Modeling and Verification of Autonomous Driving Systems under Stochastic Spatio-Temporal Constraints

Mengyuan Wang[1], Tengfei Li[2], Jing Liu[1,*], Hui Dou[4], Hongtao Chen[4], John Zhang[4,*], Lipeng Zhang[3,*]
[1]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
[2]Casco signal Ltd., Shanghai, China
[3]ECNU-CASCO Trustworthy Railway Joint Lab, East China Normal University, Shanghai, China
[4]Huawei Technology, Shanghai, China
*Corresponding authors: Jing Liu (Email: jliu@sei.ecnu.edu.cn),
John Zhang(Email: yuhongzhang@xidian.edu.cn), Lipeng Zhang(Email: 52194501021@stu.ecnu.edu.cn)

***Abstract*—The decision-making process in autonomous driving systems encounters large uncertainties with environmental changes and needs to face the complex spatio-temporal evolution of multiple objectives. Formal analysis and verification are crucial to establishing reliable and safe standards. In this paper, we propose an extension of the clock constraint language CCSL to construct spatio-temporal constraint and autonomous driving safety specifications, leveraging various autonomous driving scenarios. Additionally, we introduce probabilistic spatio-temporal events and devise extensions for driving specifications that incorporate stochasticity. This specification is converted to the UPPAAL-SMC model for facilitating formal modeling and verification. Specific schemes and verification are given in conjunction with a typical autonomous driving scenario.**

***Keywords*—CCSL, uncertainty modeling, autonomous driving control, statistical model checking**

I. Introduction

Cyber-Physical Systems(CPS) [1] are multi-dimensional and intricate systems that integrate the physical, network, and computing environments. As previously discussed, CPS is a combination of cyber and physical elements, which gives rise to various types of uncertainties. Autonomous Driving Systems(ADS) exemplify a typical instance of CPS. The uncertainties of human behaviors and the physical environment usually result in unavoidable stochastic behaviors of ADS.

Compared with manual driving, the accident rate of autonomous driving is lower. However, due to the complexity of the driving environment, improving the safety of the autonomous driving system is still a hot spot and a difficult area of research. Driving decisions based on rules or based on data in different scenarios should have different response strategies as an intelligent body. Unlike human drivers, autonomous vehicles must timely and accurately respond to the complex and dynamic environment, adhering to spatial and clock-related restrictions. In other words, the trigger conditions in spatio-temporal systems are constrained not only by strictly temporal limitations and physical time intervals but also by logical and spatial relationships.

DOI reference number: 10.18293/SEKE2023-221

MARTE [2] extends UML by providing comprehensive support for dense and discrete time, chronometric and logical time, as well as simple and multiple time references. As the companion language of MARTE, CCSL [3] enables the specification of clock mutual dependence. Thus, as a high-level formalism in the Formal Specification Level, CCSL accurately models the causal and temporal timing behaviors of real-time embedded systems.

In this paper, we focus on the spatio-temporal probability constraint logic and use this spatio-temporal event containing uncertainty for driving decisions.

This paper makes the following contributions:

- We propose a stochastic extension of CCSL with probability clock and stochastic delay to support modeling uncertainty-aware timing behaviors.
- We represent spatial events as multiform logical clocks of stochastic CCSL, called Stochastic stCCSL.
- We propose an encoding in SHA of the semantics of Stochastic stCCSL. Then we can check the safety specifications with the statistical model checker UPPAAL-SMC.

The rest of this paper is structured as follows: In Section II, we introduce preliminaries. Section III presents our extension of CCSL and proposes some transformation rules to encode Stochastic stCCSL into SHA. Section IV presents our case study and evaluation results of various solutions with UPPAAL-SMC. In Section V, we summarize related work and conclude in Section VI.

II. Preliminaries

A. Spatial Relationship

Based on the topological space, RCC8 spatial relations are proposed. The main binary relations are shown in Fig. 1:

- $DC(AB)$: A and B are independent of each other, i.e., none of the points in A are in B, and vice versa.
- $EC(AB)$: A and B boundaries are tangent to each other, i.e., A and B just intersect.

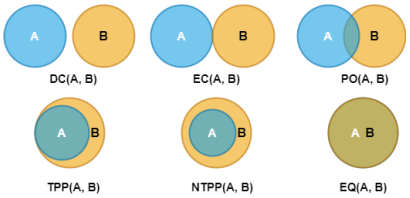


Fig. 1: $RCC8$ Spatial Relations

- $PO(AB)$: A and B partially overlap, i.e., some points in A are in B, and vice versa.
- $TPP(AB)$: A is contained in B, the points in A are in B, and the boundary of A is tangent to B.
- $NTPP(AB)$: A is contained in B, and the points in A are completely in B.
- $EQ(AB)$: All points in A are contained in B and vice versa.

B. Clock Constraint Specification Language(CCSL)

Definition 1 *(Logical Clock):* A logical clock c is defined as an infinite sequence of ticks: $(c_i)_{i=1}^{\infty}$.

Definition 2 *(Schedule):* A schedule is a function $\sigma : \mathbb{N} \to 2^C$, where C is a set of logical clocks. Given an execution step s $\in \mathbb{N}$, σ(s) denotes the set of clocks that tick at step s.

Definition 3 *(History):* Given a schedule $\sigma : \mathbb{N} \to 2^C$, a history of a schedule σ is a function $H_\sigma : C \times \mathbb{N}^+ \to \mathbb{N}$ such that for each clock $c \in C$ and natural number $n \in \mathbb{N}^+$:

$$H_\sigma(c,n) = \begin{cases} 0 \text{ if } n = 1 \\ H_\sigma(c, n-1) & \text{if } n > 1 \wedge c \notin \sigma(n-1) \\ H_\sigma(c, n-1) + 1 & \text{if } n > 1 \wedge c \in \sigma(n-1) \end{cases}$$

Intuitively, $H_\sigma(c, n)$ counts the number of c ticks before moment n.

Table I presents the syntax and semantics of CCSL operators. The semantics of CCSL is defined by the satisfaction of a schedule against corresponding constraints. Due to the page limit, we do not provide full details of the formal semantics of other CCSL constraints. Please refer to [3] for more details.

C. Stochastic Hybrid Automata(SHA)

Stochastic Hybrid Automata (SHA) [4] is described by a tuple $H = (L, l_0, C, Act, I, F, pE)$, where: 1) L is a finite set of locations, 2) l_0 is the initial location, 3) C is a finite set of clocks,Act is the set of actions, 4) $I : L \to Zones(C)$ assigns an invariant to each location, where $Zones(C)$ is the set of zone in C, 5) F is a time delay function for each location, 6) $pE \subseteq L \times Act \times Zones(C) \times prob \times 2^C \times L$ is a finite set of transactions with probability, where $prob \in [0, 1]$ is a rational number presenting the probability.

III. Stochastic Extension of Spatio-Temporal CCSL

A. Spatio-temporal constraints

Spatio-temporal constraints events are proposed based on RCC8 spatial relations and spatial logical S4u. The spatial relations are shown in Fig. 2:

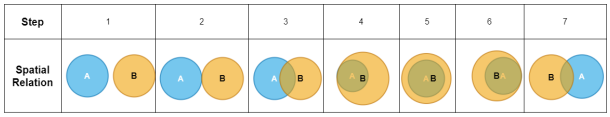


Fig. 2: Example for spatial relations evolving over time

Spatial events [5] are derived from the evolution of spatial relationships. CCSL combining time and space constraints introduce the definition of the spatial events, which are generated by the interaction between areas varying through time.

Definition 4 *(Spatial Event):* The syntax of spatial events is defined as $\mathcal{E}_{act}$, where $act = \{join(A,B), detach(A,B), include(A,B), exclude(A,B)\}$. Specific semantics of spatial events are as follows:

- $\mathcal{E}_{join(A,B)}$ is used to express the transition from relation $DC(A,B)$ to $EC(A,B) \vee PO(A,B) \vee TPP(A,B) \vee NTPP(A,B)$.
- $\mathcal{E}_{detach(A,B)}$ is used to express the transition from relation $EC(A,B) \vee PO(A,B) \vee TPP(A,B) \vee NTPP(A,B)$ to $DC(A,B)$.
- $\mathcal{E}_{include(A,B)}$ is used to express the transition from relation $DC(A,B) \vee EC(A,B) \vee PO(A,B) \vee TPP(A,B) \vee NTPP(A,B)$ to $TPP(A,B) \vee NTPP(A,B)$.
- $\mathcal{E}_{exclude(A,B)}$ is used to express the transition from relation $TPP(A,B) \vee NTPP(A,B)$ to $DC(A,B) \vee EC(A,B) \vee PO(A,B) \vee TPP(A,B) \vee NTPP(A,B)$.

The driving of autonomous vehicles involves spatio-temporal evolution, where their spatial position changes over time. Therefore, the driving protocol requires both logical time constraints to limit the system's response migration time and spatial event constraints. The CCSL provides a logical clock with strong expressive and reasoning abilities.

Fig. 3 shows where Ego, an autonomous vehicle, typically follows vehicle $Preceding$. However, there is a potential risk of rear-end collisions when Ego is driving at high speed. The autonomous vehicle in this scenario must determine the logical state of the spatio-temporal relationship between the two vehicles and modify its driving behavior accordingly.

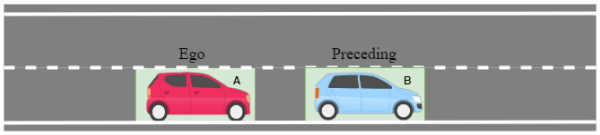


Fig. 3: A scenario in straight lane

The algorithm of generating Spatial Event $\mathcal{E}_{detach(A,B)}$ is shown in Algorithm 1.

In this function, $obstacles$ encompass either vehicles or other obstacles. If there is a vehicle in the same lane as the autonomous vehicle, the function calculates the braking distance ($brakingD$) and the distance that the front vehicle can cover within the autonomous vehicle's braking time ($obstacleD$). On the other hand, if there is an obstacle in the same lane, the function only considers the autonomous vehicle's speed for

TABLE I: The syntax and semantics of CCSL

Name	*Constraint*	*Semantics*
Precedence	$a[d] \prec b$	$\forall n \in \mathbb{N}^+. (H_\sigma(b,n) - H_\sigma(a,n) = d) \Rightarrow (b \notin \sigma(n))$
Causality	$a \preceq b$	$\forall n \in \mathbb{N}^+ H_\sigma(a,n) \geq H_\sigma(b,n)$
Subclock	$a \subseteq b$	$\forall n \in \mathbb{N}^+.(a \in \sigma(n)) \Rightarrow (b \in \sigma(n))$
Exclusion	$a \# b$	$\forall n \in \mathbb{N}^+.(a \notin \sigma(n)) \vee (b \notin \sigma(n))$
Union	$a \triangleq b + c$	$\forall n \in \mathbb{N}^+.(a \in \sigma(n)) \Leftrightarrow (b \in \sigma(n) \vee c \in \sigma(n))$
Intersection	$a \triangleq b * c$	$\forall n \in \mathbb{N}^+.(a \in \sigma(n)) \Leftrightarrow (b \in \sigma(n)) \wedge (c \in \sigma(n))$
Infimum	$a \triangleq b \wedge c$	$\forall n \in \mathbb{N}^+.H_\sigma(a,n) = \max(H_\sigma(b,n), H_\sigma(c,n))$
Supremum	$a \triangleq b \vee c$	$\forall n \in \mathbb{N}^+.H_\sigma(a,n) = \min(H_\sigma(b,n), H_\sigma(c,n))$
Delay	$a \triangleq b\$d$	$\forall n \in \mathbb{N}^+.H_\sigma(a,n) = \max(H_\sigma(b,n) - d, 0)$
DelayFor	$a \triangleq b\$d$ on c	$\forall n \in \mathbb{N}^+.(a \in \sigma(n)) \Leftrightarrow \big(c \in \sigma(n) \wedge \exists m \in \mathbb{N}^+. (b \in \sigma(m) \wedge H'_\sigma(c,n,m) = d)\big)$
Periodicity	$a \triangleq b \propto p$	$\forall n \in \mathbb{N}^+.(a \in \sigma(n)) \Leftrightarrow (b \in \sigma(n) \wedge (H_\sigma(b,n) + 1) \mod p = 0)$

Algorithm 1 Generate Spatial Event $\mathcal{E}_{detach(A,B)}$

Input:
Ensemble of obstacles, $O(i), i \in n$;
The gap between the autonomous vehicles, gap;
The current lane by the autonomous vehicle,V_{lane}
The position of the autonomous vehicle V_{pos}
Output:
True or False;
Extracting the current lane and pos of obstacles(i);
while $i < n$ **do**
  if $V_{lane} == O(i).lane$ **then**
    if $gap <= g_{safe}$ **then**
      $V_{new} = V_{pos} + brakingD(vMax)$;
      $O_{new} = O(i).pos + obstacleD(vMax)$;
      if $O_{new} - V_{new} >= g_{safe}$ &&!$O(i).static$ **then**
        return True;
      else
        if $O(i).pos - V_{new} >= g_{safe}$&&$O(i).static$ **then**;
          return True;
return False;

calculation. If the distance between the autonomous vehicle and the obstacle exceeds the safety threshold, the function returns true, otherwise, it returns false.

B. Proposed extension of CCSL: Stochastic stCCSL

Based on the set of these spatial events (In Section III-A) occurring in time and combined with the concept of logical clocks, spatial events can be directly transformed into logical clocks for processing [5].

In this paper, we consider extending the CCSL operators with stochastic characteristics. This stochastic characterization helps to clarify the uncertainty of the environment.

Definition 5 *(Probability Clock):*A logical clock c(p) where $p \in [0,1]$ is a rational number presenting the probability.

Definition 6 *(Interval Parametric DelayFor):*

$$a \triangleq b \ \$ \ [lower, upper] \text{ on } c \tag{1}$$

where parameter $lower$, $upper(lower < upper)$ are two natural numbers representing the lower and upper delay bounds.

Definition 7 *(Stochastic DelayFor):*

$$a \triangleq b \ \$ \ delay(F) \text{ on } c \tag{2}$$

where delay function F is defined as two types of probability density function:$Normal(\mu, \sigma)$, $Exp(\theta)$. $delay(F)$ describes the probability distribution of the waiting period before the timeout is reached.

Semantics of Stochastic stCCSL To conduct thorough analyzes on CCSL specifications, [6] propose to represent the semantics using transition systems.

Definition 8 *(Labeled Transition System):*Labeled Transition System(LTS) is defined as a tuple $\mathcal{A} = (S, s_0, T, A)$,where: 1) S is a set of states, 2) $s_0 \in S$ is the initial state, 3) A is a set of labels, 4) $T \subseteq S \times A \times S$ is a set of transitions.

Definition 9 *(Clock-Labeled Transition System):*Clock-Labeled Transition System (CLTS) is defined as a tuple $\mathcal{A} = (S, s_0, T, C)$, where: 1) S is a set of states, 2) $s_0 \in S$ is the initial state, 3) C is a finite set of clocks, 4) $T \subseteq S \times 2^C \times S$ is a set of transitions.

Definition 10 *(Stochastic Clock-Labeled Transition System):* Stochastic Clock-Labeled Transition System (SCLTS) is defined as a tuple $\mathcal{A} = (S, s_0, C, P, d, T)$, where: 1) S is a set of states, 2) $s_0 \in S$ is the initial state, 3) C is a finite set of clocks, 4) $P \subseteq \mathbb{Q}$ is the set of rational numbers between 0 and 1, 5) d is a stochastic variant that follows the exponential distribution, 6) $T \subseteq S \times 2^C \times P \times S$ is a set of transitions.

For instance, the constraint Subclock $a \subseteq b$, its transition system is given in Fig. 4. The constraint Delay $a \triangleq b\$d$, its transition system is given in Fig. 5.

C. Transform Stochastic stCCSL to SHA

Some previous work [7] has considered the encoding of Mode/State-based MARTE/CCSL behavior into Timed Automata (TA). In this section, we incorporate stochastic and continuous behavior into SHA to reinforce it. In general, the mapping rules are summarized in table II.

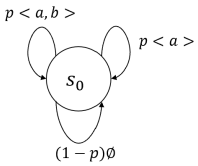


Fig. 4: Example for SCLTS:$a \subseteq b$

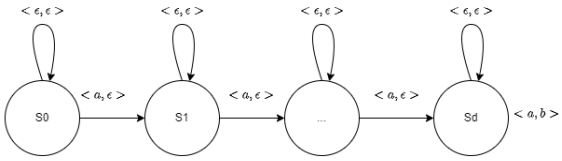
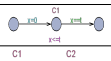
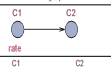
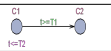
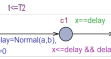
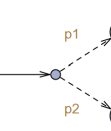


Fig. 5: Example for SCLTS: $a \triangleq b\$d$

Meanwhile, CCSL utilizes logical clocks to define partial orders and causal relationships among events. In this study, we incorporate logical clocks to capture spatial and temporal constraints during the specification process. Specifically, the formulation of the spatial events and temporal constraints is achieved through the Stochastic stCCSL.

TABLE II: The mapping rules between Stochastic stCCSL and SHA.

Stochastic stCCSL	***SHA***	***Remarks***
$c1 \triangleq c1\$t$		It denotes that the time delay in state c1 is t
$c2 \triangleq c1\$delay(Exp(rate))$		It denotes exponential time delay with parameter $rate$
$c2 \triangleq c1\$[T1,T2]$		It denotes uniform distribution interval time delay
$c2 \triangleq c1\$delay(Normal(a,b))$		It denotes Gaussian distribution time delay
$c2(p1)\#c3(p2)$		It denotes probabilistic choice. $p1+p2=1(P1,P2 \in [0,1])$

IV. Case Study

A. A right turn scenario for Autonomous Vehicles

The perception devices in autonomous vehicles acquire environmental information from the surroundings, transmitting it to the recognizer. The recognizer identifies various elements such as traffic signs, obstacles, pedestrians, and surrounding vehicles, and makes judgments. These judgment results are transmitted to the controller, which generates corresponding control actions.

Fig. 6 shows a scenario for an autonomous vehicle taking a right turn at the intersection. In this scenario, the gray vehicle in the straight line is a human-driven car, and the green vehicle is an autonomous vehicle. Note that according to traffic rules, green Vehicle has a higher priority.

First of all, we give several requirements utilizing natural language. The requirements are as follows:

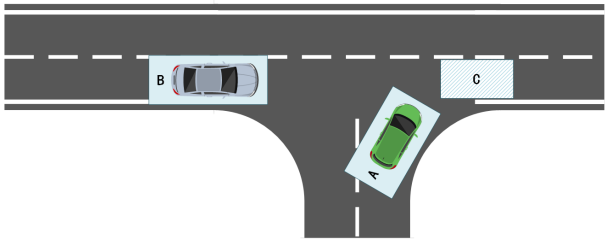


Fig. 6: A right turn scenario at the intersection

a) : The autonomous vehicle probabilistically executes a right turn considering the time delay associated with environmental information.

b) : Upon receiving the $check$ command, the autonomous vehicle's sensors collect environmental information every 50 ms with a jitter.

c) : The classifier computes the spatial relationship between the autonomous vehicle and the vehicles in the through lane based on the collected environmental information, assuming the merging process concludes within $[25, 30]$ ms.

d) : Area A represents the autonomous vehicle area, area B represents the through lane area, and area C represents the merging area. A must merge into and leave the C area before B can enter.

e) : If A has not completely left the C area when B is about to enter the area, A needs to accelerate through the C area.

f) : If the classifier predicts that both A and B will enter the C area simultaneously, A will not turn right and will decelerate until it stops.

g) : The classifier sends the classification result to the controller within 30 ms.

h) : The three requirements of d, e, and f for right turns are mutually exclusive, although execution can also be probabilistic.

B. Build the Stochastic stCCSL and map to UPPAAL-SMC model

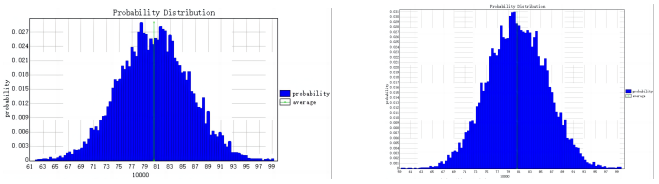
We describe the requirement as the Stochastic stCCSL. Table III presents the specifications a–h and their corresponding verification results obtained after performing 1000 simulation runs.

Specification a involves receiving a right turn signal ($TurnR$) and an environment detection command. Specification b indicates a sensor's completion of environmental data collection, introducing a delay centered around 50ms with a Gaussian distribution (mean $\mu = 50$, deviation $\sigma = 5$). The specification of c calculates spatial relationships within a 25-30ms timeframe. The specification d-f indicates spatio-temporal constraint. Based on the calculated spatial event logical relationship function, d indicates that turning is safe, that is, $exclude(A, C) \prec join(B, C)$, where $exclude(A, C)$ and $join(B, C)$ are obtained from the aforementioned spatio-temporal function calculation. If this spatio-temporal event logical relationship holds, the detector issues a $turn\_safe$

signal. Similarly, if the e logical relationship is satisfied, the detector issues a $turn\_risksignal$. If f is satisfied, the detector issues a $turn\_danger$ signal. g indicates that these signals are sent to the controller within 30ms. h expresses the mutual exclusion relationship of the first three spatial events.

C. Build the UPPAAL-SMC model for the whole system

Based on the establishment of Stochastic stCCSL and the verification of the foundation discussed earlier, Fig. 8 shows the overall UPPAAL model of the right turn scenario. The system calculates the transition to three states: ok_turn, $risk\_turn$, and $slowdown$, based on the spatio-temporal evolution relationship. The vehicle continues normal operation during the transition from the ok_turn state to the $normal$ state. When a vehicle transitions to the $risk\_turn$ state, it signifies a potential rear-end collision with the following vehicle. Consequently, the vehicle must accelerate to exit the $risk\_turn$ state and return to the $normal$ state. If the vehicle enters the $slowdown$ state, it slows down while receiving the periodic signal $check$. The transition of these three parallel states is currently based on logical spatio-temporal constraints. It is essential to emphasize the probabilistic nature of right turns, which are influenced by various environmental factors. Therefore, assessing the risk associated with right turns and choosing between normal, aggressive, or impossible options entails probabilistic decision-making in an uncertain environment.



(a) (b)

Fig. 7: Probability distribution with confidence 95% and 98% in the query

In order to verify the probability distribution of the time interval from when the system sends a right-turn signal to when it makes a decision, we illustrate a query of the model.

$Pr[<= 100](<> Process.decide)$

The query means the probability distribution of reaching state decide within 100-time units

We conducted two trials for this run using various statistical settings: 1) We set the statistical parameters of UPPAAL-SMC with $\alpha = 0.05$, $\epsilon = 0.04$. By simulating 10000 runs, the quantitative result is shown in Fig. 7(a). We can get a confidence interval $[61.8, 99.3]$ with a confidence 95%. 2) We set the statistical parameters of UPPAAL-SMC with $\alpha = 0.02$, $\epsilon = 0.01$. By simulating 10000 runs, the quantitative result is shown in Fig. 7(b). We can get a confidence interval $[59.2, 99.8]$ with a confidence 98%. In these figs, the x-axis indicates the time limit, and the y-axis denotes the probability density distribution.

V. Related Work

This section compares our approach to related works. Du $et\ al.$ [8] proposed pCCSL, a stochastic extension to MARTE/CCSL for modeling uncertainty in CPS, and used SMC to explore alternative solutions and drive the refinement process. They illustrate their proposition by modeling an energy-aware building. Huang $et\ al.$ [9] proposed an extension of PrCCSL, called PrCCSL*, to specify and verify dynamic and stochastic behaviors for automotive systems using UPPAAL-SMC. Gao $et\ al.$ [10] enhanced CCSL by adding parameters to constraints in order to represent uncertainties in temporal behaviors. Compared to their approach, our approach considers both spatio-temporal constraints and stochastic behavior simultaneously.

VI. Conclusion and Future Work

In this paper, CCSL is expanded to propose time delay constraint relations including probability intervals and density functions. We introduce time-like temporal constraints and develop logical relation functions for autonomous driving. A mapping method is proposed for this expanded constraint language, facilitating the conversion and verification using the UPPAAL model. This spatio-temporal probabilistic language is applied to the right turn example, involving formal modeling, distribution verification, and overall model evaluation for the system. These advancements enable more precise modeling of uncertainties in intelligent vehicle systems

In our future work, we propose integrating deep learning-based uncertainty models with rule-based mathematical models to construct comprehensive traffic regulations for typical autonomous driving scenarios. Additionally, we plan to develop refined models to enhance evaluation and validation.

Acknowledgment

This work was supported in part by the National Key Research and Development under Project 2022YFB3305202, the NSFC under Project 61972150, Shanghai Trusted Industry Internet Software Collaborative Innovation Center, and Shanghai Post-Doctoral Excellence Program 2021146.

References

[1] E. A. Lee, "Cyber Physical Systems: Design Challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, May 2008, pp. 363–369.

[2] M. Faugere, T. Bourbeau, R. De Simone, and S. Gerard, "Marte: Also an uml profile for modeling aadl applications," in *12th IEEE International Conference on Engineering Complex Computer Systems (ICECCS 2007)*, 2007, pp. 359–364.

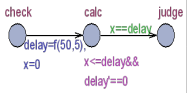
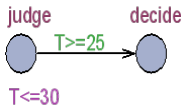
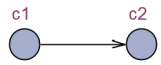
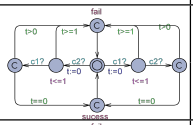
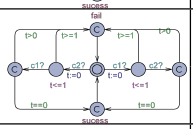
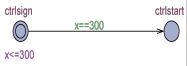
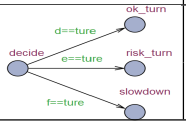
[3] C. André, "Syntax and Semantics of the Clock Constraint Specification Language (CCSL)," report, INRIA, 2009.

[4] M. Fr¨anzle, E. M. Hahn, H. Hermanns, N. Wolovick, and L. Zhang, "Measurability and safety verification for stochastic hybrid systems," in Proceedings of the 14th international conference on Hybrid systems: computation and control, 2011, pp. 43–52.

[5] Q. Liu et al., "Multiform Logical Time Space for Mobile Cyber-Physical System With Automated Driving Assistance System," in *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*, Dec. 2020, pp. 415–424.

[6] F. Mallet and R. de Simone, "Correctness issues on MARTE/CCSL constraints," *Science of Computer Programming*, vol. 106, pp. 78–92, Aug. 2015.

TABLE III: Verification Results

Req	*Specification*	*Expression*	*UPPAAL*	*Result*	*Time (ms)*	*Mem (Mb)*
a	Stochastic stCCSL	$check \triangleq TurnR\$delay(Exp(rate))$		valid	3.91	8.95
	UPPAAL	$Pr[<= 100](<> a.check)$				
b	Stochastic stCCSL	$judge \triangleq check\$delay(Normal(50,5))on\ ms$		valid	4.01	9.25
	UPPAAL	$Pr[<= 100](<> b.judge)$				
c	Stochastic stCCSL	$decide \triangleq judge\$[25,30]on\ ms$		valid	16.0	9.56
	UPPAAL	$Pr[<= 40](<> c.decide)$				
d	Stochastic stCCSL	$exclude(A,C) \prec join(B,C)$		valid	3.03	9.02
	UPPAAL	$Pr[<= 20](<> d.c2)$				
e	Stochastic stCCSL	$detach(A,B) \triangleq join(B,C)$		valid	15.0	9.45
	UPPAAL	$Pr[<= 10;1000](<> e.sucess\|\|t == 0)$				
f	Stochastic stCCSL	$join(A,C) \triangleq join(B,C)$		valid	47.0	9.62
	UPPAAL	$Pr[<= 10;10000](<> f.sucess\|\|t > 0.1)$				
g	Stochastic stCCSL	$ctrlstart \triangleq ctrlsign\ \$\ [0,30]\ on\ ms$		valid	3.94	9.52
	UPPAAL	$Pr[<= 400](<> g.ctrlstart)$				
h	Stochastic stCCSL	$ok\_turn\#risk\_turn\#slowdown$		valid	16	9.29
	UPPAAL	$Pr[<= 10;500](<> h.ok\_turn\|\|e == true)$				

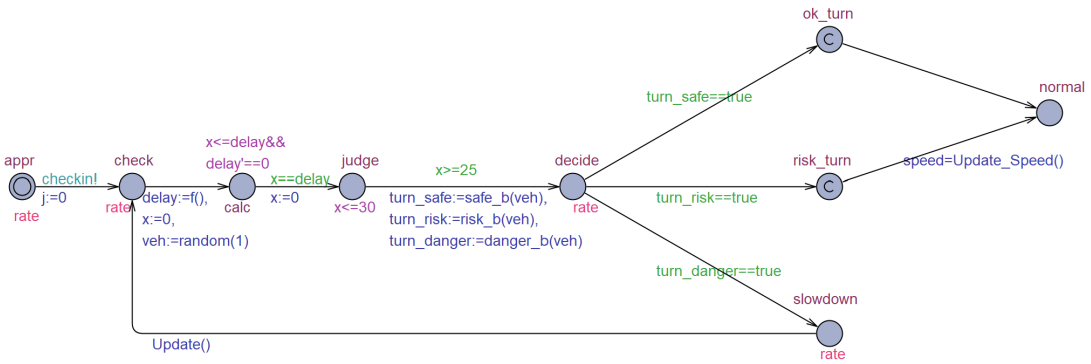


Fig. 8: Autonomous Vehicle behaviors model in UPPAAL-SMC

[7] J. Suryadevara, C. Seceleanu, F. Mallet, and P. Pettersson, "Verifying MARTE/CCSL Mode Behaviors Using UPPAAL," in *Software Engineering and Formal Methods*, Berlin, Heidelberg, 2013, pp. 1–15.

[8] D. Du, P. Huang, K. Jiang, and F. Mallet, "pCSSL: A stochastic extension to MARTE/CCSL for modeling uncertainty in Cyber Physical Systems," *Science of Computer Programming*, vol. 166, pp. 71–88, Nov. 2018.

[9] L. Huang, T. Liang, and E.-Y. Kang, "Formal Verification of Dynamic and Stochastic Behaviors for Automotive Systems," in *2019 24th International Conference on Engineering of Complex Computer Systems (ICECCS)*, Nov. 2019, pp. 11–20.

[10] F. Gao, F. Mallet, M. Zhang, and M. Chen, "Modeling and Verifying Uncertainty-Aware Timing Behaviors using Parametric Logical Time Constraint," in *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, Mar. 2020, pp. 376–381.