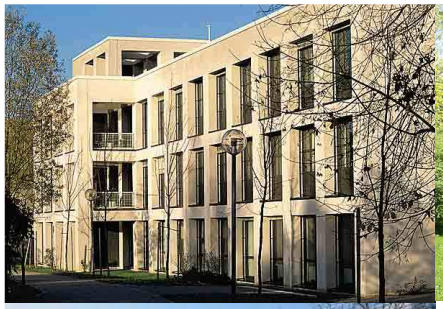






Formal Modeling, Safety Analysis, and Verification of Organic Computing Applications

SAVE ORCA

Prof. Dr. Wolfgang Reif
University of Augsburg

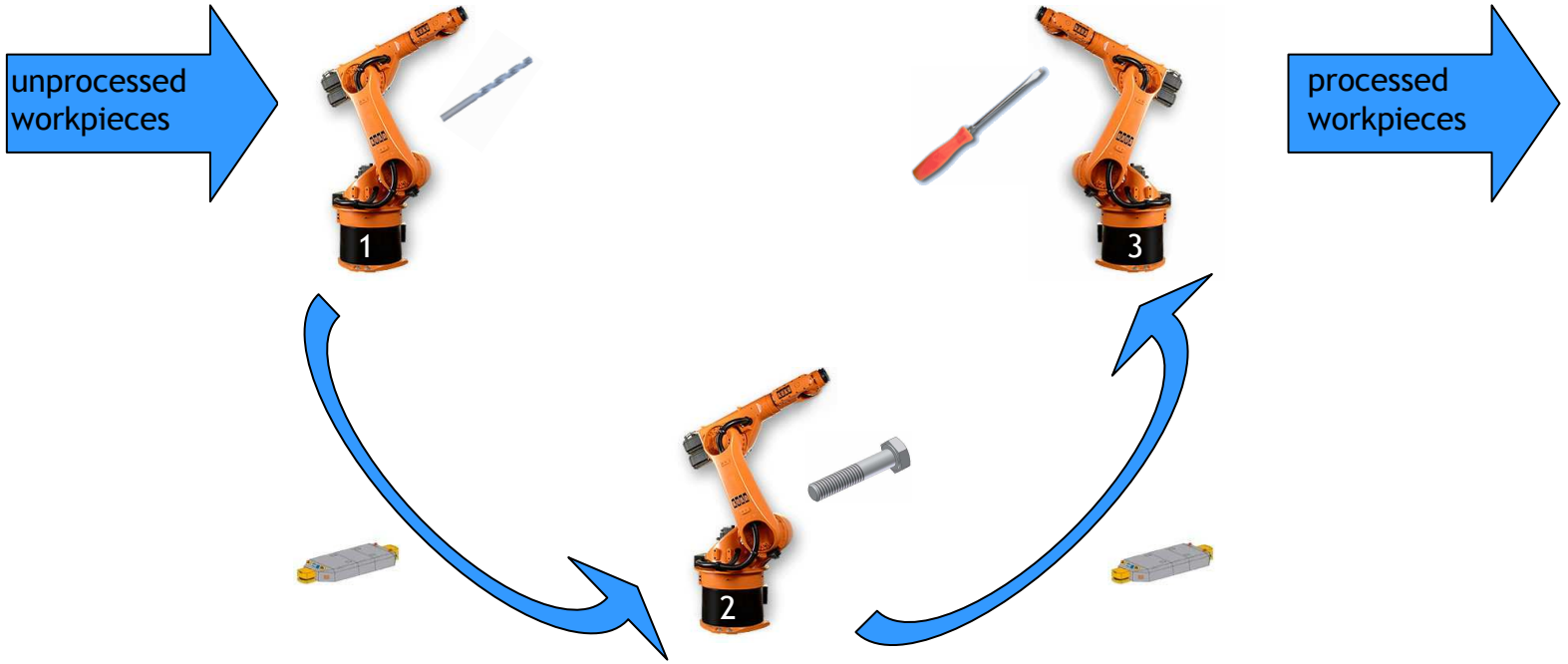
Background

- Software- and Systems-Engineering
- Formal foundations
- Embedded Applications
- Safety, Security, Sensitivity, Failure Analysis
- Application of formal models and verification
- Industrial cooperations:

  automotive, avionics / space, rail, process automation, mechatronics, …

Example: Organic production cell

- 3 robots, each with 3 different tools (drilling a hole, inserting a screw and tightening the screw)
- Holonic transport units (carts) between robots
- Self-organizing (wrt failure and changes)



Example: Organic production cell

Relevant questions:

- Functional correctness:
  - Is processing of workpieces still/again possible after reconfiguration?
  - Is no piece treated in a wrong order? Will some piece be wasted?
- Safety:
  - Is there no collision between carts?
- Reconfiguration and Optimization potential:
  - How long does reconfiguration take?
  - Does the new configuration have the maximal throughput?
  - What is the maximum number of tolerable failures?

Goal of the project

(Top-Down) Design Framework for highly reliable and adaptive organic computing application

- Including: modeling, refinement, validation and verification
- Reliability = functional correctness and safety under unexpected disturbances and component failures
- Adaptive = adaptive system behavior under changing requirements and modified tasks
- Component architecture and substitutivity
- Self-x properties: self-adaptive, self-healing, self-optimization
- Quantitative assessment of self-x properties

Work Plan (1)

- Descriptive Modeling
  - Evaluation of specification formalisms
  - Extending formalisms to self-x properties
  - Semantic foundation
- A logic to prove self-x properties
- Safety analysis for OC systems
- Extension to continously self-adapting systems

Work Plan (2)

- Quantitive Analysis
- Refining self-x capabilities
- Generation of test cases and code
- System engineering methodology
- Case studies
- Prototypical tool support

Desired Cooperations

- Digital On-Demand Computing Organism for Real-Time Systems (Becker, Brinkschulte, Henkel, Karl, Wörn)
- Model-Driven Development of Self-Organizing Control Applications (Heiß, Mühl, Weis)
- Organic Fault-Tolerant Control Architecture for Robotic Applications (Maehle, Brockmann, Großpietsch)
- Smart Teams: Local, Distributed Strategies for Self-Organizing Robotic Exploration teams (Meyer auf der Heide, Schindelhauer)
- Quantitative Emergence - Metrics, Observation and Control Tools for Complex Organic Ensembles (Schmeck, Müller-Schloer, Branke)
- Organic Computing Middleware for Ubiquitous Environments (Ungerer)