SAVE ORCA

Formal Modeling, Safety Analysis, and Verification of Organic Computing Applications

Hella Seebach, Florian Nafz and Wolfgang Reif

Motivation and goals

Institute for Software & Systems Engineering

- Software & Verification Co-Design for highly reliable Organic Computing applications
  - Design and construction
    - Top-Down design methodology
    - Extensible generic runtime environment
    - Integrated Software Development Process
  - Methods and tools for formal analysis and verification
    - Correctness and behavioral guarantees despite self-organization
    - Qualitative and quantitative analysis

Target systems: Resource-Flow Systems

Institute for Software & Systems Engineering

- Applications
  - Production automation
  - Logistics

- Software intensive applications that are
  - particularly resilient against disturbances and component failures (w.r.t. functional correctness, safety, security)
  - adaptive to changing requirements and modified tasks

- Agent / role based systems
  - Each agent has several capabilities
  - Each task needs different processing steps
  - Processing steps are a given sequence of capabilities

Challenges in the software engineering part

Institute for Software & Systems Engineering

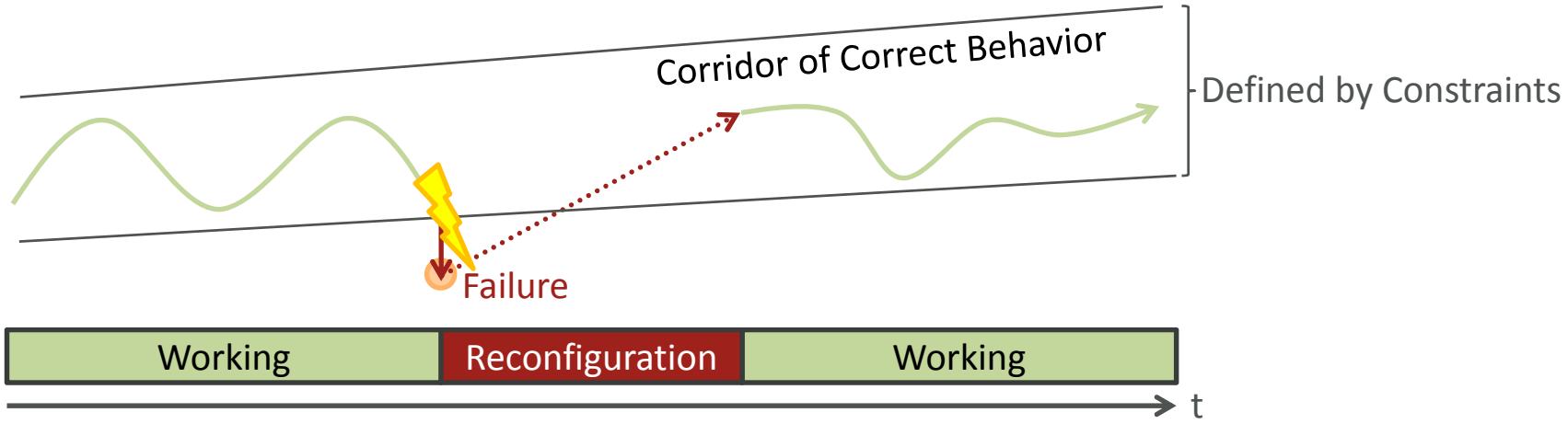
- Self-organization vs. correct system behavior ?
- How to design self-organizing systems ?
- Scalability through local reconfiguration ?

Self-organization vs. correct system behavior

Challenge 1: Self-organization vs. correct system behavior

Institute for Software & Systems Engineering

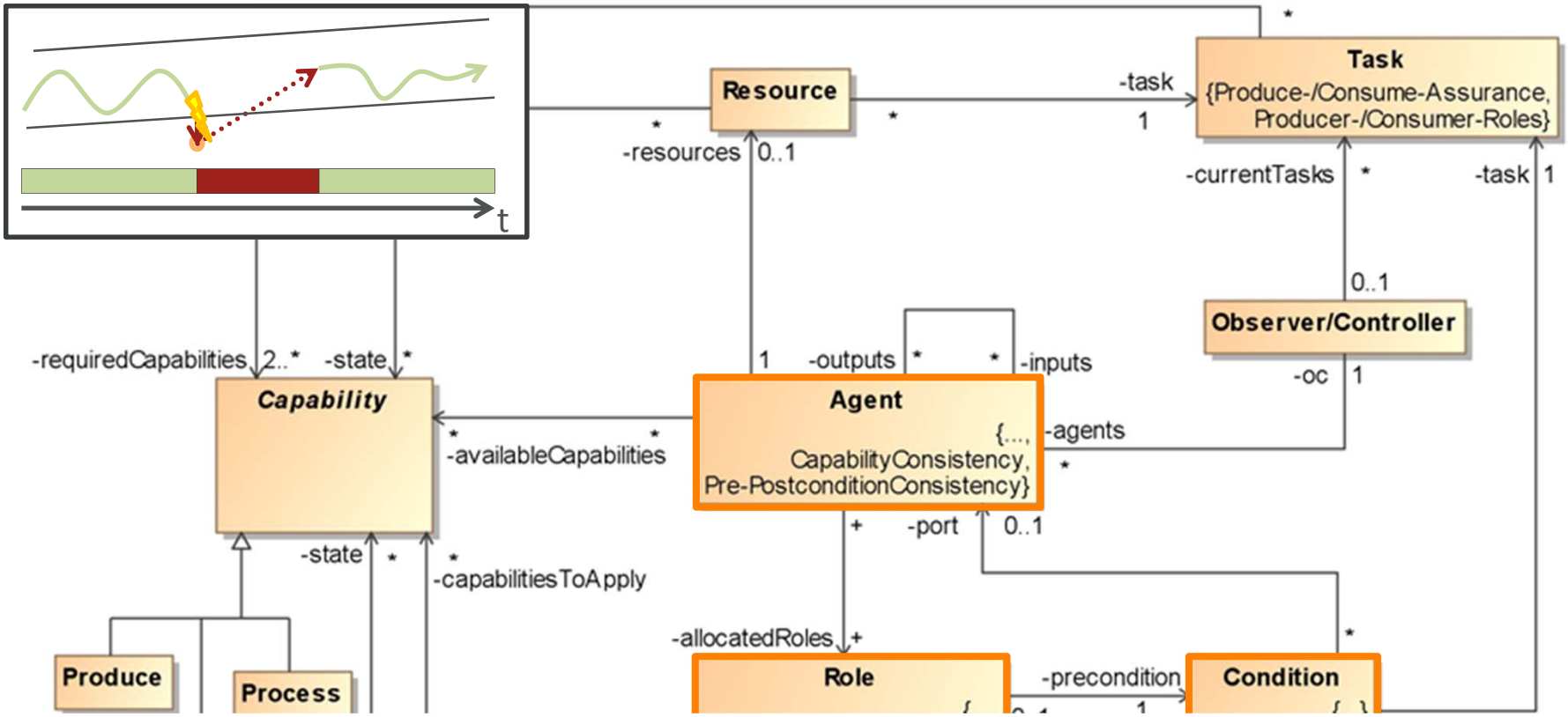
- Basic Idea: Restore Invariant Approach
- Constraints define corridor of correct behavior



[SASO08]

Organic Design Pattern (ODP) – system structure

Challenge 2: How to design self-organizing systems?



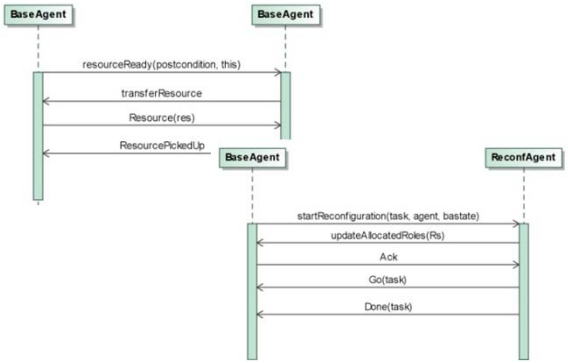
→Every ODP-system, that meets all constraints, guarantees a correct resource-flow

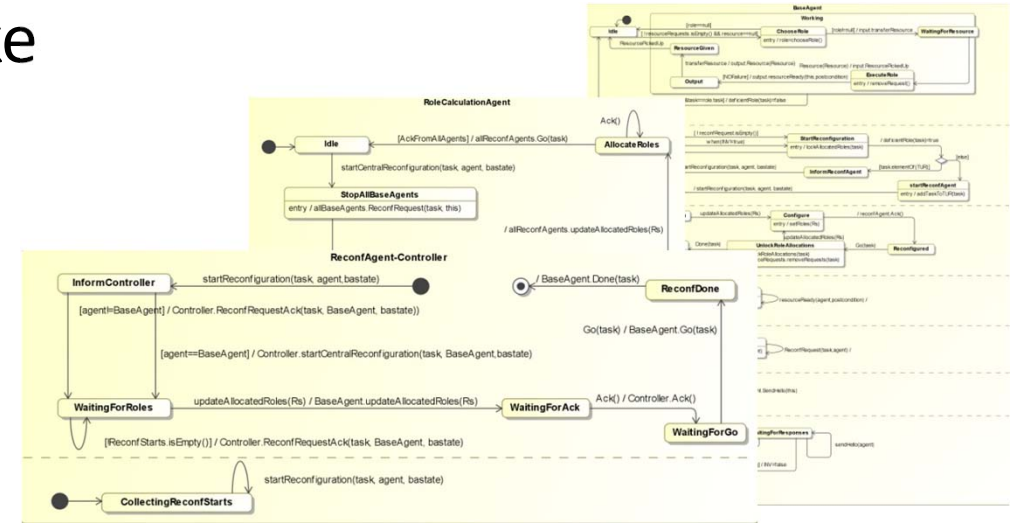
[CEC07]

Agent behavior

Institute for Software & Systems Engineering

- Fixed dynamics (statemachines) of ODP-agents
- Communication protocols (sequence diagrams) e.g. resource-handshake





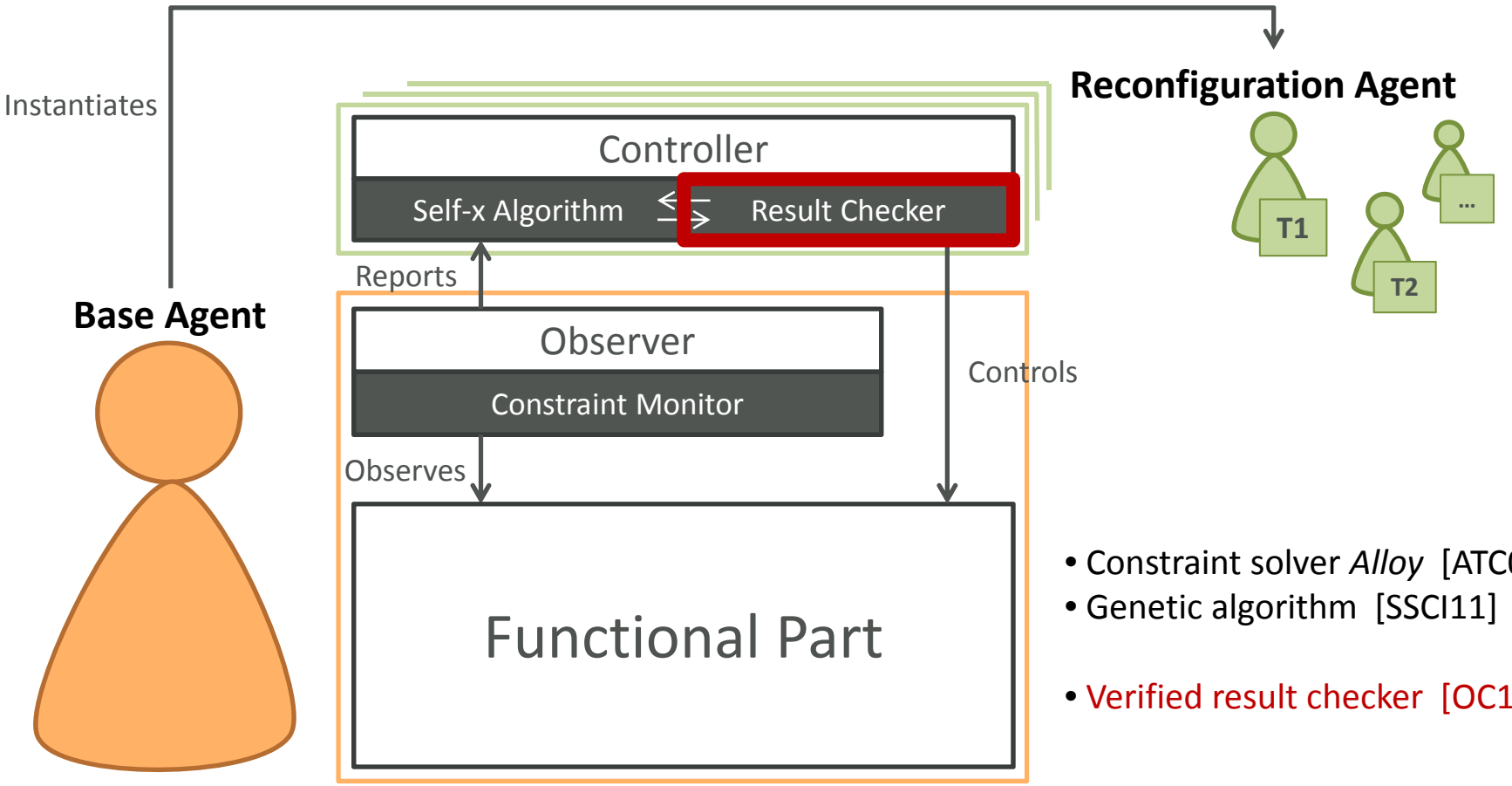
Advantages:
→ Dynamics defined for whole system class
→ Verification on system class level possible
→ Implementation — ODP Runtime Environment

[SPPOC11b]

Decentralized Observer/Controller Architecture

Institute for Software & Systems Engineering

- Constraints can be observed locally
- We distinguish between **Base Agents** and **Reconfiguration Agents**



- Constraint solver *Alloy* [ATC09]
- Genetic algorithm [SSCI11]

- Verified result checker [OC11]

Coalition formation

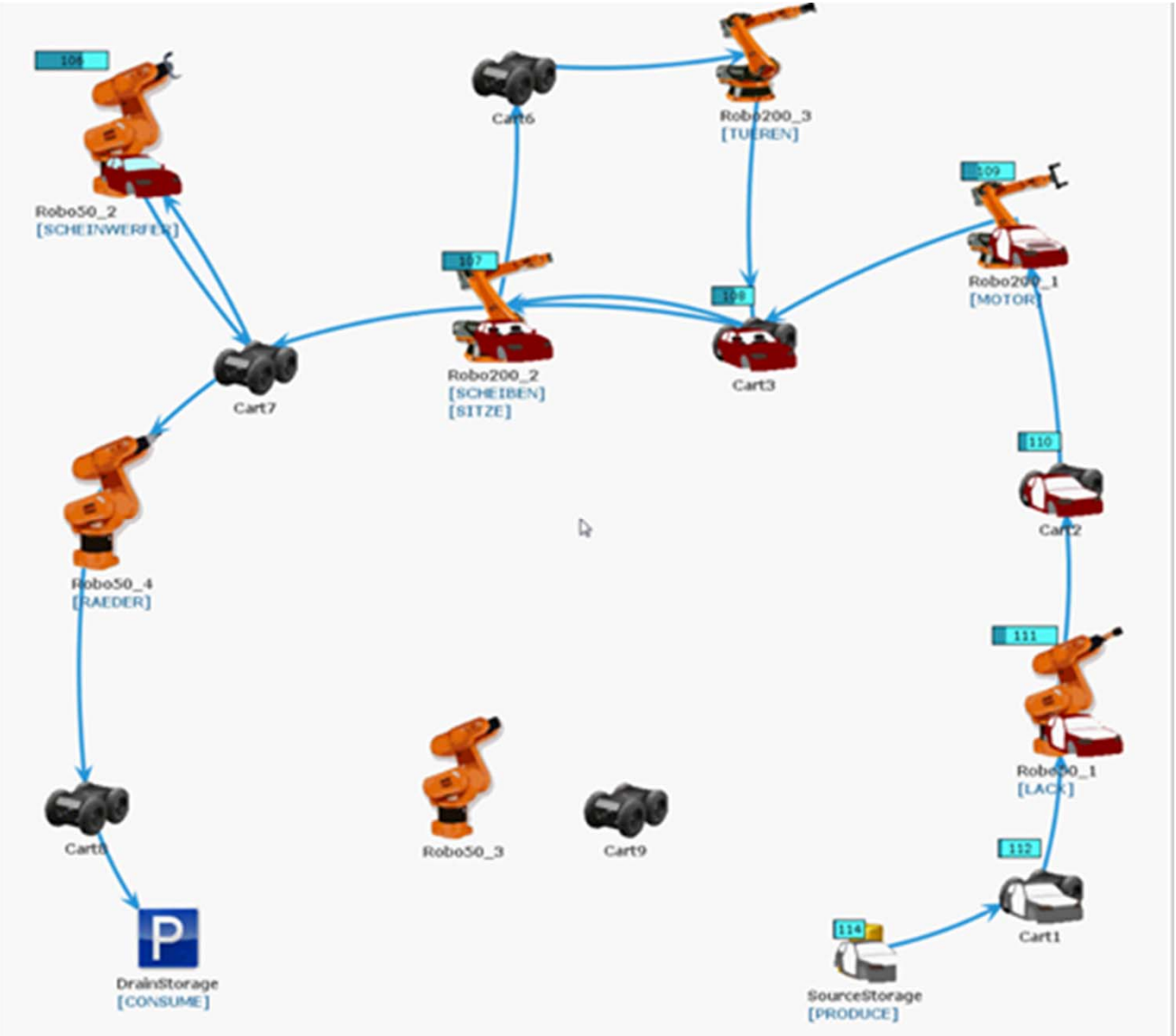
Challenge 3: Scalability through local reconfiguration

Institute for Software & Systems Engineering

- Form groups of agents that can reconfigure a part of the system with local knowledge only
- Groups are called **coalitions**
- Each coalition has a leader that coordinates the process of reconfiguration
- Local knowledge:
  - No agent has knowledge about the abilities and configuration of other agents (capabilities, inputs, outputs, allocated roles, …) as long as they are not part of the same coalition
  - Each agent only knows those agents contained in its inputs or outputs
- Make use of the underlying system structure

[EASe11]

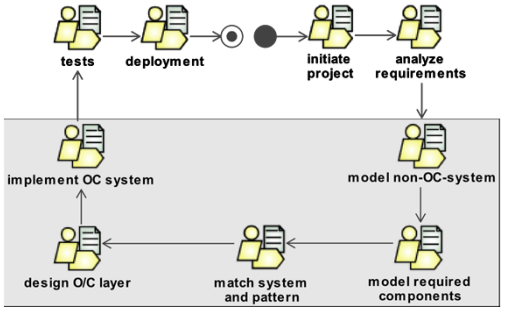
VIDEO


106
Robo50_2
[SCHEINWERFER]
Cart6
Robo200_3
[TUEREN]
109
Robo200_1
[MOTOR]
107
Robo200_2
[SCHEIBEN]
[SITZE]
108
Cart3
Cart7
110
Cart2
Robo50_4
[RAEDER]
111
Robo50_1
[LACK]
Cart8
Robo50_3
Cart9
112
Cart1
114
SourceStorage
[PRODUCE]
DrainStorage
[CONSUME]

Software Engineering Guideline

Institute for Software & Systems Engineering

- Domain model
- Instance model
- Selection of self-x-algorithm
- Code generation
- Domain specific adaptations



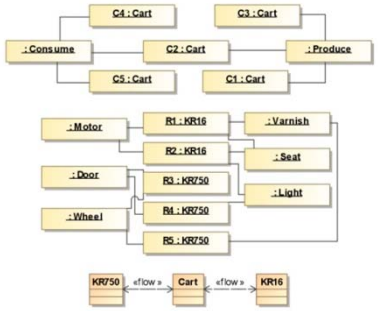
SE-Guideline

[SASO10]

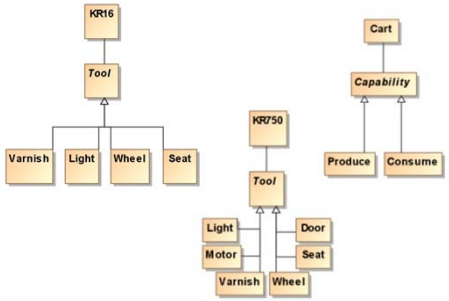
Jadex BDI Agent System

ODP Runtime Environment (ORE)

[SEAMS2009]



Instance model



Domain model

Institute for
Software & Systems
Engineering

FORMAL VERIFICATION OF ORGANIC COMPUTING APPLICATIONS

Behavioral Guarantees

Institute for Software & Systems Engineering

Goal: Correctness Assurance in OC Systems

- Provide a technique to be able to verify properties of systems despite self-x properties
  - Correctness of functional system
  - Correctness of self-x algorithms
- Systematic identification of possible failures that lead to a hazard
  - Safety Analysis
  - Quantitative properties for self-x systems

Verification Challenges

Institute for Software & Systems Engineering

- Result of a self-x phase unpredictable
- Systems have changing number of agents
- Algorithms for self-organization are hard to verify

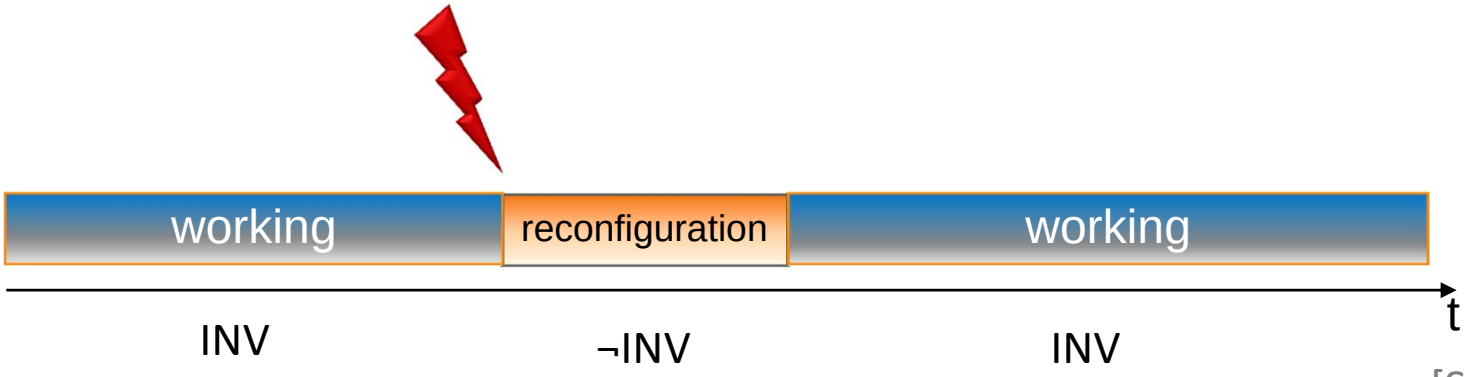
Restore-Invariant-Approach

Challenge 4: Result of a self-x phase unpredictable

Institute for Software & Systems Engineering

- Corridor specified by predicate logic formula *INV(σ)* over system states
- System goal is that this formula should hold on the entire system trace
- Whenever *INV(σ)* is violated the system tries to restore it.

$$\Box\,(\mathrm{INV} \vee (\neg\,\mathrm{INV} \rightarrow \Diamond\,(\mathrm{INV} \vee \Theta\,)\,)\,)$$



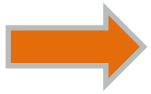
[SPPOC11a]

Seperation of Concerns

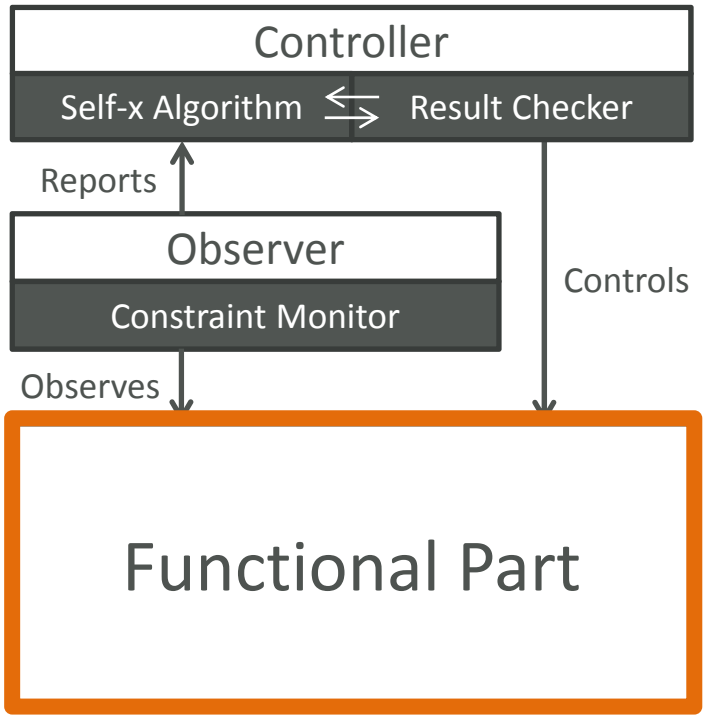
Challenge 4: Result of a self-x phase unpredictable

Institute for Software & Systems Engineering

Theorem:
The expected properties *Prop* hold in System as long as the invariant can be restored correctly by a reconfiguration mechanism.



Decoupling of self-x and functional behavior



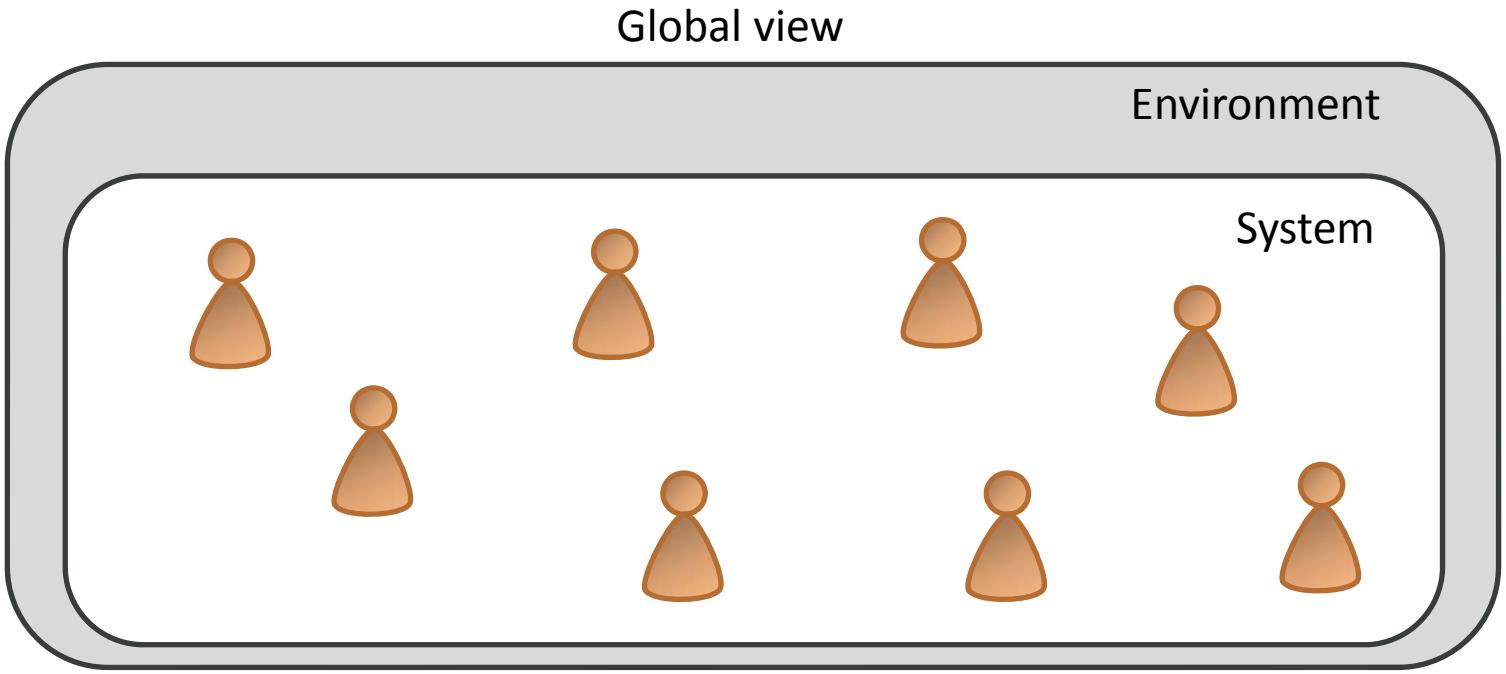
[SASO08]

Verification of functional part

Challenge 5: Systems have changing number of agents

Institute for Software & Systems Engineering

- Problem:
  - Number of agents not known at design time
  - Arbitrarily large number of agents



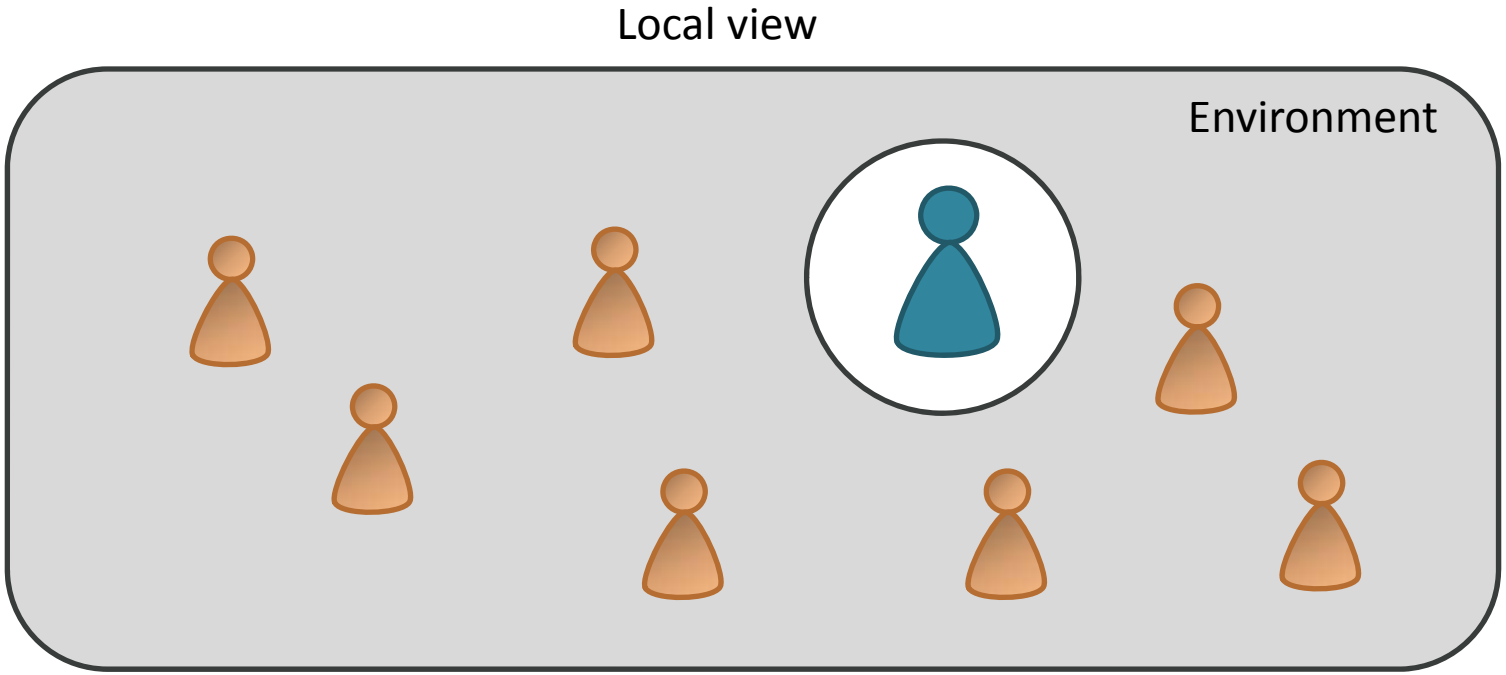
Verification of functional part

Challenge 5: Systems have changing number of agents

Institute for Software & Systems Engineering

- Solution: Compositional Reasoning

  Verification of parallel system is reduced to proving properties of the single agents

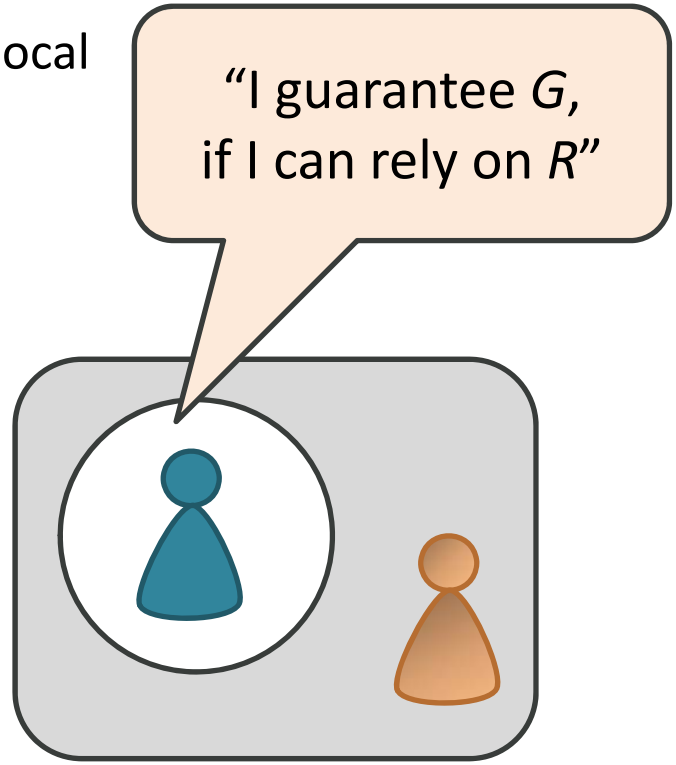


Rely/Guarantee Formalism

Challenge 5: Systems have changing number of agents

Institute for Software & Systems Engineering

- Each agent gives guarantees to its environment about the individual behavior (Guarantee), if it can rely on some properties of the environment (Rely)
  - Typical Relies R:
    - "environment doesn't change the agents local variables"
    - "incoming resources have valid state"
    - "If O/C monitors and restores invariant correctly"
  - Guarantees G:
    - "resource is produced correctly"
    - "outgoing resources have "valid" state"



[ATC10a]

Compositionality theorem for reasoning about global properties.

Correctness of Self-x Algorithm

Challenge 6 : Algorithms for self-organization are hard to verify

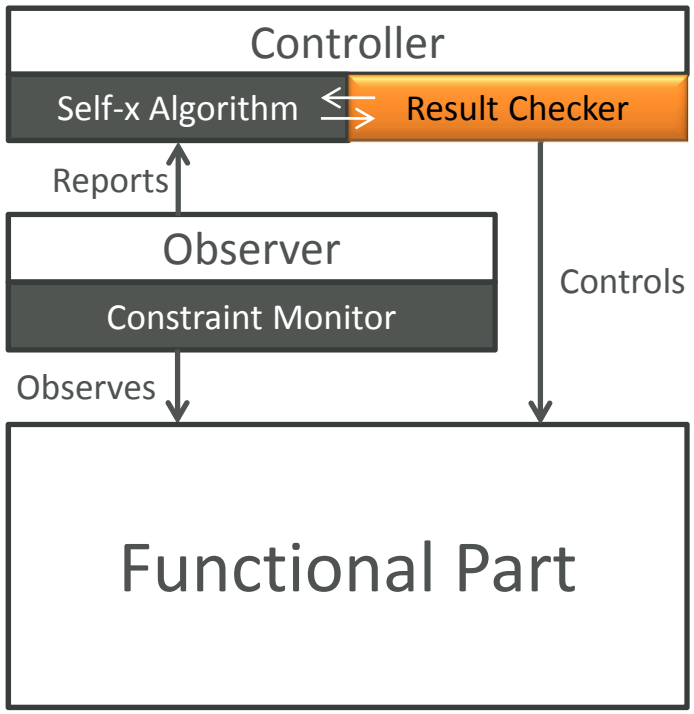
Institute for Software & Systems Engineering

- Algorithms are often complex or unsound
  - Learning techniques
  - Neural Networks
  - Genetic Algorithms

⇒ Hard or unfeasible to verify !

- Idea: Result Checker

  A component within the Controller ensures that only correct configurations are forwarded to the System



[OC11]

Verified Result Checking

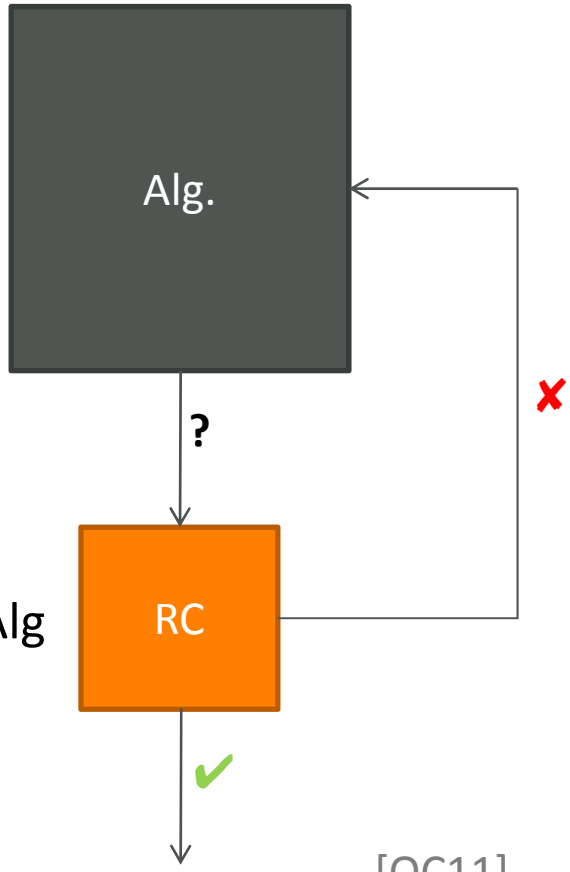
Challenge 6: Algorithms for self-organization are hard to verify

Institute for Software & Systems Engineering

- Ensure correctness of an algorithm (Alg) by an additional program, called result checker (RC)
- RC checks
  - Correctness of results
  - Not: Correctness of algorithm
- Soundness by verifying RC

- Advantages
  - *(Unlike testing)* All inputs of Alg are checked
  - *(Unlike verification)* Verification of RC, instead of Alg

    → easier task because less complex
  - Alg can be exchanged, even at runtime



[OC11]

Project summary

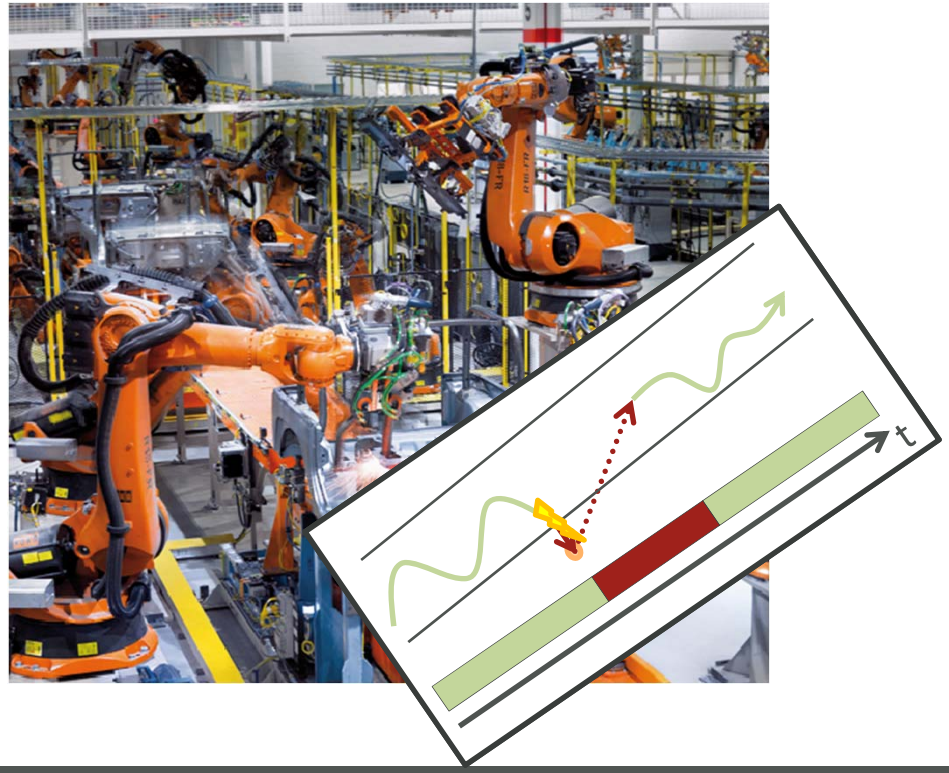
Institute for Software & Systems Engineering

- Generic verification mechanism for self-organizing systems
  - Restore Invariant Approach
- Definition of a class of systems, where all challenges were solved (Self-organizing Resource-Flow Systems)
  - Behavioral guarantees despite self-organisation
  - Top-Down Model-Driven Development
  - ODP Runtime Environment
  - Steps towards scalability: coalition formation
- Ongoing:
  - Self-optimization
  - Further work on scalability

SAVE ORCA*

Institute for Software & Systems Engineering

- 2 Ph.D. Theses
- 23 reviewed publications
- 2 technical reports
- 13 Diploma-, Master-, Bachelor-Theses



*2005-2011: one sponsored research position

Publications – Phase III

Institute for

- **[EASe11] Decentralized Reconfiguration for Self-Organizing Resource-Flow Systems Based on Local Knowledge**
  Gerrit Anders, Hella Seebach, Florian Nafz, Jan-Philipp Steghöfer, and Wolfgang Reif
  Proceedings of the 8th IEEE Conference and Workshops on Engineering of Autonomic and Autonomous Systems (EASe 2011)
- **[OC11] Ensuring correct self-reconfiguration in safety-critical applications by verified result checking.**
  Peter Fischer, Florian Nafz, Hella Seebach, and Wolfgang Reif.
  In Proceedings of the 2011 workshop on Organic computing (OC '11). ACM, New York, NY, USA,
- **[SSCI11] A Genetic Algorithm for Self-Optimization in Safety-Critical Resource-Flow Systems**
  Florian Siefert, Florian Nafz, Hella Seebach, Wolfgang Reif
  IEEE Symposium Series in Computational Intelligence 2011 (SSCI 2011)
- **[SPPOC11a] Constraining Self-organisation Through Corridors of Correct Behaviour: The Restore Invariant Approach**
  Florian Nafz, Hella Seebach, Jan-Philipp Steghöfer, Gerrit Anders, und Wolfgang Reif
  Christian Müller-Schloer, Hartmut Schmeck und Theo Ungerer (Ed.): Organic Computing — A Paradigm Shift for Complex Systems, Autonomic Systems, Birkhäuser, Springer
- **[SPPOC11b]How to Design and Implement Self-organising Resource-Flow Systems**
  Hella Seebach, Florian Nafz, Jan-Philipp Steghöfer und Wolfgang Reif
  Christian Müller-Schloer, Hartmut Schmeck und Theo Ungerer (Ed.): Organic Computing — A Paradigm Shift for Complex Systems, Autonomic Systems, Birkhäuser, Springer
- **[SORS11] Developing Self-Organizing Robotic Cells using Organic Computing Principles**
  Alwin Hoffmann, Florian Nafz, Hella Seebach, Andreas Schierl, and Wolfgang Reif
  Yan Meng and Yaochu Jin (Ed.): Bio-Inspired Self-Organizing Robotic Systems, Studies in Computational Intelligence, Volume 355, Springer-Verlag, Berlin/Heidelberg
- **[ATC10a] A Formal Framework for Compositional Verification of Organic Computing Systems**
  Florian Nafz, Hella Seebach, Jan-Philipp Steghöfer, Simon Bäumler, and Wolfgang Reif
  accepted for: Proceedings of the 7th International Conference on Autonomic and Trusted Computing (ATC 2010), Springer

Publications – Phase III

Institute for Software & Systems

- **[ATC10b] Designing Self-Healing in Automotive Systems**
  Hella Seebach, Florian Nafz, Jörg Holtmann, Jan Meyer, Matthias Tichy, Wolfgang Reif, and Wilhelm Schäfer
  accepted for: Proceedings of the 7th International Conference on Autonomic and Trusted Computing (ATC 2010), Springer
- **[SASO10] A Software Engineering Guideline for Self-organizing Resource-Flow Systems**
  Hella Seebach, Florian Nafz, Jan-Philipp Steghöfer, and Wolfgang Reif
  Proceedings of the Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2010)
- **[ICRA10] Developing Self-Organizing Robotic Cells using Organic Computing Principles**
  Alwin Hoffmann, Florian Nafz, Hella Seebach, Andreas Schierl, and Wolfgang Reif
  Workshop on Bio-Inspired Self-Organizing Robotic Systems, 2010 IEEE International Conference on Robotics and Automation (ICRA 2010), Anchorage, Alaska, USA, May 3-8, 2
- **[ARCS10]On Deadlocks and Fairness in Self-organizing Resource-Flow Systems**
  Jan-Philipp Steghöfer, Pratik Mandrekar, Florian Nafz, Hella Seebach, Wolfgang Reif
  Proceedings of ARCS 2010 - Architecture of Computing Systems, Springer
- **[MAS&S10]Design and Simulation of a Wave-like Self-Organization Strategy for Resource-Flow Systems**
  Jan Sudeikat, Jan-Philipp Steghöfer, Hella Seebach, Wolfgang Reif, Wolfgang Renz, Thomas Preisler, and Peter Salchow
  accepted for: Proceedings of the 4th International Workshop on Multi-Agent Systems and Simulation
- **[ICSE09]A generic software framework for role-based Organic Computing systems**
  Florian Nafz, Frank Ortmeier, Hella Seebach, Jan-Philipp Steghöfer and Wolfgang Reif
  SEAMS 2009: ICSE 2009 Workshop Software Engineering for Adaptive and Self-Managing Systems
- **[ATC09] A universal self-organization mechanism for role-based Organic Computing systems (best paper award)**
  Florian Nafz, Frank Ortmeier, Hella Seebach, Jan-Philipp Steghöfer and Wolfgang Reif
  Proceedings of the Sixth International Conference on Autonomic and Trusted Computing (ATC-09)

Publications – Phase I and II

Institute for Software & Systems

- **[SASO08] A specification and construction paradigm for Organic Computing systems**
  M. Güdemann, F.Nafz, F.Ortmeier, H.Seebach and W.Reif
  Proceedings of the Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2008), IEEE Computer Society Press (2008)
- **[HINF08] Organic Computing for Health Care Systems**
  F. Nafz, F. Ortmeier, H. Seebach, and W. Reif
  Proceedings of International Conference on Health Informatics
- **[ENASE08] Implementing Organic Computing Systems with Agentservice**
  Florian Nafz, Frank Ortmeier, Hella Seebach, Jan-Philipp Steghöfer and Wolfgang Reif
  3rd International Conference on Evaluation of Novel Approaches to Software Engineering
- **[CEC07]Design and Construction of Organic Computing Systems**
  Hella Seebach, Frank Ortmeier, Wolfgang Reif
  Proceedings of 2007 IEEE Congress on Evolutionary Computation, IEEE Computer Society Press 2007
- **[ISCAS07]Modeling of self-adaptive systems with SCADE**
  Matthias Güdemann, Andreas Angerer, Frank Ortmeier, Wolfgang Reif
  Proceedings of 2007 IEEE International Symposium on Circuits and Systems, IEEE Computer Society Press 2007
- **[ISOLA06] Safety and Dependability Analysis of Self-Adaptive Systems**
  M. Güdemann, F. Ortmeier, W. Reif
  Proceedings of ISoLA 2006, 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation, IEEE Computer Society Press 2006
- **[GI06]Towards Safe and Secure Organic Computing Applications**
  Matthias Güdemann, Florian Nafz, Wolfgang Reif and Hella Seebach
  INFORMATIK 2006 – Informatik für Menschen, volume P-93 of GI-Edition – Lecture Notes in Informatics
- **[ATC06] Formal Modeling and Verification of Systems with Self-x Properties**
  Matthias Güdemann, Frank Ortmeier and Wolfgang Reif
  Proceedings of the Third International Conference on Autonomic and Trusted Computing (ATC-06)

Institute for
Software & Systems
Engineering

THANKS FOR YOUR ATTENTION