The EPOC Contracting Architecture

Steffen Stein, Moritz Neukirchner, Rolf Ernst

INSTITUTE OF COMPUTER AND COMMUNICATION NETWORK ENGINEERING

TECHNISCHE UNIVERSITÄT CAROLO-WILHELMINA ZU BRAUNSCHWEIG

Outline

- Motivation
- In-System Contracting
- EPOC Contracting Architecture
- Conclusion

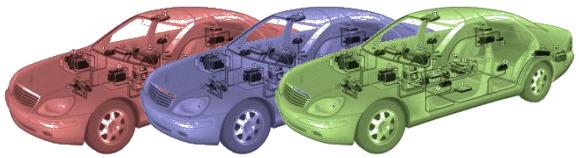
Motivation

- Already today updates for real-time systems are common practice.
- The system has to be verified for every update.

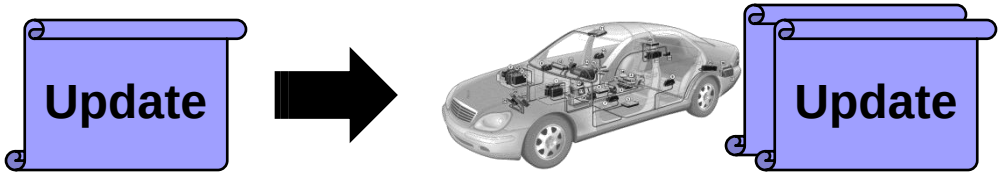
Motivation

Why do we need self-protection?
We have system level verification!

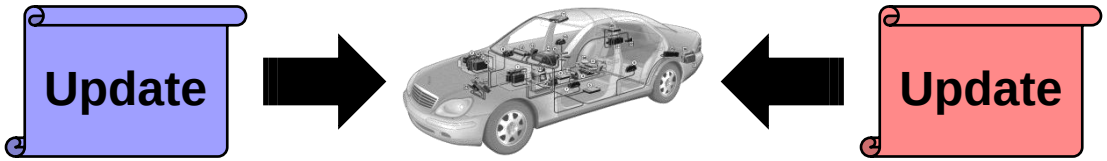
- Large number of variants (e.g. automobile) makes design-time verification costly



- Unknown update history of the specific system



- Software of different vendors running on the same system

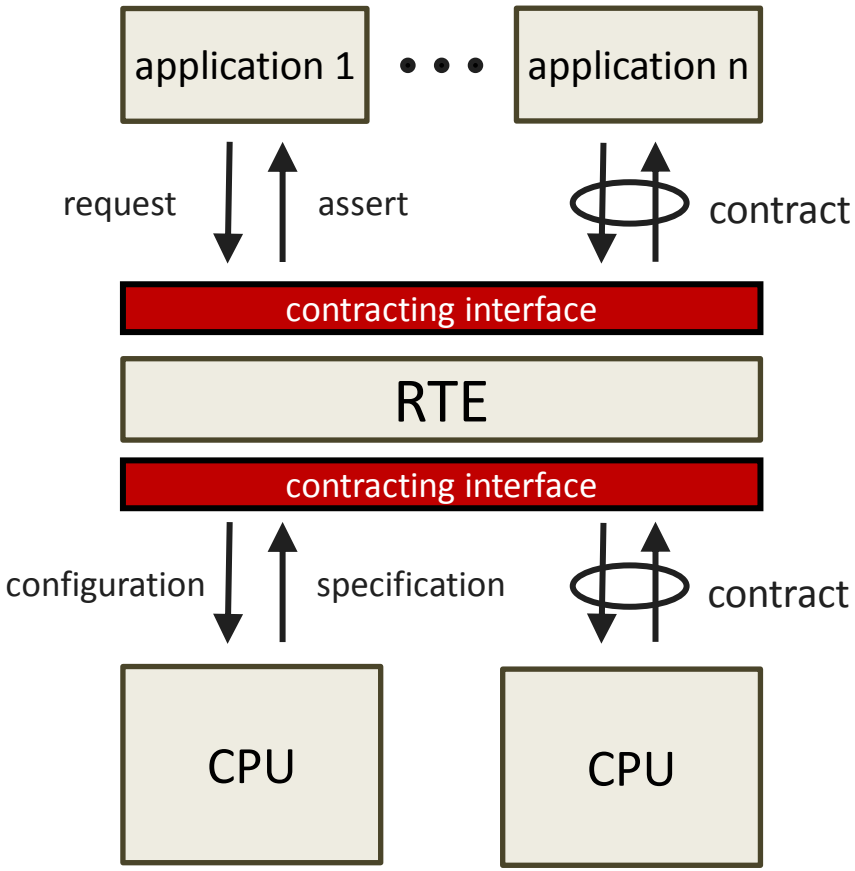


In-System Contracting

Approach:

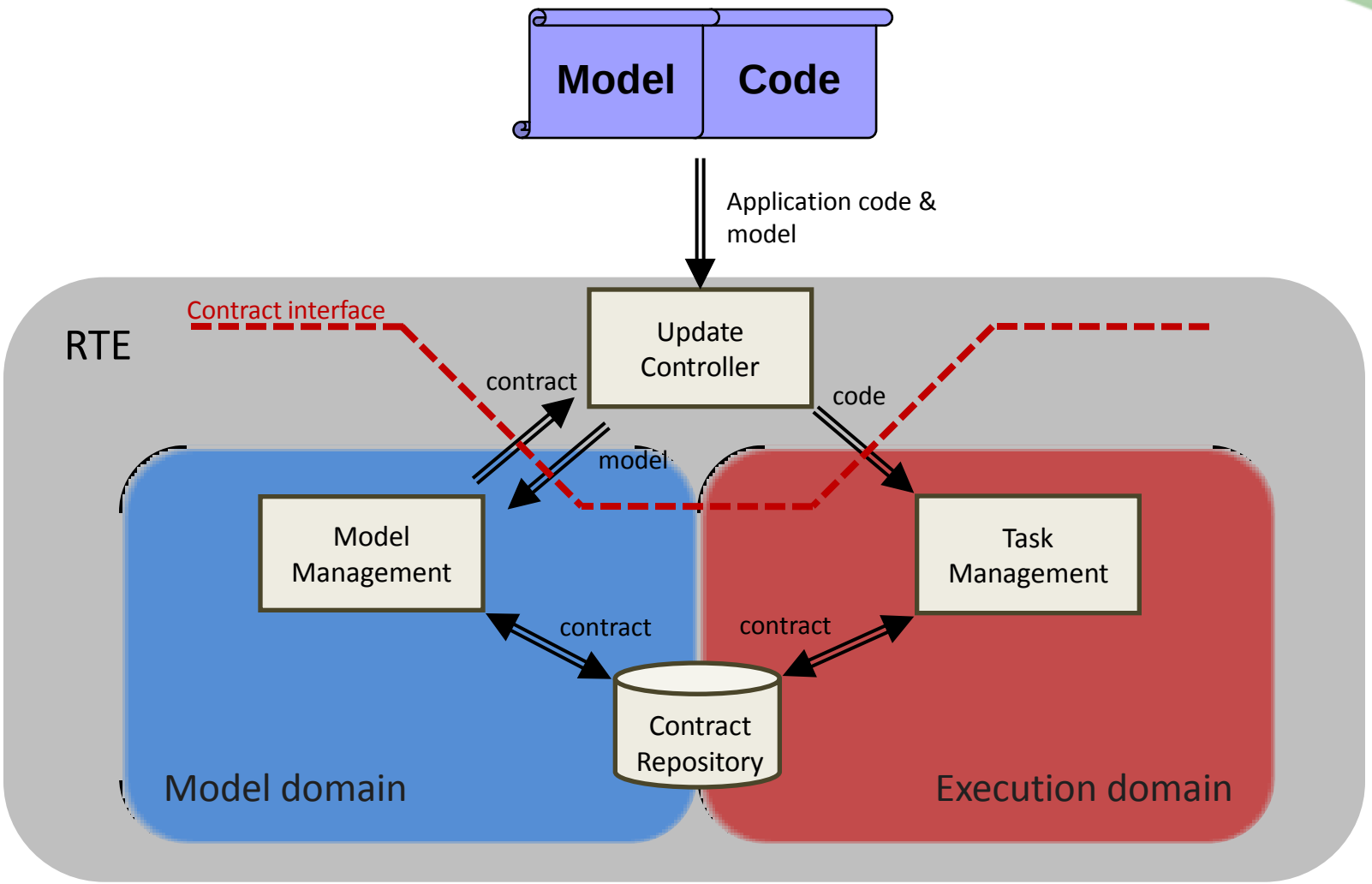
- In-System Contracting
  - Software has to declare its behavior, requirements and constraints prior to execution on the platform
  - The system verifies itself based on software description and platform capabilities
  - The system supervises adherence of the software to its descriptions

In-System Contracting



- Applications request Service from RTE
- RTE asserts properties of Applications
- Extend RTE by Contracting Layer
  - For applications
  - For platform
- Violation → Renegotiation

EPOC Contracting Architecture



Key Architectural Aspects

- Contracting towards Application and Platform
  - Decoupling of Platform and Functionality
  - Contract Information used for Monitoring

- Strict Separation of Model and Execution
  - Enables Model-Based Exploration without affecting execution

Advantages - Conclusion

- In-System Contracting allows to let the system verify itself
- Model-based verification can be performed independently of the application execution
- Application execution can be controlled and supervised using contract data

THANK YOU FOR YOUR ATTENTION