itemis

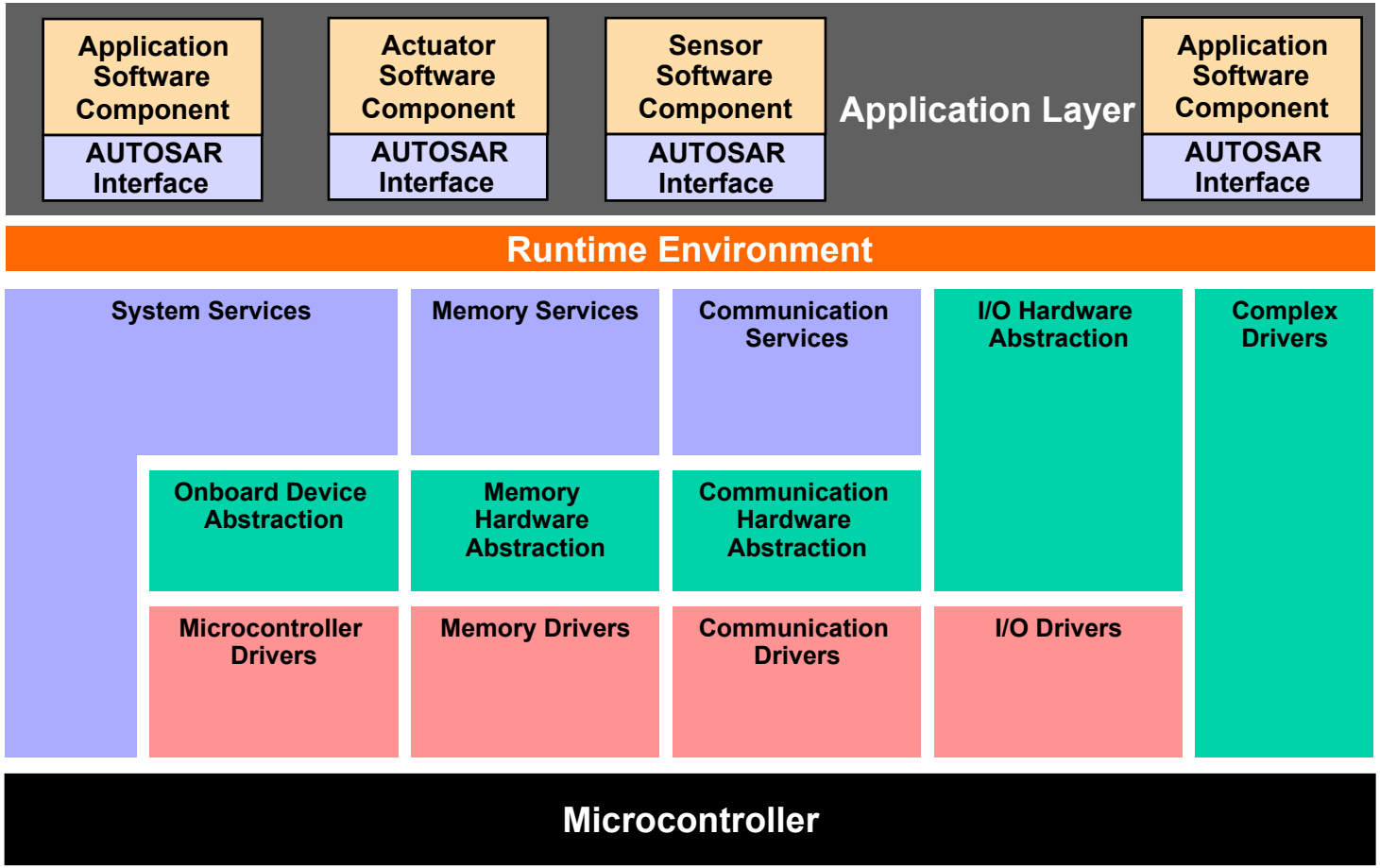
Model-based Automotive Software Development using Autosar, UML, and Domain-Specific Languages

Dr. Alexander Nyßen

Embedded World 2013, Nürnberg

AUTOSAR (AUTomotive Open System Architecture)

Architectural modeling standard and reference architecture



Composed of images from: [http://www.autosar.org/download/conferencedocs/03_AUTOSAR_Tutorial.pdf]

What is Software Architecture?

[Bass, Clements, Kazman (1997): Software Architecture in Practice]

„The **software architecture** of a program or computing system is the **structure or structures** of the system, which comprise software **components**, the **externally visible properties** of those components, and the **relationships** among them.“

What is Software Architecture?

[Bass, Clements, Kazman (1997): Software Architecture in Practice]

- Structure or structures of a system
  - Structural aspects: Decomposition, Uses, Layered, Modules
  - Behavioral aspects: Processes, Concurrency, Shared Data
  - Allocation aspects: Deployment, Artifacts, Work assignment
- Externally visible properties
  - „Assumptions other components can make of a component, such as its provided services, performance characteristics, fault handling, shared resource usage, and so on.“

Implications of Software Architecture Definition

[Bass, Clements, Kazman (1997): Software Architecture in Practice]

> "Every computing system with software has a software architecture. […] It does not necessarily follow that the architecture is known to anyone."

➡ The architecture has to be properly documented.

> "Unfortunately, an architecture can exist independently of its description or specification."

➡ The documentation has to be consistent.

> "The behavior of each element is part of the architecture."

➡ The documentation has to cover all relevant aspects.

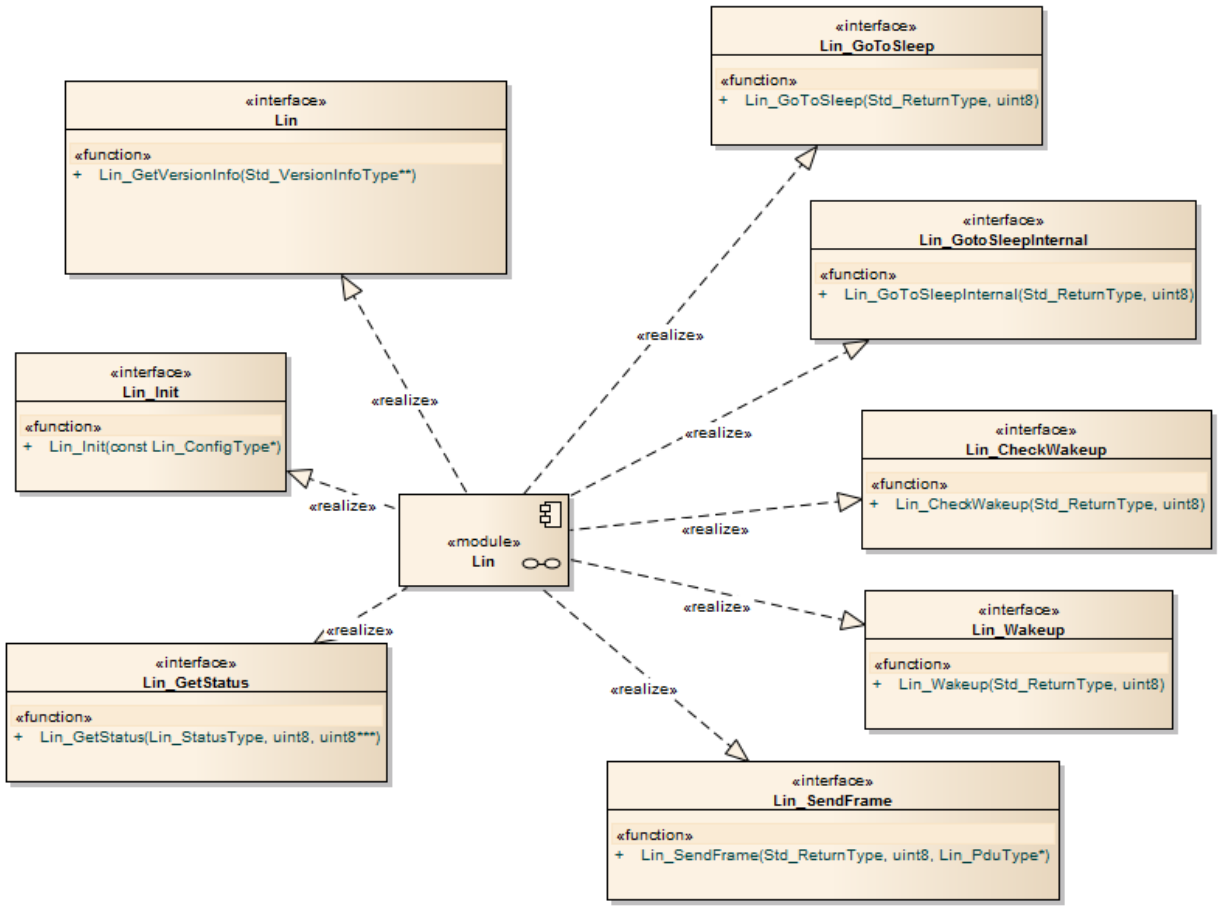
Autosar-based Software Architectures

Different abstraction levels for BSW and SW-C

- Autosar employs different concepts within specification of BSW and SW-C:
  - BSW: modules & realized/used (operation-based) interfaces
  - SW-C: components with ports and (different kinds of) interfaces, hierarchical decomposition in terms of composition types and component prototypes
- Accordingly, Autosar proposes different means to document BSW and SW-C:
  - UML for BSW
  - Autosar Graphical Notation for SW-C

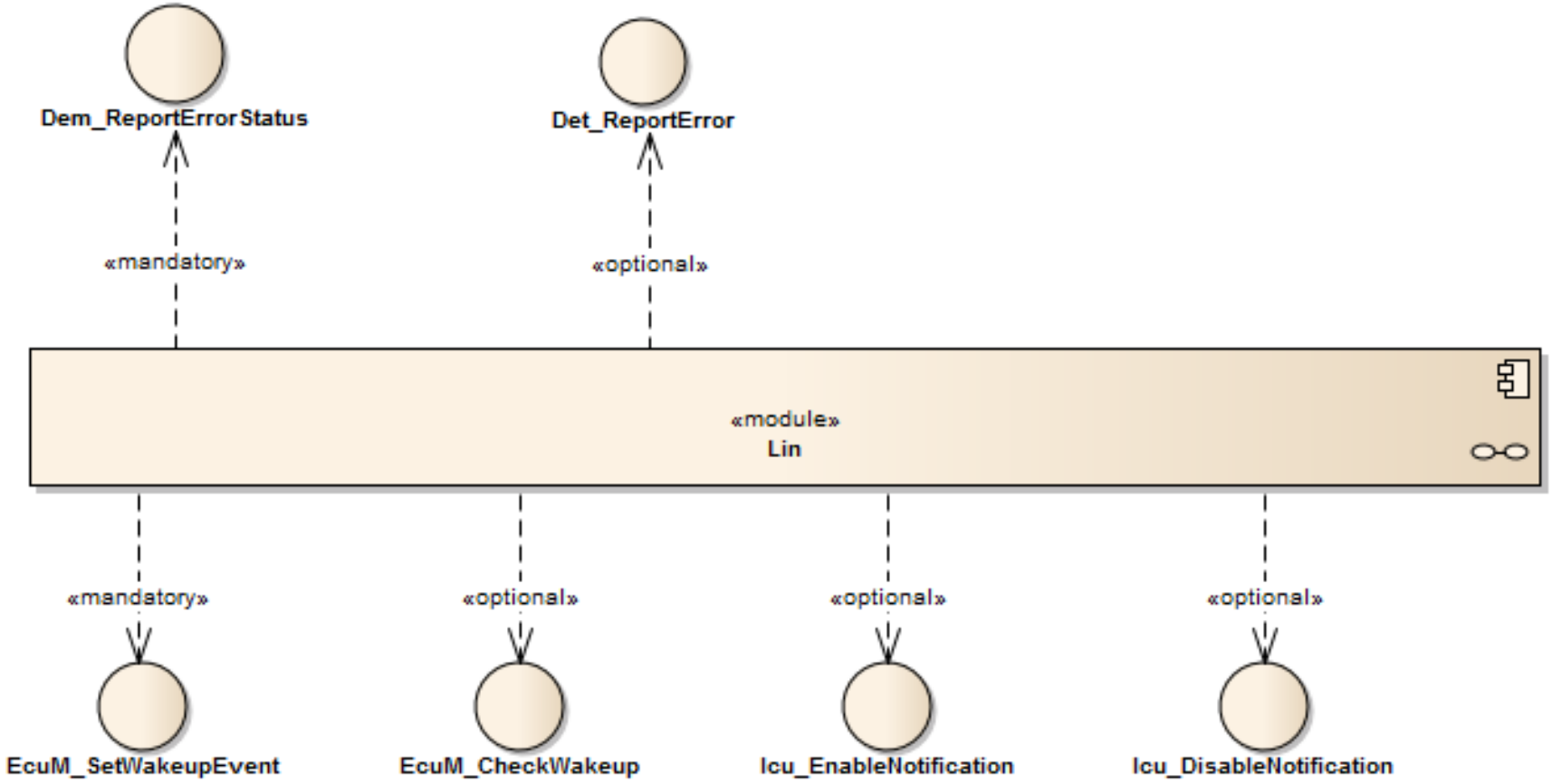
BSW & UML – Structural Aspects

Example: BSW module and realized interfaces



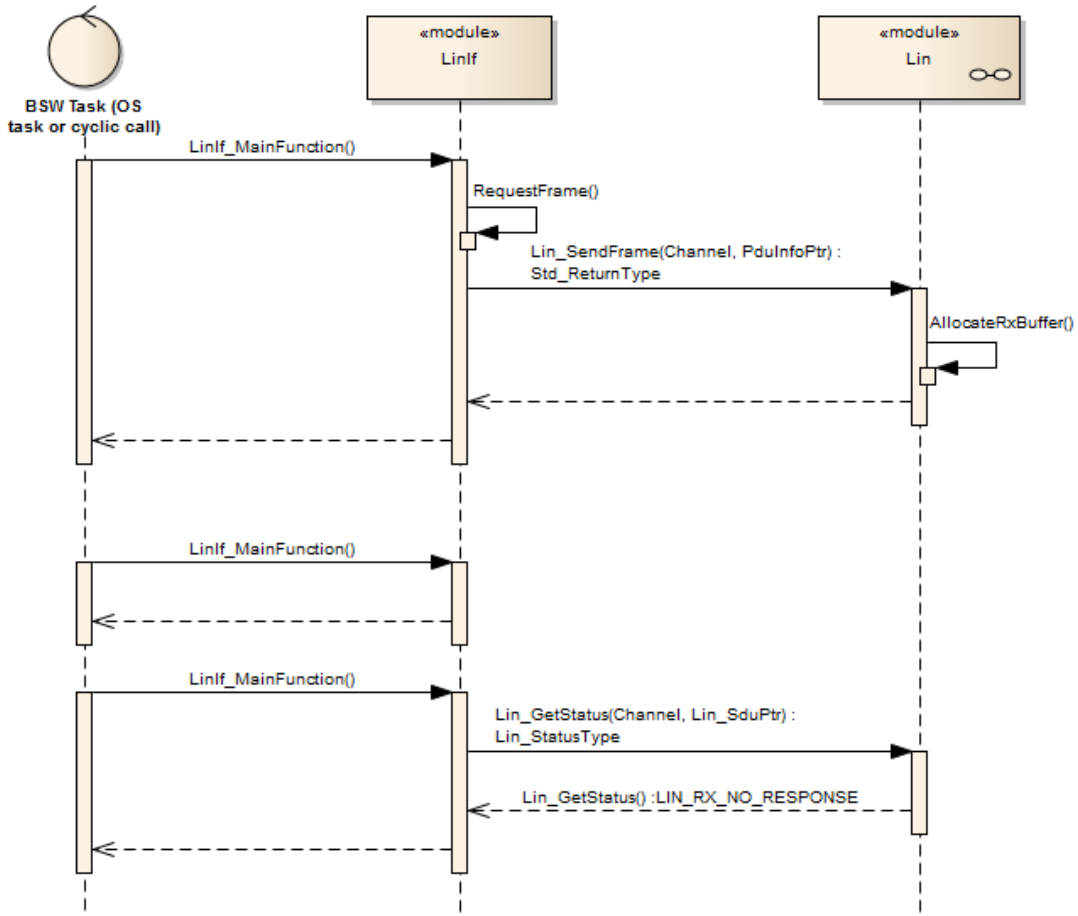
BSW & UML – Structural Aspects

Example: BSW module and required interfaces



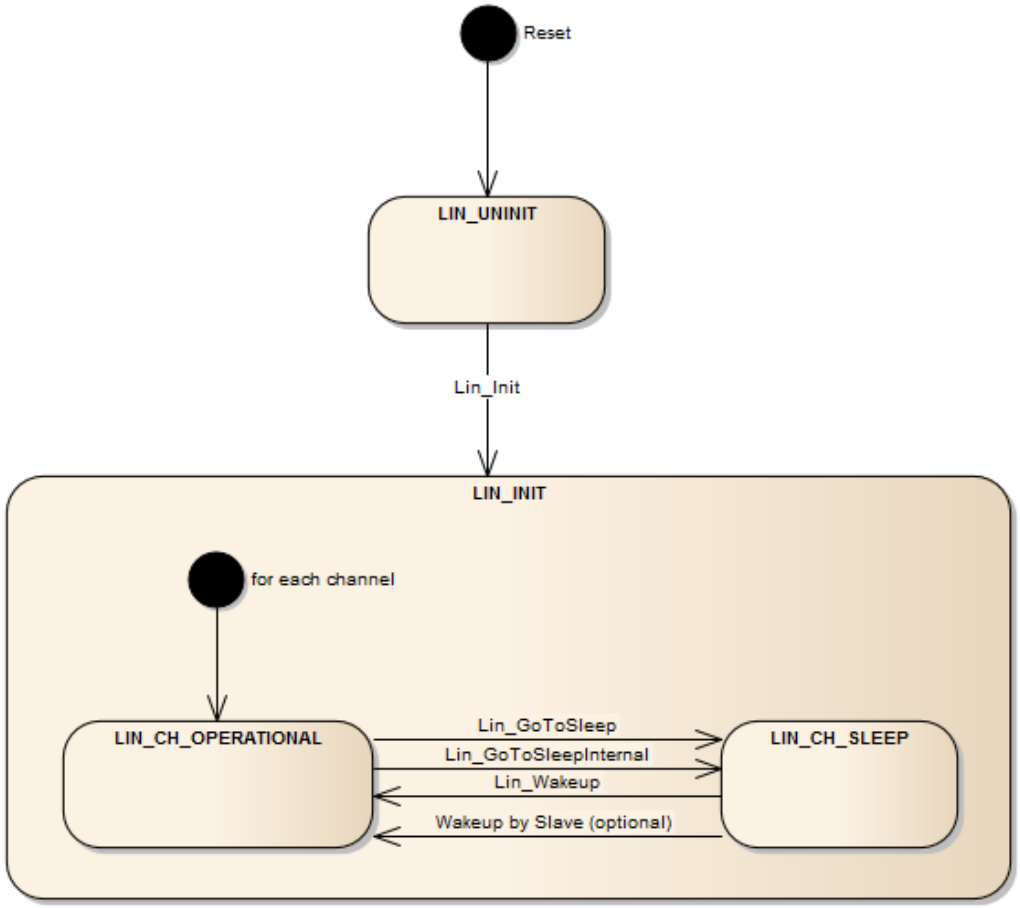
BSW & UML – Behavioral Aspects

Example: BSW module interaction



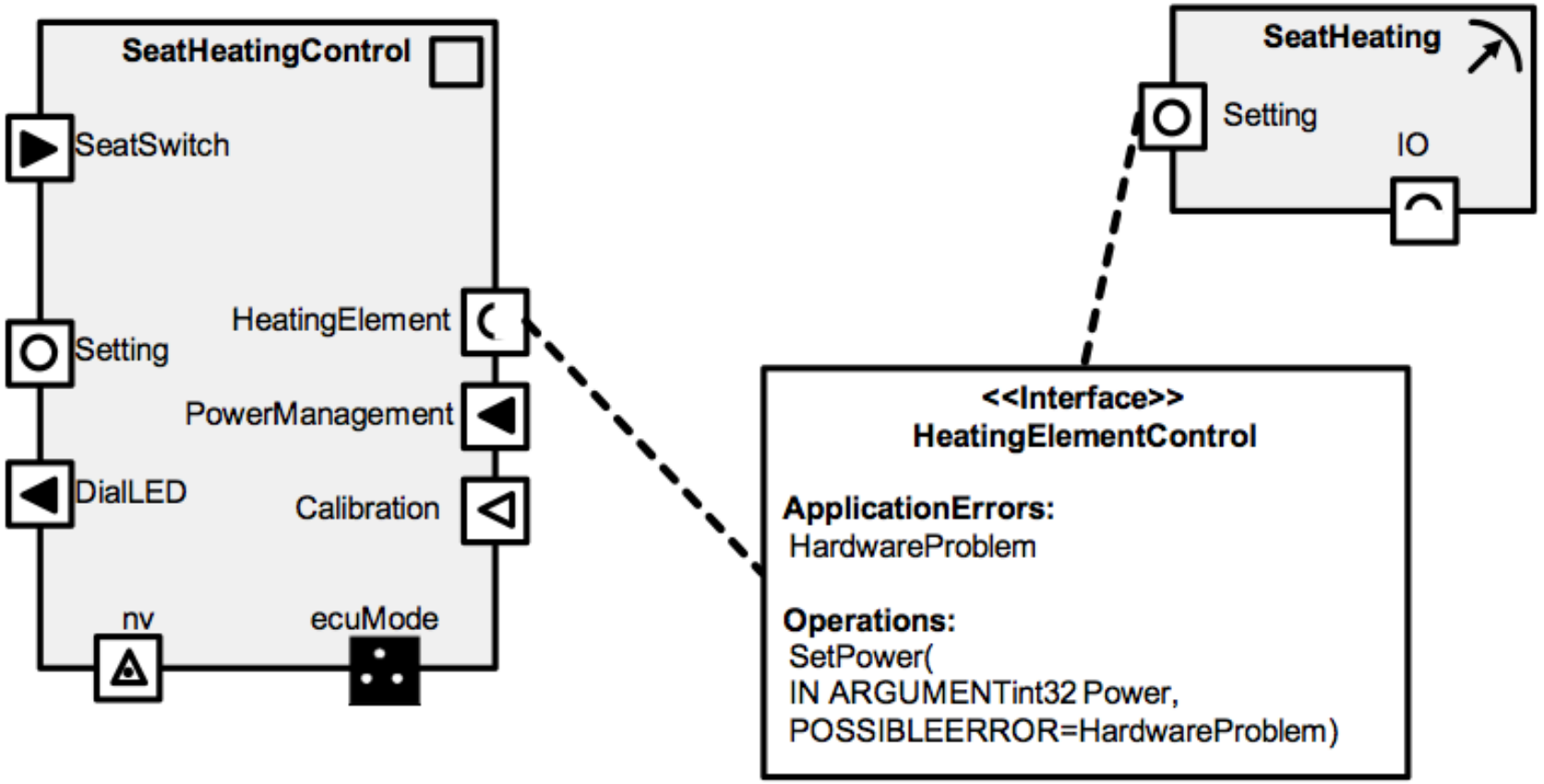
BSW & UML – Behavioral Aspects

Example: internal BSW module behavior



SW-C & Autosar Graphical Notation

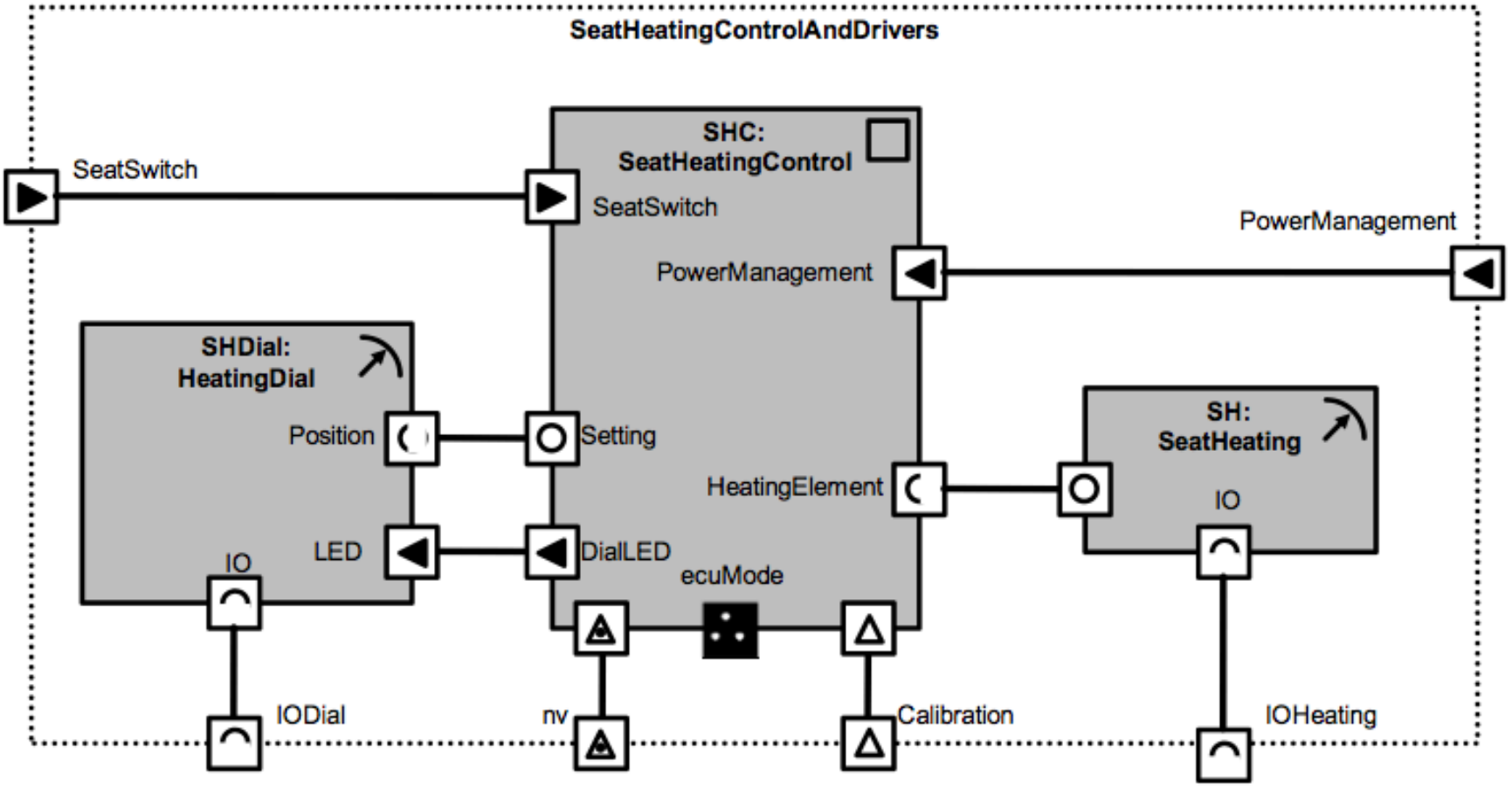
Example: Specifying interfaces of SW-C



Source: [Specification of Virtual Functional Bus, V2.2.0, R4.0 Rev.3; www.autosar.org/download/R4.0/AUTOSAR_EXP_VFB.pdf]

SW-C & Autosar Graphical Notation

Example: Specifying composition of SW-Cs



Things that Autosar does not address…

... but that still have to be addressed.

- AUTOSAR concentrates on the structural aspects of the application software architecture and mostly omits behavioral aspects.
- AUTOSAR only covers the definition of architectural aspects and leaves the transition to detailed design pretty much open.

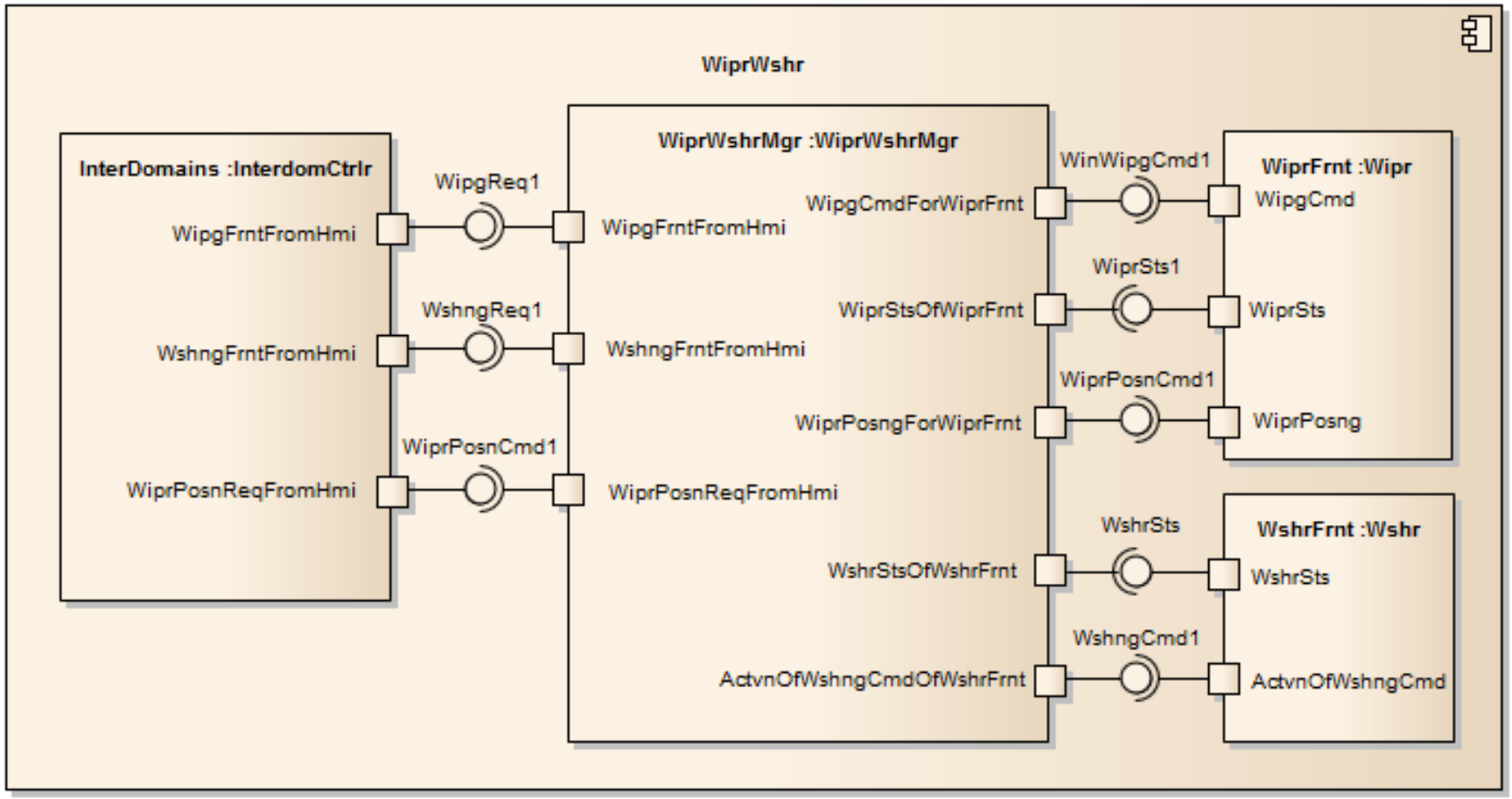
SW-C Architectural Design & UML

UML may be used to augment architectural design of SW-C

- UML can support by augmenting models with behavioral specifications (interactions, protocol state machines).
- To achieve this, a certain subset of the “structural” information has to be represented in both worlds, because UML behavior is defined on top of structure.

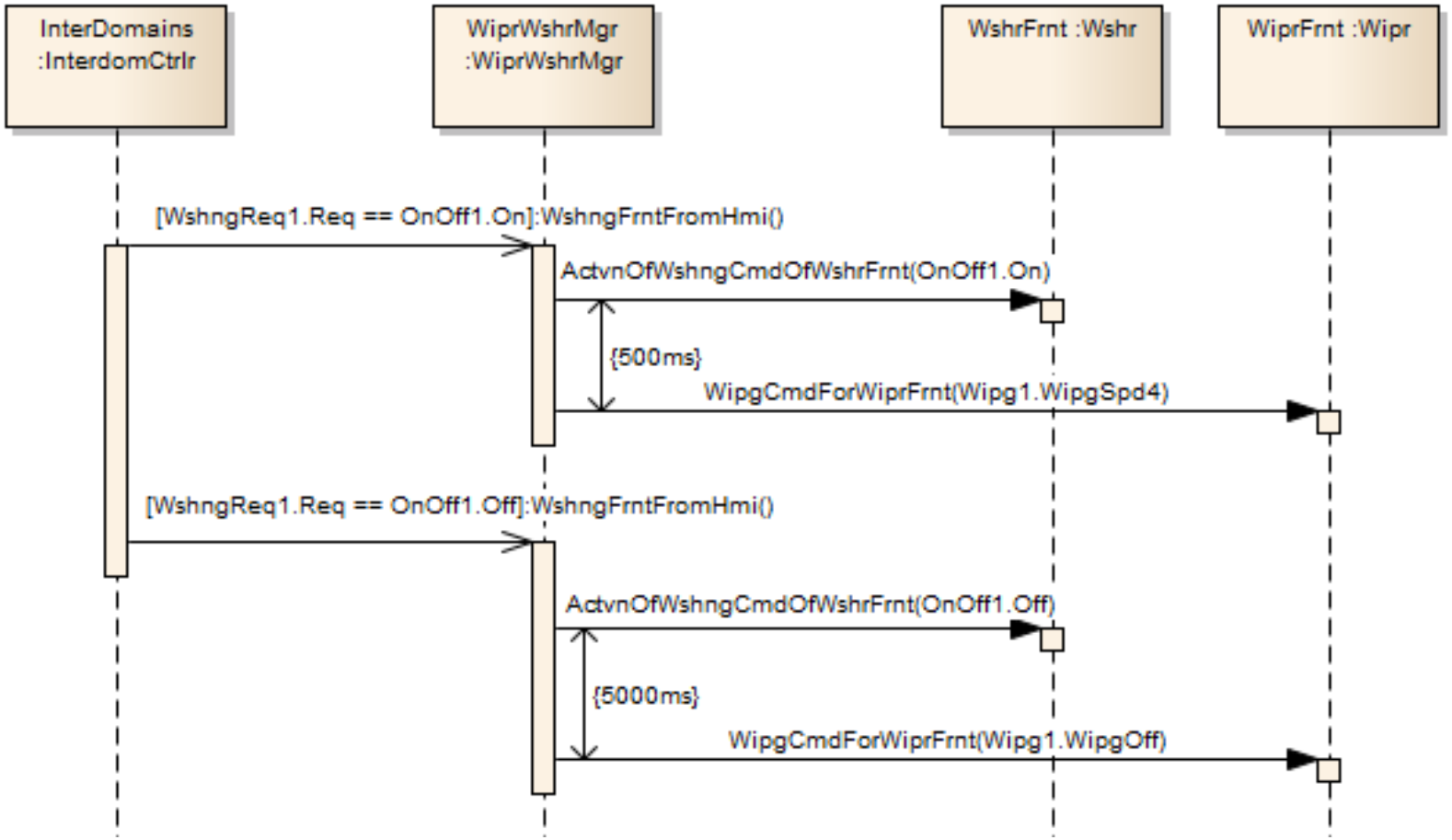
SW-C & UML - Structural Aspects

Example: internal decomposition (excerpt) of a composition type



SW-C & UML – Behavioral Aspects

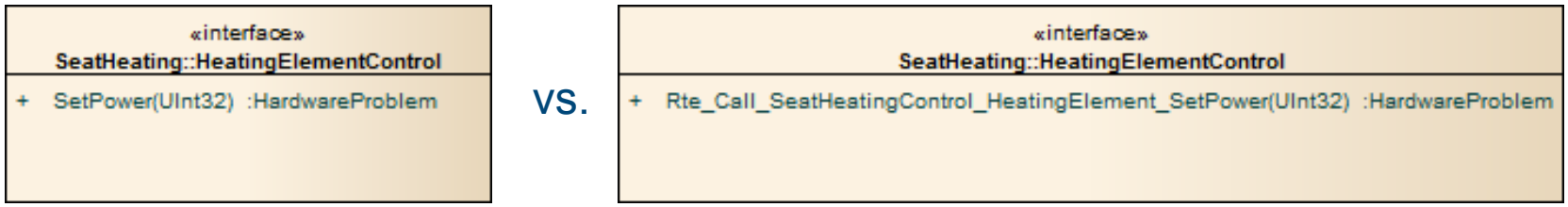
Example: interaction between component prototypes



Implications of using UML for Autosar (I)

UML has to be used on the “right“ level of abstraction

- Using the same level of abstraction as with BSW is not appropriate for SW-C because:
  - Transformation between UML and Autosar should be performed on the same level of abstraction (traceability!!)
  - Reduction of abstraction level can be encapsulated within code generators (automation!!)
  - Models would become cluttered if they are “spammed” with code abstraction details



Implications of using UML for Autosar (II)

Need for a UML-Profile and concise modeling methodology

- To cover certain aspects of an AUTOSAR architecture, a UML profile is needed.
  - Note that an AUTOSAR-UML-Profile was provided with early versions of the standard but has not been further maintained.
  - In case UML is just needed to augment AUTOSAR models (behavior, transition to detailed design), the profile's scope may however be much restricted.
- UML is widely applicable, so in addition, a concise modeling methodology and conventions have to be defined (and implemented in tooling)

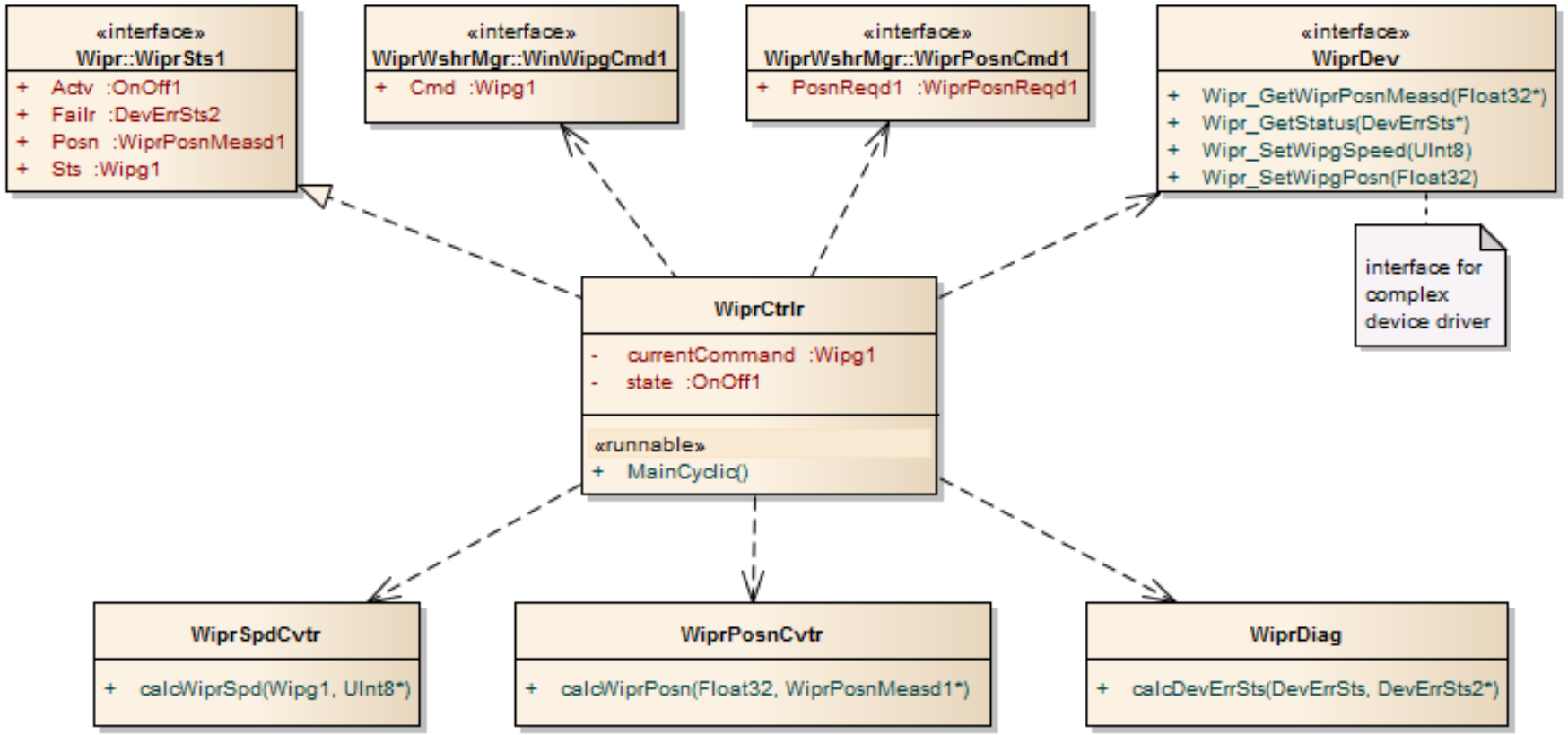
SW-C Detailed Design & UML

UML may also be used for Detailed Design of SW-C

- If software architecture is already represented, the detailed design of an atomic SW-C may also be modeled using UML:
  - Class diagrams can be used to model the structural decomposition in terms of implementation modules (.h/.c)
  - Behavior diagrams may be employed as in BSW (sequence diagrams, state machines, etc.)

SW-C Detailed Design & UML

Example: Specifying internal decomposition of atomic components



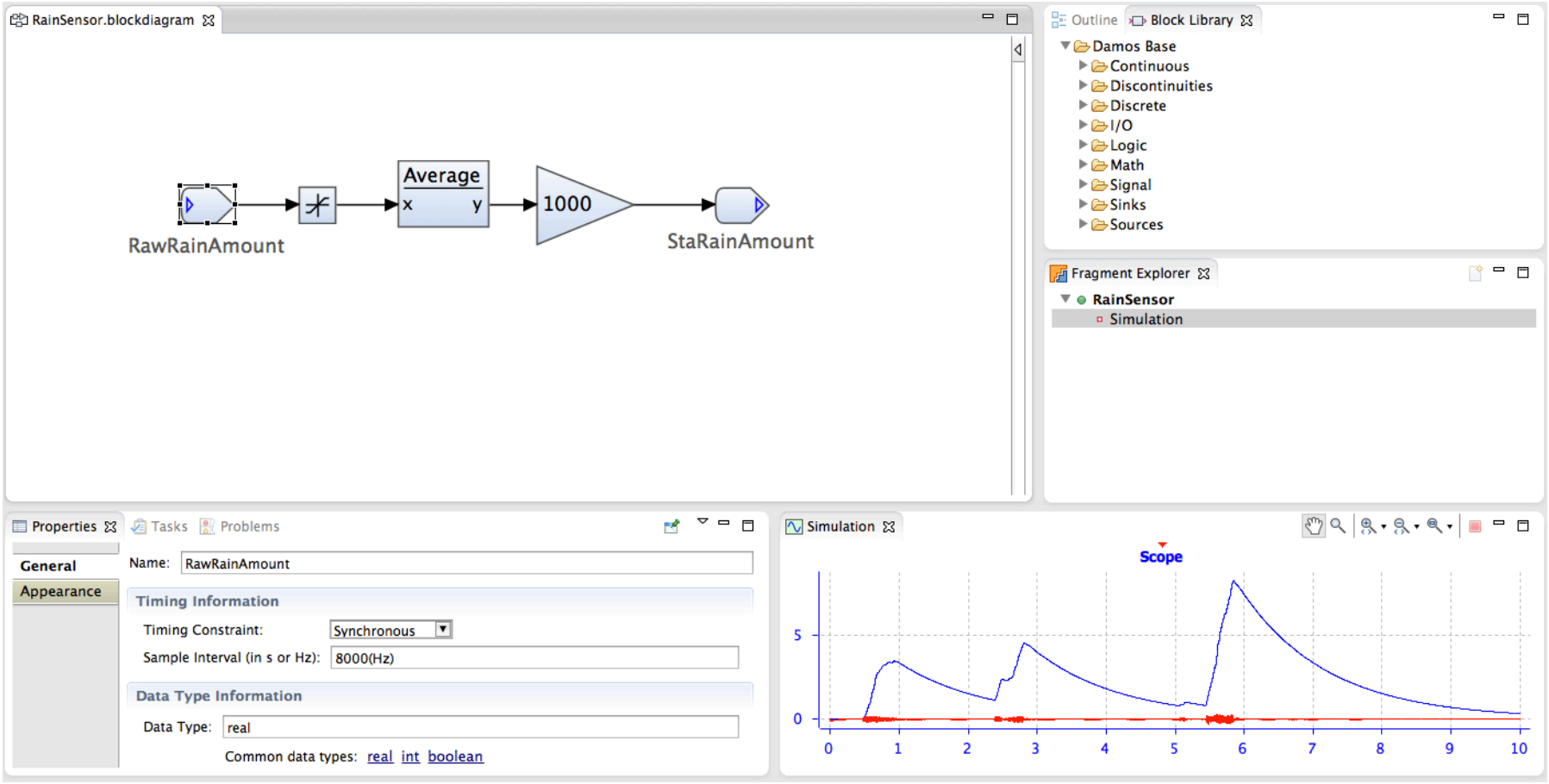
When UML is not the adequate way to go…

… DSLs might help

- Classical (implementation-)module-based decomposition of atomic SW-C may not always be the best way to go.
  - Continuous controller components may be best described by means of a dynamical systems model.
  - State-based controller components may be best described by means of a state machine.
- UML does not cover control theory, and UML tools are not capable of appropriate simulation and code generation for state machines either.
  - Using Domain-Specific Languages and dedicated DSL tools might be the better option in such cases.

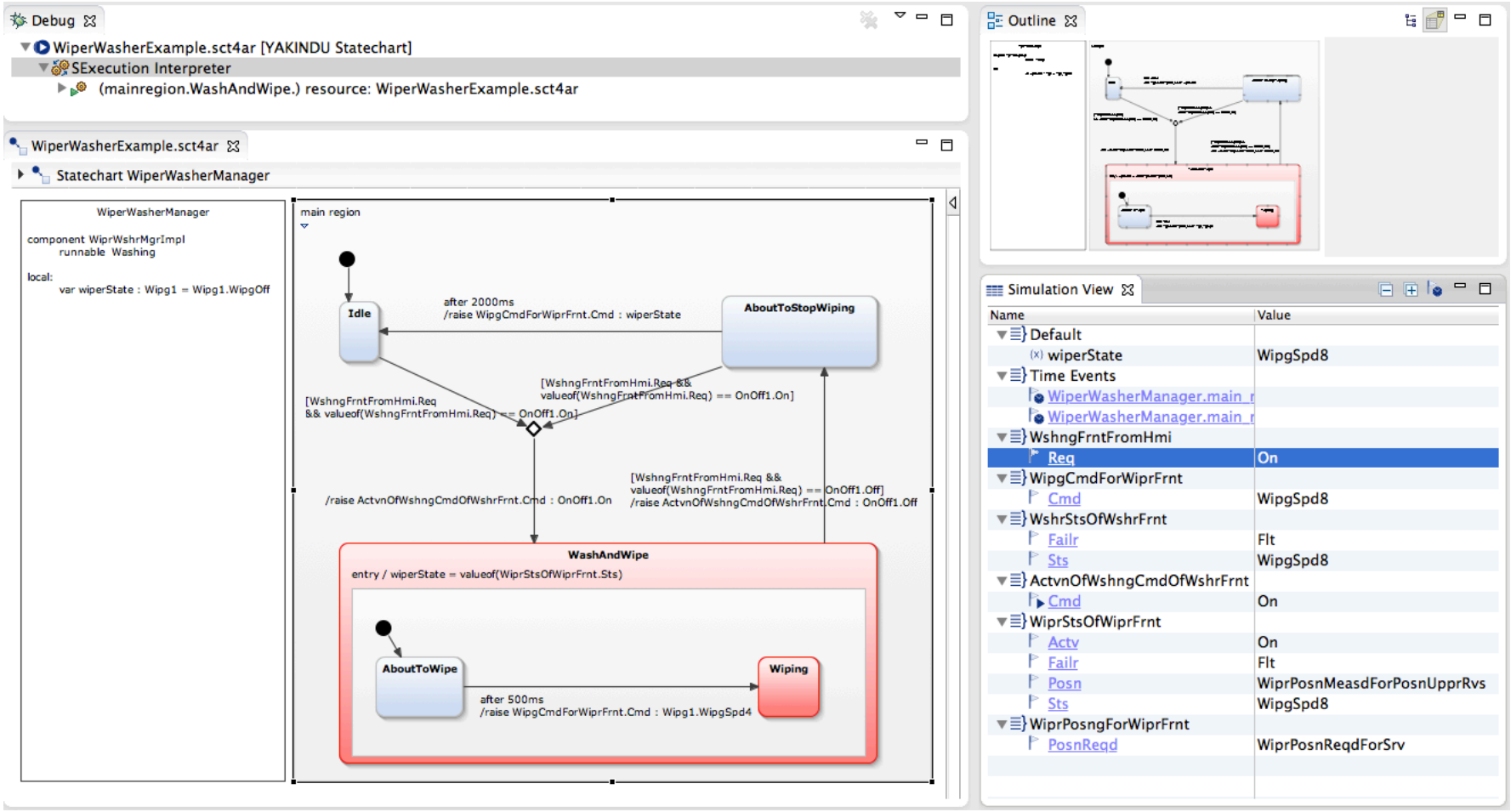
Using Domain-Specific Languages (I)

Example: Eclipse Damos



Using Domain-Specific Languages (II)

Example: Yakindu Statechart Tools (SCT)

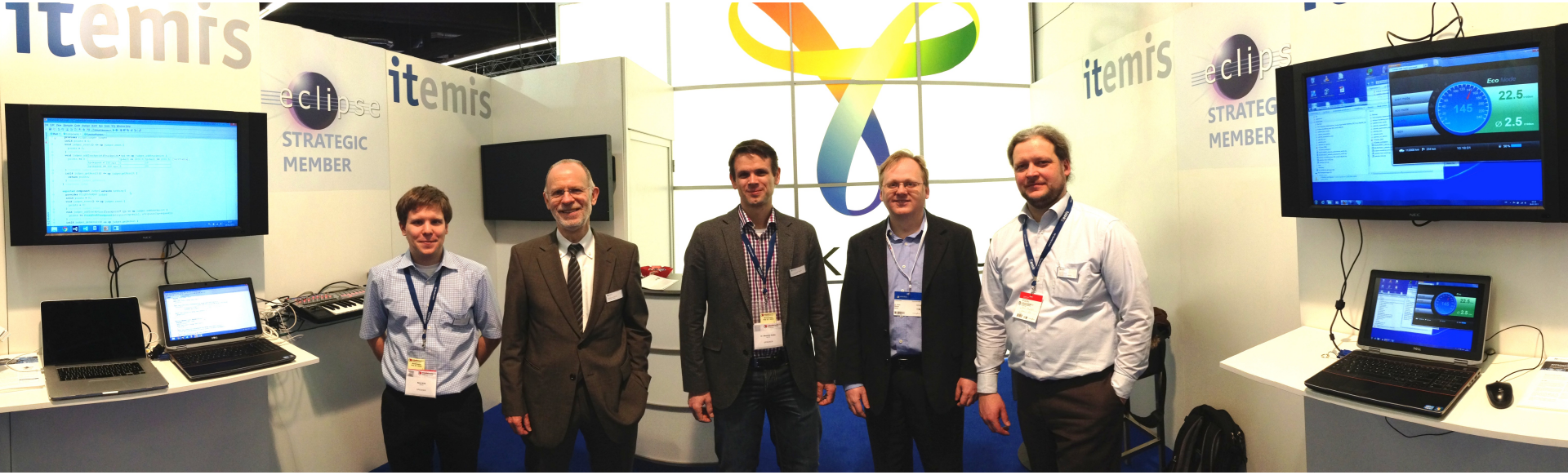


Summary & Conclusion

Autosar, UML, and DSLs

- UML may be employed to augment Autosar architectural models with missing details (behavior)
- UML may be used to specify/document the detailed design of SW-C
- Domain-Specific Languages (DSL) may alternatively be used for detailed design as well.
  - Customized solutions can also be realized using open-source technologies, e.g. Eclipse Damos or Yakindu Statechart Tools

TOOLS AND METHODS FOR SEAMLESS SOFTWARE AND SYSTEMS ENGINEERING



WE LOOK FORWARD TO SEEING YOU:

HALL 4, STAND 106