IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 28, 2019, accepted September 19, 2019, date of publication September 23, 2019, date of current version October 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943184

A Scenario-Based Approach for Formal Modelling and Verification of Safety Properties in Automated Driving

BINGQING XU[1], QIN LI[1,2,3], TONG GUO[1], AND DEHUI DU[1]

[1]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China
[2]National Trusted Embedded Software Engineering Technology Research Center, East China Normal University, Shanghai 200062, China
[3]Shanghai Institute of Intelligent Science and Technology, Tongji University, Shanghai 200092, China

Corresponding author: Qin Li (qli@sei.ecnu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB2101300, in part by the Shanghai Science and Technology Committee Rising-Star Program under Grant 18QB1402000, in part by the Science and Technology Commission of Shanghai Municipality Project under Grant 18ZR1411600, in part by the National Natural Science Foundation of China under Grant 61602178 and Grant 61602177, in part by the China HGJ Project under Grant 2017ZX01038102-002, and in part by the National Defense Basic Scientific Research Program of China under Grant JCKY2016204B503.

ABSTRACT Considering diverse scenarios in urban traffic, the safety assessment of the decision-making process in automated driving is of great concern for years. And the difficulties of assessing safety lies in the computation of massive spatio-temporal data, the classification of scenarios, and the representation of the uncertainty in the environment with mixed traffic of manned and automated driving. Formal methods are often advocated as the way of increasing confidence in the safety-critical systems via its rigorous mathematical logic. Thus, we propose an assessment scheme which involves: i) the abstract model for decision-making, ii) characterization of composable scenarios, and iii) a corresponding formal verification method to assess safety. The abstract model captures features from the real-time observation and the estimation of the feasible driving alternatives of the surrounding vehicles, as the scenario is regarded as the dynamic evolution of the spatio-temporal data in the static road geometry over time. These features enable the specification and reasoning of the spatial guard conditions and safety properties, and also contribute to the connection and composability of the scenarios. Due to our scenario-based model verification method, we can assess the safety of decisions in scenario transitions by quantitative verification on the probability of the satisfaction of safety property through mapping from our approach to UPPAAL SMC. For illustration, case studies in the fundamental scene structures and the multi-lane roundabout are introduced.

INDEX TERMS Assessment scheme, automated driving, decision-making model, formal verification, scenario.

I. INTRODUCTION

The Automated Driving System (ADS) is with growing intelligence and the obtained results are likely to ease traffic congestion and enhance driving safety effectively. Back to the DARPA Urban Challenge (DUC) with the closed urban course, the autonomous cars *Boss* from Stanford [1] and *Junior* from Carnegie Mellon [2] came out on top by their advanced techniques. Afterwards, technology companies like Waymo, traditional automotive manufacturers like Audi, and internet firms like Baidu have developed their own autonomous cars and implemented road testing [3]. As for the vision-based techniques in traffic image processing, the KITTI Vision Benchmark Suite is proposed [4]. However, there is no doubt that the autonomous car does encounter difficulties to make safe driving decisions in real driving scenarios. Google confirmed that the vehicle was rear-ended in 8 collisions and side-swiped in 2 collisions by another driver among the 12 collisions till 2015 [5]. A driver was killed in a crash with a tractor-trailer in Tesla autopilot mode in 2016 since the brake was not applied with the failure to notice the white side of the tractor-trailer against a brightly lit sky [5]. These reveal the deficiency in safety assessment scheme with regard to the decision-making process in

The associate editor coordinating the review of this manuscript and approving it for publication was Kan Zheng.

automated driving in the face of the complex urban traffic environment.

The decision-making process can be divided into three levels of skills and control: strategical (i.e. route planning), tactical(manoeuvring such as whether and when to make a left turn), and operational (steering control) [6]. In this paper, the decision-making process is considered as a **periodic tactical control system** with guarded conditions. In each control period, it works with a pattern including i) gathering sensor information by onboard equipment such as LIDARs etc. and wayside stations to detect, recognize and track the moving obstacles, ii) calculating the decision based on the perception from sensor data and estimation of the possible behaviours of the other vehicles, and iii) emitting control signals. The prior probability values of the stochastic behaviours for the surrounding vehicle are calculated from the collected historical information kept in the knowledge library. For urban traffic environment with mixed traffic, the surrounding vehicles consist of manned and autonomous cars at different automatic level [7] in this paper, which may lead to the unavailable vehicle-to-vehicle communication and the unmeasurable driving intentions.

For safety assessment of the driving decisions, the following three main challenges should be considered in the automated driving.

Challenges:

i) **Characterization of the scenarios.** Many studies have shown the decision-making process in various urban scenarios [8], [9]. While these scenarios are mostly regarded as the **separate fixed road types** associated with vehicle movements such as crossroads, motorways, platooning etc. There will exist countless scenarios due to the dynamic evolution of traffic data and the changeable behaviours of vehicles. To characterize general scenarios, but not for a separate case, the structured scenario features, the connection and composability of scenarios should also be taken into account in the characterized scenarios.

ii) **Expression of the uncertainty in the driving environment.** Uncertainty is a dangerous situation which involves imperfect or unknown information from the driving environment. It applies to predictions of future movements of surrounding vehicles, and to physical measurements that are from the sensor devices [10], [11]. Representing uncertainty explicitly is a feasible solution to demonstrate the reasonable estimation of the driving decisions of other vehicles [12].

iii) **Absence in approaches for assessing the safety of driving decisions.** A majority of the current approaches for assessing the safety of decision-making in automated driving is based on testing [13]. A more rigorous approach is needed for the safety-critical systems such as verification approach [14].

We propose a scenario-based formal modelling and verification approach for the safety assessment of the decision-making process in automated driving.

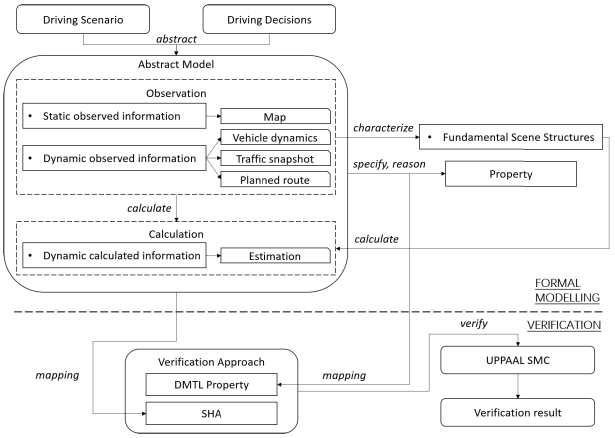


FIGURE 1. Roadmap for the construction of the safety assessment scheme.

Contributions:

1) We propose the formal definition of the scenario by considering it as the dynamic evolution of spatio-temporal data in the static road geometry over time. And we also define three fundamental scene structures as the unit scenarios satisfying corresponding spatial properties. Thus, the **connection and composability** of scenarios are enabled according to the road topological structure and dynamic features. Then it is possible to compose the complex scenario and reason the safety properties in a composite scenario from a simple one.
2) As the basis of making driving decisions, we construct the abstract model for describing the structured features of scenario and stochastic driving behaviours of vehicles. **The abstract model captures the static road geometry, the dynamic evolution of spatio-temporal data including the real-time observation and the estimation of probabilities of the feasible driving alternatives of the surrounding vehicles**. These abstract structures enable a great reduction in state space and facilitate the reasoning and formal verification.
3) We propose the safety assessment scheme where **quantitative properties verification** and the trade-off between properties can be achieved. Due to the abstract model, the corresponding **verification method** is constructed **considering scenario transitions**. By mapping the abstract model to Stochastic Hybrid Automaton (SHA) in the automated model checking tool UPPAAL SMC, it can facilitate the application of our assessment scheme in industry and achieve automated model verification.

Figure 1 shows an overview of our approach to the construction of a safety assessment scheme as described above. The scheme consists of formal modelling and verification. Given the driving scenario and driving decisions, the abstract model is constructed as the basis for decision-making. It involves abstract features based on observation and calculation. The scenarios are characterized according to the observed information. And the estimation is calculated on

FIGURE 2. Multi-lane roundabout scenario. Photo by Luiz Felipe Castro. https://www.vcg.com/creative/811237205.

the basis of observation and characterization of the scenario. Mapping from the formal description in abstract model and property to the verification approach based on the principle of SHA, feeding the model to the automated verification tool UPPAAL SMC, the assessment of driving decisions can be achieved by the verification of qualitative and quantitative properties.

The rest of the paper is organized as follows: Section II presents the observed features in the abstract model. Section III defines the scenario and puts forward the fundamental scene structures. Section IV shows the calculated features in the estimation based on the observation in the abstract model. Section V delivers the verification method. Section VI shows the way our scenario-based approach works in detail and illustrates by cases in the fundamental scene structures. Section VII elaborates the case study in the multi-lane roundabout and analyses the verification result. Section VIII introduces the related work. Section IX concludes this paper.

II. OBSERVED INFORMATION IN THE SCENARIO-BASED ABSTRACT MODEL

It is vital to partition and abstract the driving scenarios in the complex urban traffic environment. In our abstract model based on the scenario, the characterized features can facilitate the specification and reasoning of properties.

A. FEATURES IN THE SCENARIO

The scenario is considered as the dynamic evolution of the spatio-temporal data in the static road geometry over a period of time. Figure 2 shows a typical scenario of a multi-lane roundabout. A roundabout is a type of circular intersection in which road traffic is permitted to flow in one direction around a central island without the use of traffic signals [15]. Some principles should be obeyed in the roundabout that vehicles entering should give way to traffic already within the roundabout; vehicles should leave with the signal on.

Obviously, the roundabout is a composite scenario according to the road geometry. It is necessary to express the composability of the scenario. Since **the composite scenario can be decomposed into several linked sub-scenarios which are fundamental scene structures, the safety assessment of the driving decisions can be realized in these sub-scenarios in sequence.** We select the roundabout scenario as a running example in this paper since it consists of all the fundamental scene structures.

Based on the characterization of the scenario, this section shows the **static observed features** and **dynamic observed features** in the abstract model for decision-making in automated driving.

B. STATIC OBSERVED INFORMATION

Static features are the fixed information describing the partitioning, positioning and connectivity of roads through the mapping from the physical world to the abstract model. Such a description in the abstract model consists of the road and the road topological structure.

To enable spatial reasoning and reduce state space, we define *Segment* as the finite set of abstract road segments for the mapping from physical locations in the real world. SM is the set indicating the infinite real locations on $R \times R$ in the rectangular coordinate system. All the **drivable locations** in SM can be mapped to the road segments in *Segment*. And we can retrieve the corresponding road segment of one real location via function $divSeg : SM \nrightarrow Segment$.

To express the positions of vehicles inside a segment, their positions are abstracted to the points on the centre lines according to the allowed driving directions inside each segment in *Segment*. CP is the subset of SM. CP includes the points on the centre lines of *Segment*. Function *centre* is used to define the centre lines of *Segment*. Apparently, multiple centre lines exist in the intersection of roads in accordance with the driving directions.

- $centre : \mathcal{P}(CP) \rightarrow Segment$ is the function denoting the centre lines of *Segment* where $\mathcal{P}(CP)$ stands for the sets of centre lines consisting of points in CP.

Then the longitudinal lengths can be calculated due to the accumulated distance of adjacent points on the centre line, according to *centre* and driving directions inside each segment within *Segment*. Similarly, the lateral widths of each segment within *Segment* can be calculated by points on the borderline of *Segment*. The division of the segments in length is introduced in Section II-C.

Based on *Segment*, the road is formed upon the connectivity of road segments in *Segment*, and then the whole map is build up.

Definition 1 (Map): $Map = (Segment, Road, Arc)$ *indicates the drivable area upon the directed connections between road segments where*:

- *Road is the finite set of roads. An element road of Road is a set containing segments in the same road.*
- *Arc is the finite set of directed connections between adjacent segments with shared boundaries where the driving direction is allowed. Each connection is described as an ordered pair* (u, v) *where* $u \neq v$ *and* $u, v \in Segment$. Arc_s *is the finite set of directed connections between the*

segments on the same road. $Arc_d = Arc - Arc_s$ *is the set of directed connections between road segments on different roads.*

Based on *Arc*, function *Next* extracts the successor segments connected with the segment, and function *Former* shows the previous segments connected to the segment.

- $Next(v) = \{u \mid (v, u) \in Arc\}$
- $Former(v) = \{u \mid (u, v) \in Arc\}$

We can figure out the fundamental scene structures via the connectivity between segments in Section III, and figure out the existence and quantity of the interference segment in the estimation process in Section IV later.

In fact, *Segment* is the union of three disjoint subsets.

- $Segment = Cross \cup Critical \cup Normal$

For the connections between the segments on the same road, a segment is an element in *Cross* when the connection from the other segments to it is more than 1, and the connection from it to the other segments is equal to or more than 1.

- $Cross = \{u \mid card(\{v \mid (u, v) \in Arc_s\}) \geq 1 \wedge card(\{w \mid (w, u) \in Arc_s\}) > 1\}$ where *card*() counts the elements.

Other than *Cross* defined above, *Critical* indicates segments adjacent to *Cross* and which should be passed through before entering the *Cross*.

- $Critical = \{u \mid (u, v) \in Arc \wedge u \notin Cross \wedge v \in Cross\}$

Normal denotes the set of remained segments.

According to the *Map*, the available driving paths can be described. A single driving path is an ordered list of connected segments in *Segment*.

Definition 2 (Path): Let Map be the map, Path be the set of path on the Map. $path : < S_n >$ *is a sequence of segments with finite length n.* S_0 *stands for the starting segment, and* S_n *represents the destination segment.*

- $path : < S_n >= \{S_n \mid S_n \in Segment \wedge (S_n, S_{n+1}) \in Arc \wedge n \in \mathbb{N}\}$

Figure 3 shows the static observed information for the scenario of a multi-lane roundabout. The static information in this scenario is as follows:

- $Segment = \{A, B, C, D, E, F, G, H, I, J, K, L, M\}$
- $\{A, B, C, D, E, F\} \in Road$
- $(L, M) \in Arc_s$
- $(C, K) \in Arc_d, (K, C) \in Arc_d$
- $B \in Cross, A \in Critical, G \in Normal$
- $Next(B) = \{C, J, K\}, Former(B) = \{A, H\}$

C. DYNAMIC OBSERVED INFORMATION

Based on the static features, dynamic features contains the time-dependent traffic data obtained from the observation and calculation. From the observation, real-time traffic data from the onboard and wayside stations are recognition and tracking of moving obstacles. It indicates the instantaneous driving statuses of all the vehicles on the *Map*, e.g. the vehicle ahead is with the left turning light on etc. The estimation is dynamic calculated information according to the traffic

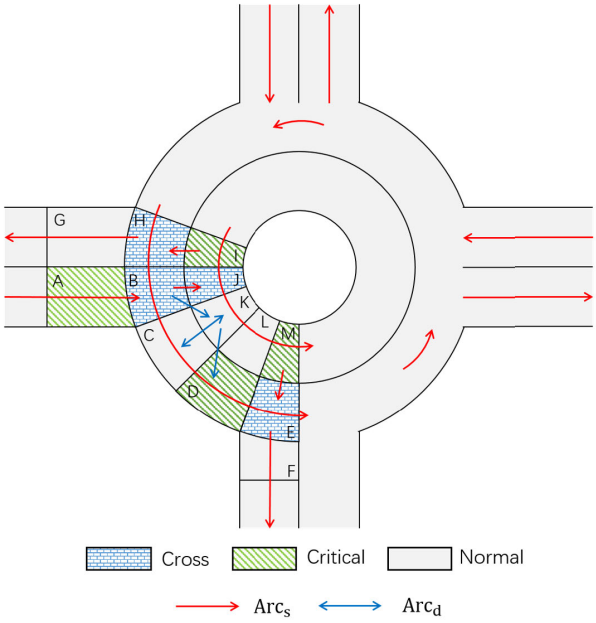


FIGURE 3. Static information in the abstract model for the multi-lane roundabout scenario.

data from observation, e.g. the vehicle ahead has different possibilities to turn left, turn right and go straight. Then we can specify the guard conditions via traffic data from both observation and reasonable calculation. In this part, we focus on the dynamic observed features, the estimation information is studied later in Section IV.

As defined below, the traffic snapshot structure captures dynamic observed information including the driving environment and the host vehicle itself, and it is updated once in every control period. The obstacles considered in this paper are only vehicles.

Definition 3 (Traffic Snapshot): Let Map be the map, self be the host vehicle with ADS, $\mathbb{C}$ *be the set of the other vehicles driving on the Map. The traffic snapshot is* $TS = (position, speed, acceleration, left, right)$ *where*

- $position : \mathbb{C} \cup \{self\} \rightarrow SM$ *is the function indicating the positions of vehicles in* $\mathbb{C} \cup \{self\}$.
- $speed : \mathbb{C} \cup \{self\} \rightarrow \mathbb{R}^+ \cup 0$ *is the function denoting the distance passed in a* ***control period***.
- $acceleration : \mathbb{C} \cup \{self\} \rightarrow \mathbb{R}$ *is the function that shows the value of acceleration by calculation upon captured speeds at this control period* t *and the former control period* $t - 1$. *For a vehicle* $c \in \mathbb{C}$, $acceleration(c) = (speed_t(c) - speed_{t-1}(c)) / n$ *where* n *is the cycle time in one period.* $n = 1$ *in this paper.*
- $left, right : \mathbb{C} \cup \{self\} \rightarrow Bool$ *is the boolean function representing the on-off states of the left and right turn signals respectively.*

We define the route for automated driving as a given *path* acquired from the routing module and observe the route every period to get the next target segment. The starting segment and the destination segment of the driving task are determined.

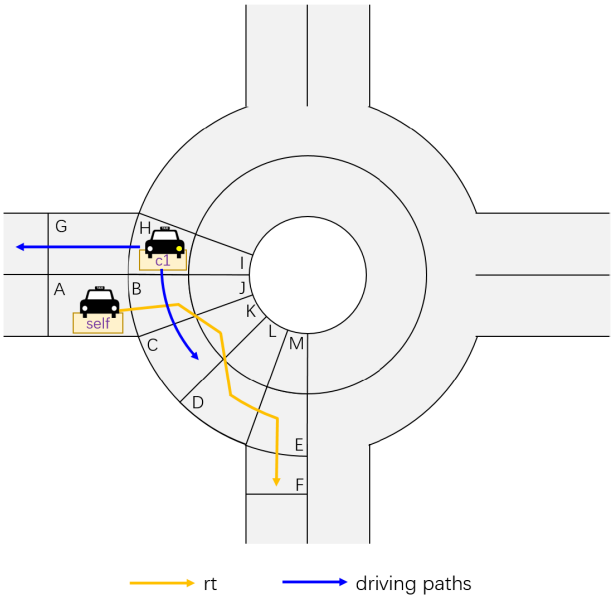


FIGURE 4. Dynamic information in the abstract model for the multi-lane roundabout scenario.

Definition 4 (Route): Let Map be the map, Path be the set of path on the Map. $rt :< rt_n >$ *is the specific path with finite length in Path.* rt_0 *is a determined starting segment and* rt_n *is a determined destination segment.*

- $rt :< rt_n >= \{rt_n \mid rt_n \in Segment \wedge (rt_n, rt_{n+1}) \in Arc \wedge n \in \mathbb{N}\}$

In this paper, we don't look into the process of the route planning, and the route is given during the driving task.

The dynamic observed information about the vehicle $c1$ for the moment is shown in Figure 4 where:

- $divSeg(position(c1)) = \{H\}$
- $speed(c1) = 10$
- $acceleration(c1) = 2$
- $left(c1) = false$
- $right(c1) = true$
- $rt :< A, B, K, D, E, F >$ is the assumed route where A is the starting segment and F is the destination segment.

TS captures the essential dynamic observed information, and the division of the segments in the length is exactly upon the speed and acceleration information of the host vehicle $self$. In addition, the speed and acceleration of any vehicle can not increase or decrease infinitely, their values are limited due to the driving performance. We divide the segment by the safe distance SD of the host vehicle. Now that the safe distance is composed of **stopping distance**, **reaction distance**, and **spacing distance**. Stopping distance is the distance from the time the brake operation works to the time the host vehicle stops. It can be defined as $sd = \frac{speed(self)^2}{2acceleration(self)_{max}}$ where $acceleration(self)_{max}$ is the maximum brake acceleration of $self$. Reaction distance is the distance moved during the reaction time for activating the brake operation according to the current speed. It can be defined as the $rd = speed(self) \times t + 0.5acceleration(self) \times t^2$ where t is the reaction time. Spacing distance θ is the intrinsic distance composed of length of the vehicle and the shortest space gap between the vehicles. Then, based on the safe distance $SD = sd + rd + \theta$, the length of the segment should be settled upon SD that vehicles can not go through two road segments in one period according to its maximum brake acceleration. Based on this assumption, the segments are set to the equal length for simplification of the calculation in the case study in Section VII.

III. CHARACTERIZATION OF SCENARIO

Speaking of keeping safety in the diverse driving task with different routes, we intend to find the divide-and-conquer method to decompose the whole driving task into subtasks related to the type of scenario. According to the observed features in the previous section, we can define the formal description of the scenario and the fundamental types of the scenario which serve as the base of scenario composition.

A. SCENARIO

The scenario can be defined if given the static connectivity between the segments based on the dynamic position of the host vehicle $divSeg(position(self))$. Reaching a segment in rt implies being in a scenario, so the type of scenario should be checked immediately once the occupied segment of host vehicle changes.

Here we give the **general formal definition for scenario** SS **along** rt, and we define the scenario as the set of the segment occupied by $self$, connected segments from the position of $self$ including the target segment, other segments linked to the target segment (interference segments), and the other segments connected to the interference segments except for the target segment.

Definition 5 (Scenario): SS denotes a scenario on the given Map. $n \in \mathbb{N}^+$, $TAR \in rt$, *and TAR denotes the target segment which satisfies* $TAR \in Next(divSeg(position(self)))$.

$$
\begin{aligned}
SS = {} & divSeg(position(self)) \\
& \cup Next(divSeg(position(self))) \\
& \cup \cdots \\
& \cup Next^n(divSeg(position(self))) \\
& \cup (Former(TAR) \setminus divSeg(position(self))) \\
& \cup \cdots \\
& \cup Former^{n-1}(Former(TAR) \setminus divSeg(position(self))) \\
& \cup (Next(Former(TAR) \setminus divSeg(\\
& \quad position(self))) \setminus \{TAR\}) \\
& \cup \cdots \\
& \cup Next(Former^{n-1}(Former(TAR) \setminus divSeg(\\
& \quad position(self)))) \setminus \{TAR\}
\end{aligned}
$$

Suppose that the vehicle can move through n segments in a period, which indicates the n layers in the observation range, the target segments in sequence contained in $Next(divSeg(position(self))) \cup \cdots \cup Next^n(divSeg(position(self)))$

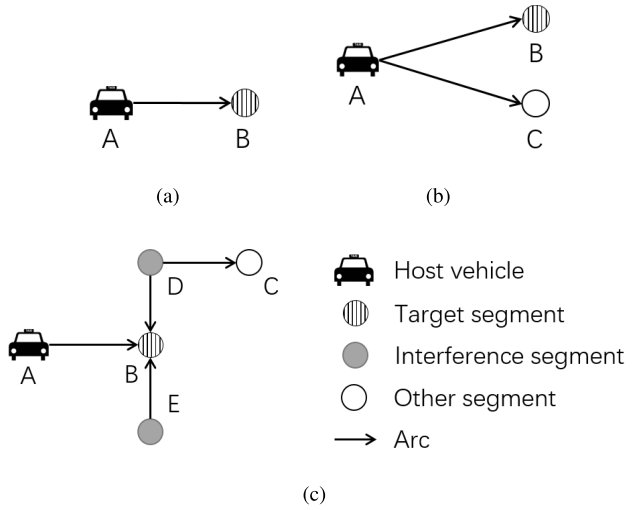


FIGURE 5. (a) is the single road structure. (b) is the fork structure. (c) is the junction structure with multiple interference segments.

should be considered. More segments connected to the interference segments in $(Former(TAR) \setminus divSeg(position(self))) \cup \cdots \cup Former^{n-1}(Former(TAR) \setminus divSeg(position(self)))$ should be taken into account. The other possible segments the interference vehicle may enter under sense area except the path towards the target segment are in $(Next(Former(TAR) \setminus divSeg(position(self))) \setminus \{TAR\}) \cup \cdots \cup Next(Former^{n-1}(Former(TAR) \setminus divSeg(position(self)))) \setminus \{TAR\}$. Then traffic data in more segments would be calculated during estimation for one period. To be clear, these segments can be detected if they are on the *Map*, and they can not be noticed if they are not on the *Map* even if they satisfy the connection relationship.

Upon the partition of segments put forward in Section II-C, the vehicle can only move from one segment to another in a period ($n = 1$), then **the scenario SS considered in this paper** is as follows:

$$\begin{aligned} SS = {} & divSeg(position(self)) \\ & \cup Next(divSeg(position(self))) \\ & \cup Former(TAR) \setminus divSeg(position(self)) \\ & \cup Next(Former(TAR) \setminus divSeg(\\ & \quad position(self))) \setminus \{TAR\} \end{aligned}$$

B. FUNDAMENTAL SCENE STRUCTURES

Fundamental scene structures are the basic units composing the complex scenarios based on both static and dynamic features. That is to say, the entire driving task is regarded as the composition of the subtasks in the subscenarios based on the fundamental scene structures along the rt.

As shown in Figure 5, A stands for the occupied segment of the host vehicle $self$, B represents the target segment of the host vehicle, C is the other irrelevant segment, and D, E are the interference segments that are connected to the same target segment with the host vehicle.

Then the **three fundamental scene structures are scenarios satisfying particular conditions respectively**.

1) SINGLE ROAD STRUCTURE

The single road structure meets the following conditions:

$$\begin{aligned} SS \models {} & Next(divSeg(position(self))) = \{TAR\} \\ & \wedge\ Former(TAR) \setminus divSeg(position(self)) = \emptyset \end{aligned}$$

As Figure 5(a) shows, the target segment based on rt is the only connected segment ahead of the host vehicle, and no other segment is connected to the target segment except the segment occupied by the host vehicle. So there are no interference segments and interference vehicles. In this structure, we could only observe the traffic in the target segment, and move forward if no vehicle is in the target segment TAR ahead.

2) FORK STRUCTURE

The fork structure meets the following conditions:

$$\begin{aligned} SS \models {} & card(Next(divSeg(position(self)))) > 1 \\ & \wedge Former(TAR) \setminus divSeg(position(self)) = \emptyset \end{aligned}$$

As Figure 5(b) shows, the occupied segment of the host vehicle is connected to multiple segments including the target segment. Since the vehicle will move cling to the planned route rt, the other connected segments are irrelevant, and no other segments are connected to the target segment. So there are no interference segments and interference vehicles. In this structure, we could only observe the traffic in the target segment just as in the single road structure.

3) JUNCTION STRUCTURE

The junction structure meets the following conditions:

$$SS \models card(Former(TAR) \setminus divSeg(position(self))) \geq 1$$

As Figure 5(c) shows, this is the junction structure with two interference segments. In the junction structure, there exist interference segments connected to the target segment other than the segment occupied by the host vehicle. Traffic in the target segment and the interference segments in $Former(TAR)$ should be observed to make a considerate and reasonable driving decision. First of all, check the quantity of the interference segment. Then, confirm the existence of vehicles in the interference segments and the target segment, and calculate the driving decisions of the interference vehicles one by one if the interference vehicles exist. Finally, make the driving decisions considering the observed information and estimation information. The occupation of the target segment or the high possibility for the interference vehicle to enter the target segment will stop the host vehicle for safety guarantee.

With the help of the fundamental scene structures, the estimation calculated from the observation according to the scenarios is patterned.

IV. ESTIMATION INFORMATION IN THE SCENARIO-BASED ABSTRACT MODEL

Absolute safety is elusive due to the inherent uncertainty in the driving environment, even the observed information can tell the current driving statuses of the host vehicle *self* and the other vehicles in $\mathbb{C}$. **The uncertainty mainly comes from the following two aspects**: **i) the driving decisions of other vehicles are uncertain and unobservable, and ii) the cognition from the observed data may be inconsistent with the intentions of the vehicles.** For instance, the vehicle in the interference segment may enter the target segment of *self* when *self* approaches the target segment at the same time, the manned vehicle turns right with the left turn light on, etc. All these unexpected stochastic behaviours and misconception contribute to the difficulties of making safe driving decisions. So achieving relative safety as far as possible is studied in this paper.

To cope with the uncertainty, estimation information is proposed in the abstract model. **To estimate the probabilities of the driving decisions of the surrounded vehicles, estimation is the calculated information based on the current observation at each period and updated in accordance with the latest observation.**

A. SCENARIO-BASED ESTIMATION PROCESS

The estimation process is in **two main steps** as follows. Estimation should obey the rules that the segment can only be occupied with one vehicle at one time.

1) FIGURE OUT THE INTERFERENCE SEGMENTS AND INTERFERENCE VEHICLES BASED ON THE SCENE STRUCTURES

Based on the driving intentions of *self* and scene structures, we can easily figure out the interference segments and interference vehicles.

a: SINGLE ROAD, FORK

In the single road structure and fork structure, there are no interference segments. So the final driving decision depends on the observation only, which means the vacancy in the target segment allows for moving forward and the following estimation step is unnecessary.

b: JUNCTION

In the junction structure, there is at least one interference segment. First, check if there exists a vehicle in the target segment. The following steps can be omitted if the target segment is already occupied. Then, check if there is an interference vehicle in each interference segment. If the interference segment is vacant, just calculate the moving distance of *self* in *movDistance*() defined below to decide to move out or not. If the interference segment is occupied, take the next step described below.

Here, we take the multi-lane roundabout scenario in Figure 4 as an example. At the current step, in this junction with one interference segment, we get the observed information as follows:

i) occupied segment of *self*: $divSeg(position(self)) = A$,
ii) the target segment: B,
iii) the interference segment: H, and
iv) the interference vehicle: $c1$.

2) CALCULATE THE PROBABILITIES OF THE POSSIBLE DRIVING DECISIONS FOR THE INTERFERENCE VEHICLES

When in a junction, the target segment is empty and at least one interference vehicle exists, we follow the second step. We model the estimated probabilities for driving decisions of the interference vehicles in *Prob*. It depends on the estimation process based on the calculation of possible moving distances *movDistance*, possible driving paths *drivingPath* and probability distribution of various reachable segments upon *accesPosition*. In *Prob*, we adopt probability distribution as an explicit expression of the stochastic driving decisions of the interference vehicles. The data in the *Prob* combined with the observation in the abstract model can be used to judge the satisfaction of the guard conditions for the corresponding driving decisions.

First, compute how long the interference vehicle $c1$ and the host vehicle *self* can drive during the next period by *movDistance*(). The results show whether the current speed of *self* and $c1$ can enable them to move out of the current segment. As described in Section II, the passing distance inside the segment can be regarded as the distance on the centre line according to the driving direction. To calculate the accumulative distance based on the computational method, the formula for uniformly accelerated motion is used ($movDistance(c1) = speed(c1) \times t + 0.5 acceleration(c1) \times t^2$, t is one period) in this paper, while other kinds of formulas can be used due to the requirement of the system. Given the *Map* and *Path*, *TS* at this control period is acquired, $\mathcal{C}$ is the set of the interference vehicles in the scene structure that $\mathcal{C} \subset \mathbb{C}$.

- $movDistance : \mathcal{C} \cup \{self\} \rightarrow \mathbb{R}^+$ is the function extracts the distances the interference vehicle and *self* pass according to the computation result on the basis of *position*, *speed*, and *acceleration* data during the current period.

If it is possible for *self* and $c1$ to reach the target segment in the next period, we should calculate the value of probabilities for the driving decisions of the interference vehicle in each interference segment.

Then, find out the stochastic driving decisions of the interference vehicle. Actually, the driving decisions can be described as the different sequences of passed segments $< H, G, \cdots >, < H, B, \cdots >$ extracted by $drivingPath(c1)$ from the position of the interference vehicle.

- $drivingPath : \mathcal{C} \rightarrow \mathcal{P}(Path)$ denotes the possible driving paths for each $c \in \mathcal{C}$ starting from the present position $divSeg(position(c))$.

Now, we can calculate the new positions of the interference vehicle based on $drivingPath(c1)$ and $movDistance(c1)$. If the

value of distance in $movDistance(c1)$ exceeds the length of the current segment according to the driving path, its new position will be in a new segment; otherwise, it will be in the new position in the current segment. *accesPosition* is defined to estimate the possible positions which can be abstracted to the possible segments.

- $accesPosition : \mathcal{C} \times drivingPath(\mathcal{C}) \times movDistance(\mathcal{C}) \rightarrow \mathcal{P}(SM)$ is the function indicating the possible positions for each $c \in \mathcal{C}$, after moving through $movDistance(c)$ in various routes in $drivingPath(c)$.

Then possible paths could be reduced to paths ended within segments in $divSeg(accesPosition(c, drivingPath(c), movDistance(c)))$.

After the series of calculation, we can describe the probabilities for stochastic driving decisions of each interference vehicle in *Prob*.

Definition 6 (Probability Distribution for Driving Decisions): Given the Map and Path, TS at this control period is acquired, $\mathcal{C}$ is the set of the interference vehicles driving on the Map. For the host vehicle self, its estimation of the driving decisions of the interference vehicles is in Prob.

- $Prob : \mathcal{C} \times divSeg(accesPosition(\mathcal{C}, drivingPath(\mathcal{C}), movDistance(\mathcal{C}))) \rightarrow [0, 1]$ *denotes the estimated probability distribution of possible reachable segments along the various driving paths for each interference vehicle in the next period.*

$\exists c \in \mathcal{C} \cdot \Sigma_{s \in divSeg(accesPosition(c, drivingPath(c), movDistance(c)))} Prob(c, s) = 1.$

For the interference vehicle $c1$, it is obvious in Figure 4 that:

- $movDistance(c1) = speed(c1) \times 1 + 0.5 acceleration(c1) \times 1^2$
- $divSeg(accesPosition(c1, drivingPath(c1), movDistance(c1))) = \{G, B\}$
- $Prob(c1, G) + Prob(c1, B) = 1$

Now that we have got the description of the stochastic driving decisions of the interference vehicle, the value of *Prob* is obtained from the prior probability distribution and the successor modification.

B. CALCULATION OF PROBABILITIES FOR STOCHASTIC DRIVING DECISIONS

As shown in Figure 4, we can get the prior probability distribution of driving behaviours from the historical data as the basis of probability modification, due to the possible reachable segments in $drivingPath(c1)$ calculated from $accesPosition(c1, drivingPath(c1), movDistance(c1))$ in the next period.

1) THE PRIOR PROBABILITY DISTRIBUTION OF DRIVING DECISIONS FOR THE INTERFERENCE VEHICLE

For each interference vehicle in the interference segment, the possible driving paths from the current segment and the feasible vehicle behaviours are determinant. Thus, we define the probability distribution as follows.

Definition 7 (Probability Distribution of Driving Decisions): $ST = \{A_1, A_2, \ldots, A_n\}$ is the universal set of driving decisions of one interference vehicle c based on the its current occupied segment, and each element in ST means a decision of moving to the estimated segment $divSeg(accesPosition(c, drivingPath(c), movDistance(c)))$. $A_1, A_2, \ldots, A_n$ are collectively exhaustive since $A_1, A_2, \ldots, A_n$ are mutually exclusive, and the union of decisions is the universal set where

(1) $A_i \cap A_j = \emptyset$ $(i \neq j \wedge i, j \in \mathbb{N}^+)$;

(2) $A_1 \cup A_2 \cup \ldots \cup A_n = ST$. The probability distribution for $A_1, A_2, \ldots, A_n$ is that $\Sigma_{i=1}^{n} P(A_i) = 1$ $(P(A_i) > 0)$ where $P(A_i)$ stands for the probability of driving decision A_i for the interference vehicle.

The prior values for the probability distribution is from the knowledge library. It is necessary to note that, we assume that there is a knowledge library (or maybe a cloud centre, a database in the wayside station, etc.) which stores and updates history data for probability distributions in this paper. The probability distribution has two main sources. One is the comprehensive statistical data of the interference vehicle collected from similar scenarios based on its history driving decisions. The other is the statistical data of other vehicles collected from similar scenarios for reference, and these data are set as the reference data in case the interference vehicle has no such data in the scenario. This prior probability distribution is used as the base of modification.

In Figure 4, $ST = \{turn, forward\}$. The value of $P(turn)$ is obtained for approachable segment G, and the value of $P(forward)$ is obtained for approachable segment B. Definitely, $P(turn) + P(forward) = 1$.

2) MODIFICATION OF PROBABILITY BASED ON TS AND APPROACHABLE DISTANCE

Probability is modified upon **observed traffic data** and the **possible moving distance** of the interference vehicles. In the following part, a rational modification is introduced, while the modification method can be varied due to the system requirement based on our estimation scheme.

(1) To achieve a more precise estimation, light status is considered to modify the prior probability distribution. For the interference vehicle c, it is feasible that the probability of heading left rises when $left(c) = true$. Correspondingly, the probabilities of other decisions of c must be decreased to maintain the overall sum of probabilities as 1.

(2) Now we further consider the emergencies in the same driving path. For instance, the interference vehicle decides to turn left with a sudden acceleration at the maximum speed, deceleration at the maximum speed or original acceleration. We consider these sudden situations and adopt the same modification method as light status. That is to say, the probability of the vehicle moving to the next segment rises when the vehicle is able to move outside the current segment even by deceleration. The rational estimation of $movDistance(c)$ with sudden speed change is for addition when the intention is determined. And this process is put forward to keep the

high autonomy of the host vehicle while relative safety is guaranteed.

Definition 8 (Modification Method of the Probability Distribution): Let $ST = \{A_1, A_2, \ldots, A_n\}$ *be the universal set of driving decisions of one interference vehicle c based on its current occupied segment, and each decision of moving to the estimated segment* $divSeg(accesPosition(c, drivingPath(c), movDistance(c)))$ *is an element in ST.* $p(A_i) \in (-1, 1)$ $(i = 1, 2, \ldots, n)$ *denotes the modification parameter of the corresponding probability. And the instructions of modification according to the* ***observed traffic data*** *and the* ***possible moving distance*** *is for example in guarded command language as follows.* ϕ *is the specification of guard condition from observation or estimation.*

$P(A_1) + P(A_2) + \ldots + P(A_n) := 1$

$p(A_1) + p(A_2) + \ldots + p(A_n) := 0$

if $\phi \rightarrow$

$P(A_1), P(A_2), \ldots, P(A_n) := P(A_1) + p(A_1), P(A_2) + p(A_2), \ldots, P(A_n) + p(A_n)$

fi.

For ϕ in different types, ϕ_{obs} is the specification of observed condition such as light status, while ϕ_{cal} is the specification of estimated condition such as possible moving distances of the interference vehicle. Suppose that A_1 is a decision of c for turning left in this case, $p(A_1)$ must be positive if ϕ_{obs} : $left(c) = true$; otherwise, $p(A_1)$ must be negative if ϕ_{cal} : $movDistance_{maxacc}(c) < segLen$. It means that c is not able to move out of the current segment even it accelerates. *maxacc* stands for the maximum acceleration, and *segLen* is the length of current segment.

The estimation process is calculated based on the observed data to cope with the stochastic driving behaviours of the interference vehicles. For the host vehicle, it is much considerate to consider both traffic data from observation and estimation before making the driving decision.

V. VERIFICATION

Now we have got the abstract model which includes the static (*Map*) and dynamic traffic data (*TS*, *rt*, *Prob*) from real-time observation and estimation, and three fundamental scene structures which are scenarios satisfying specific spatial conditions, according to the decision-making process. That is to say, we are in need of the model verification method upon our abstract model to complete the safety assessment scheme. In this section, we first propose our model verification method for the abstract model and then put forward the mapping rules from the abstract model to SHA for accomplishing the automatic verification.

A. MODEL VERIFICATION

During the driving task, the host vehicle *self* makes a sequence of driving decisions due to traffic condition along *rt*. To verify driving decisions means checking whether the safety property is satisfied in the sequence of driving decisions in the sequential scenario transitions until arriving at the destination in *rt*.

A driving decision of *self* considers both observation and estimation including: i) the observed information received at the current control period, ii) local knowledge inherited from the previous state, and iii) the reasonable driving decisions of vehicles surrounding *self*. As mentioned in Section I, decision-making process for automated driving is a **periodic control system** according to the **scenario-based abstract model** in **each period**. Apparently, the process operates as the combination of continuous vehicle dynamics evolution and discrete periodic control signals. In addition, the stochastic driving behaviours of the other vehicles give rise to the uncertainty of the environment, from the perspective of the host vehicle *self*.

The aim of the model verification here is relative safety. In other words, we verify that the safety property with probability holds in all reachable states of our abstract model. The value of probability is a given safety threshold, e.g. probability of the collision rate should be lower than 0.2. Suppose that *SAFE* stands for the safety property should hold in the model, *self* stands for the host vehicle, we assess safety by checking whether the following formula holds:

$$Map, TS, rt, Prob, self \models SAFE$$

It means that the model is safe enough for the host vehicle *self* driving according to the route *rt* if this formula holds. According to our divide-and-conquer and scenario-based verification method, we should guarantee that *SAFE* holds in the same scenario and between scenario transitions. Then there exist three conditions to verify.

1) The first is the initial state in the initial scenario, then

$$Map, TS_0, rt, Prob, self \models SAFE.$$

2) The second is the transition in the same scenario based on the same occupied segment seg_1 of *self*. In this condition, we first check the type of scenario which is one of the fundamental scene structures according to Section III. Since *Map* is static and *self* is the host vehicle, for each transition $TS, rt, Prob \xrightarrow{guard, action} TS', rt', Prob'$, we assume that $AM = (TS, rt, Prob)$. And the jumping function for evolution of continuous variable and the weight of the uncertain behaviour are implied in *action* here. For each transition, $TS_1 \neq TS_2$, $seg_1 \rightarrow TS_1$, and $seg_1 \rightarrow TS_2$ where the occupied segment seg_1 of the host vehicle is calculated from $seg = divSeg(position(self))$.

$$Map, TS_1, rt, Prob, self \models (SAFE \wedge guard) \wedge action(AM, AM') \Rightarrow Map, TS_2, rt', Prob', self \models SAFE$$

3) The third is the transition from the original segment seg_1 to the new segment seg_2 when transiting to the new scenario. For each transition, $TS_1 \neq TS_2$, $seg_1 \neq seg_2$, $seg_1 \rightarrow TS_1$, $seg_2 \rightarrow TS_2$, and $seg_2 \in Next(seg_1)$ where the values of occupied segment of the host vehicle seg_1 and seg_2 are

calculated from $seg = divSeg(position(self))$.

$$Map, TS_1, rt, Prob, self \models (SAFE \wedge guard) \wedge action(AM, AM') \Rightarrow Map, TS_2, rt', Prob', self \models SAFE$$

The driving environment is the synchronous actions of the host vehicle and the other vehicles. When *self* reaches the target segment in this scenario, the original target segment becomes the currently occupied segment, and the target segment is updated. We can control the host vehicle *self*, but only observe and get information about the other vehicles in each period. To conclude, we should guarantee that the safety property holds in the scenario and scenario transitions between fundamental scene structures. Then we can verify the model upon the composite scenario composed of sub-scenarios.

To enable automated model checking of the abstract model with hybrid and stochastic features, and make the approach applicable in the industry, the automated model checking tool is needed. SHA combines continuous system dynamics, stochastic alternatives, and real-time behaviours etc. At the core of UPPAAL SMC, via broadcast channels and shared variables, the model runs as the network of dynamic instantiation of templates which are defined as SHA [16]. Our model is defined as a periodic system and run as the simultaneous processes of the host vehicle and the environment (other surrounding vehicles). In each period, the traffic data from the synchronous driving vehicles are fetched in this scenario. When transiting to the next scenario type in sequence, new traffic data and scenario information are fetched, though the calculation of data is varied. Therefore, we apply the verification approach by mapping from the abstract model to SHA which is supported in UPPAAL SMC.

B. MAPPING RELATION FROM ABSTRACT MODEL TO SHA

In this part, we suggest the verification approach based on UPPAAL SMC by mapping from the abstract model to SHA. This is the advanced and more specific approach based on our previous work [17], [18].

Definition 9 (Verification Approach): A stochastic hybrid automaton is described by a tuple $M = (Loc, loc_0, \mathcal{Q}, Var, Inv, \mathcal{A}, \mathcal{G}, \mathcal{T})$ *where the meanings of the symbols and the mapping between the abstract model and SHA is as follows based on given Map, traffic snapshot TS, estimation for stochastic driving behaviours in Prob, the host vehicle self and the set of vehicles* $\mathbb{C}$ *in the driving environment.*

- $Loc = divSeg(position(\mathbb{C} \cup \{self\})) \cup \circledcirc$ is the finite set for discrete states or locations. It indicates the occupied segments of the vehicles according to *Segment* in the *Map*. $\circledcirc$ denotes the non-existence of vehicles.
- loc_0 is the initial location, which denotes the initial locations of the vehicles in $\mathbb{C} \cup \{self\}$.
- $\mathcal{Q} = \{q | q(t) = TS \wedge t \in [0, \epsilon]\}$ is the **continuous state space** of the SHA over the time interval $[0, \epsilon]$ where ϵ is a non-negative number. In $\mathcal{Q}$, the continuous state variables in the **continuous state** q take their values at time t based on the vehicle dynamics in $TS = (position, speed, acceleration, left, right)$ of all vehicles on the *Map*.
- Var is the finite set of discrete variables such as static information in the *Map*.
- $Inv : Loc \to \mathcal{P}(\mathcal{Q})$ is a mapping from the locations in Loc to the set of subsets of $\mathcal{Q}$. It means that $Inv(loc) \subset \mathcal{Q}$ ($loc \in Loc$), and $Inv(loc)$ is the **location invariant** for $loc \in Loc$. When the system reaches location loc, the continuous state q must satisfy $q \in Inv(loc)$. In our model, $loc = divSeg(position(c))$ is the location for the vehicle c, then its passed distance in the location loc should not exceed the length of loc; otherwise, c violate the location variant.
- $\mathcal{A} = Act_{self} \cup Act_{\mathbb{C}}$ is the finite set of actions of vehicles where Act_{self} denotes the actions of the host vehicle *self*, $Act_{\mathbb{C}}$ denotes the actions of the other vehicles in $\mathbb{C}$.

For *self*, its decisions are determined, the decision $a \in Act_{self}$ can only be triggered when the corresponding $g \in \mathcal{G}$ is satisfied.

While for the vehicle $c \in \mathbb{C}$, its stochastic driving decisions leads to the uncertain subsequent locations. Based on $loc = divSeg(position(c))$, they can make several reasonable decisions A_i in ST with the weight w_i as put forward in Section IV and $ST \subset Act_{\mathbb{C}}$.

When action a is taken, such as entering the roundabout, it implies the activation of the continuous vehicle dynamics change in $\mathcal{Q}$ such as speed etc.

- $\mathcal{G} = \mathcal{Q}_{sub} \cup Var_{sub} \cup Prob$ is the set of guard conditions depends on the spatial conditions consisting of **observation** of the continuous variables in $\mathcal{Q}_{sub}$ and discrete variables in Var_{sub} and **estimation** in $Prob$ where $\mathcal{Q}_{sub} \subset \mathcal{Q}$, $Var_{sub} \subset Var$.
- $\mathcal{T} \subseteq Loc \times \mathcal{G} \times \mathcal{P}(\mathcal{A}) \times \mathcal{P}(\mathcal{J}) \times \mathcal{P}(W(\mathcal{A} \times Loc)) \times \mathcal{P}(Loc)$ is the finite set of transitions in M.

$\mathcal{P}$ is the power set identifier. $\mathcal{J} = \{(q, q') | (q, q') \subset \mathcal{Q} \times \mathcal{Q}\}$ $(q, q' \in \mathcal{Q})$ is the set of relations on the continuous states. In function $W : \mathcal{A} \times Loc \to \mathbb{N}^+$, the weight is attached to actions and their corresponding post locations. In the transition $tr \in \mathcal{T}$ of M, it means the corresponding g is satisfied in location loc, the allowed action a with the weight $w \in W(a, loc')$ is triggered, the transformation $j \in \mathcal{J}$ from the continuous state q to q' is achieved, and it is transited to the subsequent location loc'.

Because the stochastic behaviours are from the other vehicles in the environment while the decision of *self* is determined, the actions with weight values are modelled in automata of the other vehicles.

Given a state $loc \in Loc$ at the current period, the transition of this automaton transfers from loc to a post-state $loc' \in Loc$ of loc with satisfied guard g and the corresponding action a with the weight w. From the state loc_0 of vehicle c, for instance, when g is satisfied, a_1, a_2 can be activated with weight $w_1 \in W(a_1, loc_1)$ and $w_2 \in W(a_2, loc_2)$, so $loc_0 \xrightarrow{g, a_1, j_1, w_1} loc_1$ and $loc_0 \xrightarrow{g, a_2, j_2, w_2} loc_2$. As $Prob(c, segment_1) = 0.5$ and $Prob(c, segment_2) = 0.5$

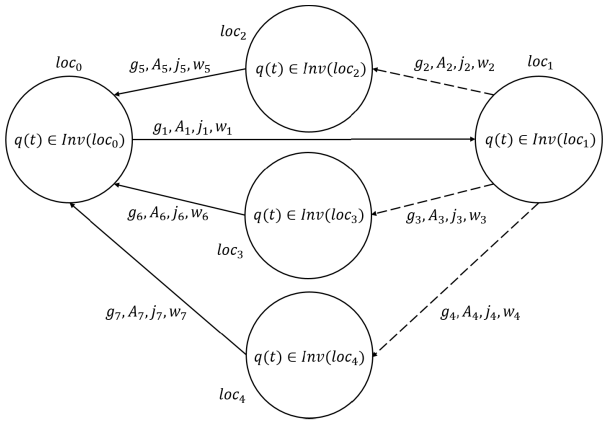


FIGURE 6. Mapping from the abstract model to SHA.

in *Prob* are probability values, we change these values to weight values through the probabilistic representation from $\frac{w_1}{w_1+w_2}$ and $\frac{w_2}{w_1+w_2}$ to enable the weight representation in the verification.

Figure 6 shows the description of our model in SHA after mapping from the abstract model. It is obviously an automaton for the interference vehicle with stochastic behaviours where $w_1 = w_5 = w_6 = w_7 = 1$, $ST = \{A_2, A_3, A_4\}$, and $w_2 + w_3 + w_4 = 1$.

In fact, we do not dig into the evolution (ODEs) of vehicle dynamics but focus on the control, so we do not give the explicit calculation method of the continuous variables due to its variety. For the network of SHAs, we construct the communication among SHAs through broadcast channel and shared variables.

According to the definition mentioned above, we also summarize the mapping relation between features in our abstract model and SHA as shown in Table 1, in order to explain generality and help the users to complete the mapping when the features in the abstract model have been obtained. For the host vehicle *self*, its values of the variables should be consistent with its driving route until it reaches the destination. But for the other vehicles, the reachable segments are only related to the road topological structure. The mapping relation is applicable for all the vehicles appearing on the given *Map*.

C. PROPERTY

For the model as the network of SHAs *MM*, statistical model checking monitors runs of the model with respect to some properties, and then use results from the statistics to get an overall estimate of the correctness of the design. The approach has been applied to verify both qualitative and quantitative properties. Qualitative property checks the probability that a random run of model satisfies the property greater or equal to a certain threshold. Quantitative property verifies what is the probability that a random run of model satisfies the property.

In this paper, we define the safety as the only presence of one vehicle in a segment in a period. Then, the presence

TABLE 1. Mapping relation between features in the abstract model and SHA.

The Abstract Model	SHA
the reachable segments of all vehicles	Loc
the starting segments of all vehicles	loc_0
the values of continuous variables	$\mathcal{Q}$
the values of discrete variables	Var
the values of the continuous variables should be satisfied in the corresponding reachable segments	Inv
the driving decisions	$\mathcal{A}$
the guard conditions where the values of the discrete and continuous variables and estimated probabilities of decisions of all vehicles can be observed and calculated to allow an action	$\mathcal{G}$
the transition between the reachable segments with guard, action, jumping functions for evolution of the continuous variables, and the probability of the transition	$\mathcal{T}$

of two vehicles in the same segment in the same period indicates a (potential) collision. To verify the safety of the driving decisions, we assume the vehicles in the scenario all obey the traffic rules, and the danger is mainly caused by the unexpected behaviours in the driving environment. When $c1 \in \mathbb{C}$, then we specify the safety property as

$$\psi : \neg\exists c1 \neq self \cdot divSeg(position(c1)) \cap divSeg(position(self)) \neq \emptyset$$

According to the uncertainty of the driving environment mentioned before, how much ψ is satisfied in the sequence of driving decisions is the measure of relative safety of the driving decisions.

The expression of properties in UPPAAL SMC is a simplified version of Dynamic Metric Temporal Logic (DMTL) [16], [19] which can specify state formulae and path formulae. State formulae describe individual states, whereas path formulae quantify over paths or traces of the model. With regard to the symbolic model checking, relative safety can be checked in UPPAAL SMC. In addition to the stochastic behaviours, we can check three kinds of properties mapping from ψ. φ and ϕ are the properties specified in DMTL. Mapping from ψ to φ in DMTL, we check the probability confidence interval of collision in

$$\begin{aligned}\varphi : \ & divSeg(position(c1)) == \{TAR\} \\ \&\& \ & divSeg(position(self)) == \{TAR\} \\ \&\& \ & c1 \neq self \\ \&\& \ & c1 \in \mathbb{C}\end{aligned}$$

conversely within the time bound of clock x.

$Pr_{MM}(\diamond_{x \leq bound}\varphi) >= \gamma$: the probability that there is another vehicle in the same segment with *self* is greater than

the probability γ, through runs of paths in *MM* before the time bound *bound* for the approximation of estimation of probability. The satisfaction of this property indicates the unsafe driving decisions leading to high probability of collision compared to the threshold.

$Pr_{MM}(\diamond_{x\leq bound_1}\varphi) >= Pr_{MM}(\diamond_{y\leq bound_2}\phi)$: without calculation of value of the probabilities, compare the probability values of φ and ϕ through runs of paths in *MM* before the time bound $bound_1$ and $bound_2$ respectively, x and y are clocks.

$Pr_{MM}(\diamond_{x\leq bound}\varphi)$: the probability confidence interval that there is another vehicle in the same segment with *self* through runs of paths in *MM* before the time bound *bound*. Compared to the first two specifications, we obtain the exact interval values of potential collision.

Mapping from the abstract model and property to SHAs and DMTL property, our safety assessment scheme is applicable in the case with the help of automated verification tool based on the formal model.

VI. SCENARIO-BASED APPROACH

Now that we have introduced the structure and content of our safety assessment scheme as shown in Figure 1 in the previous sections, we will give a comprehensive description of the processing steps for our approach in detail. In this section, we will show how the scenario works in the modelling phase and verification phase, and show the composability of the scenarios in multi-lane roundabout shown in Figure 3 based upon three cases on fundamental scene structures.

A. PROCESSING STEPS

As a preliminary, we obtain the identifier *self* for the autonomous vehicle and its starting and destination position. The real map in coordinates is required and the abstraction of the real positions and the naming system are done in advance. Then the set of segments and the connection relationship can be stored in *Map* in our abstract formal model. Now the initial route *rt* is also determined due to the *path* on the *Map*. At the same time, some historical records of *self* are copied. The features and structures of *Map*, *TS*, *Prob* in the abstract model are fixed. Other real-time data *TS* from observation and estimation *Prob* can only be gathered and calculated upon the coming period when running the model since our model is a periodic system.

1) PROCESSING STEPS IN THE MODEL

i) When a new period comes, gather the real-time sensor data of *self* and the surrounding vehicles from the onboard and wayside sensors, and feed the data into *TS*. Check the target segment in *rt*.

ii) Calculate the occupied segment of *self* upon position information in *TS*, check the type of scenario based on *Map*, and calculate *Prob* based on *TS* in different procedure shown in Section IV.

a) If the scenario is a single road or a fork, omit the calculation of *Prod*.

b) If the scenario is a junction, calculate *Prob* if there is an interference vehicle in the interference segment. Repeat the calculation if there are multiple interference vehicles in the multiple interference segments.

iii) A decision is given according to real-time observation *TS* and the probabilities of all possible driving decisions for all interference vehicles in *Prob*.

iv) Repeat step i) to iii) in each period until *self* reaches the destination in *rt*.

According to the steps above, writing computer programs to feed data into the model is necessary. While manual intervention is not required since the structure and content of our abstract model is determined.

To verify that the safety property holds in the sequence of decisions, the processing steps are shown below.

2) PROCESSING STEPS IN VERIFICATION

i) Specify the safety properties based on the system requirements.

ii) Mapping the abstract model and property to the automated model checking tool UPPAAL SMC.

iii) Check automatically by verifying the safety of each decision in the connected scenario in the simulation of running paths as shown in Section V.

By the above steps, we can assess the safety of the driving decision. In the verification phase, manual intervention is needed in step i) and ii), while iii) is automatic by the tool.

B. APPLICATION IN FUNDAMENTAL SCENE STRUCTURES

To illustrate our approach, we show three cases of fundamental scene structures extracted from the multi-lane roundabout discussed in Section VII. Through these cases, we show the composability of scenario based upon the connection of fundamental scene structures.

Since the driving route *rt* for *self* is $< A, B, K, D, E, F >$, we divide the driving task into subtasks on $< A, B >$, $< B, K > \ldots < E, F >$ as shown in Figure 7. If the subtask in each scenario is verified that the collision rate is lower than 0.2, then the whole task is safe. As Figure 5 shows in Section III, we can judge the scenario respectively in these three subtasks easily, according to the occupied segment of *self*.

Figure 7(a) is the junction structure with one interference segment H. A stands for the occupied segment of the host vehicle *self*, B represents the target segment of the host vehicle, and G is the other segment for the interference vehicle (if exist) to leave. Figure 8 shows that the collision rate is low (referred to the result of the second property) while completing the subtask $< A, B >$ (referred to the result of the first property), so it is safe in this scenario.

Figure 7(b) is the junction structure with multiple interference segments C and J. B stands for the occupied segment of the host vehicle *self*, K represents the target segment of the host vehicle, and D is the other segment for the interference vehicle in C (if exist) to leave. Figure 9 shows that

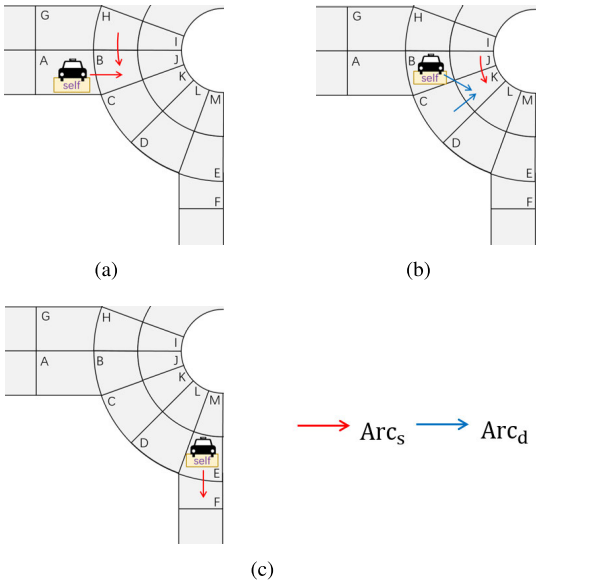


FIGURE 7. (a) is the junction structure with one interference segment. (b) is the junction structure with multiple interference segments. (c) is the single road structure.

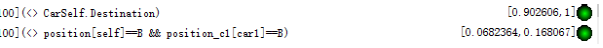


FIGURE 8. Verification results for properties in junction structure with single interference segments in 100 time units.

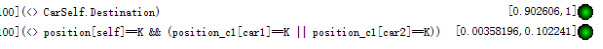


FIGURE 9. Verification results for properties in junction structure with multiple interference segments in 100 time units.

FIGURE 10. Verification results for properties in single road structure and fork structure in 100 time units.

the collision rate is low (referred to the result of the second property) while completing the subtask $< B, K >$ (referred to the result of the first property), so it is safe in this scenario.

Figure 7(c) is the single road structure. E stands for the occupied segment of the host vehicle *self*, F represents the target segment of the host vehicle, and there is no interference segments. Since the fork structure with a determined target segment is similar to the single road structure, the result of this case is also applicable for the fork structure. Figure 10 shows that the collision rate is little (referred to the result of the second property) while completing the subtask $< E, F >$ (referred to the result of the first property), so it is safe in this scenario.

As the three cases and processing steps mentioned above, the model of the host vehicle (with determined judge logic) and the model of traffic data in the target segment (with random simulation) are the same as the ones in Section VII as shown in Figure 11. While the models for traffic condition in the interference segments are different parts of the model shown in Figure 11(d) which contains all possible situations when considering traffic conditions in the interference segment. In the scenario shown in Figure 7(a), the interference vehicle has the other segment G to leave if it exists. Then *ToTgtSeg*1, *ToOutSeg*1, *ToStay*1, *ToOutSeg*2, and *ToStay*2 are reachable through part of the paths from the initial state. In the scenario shown in Figure 7(b), the interference segment C has another segment to leave, while segment J does not. Then all the states are reachable, its corresponding model should cover all possibilities as shown in Figure 11(d). In the scenario shown in Figure 7(c), there is no interference segment. So only *NoKeySeg* is reachable.

To summarize, the model in Figure 11(d) composes all possible states in various fundamental scene structures for the traffic conditions in the interference segment. **Then any scenario that can be described as the composition of fundamental scene structures can be modelled as Figure 11.** Since the safety property holds in $< A, B >$,$< B, K >$ and $< E, F >$, the model is safe if it is still satisfied in the corresponding scenarios of the subtask $< K, D >$ and $< D, E >$. To find whether we can achieve this, later in Section VII, we prove that the high collision rate in one scenario of a subtask will bring the danger to the whole driving task.

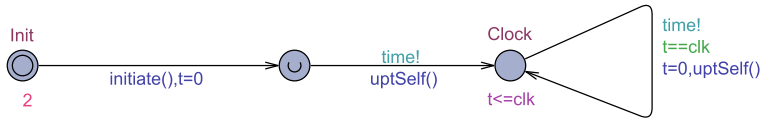
VII. CASE: MULTI-LANE ROUNDABOUT

Now that we have put forward the safety assessment scheme of the driving decisions in automated driving based on the formal modelling and verification approach, the modelling and verification of the decision-making in the multi-lane roundabout scenario are illustrated in UPPAAL SMC as a case study in this section.

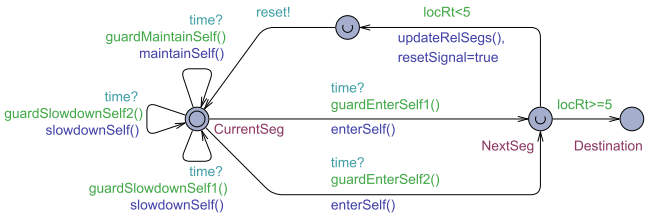
In the multi-lane roundabout scenario shown in Figure 4, the host vehicle *self* is initially at segment A approaching the roundabout, and its driving task is driving through the roundabout along the planned route $rt :< A, B, K, D, E, F >$ and finally reach the destination F. The segment is set to the equal length as 100 in the case. During the driving task, *self* obeys the traffic rules that it should not enter the target segment that has already been occupied by another vehicle, give way to the vehicles already in the roundabout, and vice versa for the other vehicles. Only *self* utilizes the estimation process based on the prior probability distributions of driving decisions for the interference vehicles, while the decision-making process of the interference vehicles is unknown to *self*. The prior probability distributions obtained from the knowledge library is not updated in this driving task. The final driving decisions and latest probability distributions of the interference vehicles are unobservable to *self*.

A. MODELLING IN UPPAAL SMC

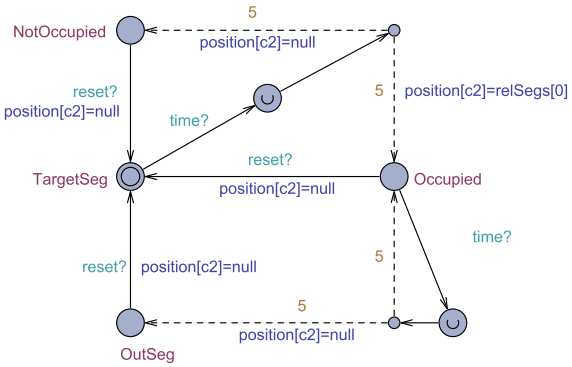
Single road structure, junction structure and fork structure are fundamental scene structures composing the multi-lane roundabout. Then the whole system can be considered as the network of automata where the traffic operating simultaneously in the occupied segment of the host vehicle, the target segment of the host vehicle, the interference segment(s) of the host vehicle and a period controller, due to the previous description of driving scenario and fundamental



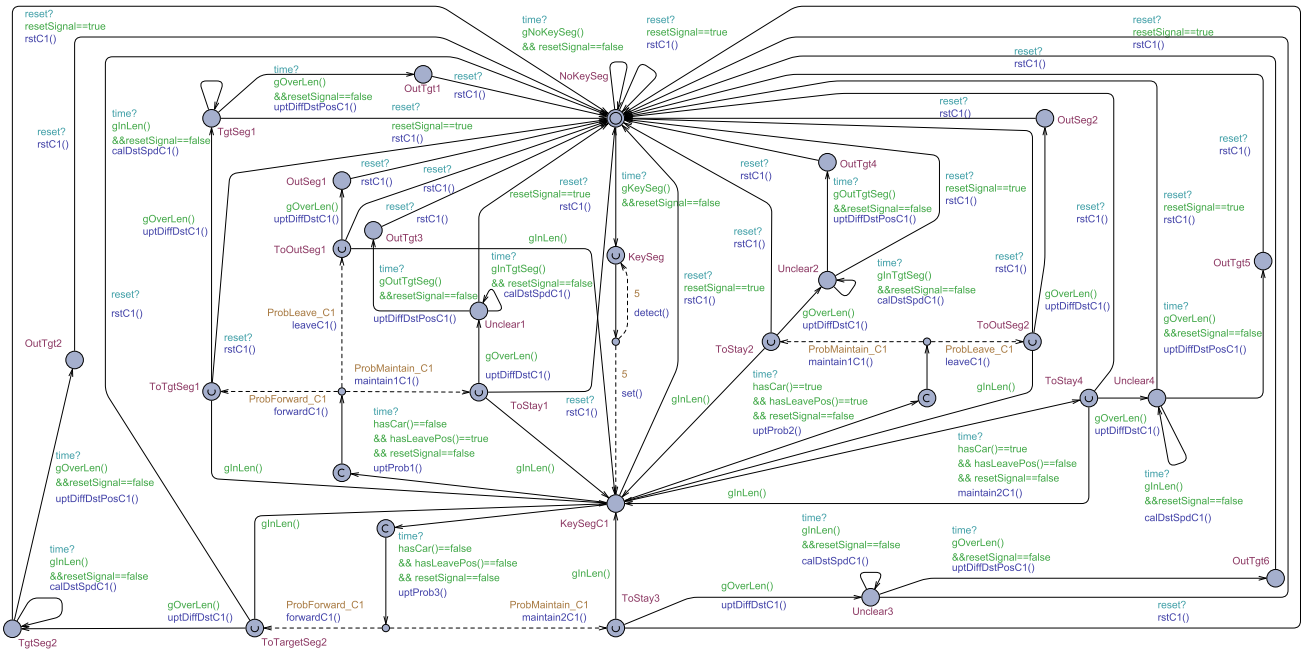
(a) Automaton $Period$ for period control.



(b) Automaton $CarSelf$ for the host vehicle with rational estimation.



(c) Automaton $CarC2InTargetSeg$ for the traffic in the target segment.



(d) Automaton $CarC1InKeySegs$ for the traffic in the interference segment.

FIGURE 11. Modelling of decision-making in multi-lane roundabout scenario in UPPAAL SMC. (a) Automaton *Period* for period control. (b) Automaton *CarSelf* for the host vehicle with rational estimation. (c) Automaton *CarC2InTargetSeg* for the traffic in the target segment. (d) Automaton *CarC1InKeySegs* for the traffic in the interference segment.

scene structures. The existence of the interference segment and the target segment refers to the observation of *Map* in every control period. And the presence of vehicles on the target segment and the interference segment is controlled by the random function and observed in *TS* in every control period. The synchronized network of the four automata via broadcast channels *time* and *reset* are as follows and shown in Figure 11:

$$CarSelf||CarC2InTargetSeg||CarC1InKeySegs||Period$$

1) *PERIOD*

Figure 11(a) is the automaton *Period* for a period controller. Through the broadcast channel *time*, all four automata are synchronized, and this signal is sent when it comes to a period. One period equals to one cycle *clk* in this case. *initiate*() initializes the traffic data at first, and *uptSelf*() ensures that *self* always has an acceleration to start. The period controller is spawned according to an exponential distribution with rate 2 by the SHA, and the function *initiate*() is executed only once during the driving task.

2) CarSelf

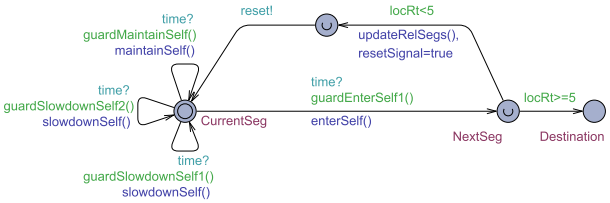
As Figure 11(b) shows, *CarSelf* is the automaton for the host vehicle *self* with rational estimation. Rational estimation is based on the calculation of running parameters of the interference vehicles as described in Section IV. The driving decision of *self* depends on the satisfaction of the guard conditions of their corresponding decisions as written in the tool below.

$$
\begin{aligned}
&bool\ guardSlowdownSelf2()\\
&\{\ \ return\ calMovDstSelf() >= segLen\\
&\&\& \ position[c2]! = rt[locRt + 1]\\
&\&\& \ relSegsLen >= 2\\
&\&\& \ position\_c1[car2]! = rt[locRt + 1]\\
&\&\& \ position\_c1[car1] == relSegs[1]\\
&\&\& \ estimate1() >= 30 \ \&\& \ estimate2() >= 30;\ \}
\end{aligned}
$$

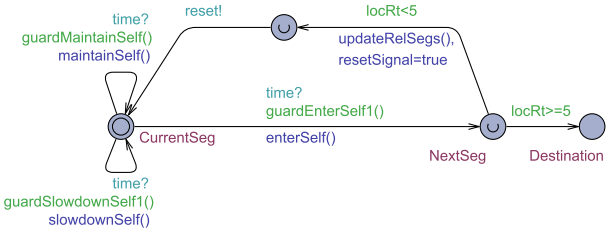
The guard conditions of *self* concern the observed traffic data and estimation of driving decisions of the interference vehicles. $guardSlowdownSelf2()$ is the boolean function of the guard condition g on the transition $CurrentSeg \xrightarrow{g,slowdownSelf(),j_{self},1} CurrentSeg$ with all weights to slow down where j_{self} indicates the transformation of continuous states. In g, as mentioned in the previous section, we focus on the continuous state in $\mathcal{Q}_{sub}$ where the target segment $rt[locRt + 1]$ is not occupied and at least one interference vehicle $car1$ exists checked in $position[c2]! = rt[locRt + 1]$, $position\_c1[car2]! = rt[locRt + 1]$ and $position\_c1[car1] == relSegs[1]$; $relSegsLen >= 2$ shows the discrete variables in Var_{sub} that this is a junction structure with at least one interference segment; $calMovDstSelf() >= segLen$ indicates that *self* is able to move out the current segment in the next period due to the estimation process of *Prob*, at the same time $estimate1() >= 30$ and $estimate2() >= 30$ shows the great probabilities for moving towards the target segment in the next period which is higher than the threshold for both interference vehicles $car1$ and $car2$ if exist. Definitely, *self* will slowdown if g is satisfied.

Figure 12(a) shows the automaton for *self* with conservative estimation. Conservative estimation means immediate slowing down whenever there exist interference vehicles. Figure 12(b) shows the automaton for *self* with aggressive estimation. Aggressive estimation means immediate entering whenever there exist interference vehicles. Apparently, the estimation style is not limited to rational, conservative, and aggressive style. We propose the three typical styles in this paper for comparison as shown in the Table 2. The host vehicle has different decisions with various estimation style when there are no vehicles in the target segment and the interference segment is occupied at the same time.

The planned route *rt* of *self* is divided into intervals between adjacent segments where *CurrentSeg* is the starting location while *NextSeg* is the end location in each interval. Synchronization signal for updating new observation of the environment is sent through broadcast channel *reset*. After reset, traffic data in the target segment in



(a) Automaton $CarSelf$ for the host vehicle with conservative estimation.



(b) Automaton $CarSelf$ for the host vehicle with aggressive estimation.

FIGURE 12. Automaton *CarSelf* with various styles of estimations. (a) Automaton *CarSelf* for the host vehicle with conservative estimation. (b) Automaton *CarSelf* for the host vehicle with aggressive estimation.

TABLE 2. Comparison of estimation styles.

Estimation style /Traffic condition	Has vehicle in the interference segment
conservative	slow down (stop)
rational	slow down/stay/enter based on calculation
aggressive	enter

CarC2InTargetSeg and traffic data in the interference segment in *CarC1InKeySegs* is updated based on the current location of *self* and the scene structure. According to the guards with the estimation of traffic in the interference segments and observation, *self* can keep moving in the original segment (*maintainSelf*()), slow down (*slowdownSelf*()), and enter the target segment (*enterSelf*()) until *self* reaches the location *Destination* in F.

3) CarC2InTargetSeg

Figure 11(c) is the automaton *CarC2InTargetSeg* for traffic in the target segment of *self*, and the vehicle originally in this segment is $c2$ by a random function. For the target segment of *self*, we check whether $c2$ is already in this segment. Once existed, it will leave in random periods by a random function. If the target segment is vacant, the target segment keeps unoccupied until the interference vehicle or the host vehicle reaches the segment.

4) CarC1InKeySegs

Figure 11(d) is the automaton *CarC1InKeySegs* for traffic in the interference segment of *self*. At first, search the *Map* and *rt* to find the existence of the interference segment (when the topology infers a junction structure) in location *NoKeySeg*,

and random function is utilized to randomize the existence of interference vehicle $c1$ if the interference segment exists in location *KeySeg*. The parameter *id* is introduced to identify the instance of $c1$ in the interference segments in location $KeySegC1$. The identifiers for the interference vehicles are $car1$ and $car2$ since there are at most two interference segments in the junctions in the multi-lane roundabout, i.e. *self* is in B, the target segment is K, $car1$ is in C, and $car2$ is in J.

Considering the assumption that $c1$ obeys the traffic rules, for instance, $c1$ may *leave*, *maintain*, or *forward* if the guard $hasCar() == false$ and $hasLeavePos() == true$ are satisfied. For the transition $KeySegC1 \xrightarrow{g_{c1},a_{c1},j_{c1},ProbForward\_C1} ToTgtSeg1$ where g_{c1} is its guard condition as written in the modelling language in the tool below.

$$\begin{aligned} g_{c1} &= hasCar() == false \\ &\&\& \ hasLeavePos() == true \\ &\&\& \ resetSignal == false \end{aligned}$$

Except for listening to the *time* signal, in g_{c1}, the target segment is not occupied as described in $hasCar() == false$, and $hasLeavePos() == true$ means $c1$ has another segment for leaving except the target segment in the observation of $c1$. $resetSignal == false$ means there is no signal on the broadcast channel *reset* which activates the reset. And with probability $ProbForward\_C1$ to move to the target segment, $c1$ completes the actions in a_{c1} which update the new probability of moving forward in $uptProb1()$ and calculates the new covering distance in $forwardC1()$ according to the **intention** of moving forward. $ProbLeave\_C1 + ProbForward\_C1 + ProbMaintain\_C1 = 1$ and each of them is the weight value of the **intention** of the interference vehicle. The successor locations are in accordance with the intentions when the corresponding guard conditions are satisfied, otherwise, transmit to the unexpected outcomes. These values are refreshed by one of the operations in $uptProb1$, $uptProb2$, and $uptProb3$ to simulate the stochastic probability values.

The location marked with **U** means an urgent state does not consume time. For location $ToTgtSeg1$, it indicates the moving forward intention of interference vehicle $c1$ and does not cost time. If $c1$ is able to move out of the current segment, the successor state will be $TgtSeg1$. The traffic data in $CarC1InKeySegs$ is updated when receiving the signal from the channel *reset*.

All the model files are open to the public in the case study repository on: **https://github.com/JoyaXu?tab=repositories**.

B. QUALITATIVE AND QUANTITATIVE VERIFICATION BY STATISTICAL MODEL CHECKING

The safety of the driving decisions upon this model can be assessed through the verification of safety properties. Qualitative and quantitative properties can be verified in the built-in verifier in UPPAAL SMC through statistical model checking.

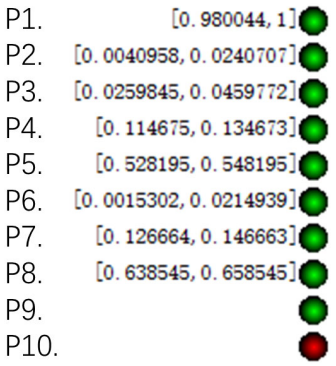


FIGURE 13. Verification results for safety properties with rational estimation.

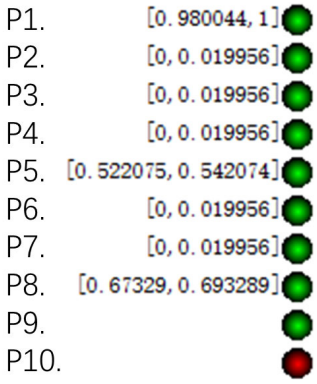


FIGURE 14. Verification results for safety properties with conservative estimation.

Considering safety in the driving task according to rt, the following requirements are essential and should be verified. One is that *self* should reach the destination in rt; the other is that the driving decisions of *self* ensure to maximize the probability of safety during the driving task and avoids collision with the other vehicles. Since safety is considered between the host vehicle and its environment, the dangerous situation between the interference vehicles is not studied in this paper.

Several fundamental qualitative and quantitative properties verified by SMC demonstrate the relative safety via the expression of probability value. The list of the safety properties is in Table 3 below. Figure 13, 14, and 15 shows the verification results of qualitative and quantitative properties with rational estimation, conservative estimation, and aggression estimation respectively. Here, all the properties are verified within 100 in time. The red spot indicates the failure of satisfaction of the property while the green spot is a success. And we demonstrate the typical ones as follows.

P1 verifies the probability that *self* reaches the destination according to rt, which is the basic requirement of the driving task. The probability value of the verification result ensures the high probability of arriving at the destination within the time bound in three versions of the model.

P2 indicates the tiny possibility that *self* and the interference vehicle $car2$ collide in the target segment. When the

TABLE 3. List of safety properties verified in multi-lane roundabout.

No.	Properties
P1	Pr[<=100](<> CarSelf.Destination)
P2	Pr[<=100](<> CarSelf.NextSeg && CarC1InKeySegs(2).TgtSeg2)
P3	Pr[<=100](<> position[self]==position_c1[car2])
P4	Pr[<=100](<> position[self]==B && position_c1[car1]==B)
P5	Pr[<=100](<> position[self]==K && position_c1[car1]==K)
P6	Pr[<=100](<> position[self]==D && position_c1[car1]==D)
P7	Pr[<=100](<> position[self]==E && position_c1[car1]==E)
P8	Pr[<=100](<> position_c1[car1] ==K && position_c1[car2]==K)
P9	Pr[<=100](<> CarC1InKeySegs(1).TgtSeg1) >= Pr[<=100](<> CarC1InKeySegs(1).Unclear1)
P10	Pr[<=100](<> CarSelf.NextSeg && CarC1InKeySegs(1).TgtSeg1) >= 0.18

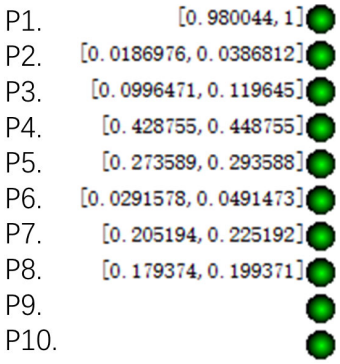


FIGURE 15. Verification results for safety properties with aggressive estimation.

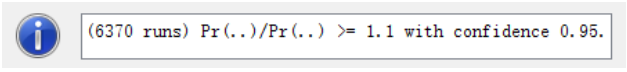


FIGURE 16. Verification result for probability comparison.

estimation is conservative, no collision will happen due to the verification results.

P5 verifies the probability that *self* and the interference vehicle *car*1 collide in the target segment K. Obviously, the great value of probability means high possibility of collision. Then we will expose the reason why it is likely to collide in segment K compared to B, D, E which is also the target segment of the interval in scene structures. Then we analyse the responsibility of collision and relationship of collision and scene structure in Section VII-C later.

P8 indicates the high probability of collision between the two inference vehicles. It is easy to conclude that the estimation information is a vital part of keeping safety since the high probability of collision results from the lack of estimation of driving behaviours between the interference vehicles.

P9 verifies the quantitative comparison as Figure 16 shows in three versions of the model. The probability of *car*1 moving to the target segment is larger than the probability of *car*1 colliding with *self* caused by a sudden behaviour change which is not expected according to its original intention.

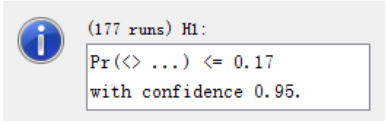


FIGURE 17. Verification result for probability hypothesis with rational and conservative estimation.

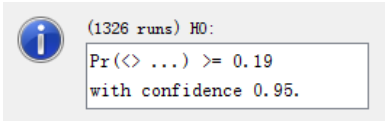


FIGURE 18. Verification result for probability hypothesis with aggressive estimation.

And this indicates that unexpected event rarely happens based on the estimation.

P10 is to verify whether the probability is larger than 0.18 that *self* reaches the target segment while *car*1 happens to move to the same segment in the same period. The red spots in Figure 13 and 14 shows that the probability is lower than 0.18 which indicates a safer situation. As shown in Figure 17, the property is not satisfied in the rational and conservative estimation since it meets the request of safety that the probability of collision is smaller than 0.17. That means the rational and conservative really contribute to the safety of the driving decisions. Figure 18 shows that this property is satisfied in the model with aggressive estimation. This indicates that aggressive estimation is more likely to lead to a dangerous state if compared to a determined safety threshold of 0.18.

C. SAFETY RESPONSIBILITY IN THE MULTI-LANE ROUNDABOUT

This part reveals the practical application of safety assessment scheme of driving decisions based on the scene structures, such as the discussion on the responsibility of collision. From the high probability of collision in segment K verified in **P5: Pr[<=100](<> position[self]==K && position_c1[car1]==K)**, we discover that *self* and the interference vehicle *car*1 will collide in K with high probability while the collision probability is low in B, D, and E. And it

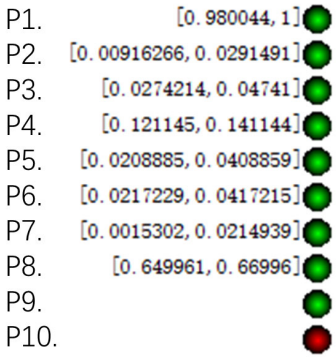


FIGURE 19. Verification results for safety properties with advanced rational estimation.

is definitely related to the scene structure. According to the *Map* in Figure 4, B, K, D, and E are the target segments in the junction structures according to the occupied segments of *self*. The collision is about to happen in K in two cases.

1) When *self* is in K, C is the interference segment that $car1$ is in, D is the target segment, and K is the segment for $car1$ to leave. In this scene structure with one interference segment, the segment for leaving and the position of *self* is the same segment.
2) When *self* is in B, C is the interference segment that $car1$ is in, J is the interference segment that $car2$ is in, K is the target segment, and D is the segment for $car1$ to leave. And this is the scene structure with two interference segments.

We mainly focus on the driving decisions of *self*, but not control the decision-making process of the interference vehicle, e.g. $car1$ enters the other segment to leave. Therefore, we doubt that the high probability of collision is caused by a rear-end collision by $car1$ due to the lack of observation and its stochastic behaviour in the first case described above. To confirm the speculation, we add the judging process for $car1$ when it enters the segment for leaving. The collision rate sharply decreases in the model with rational, conservative, and aggressive estimation as shown in Figure 19, 20, and 21. It is believed that the characterized scenarios and verification results for quantitative safety properties can be used to help to analyse the cause of the collision.

And for the property **P10**, it still remains the same verification result as shown in Figure 13, 14 and 15, and aggressive estimation leads to a higher collision rate compared to the rational and conservative estimation.

D. TRADE-OFF BETWEEN SAFETY AND EFFICIENCY IN FUNDAMENTAL SCENE STRUCTURES

In the industrial field, the requirement of driving task is not limited to safety, but also efficiency. As the verification above is based on the 100 time units, it is more practical to study the balance between safety and efficiency in three scene structures if less time is allowed.

According to the scene structures in Figure 5, we verify the properties in 15 and 20 time units. Through these experiments, we come to the conclusion regarding both safety

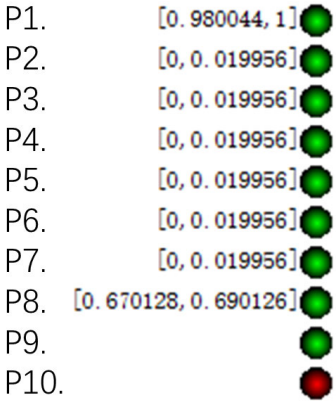


FIGURE 20. Verification results for safety properties with advanced conservative estimation.

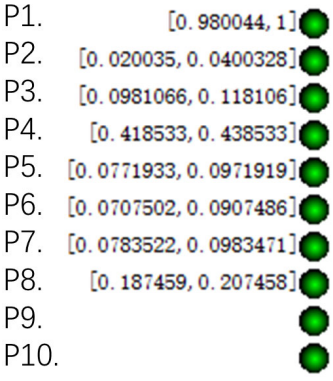


FIGURE 21. Verification results for safety properties with advanced aggressive estimation.

and efficiency in the decision-making process. For instance, the adoption of combined estimation styles achieves better performance in safety and efficiency in the composite driving scenario composed of fundamental scene structures. The concerned properties are in two types.

Properties denoting completion of the driving task:

- **Pr[bound](<> CarSelf.Destination)**

Safety properties:

- **Pr[<=15](<> position[self]==B && position[c2]==B)**
- **Pr[<=15](<> position[self]==B && position_c1[car1]==B)**
- **Pr[bound](<> position[self]==B && (position_c1[car1]==B || position_c1[car2]==B))**

1. For scene structures without interference segments like single road and fork, the estimation process can be removed from the abstract model as mentioned in Section IV to save the computation time as it is safe enough as shown in Figure 22.

2. For the junction with one interference segment, a rational estimation is recommended to keep safety in the driving task. As shown in Figure 23, in the given time, the conservative estimation will not cause a collision but has a lower probability in the completion of the task. At the same time, the collision rate is too high using the aggressive estimation.

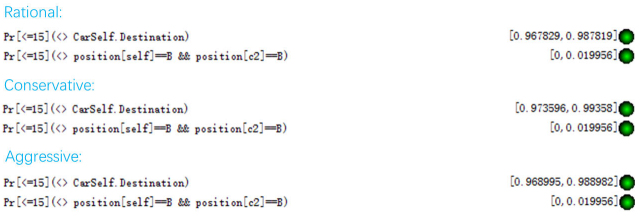


FIGURE 22. Verification results for properties in single road structure and fork structure in 15 time units.

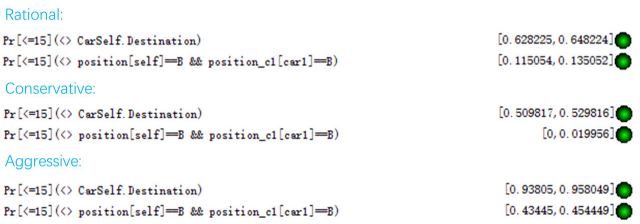


FIGURE 23. Verification results for properties in junction structure with single interference segment in 15 time units.

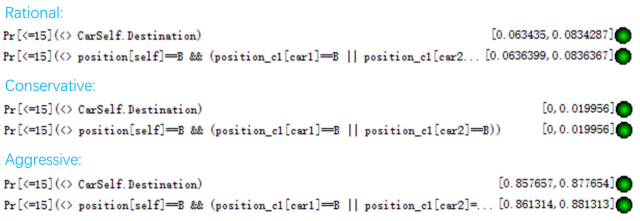


FIGURE 24. Verification results for properties in junction structure with multiple interference segments in 15 time units.

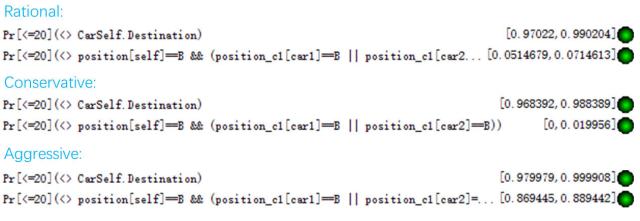


FIGURE 25. Verification results for properties in junction structure with multiple interference segments in 20 time units.

If given the safety threshold and arrival rate according to the industrial standard, we can adjust the time bound specified in the property to find out the acceptable time interval in which safety is guaranteed through the quantitative verification.

3. For the junction with multiple interference segments, analysing the rational part in Figure 24, the arrival rate is obviously decreased as it costs more time on estimation compared with the junction with single interference segment in Figure 23 in the same given time.

4. For the junction with multiple interference segments, the arrival rate increases if given more time as shown in Figure 24 and Figure 25.

Given the driving decisions under our abstract model of decision-making process in the composite driving scenario and our verification method, we can infer how the element in our abstract model influences safety such as estimation style or the observation. For each scene structure, we can extract the necessary structure and the corresponding style in the abstract model to make the decision with respect to the balance between safety and efficiency. According to the functional safety requirement of vehicles in the industrial standard like ISO 26262, our verification method provides the solution for safety assessment based on the safety threshold.

VIII. RELATED WORK

In the previous studies, from different perspectives, there exist numerous work on scenario-based spatial modelling and safety assessment methods of driving behaviours in automated driving.

A. MODELLING DRIVING SCENARIOS

The driving performance of a vehicle is studied regarding its driving scenarios, and the existing approaches of modelling driving scenario are under a hierarchical structure or by features classification based on domain knowledge.

The researchers from the Institute for Automotive Engineering at RWTH Aachen University adopt the logical scenario as a six-layer model of different properties for scenario-based testing. Sensor data including road geometry, moving objects, environmental conditions etc. is segmented into each layer of the model [20]. Kettani et al. propose notions of the spatial conceptual map (data structure of mental images) and object's influence areas (neighbourhood space around spatial objects). And the two notions enable the formal definition of the properties of neighbourhood, orientation and distance in a qualitative way [21]. Bagschik et al. put forward a knowledge-based scene creation applied in traffic scenes for automated vehicles by the approach of ontology [22]. Uwe et al. introduce the Stop&Go system to cope with complex urban traffic scenario rather than highway traffic and extract spatial features based on object detection, tracking and recognition [23]. For driving tasks as lane change and overtaking on the highways, the Multi-lane Spatial Logic (MLSL) [24] is proposed for the specification, reasoning, and verification of spatial property based on the sequence model of lanes. Length measurement and dynamic modality are later introduced to refine MLSL as an extended EMLSL [25]. Based on the one-directional spatial model in MLSL, we expand it into a two-directional one which can specify and verify spatial properties in the crossroads [26]. Later, a generic topology of urban traffic networks is put forward to modify the abstract model of the typical crossroads [27]. Simulators such as CARLA [28] and PTV Vissim [29] provide maps generation, sensor data, environmental conditions considering the weather, driving behaviours etc. for scenario description.

In this paper, we link the spatial and temporal properties with the traffic data and characterize these data based on the domain knowledge in the automated driving.

B. AUTOMATED TOOLS FOR QUANTITATIVE VERIFICATION

For the formalized model of the stochastic complex system, automated model checking tools make the formal verification

applicable in the industry with higher safety. Especially the ones that can check both qualitative and quantitative properties.

PRISM is suitable for quantitative verification in the probabilistic systems [30], [31] and stochastic multi-player games. Chen et al. propose the temporal logic rPATL reasoning out the probability of an event's occurrence or the expected amount of accumulated cost/reward when a set of players achieving a goal [32]. They also study the strategy synthesis for stochastic two-player games where each property in the conjunction can be either an LTL formula or a reward function [33]. MoDeST toolset is ideal for modelling and verification of Stochastic Timed Automaton (STA) [34] and SHA [35]. UPPAAL SMC is the Statistical Model Checking (SMC) extension [19] of UPPAAL, which can achieve performance analysis by quantitative property verification. It is an integrated tool environment for modelling, validation, and verification of real-time systems modelled as networks of timed automata [36]. COSMOS, a statistical model checker for the Hybrid Automata Stochastic Logic(HASL), it takes a Generalized Stochastic Petri Net, an LHA and an expression Z representing the quantity to be estimated as the input [37].

We choose UPPAAL SMC in this paper due to its strength on real-time behaviours and its performance analysis that can reveal the relative safety of the host vehicle with high autonomy.

C. SAFETY ASSESSMENT APPROACHES OF AUTOMATED DRIVING

For standardization of safety assurance in automated driving, Responsibility-Sensitive Safety (RSS) is proposed as a white-box, interpretable, mathematical model for safety assurance. It reveals the safety standard that the autonomous car should never take the initiative action leading to crash according to the current traffic condition [38]. For classic safety assessment, safety descriptors use only the time to collision and the vehicle distance gap. The Dwell Time descriptor is extended to assess safety from both time and distance criteria on driver safety under mixed traffic styles [39]. Based on the assessment by virtual testing and hardware in-the-loop testing, Gelder et al. present a data-driven method to generate test cases from real-life driving data, and compute the probability of the occurrence of unsafe situations in real scenarios [40]. ESACS platform integrates the system design and the system safety assessment processes where formal notations are the common and shared language. An application in the embedded controllers of Secondary Power System for the Eurofighter Typhoon aircraft is demonstrated [41]. Besides the testing method, model checking method is also studied for the safety assessment. With respect to both continuous and discrete aspect of the self-driving process, to verify the driving behaviours of the autonomous car in platoons on highways, how combined verification approaches work is presented based on the driving data from TORC [42]. Wang et al. from the National Highway Traffic Safety Administration (NHTSA) present a technique used by BMW for the safety assessment of highly automated driving functions. However, an international consensus on safety standard and methodological issues of the safety assessment of automated driving is still lacking [43]. To assess the safety of driving decisions, it is believed that an integrated theoretical framework by the formal model should combine the probabilistic perception and deterministic control to handle driving behaviours better in the uncertain urban environment [44]. For the black box problem of the fully data-driven decision-making approaches, Sifakis proposes the hybrid architecture of formal model as the combination of data-driven modules and model-driven modules for automated driving [45]. Accordingly, using formal methods is convinced as a promising way to improve safety in the safety-critical systems of industrial applications [46].

IX. CONCLUSION

This paper studies a safety assessment scheme of decision-making in automated driving, including the scenario-based formal modelling and verification approach. We construct the abstract model to describe the essential spatio-temporal features in the decision-making process. According to the observation of static road geometry and time-dependent dynamic traffic in the abstract model, the general formal description of the scenario is defined. Moreover, we characterize three fundamental scene structures, which indicates the basic unit scenarios that can connect and compose a composite scenario together. Based on the classified scene structures, the scenario-based estimation of stochastic driving decisions of other vehicles in the probabilistic representation is calculated by the observation in the abstract model. Due to the features in the abstract model, we can specify and reason the spatial properties and guard conditions considering both the observed data and estimation. The corresponding model verification method is suggested based on the successive verification in the connected fundamental scene structures along the driving route. Then any scenario that is a composition of fundamental scene structures does not need a new model, because the model and verification method of the fundamental scene structure is fixed. Mapping from the abstract model to the network of SHAs, both qualitative and quantitative properties can be verified in the automated verification tool UPPAAL SMC, and find a way to apply the approach in the industry. Based on the verification result, it is applicable to analyse the trade-off between various properties in the case of the system requirements and discuss the responsibility of danger under the scene structure.

In this paper and our previous work [17], [18], [26], the observed local traffic of the host vehicle is the basis of spatio-temporal logic and scenario-based safety assessment. While the shared traffic data by communication in the collaboration with the other vehicles should be concerned in both theoretical and experimental research. In the future, we shall consider the integration of the individual local spatial data and the mapping from local knowledge to the global consistent knowledge through spatial reasoning.

REFERENCES

[1] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, and M. Gittleman, "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 425–466, 2008.

[2] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Baden-Baden, Germany, Jun. 2011, pp. 163–168. doi: 10.1109/IVS.2011.5940562.

[3] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. F. R. Jesus, R. F. Berriel, T. M. Paixão, F. Mutz, T. Oliveira-Santos, and A. F. De Souza, "Self-driving cars: A survey," 2019, *arXiv:1901.04407*. [Online]. Available: https://arxiv.org/abs/1901.04407

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 3354–3361. doi: 10.1109/CVPR.2012.6248074.

[5] *Waymo*. Accessed: Dec. 20, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Waymo

[6] J. A. Michon, "A critical view of driver behavior models: What do we know, what should we do?" in *Human Behavior and Traffic Safety*, L. Evans and R. C. Schwing, Eds. Boston, MA, USA: Springer, 1985, pp. 485–524. doi: 10.1007/978-1-4613-2173-6_19.

[7] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, SAE, Standard J3016_201806, Jun. 2018.

[8] D. Dolgov and S. Thrun, "Autonomous driving in semi-structured environments: Mapping and planning," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Kobe, Japan, May 2009, pp. 3407–3414. doi: 10.1109/ROBOT.2009.5152682.

[9] J. Choi, J. Lee, D. Kim, G. Soprani, P. Cerri, A. Broggi, and K. Yi, "Environment-detection-and-mapping algorithm for autonomous driving in rural or off-road environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 974–982, Jun. 2012. doi: 10.1109/TITS.2011.2179802.

[10] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Qingdao, China, Oct. 2014, pp. 392–399. doi: 10.1109/ITSC.2014.6957722.

[11] G. Elsayed, S. Shankar, B. Cheung, N. Papernot, A. Kurakin, I. Goodfellow, and J. Sohl-Dickstein, "Adversarial examples that fool both computer vision and time-limited humans," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2018, pp. 3914–3924. [Online]. Available: http://papers.nips.cc/paper/7647-adversarial-examples-that-fool-both-computer-vision-and-time-limited-humans

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[13] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 299–310, Jun. 2009. doi: 10.1109/TITS.2009.2018966.

[14] C. Berger and B. Rumpe, "Autonomous driving-5 years after the urban challenge: The anticipatory vehicle as a cyber-physical system," 2014, *arXiv:1409.0413*. [Online]. Available: https://arxiv.org/abs/1409.0413

[15] *Roundabouts: An Informational Guide*, U.S. Dept. Transp., Federal Highway Admin., Washington, DC, USA, 2000.

[16] A. David, K. G. Larsen, A. Legay, and D. B. Poulsen, "Statistical model checking of dynamic networks of stochastic hybrid automata," in *Proc. ECEASST*, vol. 66, 2013, pp. 1–15. doi: 10.14279/tuj.eceasst.66.893.

[17] B. Xu and Q. Li, "A bounded multi-dimensional modal logic for autonomous cars based on local traffic and estimation," in *Proc. 11th Int. Symp. Theor. Aspects Softw. Eng. (TASE)*, Sophia Antipolis, France, Sep. 2017, pp. 1–8. doi: 10.1109/TASE.2017.8285637.

[18] B. Xu, Q. Li, T. Guo, Y. Ao, and D. Du, "A quantitative safety verification approach for the decision-making process of autonomous driving," in *Proc. 13th Int. Symp. Theor. Aspects Softw. Eng. (TASE)*, Guilin, China, Jul./Aug 2019.

[19] A. David, K. G. Larsen, A. Legay, M. Mikučionis, and D. B. Poulsen, "Uppaal SMC tutorial," *Int. J. Softw. Tools Technol. Transf.*, vol. 17, no. 4, pp. 397–415, 2015. doi: 10.1007/s10009-014-0361-y.

[20] J. Sauerbier, J. Bock, H. Weber, and L. Eckstein, "Definition of scenarios for safety validation of automated driving functions," *ATZ Worldwide*, vol. 121, no. 1, pp. 42–45, 2019.

[21] D. Kettani and B. Moulin, "A spatial model based on the notions of spatial conceptual map and of object's influence areas," in *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science*. Berlin, Germany: Springer, Aug. 1999, pp. 401–416. doi: 10.1007/3-540-48384-5_26.

[22] G. Bagschik, T. Menzel, and M. Maurer, "Ontology based scene creation for the development of automated vehicles," 2017, *arXiv:1704.01006*. [Online]. Available: https://arxiv.org/abs/1704.01006

[23] U. Franke, D. Gavrila, S. Görzig, F. Lindner, F. Puetzold, and C. Wöhler, "Autonomous driving goes downtown," *IEEE Intell. Syst.*, vol. 13, no. 6, pp. 40–48, Nov./Dec. 1998. doi: 10.1109/5254.736001.

[24] M. Hilscher, S. Linker, E.-R. Olderog, and A. P. Ravn, "An abstract model for proving safety of multi-lane traffic manoeuvres," in *Proc. 13th Int. Conf. Formal Eng. Methods (ICFEM)*, Durham, U.K., Oct. 2011, pp. 404–419. doi: 10.1007/978-3-642-24559-6_28.

[25] S. Linker and M. Hilscher, "Proof theory of a multi-lane spatial logic," in *Proc. 10th Int. Colloq. Theor. Aspects Comput.*, Shanghai, China, Sep. 2013, pp. 231–248. doi: 10.1007/978-3-642-39718-9_14.

[26] B. Xu and Q. Li, "A spatial logic for modeling and verification of collision-free control of vehicles," in *Proc. 21st Int. Conf. Eng. Complex Comput. Syst. (ICECCS)*, Dubai, United Arab Emirates, Nov. 2016, pp. 33–42. doi: 10.1109/ICECCS.2016.014.

[27] M. Hilscher and M. Schwammberger, "An abstract model for proving safety of autonomous urban traffic," in *Proc. 13th Int. Colloq. Theor. Aspects Comput.*, Taipei, Taiwan, Oct. 2016, pp. 274–292. doi: 10.1007/978-3-319-46750-4_16.

[28] A. Dosovitskiy, G. Ros, F. Codevilla, A. López, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn. (CoRL)*, Mountain View, CA, USA, Nov. 2017, pp. 1–16. [Online]. Available: http://proceedings.mlr.press/v78/dosovitskiy17a.html

[29] M. Fellendorf and P. Vortisch, "Microscopic traffic flow simulator VISSIM," in *Fundamentals of Traffic Simulation*. New York, NY, USA: Springer, 2010, pp. 63–93.

[30] M. Z. Kwiatkowska, G. Norman, and D. Parker, "Quantitative analysis with the probabilistic model checker PRISM," *Elect. Notes Theor. Comput. Sci.*, vol. 153, no. 2, pp. 5–31, 2006. doi: 10.1016/j.entcs.2005.10.030.

[31] M. Z. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Proc. 23rd Int. Conf. Comput. Aided Verification (CAV)*, Snowbird, UT, USA, Jul. 2011, pp. 585–591. doi: 10.1007/978-3-642-22110-1_47.

[32] T. Chen, V. Forejt, M. Z. Kwiatkowska, D. Parker, and A. Simaitis, "Automatic verification of competitive stochastic systems," *Formal Methods Syst. Design*, vol. 43, no. 1, pp. 61–92, 2013. doi: 10.1007/s10703-013-0183-7.

[33] T. Chen, M. Kwiatkowska, A. Simaitis, and C. Wiltsche, "Synthesis for multi-objective stochastic games: An application to autonomous urban driving," in *Proc. 10th Int. Conf. Quant. Eval. Syst. (QEST)*, Buenos Aires, Argentina, Aug. 2013, pp. 322–337. doi: 10.1007/978-3-642-40196-1_28.

[34] H. C. Bohnenkamp, P. R. D'Argenio, H. Hermanns, and J.-P. Katoen, "MODEST: A compositional modeling formalism for hard and softly timed systems," *IEEE Trans. Softw. Eng.*, vol. 32, no. 10, pp. 812–830, Oct. 2006. doi: 10.1109/TSE.2006.104.

[35] E. M. Hahn, A. Hartmanns, H. Hermanns, and J.-P. Katoen, "A compositional modelling and analysis framework for stochastic hybrid systems," *Formal Methods Syst. Design*, vol. 43, no. 2, pp. 191–232, 2013. doi: 10.1007/s10703-012-0167-z.

[36] G. Behrmann, A. David, and K. G. Larsen, "A tutorial on uppaal," in *Proc. Formal Methods Design Real-Time Syst.*, Bertinoro, Italy, Sep. 2004, pp. 200–236. doi: 10.1007/978-3-540-30080-9_7.

[37] P. Ballarini, H. Djafri, M. Duflot, S. Haddad, and N. Pekergin, "COSMOS: A statistical model checker for the hybrid automata stochastic logic," in *Proc. 8th Int. Conf. Quant. Eval. Syst. (QEST)*, Aachen, Germany, Sep. 2011, pp. 143–144. doi: 10.1109/QEST.2011.24.

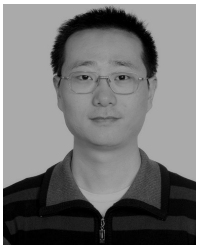
[38] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," 2017, *arXiv:1708.06374*. [Online]. Available: https://arxiv.org/abs/1708.06374

[39] O. Derbel, B. Mourllion, and M. Basset, "Extended safety descriptor measurements for relative safety assessment in mixed road traffic," in *Proc. Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 752–757.

[40] E. de Gelder and J.-P. Paardekooper, "Assessment of automated driving systems using real-life scenarios," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Los Angeles, CA, USA, Jun. 2017, pp. 589–594. doi: 10.1109/IVS.2017.7995782.

[41] M. Bozzano, A. Cavallo, M. Cifaldi, L. Valacca, and A. Villafiorita, "Improving safety assessment of complex systems: An industrial case study," in *Proc. Int. Symp. Formal Methods Eur.*, Pisa, Italy, Sep. 2003, pp. 208–222. doi: 10.1007/978-3-540-45236-2_13.
[42] M. Kamali, L. A. Dennis, O. McAree, M. Fisher, and S. M. Veres, "Formal verification of autonomous vehicle platooning," *Sci. Comput. Program.*, vol. 148, pp. 88–106, Nov. 2017. doi: 10.1016/j.scico.2017.05.006.
[43] L. Wang, F. Fahrenkrog, T. Vogt, O. Jung, and R. Kates, "Prospective safety assessment of highly automated driving functions using stochastic traffic simulation," in *Proc. 25th Int. Tech. Conf. Enhanced Saf. Vehicles (ESV)*, 2017, pp. 1–13.
[44] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: Approaches, lessons and challenges," *Philos. Trans. Roy. Soc. London A, Math. Phys. Sci.*, vol. 368, no. 1928, pp. 4649–4672, 2010.
[45] J. Sifakis, "Autonomous systems—An architectural characterization," 2018, *arXiv:1811.10277*. [Online]. Available: https://arxiv.org/abs/1811.10277
[46] J. Bowen and V. Stavridou, "Safety-critical systems, formal methods and standards," *Softw. Eng. J.*, vol. 8, no. 4, pp. 189–209, Jul. 1993. doi: 10.1049/sej.1993.0025.

BINGQING XU received the B.S. degree in software engineering from East China Normal University, Shanghai, China, in 2013, where she is currently pursuing the Ph.D. degree with the School of Software Engineering. Her research interest includes modeling and verification of cyber-physical systems.



QIN LI received the Ph.D. degree from East China Normal University, Shanghai, China, in 2011, where he is currently an Associate Professor with the School of Software Engineering. His research interests include formal modeling and verification of cyber-physical systems, and trustworthy artificial intelligence systems.

TONG GUO received the B.S. degree from the Shandong University of Science and Technology, Qingdao, Shandong, China, in 2018. She is currently pursuing the M.S. degree in software engineering with East China Normal University, Shanghai. Her research interest includes modeling and verification of cyber-physical systems.

DEHUI DU was born in Henan, China, in 1979. She received the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 2007. She is currently a Research with East China Normal University, as an Associate Professor. Her research interests include the formal method, model-driven development, and verification of cyber-physical systems.