Mathematical Definitions of Scene and Scenario for Analysis of Automated Driving Systems in Mixed-Traffic Simulations

Downloaded from: https://research.chalmers.se, 2022-08-28 17:31 UTC

Citation for the original published paper (version of record):

N.B. When citing this work, cite the original published paper.
Mathematical Definitions of Scene and Scenario for Analysis of Automated Driving Systems in Mixed-Traffic Simulations

Eleonora Andreotti, Pinar Boyraz, and Selpi

Abstract—This paper introduces a unified mathematical definition for describing commonly used terms encountered in systematical analysis of automated driving systems in mixed-traffic simulations. The most significant contribution of this work is in translating the terms that are clarified previously in literature into a mathematical set and function based format. Our work can be seen as an incremental step towards further formalisation of Domain-Specific-Language (DSL) for scenario representation. We also extended the previous work in the literature to allow more complex scenarios by extending the model-incompliant information using set-theory to represent the perception capacity of the road-user agents. With this dynamic perception definition, we also support interactive scenarios and are not limited to reactive and pre-defined agent behavior. Our main focus is to give a framework to represent realistic road-user behavior to be used in simulation or computational tool to examine interaction patterns in mixed-traffic conditions. We believe that, by formalising the verbose definitions and extending the work in DSL, we can support automatic scenario generation and dynamic/evolving agent behavior models for simulating mixed traffic situations and scenarios. In addition, we can obtain scenarios that are realistic but also can represent rare-conditions that are difficult to extract from field-tests and real driving data repositories.

Index Terms—scene, scenario, situation, scenery, formal definitions, interactive agents, automated driving, mixed-traffic, traffic simulations

I. INTRODUCTION

There are two motivations driving the research efforts in definition of a formal language for describing the complex situations in mixed-traffic simulations: (i) Development of automated driving systems require extensive scenario testing before deployment and the common language allows different systems to be bench-marked on a fair manner, (ii) The co-existence of human-operated vehicles (HOV), autonomous vehicles (AV) and vulnerable road users (VRU) (i.e. bikers, cyclists, pedestrians, e-vehicle users) require the public authorities to examine complex scenarios for better traffic and transportation management as well as digital and physical infrastructure design. In fact, the milestone of such efforts has started with [1] in which scene, scenario and situation terms were clarified and their scope was determined. Building upon these clear definitions and separation between the scenario and situation, [2] developed a Domain-Specific-Language (DSL), named ‘GeoScenario’ to be used in scenario representation to substantiate test cases for automated driving systems (ADS). GeoScenario is a perfect DSL for providing ways of reproducing trajectory of road-user agents and also carefully designed mechanisms of coordinating their actions (Please see Section C, last paragraph in [2]). However, in its current state, to the best of authors’ knowledge, it cannot provide complete and realistic intelligent behavior of road-user agents, featuring dynamic models of road-user agents, yet. In addition to this, this DSL does not specify details of vehicle model dynamics, therefore may not shed light in terms of individual vehicle parameters and their effect in microscopic traffic simulations. For example, the avoidance manoeuvres cannot be fully represented with their exact trajectories in such platforms at this scale. Next to the GeoScenario, another approach for the scenario creation is based on computational ontology, Studer et al. [3]. Guarino et al. [4] make use of set theory to provide a more precise and formal view of the aspects of Studer’s definition of computational ontology. Bagschik et al. [5] also propose a process for a computational ontology based scene creation for the development of Automated Vehicles, subsequently adapted to the ontology based scenario creation [6]. This approach uses a 5-layer-model, expanding a 4-layer-model proposed by Schuldt et al. [7], combined with Ulbrich’s scene definition [1]. Each entity (represented by a word) within a layer represents multiple relations to parameters in the physical state space and to represent interactions of layers the authors propose to annotate if an entity includes or influences a parameter. This approach, proposed for automated vehicles scene, limits the interaction between objects at the given instant of the scene and does not consider that every “thinking entity” (automated or not) of the scene makes its own projections about how it expects the other entities to behave in the near future, and that its reactions are based on its expectations. Beside that, a connection between the functional scenario (described in a linguistic way) and the logical (made up of parameters and parameter ranges) and concrete scenarios (composed by concrete value for each parameter) is necessary [8], [9]. There are also great efforts in designing Open simulation platforms, which can enable multi-agent simulations to estimate the impact of ADS with a focus on safety. In [10], such a platform is used to see the effect of penetration of SAE Level 1 and 2 technologies, such as automated emergency braking (AEB) and lane departure warning (LDW), on safety. It was also possible to demonstrate how real road-network and field
data from reported accidents can be combined to obtain a larger assessment of ADS. Although, they employ an advanced platform to simulate this large-scale mixed-traffic with multi-agent approach, they do not dwell much on the scenario and situation definitions and the software structure is not given in detail yet. Another important aspect of agent-based mixed-traffic simulations is their level of rigour when defining traffic flow or safety metrics to identify potential problems related to congestion and traffic safety. In order to represent these aspects in a quantitative and holistic way, [11] has borrowed the 'potential field' approach that is used to solve the problem of path-planning in robotics, avoiding static or dynamic obstacles. In that work, concept of potential field was extended to include fields of risk from vehicle’s motion, driver’s perception and environmental factors. In order to include the perceptions of agents, several efforts have been made also in SUMO and MITSIMLab [12], [13]. While, in [14], the agents perform tactical-level driving and the manoeuvres performed in every situation are decided in real time. However, the forces are not modeled and the consequences of weather, for example, cannot be taken into account, e.g. the vehicles will perform manoeuvres without slipping. The most recent efforts in the microscopic traffic simulation’s field are well summarized in [15].

In this paper, we propose mathematical definitions of scene, scenario, and situation, using the verbal definitions proposed by [1] as a starting point. Due to our mathematical definitions, we can create a link between spoken language and mathematical models, which can be used to expand the already existing micro-scale traffic simulations. In this context, the set theory is useful to identify objects sharing common properties, and thereby grouping them together.

In doing so, we extend the model-incompliant information to represent the perception capacity of the road-users agents, i.e. the awareness of things through the physical senses. The perception of the road-user can be human perception (human road-user), machine perception (i.e. for autonomous vehicle), or a mixture of both. Therefore, these perceptions depend on the presence/absence and manufactured functionality of the vehicle’s sensors, on the driver emotional state (gloomy/happy, worried/carefree, nervous/calm, bored/interested, angry/peaceful) and on driver personality type (lazy/dynamic, irresolute/resolute, impatient/patient, impulsive/reflective, superficial/meticulous) and driver physical condition (tired/refreshed, sleepy/awake, sick/healthy), see [16], [17] for detailed information. In [18], it is shown that the 90-95% of accidents are caused by human errors in information processing and inattention/lack of attention, while in [19] the authors highlighted that the more experienced drivers have a better understanding of other road users’ communicative signals and are more quickly to detect road hazards.

Our paper is organized as follows: in Sections II, III and IV, we respectively introduce the mathematical definitions of scene, situation and scenario, and illustrate these definitions using examples. In Section V, we show how our definitions could be used to describe a scenario, and the scenario from each road-users’ different point-of-view, therefore extending the perception component in scenario representation. We have compiled discussion points in Section VI to identify unresolved problems and set path for future work in evaluation of ADS using multi-agent traffic simulations. Finally, conclusions are drawn in Section VII.

II. DEFINING THE TERM SCENE

In order to define the term scene, let’s start by defining some preliminary concepts that are useful for our purpose. Let us start by considering a portion of space, $S$, in the real world. We define the *static object set* $V^s(S)$ as the set of the whole motionless objects (lane network, stationary elements, vertical elevation) which are located in the portion of space $S$. By motionless objects, we mean those objects that stand still for a sufficiently long time, or whose movements are almost imperceptible.

**Example 1. Static object set, $V^s(S)$**

The portion of space $S$, in the real world, can be represented as the region delimited by the black perimeter in Fig.1a. In order to represent the static object set of the portion of space $S$, $V^s(S)$, we should delete from the picture all the dynamic elements, i.e. all the objects in Fig.1a whose position changes as a function of time. In Fig.1b we removed the vehicle and bicycles, thus we get the set of the whole motionless road-objects be located in the portion of space $S$, i.e. the static object set.

![Fig. 1: Real (a) and static (b) object set representations of a portion of space $S$ (delimited by the black perimeter).](image)

Let $V^d(S,t)$ be the set of the whole dynamic elements (i.e. pedestrians, cyclists, HOVS, AVs) be located in the portion of space $S$ at time $t$. Each element of the set $V^d(S,t)$ is called *actor*. The union of the sets $V^d(S,t)$ and $V^s(S)$ is denoted as $V(S,t) := V^s(S) \cup V^d(S,t)$ and called the object set. Each element of $V(S,t)$ is called object. Throughout the paper, when there is no scope for ambiguity, we omit the letter $S$ and $t$ from set’s symbols and write, for example, $V^s$ and $V^d$ instead of $V^s(S)$ and $V^d(S,t)$.

In [1, p. 983], the authors define a scene as “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”. Until now, we have defined the object set which includes all the static and dynamic elements. In order to define a scene we need to define “the scenery” as well as “the relationships among those entities” and “all actors’ and observers’ self representations”, and to achieve this, it is useful to introduce a position function. Typically $S$ can be a portion of space given in geographical coordinate system, or in any other coordinate system, and the function $C : V(S,t) \rightarrow S$ associate each
object in \( \mathcal{V}(S,t) \) to its \textit{dimensional coordinates} in the coordinate system of \( S \), Fig.2.

Let \( x \in \mathcal{V}(S,t) \) then

\[
C : \mathcal{V}(S,t) \rightarrow S, \quad x \mapsto C(x).
\]  

(1)

From function (1) we can derive the surface occupied by the object \( x \) and the mutual relations among two or more objects. This function \( C \) is called position function and the set \( C(\mathcal{V}) \) is the position set. Sometimes it is useful to write \( x(t) \) to mean that we are considering the object \( x \) at time \( t \), and then through \( C \) we associate \( x \) to its position at time \( t \).

Example 2. Position set

Each object \( x \in \mathcal{V} \) is associated, through function \( C \), to its dimensional coordinate in \( S \). For instance, in the example shown in Fig.2, each \( x \) can be associated to coordinates of 4 points, \( P', P'' \) and \( P''' \) (in a 2-dimensional space \( S \)) which represent the vertices of the minimum quadrangle that contains the object \( x \), i.e. \( C : \mathcal{V} \rightarrow S, x \mapsto C(x) = \{P, P', P'', P'''\} \). To be more general we could consider polygons with more than 4 vertices.

Remark 1. The vertices in Example 2 delimit the occupancy area. We could work in 3 dimensions, and in such a case we would have the vertices (by extending the quadrangle we could consider parallelepipeds) to represent the occupancy volume.

Fig. 2: Position set \( C(\mathcal{V}) \). (a) Set \( \mathcal{V} \) of the portion of space \( S \) at time \( t \). (b) Set \( \mathcal{V} \) in dimensional coordinate representation \( C(\mathcal{V}) \), i.e. the position set.

The couple \((\mathcal{V}, C)\) provides us the real (objective, ground truth) snapshot of the portion of space \( S \) and the objects contained in it at time \( t \). However each actor \( x \) (pedestrian, cyclist, HOV, AV, etc.) may not see and perceive the whole information contained in the couple \((\mathcal{V}, C)\). Each actor has a role as an observer and within that role has its own representation (i.e. what it can see or perceive).

Moreover, the observer’s perception may depend on what type of observer it is, and from the moment in which it finds itself. For example an automated vehicle (AV) observer will have different perceptions than an elderly driver of a non-automated vehicle observer or an observer who checks the status of road from navigation system. Consistent with the previous definitions, we define the object set from \( x \)’s point of view, or \( x \)-object set, \( \mathcal{V}_x \), as the subset of \( \mathcal{V} \) of the objects that \( x \) sees or perceives at time \( t \), Fig.3.

Example 3. Object set from \( x \)’s point of view, \( \mathcal{V}_x \)

Fig.3 represents the object sets from car (\( x \)) point of view, \( \mathcal{V}_x \), and from bicycle (\( y \)) point of view, \( \mathcal{V}_y \). Using Example 1 we get \( \mathcal{V}_x \neq \mathcal{V} \) and \( \mathcal{V}_x \neq \mathcal{V}_y \) and this is due to the fact that car is not able to see the bicycle because of the presence of the tree.

Aligned with the previous definitions and in order to define a

![Fig. 3](image-url)

Fig. 3: Object sets from car’s driver (a) and cyclist (b) point of view, \( \mathcal{V}_x \) and \( \mathcal{V}_y \) respectively.

Scene it turns out to be crucial to define the position function from \( x \)’s point of view. We consider the \( x \)’s point of view in the coordinate system of \( S \): we define \( C_x \) as the position from \( x \)’s point of view function, or \( x \)-position function as follows:

\[
C_x : \mathcal{V}_x \rightarrow S, \quad y \mapsto C_x(y).
\]  

(2)

Sometimes it is useful to make the time dependence of \( y \) explicit and write \( y(t) \) to mean that we are considering the object \( y \) at time \( t \), and then through \( C_x \) we associate \( y \) to its position at time \( t \). The image set of \( \mathcal{V}_x \) under the \( x \)-position function, \( C_x(\mathcal{V}_x) \), is called position from \( x \)’s point of view set, or \( x \)-position set for brevity. In general it is not true that \( y \in \mathcal{V}_x \subset \mathcal{V} \Rightarrow C_x(y) = C(y) \). Thus, even if an object \( y \) can be seen or perceived from the ego-vehicle \( x \) it does not necessarily mean that \( x \) has an objective view of \( y \) position or size (for example perception loss due to faulty or malfunctioning sensors); see illustrations in Fig.4 and Example 4.

Example 4. Model-incompliant information

In scene definition, the static and dynamic model incompliant information are implicitly taken into consideration. Indeed, for instance, the set \( \mathcal{V}_x \) collect all the objects that \( x \) can see and perceive, but this doesn’t mean that \( x \) really sees or perceives them. The “cognitive” aspect of the model and, more generally, the model incompliant information are represented by the function \( C_x \). Let us suppose to be in the case described in Fig.(3), where the car driver cannot see or perceive the cyclist. Then we get an \( x \)-position set \( C_x(\mathcal{V}_x) \) which does not include the bike. Now let us suppose the bike is coming from the left of the car (Fig.4b). In this case the car driver is supposed to be able to see it, but for some reasons the car driver is not able to see it, due to e.g. failure of sensors for automated vehicles (AV) or inattentive driver for human-operated vehicles (HOV). In this case the bike belongs to \( \mathcal{V}_x \) but the function \( C_x \) associates an empty set to it.

So far, we have defined the set of objects and the function
that defines their positions, both from the point of view of an omniscient observer and from the point of view of any other observer. Now we are going to define an equivalence relation in order to categorise the objects within the set of objects. Indeed, we cannot expect pedestrian objects to have the same behavior or characteristics as cyclist objects, as well as HOV objects with AV objects.

Let us define \( \sim \) the equivalence relation as "is the same type of object", and \( \mathcal{V} \) (or \( \mathcal{V}_x \)) the quotient sets of \( \mathcal{V} \) (or \( \mathcal{V}_x \)) by \( \sim \), i.e. the set of all possible equivalence classes of \( \mathcal{V} \) (or \( \mathcal{V}_x \)) by \( \sim \). Let \( [x] := \{ y \in \mathcal{V} \mid x \sim y \} \) denote the equivalence class to which \( x \) belongs. All elements of \( \mathcal{V} \) equivalent to each other are also elements of the same equivalence class.

**Remark 2.** The equivalence relation \( \sim \) can be more or less stringent. For example a relation "is the same type of object" may manifest the equivalence class set composed of vehicle objects, cyclist objects, static objects, or it may manifest the equivalence class set composed of aggressive car’s driver objects, AV objects, absent-minded car’s driver objects [20], [21], electric bicycle’s driver objects, and so on.

Let us define the states & attributes of the equivalence class of \( x \) set \( A^{[x]} \) as the set of the entail dynamic motion information (like moving forward, being still, turning right) and the information indicating an immediate action that is taking place while the snapshot in time is being taken (like indicator activated and honking, raining). Let \( \mathcal{P}(A^{[x]}) \) be the power set of \( A^{[x]} \) and \( A := \bigcup_{x \in \mathcal{V}} \mathcal{P}(A^{[x]}) \) the states & attributes set, then we define the states & attributes function as follows:

\[
f_{sa} : \mathcal{V} \to A, \quad x \mapsto f_{sa}(x) \tag{3}
\]

where each \( x \in \mathcal{V} \) is associated to an element of \( A \), i.e. to a subset of states & attributes of the equivalence class of \( x \) set, Fig.5. Similarly we define the states & attributes from \( x \)'s point of view function, or \( x \)-states & attributes function as follows:

\[
f_{sa}^x : \mathcal{V}_x \to A_x, \quad y \mapsto f_{sa}^x(y) \tag{4}
\]

where \( A_x := \bigcup_{y \in \mathcal{V}_x} \mathcal{P}(A^{[y]}) \), \( A^{[y]} \) is the states & attributes of the equivalence class of \( y \) from \( x \)'s point of view set (or \( x \)-states & attributes of the equivalence class of \( y \) set) and \( f_{sa}^x(y) \) is the states & attributes from \( x \)'s point of view set, or \( x \)-states & attributes set, i.e. each \( y \in \mathcal{V}_x \) is associated to a subset of \( A_x \).

**Remark 3.** Let \( x \) be an observer, then the set \( A_x \), unlike the set \( A \), may not completely characterise the objects in \( \mathcal{V}_x \). In fact, \( x \) may not be able (not having all the capabilities) to attribute all the states and attributes necessary to characterise the objects it sees/perceives. For example, it may not be able to identify wind speed and the amount of precipitation.

**Example 5. States & attributes set**

Let us suppose the states & attributes of the equivalence class of \( x \) set, \( A^{[x]} \), where \( [x] \) is the equivalence class of vehicle, is made up of the elements \{i, f, s, r, l, h\} (where i=indicator activated, f=moving forward, s=being still, r=turning right, l=turning left, h= honking). Then in the example shown in Fig.1a we could assign the states & attributes set \( f_{sa}(x) = \{f\} \) to the car \( x \). If the tree hadn’t been in the scenery and the car driver had seen the bicycle it could sound the horn, and then the states & attributes set could take the form \( f_{sa}(x) = \{f, h\} \). In this case if the cyclist \( y \) is deaf, for instance, it would attribute the set \( f_{sa}^x(y) = \{f\} \).

**Example 6. States & attributes set, Environmental Conditions.**

Let us consider now the Environmental Conditions of the ODD (Operational Design Domain) classification, [22], which are divided into four subcategories: weather, illumination, particulate matter, and road weather. All these subcategories are objects of the set \( \mathcal{V}^w \). As such each of these subcategories have a position, given by the function \( C \). This position can be a region of \( S \), or even the whole portion of space \( S \). In addition to the position, we can also assign the states & attributes function and set. For instance, the states & attributes of the equivalence class of \( x \) set \( A^{[x]} \) for the object weather can be composed by the elements rain, temperature, wind, and snow, which can be characterised by some adjectives, like low, moderate and heavy. Then the object weather can be associated to one or more of the elements \{rain low, rain moderate, rain heavy,..., snow heavy\} = \( A^{[x]} \) by the states & attributes function.

At this point we have all the necessary ingredients to define a scene.
Definition 1 (Scene). A scene \( \mathcal{E}_x(S, t) \) of the portion of space \( S \) at time \( t \) from \( x \)'s point of view is defined as the 3-tuples \( \mathcal{E}_x(S, t) := (V_x, C_x(V_x), f_{sa}(V_x)) \) where \( V_x \) is the object set from \( x \)'s point of view at time \( t \), \( C_x(V_x) \) and \( f_{sa}(V_x) \) are the \( x \)-position set and \( x \)-states & attributes set respectively.

Remark 4. If we are in a simulated world where “a scene can be complete and uncertainty-free as from an omniscient observer’s point of view”, \([1, p. 983]\), then \( x \)'s point of view sets have to be replaced by omniscient observer’s point of view sets by obtaining the complete scene \( \mathcal{E}_x(S, t) := (V, C(V), f_{sa}(V)) \) where \( V \) is the object set at time \( t \), \( C(V) \) and \( f_{sa}(V) \) are the position set and states & attributes set respectively.

Definition 2 (Scenery). A scenery \( \mathcal{V}_x(S) \) of the portion of space \( S \) is defined as the 3-tuples \( \mathcal{V}_x(S) := (V^s_x, C_x(V^s_x), f_{sa}(V^s_x)) \) where \( V^s_x \) is the set of the \( x \)-static object, \( C_x(V^s_x) \) and \( f_{sa}(V^s_x) \) are the \( x \)-static position set (the \( x \)-position set of the set of the \( x \)-static object) and \( x \)-static states & attributes set (the states & attributes set of the set of the \( x \)-static object) respectively.

Remark 5. Similarly to the definition of a scene, even for the definition of scenery we can consider the viewpoint of the omniscient observer, and therefore the complete and uncertainty-free scenery is defined by replacing the \( x \)-point of view sets by omniscient observer’s point of view sets: \( \mathcal{V}(S) := (V_s, C(V_s), f_{sa}(V_s)) \) where \( V_s \) is the set of the static object, \( C(V_s) \) and \( f_{sa}(V_s) \) are the static position set (the position set of the set of the static object) and static states & attributes set (the states & attributes set of the set of the static object) respectively.

In this way, we have the scenery implicitly included in the scene definition.

III. DEFINING THE TERM SITUATION

In order to define the term situation we introduce the goals & values position functions.

Let us firstly define the goals & values position from \( x \)'s point of view function, or \( x \)-goals & values position function, as the function

\[
\mathcal{C}_x : V_x \rightarrow S, \quad y \mapsto \mathcal{C}_x(y).
\] (5)

which associate each object \( y \in V_x \) to the position of it in the near future, based on \( x \)'s expectation (see Fig.7a).

Remark 6. The position of \( y \in V_x \) in the near future, based on \( x \)'s expectation, will depend not only on \( y \)'s goal, but also on the values that \( x \) attributes to \( y \) (see Example 8).

Example 7. Goals & values positions

Let us consider the example illustrated in Example 1 and the object set from car driver’s (\( x \)) point of view, \( V_x \), represented in Fig.3a.

Because \( x \) is not able to see the bicycle \( y \) behind the tree, it cannot predict a position for \( y \) in the near future: in the future it sees itself crossing the intersection without braking.

In reality, however, the bicycle exists. What happens in the future is an accident, as shown in Fig.7b.

Example 8. Goals & values positions

If a car driver \( y \) does not turn on the indicator, then our observer \( x \) will expect that the car will go straight in the near future. But let us suppose that \( x \) gives \( y \) the value of being a unruly driver, then \( x \) might expect \( y \) to turn right or turn left equally.

Example 9. Goals & values positions (change over time)

Now let us consider the car driver \( y \) does not turn the indicator on at time \( t \) and our subject \( x \) expects the car to go straight in the near future. However, it could happen the car driver \( y \) turns right/left at time \( t + \Delta t \). It is even possible that our subject \( x \), some seconds later, at time \( t + \Delta t/2 \), could expect the car driver \( y \) to turn somewhere because it realises the car driver \( y \) decelerates. Then, the image set of \( V_x \) under the \( x \)-goals & values position function can change over time.

The set \( \mathcal{C}_x(V_x) \) is called goals & values position from \( x \)'s point of view set, or \( x \)-goals & values position set for brevity.

In general we expect that \( \mathcal{C}_x(y) = \mathcal{C}_x(y) \) if \( y \in V_x \), even if this is not always true: a tree, which belongs to the set \( V^s \), can break and fall, for instance.
Let us consider the couple \((V(S,t), C(V(S,t)))\). If we want to predict the future of this couple, as real snapshot, we could think of considering the couple \((V(S,t + \Delta t), C(V(S,t + \Delta t)))\). But, this couple just predicts the future at an exact time, it is not the real goal of the objects in \(V(S,t)\) and moreover the set \(V(S,t)\) could be different from \(V(S,t + \Delta t)\).

For this purpose we define the goals & values position function from a real point of view as follows:

\[
\bar{C} : V \rightarrow S, \quad y \mapsto \bar{C}(y)
\]  

(6)

where the goals & values position function associates each object \(y\) in \(V\) to the dimensional coordinate of \(y\) in the near future, based on \(y\)'s expectation, as illustrated in Fig.7b and Example 7. The goals & values position function knows the goals of each object at time \(t\), i.e. what each object in \(V\) is going to do. In order to pursue the definitions given in [11] we need to introduce a relevant function which filters the relevant information from a given set of the whole information.

Let us firstly define the enlarged situation set, \(N^e_x(S,t)\) (or \(N^e_x\) if no ambiguity concurring), as the 4-tuples made up of the \(x\)-object, \(x\)-position, \(x\)-states & attributions and \(x\)-goals & values position sets \(N^e_x := (V_x, C_x(V_x), f_{sa}^x(V_x), \bar{C}_x(V_x))\).

The relevant function operates on the enlarged situation set as a composition of two functions: firstly the relevant object function filters out all the irrelevant objects from \(N^e_x\) and all the information about them; secondly, on the remaining sets, the relevant information function filters out all the irrelevant information which are not related with the presence or absence of the objects.

We define a \(x\)-relevant function any function defined as follows:

**Definition 3. Relevant function, \(f^x_r\)**

Let \(N^e_x := (V_x, C_x(V_x), f_{sa}^x(V_x), \bar{C}_x(V_x))\) be the enlarged situation of the \(x\)-object set \(V_x\). A relevant function \(f^x_r\) is defined as each function which can be rewritten as the composition of two functions which operate as follows:

\[
f^x_r : N^e_x \xrightarrow{f_{sa}^x} N^e_{xo} \xrightarrow{\bar{C}_x} N_x
\]  

(7)

where \(N^e_{xo} := (V^e_{zo}, C_x(V^e_{zo}), f_{sa}^e(V^e_{zo}), \bar{C}_x(V^e_{zo}))\), with \(V^e_{zo}\) any subset of \(V_x\); and \(N_x := (V_x, C_x(V_x), f_{sa}^x(V_x), \bar{C}_x(V_x))\), with \(C^r(V_x)\), \(f_{sa}^x(V_x)\) and \(\bar{C}_x(V_x)\) any subset of \(C(V_x)\), \(f_{sa}^x(V_x)\) and \(\bar{C}_x(V_x)\) respectively, a situation \(N(S,t)\), or simply \(N\), of the portion of space \(S\) at time \(t\) is defined as the 4-tuples \(N(S,t) := (V^r, C^r(V^r), f_{sa}^r(V^r), \bar{C}_x(V^r))\) where \(V^r\) is a relevant object set, \(C^r(V^r)\), \(f_{sa}^r(V^r)\) and \(\bar{C}_x(V^r)\) are relevant position, relevant states & attributions and relevant goals & values position sets respectively.

**IV. DEFINING THE TERM SCENARIO**

In order to define a scenario we have to define goals & values and actions & events functions first, Fig.11.
Definition 5 (Goals & values function). A goals & values function is the function that describes the path that the object \( y \) takes in order to reach its goal, i.e. to reach the position \( \tilde{C}(y) \)

\[
\mathbf{f}_{gv} : \mathcal{C}(V) \rightarrow \tilde{\mathcal{C}}(V) \times T, \quad \mathcal{C}(y) \rightarrow (\tilde{\mathcal{C}}(y), \bar{t}).
\]  

(9)

Because the function \( \mathcal{C} \) depends on time \( t \) as well as the dimensional coordinate of \( y \), the function \( \mathbf{f}_{gv} \) is able to compute the goal-time \( \bar{t} \), the time in which the object \( y \) reaches the position \( \mathcal{C}(y) \).

Remark 8. The goals & values function is updated at each time step, and its form depends on the interactions of the object with the environment. Let us suppose that the ego vehicle \( y \) is in a car following condition at time \( t_1 \). At time \( t_1 \) the function \( \mathbf{f}_{gv} \) will be given by \( \mathbf{f}_{gv}(\mathcal{C}(y(t_1))) = \left( \mathcal{C}(y(t_1)) + v_y \bar{t}, \bar{t} \right) \), where \( v_y \) is the constant speed of the vehicle \( y \). Let us suppose now that the lead vehicle brakes suddenly at time \( t_2 \). Then, after a reasonable reaction time \( \tau \) (between about 0.7 to 2 seconds for HOVs [23], and about 0.5 for AVs [24]), vehicle \( y \) will also start to brake. Therefore, at time \( t_2 + \tau \) the function \( \mathbf{f}_{gv}(\mathcal{C}(y)) \) will change its form or the value of the parameters. It could decrease the speed \( v_y \) or it could involve a deceleration which in the car following was not relevant. Furthermore, the form/values of the parameters may be also different for AV compared to HOV, due to their different reaction times.

Definition 6 (Actions & events function). A actions & events function is the function that describes the path that the object \( y \), located in \( \mathcal{C}(y(t)) \) at time \( t \), takes in order to reach its position at time \( t + \Delta t \), i.e. to reach the position \( \bar{C}(y(t + \Delta t)) \)

\[
\mathbf{f}_{ae} : \mathcal{V}(S, t) \times T \rightarrow \mathcal{C}(V(S, t + \Delta t)),
\]

(10)

\[
(C(y(t)), t + \Delta t) \rightarrow \mathcal{C}(y(t + \Delta t)).
\]

Remark 9. In general it is not true that \( \mathcal{C}(y(t)) = \mathcal{C}(y(t)) \). Indeed, the object \( y \) could change its goals over the time while \( \mathcal{C}(y(t)) \) represents the real position it reaches at time \( t \).

At this point we define the goals & values from \( x \)'s point of view function, or \( x \)-goals & values function, as follows:

Definition 7 (x-goals & values function). A \( x \)-goals & values function is the function that describes the path that the object \( y \) takes in order to reach the position \( \tilde{C}_x(y) \) from \( x \)'s point of view

\[
\mathbf{f}_{gv}^x : \mathcal{C}_x(V_x) \rightarrow \tilde{\mathcal{C}}_x(V_x) \times T, \quad \mathcal{C}_x(y) \rightarrow (\tilde{\mathcal{C}}_x(y), \bar{t}_x)
\]

(11)

where \( \bar{t}_x \) is the time the object \( y \) reaches the position \( \tilde{C}_x(y) \) based on \( x \)'s expectation.

Definition 8 (x-actions & events function). A \( x \)-actions & events function is the function that describes the path that the object \( y \), located in \( \mathcal{C}_x(y(t)) \) at time \( t \), takes in order to reach its position at time \( t + \Delta t \), i.e. to reach the position \( \mathcal{C}_x(y(t + \Delta t)) \) from \( x \)'s point of view

\[
\mathbf{f}_{ae}^x : \mathcal{C}_x(V_x) \times T \rightarrow \mathcal{C}_x(V_x),
\]

(12)

\[
(C_x(y(t)), t + \Delta t) \rightarrow \mathcal{C}_x(y(t + \Delta t)).
\]

The function \( \mathbf{f}_{gv}^x \), takes the name of \( x \)-goals & values function because intrinsically contains the information of the values of the object \( y \) (from \( x \)'s point of view).

Example 10. \( x \)-goals & values function

In order to make more explicit how \( x \)-goals & values function can represent the values of the object \( y \) let us illustrate an example. Let us suppose the observer \( x \) considers \( y_1 \) an aggressive driver and \( y_2 \) a distracted driver, then the function \( \mathbf{f}_{gv}^x(y_1) \) should take a form which is different from \( \mathbf{f}_{gv}^x(y_2) \). A possible choice to include these information in the model is to describe the function \( \mathbf{f}_{gv}^x \) as a piecewise function depending on the object \( y \) and the values that the observer \( x \) decides to give to \( y \). Let us denote, for instance, \( [y_1], [y_2], [y_3] \) be the set of the aggressive driver, distracted driver and driverless respectively, then we can define the function \( \mathbf{f}_{gv}^x \) in the following way

\[
\mathbf{f}_{gv}^x(C(y)) = \begin{cases} 
  g_1(y) & \text{if } x, y \in [y_1] \\
  g_2(y) & \text{if } x \in [y_1], y \in [y_2] \\
  g_3(y) & \text{if } x \in [y_1], y \in [y_3] \\
  g_4(y) & \text{if } x \in [y_3], y \in [y_1] \\
  g_5(y) & \text{if } x \in [y_3], y \in [y_2] \\
  g_6(y) & \text{if } x, y \in [y_3] \\
  g_7(y) & \text{if } x \in [y_2]
\end{cases}
\]

(13)

where if \( x \in [y_2] \) (a distracted observer) it is not able to distinguish the different modes of the objects it sees (otherwise we could introduce a stochastic parameter in order to define an almost distracted driver). Now let us suppose \( z \) is an honking driver, then we can expect it affects the guide of the distracted driver and the perception of \( x \). We can include the information in the function by substituting the functions \( g_2, g_5 \) and \( g_7 \) with \( g_2(y|z \in V_z), g_5(y|z \in V_z) \)

When the \( x \)-goals & values of object \( y \) and the goals & values of \( y \) does not overlap, or deviate too much, then the probability of conflict may increase. In this way the number of conflicts or safety critical events (SCEs) can be counted and followed-up in the simulation.

Definition 9 (Scenario). A scenario \( \mathcal{O}^x_x(S) \) on the portion of space \( S \) in the time set \( T \) from \( x \)'s point of view is defined as the \( \mathcal{O}^x_x(S) := (\tilde{\mathcal{C}}_x^x(S), \mathbf{f}_{gv}^x, \mathbf{f}_{ae}^x)_{t \in T} \), where the time set is a set of ordered sequence of times \( T = \{ t_i : t_i < t_{i+1}, i = 0, ..., n - 1 \} \).
Remark 10. Differently from scene and situation definitions, which have been defined through set tuples, in the case of scenario definition it is necessary to know the functions $f_{gv}$, $f_{ae}$, which describe how the evolution of scene happens.

V. SCENARIO FROM REAL DRIVING DATA

In order to show the framework capability of representing realistic and rare-conditions, let us consider two public repositories of real driving data: UAH-DriveSet [25], and DR(eye)VE [26]. In the first one the data are recorded by using the smartphone application DriveSafe and the sensors on the smartphone (inertial sensors, GPS, camera and internet access) are used to log and recognize driving maneuvers and infer behaviors from them. In the second one, the data are acquired both from the driver gaze through an eye tracking device, and from a roof-mounted camera, in order to have both the driver’s and the vehicle’s point of view, and to predict the driver’s focus of attention. The main aim of this section is to consider some scenarios described in these datasets and show how these scenarios can be rewritten according to our definitions.

1) Object and states & attributes sets: In TABLE I, the objects and their respective states & attributes are shown. We observe that the attributes can belong to different classes: in the UAH-DriveSet the ego-object has attributes related to its behavior, while in the DR(eye)VE to the attention it shows to the various events. However, our definition allows us to consider more classes simultaneously.

2) Position set: Both datasets provide precise positions of the ego-vehicle through geographic coordinates, speed at any time step, the course and accelerations along the axes of the ego-vehicle. The first information (geographic coordinates) is an element of the position set, precisely the position of the ego-object: in both datasets the position is given through the latitude and longitude coordinates. In addition to those coordinates, the first dataset also has information on altitude.

3) $f_{ae}$ and $f_{gv}$ functions: The other three information sets (speed, course and acceleration) are the parameters that define the functions $f_{ae}$ and $f_{gv}$. The acceleration, in the first dataset, also takes into account the acceleration along the vertical axis, and this aspect, together with the altitude information, makes the functions $f_{ae}$ and $f_{gv}$ able to describe the movements of the ego-vehicle in going up and down from hills. Due to this piece of information, we can consider the perceptions of the ego-speed when going up and down from a hill through the function $f_{gv}$, [27]. In Fig. (12) we show how the speed of the vehicle D4 (UAH-DriveSet) varies as the altitude changes. The elevation parameter can be added to SUMO in several ways [28], but the effects on driving are not taken into account yet, such as the perception of speed.

![Fig. 11: Scenario diagram](Image)

![Fig. 12: Altitude (top) and relative driver D4’s speed (bottom).](Image)

VI. DISCUSSION

By comparing our definitions of scene/scenario with, for example, the scene/scenario build with the open source
SUMO software, we note that in several aspects there is a natural overlap. Each object in SUMO belongs to a class of equivalence called type: each vehicle, according to SUMO, can be a passenger, bicycle, pedestrian, truck, tram or rail, as well as each edge can be a highway, bridleway, bus guideway, cycleway or pedestrian. Moreover, according to the equivalence class to which the object belongs, it is characterised by specific attributes, exactly as the \( f_{sa} \), and by a position, which can be changeable over time or not.

SUMO also includes the ‘agents’ more or less in ‘reactive’ mode, the agents/objects are not generally ‘interactive’. The driver behavior is simulated in compliance with the ACME-Driver Model [12], i.e. the model is able to produce elementary forms of strategic-level behavior, such as individual routing capability or the ability of drivers to recognize high congestion and use alternative routes, but a more comprehensive perception mechanism is not yet implemented. This is a limitation in creating situations where the human/machine perception and relative point of view or ‘model in-compliant info’ cannot be represented. In our definitions, we actually can represent interactive agent behavior, relative perception capacity and similar advanced dynamic features of road-user behavior.

**VII. CONCLUSIONS**

The approach presented in this work gives a mathematical form to the definitions given in [1], which makes it possible to consider the point of view of each object present in the scenario and to treat it as an ego-object (ego-HOV, ego-AV and ego -VRU), i.e. a subject of the scenario. As previous studies shown, the driver’s choices (to let the pedestrians cross the streets, overtake the bicycle, sound the horn, etc.) and the driving style change according to the country [30], [31], the topology of the road [32], [33], the type, the condition as well as the perception of the driver [34], [35].

Through our definitions, the ego-object can be implemented taking into account not only its characteristics (i.e. whether it is aggressive, impulsive, cooperative or strategic, autonomous, automated or human-operated [36], [37], [38], [39], [40]), but also what it is able to see and perceive, its cognitive aspects, and what it expects the other objects present in the scenario to behave (which does not always correspond to the truth) so that the ego-object can react accordingly. In this paper, we have taken advantage of the flexibility of the set theory to design a framework that may be adapted to any software and in future research development. In the near future, we will use this new approach to build complex scenarios composed of dynamic objects with their subjective point of views. Moreover, these dynamic objects will perform manoeuvres taking into account their point of views and the manoeuvres of the other dynamic objects placed in the scenario, as if they were pieces of a larger and more complex puzzle. The point of views of each of the objects within the big scenario can be shared with other objects (e.g., in the context of V2x) leading to a collective cooperation between communicating objects. Consequently the decisions of the objects to deviate path, overtake, etc.) may depend not only on what happens within the scenario, but also on these communications and thus on what happens in the macroscopic scenario description.

**ACKNOWLEDGMENT**

This work is financially supported by the Swedish Innovation Agency Vinnova for grant 2018-02891 and Chalmers Area of Advance Transport for DS-Auto.

**REFERENCES**


