Parameterization of Fail-Operational Architectural Patterns

Dulcineia Penha (dulcineia.penha@esk.fraunhofer.de), Gereon Weiss
Fraunhofer ESK

15th April 2015, 30th ACM/SIGAPP Symposium On Applied Computing (SAC 2015 Salamanca, Spain)

Safety-Critical Networked Embedded Systems (SCNES)

Characteristics of today's SCNES
- dense amount of software deployed
- pushing borders on **safety and reliability**

During development engineers must
- deal with **fail-operational requirements**
- recur to redundancy, monitoring, special shutdown procedures, etc.

Self-adaptation
- fulfill fail-operational requirements
- enable optimized resource utilization

❖ **Challenge** → transfer knowledge between several design levels

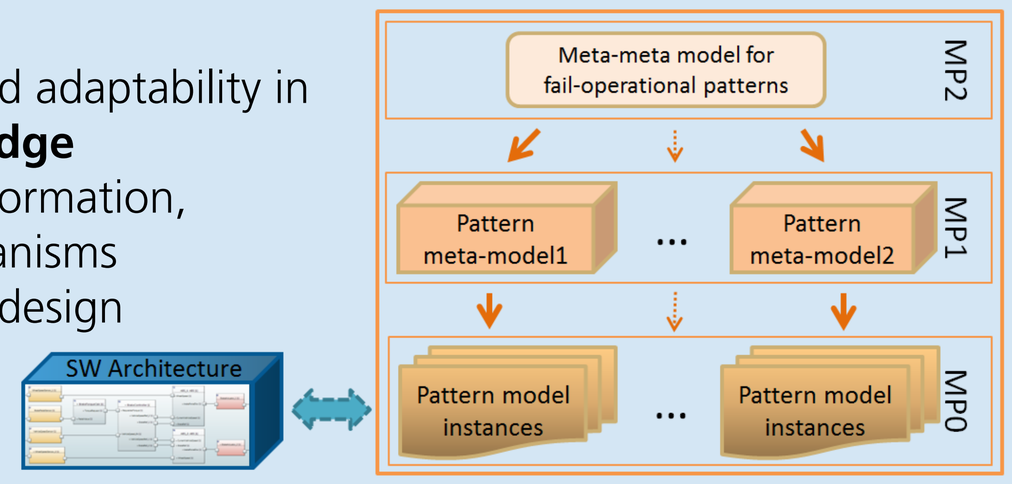
❖ **Lacking support for engineers** → during system development to apply fail-operational and adaptability concepts

Fail-Operational (FO) Architectural Patterns

Support engineers to realize safety, reliability, and adaptability requirements via patterns
- realization of patterns using models
  - in contrast to the classic definition assuming textual form
- integration of adaptability and FO concerns into model-based development

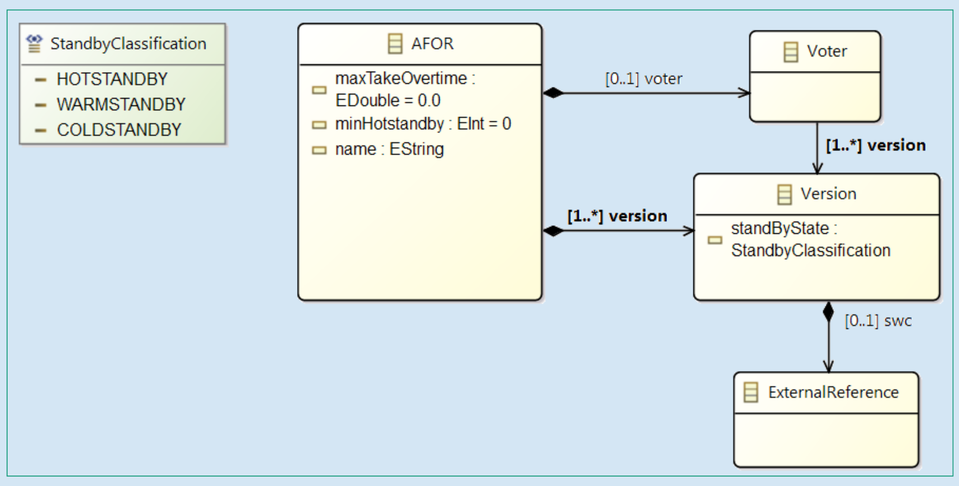
Patterns defined as meta-model extensions and instantiated at the architectural level
- better management of system-wide fail-operational and adaptability requirements

Capabilities
❖ Arguments for safety and adaptability in form of **stored knowledge**
❖ **Facilitate tracing** of information, requirements and mechanisms
→ Avoid failures during design



Overview of architectural pattern definition and meta-patterns levels

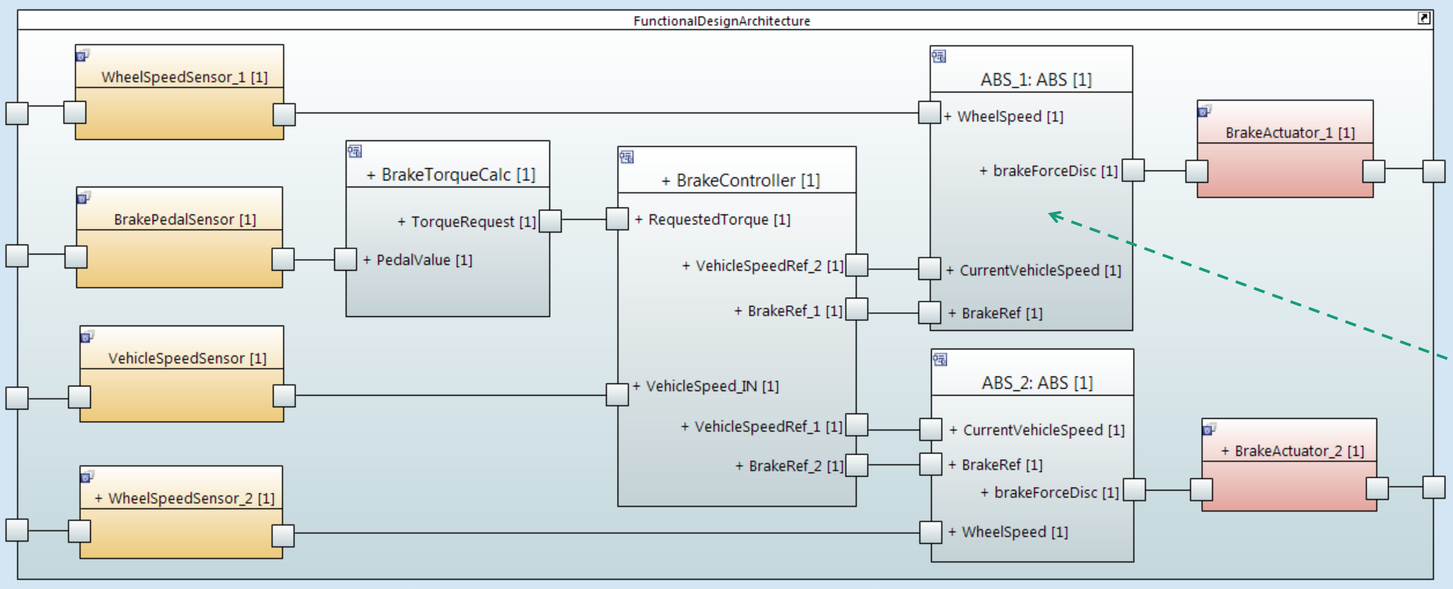
Example: Adaptive Fail-Operational Redundancy (AFOR) Pattern



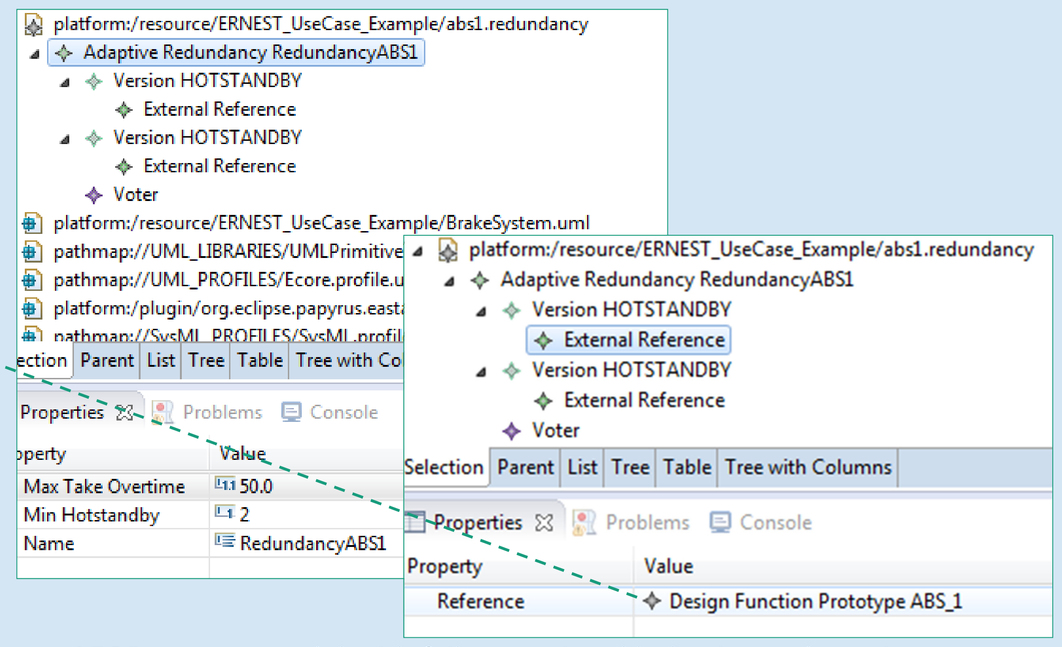
Meta-model for the AFOR pattern

- ***maxTakeOverTime***: maximum time interval in which a standby version has to take over in case of a failure
- *N* redundant versions
  - ***standbyClassification****:* hot-standby, warm-standby, cold-standby
- External Reference:
  - reference to software components in the architecture
  - enable **tracing and coupling** of the fail-operational requirements with the architecture
  - **heterogeneous or homogeneous versions**
- Voter

Automotive Case-Study (Brake-System)



SW Architecture in EAST-ADL – Functional Design Architecture



AFOR pattern model with fail-operational behavior and requirements, inclusive external reference to the SW Architecture

Scenario
- Redundancy strategy because of the high risks imposed by a failure in the ABS components
- Depicted: AFOR instantiated for ABS_1
  - *FTTI* = 50 milliseconds → hot standby
  - two hot-standby versions of ABS_1
    → reference to the individual redundant component in the software architecture
  - Voter

Advantages of the approach
❖ Reuse for different applications
❖ Enrich the SW architecture
❖ Machine readable
❖ Enable automatic generation of more detailed SW architectures
❖ Allow traceability from requirements
❖ Integrate well into modern MDD methods

Conclusion

- Engineers are supported to assure that safety requirements are considered at the architecture level
  - Explicit modeling of FO-related information and adaptive behavior
- The effort to develop adaptive SCNES is reduced by utilizing FO architectural patterns for general and reoccurring safety-relevant mechanisms
- FO patterns are machine readable, allow automatized approaches, and support traceability towards requirements

Acknowledgements: The research leading to these results has received funding from the European Commission within the Seventh Framework Programme ([FP7/2007-2013] [FP7/2007-2011]) as part of the SafeAdapt project under grant agreement number 608945.