

Model Level Validation of Component Based Software for Distributed Embedded Systems

Dietmar Schreiner and Karl M. Göschka
 {d.schreiner,k.goeschka}@infosys.tuwien.ac.at
 Vienna University of Technology



Abstract

When building a component based application for distributed embedded systems, its overall behavior depends not only on the contracts applying to the components and their interfaces, but even more so on explicit as well as implicit connectors emerging from component composition, deployment and interaction. We introduce explicit connectors at model level to perform validation of component and contract composition beyond simple interface matching. Moreover, we provide a classification of typical component connectors to simplify application development for distributed embedded systems. This avoids the need of extensive knowledge of communication subsystems and the existence of any heavy weight middle-ware.

Model Driven Development

Composition

- 1 ● Component contracts
- Interface contracts
- Abstract connectors

Specification of Deployment

- 2 ● ECU constraints and contracts
- Bus constraints and contracts

Model Transformation

- 3 ● OTS connector selection
- Emerging contracts
- Glue code generation
- Model validation

Components and Connectors

Connectors

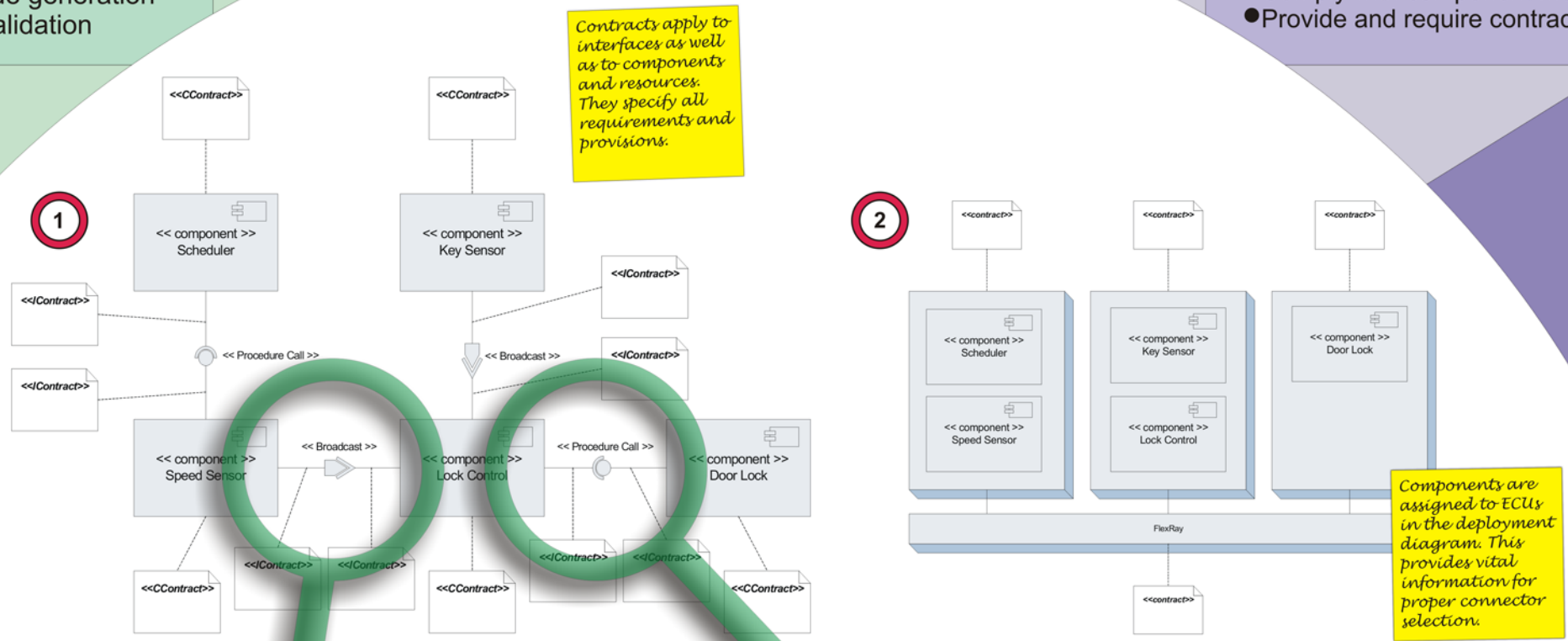
- Encapsulate communication and interaction logic
- Special purpose components

Contracts

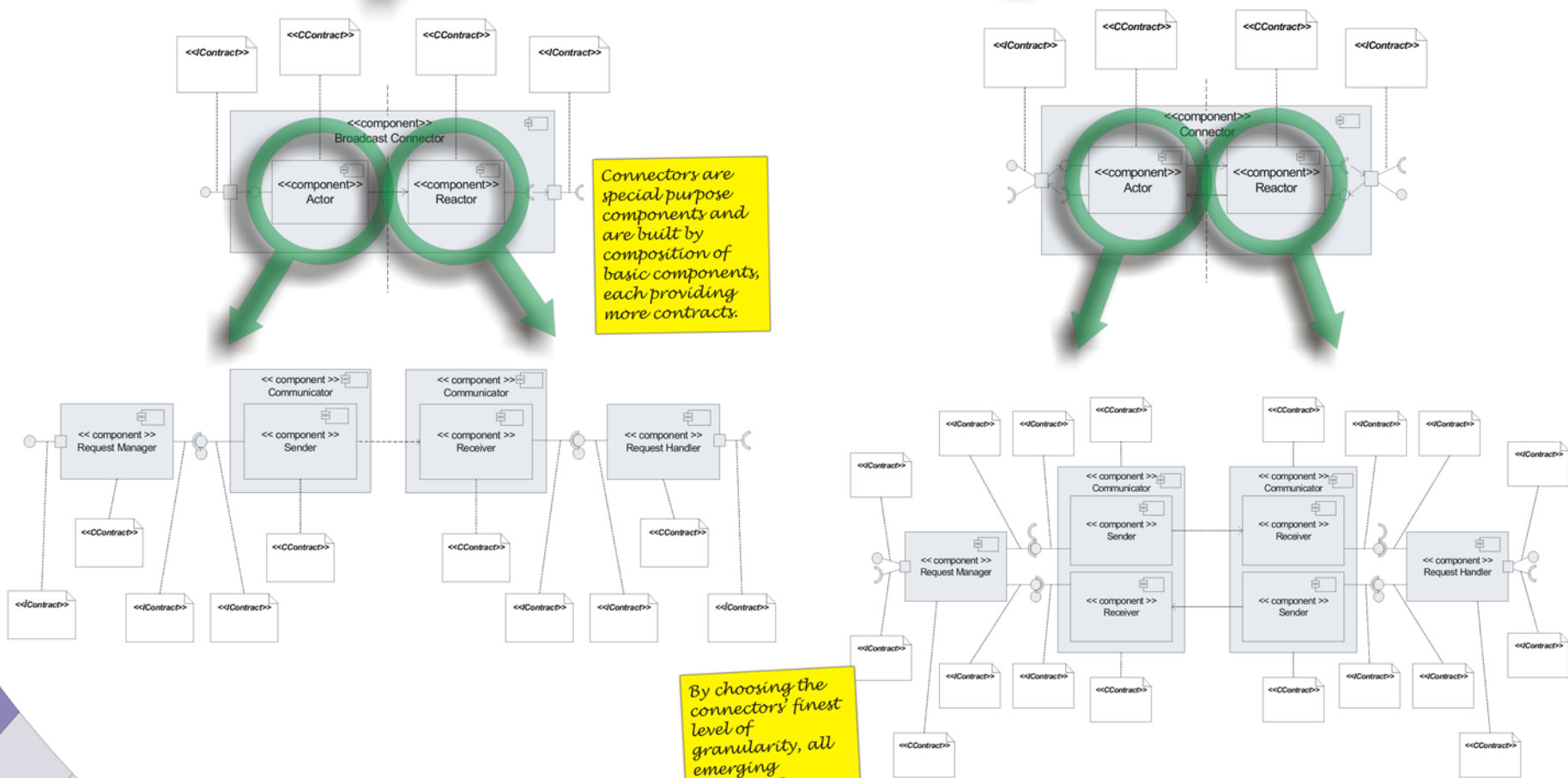
- Component Contracts
- Interface Contracts
- Behavior Protocols

Components

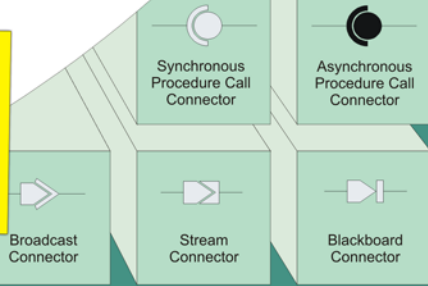
- Composable units of execution with well defined contracts
- Comply with component model
- Provide and require contracts



Transformation



The connectors' communication style, as seen by the application developer, is specified using this predefined symbols.



<http://www.infosys.tuwien.ac.at/compass>