TECHNISCHE UNIVERSITÄT DRESDEN

2nd International Workshop on Multi-Paradigm Modelling for Cyber-Physical Systems (MPM4CPS2020)

Connecting Conceptual Models using Relational Reference Attribute Grammars

René Schöne, Johannes Mey, Sebastian Ebert, and Uwe Aßmann

Technische Universität Dresden

connector.relational-rags.eu

Problem: Constructing cyber-physical systems (CPS)

Challenges for connecting runtime models

Distribution: transparent communication with locally and remotely accessible models

Multi-Paradigm: support for different paradigms and (programming as well as modelling) languages

Fast, reactive behaviour: changes in input lead to automatic recomputation for fast reaction

Goals for realizing connected runtime models

① **Minimize development effort:** reduce workload of developers, but also minimize errors and redundancy

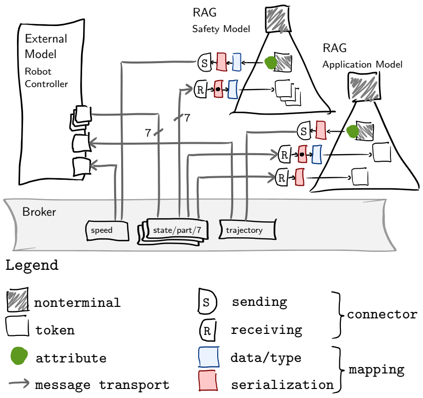
② **Minimize network usage:** cope with many updates, and/or little network bandwidth, prevent potential congestion

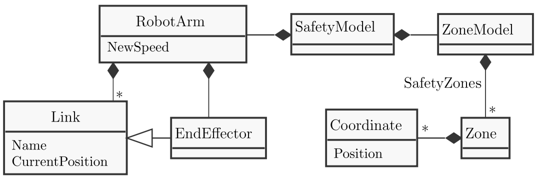
③ **Minimize computation effort:** enable faster reaction times

④ **Maximize interoperability:** not constrain models to be connected

Solution: Generation of model connectors

Idea: Use a small, dedicated language to specify endpoints (and necessary transformations) for a connection between two runtime models.





```
receive Link.CurrentPosition using
ParseState, Transform;
```

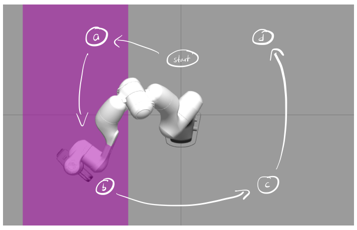
```
send RobotArm.NewSpeed using
CreateSpeedMessage, SerializeRobotConfig;
```

```
ParseState maps byte[] bytes to RobotState {:
  return RobotState.parseFrom(bytes);
:}
```

shared by both models

- Robot controller as external model controls robot
- Safety model describes position of robot parts and safety zones → sends speed updates as soon as one part is in a zone
- Application model steers movement of robot using a workflow → sends trajectory updates when one step in workflow is finished
- Recomputation only triggered if new input available (similar to reactive programming, unusual for RAGs)
- Generation with relational RAGs [1, 2]

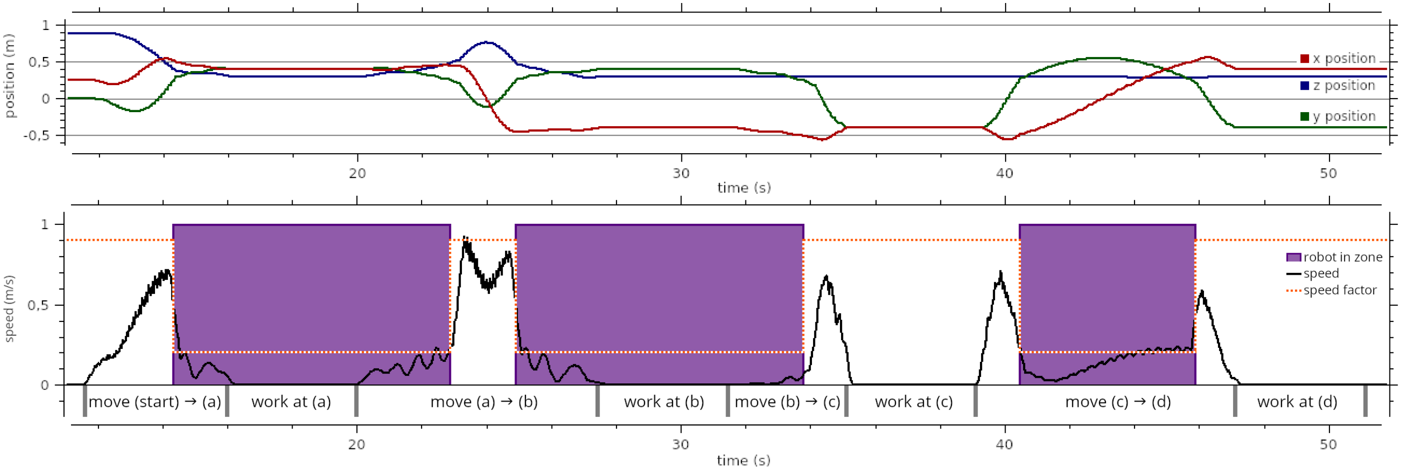
Evaluation with a robotic use case



← *View from above of the robot, showing its work plan and safety zone(s) in purple*

- With 18 and 28 LOC DSL-Code necessary 281 and 701 lines Java connection code is generated (safety model and application model, respectively) ①
- Over 38 000 position updates $\underset{③}{\rightarrow}$ 56 safety zone recomputations $\underset{②}{\rightarrow}$ necessary six speed updates sent
- External models connectable using standard communication protocols (e.g., MQTT) and any serialization format (e.g., Protocol Buffers) ④

coordinates of robot arm and when in safety zone



References

[1] Görel Hedin and Eva Magnusson. "JastAdd: an aspect-oriented compiler construction system". In: *Science of Computer Programming* 47.1 (2003), pp. 37–58. ISSN: 0167-6423. DOI: 10.1016/S0167-6423(02)00109-0.

[2] Johannes Mey et al. "Relational Reference Attribute Grammars: Improving Continuous Model Validation". In: *Journal of Computer Languages* 57 (Jan. 20, 2020), p. 21. ISSN: 2590-1184. DOI: 10.1016/j.cola.2019.100940.

Contact:

René Schöne	rene.schoene@tu-dresden.de
Johannes Mey	johannes.mey@tu-dresden.de
Sebastian Ebert	sebastian.ebert@tu-dresden.de
Uwe Aßmann	uwe.assmann@tu-dresden.de

Acknowledgments:

This work is partly supported by the German Research Foundation (DFG) as part of Germany's Excellence Strategy – EXC 2050/1 – Project 390696704 – "Centre for Tactile Internet with Human-in-the-Loop" (CeTI), the project "RISCOS" (project number 280611750) and the project "HybridPPS" (project number 418727532), and using tax money based on the budget approved by the Saxon state parliament for the project "PROSPER".