

SAVE ORCA

Formal Modelling, Safety Analysis and

Verification of Organic Computing Applications



DFG SPP 1183

Goal

Integrated design framework for the development of highly reliable and adaptive Organic Computing applications

Scope

Highly dynamic resource-flow systems with self-x properties

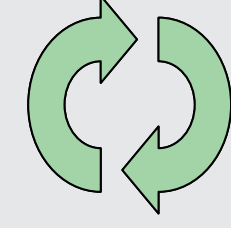
Challenges

- Decentralisation
- Unpredictability vs. Dependability
- Engineering self-x

Underlying Theory: Restore Invariant Approach

1. Behavioural corridors are described by a set of invariants. Organic Computing system monitors invariants

2. Temporary violation of invariants triggers self-x phase for restoring the invariants



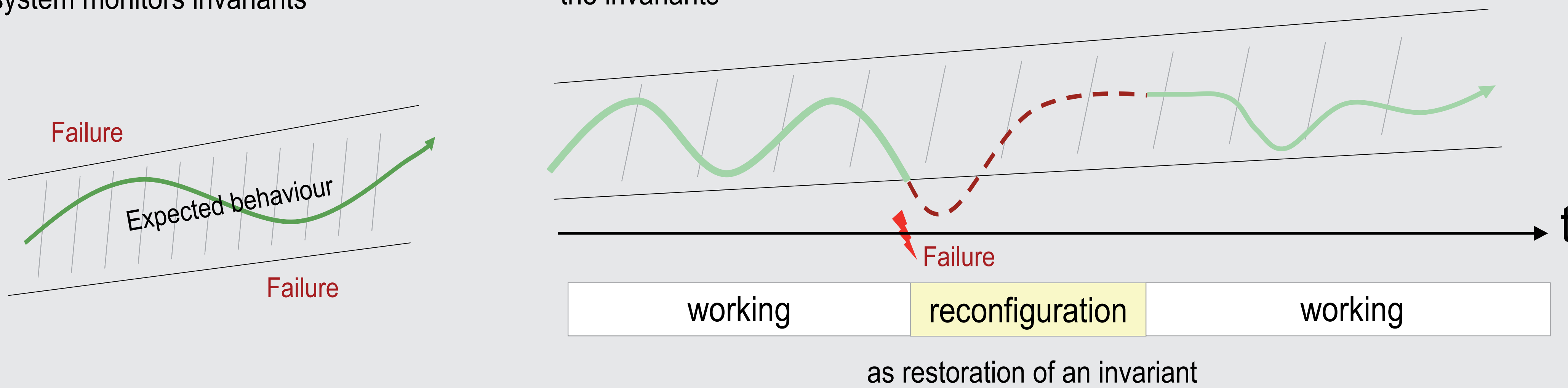
3. OC system preserves invariants as long as possible

4. Sometimes necessary:

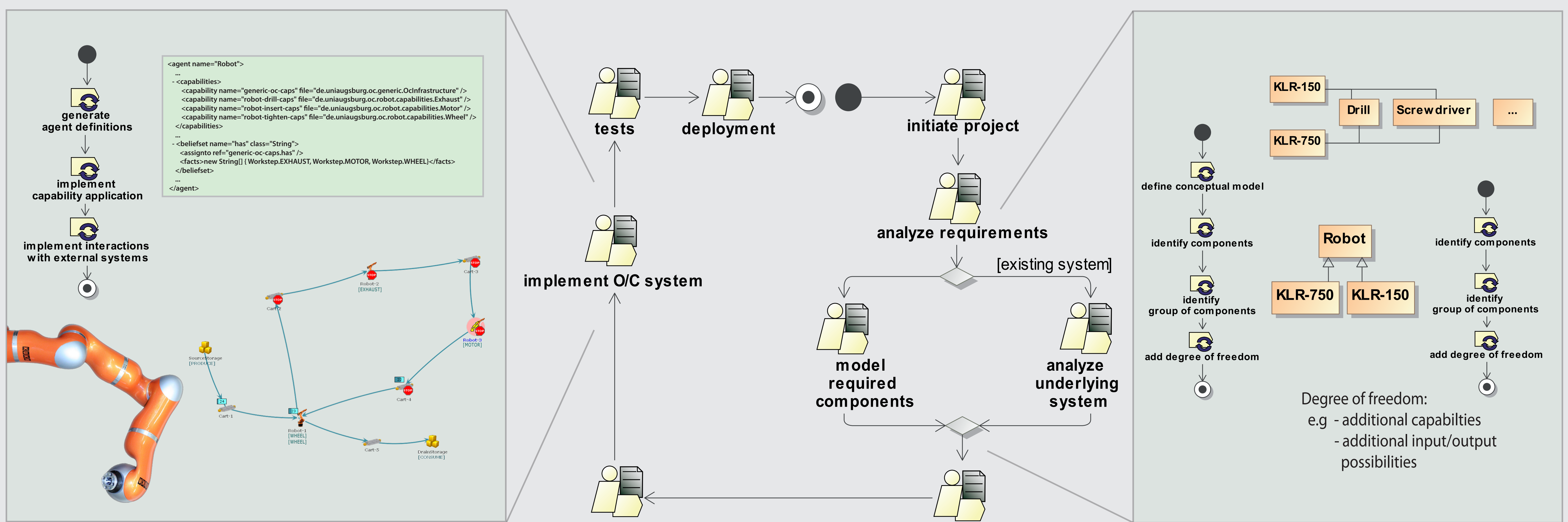
Restoration of all invariants is no longer possible.



Weakening of the invariants (e.g. priority selection)

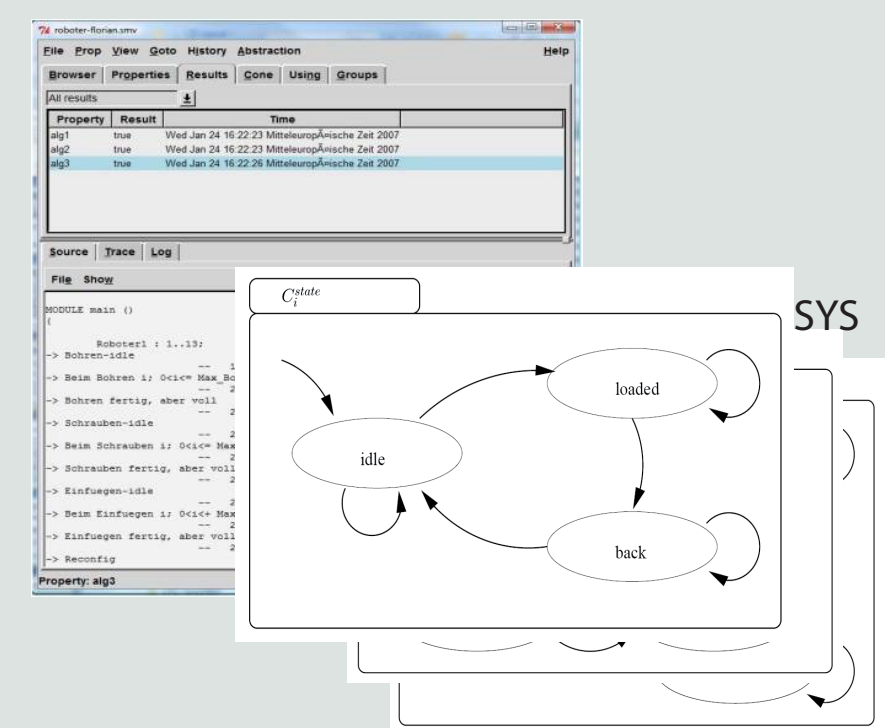


8 Steps to an Organic Computing Application



Invariants:
 $\forall a \in \text{Agent} : \forall r \in a.\text{assignedRole} : r.\text{applyCapabilities} \subseteq a.\text{availableCapabilities}$
 $\forall a \in \text{Agent} : \forall r \in a.\text{assignedRole} : r.\text{precondition.port} \in a.\text{inputs}$
 $\wedge r.\text{postcondition.port} \in a.\text{outputs}$

Formal Verification
 - Modelchecking
 - Interactive Verification



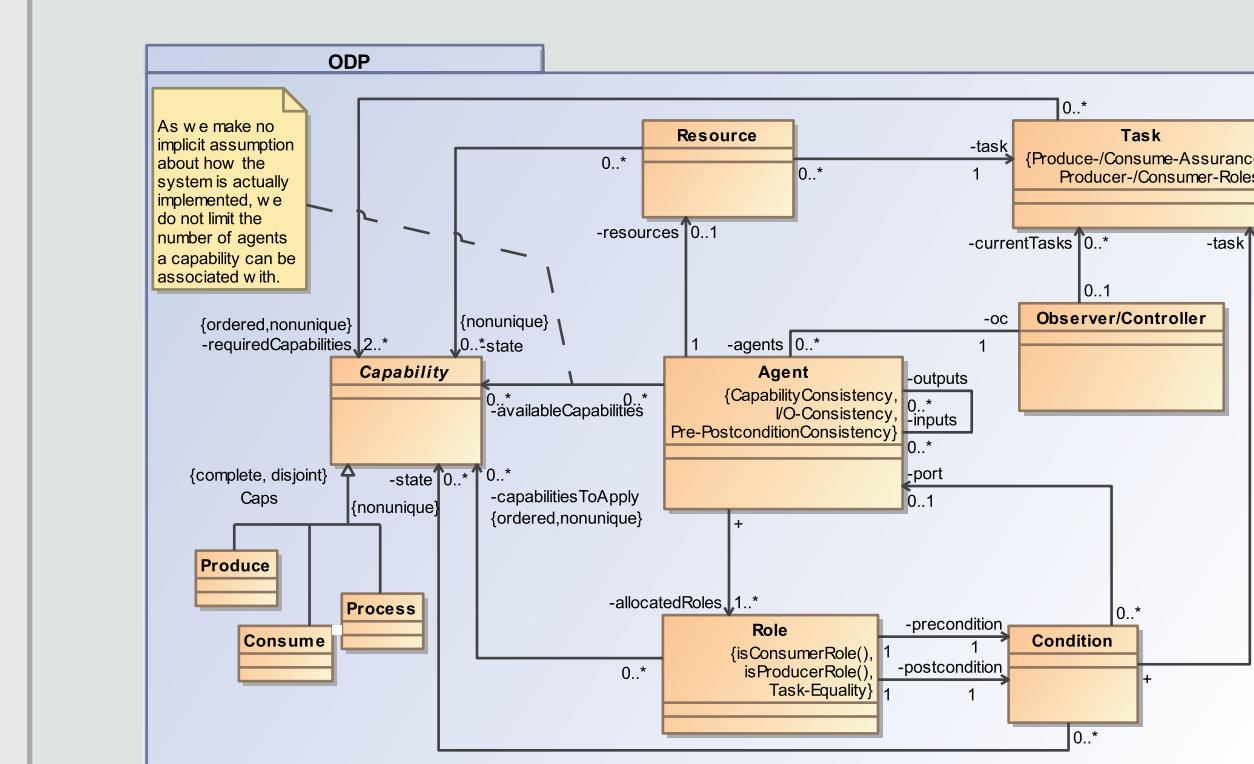
define domain specific invariants
 choose reconfiguration mechanism
 verify additional invariants
 analyze self-x properties

Qualitative Analysis
 - Fault Tree Analysis
 - adaptive DCCA

$$P \leq \sum_{CS \in \text{CSS}} \prod_{fm \in CS} P(fm)$$

Quantitative Analysis
 - multicriteria optimization
 - probabilistic modelchecking

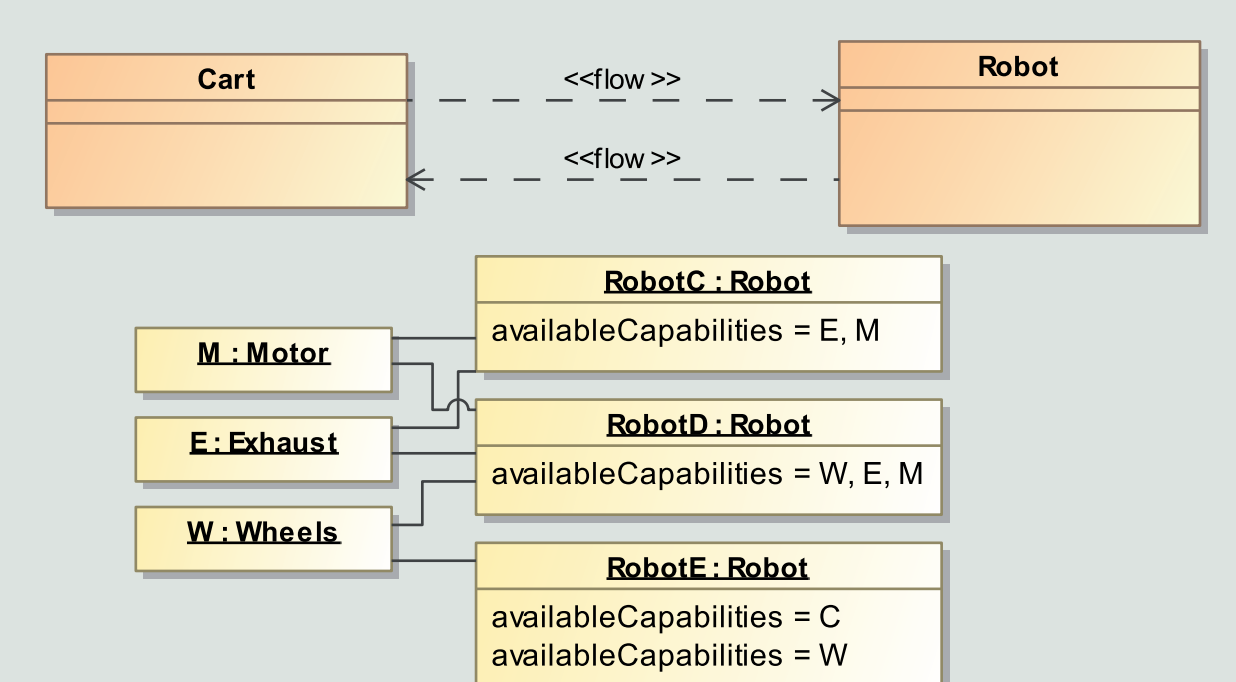
$$\text{SYS} \models E \neg (\Gamma \wedge \Delta) \text{ until } EG ((\Gamma \wedge \Delta) \wedge H)$$



map components to pattern

identify resources and tasks

instantiate concepts



Want to find out more? Contact the Organic Computing team:

Prof. Dr. Wolfgang Reif
 Hella Seebach
 Florian Nafz
 Jan-Philipp Steghöfer

Institute for Software & Systems Engineering
 Augsburg University
www.isse.de/projects
reif@informatik.uni-augsburg.de

