents

1 Introduction **1**
- 1.1 Objectives and Scope of the Task . . . 1
- 1.2 Structure of the Deliverable . . . 1

2 Task Overview and Dependencies **2**
- 2.1 Interdependencies with Other Tasks . . . 2
- 2.2 Current Work Description . . . 4
- 2.3 Challenges . . . 6
  - 2.3.1 Combinatorial Management Challenges . . . 6
  - 2.3.2 Acceptability: Traffic rules and good manners . . . 7
  - 2.3.3 Project Challenges . . . 8

3 Scenario Description **10**
- 3.1 Main definitions and scenario definition . . . 10
- 3.2 Additional definitions and scenario descriptions for POC BuSAD . . . 14
  - 3.2.1 Global description and overview of the ODD for scenario definition . . 14
  - 3.2.2 Static physical environment . . . 15
  - 3.2.3 Dynamic physical environment . . . 17
  - 3.2.4 Condition environment . . . 18
  - 3.2.5 Description of the scene concepts and scene life . . . 20
  - 3.2.6 Scenario description applied to POC BuSAD . . . 25

4 Scenario Management **27**
- 4.1 General Overview . . . 27
- 4.2 Scenario Management Methodology in MOSAR . . . 27
  - 4.2.1 Generic high-level concepts on Scenario Management in MOSAR . . . 28
  - 4.2.2 Scenario Classification . . . 28
  - 4.2.3 Scenario Management from Different Sources in PRISSMA . . . 29
  - 4.2.4 Statistical Analysis . . . 29
  - 4.2.5 MOSAR in the PRISSMA Scope – Potential Evolution . . . 30

5 From Descriptive Scenarios Towards Simulation **31**
- 5.1 From Scenarios in MOSAR towards Simulation Test Cases . . . 31
- 5.2 Implementation of scenario in simulation environment: Open-Drive and OpenScenario . . . 33
- 5.3 3D Environment and Digital Twins for Scenarios . . . 34
- 5.4 Scenario description implementation and use in Pro-SiVIC . . . 35

6 Scenario Combinatorial Management **37**
- 6.1 Strategic choices to manage combinatorial explosion . . . 37
- 6.2 Combinatorial Management: Ansys optiSLang for Process Integration & Scenario Parameter Variation . . . 37

7 Next Steps **40**

List of Figures

1 Interdependencies of task 2.4 . . . 3
2 Approach from high-level AI requirements (Task 1.2) to descriptive scenarios to simulation . . . 4
3 Scenario Storyboard . . . 12
4 Scenario Storyboard Actors' parameters . . . 13
5 The taxonomy proposed in PRISSMA for ODD (work from WP8) . . . 13
6 Global architecture with the scenario management in an upstream position (Source UGE) . . . 14
7 Description of the ODD domains used for the scenario description (source UGE) 15
8 Taxonomy for the environment, ego-vehicle, and event management in the building of evaluation and validation scenarios (source UGE). . . . 16
9 The sub domains and components of the static physical environment (source UGE) . . . 17
10 The sub domains and components of the dynamic physical environment (source UGE) . . . 18
11 The sub domains and components of the environment conditions (source UGE) 19
12 Modeling of categories of disturbers having an impact on the performances of sensors and data quality and involved in the domain dedicated to condition environment (source UGE). . . . 19
13 Description of a scene life and parameter of state for dynamic actors and ego-vehicle (source UGE) . . . 20
14 Description of the main parameters for the configuration of both the actors and the ego-vehicle state (source UGE) . . . 21
15 Description of the different states and modes for actions, behaviors, and activities. (Source UGE) . . . 23
16 Diagram of the different relations, links, and interactions between the components and the concepts handled and managed in a scene (source UGE). . 24
17 Diagram of the different domains and concepts involved in the scene description (source UGE) . . . 25
18 Bus station automated desert service with 6 possible scenes (source UGE). . . . 25
19 Example of sentences for static physical environment (source UGE). . . . 26
20 Container Management in MOSAR . . . 28
21 MOSAR - Scenario Manager . . . 29
22 MOSAR - Scenarios occurrences analysis tool . . . 30
23 Parameter annotation in MOSAR - Aiming toward simulation - parameters for cutter actor in a cut-in scenario . . . 32
24 Parameter annotation in MOSAR in a cut-in scenario - Aiming toward simulation 32
25 OpenDRIVE road network description . . . 33
26 OpenSCENARIO scenario example . . . 34
27 Overview of scenario management with the different modules allowing to generate a set of scenes with a generic modelling/language. In Pro-SiVIC, a specific module called sivicScenarioManager will be used in order to consider the scene and scenario provided by third party software and will convert these rules and sentences in a set of Pro-SiVIC set of commands. (Source UGE) . . . 35
28 Functional architecture of event mechanism implemented in Pro-SiVIC platform 36

29 Principles to be followed to derive scenarios relevant for the ODD of the ADS – Regulation EU 2022/1426 ([1]) – annex III part 1 appendix 1 40

List of Tables

1 Example of a high-level AI requirement processing in order to guide scenario management 5

1 Introduction

This deliverable is issued from the work in task 2.4, titled ‘Management, Selection and Use of Scenarios, Parameters and Evaluation Variables’, as part of Work Package 2 of the PRISSMA project.

As a general overview, Work Package 2 (WP2), titled “Pertinent Uses of Simulation – Possibilities and Limitations” addresses the subject of the evaluation of AI-based systems for automated transportation through simulation platforms. This includes providing recommendations for the generic processes of evaluating the simulation tools, defining test plans, guaranteeing interoperability of sub-modules and instantiating such methodologies through Proofs of Concept (POC) for such systems in simulation environments.

1.1 Objectives and Scope of the Task

Task 2.4, being included within the scope of WP2, addresses the challenges related to scenario management in order to ensure the proper use of pertinent parameters and variables allowing to evaluate IA-based systems on the grounds of simulation. This task aims at providing a methodology to manage scenarios so that the main properties of the system under test (SUT) are well represented, the combinatorial explosion of cases to test is managed coherently and therefore the simulation teams receive a first scenario-based proposal that they can enrich with specific capabilities. The aim is that the work can be generic and applicable to any simulation tool or platform.

Task 2.4 is a backbone task which aims at incorporating best practices oriented toward certification of AI systems based on the state of the art for scenario management processes. It is tightly connected to task 1.2 in WP1 which addresses the scenario generation process for the project.

This task is led by IRT SystemX and co-led by INRIA. The remaining contributors include: Université Gustave Eiffel, LNE, APSYS, ANSYS, AVS, ESI, STRMTG, SPHEREA, and Transpolis.

1.2 Structure of the Deliverable

This deliverable is the first iteration for task 2.4 and synthesizes the work up to the M18 milestone of the PRISSMA project. A new version of this document will present the results of the following iteration of activities of the task.

In this deliverable, section 2 provides an overview of the task, its interdependencies with other tasks in the project, the description of the work carried out and the challenges encountered.

The following sections detail key aspects of the task starting by the basis of scenario description on section 3, to then enlarge the scope to scenario management on section 4. Section 5 addresses the process of going from descriptive scenarios to simulation and finally section 6 focuses on the challenge of the combinatorial explosion when managing scenarios. In this sense, from section 4 to section 6, this deliverable presents the process of going from a scenario description to the management of many scenarios to the simulation of test cases that derive from the combinatorial manipulation of the relevant parameters for the defined scenarios and the system under test.

Finally, the section 7 presents the next steps for the task.

2 Task Overview and Dependencies

In general, task 2.4 should propose procedures and management mechanisms in order to select and use scenarios as well as their associated parameters and variables, this in the view of certification of AI-based systems. Inter-dependencies of this task are detailed below in sub-section 3.2; however, broadly speaking, there is a link between this task and WP1 which should define validation requirements for the systems under test.

The task's direction has proposed then a preliminary workplan based on the high-level requirements that are to be defined on WP1 with the goal of deriving from these requirements the aspects related to scenario management that should be addressed in this task. This has been the foundation of the work carried out in the task to this day and in the following sub-sections the interdependencies of the task are detailed, as well as the workplan, current status and both scientific and technical challenges encountered.

2.1 Interdependencies with Other Tasks

As explained in the Deliverable L1.4, the NATM (New Assessment/Test Method for automated driving) is a text defined by the VMAD, an UNECE working group, see [2]. This method is to become the reference for Automated driving validation and regulations as the R157 or the UE ADS act (EU 2022/1426 [1]) are driven by this work. It is based on 6 pillars:

- (a) A scenario catalogue
- (b) Simulation / virtual testing
- (c) Track testing
- (d) Real world testing
- (e) Audit / assessment
- (f) In-service monitoring and reporting.

Roughly speaking, PRISSMA main WPs are based on the NATM multi-pillar approach. Pillar ”(b) Simulation / virtual testing” corresponds to PRISSMA WP2, pillar ”(c) track testing” to WP3, pillar ”(d) Real world testing” to WP4, pillar ”(e) Audit / assessment” should be treated in WP6 and pillar (f) in WP7.

The pillar ”(a) A scenario catalogue”, is not treated as a WP in PRISSMA. However, the definition of this catalogue or database and its management are a transversal activity that has consequences on the 5 other pillars. L1.4 gives some guidelines to build the catalogue and clauses to define the use of the scenarios according to the pillar, see [3]. However, L1.4 recommendations are very theoretical, and the requirements are very high level

Interdependencies for task 2.4 are displayed on the Figure 1. In general, task 2.4 should provide best practices in order to manage scenarios to ensure validation of AI-based systems. For this purpose, and in a chronological setup, this task should work on the basis of the requirements for testing and audit defined on task 1.2 of WP1 (“Objectives Assigned to AI-based Systems”). This task specifically includes a working group called “scenario generation” which should provide the basis of the scenarios to be managed by task 2.4. In the scope of scenario generation, at this stage of the project, deliverable L1.4 section 4.2.1 [3] is providing a preliminary approach based on recently published French regulations that should serve as a reference to generate scenarios in the PRISSMA project and provide the necessary input for task 2.4, see [4] for the official document French General Administration for Infrastructure, Transport and Mobility - DGITM . To be detailed in a further version of this deliverable once the DGITM's documentation will be integrated in the overall scope of PRISSMA.

Task 2.4 focuses on simulation and therefore it should provide and receive feedback from WP3 which addresses testing under controlled conditions whether it is in the sense of complementarity of simulation versus controlled testing of AI-systems or in the sense of comparison to ensure representativity. This very same interaction holds between task 2.4 and WP4 which in turn addresses testing and validation of the same systems in real driving conditions. Finally, task 2.4, depends on the taxonomy that should be defined in WP8 in order to describe the ODD for such systems and therefore enabling the description of the scenarios recommended to be used for evaluation and validation. WP8 should also provide use cases allowing to instantiate the methodology proposed in the task.

Otherwise stated: task 2.4 interfaces with other tasks in the project as follows:

Inputs: WP1 (scenario generation, validation requirements), WP8 (taxonomy and use cases), WP3/WP4 (for test cases that should be compared with results in simulation)

Outputs: WP3/WP4 (for test cases in simulation that should be tested on driving conditions either real or in controlled environments)

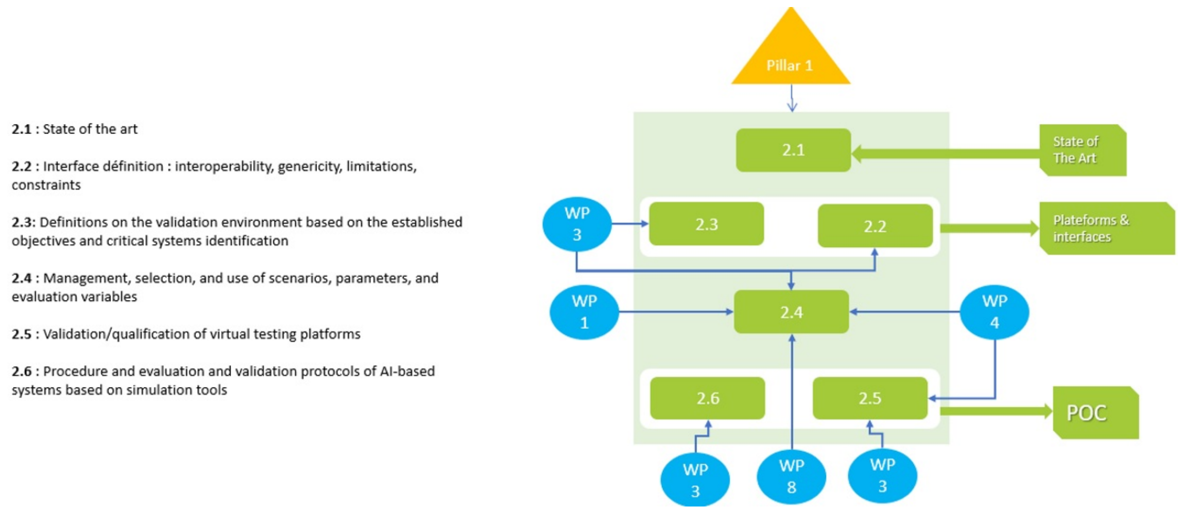

Figure 1: Interdependencies of task 2.4

Required concrete inputs are:

- Generated high-level scenarios coming from task 1.2 based on identified requirements for AI-based systems.

- ODD taxonomy in order to refine scenarios (WP8)

- List of requirements (Task 1.2) regarding types of AI: that should give guidelines on which specifications not to miss on the validation process and therefore the means of exploiting scenario catalogues.

This should narrow the search space and allow focus on specific scenarios dealing with limitations of the AI bricks under validation. Analyzing and managing scenarios that focus on the boundaries of functional states of the evaluated system allows to better assess safety since functional safety should be assessed by the proprietary of such system.

In this sense, the SOTIF (Safety of the Intended Functionality) standard [5], provides guidelines for the process and the elements necessary to be taken into consideration in order to implement a scenario driven safety assessment methodology. Even though it is not oriented exclusively on AI-based systems, it applies perfectly on these. In this sense the necessary inputs for task 2.4 are:

- System description
- Catalogue on triggering conditions
- System's ODD

More details on the SOTIF process and requirements can be found in PRISSMA's deliverable L1.4: Test and Audit Requirements – Initial Report [3], specifically on section AI requirements, criteria and missions.

2.2 Current Work Description

The workplan proposal for the task has been to take as a starting point the high-level requirements stemming from WP1 and to try to derive, from these requirements, information that could be related to scenario management and that could guide the use of these scenarios in order to ensure proper certification procedures based on simulation test cases.

One main contribution in the scope of the project and in coherence with the work of task 2.4 is the synchronisation of WP8 and task 2.4 regarding the description of the ODD. The used taxonomy is indeed the one that shall be used to describe the scenarios when being generated (task 1.2) and managed (task 2.4). This taxonomy (see Figure 5) is the result of the work of WP8 and it includes the input from OEMs as well as regulations requirements in the french and the international scope since it is aligned with the one used in MOSAR.

The overall process is depicted on Figure 1 within the simulation scope, and the work carried out to this day covers the processing of high-level requirements in order to obtain pertinent information for scenario generation and description (Task 1.2) and scenario management (Task 2.4). As a reminder, crucial related subjects (not depicted on Figure 2) are *scenario generation* which is addressed on PRISSMA's deliverable [3] and *combinatorial management* which is to be addressed in the following phase of task 2.4 and for which preliminary discussions are presented in this document in section 2.3.1 regarding the challenges of the project.

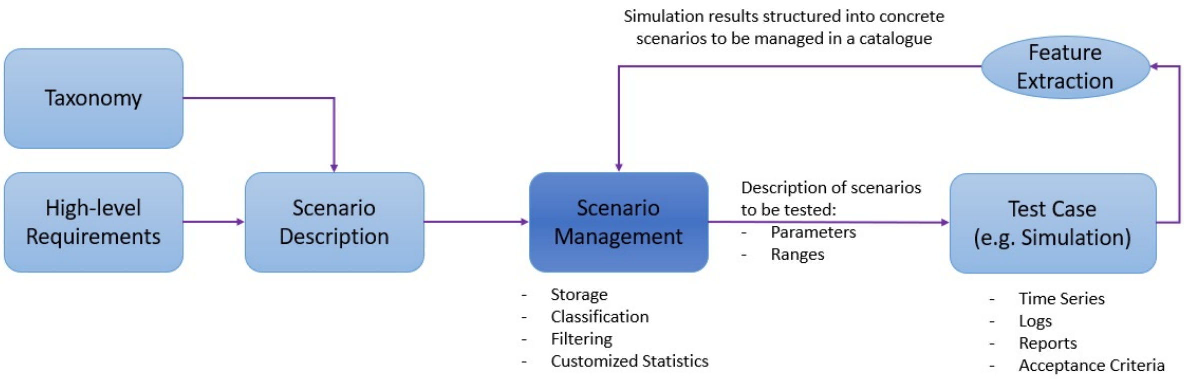

Figure 2: Approach from high-level AI requirements (Task 1.2) to descriptive scenarios to simulation

Regarding the processing of high-level requirements, for the entire list of contributors of the task, and based on the total number of high-level requirements provided by WP1, a subset of requirements was assigned to each contributor ac-cording to the number of PM assigned for the task.

For each requirement, a pre-filtering was made based on the pertinence of the requirement to scenario-related activities, i.e. generation, combinatorial reduction, among others. The requirements that were considered as non-relevant for scenario activities were however kept in the distribution so that other contributors could make a revision and modification if necessary. The information that was asked to contributors to be declined based for each provided requirement was:

- Physical components (impacted by the requirement)
- Software components (impacted by the requirement)
- Selected AI types (to be certified and related to the requirement)
- Scenario Creation (Yes/No if the requirement is used to create new scenarios)
- Combinatorial Reduction (Yes/No if the requirement can be used to reduce the combinatorial explosion of parameters when managing scenarios)
- Dependency on other requirements (to explicit these dependencies)
- Variables/Parameters (to explicit the ones that stem from the requirement)
- AI Influence Factors
- Validation Criteria for the Requirement
- Refinement by (if the requirement is too general to be treated, specify the working group in the project that should refine or detail it into more specific requirements to be analyzed in order to manage scenarios)

An example of this first phase of requirement processing is presented on Table 1.

High-level AI Requirement	The degraded mode operating domain of the AI functionality must be characterized
Physical components	sensors
Software components	Perception / decision
Selected AI types	All
Scenario Creation	All critical scenarios
Combinatorial Reduction	No
Dependency on other requirements	Definition of corner cases or outside ODD
Variables/Parameters	-
AI Influence Factors	-
Validation Criteria for the Requirement	Coherent characterizations
Refinement by	Work group: Scenario Generation (T1.2)

Table 1: Example of a high-level AI requirement processing in order to guide scenario management

For this example, the high-level AI requirement is that the degraded mode operating domain of the AI functionality must be characterized. After identifying the components and AI types impacted by this requirement, it is stated that the requirement indeed induces scenario creation, namely all critical scenarios, and that it does not allow reducing the combinatorial expansion. Such requirement depends on other requirements such as the definition of corner cases or what is considered outside the ODD. Specific variables, parameters and AI influence factors cannot be identified at this point, the validation criteria are to have coherent characterizations and, finally and most importantly, this requirement needs refining by the working group: Scenario Generation in task 1.2. The requirement has been detailed as much as possible for the needs of task 2.4 and, as planned by the roadmap of the project, it should now be refined by a specific working group, in this case: Scenario Generation in task 1.2.

This example illustrates one of the main challenges of the current work of the task, which is the fact that requirements are at this point very high-level and therefore it is not trivial to propose a generic strategy for scenario management without considering the requirements that are associated to the specific System Under Test (SUT). This challenge has been confirmed by the other contributors based on the requirements that were assigned to them where, intuitively, it is quite difficult to specify best practices on what and how to test a specific system without knowing the system in itself. The counterpart is the need to be as generic as possible and to provide a methodology suitable to be adapted to various systems. For this reason the following step is to receive the refinement performed by all the different working groups and to propose a first draft of a methodology for scenario management based on more specific requirements and the necessary hypotheses in the absence of a SUT.

In general, Figure 2 provides a high-level view of the operational workflow which should lead from the system's requirements to scenario description and management, based on a defined taxonomy. This procedure allows reaching an objective which is to generate and manage the test cases necessary in simulation in order to allow the application of certification procedures. These test cases must then reflect reliably and sufficiently the system's behavior and they are obtained though the specification of:

- The AI-related requirements and limitations,
- the system's expected behavior,
- the specified ODD
- safety requirements (according to regulatory bodies or norms)

All of these sources of information provide the means for the construction of pertinent simulation campaigns that should ensure the representativity of the system's behavior in order to be certified.

In the absence of a SUT, a generic process can be proposed in order to provide the preliminary means in order to carry out certification. This process will entail obvious limitations that will need to be addressed when instantiating it on one specific system.

2.3 Challenges

2.3.1 Combinatorial Management Challenges

The combinatorial management has 2 main challenges:

- **Need for exhaustivity in simulation vs combinatorial expansion**

  Ideally, in order to establish the correct response of an ego vehicle in all circumstances, one should simulate the ego vehicle behaviour in all possible configurations (scene map, trajectory of the ego vehicle, other static and mobile obstacles, external conditions, etc.) [2]. However, the space of possible configurations increases rapidly with the scene complexity (number of obstacles and their trajectories) and the various other parameters, and this combinatorial explosion makes the exhaustive simulation prohibitively costly in terms of time and computing resources.

Therefore, in practice a (sufficiently large) set of simulation scenarios are selected, with the goal of covering all relevant situations in which the behaviour of the ego vehicle should be validated [6]. The challenge is to determine the appropriate trade-off between exhaustivity and combinatorial explosion, taking into account the cost of the simulation campaign and the coverage of the relevant situations by the set of selected simulation scenarios.

- **Need for modelling scenarios in an *intermediate level* in a structured catalogue.**

  This challenge refers to the fact that in a proper catalogue, scenarios should be structured, defined, and categorized in a manner that such catalogue can be constructed by collaborators of different backgrounds compatible but not restricted to simulation. As an example, scenarios can be defined by automotive research engineers, project managers, test trial coordinators and any other professional managing driving scenarios without having to master simulation specificities. Proper scenario catalogues can then provide scenario descriptions based on relevant metadata that can be used to evaluate the systems statistically and for which then the proper tools should be available in order to filter, categorize and characterize scenario datasets according to the criteria necessary by the specific user.

  As an example: test campaigns can be designed to challenge the SUT (and therefore the related AI-based bricks) to manage scenarios where road users unexpectedly enter the ego vehicle's path. For given use cases, different road users can be considered as the intruder for this example. Also, the distances, trajectories and velocities at which the intruder engages in the ego vehicle's path can vary as well as environmental conditions that could influence the response of the AI-based systems. From a high-level stand point, scenarios can be defined in a scenario catalogue by abstracting the situation to only a few scenes and defining the ranges of variation for the parameters of interest. After this such high-level descriptive scenarios are registered in the catalogue by users of various profiles, the test campaign performed by simulation specialists can produce the resulting hundreds of concrete scenarios that can be stored and categorized in the catalogue. This scenario dataset can now be exploited by various users including certification organisms in order to evaluate the configurations of interest that should justify the safety of the system under test. This includes for example performing statistics of the system's response when dealing with VRU's at rush hour at dusk and with heavy rain conditions, or any other condition judged as pertinent given the system's ODD.

2.3.2 Acceptability: Traffic rules and good manners

Within all the requirements the automated driving systems shall fulfil, to respect all the traffics rules and regulations is still a challenge for the developers and the verification and validation teams. This challenge is induced by the difficulty to deal with the semantic gap between software code and safety-related specification. As explained by [7], the current validations methods (using simulation, track or road testing) only target a limited set of driving rules related to basic safety requirements such as collision avoidance and safe distance on urban roads or highways. More complicated driving rules including on-road cross-walks, four-way stop signs, and intersections with traffic lights are ignored by general AV (Autonomous Vehicles) testing techniques. They referred to scenario-dependent driving rules because the expected behaviour of an AV heavily depends on its location as well as surrounding circumstance. [7] proposed a framework named AV Checker to identify violations of scenario-dependent rules. This new tool is American rule-based, but its principles could be adapted to European and specifically French rules, or at least give some clues for PRISSMA work.

On the other hand, the public acceptance and adoption of ADS should also strongly rely on the ability of ADS to adapt its behaviour and to consider “good manners”. These rules are implicit and depend on regional driving habits or common good behaviours, and they are different from the legal traffic rules. For example, if an automated vehicle stops 2m after a car that was about to reverse into a parking space, the AV will prevent the driver from doing its intended manoeuvre, creating a problem in a scenario that does not present any safety critical issue.

This challenge is also about translating very numerous implicit rules into software codes. It may also represent a special challenge for AI based on data-driven decision making. The set of data used for the code development shall be large enough and geographically specific enough to show the functioning of these rules.

The solution will rely on the contribution of ergonomists and human factor experts to define the behavioural adaptation.

The NATM introduces a validation process based on scenario databases. In a near future, it may be important to develop scenario databases dedicated to the validation of traffic rules and “good manners”. These scenario databases shall be used to validate the ADS behaviour especially during the simulation process because these scenarios may include numerous other road users and complex infrastructure configuration which are difficult and costly to reproduce on tracks and long and hazardous to observe in real word tests. However, implementing these scenarios in virtual tools also presents the challenge of assessing the quality of the ADS response. The adequacy and the acceptability of the system behaviour may be subjective matters complicated to implement in a virtual tool chain.

2.3.3 Project Challenges

- Provide a generic methodology with a scenario design top-down approach which allow to adapt the evaluation and validation process as a specific case of a generic framework.
- In the framework of the POC developed on the Satory’s test track, a full mobility service will be addressed with specific situation.
- The different layers and stages in the processing resulting in the generation of the scene description and the scenarios are presented in section 3.2.
- In this modelling, a first stage is to provide a taxonomy and an ontology of the different layers addressed (ODD, OD, OEDR, Events, among others).
- PRISSMA is dedicated to AI based system/application/algorithm evaluation and validation. In this framework, it is necessary to define the parameters, variables, events, conditions, and configurations impacting the reliability, robustness, and operation of systems and components used for the building of automated vehicle services.

- PRISSMA must consider the cyber-security on communication and IA based system. The scenario definition needs to address in a generic way this requirement.

3 Scenario Description

3.1 Main definitions and scenario definition

The purpose of this section is to provide the grounds for the description of a single scenario and allow for the following section to illustrate the scenario management process based on the description of a single scenario.

In the following, several approaches are provided regarding the description of a single scenario to ensure as much genericity as possible in task 2.4. It is however reminded that the aim of the task is not to prescribe regulation on the definition of scenarios or to state how they should be generated, this is the work of WP1 in PRISSMA (including the working group on scenario generation), see [3] for an initial report on test and audit requirements.

A descriptive scenario is a described situation where one way of describing it is through successive keyframes that define the situation. It contains information about the actors dealing with dynamic and static elements, the infrastructure carrying information about the road section and scenes carrying information about the manoeuvres that construct a storyboard. The latter, holds information such as the lateral and longitudinal position, velocity, orientation, and the actions that each actor has in each step of this described situation. The environment (weather conditions, temperature, etc.), traffic, and road state are also part of the storyboard.

A descriptive scenario of a simple situation going from text and few parameters can be transformed into a more thorough descriptive scenario depending on the available information. The more known information about the situation, i.e. actors, infrastructure, environment, manoeuvres, etc., the more scenarios can be exhaustive.

There are several approaches for AV safety assessment. Among them, the scenario-based approach is a promising one. As the number of real-world traffic situations is infinite, it is not possible to assess AV in all possible situations. However, the scenario-based approach allows to cover a wide range of existing situations and to select the most challenging ones to assess AI-based functionalities.

A scenario is a sequence of actions and events. Following are some definitions to clarify the terms scenario and event, introduced in [8].

Definition 1 (Scenario). A scenario is a quantitative description of the relevant actors, characteristics and activities of a situation. It can characterise the actors of the situation and reach the point of describing the goals of the ego vehicle(s), the static environment, the dynamic environment, and all events that are relevant within the time interval be-tween the first and the last relevant event.

Definition 2 (Action and Event). An action is an atomic behavior performed by an actor in a scene. An event corresponds to a moment at which a mode transition occurs or a system reaches a specified threshold, where the former can be induced by both internal and external causes. An event can be seen as an occurrence of a specific action with specific value at a particular point of time. An event can be represented by spatial, temporal, or semantic values involving one or several scene actors. An actor is an element of the scene acting on its own behalf.

Scenarios can be described at different levels:

- **Functional scenarios**: verbal description

- **Logical scenarios**: parameter ranges
- **Concrete scenarios**: exact parameters values

These 3 types of scenarios (functional, logical and concrete) can be obtained from a knowledge-base, made by experts, or from data-driven approach (for logical and concrete scenarios).

Functional scenarios include operating scenarios on a semantic level. The entities of the domain and the relations of those entities are described via a linguistic scenario notation. The scenarios are consistent. The vocabulary used for the description of functional scenarios is specific for the use case and the domain and can feature different levels of detail.

Logical scenarios include operating scenarios on a state space level. Logical scenarios rep-resent the entities and the relations of those entities with the help of parameter ranges in the state space. The parameter ranges can optionally be specified with probability distributions. Additionally, the relations of the parameter ranges can optionally be specified with the help of correlations or numeric conditions. A logical scenario includes a formal notation of the scenario.

Concrete scenarios distinctly depict operating scenarios on a state space level. Concrete scenarios represent entities and the relations of those entities with the help of concrete values for each parameter in the state space.

Some work also focus on defining scenarios categories, to qualitatively describe scenarios [8], [9].

Definition 3 (Scenario category). A scenario category is a qualitative description of the relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, and the dynamic environment.

Many tools allow describing and managing scenarios on the bases of the aforementioned description. In the following, an example in the tool MOSAR Scenario Manager illustrates a basic example for a scenario including the notion of the storyboard, infrastructure, actors and their dynamics and manoeuvres.

A storyboard of a scenario contains an initial scene, a final scene, and keyframes in between to reconstruct the relevant moments that form such scenario. However, between the keyframes the continuous data of the actors is unknown. Figure 3 shows an example of the format of a scenario storyboard in one tool allowing to manage scenario catalogues: MOSAR Scenario Manager.

Definition 4 (Situation). A situation is the entirety of circumstances, which are to be considered for the selection of an appropriate behavior pattern at a particular point of time. It entails all relevant conditions, options and determinants for behavior. A situation is derived from the scene by an information selection and augmentation process based on transient (e.g. mission-specific) as well as permanent goals and values. Hence, a situation is always subjective by representing an element's point of view.

Definition 5 (Scene). A scene describes a snapshot of the environment including the scenery and dynamic elements, as well as all actors' and observers' self-representations, and the relationships among those entities. Only a scene representation in a simulated world can be all-encompassing (objective scene, ground truth). In the real world it is incomplete, incorrect, uncertain, and from one or several observers' points of view (subjective scene).

Definition 6 (Scenery). The scenery subsumes all geospatially stationary aspects of the scene. This entails metric, semantic, and topological information about roads and all their components like lanes, lane markings, road surfaces, or the roads' domain types. Moreover, this subsumes information about conflict areas between lanes as well as information about their interconnections, for example, at intersections. Apart from the before mentioned environment conditions, the scenery also includes stationary elements like houses, fences, curbs, trees, traffic lights, or traffic signs.

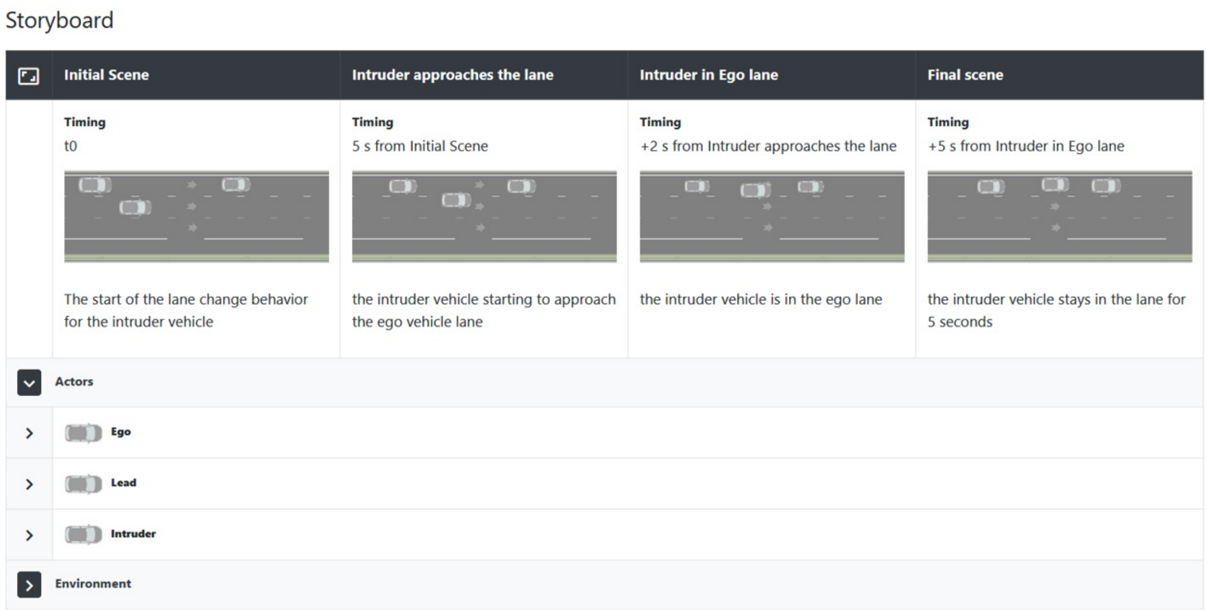

Figure 3: Scenario Storyboard

Moreover, the figure 4 shows the potential parameters that are to be assigned for each select-ed actor for this specific scenario. Values can be described as abstract, as a range, or as fixed values depending on the scenario description level (functional, logical or concrete). Manoeuvres and equipment of each actor can be structured at this stage of the scenario description.

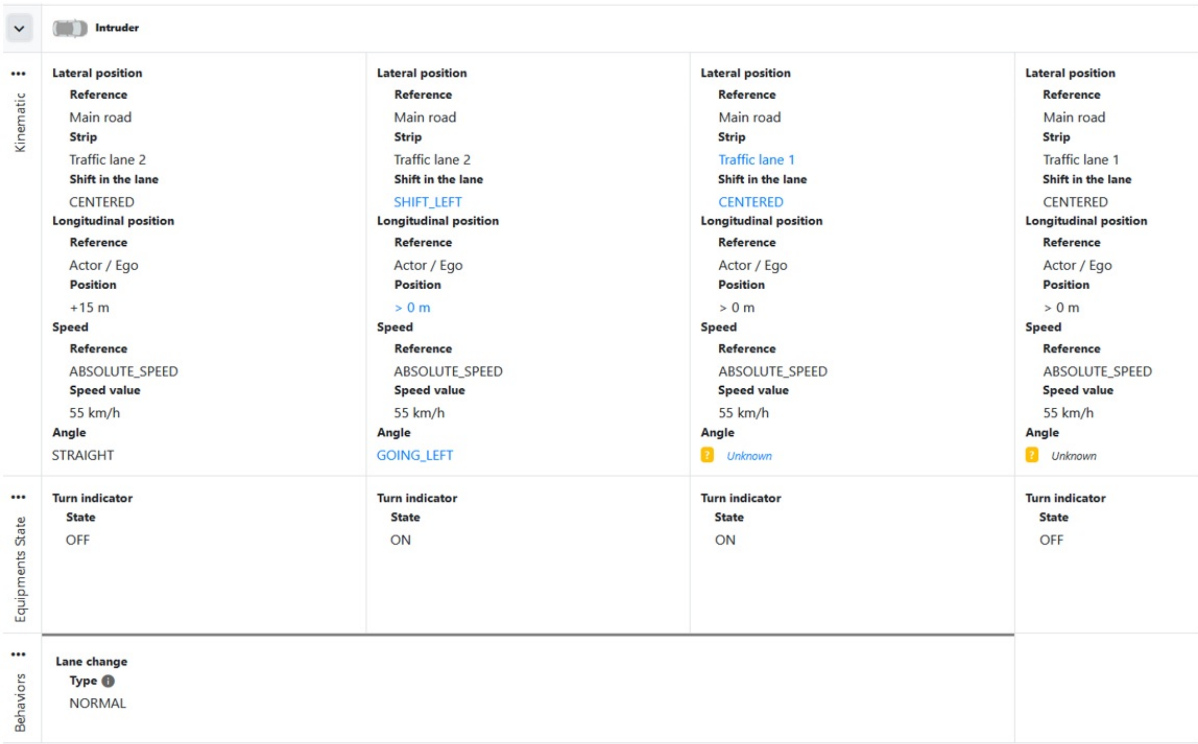

Figure 4: Scenario Storyboard Actors' parameters

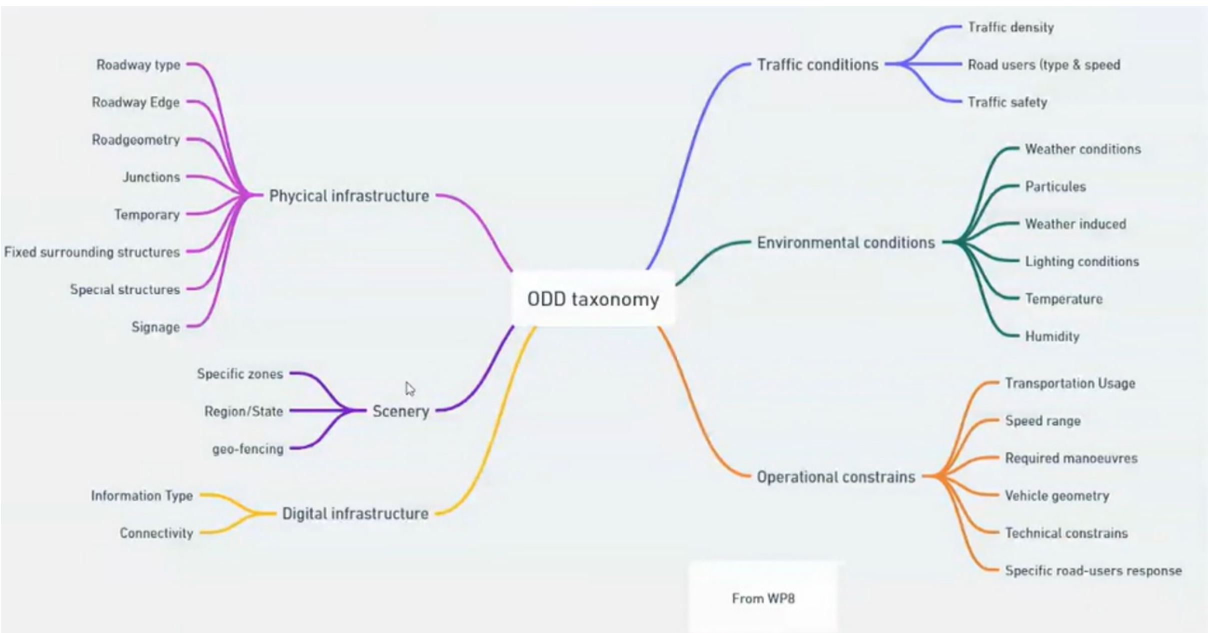

Figure 5: The taxonomy proposed in PRISSMA for ODD (work from WP8)

3.2 Additional definitions and scenario descriptions for POC BuSAD

In the POC 1 dedicated to the Bus Station Automated Desert (called BuSAD), the scenario description proposed by University Gustave Eiffel is based partially on the WP8 taxonomy but with a more accurate definition of concepts based on static elements, dynamic actors, and dynamic aspect handle in the scene. The global architecture proposed by Univ. Eiffel is presented in the figure 6. The scenario description and implementation is addressed in the upstream part dedicated to requirement, ODD, use cases, and scenario definition.

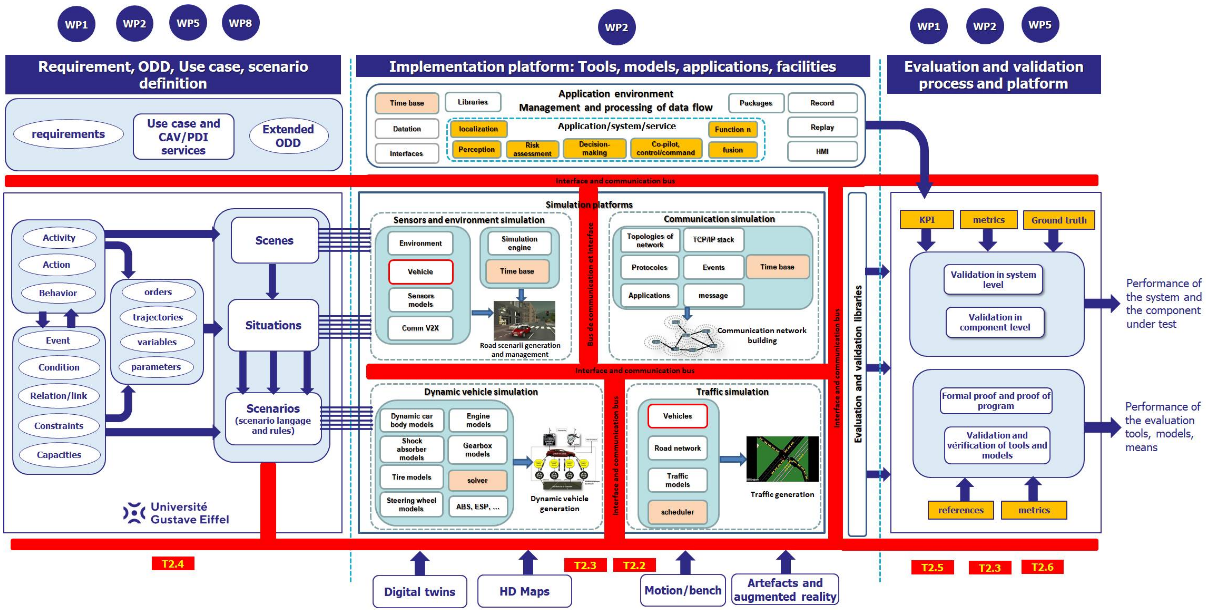

Figure 6: Global architecture with the scenario management in an upstream position (Source UGE)

3.2.1 Global description and overview of the ODD for scenario definition

In the POC 1, a scenario is seen as a set of scenes with specific initial and final states and dynamic actions sharing each scene and allowing to move from a current situation/configuration to another one. A scenario and the set of scenes involved in the scenario are made from the use of a taxonomy and an ontology modelling the environment of the automated mobility service. By extension, the systems of systems and the components involved in the mobility service under test. In this context, the proposed ODD will be defined using 3 domains:

- The static physical environment
- The environment conditions
- The dynamic physical environment involving the ego-vehicle under test

The proposed domain organisation is presented in the figure 7. In this modelling, the 3 main domains are shared in sub domains. Comparing this modelling with the ODD proposed in the WP8, it is easy to see the correlation between both. The right part of the WP8's ODD (figure 5) is represented in the static physical environment. The environment conditions are the same in both representations. The traffic conditions and operational constraints are in the dynamic physical environment.

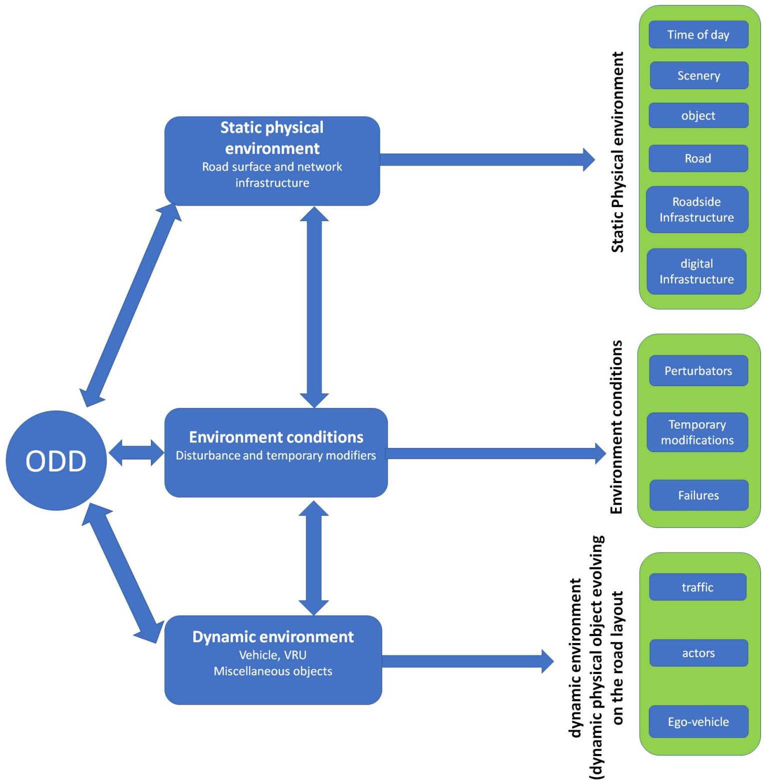

Figure 7: Description of the ODD domains used for the scenario description (source UGE)

3.2.2 Static physical environment

For the static physical environment, the different sub domains are:

- The time of day
- The scenery
- The static objects
- The road
- The roadside infrastructure
- The digital infrastructure

The figure 8 defines a part of the component for each sub domain of the static physical environment.

Figure 8: Taxonomy for the environment, ego-vehicle, and event management in the building of evaluation and validation scenarios (source UGE).

In short, the static physical environment allows to describe the components involved in a static configuration of the environment and corresponds to the set of configurable components and elements that sets up the static skeleton of a driving scene. These elements include the elements/components of the 2 first layers of the environment modeling presented in the figure 8 (adaptation and improvement of the ontology proposed in the PEGASUS project). The different components and elements of this static physical environment are presented in the figure 9 but this overview is not an exhaustive modelling.

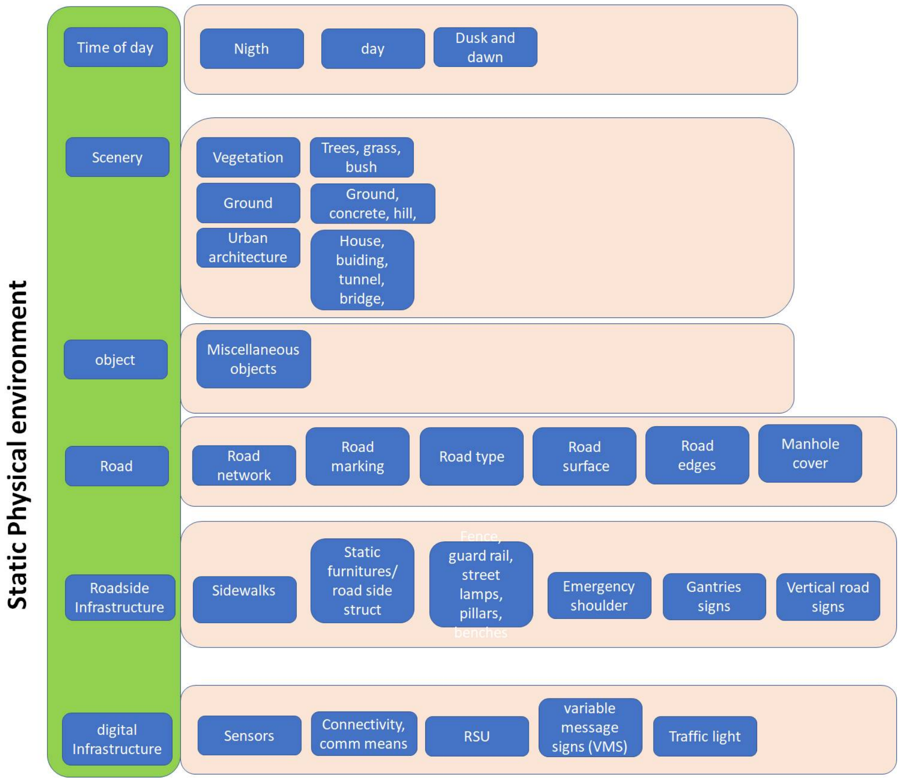

Figure 9: The sub domains and components of the static physical environment (source UGE)

3.2.3 Dynamic physical environment

The dynamic physical environment is the set of elements/components/objects with a dynamic modelling evolving during the simulation with their own behaviors. Dynamic components include multiple surrounding vehicles, VRU, controllable miscellaneous objects as balls. This domain is shared in 3 sub domains: the traffic manager for the management of dense and complex traffic configurations without complex modeling of the object dynamics. For instance, each vehicle in this part could use a simple driving model like IDM. The second sub domain is related to the physical complex actors and system on the scenario. The last sub domain is dedicated to the ego-vehicle. This sub domain is more complex and involves the different levels of components present in the real ego-vehicle. In figure 8, this sub domain is represented by the 4 level of the ego-vehicle and embedded system definition. In the scenario ODD description, this domain is built from the components enumerated in the figure 10.

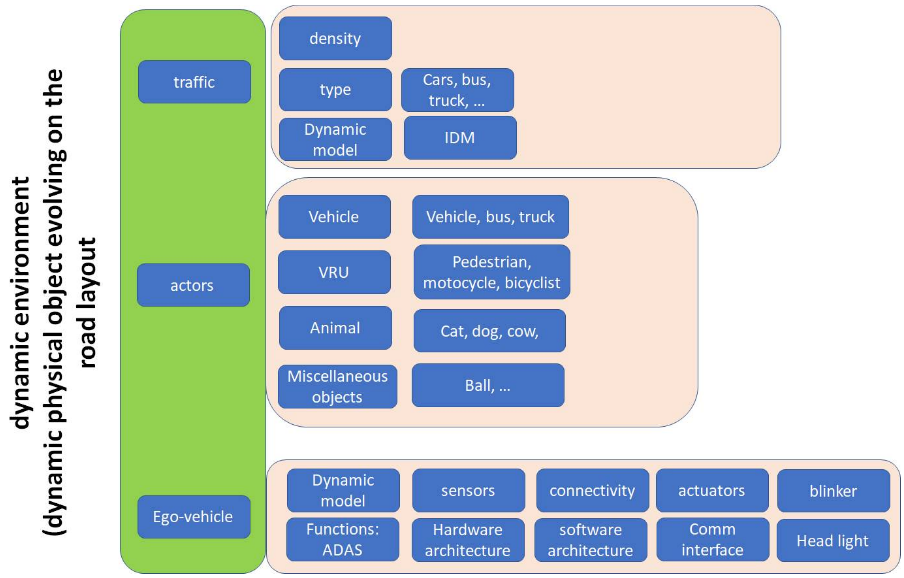

Figure 10: The sub domains and components of the dynamic physical environment (source UGE)

3.2.4 Condition environment

The third domain is dedicated to the condition impacting the current and future state of a scene and of the scenario (figure 11). This domain is composed by 3 sub domains with disturbers, temporary modification of the scenery and static physical environment, and the failures. The first sub-domain is share in a set of sub classes of disturbers. In the figure 9, only a sub part is proposed with weather conditions, illumination, and presence of particle. The second sub-domain is more focused on the road surface and roadside modification (objects, geometric aspect, source of information, . . .). For instance, these modifications can be generated by roadworks. The last sub domain is about failures on a component of the dynamic physical environment. For instance, cyber-attacks could occur on the perception aspect (capability of the perception) with the modification of road signs. A failure could also occur on one or several elements of the ego-vehicle. generating degraded and adverse conditions for the ego-vehicle operating and the system under test. About the different classes of disturbances impacting the sensors, a first proposal is done with the figure 10. This modelling will be enhanced in the second part of the PRISSMA project. Moreover, cyber-attack will be soon more detailed with the inputs of the WP5 of PRISSMA.

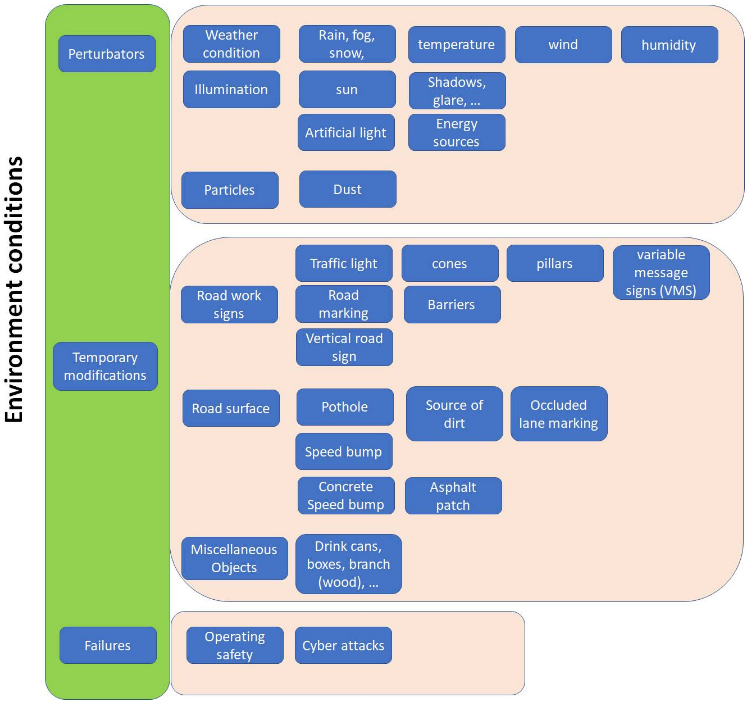

Figure 11: The sub domains and components of the environment conditions (source UGE)

For a more detailed description, the categories of disturbers to take into account in the scenes and in a scenario are given in the figure 12.

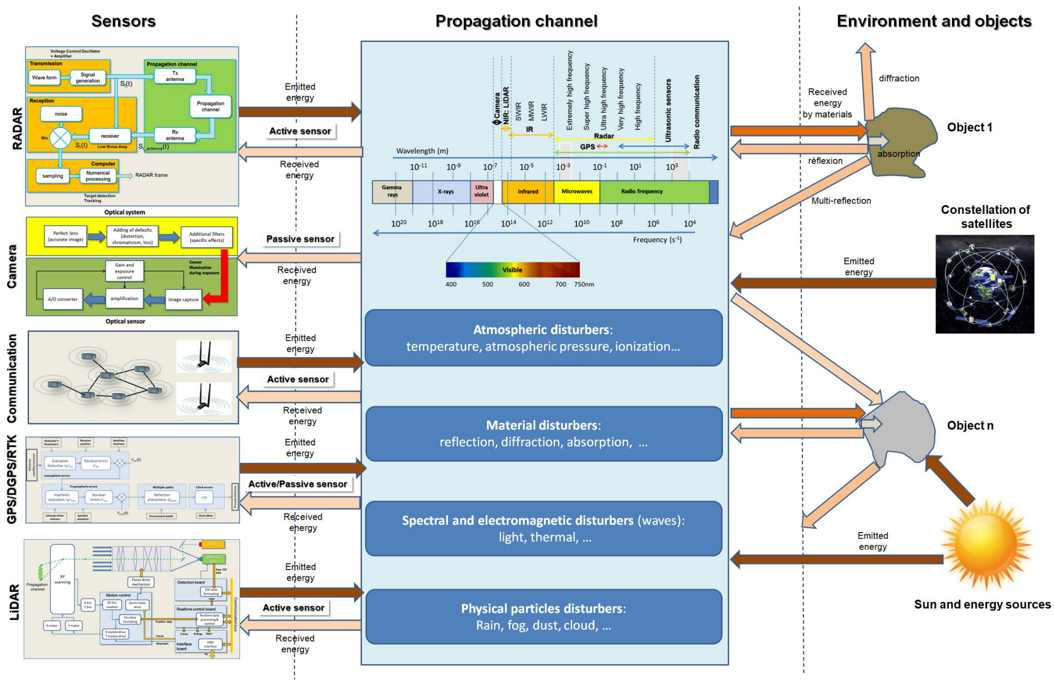

Figure 12: Modeling of categories of disturbers having an impact on the performances of sensors and data quality and involved in the domain dedicated to condition environment (source UGE).

3.2.5 Description of the scene concepts and scene life

A scenario is a series of scenes defined by a sub part of the components involved in the domains and sub-domain. A scene will start with a specific situation represented by an initial state (initial configuration) of all the elements/components/actors involved in scene (see figure 13 and figure 14). A scene is mainly focused on the ego-vehicle. In this condition, the parameters used for ego-vehicle and dynamic actors could be different. Indeed, the dynamic element could have a current state depending on the ego-vehicle current state. A scene ends with a specific final state of all the elements/components/actors involved in scene. A scene can be considered as over if an event or a set of events are reached, or if an action is generated. Often, an action is triggered after the detection of a specific event. It is important to highlight that the static physical domain contains sub-domains and components defined once and for all, without possible modification and applying to all scenes and scenarios. For instance, MOSAR defines these initial situations and configurations. Then, in order to switch to the next scene, the simulation platform using the scenario description will have to manage and to generate intermediate states and dynamic behaviours.

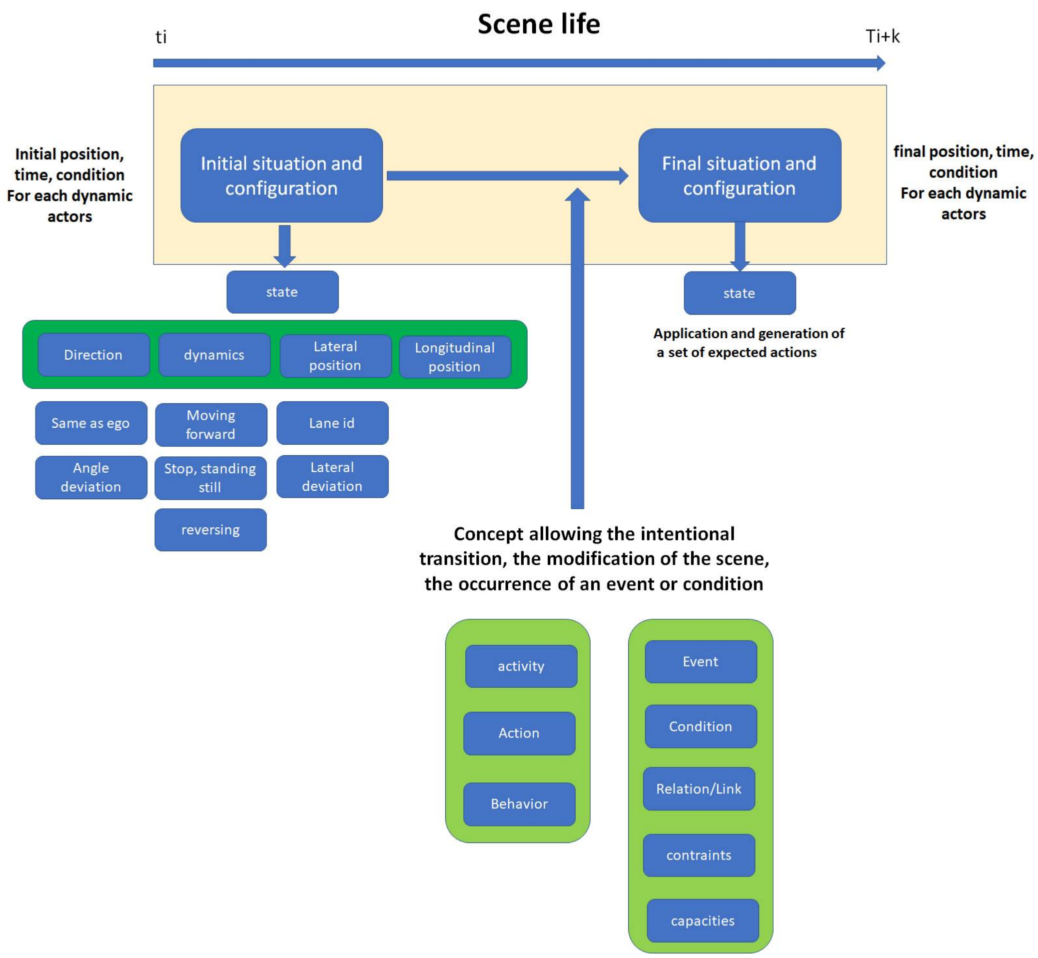

Figure 13: Description of a scene life and parameter of state for dynamic actors and ego-vehicle (source UGE)

In order to be more exhaustive about the running of a scene, it is necessary to extend the static scenario description with the dynamic aspects and the information and constraints usable by element/component/object in order to modify its state. These concepts allow to model, for instance, the intentional transition, the modification of the scene, the occurrence of an event or condition.

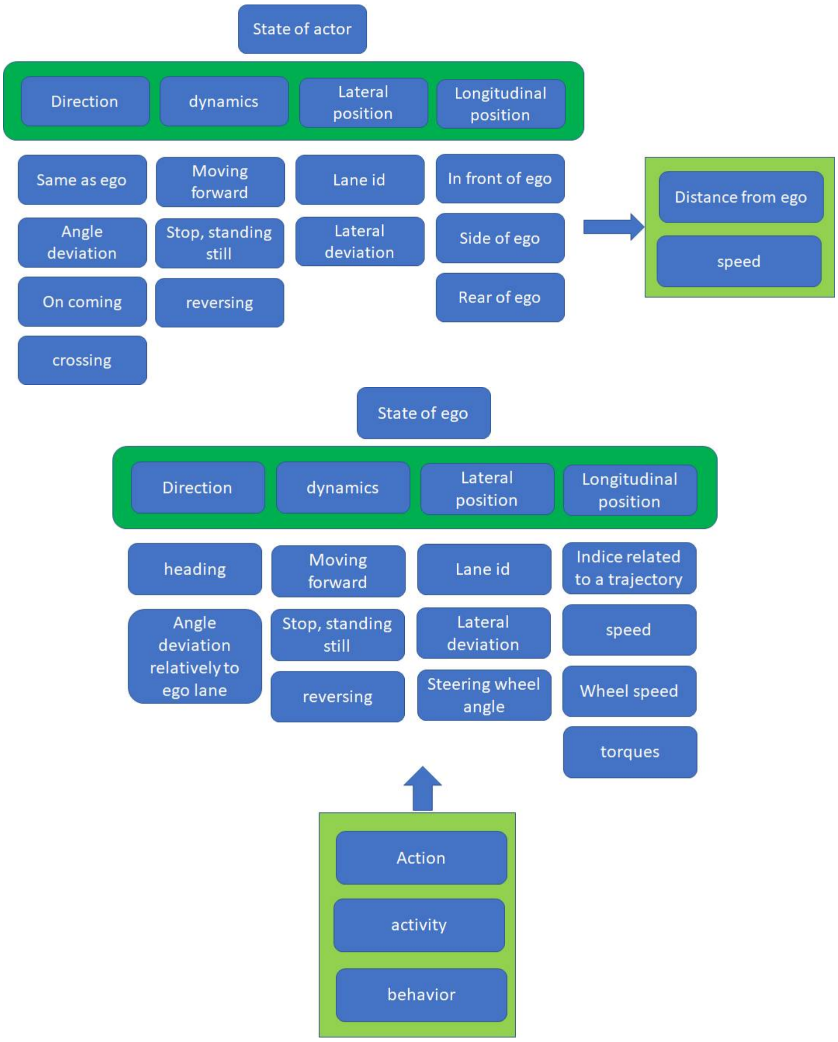

Figure 14: Description of the main parameters for the configuration of both the actors and the ego-vehicle state (source UGE)

In this context, we propose 2 sets of concepts allowing to model the type of information and the dynamic aspects of each objects and dynamic elements of the scene (see figure 15). For instance, in these definitions, we could observe the sharing of the concepts event and action. Effectively, these 2 concepts represent 2 different meanings. The first one is a punctual state of information depending of the context, some conditions, some relations. The second one is really a short term dynamic modification of a component, an actor, an element, a variable, a parameter. The type of information and the definition of this information proposed by University Gustave Eiffel are the following:

- **An event** is a specific fact building a specific situation or condition for a set of scene elements. The event is more the observation of the realisation of a configuration/conjuncture with possible conditions. From [10], an event represents anything that happens in an instant of time (frame). Therefore, any instantaneous change of state caused by an object or an element of the context can be defined as an Event.

These changes usually cause a new occurrence and, depending on the duration, this can be defined as a new event or an action. In fact, an event is an occurrence of a situation/configuration with one or several conditions at a particular time of the scene.

- **A condition** is a specific configuration (spatial, temporal, semantic) of a set of components/actors/elements allowing to obtain an event and a specific situation. A condition observation is useful for applying an action for instance.
- **A relation/link** is a specific physical relationship between an object, or a sub part of an object with another part of the scene (road, scenery, infra, ...). For instance, an em-bedded camera will have a link with the car body. A car will have a link with a road lane and the road surface.
- **A constraint** is a specific value and limit to be applied by an object (i.e. a speed limit, a forbidden sign, a traffic condition, a lane closed for specific vehicle or VRU, ...)
- **The capacity** is a set of hypotheses and parameters with defined interval of values for the actor/object/element dynamic evolution and limit of action/manoeuvre/behaviour. For instance, the capacity can define the tire grip, the engine map (motor power and speed limit), the acceleration limit, the braking torque, ...

The dynamic aspect is refined in 3 specific complementary concepts, defined as follows:

- **An activity** is a set of coordinates, acts and behaviours applying on a part (element, component, condition) of the scene during all the duration of the scene. Activity could be seen as the nominal manoeuvre. An activity occurs during a specific time interval (the duration of the scene). It is necessary to distinguish between intransitive and transitive activities because they are semantically different.
  - **Intransitive activity**: Expresses the status of objects without a specific modification in their dynamic state. For instance, the car is moving straight with the same speed, the car is parked or is stopped to a stop sign.
  - **Transitive activity**: Represents the transition from an initial state to another. For instance, the car is moving from the right to the left lane, or the car is changing lane.
- **An action** is a punctual act producing an effect, a way of acting on the current state of an element/component is order to modify its current state. Action is a set of possible acts modifying the current state of an object, actor, or weather condition to switch from the state A toward the state B by applying a model. Action could be seen as an atomic modification of the activity and/or behaviour performed by a component/elements/actor of the scene.
- **A behaviour** is a specific operating of an intern model apply to an element/actor without specific will and wish of the scenario master. A behaviour of an actor allows for the actor to have its own life and evolution in the scene. Behaviour and activity are the 2 inputs of an actor self-evolution.

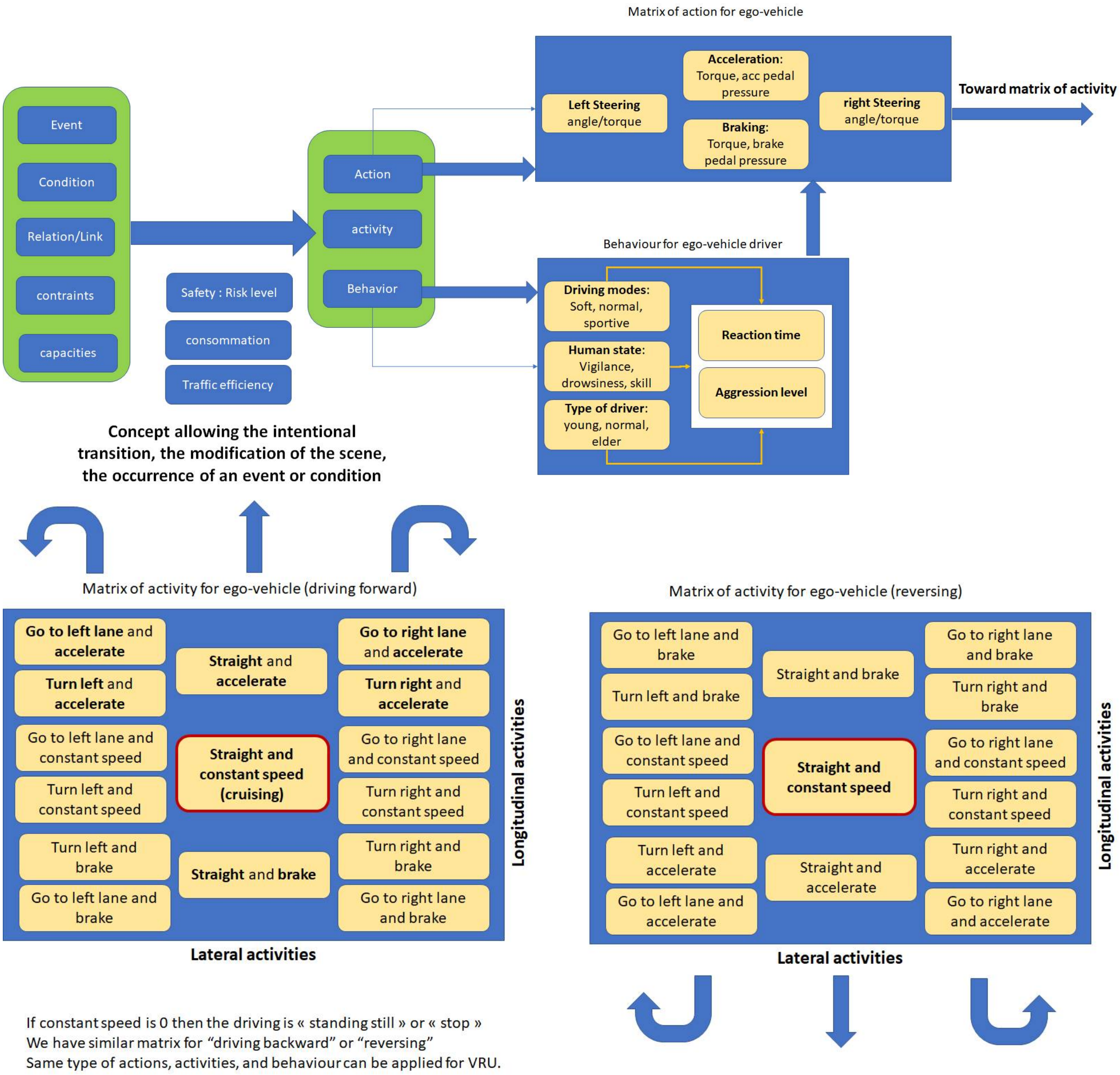

Figure 15: Description of the different states and modes for actions, behaviors, and activities. (Source UGE)

If we apply these concepts of information and dynamic state to the components of the domain and sub-domain of a scene, we can build the relation and link presented in the figure 16. The figure 17 is another way to represent the component, information, and dynamic aspects allowing to model a scene. Figure 15 presents the different states defined for the activity, the action, and the behavior. About activities, 2 matrices are proposed, the first one for the driving forward and the second one for the reversing.

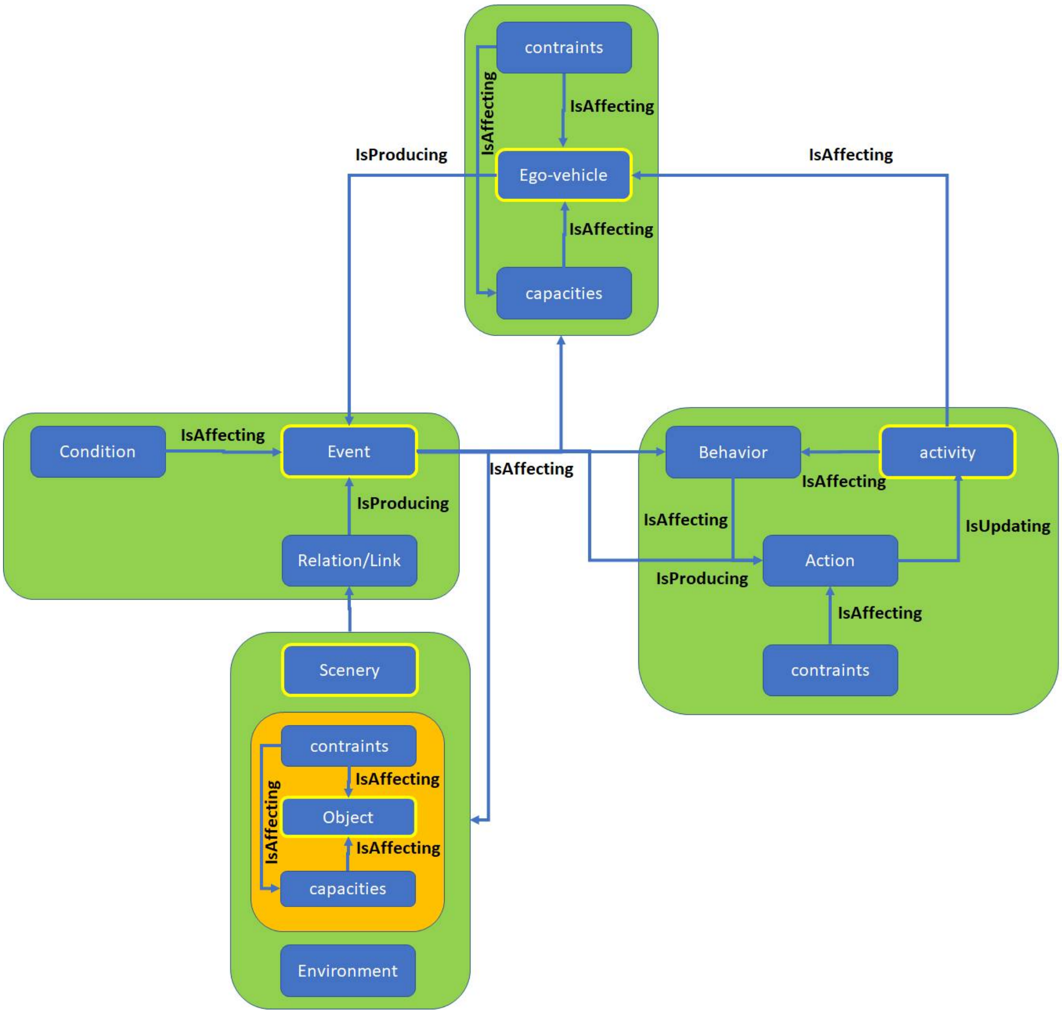

Figure 16: Diagram of the different relations, links, and interactions between the components and the concepts handled and managed in a scene (source UGE).

The switch between a scene to another one is carried out upon the detection of an event. From figure 16, an event is a specific configuration or state involving at least one component. An event occurs if a condition is reached. Then, the event will trigger an action, could modify a behavior, and at the end will modify the current activity. This event and modification of activity could be chosen as the scene changing. The figure 17 gives another way to understand the different domains and concepts involved in the scene description. In this representation, a scene is build from a set of elements defined in the ODD for both static and dynamic elements. A dynamic element (mainly the actors and the ego-vehicle) could used a dynamic model allowing to take into account specific behaviours. The overall expectation of a dynamic element is seen as an activity. A specific configuration and situation will generate a possible event leading to the generation of an appropriate action modifying the current state of the dynamic actors, elements, components.

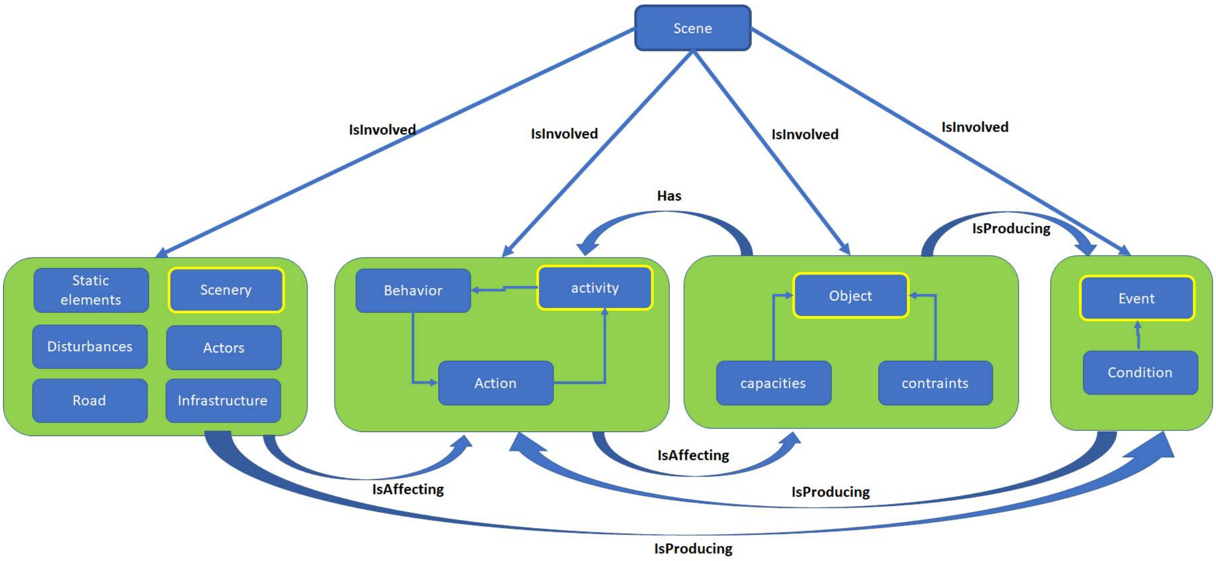

Figure 17: Diagram of the different domains and concepts involved in the scene description (source UGE)

3.2.6 Scenario description applied to POC BuSAD

If we apply this scenario and scene description to the POC BuSAD it is possible to develop a language with a set of rules for each domain, sub domain, and component.

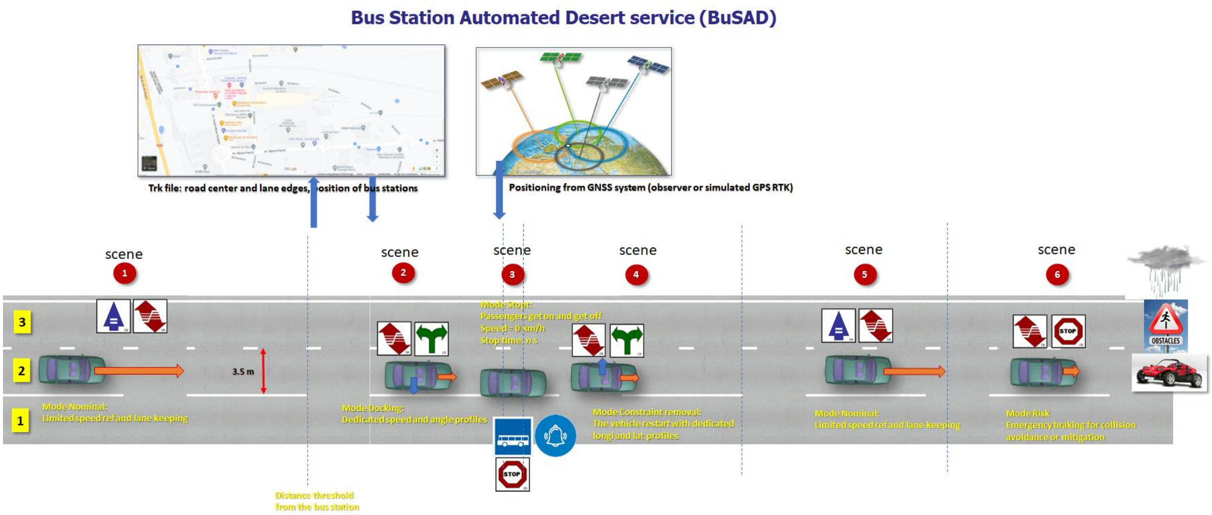

Figure 18: Bus station automated desert service with 6 possible scenes (source UGE).

The POC BuSAD can be model by at least 6 possible scenes. The figure 18 represents the upstream and downstream scenes for the docking at a bus station:

- **The first scene** corresponds to the nominal driving mode. In this mode, the shuttle or AV is driving on the right lane with a maximal speed up to 20 km/h and, in case of a front moving vehicle, it will manage the inter-distance and apply a car following manoeuvre.

- **The second scene** occurs when the distance between the ego-vehicle and the bus station is lower than a threshold and the bus station is identify by the perception system. In this second scene, the vehicle will apply dedicated both longitudinal and lateral profiles in order to reach the bus station with constraints of ego-vehicle speed and lateral deviation.
- **The third scene** concerns the stop period to the bus station allowing passengers to get on and off.
- **The fourth scene** focuses on the restarting from the bus station with dedicated longitudinal and lateral profiles allowing to reach the center on the right lane.
- **The fifth scene** is similar to the first one.
- **The last scene** concerns the reaction on a critical event like an object stopped on the right lane. In this situation, the ego-vehicle will have to apply an Emergency Braking allowing to avoid or to mitigate the collision.

It is interesting to quote that the BuSAD service will apply these 6 scenes continuously on a trajectory covering 3,4 km with 4 or 5 bus stations. The environment will have bends with very small radius of curvature and intersections that may contain vehicles entering the main road and the ego-vehicle traffic lane. Moreover, the ego-vehicle does not have the capability to apply lane changing and collision avoidance by lane changing. The ego-vehicle will keep the same lane during all the evaluation and validation process.
In the definition of the rules allowing to describe the scenes, the set of rules for the static physical environment are defined only one time. For instance, figure 19 shows a sub part of the sentences usable to define a part of the time of day, the scenery and the road configuration. The same type of sentences is built for environment conditions, dynamic physical environment, and ego vehicle. To these sentences, for each scene, we add other sentences managing the informative and dynamic parameters like events, constraint, condition, capacities and the action, activity, and behavior on elements/components/actors of the scene. The detection of a specific event reaching defined conditions will produce a specific action depending on current activity and actor behavior. For instance, the ego-vehicle reaching a distance threshold from the bus stations will generate a specific action which consists to apply a speed and lateral profile allowing to converge toward the bus station. This action will be applied depending on the behavior and the current activity.

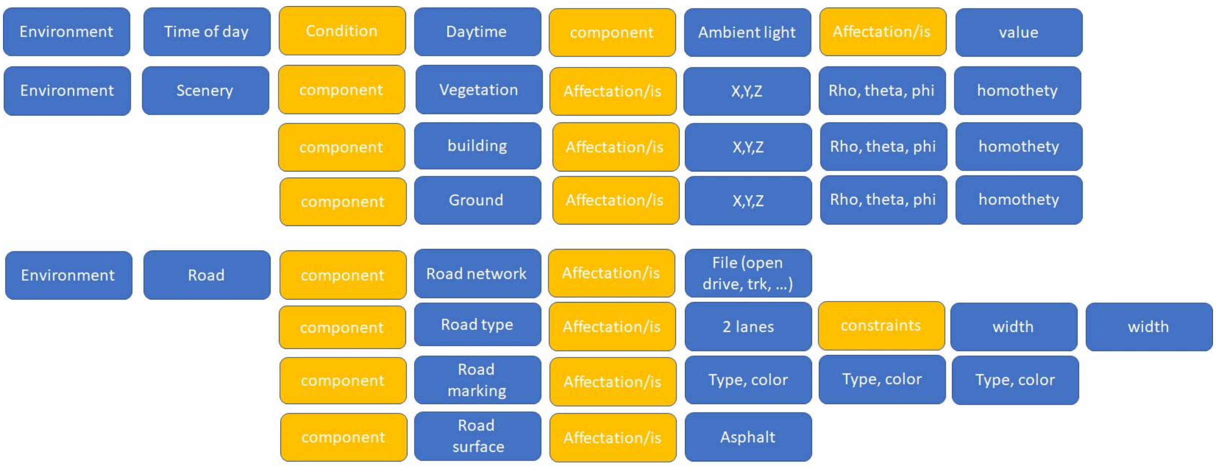

Figure 19: Example of sentences for static physical environment (source UGE).

4 Scenario Management

4.1 General Overview

The scope of task 2.4 is to propose mechanisms for scenario management, which includes their structuring, storage, manipulation, selection and parametrization for evaluation. The focus of this section is on the motivation for a proper methodology for managing a scenario catalogue, the available building blocks for PRISSMA and what is next in this management scope for the next phases of the project. A previous step to scenario management includes the scenario generation. As a reminder, scenario generation is addressed in task 1.2 of WP1.

A descriptive scenario catalogue should provide:

- Storage capability for a set of scenarios which can be characterized as needed according to the interest of the contributing parties, from a general definition to more specific parametrizations according to regulation or specific safety, performance, or comfort tests.
- Access management in order to protect confidentiality.
- Scenario labeling or categorization according to specific criteria: i.e. types of road users, regulation scenarios, among others.
- Scenario traceability in order to visualize the progress when describing, enriching and validating the scenarios.
- Means for describing and exploiting the results of the required situations for certification without the necessity of specialized knowledge on testing, whether it is on simulation, test track trials or other platforms.
- Means for describing high-level scenarios as generically as possible in order to be equally applicable for different systems and to be understandable and shareable among contributors, certification organisms, and any other interested parties.
- Means for providing scenario descriptions that can be exploited subsequently on different simulation platforms.
- A structured framework for the analysis of the results coming from testing platforms (i.e. simulation or others) according to metadata and statistics characterizing the entire scenario dataset which can contain hundreds or thousands of instances (i.e. concrete scenarios). Otherwise stated: a structured and synthesized interpretation of results based on key aspects of the test cases instead of continuous data for each test.

4.2 Scenario Management Methodology in MOSAR

The aforementioned points describe the fundamental high-level requirements to characterize and manage scenarios and can be concretely apprehended with MOSAR Scenario Manager, which is a tool precisely instantiating these principles and providing a practical overview of a methodology to describe and manage scenarios.

Scenario management is then a complex task and the MOSAR Scenario Management plat-form allows users to have a database to register, store, classify, search, trace, analyze, import and export scenarios from and into the database. This tool can be used to manage descriptive scenarios which can also be transformed into test cases for simulation in different formats (OpenScenario, SCANeR studio, etc.).

In MOSAR scenarios are defined based on a keyframe paradigm which translates into a sequence of scenes describing a time instant with different actors, equipments, infrastructure and environmental conditions as defined in section 3. Scenarios are then defined under this paradigm in order to be managed at a large scale, this holds for functional, logical and concrete scenarios.

4.2.1 Generic high-level concepts on Scenario Management in MOSAR

Scenarios are stored in specific data structures called *containers* which provide the means for storage, access restriction in order to ensure confidentiality (among other requirements), and collections access in order to describe scenarios; this is access to specific collections, i.e. elements related to infrastructure, actors, equipments, and behaviors among others.

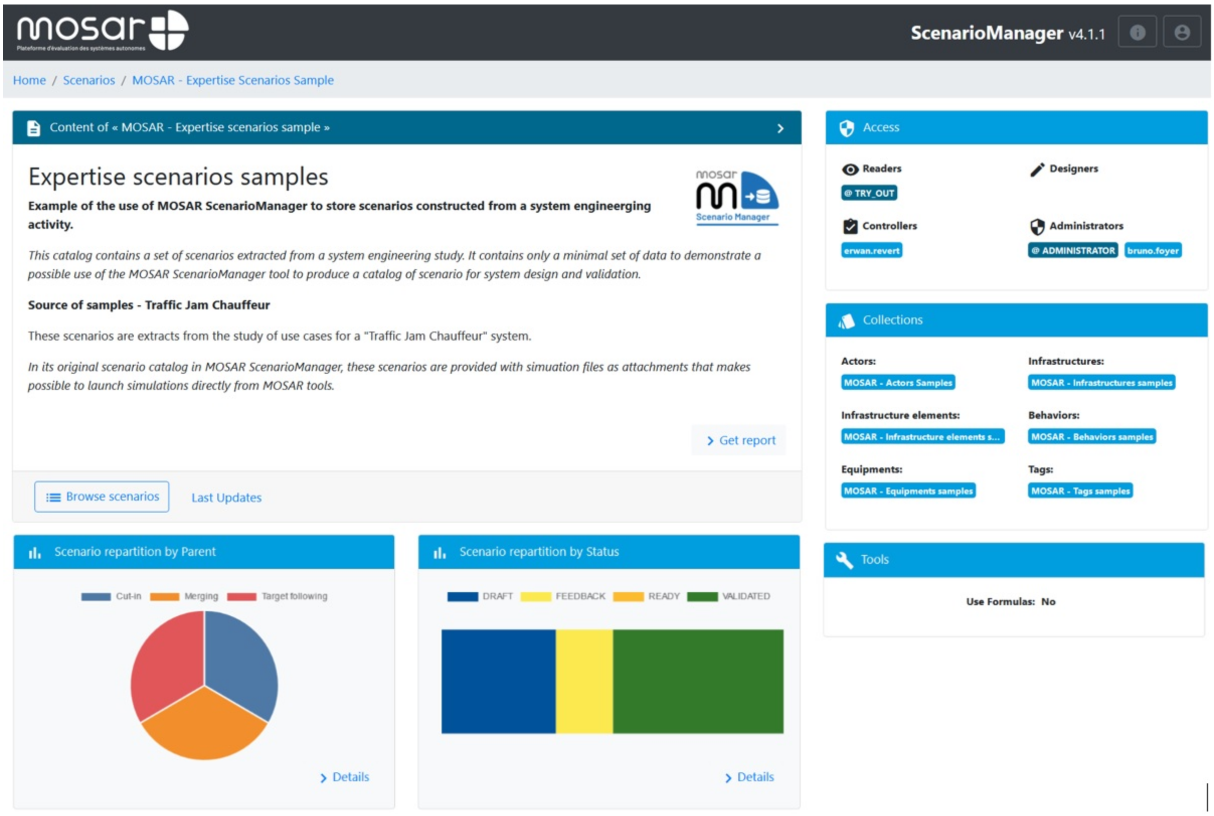

Figure 20: Container Management in MOSAR

4.2.2 Scenario Classification

To facilitate their management, scenarios are organized in a tree-like structure based on the three description levels: functional, logical and concrete. This allows for an easy overview of progress in the description of scenarios that will most likely be used in simulation but also an overview on the progress of the results retrieval for concrete scenarios, whether they come from simulation or real-world testing.

Moreover, classification of scenarios in the catalogue can be refined at a deeper level as they can be customized according to *tags*. Tags can be then used to attach an element that characterizes the scenario and through which similar scenarios in the database can then be described, filtered and analysed statistically.

4.2.3 Scenario Management from Different Sources in PRISSMA

The structure according to which scenarios are defined in MOSAR is such that data can be retrieved from different sources and be analyzed and even compared analogously. Trials can be carried out in real-world settings as well as simulation and given the structure of each scene in each scenario, feature extraction can be performed on continuous data and results can be re-injected into MOSAR as concrete scenarios. Results can then be categorised and analysed statistically. In this sense, sources for scenarios can then be: accident-related, real world driving, test track trials, safety expertise scenarios, list of scenarios provided by regulatory bodies, and any other source that can provide a description as a sequence of scenes, each scene being characterised with the proper parameters: infrastructure, per actor description and behaviour (static and dynamic), and environmental conditions, among others. Figure 21 displays this heterogeneous data retrieval and potential use in subsequent phases.

Figure 21: MOSAR - Scenario Manager

4.2.4 Statistical Analysis

The statistical analysis feature existing in MOSAR allows users to visualize the existing data in the platform based on the selection of parameters or conditions predefined by the operators. For example, a database with a high number of scenarios and parameters can be analyzed by choosing one or many particular parameters to include or exclude from the search. Moreover, the scenario catalogue can be analyzed for different purposes and therefore through different criteria. As an example, an environmental parameter can be specifically chosen to search for scenarios with only "Rainy" weather and export these data or create specific graphs and find parameters inter-dependencies.

Figure 22 provides an illustrative example of parameter manipulation; however broader manipulations are possible allowing to assess the scenario-structured data. One or more filtered parameters can be performed with precision on their values. In the example, scenarios in which rain is present and moderate are retrieved. Namely, the filter is applied in the database, resulting

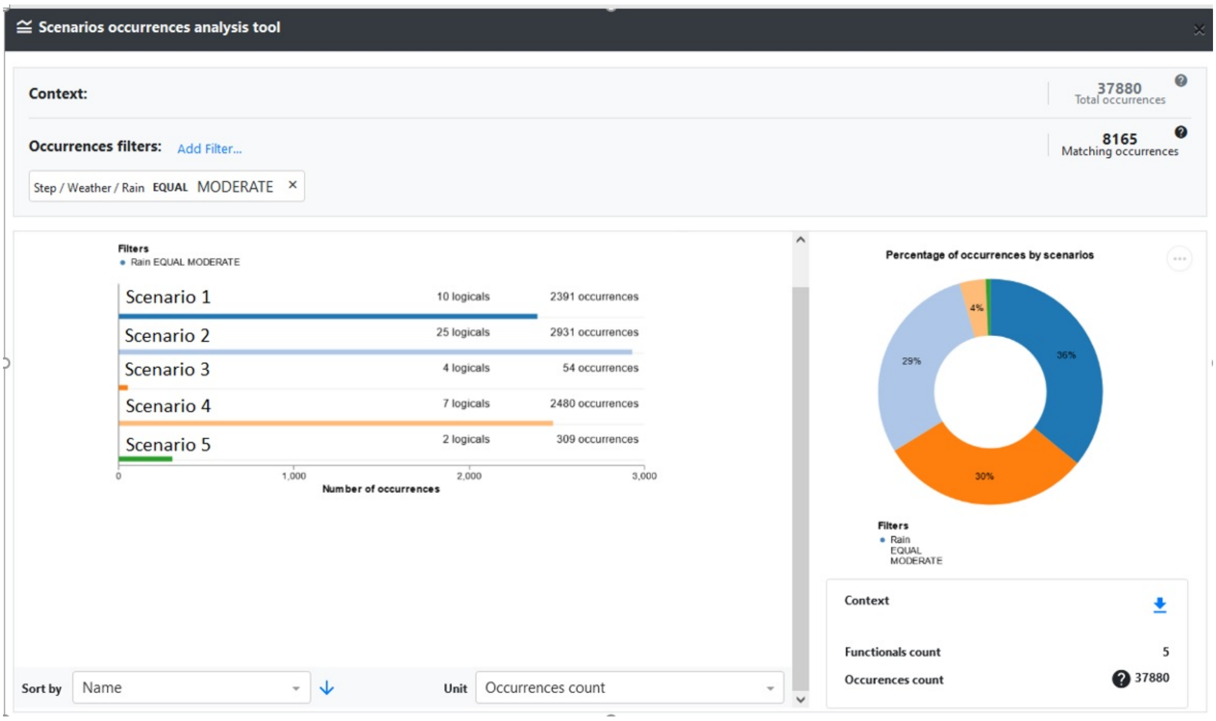

Figure 22: MOSAR - Scenarios occurrences analysis tool

in 5 functional scenarios and, for each one, the number of logical and concrete scenarios (occurrences) is retrieved and visualized in proportion the database. The tool will provide the resulting occurrences from the total number of scenarios existing in a specific catalogue. These results can also be exported in different formats (pdf, csv, JSON, among others). Naturally, many other configurations are possible, for example rainy weather and presence of a VRU (Vulnerable Road User) can be used to assess particular safety hazards for VRUs given this particular weather and the influence of this factor on the performance of a particular sensor.

4.2.5 MOSAR in the PRISSMA Scope – Potential Evolution

Regarding scenario management, and given the scope of PRISSMA, one pertinent question is the one related to how artificial intelligence is accounted for in MOSAR. As of now the management of AI-related data is implicit in MOSAR as it is possible to retrieve and structure data coming from sensors which are the result of the application of AI algorithms. MOSAR's roadmap constantly integrates feedback from OEMs as well as French regulation authorities and international reference bodies and it is open to retrieve feedback from PRISSMA regarding potential requirements to structure AI-related parameters in a suitable manner according to the identified needs in the project. The work to be carried out in the POC of the project will determine the suitable way to integrate AI-related aspects in MOSAR with the current data model of if new needs are identified.

5 From Descriptive Scenarios Towards Simulation

5.1 From Scenarios in MOSAR towards Simulation Test Cases

Descriptive scenarios in MOSAR can describe at a high-level specific situations to be structured and assessed. As aforementioned, these situations may be related with the service to be provided, the working conditions and the subsequent safety assessment.

One fundamental challenge arising when going from descriptive scenarios in MOSAR to simulation is then to fill the gaps between scenes. Scenes describe key-frames of the situation with specific configurations for actors and environmental conditions and therefore parameters. In simulation then different dynamics can lead from a scene A to a scene B.

It is then fundamental to define the system to be tested and the expected results. For a descriptive scenario to be suitable for simulation, the SUT should be specified in order to generate use cases and per use case then the parameters should be further constricted to represent the specific test to be carried out and the specific rules for acceptance.

Moreover, key parameters must be identified or ”annotated” in MOSAR’s scenarios in order to ensure simulation compatibility. These parameters, also called annotations, are as follows:

- *Initial conditions* or *input parameters* of a manoeuvre to be performed (target values). Variability should be performed in simulation for parameters annotated as inputs when a range of values is specified.
- *Triggers* are parameters determining when the transition to the following scene is triggered.
- *Expected values* indicate the expected results in simulation.
- *Actor behaviour specification.* Each actor can have one or more manoeuvres, that have to be described with specific parameters and described by a start scene of the behaviour and an end scene of the behaviour. This prevents having missing preliminary information for the actions of the actors during the scenario creation in simulation.
- *Stopping criteria.* Group of technical parameters specifying the conditions when the simulation should stop. Some examples include: imposing an end-time or condition to simulations in order to force its ending, imposing the simulation to stop if an unexpected event arises, among others.
- *Acceptance criteria.* Group of expected behaviour of the SUT, that guarantees the success of the validation scenario (test case).

Testing AI decision-making can have different expected behaviour of the ego vehicle for the same situation in black-box mode. The challenge at this stage is the management of the acceptance criteria and all the possible, feasible and pertinent combinations of parameters of the current scenario.

Figures 23 and 24 provide an overview of how this annotation process is carried out in MOSAR when describing a use case which links a scenario to a specific system. The presented example is an excerpt for a cut-in scenario and in figure 23 specifics are given for the cutter parameters in this scenario. Figure 24 presents an excerpt of annotations with the overview of the storyboard in the process of creating the test protocol for a specific use case for the cut-in scenario.

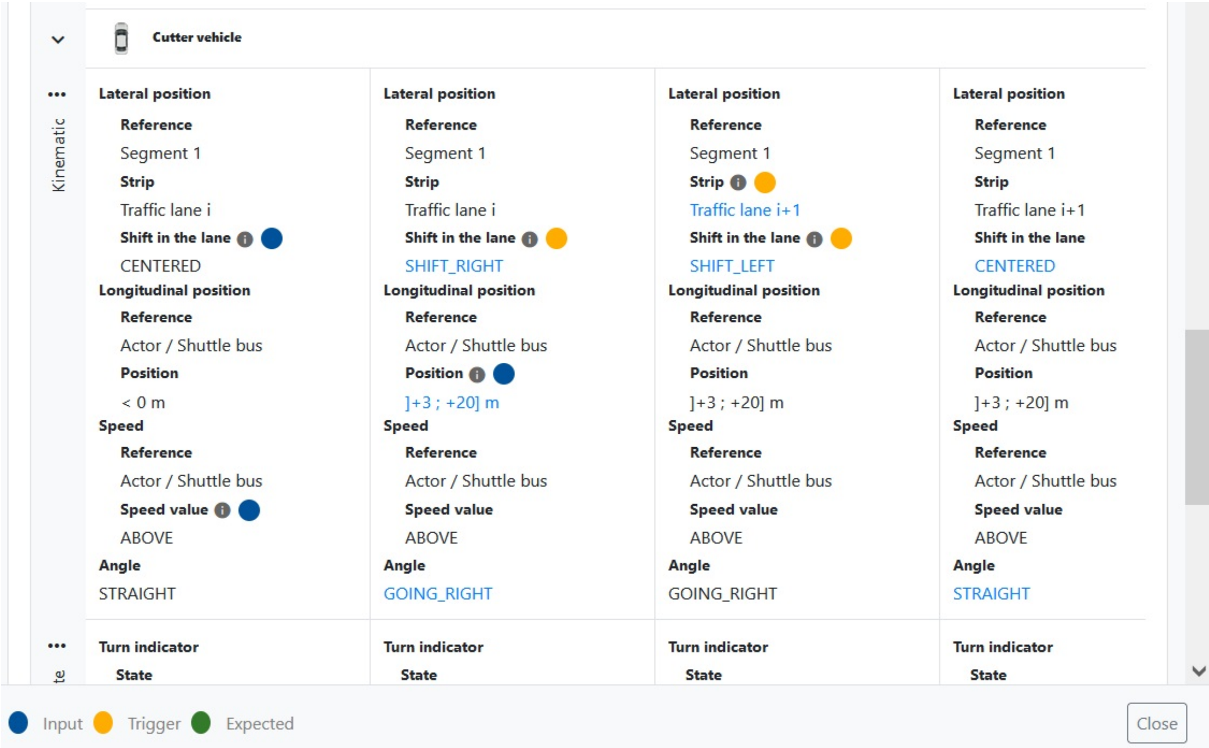

Figure 23: Parameter annotation in MOSAR - Aiming toward simulation - parameters for cutter actor in a cut-in scenario

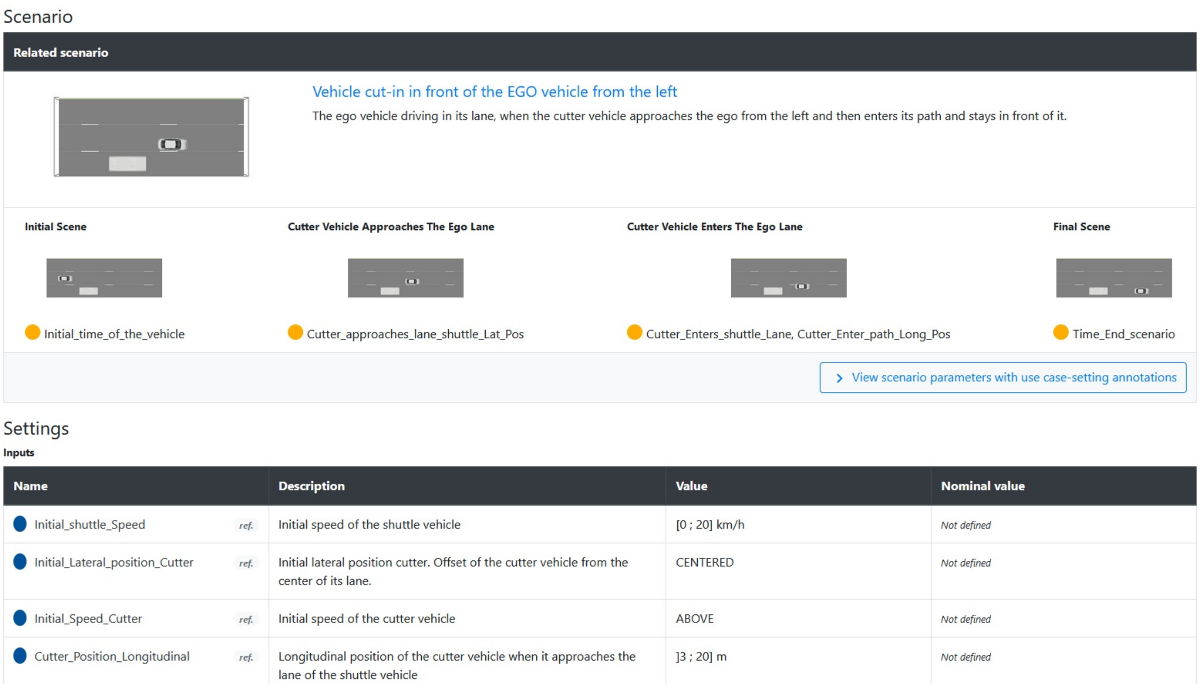

Figure 24: Parameter annotation in MOSAR in a cut-in scenario - Aiming toward simulation

5.2 Implementation of scenario in simulation environment: Open-Drive and OpenScenario

Whether it is question of real tests or simulation, to describe a scenario we use a succession of actions and maneuvers performed by actors in a certain context which is most of the time a road and its boundaries. This road will be key when we will describe what should happen during a scenario because we will use various elements and specifications belonging to the infrastructure of the road network. Here is an example we can use to express a very simple scenario: "The car follows the right lane on one hundred meters". For this simple task, we need to know what a road lane is and have a way to ensure that the traveled distance is correct and corresponds to what we asked. For that kind of description of road network, we rely on what is commonly called the semantic layer of the road. So, in simulation, to represent a road network we need two models that are intrinsically linked, the 3D model that is the visual and physical part and the semantic layer that is the description of the road network. A certain number of standard description formats exist and are used across simulation platforms, but the one that stands out is OpenDRIVE.

OpenDRIVE format provides a common base for describing road networks with extensible markup language (XML) syntax, using the file extension xodr. The data that is stored in an OpenDRIVE file describes the geometry of roads, lanes and objects, such as roadmarks on the road, as well as features along the roads, like signals. The road networks that are described in the OpenDRIVE file can either be synthetic or based on real data.

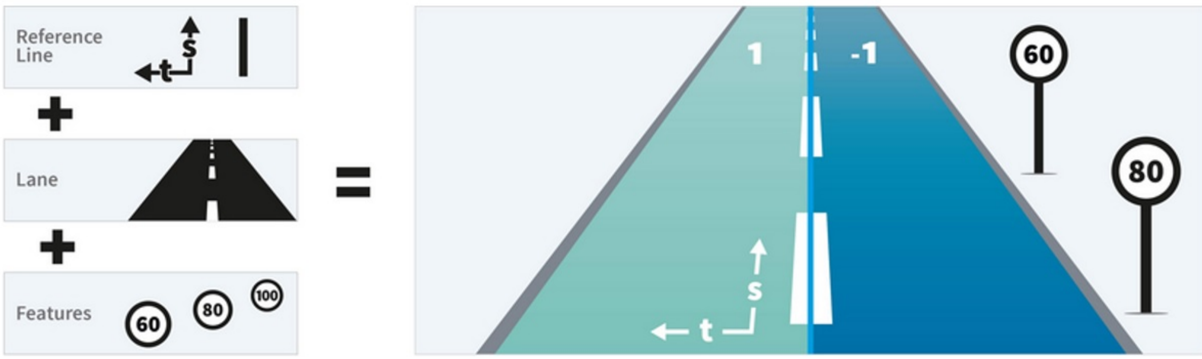

Figure 25: OpenDRIVE road network description

Thanks to OpenDRIVE, we can rely on the road network description to express actor's maneuvers and actions to define scenario at higher level of language. This is the purpose of OpenSCENARIO, the scenario is a story we want to tell, and the scene where it takes place is the road network. The OpenSCENARIO file, which is also an XML file, relies on the OpenDRIVE description and provides a set of actions and uses an event-based scenario execution model that should support all levels of scenario description, from the very abstract to the very concrete, in a suitable way.

Like OpenDRIVE, OpenSCENARIO is supported by ASAM (Association for Standardization of Automation and Measuring Systems). Although the OpenSCENARIO format is younger and still in development, its version 2.0 is on the way and is already popular and used by simulation industry actors. Both formats are here to be used by simulation platforms to have a common and inter-operable description language and representation for scenes and scenarios.

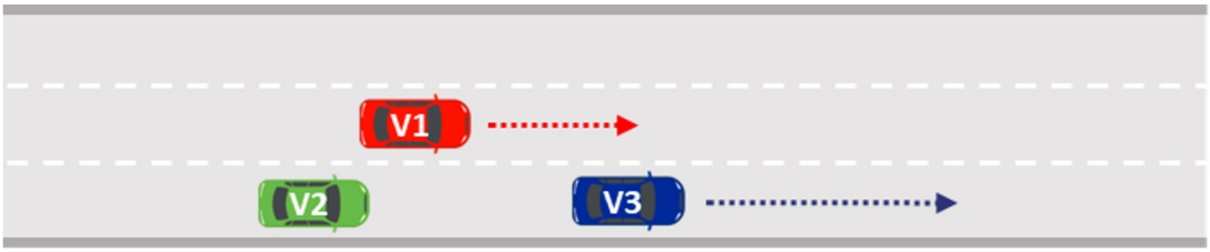

Figure 26: OpenSCENARIO scenario example

5.3 3D Environment and Digital Twins for Scenarios

Depending on the accuracy of sensors used during simulation, different qualities of the 3D environment are needed:

- ideal & stochastic sensors: a logical description of the environment is sufficient
- physics-based sensors: a 3D description environment, with 3D geometries and materials calibrated for every sensor (camera, radar, lidar, GPS, etc. . .) is required.

Scenarios can either be described on an imaginary road designed for a specific scenario or standard (like NCAP tracks), or be applied on some roads existing in a real world.

Regarding the variability of the simulation scenarios, the following considerations are pertinent.

Regarding blocking points on simulation tools:

- Require experts in multiple domains
- 3D models used to represent a scene are not scalable to scenario needs
- If real time simulation is a constraint, the accuracy of certain model may need to be decreased

Regarding hints on what can be addressed before the simulation phase:

- Identify what will be the system under test
- Define the stakeholder of the system
- Specify the level of accuracy needed for simulation models
- Describe the expectations regarding the results of a scenario

Regarding randomness in a scenario:

In simulation, we can use mechanisms to spawn actors with variable parameters, we can manage those parameters by giving them some weight. Imagine you have a way to generate traffic around the SUT, and you can define the density, the type of actors surrounding (cars, trucks, bicycles among others), their colours, the way those actors behave and many other parameters we can have.

If the system is fully random, each simulation run will differ from the others since parameters are random. At this stage, research suggests that this feature might not be compatible with the certification approach in the PRISSMA project.

5.4 Scenario description implementation and use in Pro-SiVIC

Figure 27 presents the mechanism applied in order to manage dynamically the scene, events, and scenario in Pro-SiVIC. The first stage is to provide a high-level description of the scenes and events using ODD and OEDR taxonomy in order to respect the used conditions and operating limits of the specific system under test (either an algorithm, an application, a component, a full service). The sentences sent to Pro-SiVIC in the plug-in sivicScenarioManager will be interpreted in order to generate the script commands in the Pro-SIVIC format. This script language developed for the needs of Pro-SiVIC allows to handle, modify, delete, create in real time elements and components of the simulation.

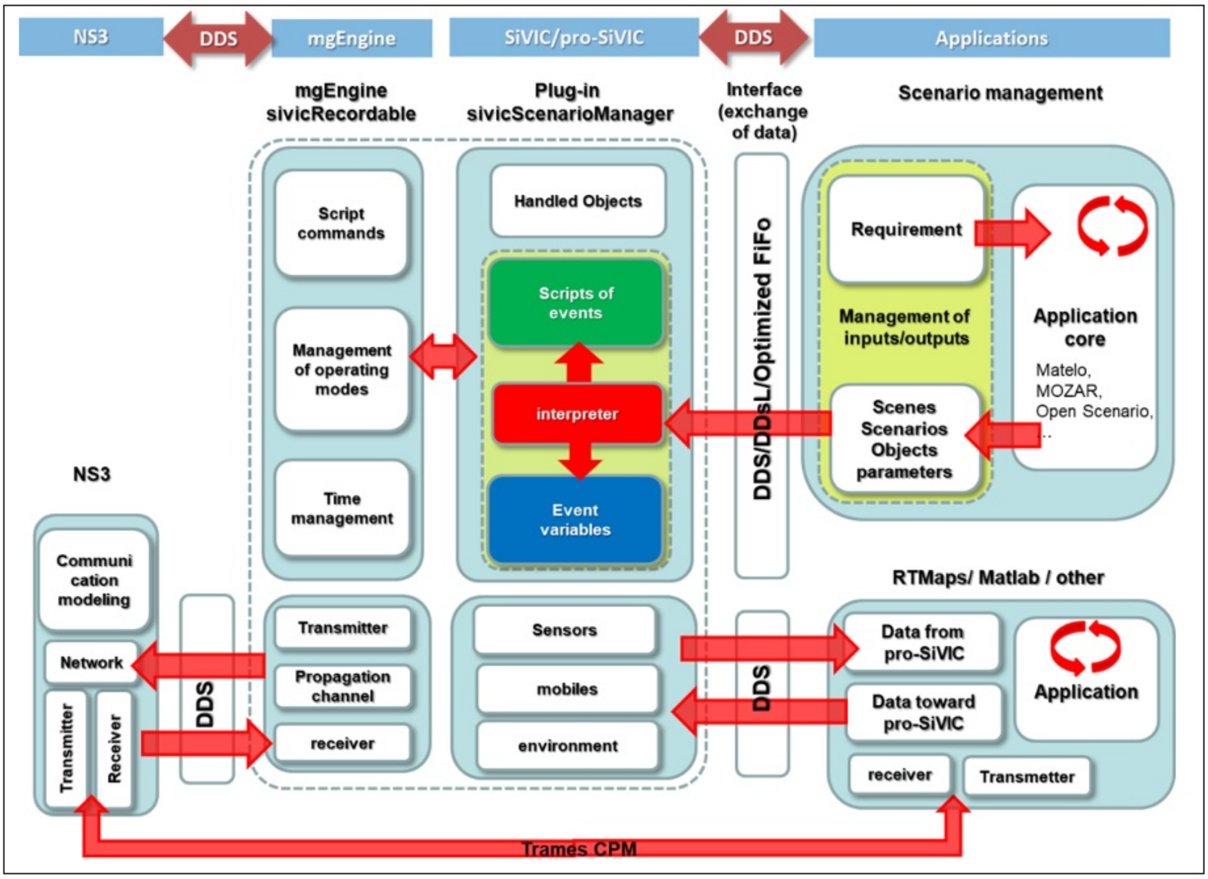

Figure 27: Overview of scenario management with the different modules allowing to generate a set of scenes with a generic modelling/language. In Pro-SiVIC, a specific module called sivicScenarioManager will be used in order to consider the scene and scenario provided by third party software and will convert these rules and sentences in a set of Pro-SiVIC set of commands. (Source UGE)

In order to be able to dynamically manage events relating to a set of criteria such as time, period, position, distance and others, an event management mechanism has been integrated into Pro-SiVIC. These events and their management are carried out using the sivicEvent plug-in (see figure 28). The main purpose of this mechanism is to be able to enrich scenarios using modifications and conditional triggers. Each type of event is constructed generically. An event is triggered (execution of an event script) only when a condition is met by a ”positionable” object or a mobile object such as a vehicle or pedestrian. For example, if the ”timesivic” criterion is selected and a period of 500ms is set, then the event script will be called and executed after an elapsed time period of 500ms. In the case of a "position" type event, the event script will be called and executed when an object or a vehicle chosen as the source of the event is within a fixed radius.

The events that can be managed are currently of the "position", "inter distance between objects", "trajectory index", "date", "time interval", "larger" and "smaller" type (outdoor mode managed from an RTMaps type application). The definition and description of these types of events are:

- **Position**: the reference object is close to the defined position
- **Distance**: the reference object is at a distance from another vehicle
- **Index**: the reference vehicle or pedestrian is at a trajectory index
- **Time**: the reference vehicle is on a date
- **Timesivic**: the reference object has operated for a time interval
- **DDSless**: conditional value less than a threshold
- **DDSgreater**: conditional value greater than a threshold

A number of iterations can be defined in order to specify the set of commands to be carried out following the realization of a specific event. This sivicEvent plug-in inherits from the operating mode and time management plug-in (sivicRecordable). This implies that it accesses the same mode of operation as all the plug-ins available in the SiVIC platform. Thus, to disable the management of an event management plug-in (sivicEvent), the *Off* mode will be selected. To activate the taking into account of an event, the *On* mode will be activated. This operating mechanism is shown in the figure 28.

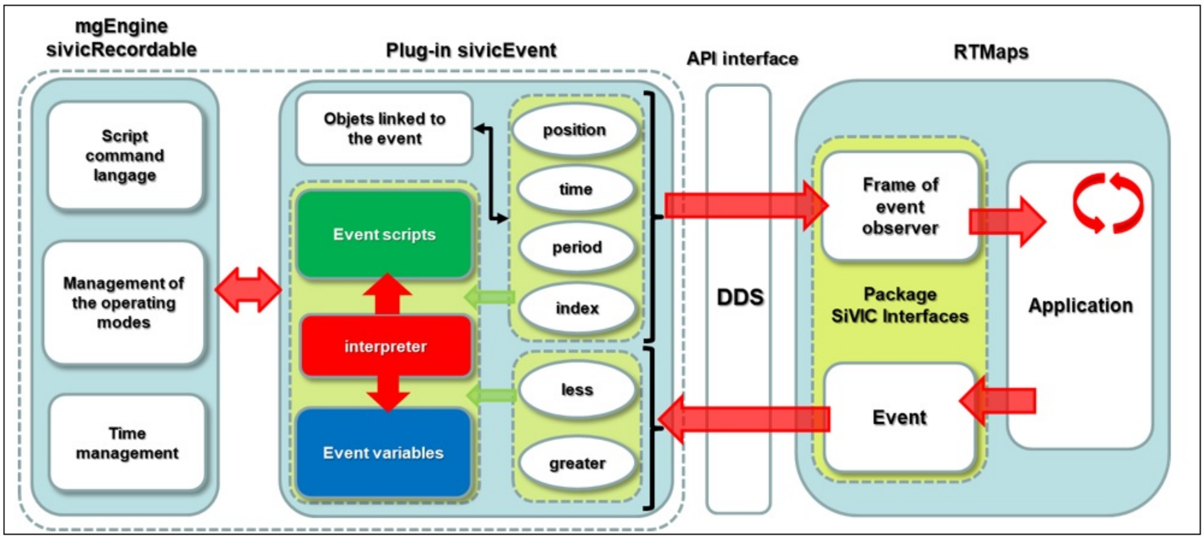

Figure 28: Functional architecture of event mechanism implemented in Pro-SiVIC platform

6 Scenario Combinatorial Management

6.1 Strategic choices to manage combinatorial explosion

Knowledge and comprehension of the SUT should help reducing the number of parameters to explore. For example, if we are testing an emergency braking feature based on radar detection, we can drop all the test concerning the day / night cycle influence since our system is not sensible to it. What we want to illustrate here is that the first approach should consist in identifying the sensitivity of the system.

Then the accuracy of the system should also give some input regarding the number of parameters to be tested. As an example, if the requirement is to validate the capability of a system to detect pedestrians, perhaps it is not necessary to test every shape, height, skin, movement style. As a first step, a proper consideration could be to evaluate only the extremes to study if the SUT is sensible to those kinds of parameters. A basic strategy could consist on "how can we address the system and refine step by step". With a list of parameters to vary and for each a few values, then to perform a first batch of simulations to identify the parameters that need to be explored.

However, in order to address the variety of real-world traffic situations, relevant test scenarios have to be selected for the AV safety assessment and different strategies have been proposed to select them while managing the combinatorial explosion, they are listed in the state of the art report from task 2.1, see [11].

The scenario generation method proposed by INRIA (see Deliverable L1.4, Section 4.2.4 [3]) is based on conformance testing guided by test purposes [12]. This is a form of model-based testing [13], which takes as input a formal behavioural model of a configuration (scene and behaviours of actors, i.e., ego vehicle and mobile obstacles) and a test purpose (automaton describing the sequences of events leading to an accepting state) and produces as output a set of test cases corresponding to the combined actor trajectories compatible with the test purpose. The number of different test cases generated from the formal model of a configuration and a test purpose is determined by the interleaving of events (actions) carried out by the actors. Therefore, the number of test cases increases with the number of actors and the complexity of their respective behaviours.

The number of test cases can be reduced by fine-tuning the parameters of the test generation process:

- Consider only test purposes that characterize critical situations, for example those corresponding to road accidents [14].
- Refine the test purposes by adding constraints (e.g., forcing an actor to pass through certain intermediate steps before the accepting state of the test purpose is reached).
- Use REFUSE directives (a technical parameter of the conformance test generation) to avoid exploring certain parts of the behavioural model that are judged irrelevant w.r.t. the test purpose.

6.2 Combinatorial Management: Ansys optiSLang for Process Integration & Scenario Parameter Variation

Ansys *optiSLang* is the tool for process integration and scenario parameter variation. The input parameters of a CAx model are varied for different purposes, typically sensitivity (or variation) studies, optimization, or reliability analysis. Some of them will be explained further.

For process integration, either a dedicated interface between Ansys optiSLang and driving simulators such as IPG CarMaker, or generic interfaces, e.g., based on structured text input and results files and batch files, such as for AVSimulation SCANeR. Combinations of different solvers in a chain or in parallel, conditional execution and even nested loops are possible. Ansys optiSLang generates sets of input parameters ("design") due to different algorithms, launches the process chain for each design, collects and evaluates the results from the solver. This process integration allows the user to easily organise the parameter variation of the scenario variables of the driving simulator used.

In the scope of a sensitivity study, scenario parameters are varied following a schematic design of experiments (DoE) [15], such as the full factorial DoE, or by random sampling. For random sampling, all parameters are assumed to be uniformly distributed within their user-defined ranges and are treated as independent. This would fulfil the requirements to a DoE, and no assumption of stochastic parameter properties is taken. Specific sampling methods such as Advanced Latin Hypercube Sampling [16] prove to be more efficient for a scan of the parameter ranges than schematic DoE.

For multi-variate statistical analysis of the relation between input parameters and driving simulator results, a mathematical model must be fed to the gathered data. The algorithm Metamodel of Optimal Prognosis (MOP) first applies a reduction of the input dimension by means of statistical criteria, then creates and evaluates different model approaches from a catalogue of available models, ranging from polynomials to neural network modes. The accuracy of each model is evaluated by the statistical procedure called cross validation. Thus, MOP automatically finds the best of all available meta-models.

Typical results of a sensitivity study are:

- Simple statistics like histograms and correlation coefficients, trend analysis
- Variance based sensitivity indices [17]. In brief, these indices tell how much one input contributes to a result, allowing for a ranking of importance.
- Cluster analysis
- Statistical analyses along with various graphical data representations enable a detailed survey of the input–output relations.

A machine learning algorithm called Adaptive Metamodel of Optimal Prognosis (AMOP) combines sampling and meta-modeling. Within several iterative steps, the algorithm learns from available results to concentrate subsequent samples for different goals, e.g. to optimise local errors or to fulfil criteria (objectives, constraints). In the context of the evaluation of a Software-in-the-Loop (SiL) of an ADAS system, it is possible to define criteria or key points of interest (KPI) and the respective threshold values, *to find eventually regions in the scanned space where the SiL does not perform as intended*, and the data analysis methods serve for root cause analysis in case there is an issue.

Robustness and reliability analysis are related in that way, as random properties of input variables are taken into account in the parameter variation, in order to study the influence of uncertainty on the input side onto the simulation results or KPI. In context of an ADAS evaluation, KPIs can be comfort criteria of the driving performance, such as maximum acceleration or deceleration, or safety criteria, often expressed as the minimum time to collision or minimum distance.

Robustness analysis mainly looks at the scatter (expressed by standard deviation or coefficient of variation) variables. Comparing the scatter of simulation results to scatter of inputs gives an idea of the stability of the process. The definition of threshold values for KPI and expressing the safety margin, i.e., the distance between the mean value of a response and the threshold value, as multiples of the standard deviation, usually abbreviated s, leads to Design For Six Sigma (DFSS): the reliability of the system under examination, thus the probability of staying within the threshold, is expressed by a simple scale, the sigma level. The data analysis options in the Ansys optiSLang post-processing application allow for such analyses.

In fact, the basic assumption in DFSS is that the observed KPI follows the Gaussian distribution, which is often not the case. The inaccuracy resulting from this assumption being wrong is acceptable for high acceptable probabilities of violating the threshold, i.e., quality assessment. For reliability analysis, or safety assessment, when the probability of failure is typically one out of a million or even lower, specialized algorithms need to be applied [18]. Typical result is the probability of failure (Pf) of the system subjected to random input, given a simple threshold value for one KPI or a more complex limit state function.

When the "gold standard" Monte Carlo method shall be applied to safety assessment, one needs roughly 100/Pf simulation runs to obtain significant, trustworthy results. This makes this method infeasible for practical applications. Ansys optiSLang offers a variety of advanced algorithms for reliability analysis, which are proven to reduce the computational effort as compared to Monte Carlo by several orders of magnitude. This could be demonstrated also in the context of ADAS safety evaluation, with a reduction of the number of scenarios required by a factor 1000 [19].

7 Next Steps

This deliverable is a first version. In August 2022, the regulation EU 2022/1426 [1] defines the procedures and technical specifications for the type-approval of automated driving system of fully automated vehicles. This regulation, also called *ADS act*, specifies a framework to define and use scenarios in virtual tests or track tests (see Figure 29).

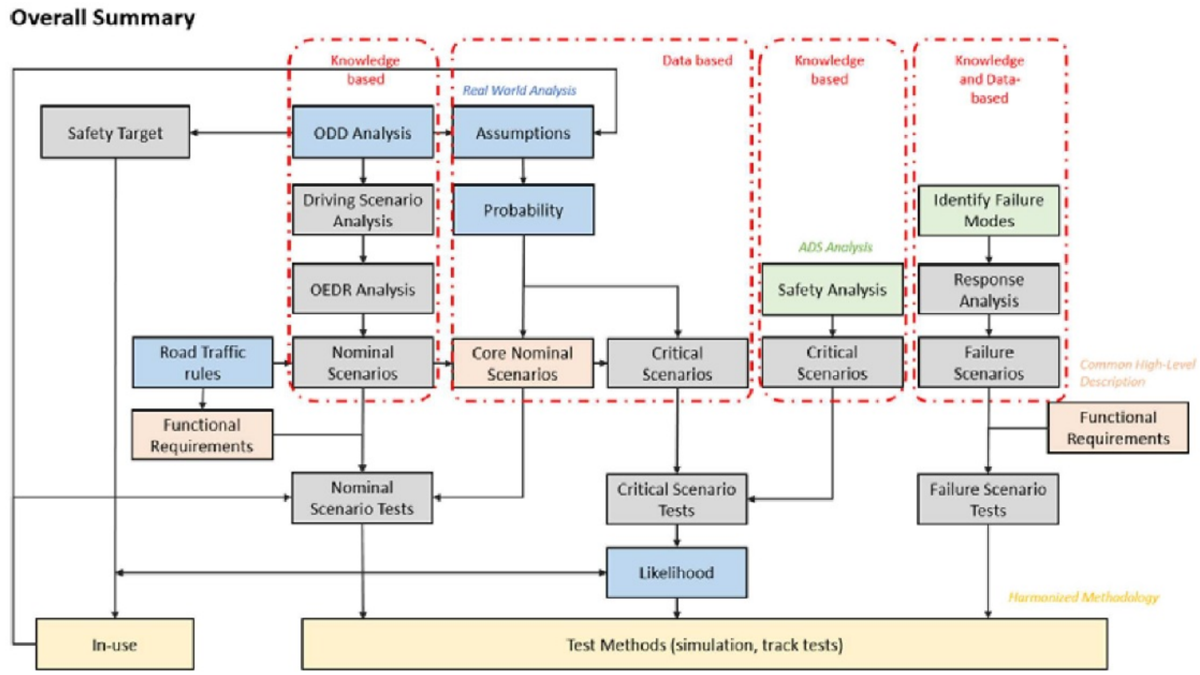

Figure 29: Principles to be followed to derive scenarios relevant for the ODD of the ADS – Regulation EU 2022/1426 ([1]) – annex III part 1 appendix 1

It also defines principles to assess the credibility of a virtual toolchain for an ADS validation in annex III part 4.

All these principles and requirements shall be used in the next version of this deliverable, including the findings of the recently published methodological report of the DGITM ([4]).

On the implementation scope:

- Ansys optiSLang will be used in a PoC with the Valeo toolchain (which includes a driving simulator and an ego car Level 3 control algorithm) for scenario variation. The objective will be to assess the performance of the toolchain for:

  - variation of the scenario variables
  - combinatorial management to calculate the probability of failure
  - analysis tools for Sensitivity, Robustness & Reliability analysis

- IRT SystemX's MOSAR will be used in order to assess scenario management for PoC in PRISSMA. Recent efforts include the exploration of the PoC led by UGE - *Université Gustave Eiffel*. The objective being to integrate scenarios regarding this PoC in MOSAR, to use the fundamentals structured in the MOSAR methodology in order to generate and manage scenarios that will later be enriched by UGE for simulation purposes.

General aspects that should be addressed in the task include the analysis of the minimization of the combinatorial explosion resulting from parameters variability (Combinatorial Management) and, to this end, the inclusion of SOTIF-related concepts into the methodology.

REFERENCES

[1] European Commission, Directorate-General for Internal Market, Industry, Entrepreneurship and SMEs, "Regulation (eu) 2022/1426 laying down rules for the application of regulation (eu) 2019/2144 of the europe-an parliament and of the council as regards uniform procedures and technical specifications for the type-approval of the automated driving system (ads) of fully automated vehicles," 2022. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32022R1426

[2] World Forum for the Harmonization of Vehicle Regulations, "(GRVA) new assessment/test method for automated driving (NATM) master document," UNECE, Tech. Rep., 2021.

[3] PRISSMA Project, "Deliverable 1.4 - tests and audit requirements - initial report," PRISSMA Task 2.4, Tech. Rep., 2022.

[4] DGITM/SAGS/EP, "Safety demonstration of automated road transport systems (ARTS): Expected contributions of the driving scenarios. methodologixal report," 2022.

[5] ISO Central Secretary, "Road vehicles - Safety of the intended functionality. Standard ISO 21448:2022," 2022.

[6] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," IEEE Access, vol. 8, pp. 87 456–87 477, 2020.

[7] Q. Zhang, D. K. Hong, Z. Zhang, Q. A. Chen, S. Mahlke, and Z. M. Mao, "A systematic framework to identify violations of scenario-dependent driving rules in autonomous vehicle software," Proc. ACM Meas. Anal. Comput. Syst., vol. 5, no. 2, jun 2021. [Online]. Available: https://doi.org/10.1145/3460082

[8] E. de Gelder, J. P. Paardekooper, A. K. Saberi, H. Elrofai, O. O. d. Camp., S. Kraines, J. Ploeg, and B. De Schutter, "Towards an ontology for scenario definition for the assessment of automated vehicles: An object-oriented framework," 2020. [Online]. Available: https://arxiv.org/abs/2001.11507

[9] d. B. N. de Gelder E, den Camp OO, "Scenario Categories for the Assessment of Automated Vehicles: Expected contributions of the driving scenarios. Methodological Report," 2020. [Online]. Available: https://cetran.sg/wp-content/uploads/2020/01/REP200121_Scenario_Categories_v1.7.pdf

[10] I. Urbieta, M. Nieto, M. García, and O. Otaegui, "Design and implementation of an ontology for semantic labeling and testing:automotive global ontology (ago)." Applied Sciences, vol. 11, pp. 1–19, August 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/17/7782

[11] PRISSMA Project, "Deliverable 2.2 - state of the art - intermediate report," PRISSMA Task 2.1, Tech. Rep., 2022.

[12] L. Marsso, R. Mateescu, and W. Serwe, "TESTOR: A modular tool for on-the-fly conformance test case generation," in Tools and Algorithms for the Construction and Analysis of Systems - 24th International Conference, TACAS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings, Part II, ser. Lecture Notes in Computer Science, D. Beyer and M. Huisman, Eds., vol. 10806. Springer, 2018, pp. 211–228. [Online]. Available: https://doi.org/10.1007/978-3-319-89963-3_13

[13] M. Utting, A. Pretschner, and B. Legeard, "A taxonomy of model-based testing approaches," Softw. Test. Verif. Reliab., vol. 22, no. 5, p. 297–312, aug 2012. [Online]. Available: https://doi.org/10.1002/stvr.456

[14] V. Ledoux, "Véhicules autonomes : situations d'interaction accidentogènes," 2019. [Online]. Available: https://surca.univ-gustave-eiffel.fr/fileadmin/contributeurs/SURCA/WP2Livrable/jfsr19_v-ledoux_identification-de-scenarios-dinteractions-accidentogenes.pdf

[15] Myers, R.H. and Montgomery, D.C., "Response surface methodology: Product and process op-timization using designed experiments," 2002.

[16] D. E. Huntington and C. S. Lyrintzis, "Improvements to and limitations of latin hypercube sampling," Probabilistic Engineering Mechanics, vol. 13, pp. 245–253, 1998.

[17] A. Saltelli, Global sensitivity analysis: the primer. John Wiley, 2008. [Online]. Available: http://books.google.at/books?id=wAssmt2vumgC

[18] C. G. Bucher, "Computational analysis of randomness in structural mechanics: Structures and infrastructures book series, vol. 3," 2009.

[19] Rasch M. et al., "Simulative validation of automated driver assistance systems using reliability analysis. weimar optimization and stochastic days wosd." 2019.