Carl von Ossietzky
Universität
Oldenburg

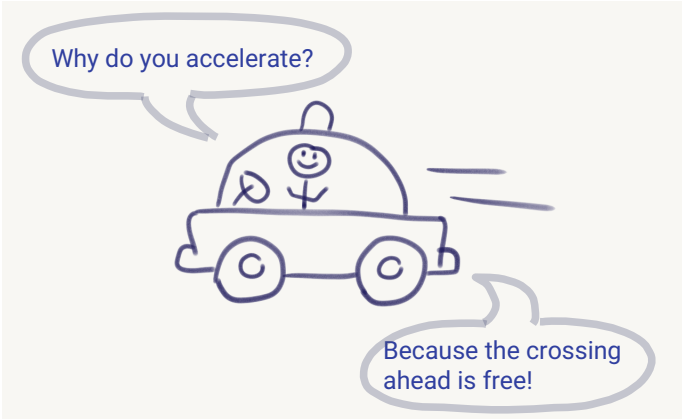
A Quest of Self-Explainability:

When Causal Diagrams meet Autonomous Urban Traffic Manoeuvres

Dr. Maike Schwammberger @RE4ES'21

21. September 2021

Motivation



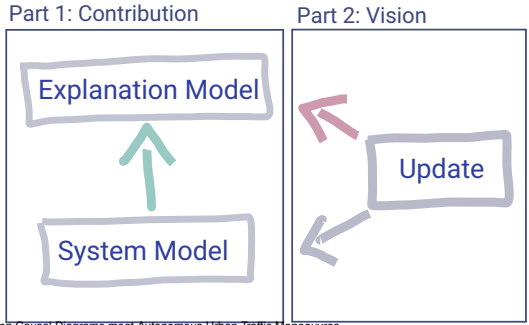
This Talk

What do I not do?

- What types of explanations are necessary?
- How should an explanation be delivered?
- ...

Instead:

- How do I retrieve an explanation model from a system model?
- Vision: How can the explanation model be updated?



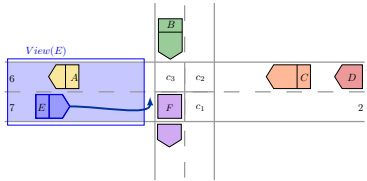
Part 1 (Contribution)

Retrieving an Explanation Model from a System Model

System Model

Crossing Controller for autonomous turn manoeuvres at intersections

- Some functional requirements have been proven:
  - Safety (collision freedom),
  - Liveness (something good happens) and
  - Fairness (no queue-jumping).



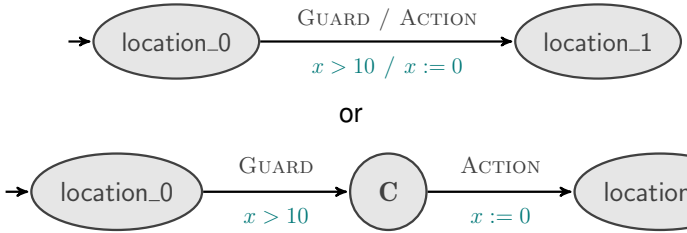
- Now: Introduce self-reflective/ non-functional capability to the Crossing Controller
  - Self-explainability (of the controller's actions)

- Semantic Model of Crossing Controller:
  - Automotive-Controlling Timed Automata (ACTA)

[S18a] S., M.: An Abstract Model for Proving Safety of Autonomous Urban Traffic (TCS Journal, 2018)
[BS19] Bischopink, C., S., M.: *Verification of fair controllers for urban traffic manoeuvres at intersections (FMAS@FMWeek19)*

System Model

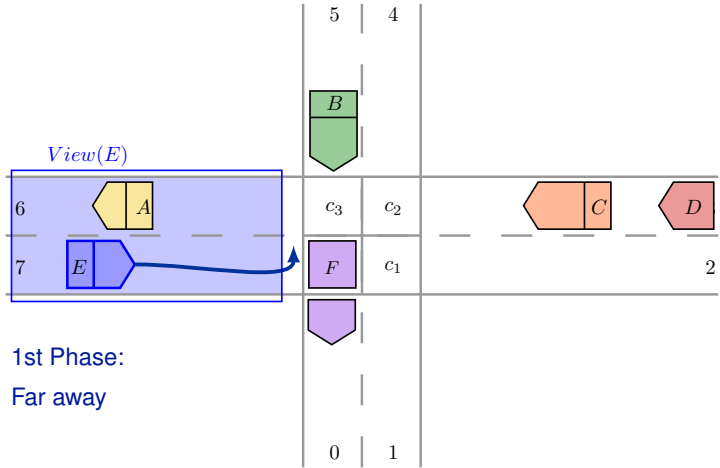
Structure of (Automotive-Controlling) Timed Automata:



Explanation type that we consider: "ACTION *happened because of* GUARD."

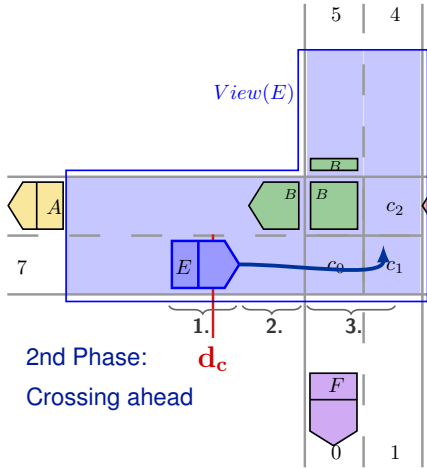
Example: "$x := 0$ *happened because of* $x > 10$."

Example: Crossing Controller



[S18a:] S., M.: An Abstract Model for Proving Safety of Autonomous Urban Traffic (TCS Journal, 2018)

Example: Crossing Controller



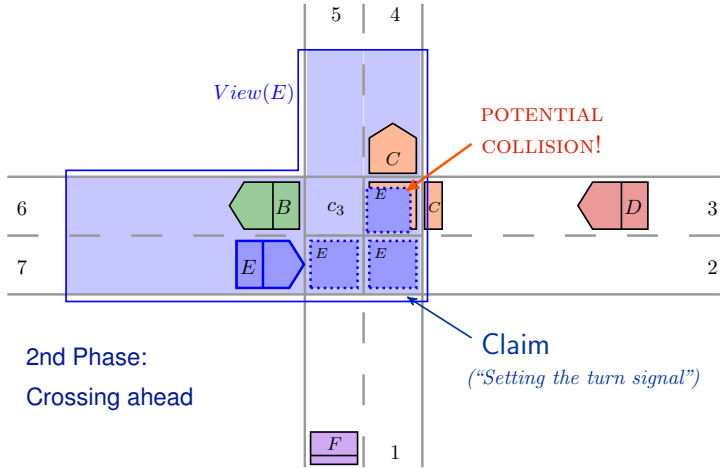
Urban Multi-lane Spatial Logic:

$$ca(E) \equiv \underbrace{re(E)}_{1.} \frown \underbrace{free^{<d_c} \wedge \neg\langle cs\rangle}_{2.} \frown \underbrace{cs}_{3.}$$

(“Crossing ahead check”)

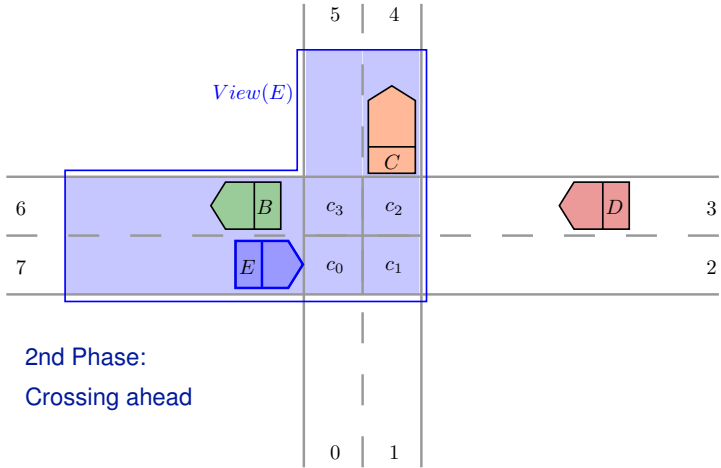
[S18a:] S., M.: An Abstract Model for Proving Safety of Autonomous Urban Traffic (TCS Journal, 2018)

Example: Crossing Controller



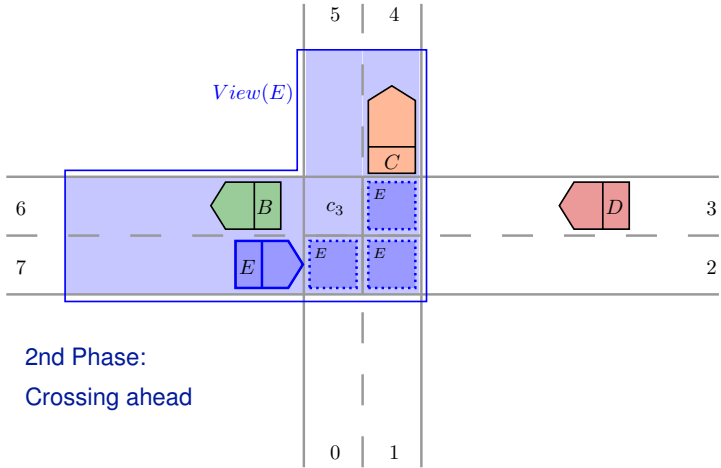
[S18a:] S., M.: An Abstract Model for Proving Safety of Autonomous Urban Traffic (TCS Journal, 2018)

Example: Crossing Controller



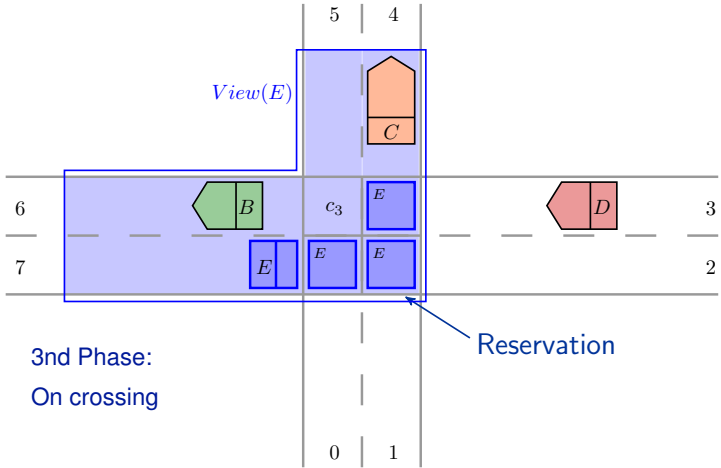
[S18a:] S., M.: An Abstract Model for Proving Safety of Autonomous Urban Traffic (TCS Journal, 2018)

Example: Crossing Controller



[S18a:] S., M.: An Abstract Model for Proving Safety of Autonomous Urban Traffic (TCS Journal, 2018)

Example: Crossing Controller



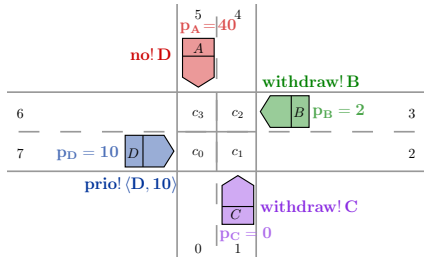
[S18a:] S., M.: An Abstract Model for Proving Safety of Autonomous Urban Traffic (TCS Journal, 2018)

Communicating Priorities

Fairness: *No car has to wait unreasonably long in front of an intersection.*

Approach:

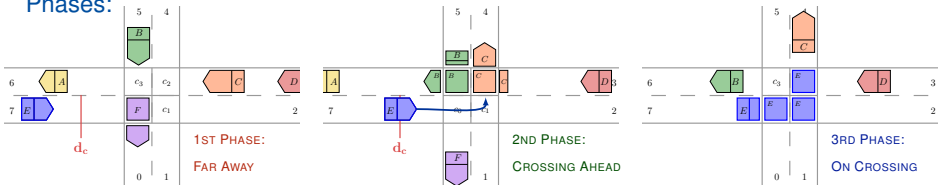
- Send priority on arriving at intersection (“prio! ⟨ D, 10 ⟩”)
- Helper Controllers determine whether priority is large enough
  - Own helper (e.g. B) determines own priority is too small: (“withdraw! ⟨ B ⟩”)
  - Other helper (e.g. A) determines D’s priority is too small: (“no! ⟨ D ⟩”)



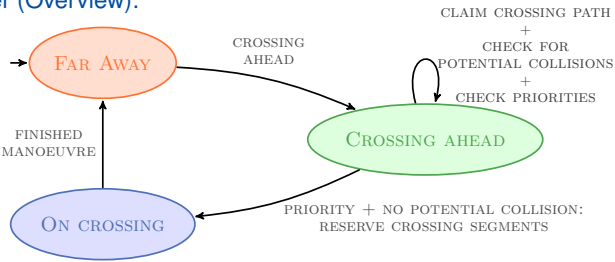
[BS19] Bischopink, C., S., M.: *Verification of fair controllers for urban traffic manoeuvres at intersections (FMAS@FMWeek19)*

Crossing Controller

Phases:



Crossing Controller (Overview):

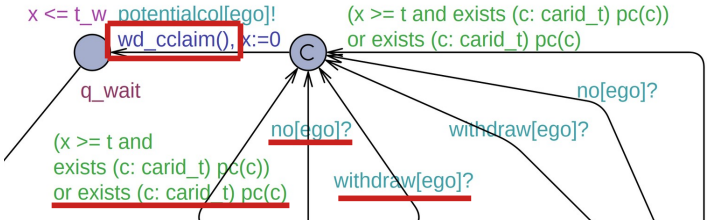


Crossing Controller

- Focus: 2nd Phase (crossing ahead), **action** “withdraw claim”
- Three possible types of transitions:

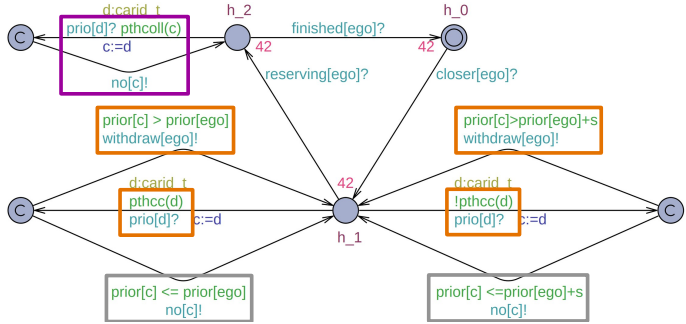
1. Message “withdraw[ego]?” received
2. Message “no[ego]?” received
3. Potential collision detected (after *t* time) (“exists (c: carid_t) pc(c)”) (**cause 1**)

⇒ “no” and “withdraw” sent by helper controller
⇒ Identify guards behing these messages in helper controller



Helper Controller

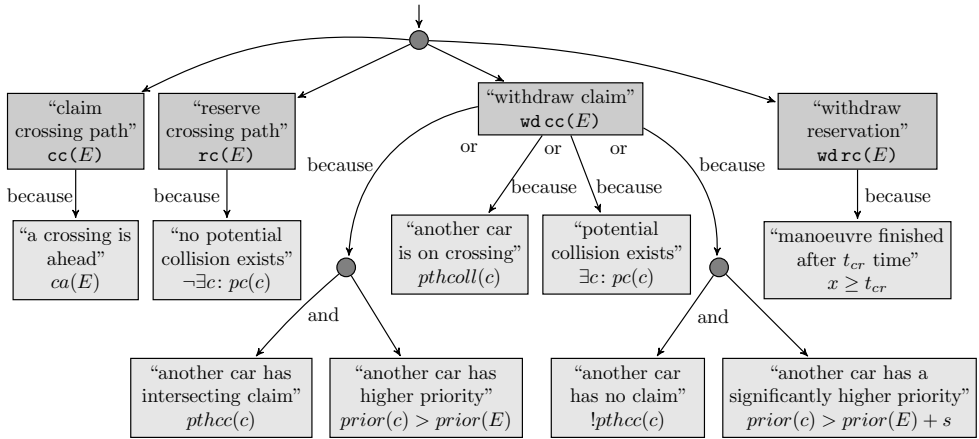
- withdraw:
  - *“Other has intersecting claim (pthcc(d))”* and *“Priority of other is larger”* (***cause 2***)
  - *“Other has no int. claim (!pthcc(d))”* and *“Priority of other is significantly (s) larger”* (***cause 3***)
- no: Similar (“inverse”) to withdraw
- no: “Other car already on crossing (pthcoll(c))” (**cause 4**)



[BS19] Bischopink, C., S., M.: *Verification of fair controllers for urban traffic manoeuvres at intersections (FMAS@FMWeek19)*

Explanation Model

Causal Diagram related to Crossing Controller:



Self-Explainability

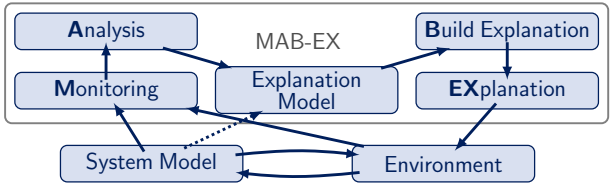
Follow the MAB-EX Framework for Self-Explainability:

Monitoring: Observe system (e.g. with Observer Automata)

Analysis: Detect need for explanation (e.g. transition was triggered)

Build Explanation: Extract explanation path from explanation model

Explain: Give explanation to recipient (user, other car,...)



Blumreiter, M., Greenyer, J., Chiyah Garcia, F. J., Klös, V., Schwammberger, M., Sommer, C., Vogelsang, A., Wortmann, A.:
Towards Self-Explainable Cyber-Physical Systems. MODELS Companion 2019

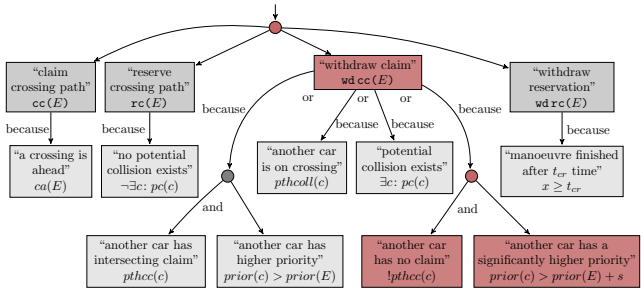
Example Explanation

Extracted explanation path:

$$\text{because}(\texttt{wd cc(E)}, \text{and}\,(!\mathit{pthcc}(c), \mathit{prior}(c) > \mathit{prior}(E) + s)),$$

Explanation:

"Car E did withdraw claim, because another car has no claim but a significantly higher priority."



Summary Part 1

- Extraction of Explanation Model from a System Model
- Answer to the “Build Explanation” Phase of MAB-EX Framework
- Universality of Semantic Model “ACTA”: Not only self-explainability for Crossing Controller

Open Questions (Part 1)

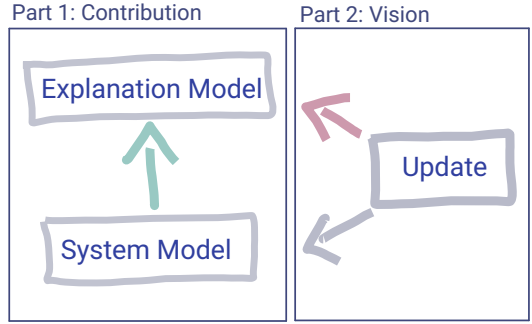
- Check technical completeness of explanation model
  - E.g. is it enough to connect guards/ causes with actions?
  - What are the requirements for an explanation model?
- Automatically extract explanation models for (automotive-controlling) timed automata
- What is provable about explanations?
  - E.g. correctness or completeness of explanation model?
- Improve presentation of explanations (e.g. through HMI methods)
- Examine the explanation model wrt actually needed explanation types
  - Technical explanation model vs. non-technical explanation model?

Part 2 (Vision)

Dynamic Updates of Explanation Models

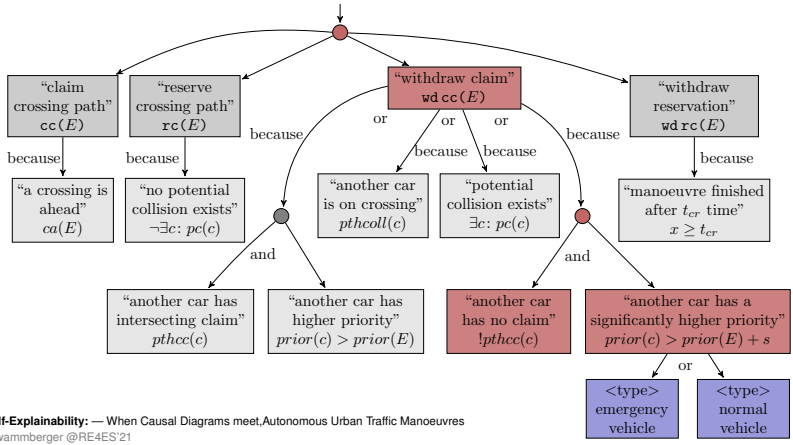
Explanation Model Updates

- System model needs to be updated at run-time (i.e. after it was sold)
  $\Rightarrow$ Different approaches for this exist (e.g. component-wise updates)
- Easy: An updated system model leads to an updated explanation model
- But what if the explanation model needs to be updated without a system model update?



Example

- Crossing Controller does not distinguish between different prioritised vehicles
  - But: Passenger might be interested which type of vehicle gets right of way (e.g. emergency vehicle)
  - Refinement of explanation model needed, but not of system model



Vision

Research Question:

– How do we identify that a run-time update of the explanation model is needed?

Possible Direction:

– Run-time/ Dynamic Requirements Engineering
  – E.g. after an unsatisfactory/ incomplete explanation was provided
  – With methods of dynamic RE, refinement of nodes in explanation model (?)
  – Benefit: Explanation model can be taylored to specific user

Open Questions

Part 2:

- How is the need for an update of the explanation model detected?
- What types of updates for the explanation model are necessary?

Open Questions

Part 2:

- How is the need for an update of the explanation model detected?
- What types of updates for the explanation model are necessary?

Part 1:

- Check technical completeness of explanation model
  - E.g. is it enough to connect guards/ causes with actions?
  - What are the requirements for an explanation model?
- Automatically extract explanation models for (automotive-controlling) timed automata
- What is provable about explanations?
  - E.g. correctness or completeness of explanation model?
- Improve presentation of explanations (e.g. through HMI methods)
- Examine the explanation model wrt actually needed explanation types
  - Technical explanation model vs. non-technical explanation model?

Literature

[BS19] Bischopink, C. and Schwammberger, M.: *Verification of fair controllers for urban traffic manoeuvres at intersections.* In *Proceedings of FMAS@FMWeek19* (2019).

[BWS+10] Bencomo, N., Whittle, J., Sawyer, P., Finkelstein, A. and Letier, E.: *Requirements reflection: Requirements as runtime entities* (2010).

[BGG+19] Blumreiter, M., Greenyer, J., Chiyah Garcia, F.J., Klös, V., Schwammberger, M., Sommer, C., Vogelsang, A. and Wortmann, A.: *Towards self-explainable cyber-physical systems.* In *Proceedings of 22nd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion* (2019).

[GEL+14] Ghosh, S., Elenius, D., Li, W., Lincoln, P., Shankar, N. and Steiner, W.: *Automatically extracting requirements specifications from natural language* (2014).

[S18a] Schwammberger, M.: *An abstract model for proving safety of autonomous urban traffic.* In *Theoretical Computing Science* journal, volume 744 (2018).