Stochastic uncertainty in CCSL

Pavlo Tokariev

Université Côte d'Azur, Inria, CNRS, i3S
Sophia Antipolis, France

Tuesday 25th November, 2025

Motivation

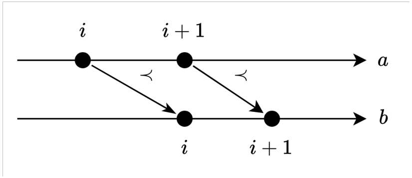
- Uncertainty is a state of partial knowledge
- Implementation is uncertain
  - From component production
  - From finite precision of testing
  - From environment
- Model and development can be uncertain
  - Not every requirement is known
  - If a requirement is not yet precise
  - By design: HAL4SDV
- Proving properties regardless of uncertainty is crucial
- Thus capturing the uncertainty is important

The Clock Constraint Specification Language (CCSL) [6, 2]

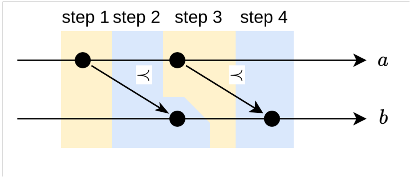
- A language that specifies the temporal behaviour of a system
- Variables represent logical clocks
- Constraints are relations between the logical clocks
- Logical clock is a possibly infinite totally ordered sequence of time instants
  $c = c_0 < c_1 < \ldots, c_i \in I$
- A time structure $\langle I, \equiv_I, \prec_I \rangle$
  - A solution that satisfies the constraints
  - Commonly, a schedule/trace, a sequence of steps, totally ordering the instants
- The problem
  - For a specification, there is a set of valid schedules
  - Find out if it is empty

Example: precedence

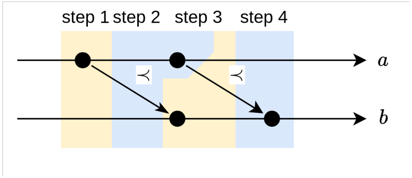
$$\langle I, \equiv_I, \prec_I \rangle \vDash \underbrace{a \prec b}_{\text{constraint}} \iff \underbrace{\forall i : a_i \prec_I b_i}_{\text{semantics}}$$



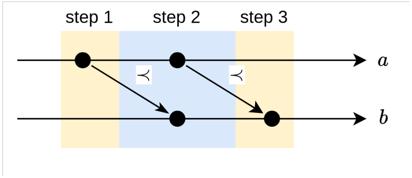
(a) Tick relations



(b) Solution 1



(c) Solution 2



(d) Solution 3

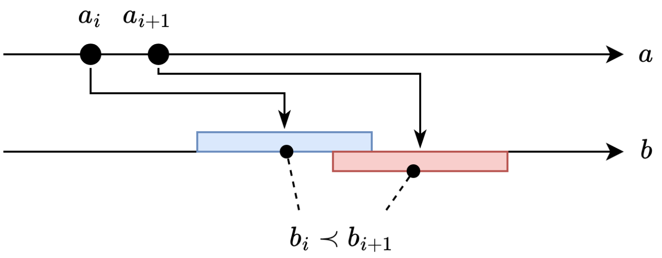
Uncertainty so far

- CCSL already can express uncertainty
  - By missing constraints
  - Steps have unknown width
  - Using auxiliary constructions
- In case of quantitative time ("delay event by 2.1ms") — complicated to solve
- With constraints of RTCCSL it is easier and simpler to express quantitative time with its uncertainty

Real-Time CCSL [9, 7]

- Expresses quantitative time and its uncertainty
- Adds 3 new constraints: real-time delay, periodic with jitter or drift
- Instants are considered to be exactly $\mathbb{Q}$
- For example, real-time delay:

$$\underbrace{\forall x \leq y \in \mathbb{Q}_{\geq 0}}_{\text{bounds}} : \langle I, \equiv_I, \prec_I \rangle \vDash \underbrace{b = \text{delay } a \text{ by } [x, y]}_{\text{constraint}} \iff \underbrace{\forall i : b_i - a_i \in [x, y]}_{\text{semantics}}$$



Modular CCSL [7]

- Based on subspecification relation $\Subset$
  - Inclusion of language projection: $A \Subset B \stackrel{\text{def}}{=} \Pi_{C(A)\cap C(B)}(A) \subseteq \Pi_{C(A)\cap C(B)}(B)$
  - Simulation relation on common subset of clocks
- Module is defined as tuple $\langle A, S, G \rangle$
  - **A**ssumption, **S**tructure, **G**uarantee
  - The module "body" $B \stackrel{\text{def}}{=} A \wedge S$
  - It is valid when $A \Subset B \Subset G$
- Modules can be chained
- Nearly not implemented right now
  - Only certain types of specifications are simple to check
  - Still useful as separation of concerns

Contribution

- Specification defines what is possible, not what is likely to happen
- Stochastic constraints guide the simulation of uncertain specification to the more probable trace
- Traces can be processed to extract representative system metrics like response time using functional chain description

Stochastic extension

- Prerequisite: uncertainty is detached from constraints
- In real-time delay:

$$\forall v = (v_0 v_1 \ldots), v_i \in \mathbb{Q} : \langle I, \equiv_I, \prec_I \rangle \models \underbrace{b = \text{delay } a \text{ by } v}_{\text{constraint}} \iff \underbrace{\forall i : b_i - a_i = v_i}_{\text{semantics}}$$

  - Periodic, logical delay, subclocking are modified this way too
  - Additional object in a specification: rational or integer sequence variable
- Uncertainty is separately specified on the sequence variables

$$\forall x = (x_0 x_1 \ldots), x_i, c \in \mathbb{Q} : x \bowtie c \iff \forall i : x_i \bowtie^{\mathbb{Q}} c$$

- Stochastic constraints define how the elements are distributed
  - Implemented few classic distributions, normal, exponential, uniform
  - `continuously distributed x as normal (mu, sigma)`

$$\forall i : x_i \sim N(\mu, \sigma)$$

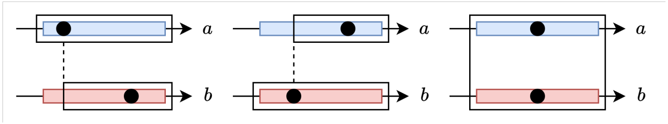
  - Uncertainty bounds + distribution result in truncated distribution

Limitations

- The distributions need to be independent and stable in their domain:
  - No double distribution on sequences
  - Bounds should be present and not modified
  - Both should be placed in assumptions
  - Assumptions guarantee that nothing can interfere with the distribution
- Variables cannot be used twice in different constraints
  - Otherwise, an arbitrary dependency between past and future quantitative time
  - Requires that sequences have to be remembered as the constraint state
- Comparison between sequences is not allowed

Simulation

- Behaviour exploration is prioritized
- When distribution is unspecified, some value is picked, uniformly, but **not** in the original interval

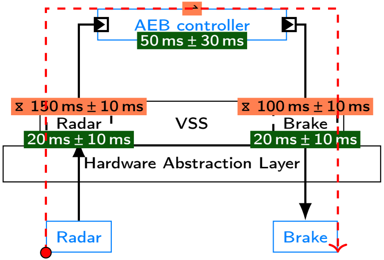


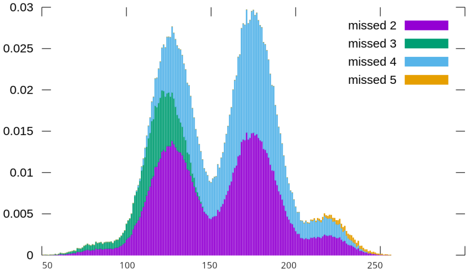
$$P(a \wedge b) = 0$$

- With stochastic constraints: mix between event-driven simulation and behaviour exploration

Application

- In HAL4SDV project [8]
  - A DSL[a] that describes a Service-oriented architecture in a Software Defined Vehicle
  - Translates into presented language and simulates[b] the system
  - So far described a version of Autonomous Emergency Braking System
- ABZ use cases: landing gear and mechanical lung ventilator





---

[a] `https://github.com/jdeantoni/SoftwareDefinedVehicleModelingLanguage`

[b] https://github.com/PaulRaUnite/mrtccsl

Related works

- Network of Stochastic Timed Automata [1]
  - Synchronization: broadcast vs rendez-vous
  - Completely independent automata vs only stochastic part
  - Additive vs subtractive description
- Probabilistic CCSL [3]
  - Defines probability of an event in a step
    - Explicitly
    - Transitively through expression constraints
  - Redefines constraints
  - Only makes sense when step is fixed size (hard requirement)
- PrCCSL [5] and PrCCSL* [4]
  - Expresses uncertainty about the constraint itself, not time or parameters
  - No recovery strategies, so a violation invalidates the whole trace

Conclusion and future work

- Conceptualized the extension that
  - Enriches uncertainty in constraints with stochastic aspect
  - Helps to guide the simulation into more "real" and fair traces
- Almost fully implemented
  - Lacking reencoding of constraints related to integer sequences
- Future work
  - Uncertain coincidence as a separate constraint
    - Rational instants are too precise
    - Previously “indeterminate size” steps were uncertain
  - Extraction of specification distributions

References I

[1] Patricia Bouyer et al. "Compositional Design of Stochastic Timed Automata". In: *Computer Science – Theory and Applications*. Ed. by Alexander S. Kulikov and Gerhard J. Woeginger. Vol. 9691. Cham: Springer International Publishing, 2016, pp. 117–130. ISBN: 978-3-319-34170-5 978-3-319-34171-2. DOI: 10.1007/978-3-319-34171-2_9. URL: http://link.springer.com/10.1007/978-3-319-34171-2_9.

[2] Julien Deantoni, Charles André, and Régis Gascon. "CCSL Denotational Semantics". report. Inria, Nov. 13, 2014, p. 29. URL: https://hal.inria.fr/hal-01082274.

[3] Dehui Du et al. "pCCSL: A Stochastic Extension to MARTE/CCSL for Modeling Uncertainty in Cyber Physical Systems". In: *Science of Computer Programming* 166 (Nov. 15, 2018), pp. 71–88. ISSN: 0167-6423. DOI: 10.1016/j.scico.2018.05.005. URL: https://www.sciencedirect.com/science/article/pii/S0167642318301916.

References II

[4] Li Huang, Tian Liang, and Eun-Young Kang. "Formal Verification of Dynamic and Stochastic Behaviors for Automotive Systems". In: *2019 24th International Conference on Engineering of Complex Computer Systems (ICECCS)*. 2019 24th International Conference on Engineering of Complex Computer Systems (ICECCS). Nov. 2019, pp. 11–20. DOI: 10.1109/ICECCS.2019.00009. URL: https://ieeexplore.ieee.org/document/8882750.

[5] Eun-Young Kang, Dongrui Mu, and Li Huang. "Probabilistic Verification of Timing Constraints in Automotive Systems Using UPPAAL-SMC". In: *Integrated Formal Methods*. Ed. by Carlo A. Furia and Kirsten Winter. Cham: Springer International Publishing, 2018, pp. 236–254. ISBN: 978-3-319-98938-9. DOI: 10.1007/978-3-319-98938-9_14.

[6] Frédéric Mallet. "Clock Constraint Specification Language: Specifying Clock Constraints with UML/MARTE". In: *Innovations in Systems and Software Engineering* 4 (Oct. 1, 2008), pp. 309–314. DOI: 10/dn4ptd.

References III

[7] Pavlo Tokariev. "Modular Real-Time Clock Constraint Specification Language". PhD thesis. Université Côte d'Azur, Dec. 13, 2024. URL: https://theses.hal.science/tel-04933243.

[8] Pavlo Tokariev, Irman Faqrizal, and Julien Deantoni. "Understandable Timing Analysis of Service-Oriented Architecture Components in Software-Defined Vehicle". In: *Communications in Computer and Information Science. CCIS*. Proceedings of the 20th Int. Conf. on Information and Communication Technologies in Education, Research, and Industrial Applications (ICTERI-2025). Nice, France, Sept. 2025. URL: https://inria.hal.science/hal-05224373.

[9] Pavlo Tokariev and Frédéric Mallet. "Real-Time CCSL: Application to the Mechanical Lung Ventilator". In: ABZ 2024 – 10th International Conference on Rigorous State Based Methods. Vol. LNCS-14759. Springer, June 25, 2024, p. 289. DOI: 10.1007/978-3-031-63790-2_24. URL: https://inria.hal.science/hal-04639949.