# Towards an Ontology for Scenario Definition for the Assessment of Automated Vehicles An Object-Oriented Framework

De Gelder, Erwin; Paardekooper, Jan Pieter; Khabbaz Saberi, Arash; Elrofai, Hala; Camp, Olaf op den; Kraines, Steven; Ploeg, Jeroen; De Schutter, Bart

**DOI**
10.1109/TIV.2022.3144803

**Publication date**
2022

**Document Version**
Final published version

**Published in**
IEEE Transactions on Intelligent Vehicles

# Towards an Ontology for Scenario Definition for the Assessment of Automated Vehicles: An Object-Oriented Framework

Erwin de Gelder, Jan-Pieter Paardekooper, Arash Khabbaz Saberi, Hala Elrofai, Olaf Op den Camp, Steven Kraines, Jeroen Ploeg, and Bart De Schutter, *Fellow, IEEE*

***Abstract*—The development of new assessment methods for the performance of automated vehicles is essential to enable the deployment of automated driving technologies, due to the complex operational domain of automated vehicles. One contributing method is scenario-based assessment in which test cases are derived from real-world road traffic scenarios obtained from driving data. Given the complexity of the reality that is being modeled in these scenarios, it is a challenge to define a structure for capturing these scenarios. An intensional definition that provides a set of characteristics that are deemed to be both necessary and sufficient to qualify as a scenario assures that the scenarios constructed are both complete and intercomparable. In this article, we develop a comprehensive and operable definition of the notion of scenario while considering existing definitions in the literature. This is achieved by proposing an object-oriented framework in which scenarios and their building blocks are defined as classes of objects having attributes, methods, and relationships with other objects. The object-oriented approach promotes clarity, modularity, reusability, and encapsulation of the objects. We provide definitions and justifications of each of the terms. Furthermore, the framework is used to translate the terms in a coding language that is publicly available.**

Manuscript received January 11, 2022; accepted January 14, 2022. Date of publication January 25, 2022; date of current version July 12, 2022. This work was supported by the Centre of Excellence for Testing and Research of Autonomous Vehicles at NTU (CETRAN), Singapore. *(Corresponding author: Erwin de Gelder.)*

Erwin de Gelder is with the Department of Integrated Vehicle Safety, TNO, 5708 JZ Helmond, The Netherlands and also with the Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: erwin.degelder@tno.nl).

Jan-Pieter Paardekooper is with the Department of Integrated Vehicle Safety, TNO, 5708 JZ Helmond, The Netherlands and also with the Donders Institute for Brain, Cognition and Behaviour, Radboud University, 6525 XZ Nijmegen, The Netherlands (e-mail: jan-pieter.paardekooper@tno.nl).

Arash Khabbaz Saberi is with TomTom, Automated Driving Product Unit, 5656 Eindhoven, The Netherlands (e-mail: arash.saberi@tomtom.com).

Hala Elrofai is with Mobile Perception Systems Lab, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands (e-mail: hala.elrofai@tno.nl).

Olaf Op den Camp is with the Department of Integrated Vehicle Safety, TNO, 5708 JZ Helmond, The Netherlands (e-mail: olaf.opdencamp@tno.nl).

Steven Kraines is with Symphony Company, Tokyo 732-0068, Japan (e-mail: steven@kraines.net).

Jeroen Ploeg is with the 2getthere, 3543 AE Utrecht, The Netherlands and also with the Department of Mechanical Engineering Dynamics and Control Group, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands (e-mail: j.ploeg@tue.nl).

Bart De Schutter is with the Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: b.deschutter@tudelft.nl).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2022.3144803.

Digital Object Identifier 10.1109/TIV.2022.3144803

***Index Terms*—Automatic testing, autonomous vehicles, intelligent vehicles, object-oriented modeling, performance evaluation, system testing, vehicle safety.**

## I. Introduction

An essential aspect in the development of Automated Vehicles (AVs) is the assessment of quality and performance aspects of the AVs, such as safety, comfort, and efficiency [1]–[8]. For legal and public acceptance of AVs, a clear definition of system performance is important, as well as quantitative measures for system quality. According to [2], traditional methods for evaluating driver assistance systems, such as [9], [10], cannot sufficiently assess quality and performance aspects of an AV, because they would require too many resources. A scenario-based approach could be a viable way to perform the AV assessment [6], [8], [11]. For a scenario-based assessment, proper specification of scenarios is crucial because

- scenarios provide the basis and justification for the tests used for the scenario-based assessment [4], [6], [12]–[15],
- it helps to arrive at an unambiguous description of scenarios that is crucial for providing standardized, repeatable, and reproducible tests [12],
- standardized descriptions of scenarios can be more easily compared and classified automatically [16],
- properly specified scenarios are the basis for evaluating the coverage of the assessment [6], and
- properly specified scenarios enable us to translate the result of a test into an assessment of the AV performance with regards to a particular Operational Design Domain (ODD) [17], [18].

Although the notion of scenario is frequently used in the context of automated driving [5]–[7], [15], [19]–[23], only rarely is an explicit definition actually given. Furthermore, even those definitions are unclear because of ambiguities and the use of other undefined terms. From the implementation perspective, describing scenarios unambiguously becomes more important given the many simulators that are recently being introduced [24]–[28]. To this end, there are several file formats and methods for defining scenarios for the assessment of AVs, such as OpenSCENARIO [29] and CommonRoad [30]. Because the focus of these implementations is on scenarios that can be simulated, these implementations describe scenarios at a quantitative level and, consequently, they do not provide concepts

for a qualitative description of a scenario. Furthermore, these implementations and other object-oriented approaches used in the field of the assessment of AVs [31]–[34] mostly lack the definitions and justifications of each of the terms.

In this work, as a starting point for developing a full ontology of scenarios, we propose a novel Object-Oriented Framework (OOF) that addresses the aforementioned shortcomings. To avoid ambiguities in the definitions, we provide intensional definitions for concepts corresponding to scenarios and all of their essential building blocks (such as activities, actors, and events). These intentional definitions give the meaning of the concepts by specifying necessary and sufficient conditions for when the concepts should be used. We base the definitions of each of the components on definitions that are commonly used in the field of the safety assessment of AVs [13], [14], [29], [35]–[37]. While being broadly consistent with existing definitions [13], [14], [38], this framework aims to be sufficiently explicit to enable the formalization of a scenario description. More specifically, because we give the characteristics of the concepts corresponding to scenarios and specify how those concepts interrelate, we can define the scenario components as objects of classes having attributes, methods, and relationships with objects that are members of other classes. In addition to the definition of a scenario, we introduce the concept of a *scenario category* that is used to qualitatively describe scenarios, i.e., an abstraction of a scenario. Scenario categories enable the categorization of scenarios in terms of the categories of their typical components. The presented OOF provides explicit guidelines for the construction of scenario descriptions that are able to effectively assess the AV performance.

The proposed approach brings several benefits. First, we provide concepts for a qualitative description of a scenario, which is useful because it enables to classify scenarios and to interpret scenarios. Second, the OOF allows for reusing and maintaining (the building blocks of) a scenario as well as performing operations on and interacting with (the building blocks of) a scenario. Third, our framework is supported with the definitions and justifications of each of the concepts. Fourth, the framework enables the translation of the concepts and their relationships into object-oriented code. This, in turn, is used to describe scenarios in a coding language that can be understood by various software agents, such as simulation tools, and that can be ported to already available formats like OpenSCENARIO [29].

To illustrate how to use the presented OOF, we have implemented the framework in a coding language that is publicly available at https://github.com/ErwindeGelder/ScenarioDomainModel.[1] This link contains real-life applications of the presented OOF, such as describing scenarios extracted from data [39]. The framework is also used as a schema for a database system for storing scenarios and scenario categories. Such a database can be used to perform scenario-based assessment of AVs[2] [40]. To further illustrate the use of the OOF, this article provides an example with a real-world case in which a vehicle approaches a pedestrian crossing. The proposed OOF provides a first step towards an ontology [41] for scenarios for the assessment of AVs. In a subsequent study, the formalized concepts presented in this article will be used to design an ontology with logical constraints that enable a computer to perform reasoning on scenarios.

The outline of the article is as follows. In Section II, we explain why an OOF is useful and what the context is. We define the notions of *scenario*, *event*, *activity*, and *scenario category* in Section III. The OOF that formalizes these definitions is presented in Section IV. In Section V, an application example is provided to illustrate the use of the framework with a real-world scenario. The article is concluded in Section VI.

## II. Background

In Section II-A, we explain why we want to present an OOF for describing scenarios and scenario categories. Section II-B provides information on the context for which we want to define scenarios.

### A. Why an Object-Oriented Framework?

According to Johnson and Foote [42], an OOF is a "set of classes that embodies an abstract design for solutions to a family of related problems." The object orientation is used for "a representation, modeling, and abstraction formalism" [43], which is why it is considered "not only useful but also fundamental" [43]. In addition, Patridge [44] notes that object-oriented modeling can provide a bridge from traditional entity-relation-based data modeling to data modeling that is fully grounded in a formalized ontology. An OOF offers the following benefits:

- *Clarity:* It provides "a common vocabulary for designers to communicate, document, and explore design alternatives" [45].
- *Modularity:* By decomposing a scenario into components, the complexity of a scenario itself is reduced. Thus, "modularity makes it easier to understand the effect of changes" [42].
- *Reusability:* An OOF promotes reusability [42], [46], [47]. For example, if two classes share certain procedures and/or properties, these procedures and/or properties could be provided by a so-called superclass from which these two classes inherit the procedures and properties, such that these procedures and properties need to be defined only once.
- *Encapsulation:* Encapsulation assures "that compatible changes can be made safely, which facilitates program evolution and maintenance" [46].
- *Possibility to translate to object-oriented programming languages:* As the framework consists of a set of classes, it can be directly used in an object-oriented coding language. The framework then specifies the relationships between the different classes and provides information on the properties of a class and the possible values.

[1] As a coding language, Python is used. The code implementation also contains more methods than presented in this article.

[2] An illustration of such an assessment is publicly available at https://github.com/ErwindeGelder/ScenarioDomainModel.

### B. Context of a Scenario

Because the notion of scenario is used in many different contexts outside of the domain of road traffic, a wide diversity in definitions of this notion exists (for an overview, see [48], [49]). Therefore, it is reasonable to assume that "there is no [generally] 'correct' scenario definition" [48]. As a result, to define the notion of scenario, it is important to consider the context in which it will be used.

In this article, the context of a scenario is the assessment of AVs, where AVs refer to vehicles equipped with a driving automation system[3]. It is assumed that the assessment methodology uses scenario-based test cases. The ultimate goal is to build a database with all relevant scenarios that an AV has to cope with when driving in the real world [6]. Hence, a scenario should be a description of a potential use case of an AV.

## III. Definitions

One of the main reasons to introduce an OOF is to enable sharing of knowledge between researchers, developers, and users. Therefore, it is important that the terms we use are clearly defined. When presenting our OOF in Section IV, we will formalize the terms such that they can be used by software agents. In this section, we define the terms *scenario*, *event*, *activity*, and *scenario category*, thereby providing insight into the terms used in the next section. We aim to provide intensional definitions that are in accordance with the common use of these terms in the literature and to provide clarity on what are the necessary and sufficient conditions for when the term should be used.

We first define the concept of a scenario in Section III-A. Next, we define two important components of a scenario: events and activities, in Sections III-B and III-C, respectively. Lastly, we present the definition of a scenario category in Section III-D. Each of the Sections III-A to III-D starts with background information. Next, we draw conclusions that lead to our proposed definition of the corresponding term. After proposing a definition, each section finishes with remarks and implications of the proposed definition. For the definitions provided in Sections III-A to III-D, use is made of the terms listed in Table I. The definitions in Table I are mostly based on literature; see Appendix A for more details.

TABLE I
Terms and Definitions That are Used in Section III. For More Details, See Appendix A

| Term | Definition |
|---|---|
| Ego vehicle | Vehicle from which the world is perceived and/or vehicle that must perform a certain task during a test |
| Physical element | Object that exists in the three-dimensional space |
| Actor | Physical element that experiences change<br>Note: An actor is a physical element, but a physical element is not necessarily an actor. |
| Static environment | Part of the environment that does not change |
| Dynamic environment | Part of the environment that does change and that comprises all actors |
| Act | Combination of an actor and an activity |
| State variables | Description of the present configuration of a system that can be used to determine the future response, given the excitation inputs and the equations describing the dynamics |
| State vector | Vector containing all $n$ state variables |
| Model | Differential and algebraic equations that describe the dynamics |
| Mode | Period in which a system does not exhibit a sudden change in an input, a model parameter, or the model |

### A. Scenario

Go and Carroll [51] describe a scenario within the field of system design. They define a scenario as "a description that contains (1) actors, (2) background information on the actors and assumptions about their environment, (3) actors' goals or objectives, and (4) sequences of actions and events. Some applications may omit one of the elements or they may simply or implicitly express it. Although, in general, the elements of scenarios are the same in any field, the use of scenarios is quite different."

Geyer *et al.* [14] describe a scenario within the context of automated driving. They use the metaphor of a movie or a storybook for describing a scenario and state that "a scenario includes at least one situation within a scene including the scenery and dynamic elements. However, [a] scenario further includes the ongoing activity of one or both actors." Geyer *et al.* [14] define a scene "by a scenery, dynamic elements, and optional driving instructions." In Geyer *et al.* [14], the meaning of activity is not detailed.

Ulbrich *et al.* [13] define a scenario as "the temporal development between several scenes in a sequence of scenes. Every scenario starts with an initial scene. Actions & events as well as goals & values may be specified to characterize this temporal development in a scenario. Other than a scene, a scenario spans a certain amount of time." The authors of [13] state that actions and events link the different scenes. A further description of actions and events is not given in [13].

Another definition of a scenario in the context of automated driving is given by Elrofai *et al.* [38]. They define a scenario as "the combination of actions and maneuvers of the host vehicle in the passive [i.e., static] environment, and the ongoing activities and maneuvers of the immediate surrounding active [i.e., dynamic] environment for a certain period of time."

Saigol *et al.* [36] define a scenario as "a description of a short interaction between an AV and other road users and/or road infrastructure".

In a concept paper on OpenSCENARIO 2.0 [52], a scenario is defined as "a 'description of the temporal development' of road users (actor entities) defined by their actions, where temporal activation (defining when) 'is regulated by' conditional 'triggers'. A scenario comprises both scenery and dynamic elements."

As a basis for constructing a comprehensive definition for the concept of scenario, we list the major characteristics contained in the above definitions as follows:

1) *A scenario corresponds to a time interval.* The aforementioned definitions [13], [14], [38], [51] state that a scenario

[3] According to [50], a driving automation system is "the hardware and software that are collectively capable of performing part or all of the dynamic driving task on a sustained basis. This term is used generically to describe any system capable of level 1-5 driving automation." Here, level 1 driving automation refers to "driver assistance" and level 5 refers to "full driving automation". For more details, see [50].

corresponds to a time interval. Van Notten *et al.* [48] call such a scenario a chain scenario ("like movies"), as opposed to a snapshot scenario, i.e., a scenario that describes the state at a given time instant ("like photos").

2) *A scenario consists of two or more events [13], [14], [48], [51], [53].* It can be helpful to develop scenarios using events [49]. Thus, a scenario could be defined as a particular sequence of events or, as Kahn [53, p. 143] writes, "a scenario results from an attempt to describe in more or less detail some hypothetical sequence of events". Furthermore, Geyer *et al.* [14] and Ulbrich *et al.* [13] use the notion of event for describing a scenario, although they do not provide a definition of the term *event*. Because a scenario contains at least a start event and an end event, the minimum number of events is two. In Section III-B, we will elaborate on the notion of *event*.
3) *Real-world traffic scenarios are quantitative scenarios.* Regarding the nature of the data, a scenario can be either qualitative or quantitative [48]. For a real-world traffic scenario to be suitable for simulation purposes, it must be described quantitatively. A scenario, however, can also be described qualitatively, such that it is readable and understandable for human experts. Providing a qualitative description of a quantitative scenario has become known as a story-and-simulation approach [54]. Note that a qualitative description of a scenario does not uniquely define a quantitative scenario. A qualitative description can be regarded as an abstraction of the quantitative scenario, see also Section III-D.
4) *The time interval of a scenario contains all relevant events.* According to [14], "the end of a scenario is defined by the first irrelevant situation with respect to the scenario". In a similar manner, we require that the time interval of a scenario should contain all relevant events. Note that 'relevant' is subjective and, therefore, an event is considered to be relevant with respect to the perspective of one or more of the participating actors, often called the "ego vehicle".
5) *A scenario includes the description of the environment.* A scenario should include the description of the static and dynamic environment. Although the description of the static environment is not a general prerequisite of a scenario, this is often included when speaking about traffic scenarios [13], [14], [19], [30], [38]. The static environment consists of all relevant[4] physical elements that do not undergo relevant changes with respect to the ego vehicle(s) within the time interval between the start and the end of the scenario. The dynamic environment consists of all relevant actors that undergo changes that are relevant to the ego vehicle(s). For example, the road may be part of the static environment, but if the change in the road temperature is relevant to the ego vehicle(s), the road is part of the dynamic environment.
6) *A scenario includes at least one ego vehicle [14], [38].* Because of the two previously mentioned characteristics, a scenario is required to include at least one ego vehicle. Note that an ego vehicle is often regarded as the device under test. In this article, however, this is not necessary because the ego vehicle is just the vehicle whose perspective is used to define what is relevant in the scenario.
7) *A scenario describes the goals or activities of the actors.* Either the activities, the goals, or a combination of activities and goals are required to determine how each actor in a scenario responds to specific events. Note that this also holds for the ego vehicle since the ego vehicle is an actor. When describing a scenario using real-world data, goals do not need to be given; e.g., Elrofai *et al.* [38] mention the activities of the actors rather than the goals. When describing a scenario that an AV has to cope with, however, the ego vehicle's goals (i.e., its driving mission [14]) could be specified rather than its activities [13]. Note that if the activities of an actor are described rather than its goals, an observer might not be able to determine whether the actor has successfully responded to the scenario.

Hence, we define a scenario as follows:

*Definition 1 (Scenario):* A scenario is a quantitative description of the relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, the dynamic environment, and all events that are relevant to the ego vehicle(s) within the time interval between the first and the last relevant event.

When applying Definition 1 in an OOF, it is possible to give the "description" of a component of a scenario simply by providing a reference to that component. A reference could be, e.g., the full name of a file, a pointer pointing to a specific part of the computer memory, or an identifier that addresses a specific entry in a database. The advantage of references is that these parts of the scenario can be exchanged across different scenarios, as these scenarios can use the same references. As an example, an OpenSCENARIO file allows to provide a reference to an OpenDRIVE file, which describes a road network [55]. As we will see in Section IV, in our proposed framework, a scenario may contain references to physical elements, activities, actors, and events.

### B. Event

As mentioned in Definition 1, a scenario consists of events. The term event is used in many different fields, e.g.:

- In computing [56], an event is an action or occurrence recognized by software. A common source of events are inputs by the software users. An event may trigger a state transition.
- In probability theory, an event is an outcome or a set of outcomes of an experiment [57]. For example, a thrown coin landing on its tail is an event.
- In the field of hybrid systems theory, "the continuous and discrete dynamics interact at 'event' or 'trigger' times when the continuous state [vector] hits certain prescribed sets in the continuous state space" [58]. Moreover, "a hybrid

[4]The term 'relevant' is subjective and depends on the use of the scenario. The composer of a scenario typically judges whether something might be relevant for the scenario.

system can be in one of several modes, [...], and the system switches from one mode to another due to the occurrence of events" [59].

- In the ISO 15926-2 standard, an ontology for long-term data integration, access, and exchange is specified in which an event is defined as "a *possible_individual*[5] with zero extent in time, which means that it occurs at an instant in time" [60].
- In event-based control, a control action is computed when an event is triggered, as opposed to the more traditional approach where a control action is periodically computed [61]. In event-based control, the event is triggered at the moment at which the system (is about to) reach a certain threshold.

Before providing the definition of an event, the following is concluded about an event, based on the aforementioned literature:

1) *An event corresponds to a time instant.* For the definition of event, we consider a hybrid-systems setting with a linear-time model [62]. Therefore, an event happens at some time instant.
2) *An event marks a mode transition or the moment a system reaches a threshold.* A mode transition may be induced by either an abrupt change of an input signal, a change of a parameter, a change in the model, or an external cause. It is also possible that the event marks the moment that a system reaches a threshold.

Hence, we define an event as follows:

*Definition 2 (Event):* 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.

Definition 2 indicates that the moment of an event can be defined in two different ways: (1) by a mode transition or (2) by the system reaching a threshold. The first type could be a mode transition caused by a sudden driver input. An event might also be induced by an external cause, such as an environmental change. The second type of event, i.e., related to the system reaching a threshold, is especially useful when describing test scenarios. For example, consider the ego vehicle approaching a pedestrian that is about to cross the road [63]. Here, the event marks the moment that the distance between the vehicle and pedestrian is less than $d_{\mathrm{v,p}}$ meters. At the moment of this event, the pedestrian starts to cross the road such that the vehicle would impact with the pedestrian if it would not change its speed or direction [63]. By using a variable threshold $d_{\mathrm{v,p}}$, the value is flexible and can be set differently to define multiple scenarios.

For the practical implementation of events, a set of conditions may be specified. In that case, the event occurs at the moment that the conditions are met. In [29], an extensive list of possible conditions that can be used to define an event is given. For example, a condition could be that the distance between the vehicle and the pedestrian is below a certain threshold.

*Remark 1:* Ulbrich *et al.* [13] and Geyer *et al.* [14] use the term *scene* to define a scenario. Like an event, we consider a scene to correspond to a temporal snapshot of the entire scenario. A scene can be obtained by taking a temporal cross-section of the entire scenario as described in Definition 1.

[5]"An entity that exists in space and time" [60].

### C. Activity

To describe the dynamic environment of a scenario, activities are used. A scenario may also describe the activities of the ego vehicle. Both the terms activity [11], [14], [35], [37], [64] and action [13], [14], [65] are used in the context of automated driving. Although, strictly speaking, the terms action and activity have a slightly different meaning, they are often used for the same purpose:

- According to [13], actions may be specified for characterizing the temporal development in a scenario.
- Elrofai *et al.* [11] consider an activity as a building block of the dynamic part of the scenario: "An activity is a time evolution of state variables such as speed and heading to describe for instance a lane change, or a braking-to-standstill."
- In a glossary for scenario catalog development [35], an activity is defined as "the state [vector] of an object over an interval of time. An activity starts with an event and ends with another event."
- In the ISO 15926-2 standard, an activity is defined as "a *possible_individual* that brings about change by causing the *event* that marks the *beginning*, or the *event* that marks the *ending* of a *possible_individual*" [60].

Before providing the definition of an activity, the following is concluded about an activity based on the aforementioned literature:

1) *An activity corresponds to an inter-event time interval.* As opposed to an event, an activity spans a certain time interval. Furthermore, the start and the end of an activity are marked by an event.
2) *An activity quantitatively describes the time evolution of one or more state variables.* Because activities are building blocks of a scenario and a scenario corresponds to a quantitative description, the activities themselves need to be quantitative as well. Therefore, an activity describes the time evolution of one or more state variables, i.e., the trajectory of one or more state variables over an inter-event time interval that corresponds to the activity, where the term state variable is defined in Table I.
3) *An activity is performed by an actor.* An activity describes the time evolution of one or more state variables and a state variable corresponds to an actor, e.g., the acceleration of a vehicle.

Hence, we define an activity as follows:

*Definition 3 (Activity):* An activity is a quantitative description of the time evolution of one or more state variables of an actor between two events.

As an example, an activity could describe the longitudinal acceleration (or, e.g., speed) during an acceleration or deceleration. Activities describing the lateral position of a vehicle with respect to the center of the corresponding lane might, e.g., be labeled with "driving straight" or "changing lane".

### D. Scenario Category

According to Definition 1, a scenario in the context of the performance assessment of an AV needs to be quantitative. However, in the literature, the term scenario is also used to refer to a collection of scenarios, where this collection of scenarios is described qualitatively. For example, in [66], a typology of pre-crash scenarios is proposed. Here, each of the pre-crash scenarios is an abstraction of many quantitative scenarios. Similar studies have been performed to describe scenarios that lead to highway accidents [67], car-cyclist accidents [68], and car-pedestrian accidents [69]. In [70], a taxonomy of scenarios is proposed to qualitatively describe challenging scenarios for automated driving. In [23], a distinction is made between so-called functional scenarios, abstract scenarios, logical scenarios, and concrete scenarios. These four types of scenario descriptions represent different levels of abstraction with functional scenarios referring to non-formal human-readable scenarios, abstract scenarios referring to formalized declarative descriptions, logical scenarios referring to parameterized scenarios with ranges and distributions of the parameters, and concrete scenarios referring to parameterized scenarios with fixed parameters values.

The aforementioned references [23], [66]–[70] show that the term *scenario* is also used to address qualitative descriptions. Since we define a scenario as a quantitative description, we need to introduce a different term to address the qualitative description. We propose to use the term *scenario category* to refer to the qualitative description of a scenario. A qualitative description can be regarded as an abstraction of a quantitative scenario, whereas a quantitative description can be regarded as a concretization of a qualitative description.

We thus define a scenario category as follows:

*Definition 4 (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.

Introducing the concept of scenario categories brings the following benefits:

- For a human, it is often easier to interpret a qualitative description than a quantitative description.
- Scenarios that have something in common can be grouped together, which enables characterization of types of scenarios and facilitates discussion of scenarios.
- The completeness of a set of scenarios can be assessed by considering the completeness of scenario categories (see, e.g., [71]) and the completeness of scenarios in each category (see, e.g., [72]).

We describe the formal relation between a scenario and a scenario category with the verb "to comprise," denoted by $\ni$. If a specific scenario category $\mathcal{C}$ is an abstraction of a specific scenario $S$, then we say that $\mathcal{C}$ comprises $S$, or simply $\mathcal{C} \ni S$. A given scenario category can comprise multiple scenarios and multiple scenario categories can comprise a specific scenario. As a consequence, as opposed to the proposed categorization of scenarios in [66], [68], [69], [73], scenario categories do not need to be mutually exclusive.

The verb "to include" is used to describe the relation between two scenario categories. A scenario category $\mathcal{C}_2$ is said to include a scenario category $\mathcal{C}_1$ if $\mathcal{C}_2$ comprises all scenarios that are comprised in $\mathcal{C}_1$. In that case, we can write $\mathcal{C}_2 \supseteq \mathcal{C}_1$. Thus we have

$$\mathcal{C}_2 \supseteq \mathcal{C}_1 \text{ if } \mathcal{C}_2 \ni S \; \forall \; \{S : \mathcal{C}_1 \ni S\}. \tag{1}$$

We propose to provide scenarios and scenario categories with additional information in the form of tags. A tag is a keyword or a keyphrase that provides extra information on a piece of data [74]. For example, items in a database can contain some tags that enable users to quickly retrieve several items that share a certain characteristic described by a tag [75]. The use of these tags brings several benefits:

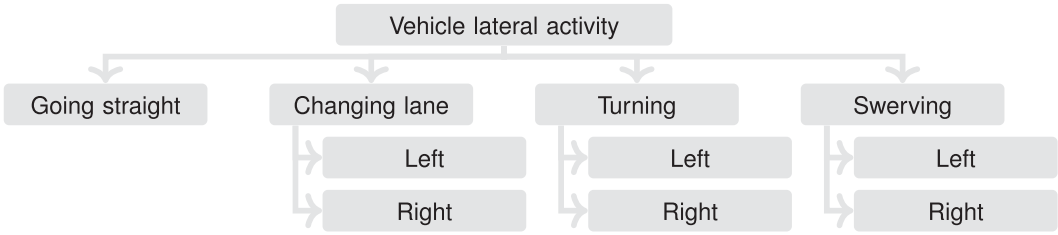
- The tags of a scenario can be helpful in determining which scenario categories do and do not comprise the scenario.
- It is easy to select scenarios from a scenario database or a scenario library by using tags or a combination of tags.

There is a balance between having generic scenario categories — and thus a wide variety among the scenarios comprised by the scenario category — and having specific scenario categories without much variety among the scenarios comprised by the scenario category. For some systems, one may be interested in a very specific set of scenarios, while for another system one might be interested in a set of scenarios with a high variety. To accommodate this, tags can be structured in hierarchical trees [76]. The different layers of the trees can be regarded as different abstraction levels [77].

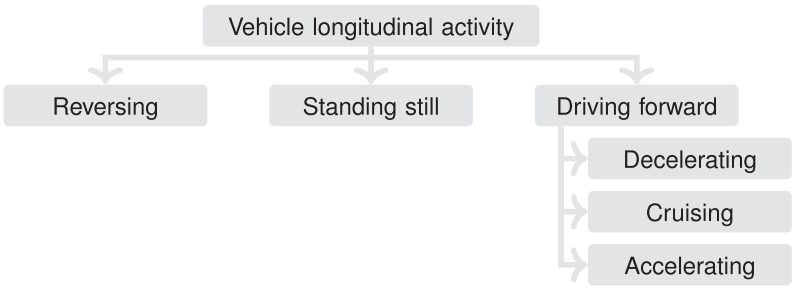
Fig. 1 shows two examples of trees of tags taken from [16]. These tags describe possible activities of a vehicle, i.e., the lateral motion control (via steering) and longitudinal motion control (via acceleration and deceleration). The tags may refer to the objective of an actor in case no activities are defined. For example, a test case in which the ego vehicle's objective is to make a left turn, the tags "Turning" and "Left" are applicable. Note that tags may be used not only to classify vehicle behavior, but also traffic and environment situations, e.g., "cut-in" or "heavy rain".

## IV. Object-Oriented Framework for Scenarios

We have already explained the use of an OOF in Section II-A. In this section, we present our OOF for scenarios for the assessment of AVs. The overview of the framework is formally represented through class diagrams that are briefly presented in Section IV-A. Next, Section IV-B explains how a scenario category is formally represented in our framework. Similarly, in Section IV-C, we describe how a scenario is formally represented. The OOF can be implemented straightforwardly in object-oriented languages such as C++ and Python, since these languages support the definition of classes, the instantiation of objects from those classes, and concepts such as inheritance and aggregation. An actual implementation of the OOF in a coding language is publicly available at https://github.com/ErwindeGelder/ScenarioDomainModel. This link also contains tutorials for the technical application of the OOF.




(a) Lateral activities of a vehicle.




(b) Longitudinal activities of a vehicle.

Fig. 1. Tags for lateral and longitudinal activities of a vehicle [16]. The lateral activity is relative to the lane in which the corresponding vehicle is driving.

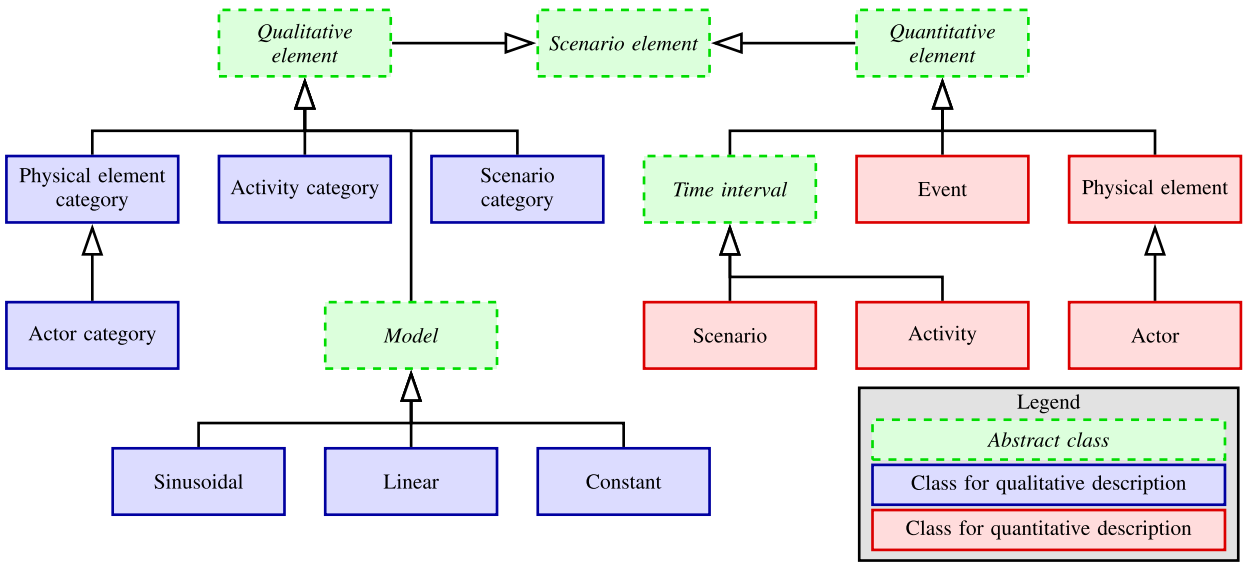



Fig. 2. Class-level relationships of most classes of our Object-Oriented Framework (OOF).

### A. Class Diagrams

In Figs. 2 and 3, the blue solid blocks represent the classes[6] that are used to describe a scenario category according to Definition 4 and the red solid blocks represent the classes that are used to describe a scenario according to Definition 1. The green dashed blocks represent so-called abstract classes. Abstract classes cannot be instantiated. Each class serves as a template for creating objects whereas an object of a particular class is referred to as the instance of that particular class.

[6]In the remainder of this paper, when referring to (an instance of) a class, italic font is used. Additionally, class names start with capital letters and instance names with lowercase letters.

Fig. 2 shows the class-level relationships while Fig. 3 shows the instance-level relationships. In Fig. 2, the arrow from, e.g., *Scenario* to *Time interval*, denotes that *Scenario* is a subclass of *Time interval*. Therefore, all properties of the *Time interval* are inherited by *Scenario*. The arrow with the diamond in Fig. 3 denotes an aggregation. This means that, e.g., an *actor*, which is an instance of the *Actor* class, has an *actor category* as an attribute. Here, the "1" at the start of the arrow from *Actor category* to *Actor* indicates that an *actor* has exactly one *actor category*. Similarly, "2" at the aggregation arrow from *Event* to *Time interval* indicates that a *time interval* contains two *events*, i.e., the events that define the start and the end of the time interval. A "$0, 1, \ldots$" at the start of an aggregation arrow

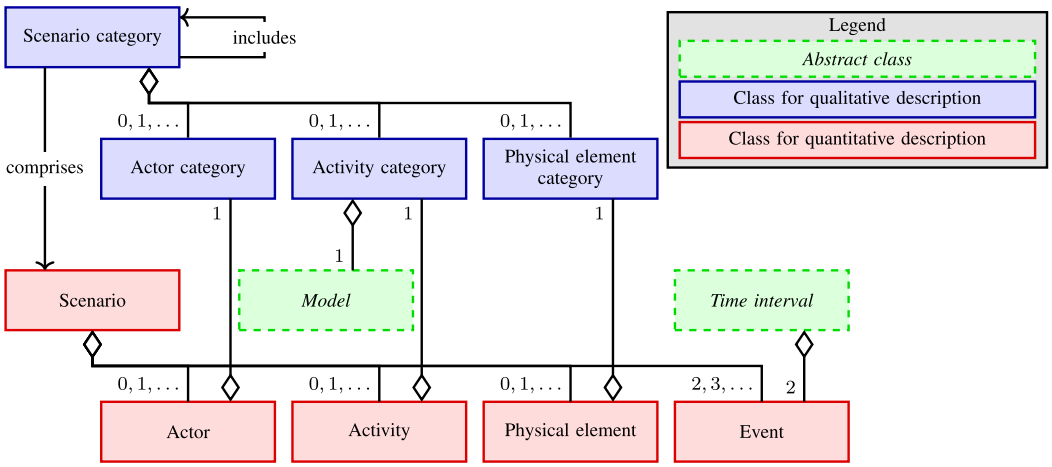



Fig. 3. Instance-level relationships of most classes of our Object-Oriented Framework (OOF).

indicates that an object has zero, one, or multiple objects of the corresponding class. The arrow with the text "comprises" and "includes" represent methods that are explained in Section III-D. Here, "comprises" can be denoted by $\ni$ and "includes" can be denoted by $\supseteq$, see (1).

### B. Scenario Category and its Attributes

Because all other classes in Fig. 2 are subclasses of *Scenario element*, these classes inherit the attributes and procedures of *Scenario element*. In our framework, a *scenario element* has a human-interpretable name, a unique ID, and possibly predefined tags that are also interpretable by a software agent. So, all other classes in Fig. 2 also have these attributes. In addition to these attributes, the *Qualitative element* class has a human-interpretable description.

The static environment is qualitatively described by one or more *physical element categories*. Because *physical element categories* qualitatively describe the static environment, they contain a human-interpretable description of the physical things they describe.

The ego vehicle(s) and the dynamic environment are qualitatively described by *activity categories* and *actor categories*. In line with Definition 3, *Activity category* includes the state variable(s). The *Model* that is used to describe the time evolution of the state variable(s) is specified. Note that *Model* is an abstract class that serves as a template for different models, such as the three examples shown in Fig. 2: *Sinusoidal*, *Linear*, and *Constant*. Let $z(t)$ denote the state variable at time $t$, then the *Sinusoidal* model is defined as follows:

$$\dot{z}(t) = \frac{\pi A}{2T} \sin\left(\frac{\pi\,(t - t_0)}{T}\right),\ t \in [t_0, t_0 + T], \tag{2}$$

$$z(t_0) = z_0. \tag{3}$$

Here, the amplitude ($A$), duration ($T$), initial time ($t_0$), and initial state ($z_0$) are parameters. The *Linear* and *Constant* models are described by the following equations, respectively:

$$\dot{z}(t) = s,\ z(t_0) = z_0, \tag{4}$$

$$z(t) = z_0. \tag{5}$$

The *Linear* model contains three parameters, i.e., the slope ($s$), initial time ($t_0$), and initial state ($z_0$). The *Constant* model only has the parameter $z_0$. Since an *activity category* is a qualitative description, the values of the parameters of its *model* are not part of the *activity category*. Note that this article only considers the models *Sinusoidal*, *Linear*, and *Constant*, but more complex models may be necessary to describe complex behavior. More complex models are out-of-scope of this article, but it is straightforward to extend the OOF with such models.

The *Actor category* is a subclass of *Physical element category* so *Actor category* inherits the properties of *Physical element category*. In addition, *Actor category* has an attribute that specifies the type of object. To indicate that an actor is an ego vehicle, the tag "Ego vehicle" is added to the list of tags of the *actor category*.

The *Scenario category* has *physical element categories*, *activity categories*, and *actor categories* as attributes. Another attribute of the *Scenario category* is the list of acts. These acts describe which actors perform which activities. Note that it is possible that one actor performs multiple activities and that one activity is performed by multiple actors.

The reader might wonder why we introduce the different classes for describing a scenario category, i.e., the blue blocks, instead of only one class for modeling a scenario category. The main advantage of the different classes is the reusability of the instances of the classes, because these instances can be exchanged among different *scenario categories*. For example, if two *scenario categories* have the same *actor categories*, we only need to define the *actor categories* once, whereas if the *actor categories* would not be instances of a class but only properties of the scenario category, we would need to define the *actor categories* twice.

### C. Scenario and its Attributes

To distinguish objects that are directly used to compose a *scenario*, these objects are instantiated from subclasses of the *Quantitative element* class. The class *Scenario* is a subclass of

*Time interval* and, therefore, it has *events* that define the start and the end of the scenario. The *Scenario* also has *physical element*, *activities*, *actors*, and *events* as attributes. The *physical elements*, *activities*, and *actors* are the quantitative counterparts of the *physical element categories*, *activity categories*, and *actor categories*, just as a *scenario* is the quantitative counterpart of a *scenario category*. As with the *Scenario category*, the *Scenario* contains a list of acts that describe which actors perform which activities.

A *physical element* has a *physical element category* and it may have multiple properties that quantitatively define the object, such as its size, weight, color, radar cross section, etc. Physical elements can be used to define, e.g., the road layout, static weather and lighting conditions, and infrastructural elements.

According to Definition 3, an activity quantitatively describes the evolution of one or more state variables in a time interval. The state variable(s) are defined by the *activity category* that the *activity* has as an attribute. Together with the *Model* that is contained by the *activity category*, the time evolution of the state variable is described by a set of parameters. The values of the parameters are part of the *activity*.

Following Definition 2, an *event* contains conditions that describe the threshold or mode transition at the time of the *event*.

Similar to a *physical element* and an *activity*, an *actor* has its qualitative counterpart — an *actor category* — as an attribute. Additionally, the *Actor* contains an initial state vector and a desired state vector, that can be used to specify the intent, as attributes. Describing the intent is especially useful for defining a test scenario in terms of the objective of the ego vehicle rather than its activities.

An advantage of having the qualitative counterparts of the *Physical element*, *Activity*, and *Actor* is that the qualitative description can be reused and exchanged. For example, there can be many different braking activities, but there needs to be only one *activity category* for qualitatively defining the braking activity. Here, it is assumed that all braking activities are modeled with the same model and that similar tags apply. If this is not the case, multiple *activity categories* need to be defined, but the number of *activity categories* will still be substantially lower than the number of *activities*.

## V. Example: Pedestrian Crossing

To illustrate the use of the OOF, we describe a scenario using objects of the classes presented in Section IV. The scenario is schematically shown in Fig. 4. The ego vehicle is driving on the right lane of a two-lane road and a pedestrian is walking on a footway that intersects the road the ego vehicle is driving on. Both the ego vehicle and the pedestrian are initially approaching the pedestrian crossing. The ego vehicle brakes and comes to a full stop in front of the pedestrian crossing. While the ego vehicle is stationary, the pedestrian crosses the road using the pedestrian crossing. When the pedestrian has passed the ego vehicle, the ego vehicle accelerates. The code of this example is publicly available[7].

[7]See https://github.com/ErwindeGelder/ScenarioDomainModel. The repository also contains other examples.

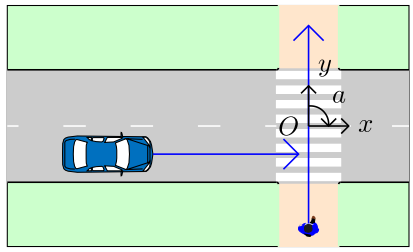



Fig. 4. Schematic overview of a scenario where both the ego vehicle and a pedestrian are approaching a non-signalized pedestrian crossing. The pedestrian has priority.

This particular scenario can be used to formulate a test scenario for the assessment of an AV. For example, when assessing a pedestrian automatic emergency braking system [63], we are interested in the behavior of the system in case the driver or automation system of the ego vehicle does not brake.

We first describe the scenario qualitatively using our proposed framework. Next, the scenario is described quantitatively in Section V-B. In Section V-C, we show which objects are reused and which objects are different if we consider an actual test scenario with a crossing pedestrian.

### A. Qualitative Description of the Pedestrian Crossing

To describe the scenario according to the presented domain model, objects are instantiated from the classes presented in Figs. 2 and 3. Fig. 5 shows the objects for describing the scenario qualitatively. There are two *actor categories:* one for the ego vehicle and one for the pedestrian. Four different *activity categories* are defined: *braking*, *stationary*, *accelerating*, and *walking straight*. The *braking*, *stationary*, and *accelerating activity categories* contain the state variable $v_{\mathrm{ego}}$, i.e., the speed of the ego vehicle, and use the *Sinusoidal* model of (2) and (3), the *Constant* model of (5), and the *Linear* model of (4), respectively. The *activity category walking straight* has the position of the pedestrian ($y_{\mathrm{ped}}$) as its state variable and uses the *Linear* model of (4).

The two *actor categories*, the four *activity categories*, and the *physical element category* that represents the crosswalk, are used by the *scenario category*. The *scenario category* has four acts. The first three acts assign the first three *activity categories* to the ego vehicle. The last act assigns the *activity category walking straight* to the pedestrian.

### B. Quantitative Description of the Pedestrian Crossing

The objects to describe the scenario quantitatively are shown in Fig. 6. The two *actors* refer to the quantitative counterparts of the *actor categories* in Fig. 5. Initial state vectors are listed for each *actor* using the coordinate frame that is shown in Fig. 4. Since we are describing a real-world scenario, there is no need to define goals or intents for the actors.

There are four *events* defined. These *events* mark the time instants that define the start and the end of the *activities*. For simplicity, it is assumed that the start of the scenario occurs at 0 s.

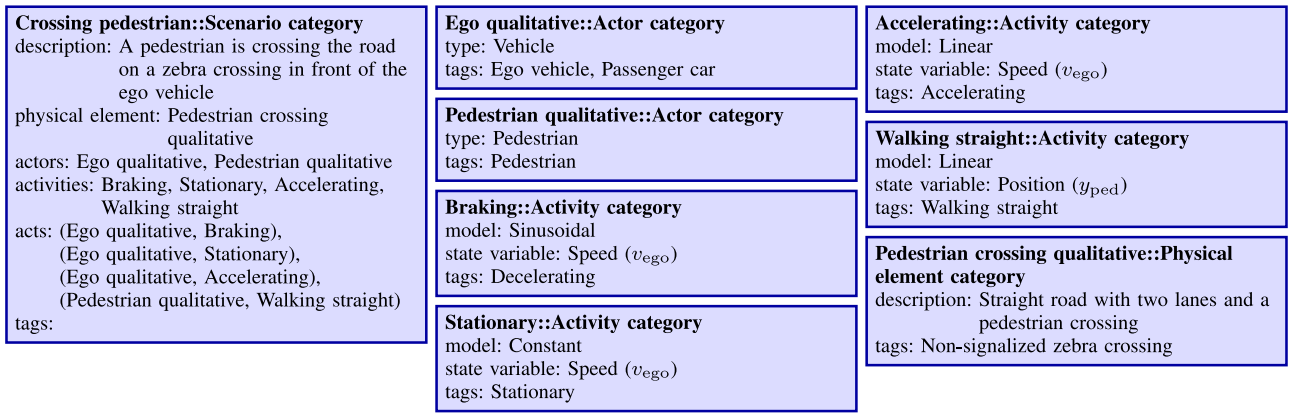



Fig. 5. Objects that are used to qualitatively describe the scenario that is schematically shown in Fig. 4. The first line of each block shows the name (before the double colon) and the class from which the object is instantiated. The following lines show the attributes of the object with the name and value of the attribute before and after the colon, respectively. For the sake of brevity, the unique ID of each object is omitted.

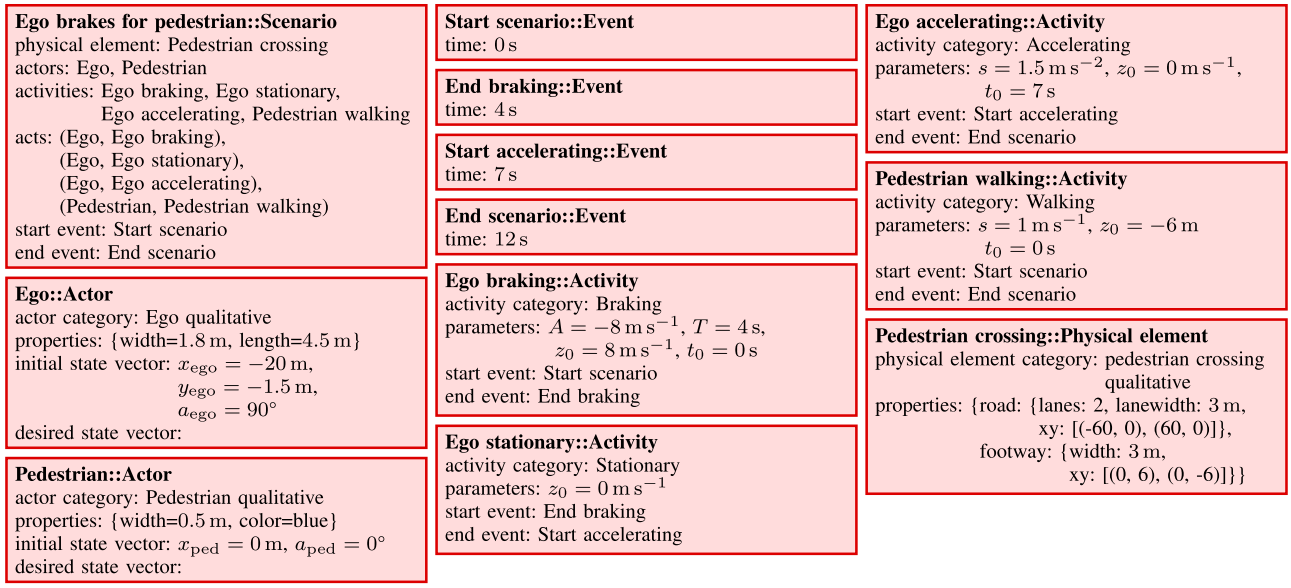



Fig. 6. Objects that are used to quantitatively describe the scenario that is schematically shown in Fig. 4. For the sake of brevity, the tags and the unique ID of each object are omitted.

There are four *activities* defined and each of these *activities* refers to its qualitative counterpart. The *activities* contain the values of the parameters as well as events that mark the start and the end of the *activities*. As described by the first *activity* (*ego braking*), the ego vehicle starts with a speed of 8 $\mathrm{m\,s^{-1}}$ and brakes in 4 s to come to a full stop. By integrating the sinusoidal function of (2) twice, it can be shown that the ego vehicle stops at 4 m from the center of the pedestrian crossing. After waiting for 3 s as described by the second *activity* (*ego stationary*), the ego vehicle accelerates with 1.5 $\mathrm{m\,s^{-2}}$ towards a speed of 7.5 $\mathrm{m\,s^{-1}}$ as described by the third *activity* (*ego accelerating*). The fourth *activity* describes the position and speed of the pedestrian.

The *pedestrian crossing* describes the entire static environment, including the main road the ego vehicle is driving on and the footway the pedestrian is walking on. The example in Fig. 6 shows some properties of the road layout to illustrate how the static environment can be described. Note that, in practice, the quantitative description of the static environment may contain many more facets than the ones mentioned in Fig. 6. As mentioned in Section III-A, it is possible to refer to another source that contains a description of (part of) the static environment, see, e.g., [55].

The *scenario* has the previously defined *physical element*, *actors*, and *activities* as attributes. The acts are used to assign the first three *activities* to the ego vehicle and the last *activity* to the pedestrian. The *scenario* also has *events* marking the start and the end of the *scenario*. A different *scenario* can be defined by, e.g., changing the parameter values. This illustrates that the *scenario category* in Fig. 5 comprises multiple *scenarios*, including the *scenarios* that only differ from the *scenario* in Fig. 6 because of different parameter values.

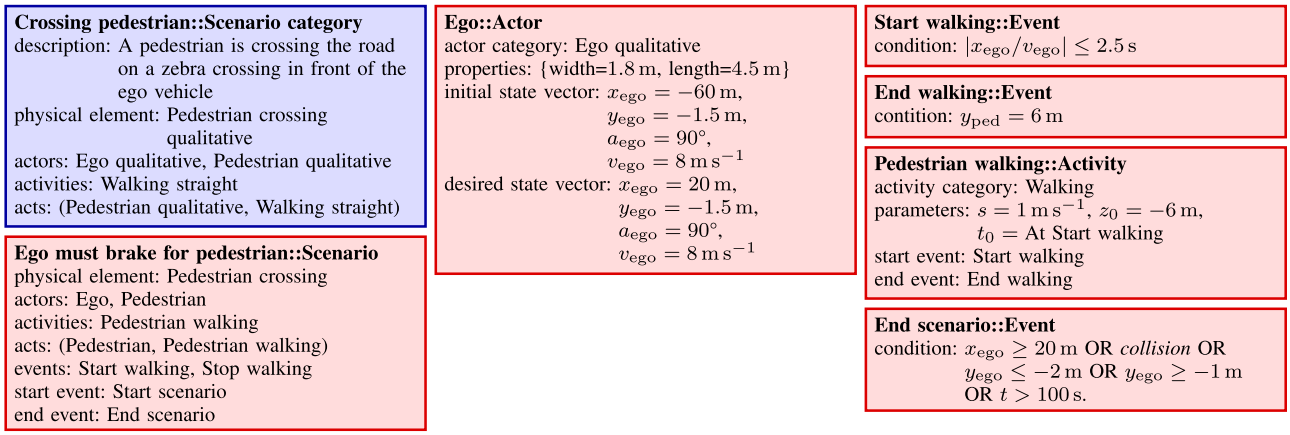



Fig. 7. The objects that, together with the objects *Ego qualitative*, *Pedestrian qualitative*, *Walking straight*, and *Pedestrian crossing qualitative* from Fig. 5 and *Start scenario*, *Pedestrian*, and *Pedestrian crossing* from Fig. 6, describe a test scenario that is schematically shown in Fig. 4. For the sake of brevity, the tags and the unique ID of each object are omitted.

### C. Test Scenario of the Pedestrian Crossing

In this example, we consider a test scenario based on the previously illustrated real-world scenario, see Fig. 4. To describe the test scenario, we reuse the two *actor categories* from Fig. 5 (*ego qualitative* and *pedestrian qualitative*) and the *actor* describing the pedestrian from Fig. 6 (*pedestrian crossing*). Fig. 7 shows the other objects that are used to describe this test scenario.

The *scenario category* only differs from the *scenario category* shown in Fig. 5 in that it does not contain *activity categories* that describe the activity of the ego vehicle.

Two attributes of the quantitative description of the ego vehicle are different. First, the initial state vector also includes the speed, denoted by $v_{\mathrm{ego}}$, at the start of the scenario and the initial position is further away from the pedestrian crossing, such that the ego vehicle's driver or automation system has more time to perceive the pedestrian. Second, because there are no activities defined for the ego vehicle, the desired state vector is defined. The goal is to reach the point 80 m in front of the ego vehicle while driving with a speed of $v_{\mathrm{ego}} = 8\,\mathrm{m\,s^{-1}}$.

The *event* that marks the start of the walking activity of the pedestrian is triggered if the ego vehicle is 2.5 s away from the center of the footway, assuming that the speed of the ego vehicle is constant. In case the ego vehicle drives with a speed of $v_{\mathrm{ego}} = 8\,\mathrm{m\,s^{-1}}$, this is at a distance of 20 m, similar to the scenario described in Fig. 6.

As with the *scenario category*, the *scenario* does not contain *activities* of the ego vehicle. Furthermore, the end event of the scenario is defined differently: now the scenario ends if the ego vehicle either reaches its destination ($x_{\mathrm{ego}} \geq 20\ \mathrm{m}$), collides with the pedestrian, deviates too much from its path ($y_{\mathrm{ego}} \leq -2\ \mathrm{m}$ or $y_{\mathrm{ego}} \geq 1\ \mathrm{m}$), or takes too long to reach the destination ($t > 100\,\mathrm{s}$).

Note that this example considers a pedestrian that crosses the road at a fixed speed ($1\,\mathrm{m\,s^{-1}}$) regardless of the proximity of the ego vehicle. To model, e.g., the case where the pedestrian notices the ego vehicle and accelerates if a collision is about to happen, an activity can be added that describes the increased speed (e.g., $2\,\mathrm{m\,s^{-1}}$) of the pedestrian. The start of this activity is at a predefined event with, e.g., the condition $|x_{\mathrm{ego}}/v_{\mathrm{ego}}| \leq 1\,\mathrm{s}$ AND $y_{\mathrm{ped}} < 0\ \mathrm{m}$.

### D. Remarks on the Example

The example illustrates the benefits of the object-oriented approach for defining a scenario, which are:

- clarity regarding the content of the scenario,
- modularity, which makes it easy to understand the changes from the real-world scenario in Fig. 6 to the test scenario in Fig. 7, and
- reusability, as is illustrated by the objects that are used more than once.

Furthermore, each object listed in Figs. 5 to 7 is directly translatable to an object in an object-oriented programming language. As a further illustration that the presented OOF is practical to use in real life, the framework is used by TNO's StreetWise program for storing real-world scenarios in a database [11][8].

In the example, two different actors are considered: the ego vehicle and the pedestrian. These are examples of traffic participants, but an actor is not necessarily a traffic participant. For example, road side units that transmit messages in an infrastructure-to-vehicle communication setting can also be actors. In this case, the transmission of messages can be considered as an activity. Another example of an actor is the road surface in case it is important for the scenario to model the changing surface temperature.

## VI. Conclusion

The performance assessment of Automated Vehicles (AVs) is essential for the legal and public acceptance of AVs as well as for the technology development of AVs. Because scenarios are crucial for the assessment, a clear definition of a scenario is required. In this work, we have proposed a new definition of the

[8]See also https://www.tno.nl/streetwise.

concept scenario in the context of the performance assessment AVs.

While our definition is consistent with other definitions from the literature, it is more concrete such that it can directly be implemented using code. We have further defined the notions of event, activity, and scenario category. To formalize the concepts of scenario, event, activity, and scenario category, an Object-Oriented Framework (OOF) has been proposed. Using the proposed framework, it is possible to describe a scenario in both a qualitative and quantitative manner. The framework, represented using class diagrams, can be directly translated into a class structure for an object-oriented software implementation. This allows us to translate scenarios into code, such that both domain experts and software programs, such as simulation tools, are able to understand the content of the scenarios. To demonstrate this, we have made our implementation in the coding language Python publicly available.

The OOF has been illustrated with an example of an urban scenario with a pedestrian crossing. We have also demonstrated how this particular scenario can be used to define a test scenario using the proposed framework. In the publicly available[9] coding implementation of the presented OOF, we have shown how to use the proposed OOF from a real application's perspective.

The presented framework is applicable for scenario mining [39], [78] and scenario-based assessment [6], [11] and, therefore, this framework provides a step towards scenario-based performance assessment of AVs. The next step is to define scenarios and scenario categories[10] that are relevant for an AV in a specific deployment area. Future work also includes creating an ontology for scenarios for the assessment of AVs. The presented OOF could be a good starting point for this [41]. An ontology allows, among others, to add properties to relationships that enable automated reasoning. In this way, an ontology enables automated classification of scenarios, thereby helping to overcome problems of data ambiguity [52].

## Appendix A
## Nomenclature

For the definition of *scenario*, several notions are adopted from the literature. In this section, the concepts of *ego vehicle*, *physical element*, *actor*, *static environment*, *dynamic environment*, *act*, *state variable*, *state vector*, *model*, and *mode*, which are adopted from literature, are detailed.

### A. Ego Vehicle

The ego vehicle is the main subject of a scenario. In particular, the ego vehicle refers to the vehicle that is perceiving the world through its sensors (see, e.g., [77]). When performing tests, the ego vehicle also refers to the vehicle that must perform a specific task (see, e.g., [30], [35]). In this case, the ego vehicle is often referred to as the system under test [4], the vehicle under test [5], or the host vehicle [5].

[9]https://github.com/ErwindeGelder/ScenarioDomainModel

[10]As a starting point, the 67 scenario categories in [16] can be used.

### B. Physical Element

A physical element refers to an object that exists in the three-dimensional space.

### C. Actor

According to [35], "actors are all dynamic components of a scenario, excluding the ego vehicle itself." Note that, in contrast to [35], in the current paper, the ego vehicle's driver, and/or automation system are considered as actors, similar to [14], because they have the same properties as another driver or automation system. While the aforementioned definition of [35] provides a good idea of what an actor could be, we use another definition in order to avoid a circular definition: an actor is a dynamic physical element, i.e., a physical element that experiences change.

*Remark 2:* An actor is also a physical element whereas a physical element is not necessarily an actor. For example, a static road sign is considered a physical element, but because it does not change during the course of a scenario, it is not an actor.

### D. Static Environment

The static environment refers to the part of the environment that does not change during a scenario. This includes geospatially stationary elements [13], such as the road network.

### E. Dynamic Environment

As opposed to the static environment, the dynamic environment refers to the part of the environment that changes during the time frame of a scenario. In practice, the dynamic environment mainly consists of the moving actors (other than the ego vehicle) that are relevant to the ego vehicle. For example, the primary use case of OpenSCENARIO [29], a file format for the description of the dynamic content of driving simulations, is to describe "complex, synchronized maneuvers that involve multiple entities like vehicles, pedestrians, and other traffic participants" [29]; so for OpenSCENARIO, these maneuvers represent the dynamic environment. Roadside units that communicate with vehicles within the communication range [79] are also part of the dynamic environment. Furthermore, changing (weather) conditions are part of the dynamic environment.

*Remark 3:* Note that it might not always be obvious whether an element of the environment belongs to the static or dynamic environment. Most important, however, is that all parts of the environment that are relevant to the assessment of an AV are described in either the static or the dynamic environment.

### F. Act

We define an act as a combination of an actor and the activity that is performed by the actor or the combination of actors and the activities they are subjected to. This is in accordance with the use of the term *act* in [29].

### G. State Variable

Dorf and Bishop [80, p. 163] write that "the state variables describe the present configuration of a system and can be used to determine the future response, given the excitation inputs and the equations describing the dynamics." In our case, "the system" could refer to an actor, a component, or a simulation. For example, a state variable could be the acceleration of an actor.

### H. State Vector

A state vector refers to "the vector containing all $n$ state variables" [80, p. 233].

### I. Model

A dynamical system is often modeled using a differential equation of the form $\dot{z}(t) = f_\theta(z(t), u(t), t)$ [81], where $z(t)$ represents the state vector at time $t$, $u(t)$ represents an external input vector, and the function $f_\theta(\cdot)$ is parameterized by $\theta$. Note that, technically speaking, $z(\cdot)$, $u(\cdot)$, $t$, and $\theta$ are inputs of the function $f$, but $\theta$ is assumed to be constant for a certain time interval. For example, the following first-order model is parameterized by $\theta = (a, b)$:

$$\dot{z}(t) = az(t) + bu(t). \tag{6}$$

### J. Mode

In some systems, the behavior of the system may suddenly change abruptly, e.g., due to a sudden change in an input, a model parameter, or the model. Such a transition is called a mode switch. In each mode, the behavior of the system is described by a model with a fixed function $f_\theta$ and smooth input $u(\cdot)$ [59].

## ACKNOWLEDGMENT

The authors would like to thank Mark van den Brand and Ludwig Friedmann for providing helpful feedback on earlier versions of this article.

## REFERENCES

[1] K. Bengler, K. Dietmayer, B. Färber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 4, pp. 6–22, Oct. 2014.

[2] W. Wachenfeld and H. Winner, "The release of autonomous vehicles," in *Autonomous Driving*, Berlin, Germany: Springer, 2016, pp. 425–449.

[3] T. Helmer, K. Kompaß, L. Wang, T. Kühbeck, and R. Kates, *Safety Performance Assessment of Assisted and Automated Driving in Traffic: Simulation as Knowledge Synthesis*. Berlin, Germany: Springer, 2017, pp. 473–494.

[4] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner, "Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, 2015, pp. 1455–1462.

[5] O. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, "Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations," *Veh. Syst. Dyn.*, vol. 44, no. 7, pp. 569–590, 2006.

[6] A. Pütz, A. Zlocki, J. Bock, and L. Eckstein, "System validation of highly automated vehicles with a database of relevant traffic scenarios," in *Proc. 12th ITS Eur. Congress*, 2017, pp. 1–8. [Online]. Available: https://www.pegasusprojekt.de/files/tmpl/pdf/12th%20ITS%20European%20Congress_Folien.pdf

[7] C. Roesener *et al.*, "A comprehensive evaluation approach for highly automated driving," in *Proc. 25th Int. Tech. Conf. Enhanced Safety Veh.*, 2017, pp. 1–13. [Online]. Available: https://www-esv.nhtsa.dot.gov/Proceedings/25/25ESV-000259.pdf

[8] 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.

[9] A. Knapp, M. Neumann, M. Brockmann, R. Walz, and T. Winkle, "Code of practice for the design and evaluation of ADAS," Eur. Automobile Manufacturers' Assoc. (ACEA), RESPONSE III: a PReVENT Project, Tech. Rep Response3_CoP_e_v5.0.doc, 2009. [Online]. Available: https://www.acea.auto/files/20090831_Code_of_Practice_ADAS.pdf

[10] *Road Vehicles - Functional Safety*, Standard ISO 26262, International Organization for Standardization, 2018. [Online]. Available: https://www.iso.org/standard/68383.html

[11] H. Elrofai, J.-P. Paardekooper, E. de Gelder, S. Kalisvaart, and O. Op den Camp, "Scenario-based safety validation of connected and automated driving," Netherlands Org. Appl. Sci. Res., TNO, Tech. Rep., 2018. [Online]. Available: http://publications.tno.nl/publication/34626550/AyT8Zc/TNO-2018-streetwise.pdf

[12] A. Aparicio *et al.*, "Pre-crash performance of collision mitigation and avoidance systems: Results from the ASSESS project," in *Proc. World Automot. Congr.*, 2013, pp. 489–505.

[13] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and substantiating the terms scene, situation, and scenario for automated driving," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, 2015, pp. 982–988.

[14] S. Geyer *et al.*, "Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance," *IET Intell. Transp. Syst.*, vol. 8, no. 3, pp. 183–189, 2014.

[15] M. R. Zofka, F. Kuhnt, R. Kohlhaas, C. Rist, T. Schamm, and J. M. Zöllner, "Data-driven simulation and parametrization of traffic scenarios for the development of advanced driver assistance systems," in *Proc. 18th Int. Conf. Inf. Fusion*, 2015, pp. 1422–1428. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7266724

[16] E. de Gelder, O. Op den Camp, and N. de Boer, "Scenario categories for the assessment of automated vehicles," CETRAN, Tech. Rep., 2020, *version 1.7*. [Online]. Available: http://cetran.sg/wp-content/uploads/2020/01/REP200121_Scenario_Categories_v1.7.pdf

[17] H. Weber *et al.*, "A framework for definition of logical scenarios for safety assurance of automated driving," *Traffic Inj. Prevention*, vol. 20, no. sup1, pp. S65–S70, 2019.

[18] M. Gyllenhammar *et al.*, "Towards an operational design domain that supports the safety argumentation of an automated driving system," in *Proc. 10th Eur. Congr. Embedded Real Time Syst.*, 2020, pp. 1–10. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1390550/FULLTEXT01.pdf

[19] A. Ebner, T. Helmer, R. R. Samaha, and P. Scullion, "Identifying and analyzing reference scenarios for the development and evaluation of active safety: Application to preventive pedestrian safety," *Int. J. Intell. Transp. Syst. Res.*, vol. 9, no. 3, pp. 128–138, 2011.

[20] W. Hulshof, I. Knight, A. Edwards, M. Avery, and C. Grover, "Autonomous emergency braking test results," in *Proc. 23rd Int. Tech. Conf. Enhanced Saf. Vehicles*, 2013, pp. 1–13. [Online]. Available: https://cdn.euroncap.com/media/1384/aeb-test-results-esv-2013-0-a837f165-3a9c-4a94-aefb-1171fbf21213.pdf

[21] Z. Xiong and J. Olstam, "Orchestration of driving simulator scenarios based on dynamic actor preparation and automated action planning," *Transp. Res. C, Emerg. Technol.*, vol. 56, pp. 120–131, 2015.

[22] Y. Shao, M. A. Mohd Zulkefli, Z. Sun, and P. Huang, "Evaluating connected and autonomous vehicles using a hardware-in-the-loop testbed and a living lab," *Transp. Res. C, Emerg. Technol.*, vol. 102, pp. 121–135, 2019.

[23] C. Neurohr, L. Westhofen, M. Butz, M. Bollmann, U. Eberle, and R. Galbas, "Criticality analysis for the verification and validation of automated vehicles," *IEEE Access*, vol. 9, pp. 18016–18041, 2021.

[24] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, "A systematic review of perception system and simulators for autonomous vehicles research," *Sensors*, vol. 19, no. 3, 2019, Art. no. 648.

[25] P. Wimmer *et al.*, "Toward harmonizing prospective effectiveness assessment for road safety: Comparing tools in standard test case simulations," *Traffic Inj. Prevention*, vol. 20, no. sup1, pp. S139–S145, 2019.

[26] Q. Chao *et al.*, "A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving," *Comput. Graph. Forum*, vol. 39, pp. 287–308, 2020.

[27] P. Kaur, S. Taghavi, Z. Tian, and W. Shi, "A survey on simulators for testing self-driving cars," 2021, arXiv:2101.05337. [Online]. Available: https://arxiv.org/abs/2101.05337

[28] A. Nehemiah, P. Fryscak, and M. Sasena, "Building an Autonomous Vehicle (AV) simulation toolchain with Simulink, RoadRunner and NVIDIA DRIVE Sim," *Blog Post*, 2021. [Online]. Available: https://blogs.mathworks.com/student-lounge/2021/04/26/building-an-autonomous-vehicle-av-simulation-toolchain-with-simulink-roadrunner-and-nvidia-drive-sim/

[29] *OpenSCENARIO*, 2021, Accessed: Dec. 2021. [Online]. Available: https://www.asam.net/standards/detail/openscenario/

[30] M. Althoff, M. Koschi, and S. Manzinger, "CommonRoad: Composable benchmarks for motion planning on roads," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 719–726.

[31] W.-T. Tsai, A. Saimi, L. Yu, and R. Paul, "Scenario-based object-oriented testing framework," in *Proc. 3rd Int. Conf. Qual. Softw.*, 2003, pp. 410–417.

[32] M. Utting, A. Pretschner, and B. Legeard, "A taxonomy of model-based testing approaches," *Softw. Testing Verification Rel.*, vol. 22, no. 5, pp. 297–312, 2012.

[33] M. R. Zofka, S. Klemm, F. Kuhnt, T. Schamm, and J. M. Zöllner, "Testing and validating high level components for automated driving: Simulation framework for traffic scenarios," in *Proc. IEEE Intell. Veh. Symp.*, 2016, pp. 144–150.

[34] D. Wittmann, M. Lienkamp, and C. Wang, "Method for comprehensive and adaptive risk analysis for the development of automated driving," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst.*, 2017, pp. 1–7.

[35] A. B. Frost, "MUSICC: Multi user scenario catalogue for CAVs, glossary of terms and definitions," *Catapult Transp. Syst*., 2018.

[36] Z. Saigol, A. Peters, M. Barton, and M. Taylor, "Regulating and accelerating development of highly automated and autonomous vehicles through simulation and modelling," *Catapult Transp. Syst*., 2018.

[37] *Special Interest Group (SIG) on SImulation and Modeling (SIM)*, "Modeling and simulation glossary," 2021, Accessed: Dec. 2021. [Online]. Available: https://sigsim.acm.org/mskr/glossary.htm

[38] H. Elrofai, D. Worm, and O. Op den Camp, "Scenario identification for validation of automated driving functions," in *Proc. Adv. Microsyst. Automotive Appl.*, 2016, pp. 153–163.

[39] E. de Gelder, J. Manders, C. Grappiolo, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, "Real-world scenario mining for the assessment of automated vehicles," in *Proc. IEEE Int. Transp. Syst. Conf.*, 2020, pp. 1073–1080.

[40] E. de Gelder, H. Elrofai, A. Khabbaz Saberi, O. Op den Camp, J.-P. Paardekooper, and B. De Schutter, "Risk quantification for automated driving systems in real-world driving scenarios," *IEEE Access*, vol. 9, pp. 168953–168970, 2021.

[41] W. V. Siricharoen, "Ontology modeling and object modeling in software engineering," *Int. J. Softw. Eng. Appl.*, vol. 3, no. 1, pp. 43–59, 2009.

[42] R. E. Johnson and B. Foote, "Designing reusable classes," *J. Object-Oriented Program.*, vol. 1, no. 2, pp. 22–35, 1988.

[43] P. Wegner, "Concepts and paradigms of object-oriented programming," *ACM SIGPLAN OOPS Messenger*, vol. 1, no. 1, pp. 7–87, 1990.

[44] C. Patridge, *Business Objects: Re-Engineering for Re-Use*. Tysons, VA, USA: The BORO Centre, 2005.

[45] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design patterns: Abstraction and reuse of object-oriented design," in *Proc. Eur. Conf. Object-Oriented Program.*, 1993, pp. 406–431.

[46] A. Snyder, "Encapsulation and inheritance in object-oriented programming languages," in *Proc. Object-Oriented Program. Syst. Lang. Appl.*, 1986, pp. 38–45.

[47] B. Meyer, "Reusability: The case for object-oriented design," *IEEE Softw.*, vol. 4, no. 2, pp. 50–64, Mar. 1987.

[48] P. W. F. van Notten, J. Rotmans, M. B. A. Van Asselt, and D. S. Rothman, "An updated scenario typology," *Futures*, vol. 35, no. 5, pp. 423–443, 2003.

[49] P. Bishop, A. Hines, and T. Collins, "The current state of scenario development: An overview of techniques," *Foresight*, vol. 9, no. 1, pp. 5–25, 2007.

[50] *SAE International*, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," Tech. Rep. J3016, Jun. 2018.

[51] K. Go and J. M. Carroll, "The blind men and the elephant: Views of scenario-based system design," *Interactions*, vol. 11, no. 6, pp. 44–53, 2004.

[52] OpenSCENARIO 2.0, 2020, Accessed: Dec. 2021. [Online]. Available: https://releases.asam.net/OpenSCENARIO/2.0-concepts/ASAM_OpenSCENARIO_2-0_Concept_Paper.html

[53] H. Kahn, *Thinking the Unthinkable*. New York, NY, USA: Horizon Press, 1962.

[54] J. Alcamo, "Scenarios as tools for international environmental assessment," Eur. Environ. Agency, Tech. Rep. 24, 2001.

[55] M. Dupuis, M. Strobl, and H. Grezlikowski, "Opendrive 2010 and beyond - Status and future of the de facto standard for the description of road networks," in *Proc. Driving Simul. Conf. Europe*, 2010, pp. 231–242.

[56] R. Breu *et al.*, "Towards a formalization of the unified modeling language," in *Proc. Eur. Conf. Object-Oriented Program.*, 1997, pp. 344–366.

[57] P. E. Pfeiffer, *Concepts of Probability Theory*. Chelmsford, MA, USA: Courier Corp., 2013.

[58] M. S. Branicky, V. S. Borkar, and S. K. Mitter, "A unified framework for hybrid control: Model and optimal control theory," *IEEE Trans. Autom. Control*, vol. 43, no. 1, pp. 31–45, Jan. 1998.

[59] B. De Schutter, "Optimal control of a class of linear hybrid systems with saturation," *SIAM J. Control Optim.*, vol. 39, no. 3, pp. 835–851, 2000.

[60] R. Batres *et al.*, "An upper ontology based on ISO 15926," *Comput. Chem. Eng.*, vol. 31, no. 5/6, pp. 519–534, 2007.

[61] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *Proc. 51st IEEE Conf. Decis. Control*, 2012, pp. 3270–3285.

[62] R. Alur and D. L. Dill, "A theory of timed automata," *Theor. Comput. Sci.*, vol. 126, no. 2, pp. 183–235, 1994.

[63] P. Seiniger, A. Hellmann, O. Bartels, M. Wisch, and J. Gail, "Test procedures and results for pedestrian AEB systems," in *Proc. 24th Int. Tech. Conf. Enhanced Saf. Veh.*, 2015, pp. 1–11.

[64] S. Childress, B. Nichols, B. Charlton, and S. Coe, "Using an activity-based model to explore the potential impacts of automated vehicles," *Transp. Res. Rec.*, vol. 2493, no. 1, pp. 99–106, 2015.

[65] G. Bagschik, T. Menzel, and M. Maurer, "Ontology based scene creation for the development of automated vehicles," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 1813–1820.

[66] W. G. Najm, J. D. Smith, and M. Yanagisawa, "Pre-crash scenario typology for crash avoidance research," U. S. Dept. Transp. Res. Innov. Technol. Admin., Tech. Rep. DOT HS 810 767, 2007.

[67] F. Fahrenkrog, L. Wang, C. Roesener, J. Sauerbier, and S. Breunig, "Deliverable D7.3 impact analysis for supervised automated driving applications," AdaptIVe, Tech. Rep. Deliverable D7.3, 2017.

[68] O. Op den Camp, A. Ranjbar, J. Uittenbogaard, E. Rosen, R. Fredriksson, and S. de Hair, "Overview of main accident scenarios in car-to-cyclist accidents for use in AEB-system test protocol," in *Proc. Int. Cycling Saf. Conf.*, 2014.

[69] J. Lenard, A. Badea-Romero, and R. Danton, "Typical pedestrian accident scenarios for the development of autonomous emergency braking test protocols," *Accident Anal. Prevention*, vol. 73, pp. 73–80, 2014.

[70] Transport Systems Catapult, "Taxonomy of scenarios for automated driving," Transp. Syst. Catapult, Tech. Rep., 2017.

[71] F. Hauer, T. Schmidt, B. Holzmüller, and A. Pretschner, "Did we test all scenarios for automated and autonomous driving systems?," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2019, pp. 2950–2955.

[72] E. de Gelder, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, "Safety assessment of automated vehicles: How to determine whether we have collected enough field data?," *Traffic Inj. Prevention*, vol. 20, no. S1, pp. 162–170, 2019.

[73] A. Lara, J. Skvarce, H. Feifel, M. Wagner, and T. Tengeiji, "Harmonized pre-crash scenarios for reaching global vision zero," in *Proc. 26th Int. Tech. Conf. Enhanced Saf. Veh.*, 2019, pp. 1–19. [Online]. Available: https://www-esv.nhtsa.dot.gov/Proceedings/26/26ESV-000110.pdf

[74] G. Smith, *Tagging: People-Powered Metadata for the Social Web*. Indianapolis, IN, USA: New Riders Publishing, 2007.

[75] D. H. Craft, P. A. Caro, J. Pasqua, and D. C. Brotsky, "Tagging data assets," U.S. Patent 6, 704, 739 B2, 2004.

[76] S. Molloy, T. Ramos, and S. Thakar, "Dynamic hierarchical tagging system and method," U.S. Patent 9, 613,0 99 B2, 2017.

[77] S. Bonnin, T. H. Weisswange, F. Kummert, and J. Schmuedderich, "General behavior prediction by a combination of scenario-specific models," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1478–1488, Aug. 2014.

[78] J.-P. Paardekooper *et al.*, "Automatic identification of critical scenarios in a public dataset of 6000 km of public-road driving," in *Proc. 26th Int. Tech. Conf. Enhanced Saf. Veh.*, 2019, pp. 1–8. [Online]. Available: https://www-esv.nhtsa.dot.gov/Proceedings/26/26ESV-000255.pdf

[79] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," *J. Netw. Comput. Appl.*, vol. 37, pp. 380–392, 2014.

[80] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, 12th ed. Englewood Cliffs, NJ, USA: Prentice Hall, 2011.

[81] S. N. Norman, *Control Systems Engineering*. Hoboken, NJ, USA:Wiley, 2011.

**Erwin de Gelder** received the M.Sc. degree (*cum laude*) in systems and control from the Delft University of Technology, Delft, The Netherlands, in 2014. While working with TNO, he is currently working toward the Ph.D. degree on this topic with the Delft University of Technology. Since 2014, he has been with the Netherlands Organization for Applied Scientific Research, TNO. From 2017 to 2019, he was with Nanyang Technological University, Singapore, to apply his research for an assessment procedure for automated vehicles in Singapore. His current research focuses on a quantitative assessment methodology for automated vehicles using scenarios obtained from real-world data.

**Jan-Pieter Paardekooper** received the M.Sc. degree in astrophysics from Leiden University, Leiden, The Netherlands, in 2006, and the Ph.D. degree in theoretical astrophysics from Leiden, in 2010.

He is currently a Researcher with TNO, The Netherlands, on the topic of artificial intelligence (AI) in connected and automated vehicles. Since 2018, he has been a part-time Research Fellow with the Department of Artificial Intelligence, Donders Institute for Brain, Cognition and Behaviour, Radboud University, Nijmegen, The Netherlands. His research interests include AI safety, neuro-symbolic AI, and situation awareness for AI.

**Arash Khabbaz Saberi** received the M.Sc. degree in embedded systems and the P.D.Eng. degree in automotive system design from the Technical University of Eindhoven, Eindhoven, The Netherlands, in 2013 and 2015, respectively. In 2020, he defended his Ph.D. thesis on application of model based software engineering for functional safety of automated driving. He is currently a Product Safety Engineering Manager with TomTom responsible for product safety of the entire TomTom product portfolio.

**Hala Elrofai** received the M.Sc. degree in applied mathematics from the University of Twente, Enschede, The Netherlands, in 2003, and the Ph.D. degree in applied mathematics from Vrije Universiteit Amsterdam, Amsterdam, The Netherlands, in 2008. Following her Ph.D., she was a Postdoctoral Fellow with the Eindhoven University of Technology, Eindhoven, The Netherlands. After that, she joined the Research and Development Department, ASML. She is currently a Senior Research Scientist with TNO. Her research interests include artificial intelligence and data analytics for development of safety assessment methods and perception and decision-making models for automated driving systems.

**Olaf Op den Camp** received the Engineering degrees in mechanical engineering (*cum laude*) and biomedical engineering (*cum laude*) from the Eindhoven University of Technology, Eindhoven, The Netherlands, and the Ph.D. degree with a study into the identification of material parameters of biological tissue, in 1996. In 1995, he joined the Netherlands Organization for Applied Scientific Research TNO. He is currently a Senior Consultant and Technical Lead of streetwise scenario database development. Since 2018, he has been working part-time with Nanyang Technological University, Singapore, to develop and implement a scenario-based safety assessment framework to support safe deployment of Autonomous Vehicles within the CETRAN Program.

**Steven Kraines** received the Ph.D. degree in chemical systems engineering from the University of Tokyo, Tokyo, Japan, in 1998. From 1998 to 2015, he was a Professor with the University of Tokyo, doing research on applying ontologies, knowledge representation and reasoning, natural language processing and semantic inference to management of knowledge resources using human-authored computer-understandable descriptors. His research interests include expert and knowledge resource matching, semantic search, knowledge mining, knowledge discovery, and knowledge network generation. He is currently a Collaborative Researcher with Symphony Company, Tokyo, Japan, and working on developing ontologies for autonomous vehicles and the road traffic domain.

**Jeroen Ploeg** received the M.Sc. degree in mechanical engineering from the Delft University of Technology, Delft, The Netherlands, in 1988, and the Ph.D. degree in mechanical engineering on the control of vehicle platoons from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2014.

He is currently with 2getthere, Utrecht, The Netherlands, where he leads the research activities in the field of cooperative automated driving for automated transit systems. Since 2017, he has been holding the position of a part-time Associate Professor with the Department of Mechanical Engineering, Eindhoven University of Technology. His research interests include control system design for cooperative and automated vehicles.

**Bart De Schutter** (Fellow, IEEE) received the Ph.D. degree in applied sciences (*summa cum laude* with congratulations of the examination jury) from Katholieke Universiteit Leuven, Leuven, Belgium, in 1996.

He is currently a Full Professor and the Head of Department with the Delft Center for Systems and Control of Delft University of Technology, Delft, The Netherlands. His current research interests include multi-level and multi-agent control, learning-based control, control of hybrid systems with applications in intelligent transportation systems and smart energy systems. He is the Senior Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL.