Organic Computing
Status, Outlook, Related Research

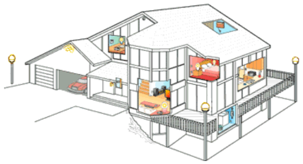
Hartmut Schmeck
Institut AIFB
Karlsruhe Institute of Technology –

Overview

- Vision
- Status
- Outlook
- Related Activities
- Conclusion

What Organic Computing is *not about*

So, what is it about?






- Collections of **intelligent (embedded) systems**
  (scenarios like **smart house, car, office, factory, shop, healthcare**,…
  …**ubiquitous, pervasive computing**).
- Potentially **unlimited networks** (large number, mobility)
- **Spontaneous local interaction,** leading to unexpected global behaviour (**emergent phenomena** as a result of **self-organisation**)
- **Robust services** in dynamically changing environments
  (e.g. mobile communication).
- **Flexible behaviour** as a reaction to varying external constraints
  (e.g. traffic light control)
- **Design, management and acceptance problems** wrt increasingly **complex systems**
  → **Controllability? Trustworthiness?**

⇒ **We have to come up with good ideas for**
  – **designing, managing, and controlling unlimited, dynamical networks of intelligent devices,**
  – **utilising the available technology for the utmost benefit to humans.**

GI/ITG Position paper 2003:

Vision for System Architecture > 2010

- **Organic Computer Systems**
  - will possess lifelike properties.
  - will consist of autonomous and cooperating sub systems and will work, as much as possible, in a self-organised way.
  - will adapt to human needs,
  - will be robust, adaptive, and flexible,
  - will be controlled by objectives (“goal-driven”),
  - will provide customized service in a user-friendly way,
  - will be trustworthy.

- **Self-organisation** allows for adaptive and context aware behaviour:
  - self-configuring
  - self-optimizing
  - self-healing
  - self-protecting
  - self-explaining
  - self-managing
  - …

Bio-inspired?

Observer/controller? EPOS? Emergent control?

MAPE? Anti agents? OCU? MAS?

Operator/controller?

Model-predictive? Controlled-self-organized?

Model-based?

Organic Computing

Organic Control Unit?

Organic middleware?

Feedback control?

Policies?

It is not the question,
whether adaptive and self-organising systems will emerge,
but ***how*** they will be designed and controlled.

Self-managing?

Autonomic?

SILKE?

Feedforward control?

Self-adaptive? Self-organized? Hormons?

Brief reminder:

DFG priority program 1183 „Organic Computing“ (2005 – 2011)

www.organic-computing.de/SPP

Program structure and objectives …

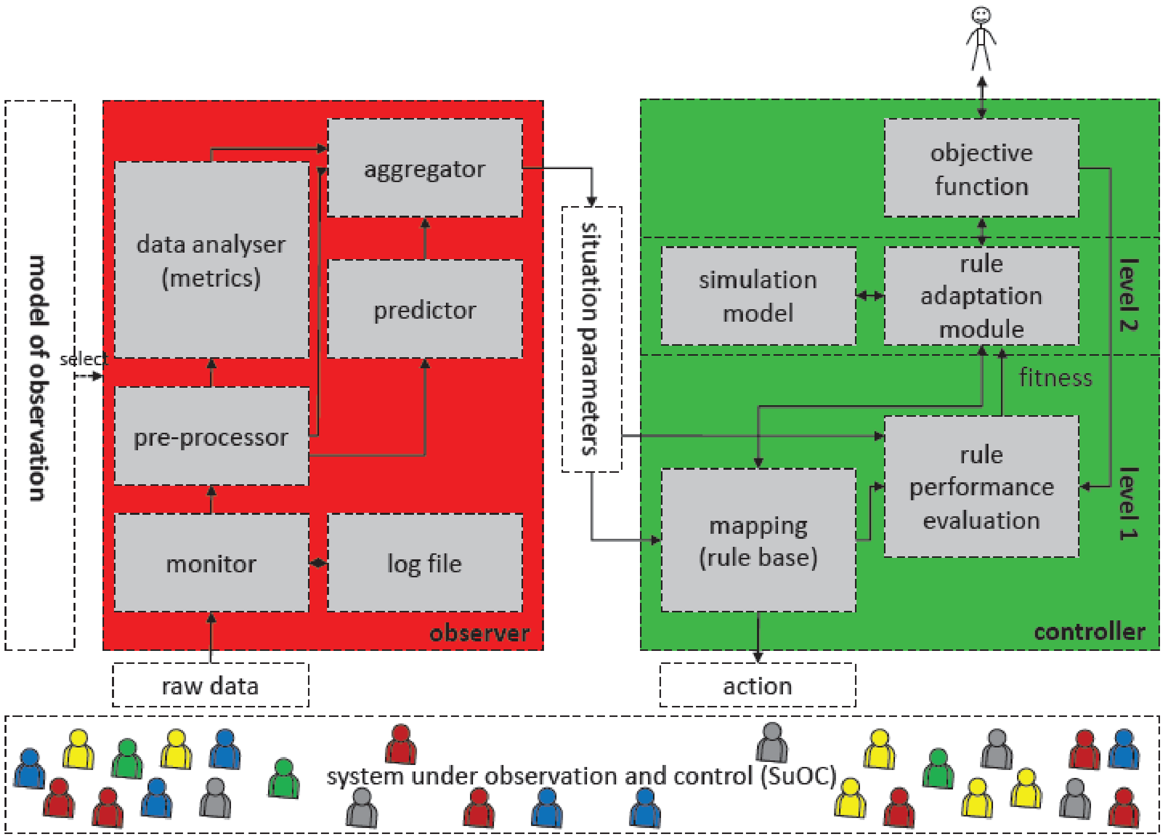
Nature	(1) OC-Principles	(2) OC-Technology	(3) Technical Applications
	Emergence Self-x	OC-toolbox	
Dissip. Struct. Ants Swarms Brain ….	Autonomy Self-x. Emergence Awareness Cooperation Competition …	Basic technologies Observer/Controller Guarding Helper Threads Embedded Learning Complete system arch. …	Car Office Telecom Factory Home Health&Care ...

Phase III: 18 Projects

- **Learning to Look at Humans**
(Würtz, Uni Bochum)
- **Model-Driven Development of Self-Organizing Control Applications**
(Heiß, Mühl, TU Berlin, Weis Uni Duisburg)
- **Organic Fault-Tolerant Control Architecture for Robotic Applications** (Maehle, Brockmann, Uni Lübeck, Großpietsch FhG, St. Augustin)
- **Smart Teams: Local, Distributed Strategies for Self-Organizing Robotic Exploration Teams**
(Meyer auf der Heide, Schindelhauer, Uni Paderborn)
- **Formal Modeling, Safety Analysis, and Verification of Organic Computing Applications – SAVE ORCA**
(Reif, Uni Augsburg)
- **Embedded Performance Analysis for Organic Computing** (Ernst, TU Braunschweig)
- **OCCS - Observation and Control of Collaborative Systems** (Branke, Schmeck, KIT; Hähner. Müller-Schloer Uni Hannover)
- **OTC² - Organic Traffic Control Collaborative**
(Hähner, Müller-Schloer, Uni Hannover, Branke, Schmeck Uni Karlsruhe)
- **AUTONOMOS: A distributed and self-regulating approach for organizing a large system of mobile objects** (Fekete, TU Braunschweig, Fischer, Uni Lübeck)
- **Organic Self-organizing Bus-based Communication Systems** (Teich, Uni Erlangen)
- **Organisation and Control of Self-Organising Systems in Technical Compounds**
(Middendorf, Uni Leipzig)
- **Architecture and Design Methodology for Autonomic System on Chip** (Rosenstiel, Uni Tübingen, Herkersdorf, TU München)
- **Multi-Opjective Intrinsic Evolution of Embedded Systems (MOVES)** (Platzner, Uni Paderborn)
- **OCμ - Organic Computing Middleware for Ubiquitous Environment** (Ungerer, Uni Augsburg)
- **The bio-chemical information processing metaphor as a programming paradigm for organic computing**
(Dittrich, Uni Jena)
- **Energy Aware Self Organized Communication in Complex Networks** (Timmermann, Uni Rostock)
- **Generic emergent computing in chip architectures**
(Fey, Uni Erlangen)
- **On-line Fusion of Functional Knowledge within Distributed Sensor Networks** (Sick, Uni Passau)
- **A Modular Approach for Evolving Societies of Learning Autonomous Systems** (Rammig. Kleinjohann, Uni Paderborn))
- **Digital On-Demand Computing Organism for Real-Time Systems** (Becker, Henkel, Karl, Uni Karlsruhe, Brinkschulte, Uni Frankfurt)
- **Emergent radio: Emergent strategies to optimise collaborative transmission schemes** (Beigl, TU Braunschweig,)

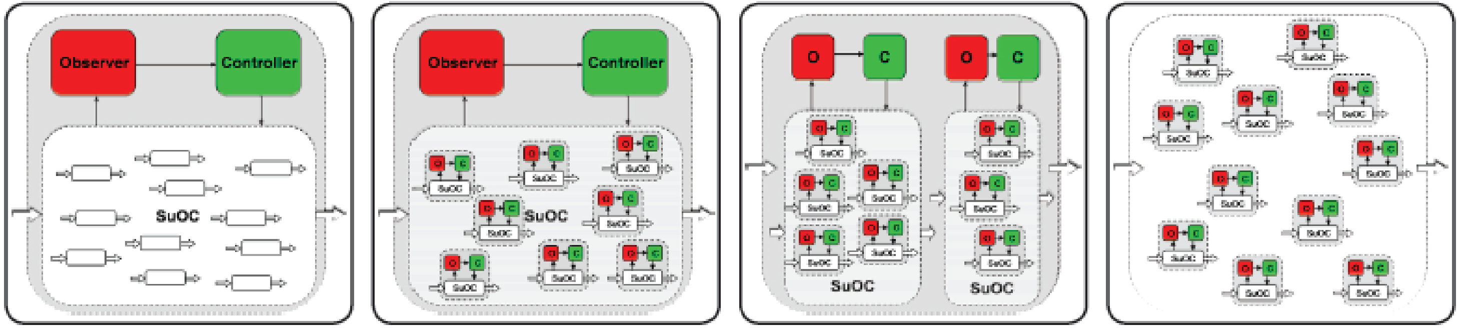
What did we achieve so far: Foundations

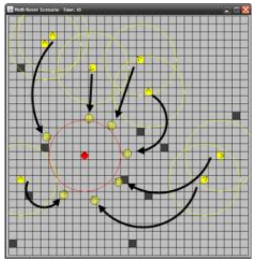
- Controlled self-organisation in technical systems
- Fundamental insights into the principles of self-organisation
- Developing a technological basis for Organic Computing systems



Design of a generic observer/controller architecture (Branke, Müller-Schloer, Schmeck et al.)

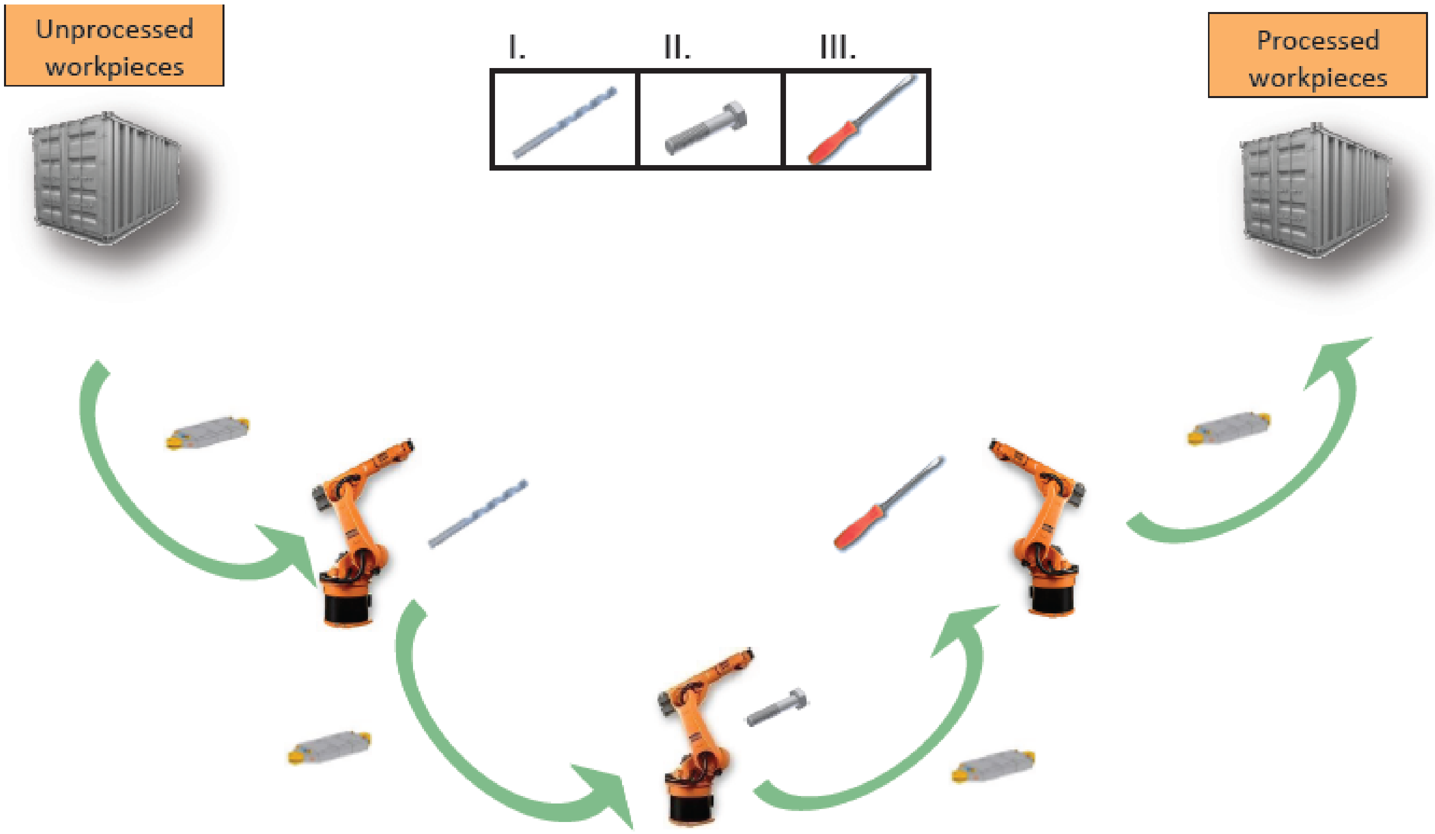
What did we achieve so far: Foundations





- Characterisation and classification of emergence in self-organ-isiging systems (entropy-based) (Müller-Schloer, Sick, et al.)
- Characterisation of basic Organic Computing properties (e. g., autonomy, controlled self-organisation, adaptivity, robustness, flexibility,…)
  (Mühl, Müller-Schloer, Ortmeier, Schmeck, et al.)
- Principles of collaboration, coordination, and learning in multi-agent systems (Hähner, Müller-Schloer, Schmeck, et al.)
- Systematic investigation of self-organising systems in nature (swarms of bees and bio-chemical information processing) (Middendorf, Dittrich et al.)

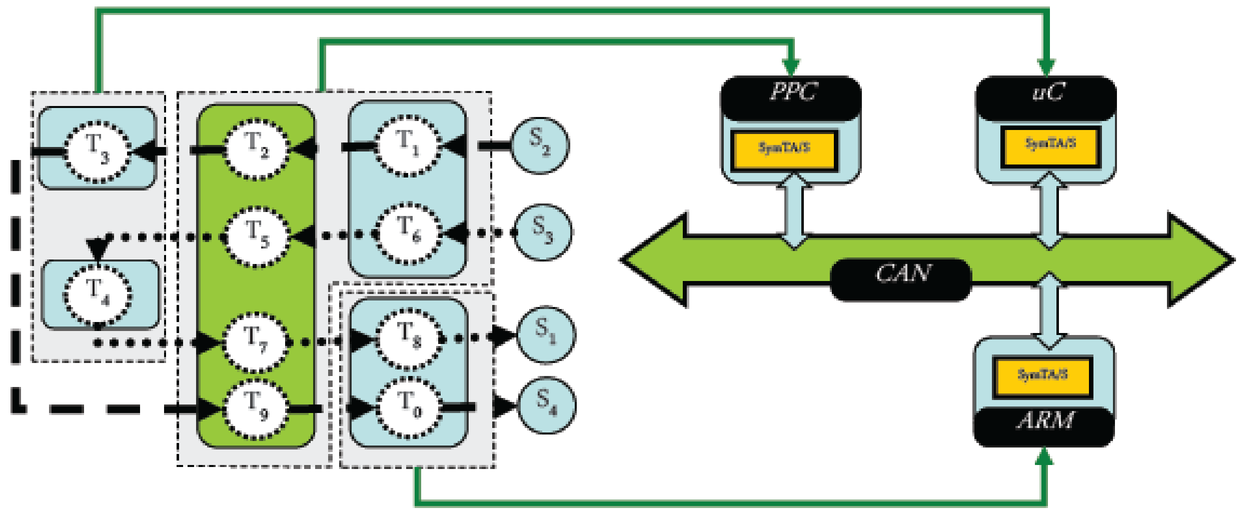
What did we achieve so far: Foundations



Safety analysis and verification of Organic Computing applications (Reif et al.)

What did we achieve so far: Technologies

- Design and experimental investigation of generic concepts
- Architectures and tools for realising organic computer systems

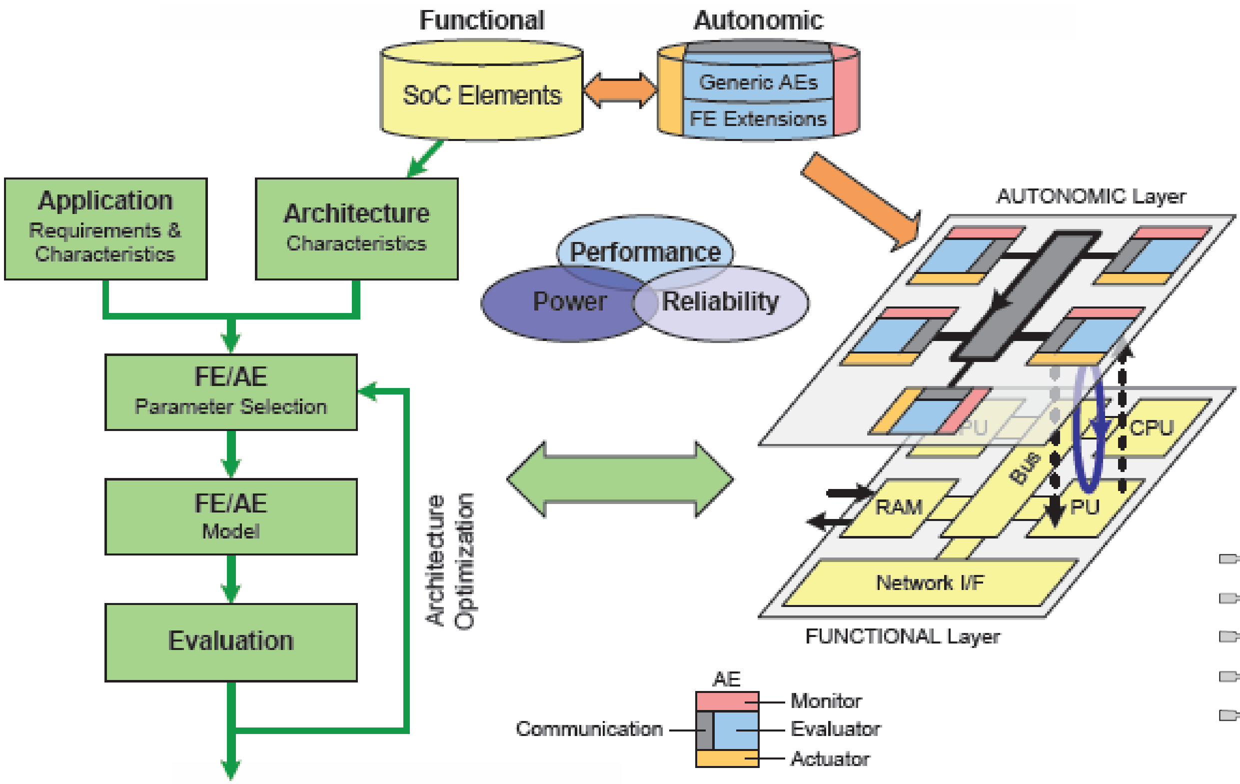


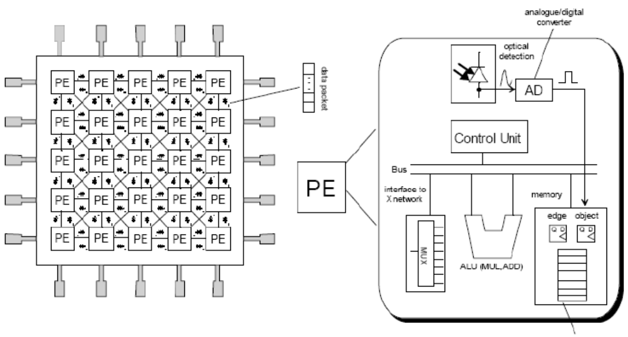
Global Model Real System

Embedded performance analysis of Organic Computing systems using a variant of the observer/controller architecture (Ernst et al.)

Organic self-organising bus-based communication systems (Teich et al.)

What did we achieve so far: Technologies

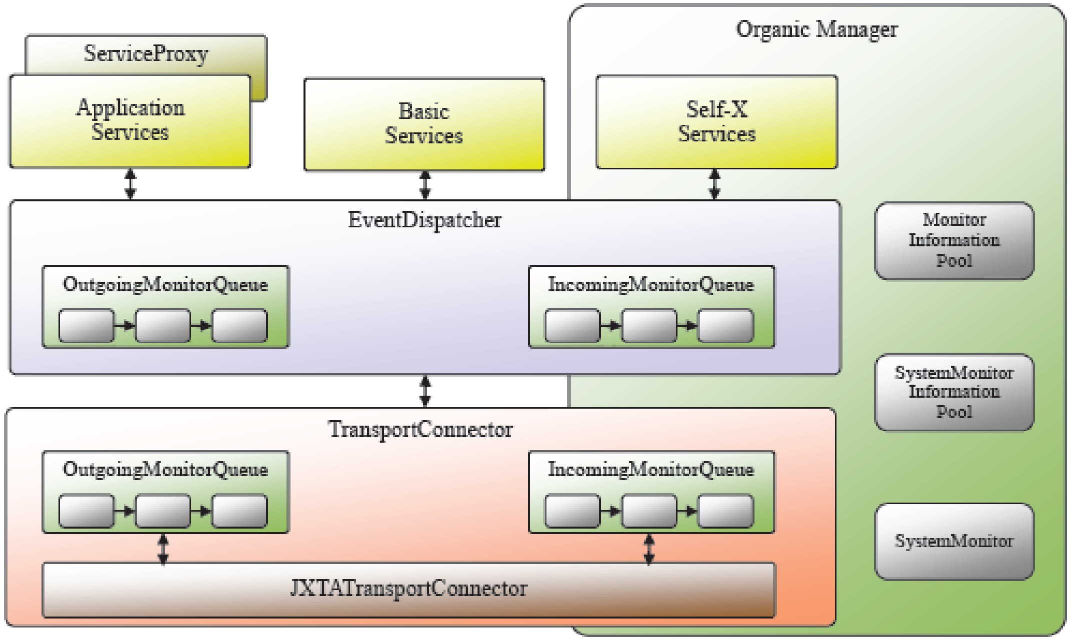


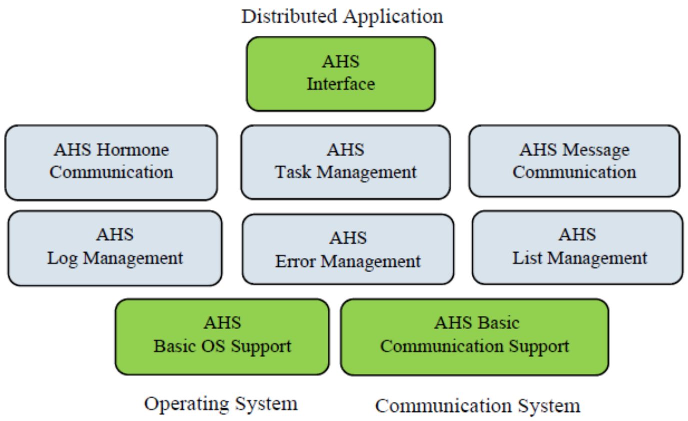


Architecture and design methodology for autonomic/organic systems on chip (Herkersdorf, Rosenstiel et al.)

Concept of marching pixels and generic emergent computing in chip architectures (Fey et al.)

What did we achieve so far: Technologies

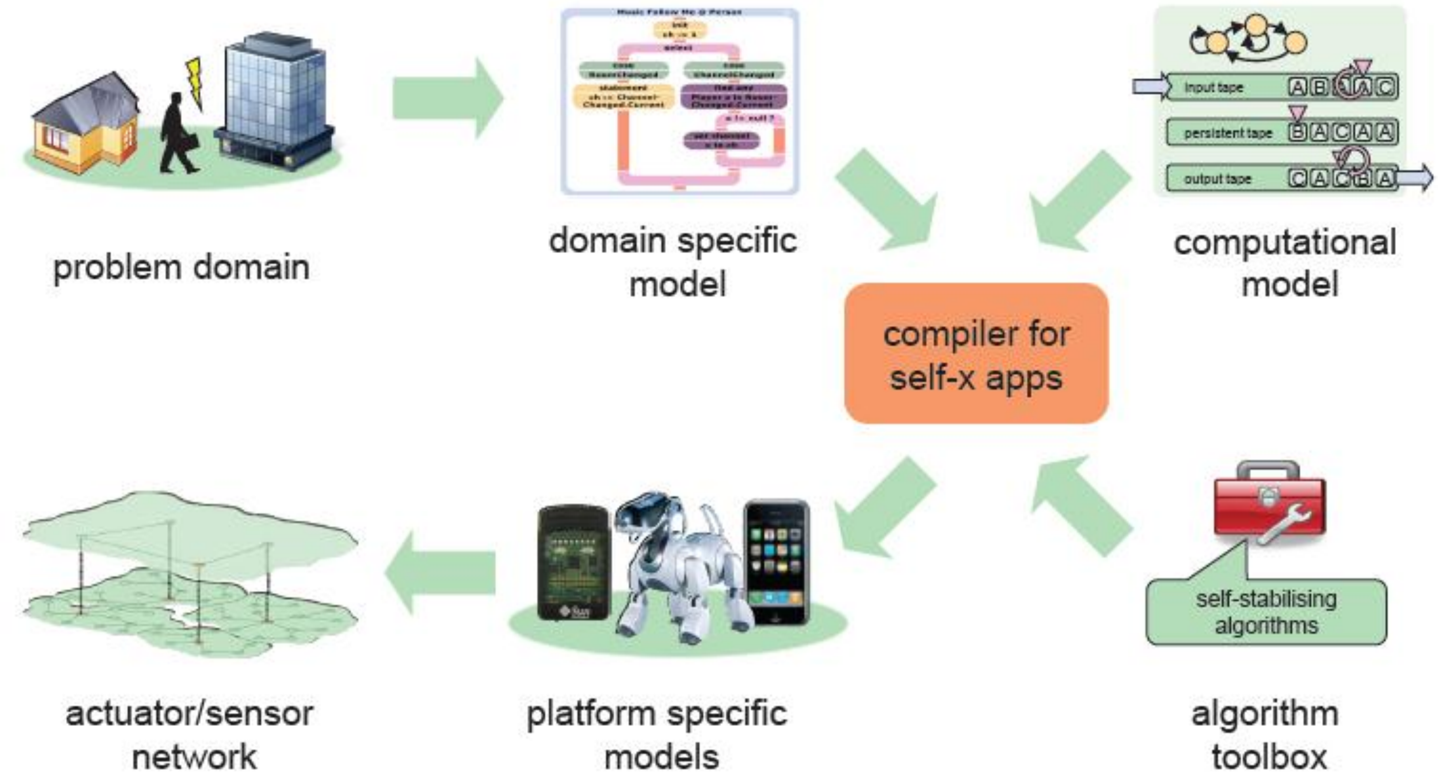




Design of generic Organic Computing middleware components (Becker, Brinkschulte, Henkel, Karl, Trumler, Ungerer et al.)

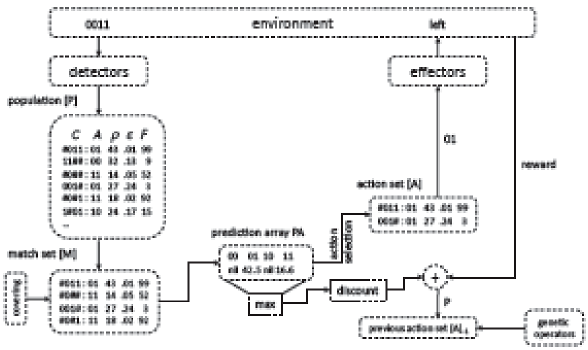
Artificial immune systems, self-protection, and self-healing (Timmermann, Trumler, Ungerer, et al.)

What did we achieve so far: Technologies

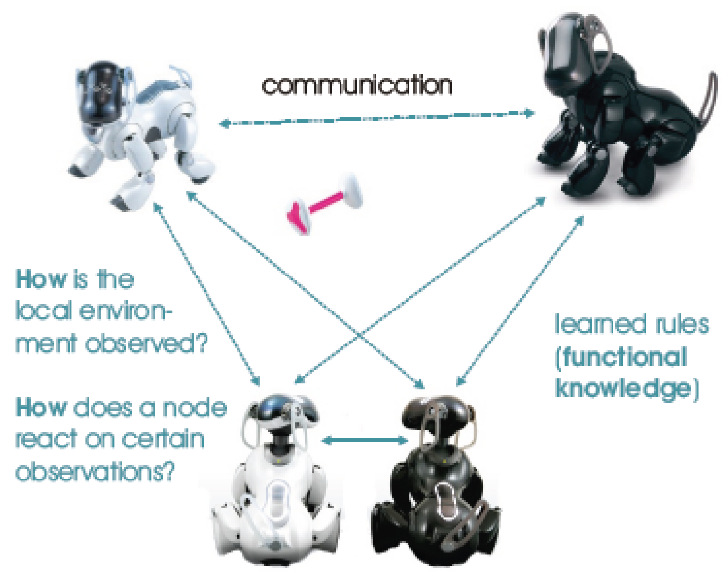


Formal definition and modelling of self-x systems; self-organising algorithms and development tools for Organic Computing applications (Heiss, Mühl, Richling, Wacker, Weis et al.)

What did we achieve so far: Technologies



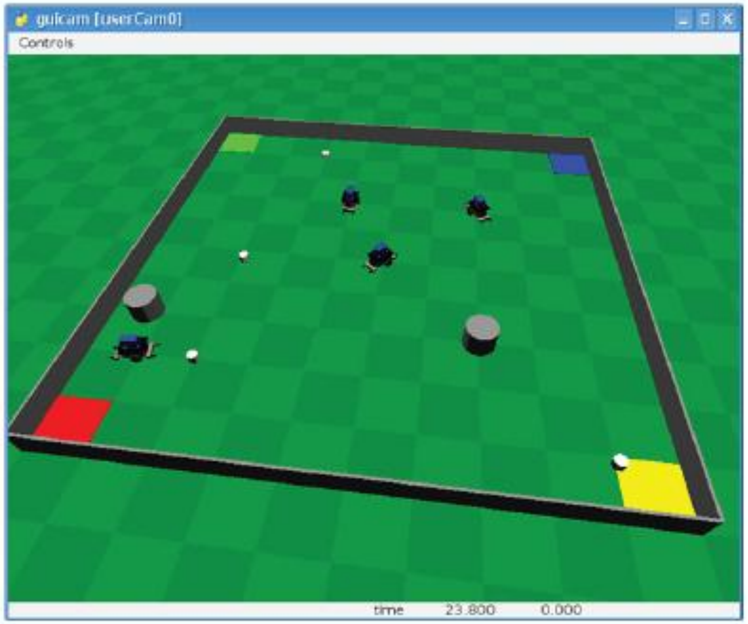
Use of classifier systems for online learning (Fredivianus, Proth-mann, Richter, Rochner, Zeppenfeld)



Autonomic learning of function-al knowledge in sensor systems (Sick et al.)

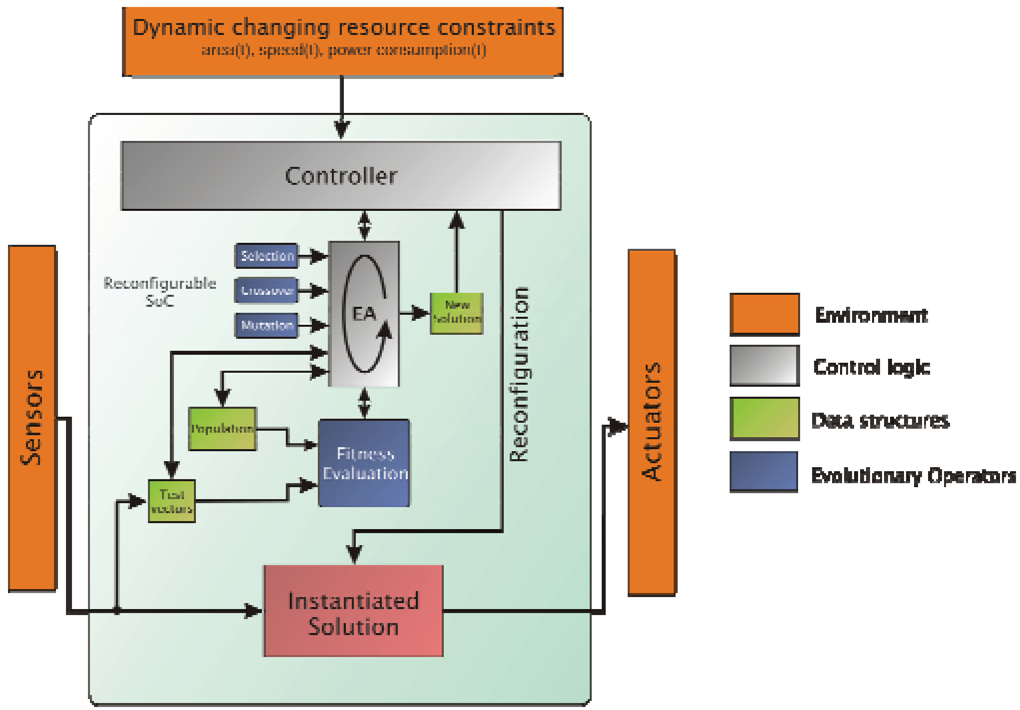
What did we achieve so far: Technologies

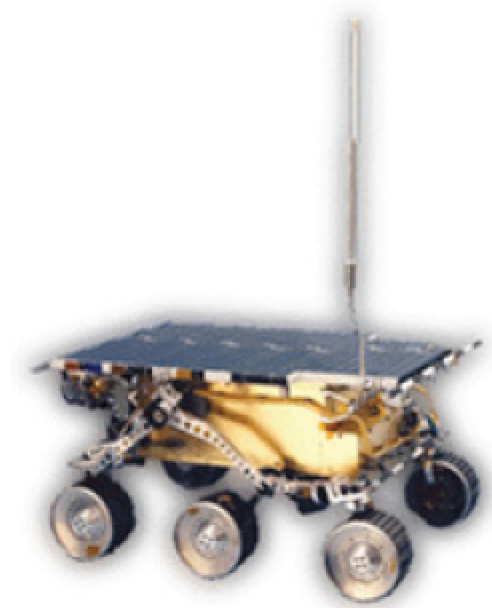
Design of a self-configuring and self-healing robot control architecture (Brockmann, Großpietsch, Maehle et al.)



Analysis of emerging behaviour in evolving societies of learning autonomous systems (Kleinjohann, Rammig et al.)

What did we achieve so far: Technologies

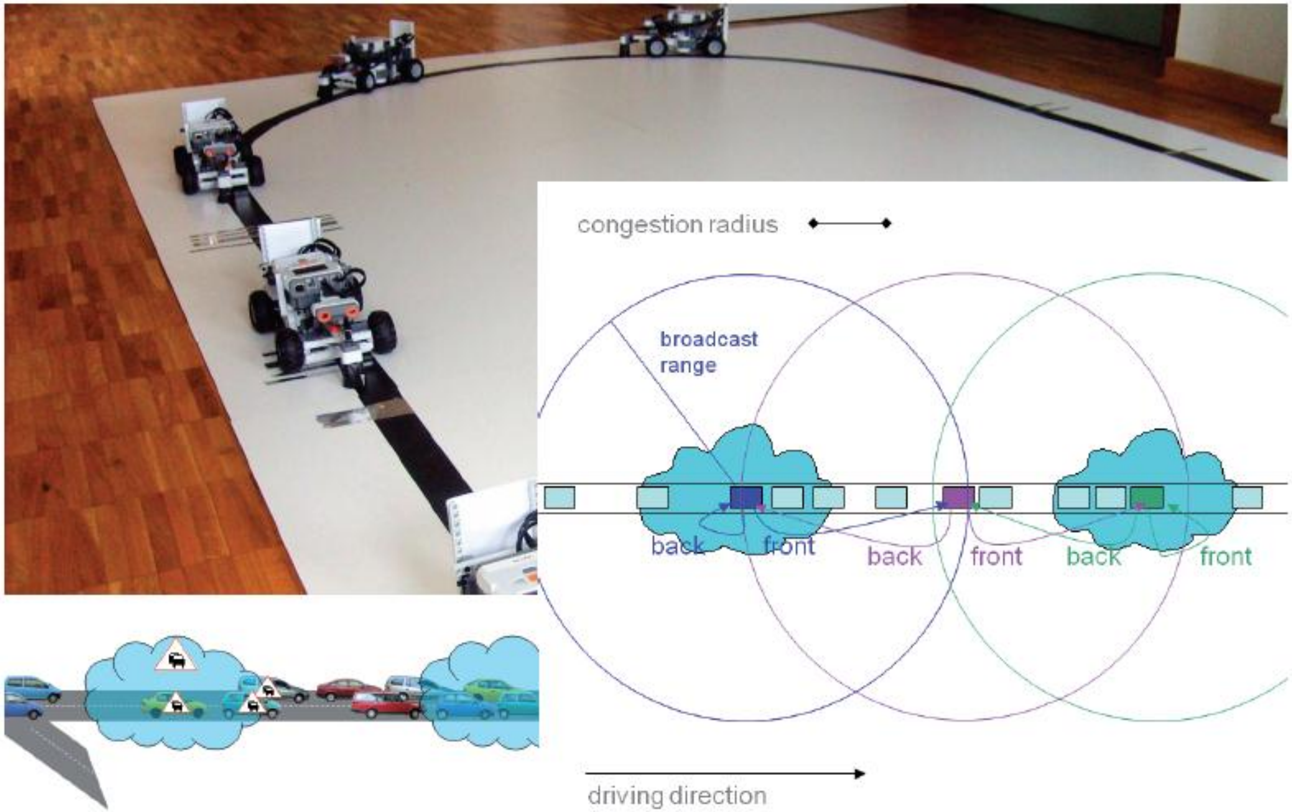




Principles, methods, and architectures for evolutionary self-adaptation to real-world problems (Platzner et al.)

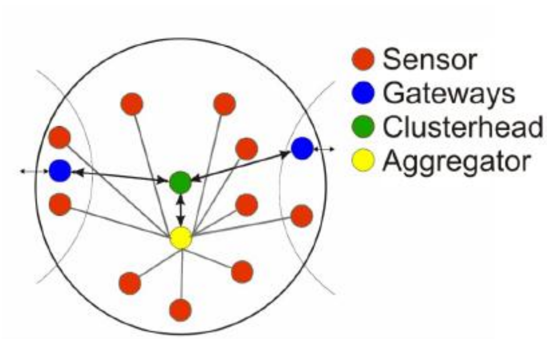
What did we achieve so far: Applications

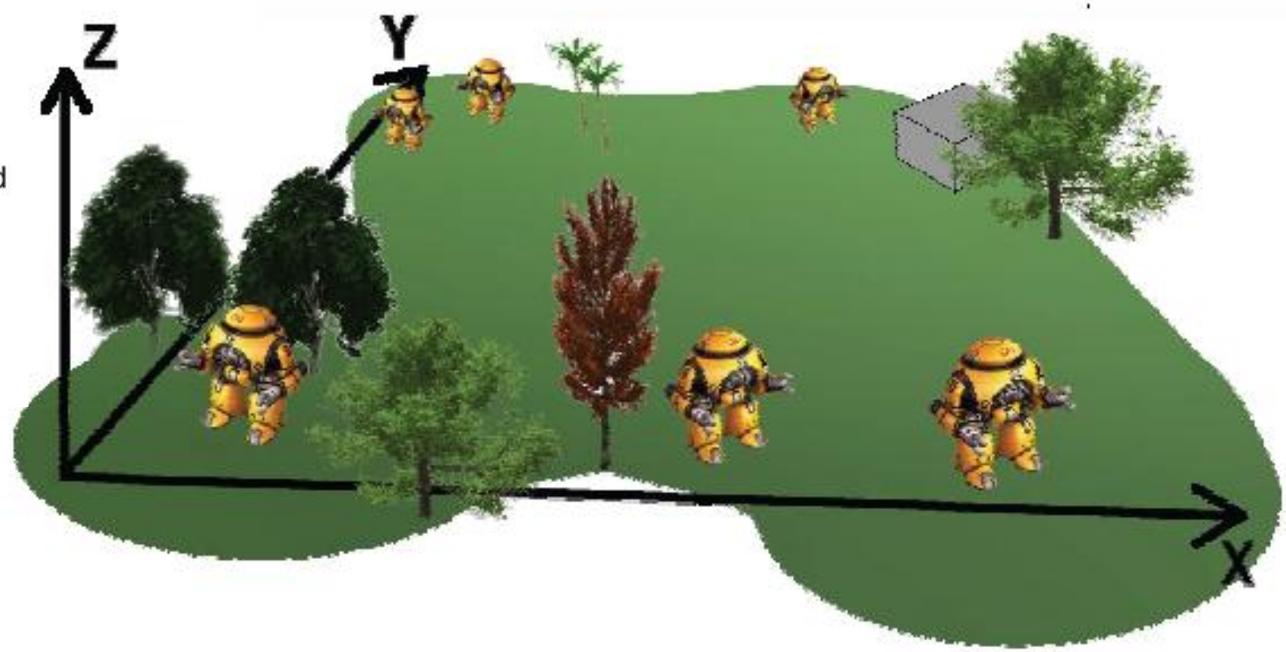
Demonstration of potential and relevance of architectures and tools with respect to demanding application scenarios



Realisation of hovering data clouds in large systems of mobile objects (Fekete, Fischer et al.)

What did we achieve so far: Applications



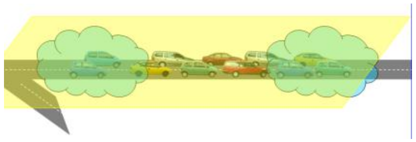


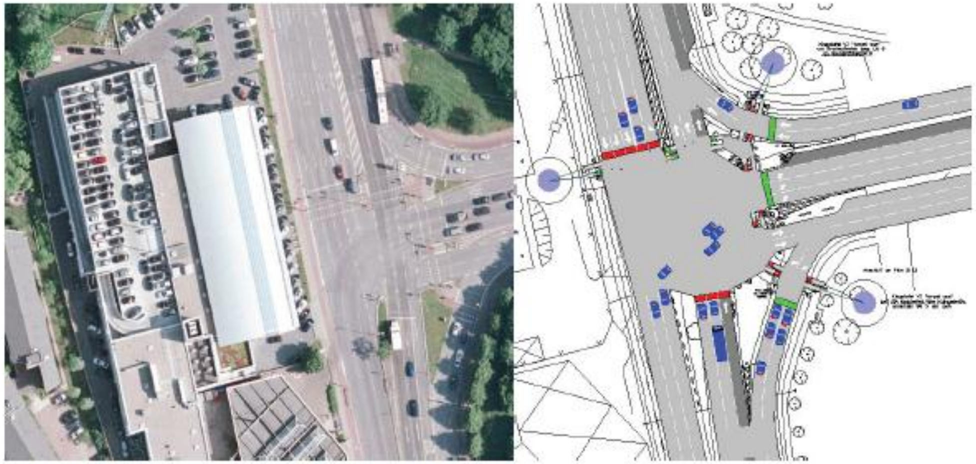
Local, distributed strategies for smart, self-organising robotic exploration teams (Meyer auf der Heide, Schindelhauer et al.)

Energy and resource aware self-organised communication and cooperation (smart teams, sensor networks)
(Meyer auf der Heide, Schindelhauer, Timmermann et al.)

Emergent strategies for optimisation of collaborative transmission schemes (Beigl, Decker et al.)

What did we achieve so far: Applications





Application of Organic Computing principles in traffic (traffic light control, progressive signal chains, organic information complexes, adaptable distributed strategies)
(Branke, Hähner, Müller-Schloer, Schmeck et al.) **(Fekete, Fischer et al.)**

Learning to find and track humans in video sequences and to recognise individuals (Würtz et al.)

What did we achieve so far: Applications

More Applications beyond SPP projects:

- Camera systems
- Energy management
- Control of off-highway machines
- Grid and Cloud Computing
- …

Lasting impact of SPP 1183?

- Fundamental insights on self-organization
- Generic architectures
- OC toolbox
- OC design principles
- OC in applications

→ Collection of papers on the lasting impact of Organic Computing

Some Related Research Activities

- Grand Challenges of Computer Engineering
- SPP 1500 Design and Architectures of Dependable Embedded Systems
- Hopefully: New SPP on Multi- and Many-core Architectures
- National Roadmap for Embedded Systems
- Software Platform for Embedded Systems - SPES
- Cyber Physical Systems

Grand Challenges in Computer Engineering

UWE BRINKSCHULTE

THEO UNGERER

ARCS TECHNICAL COMMITTEE OF VDE/ITG AND GI

Computer Engineering is everywhere

50-70 Million Lines of Code in Space Shuttle

Airbag

Pacemaker

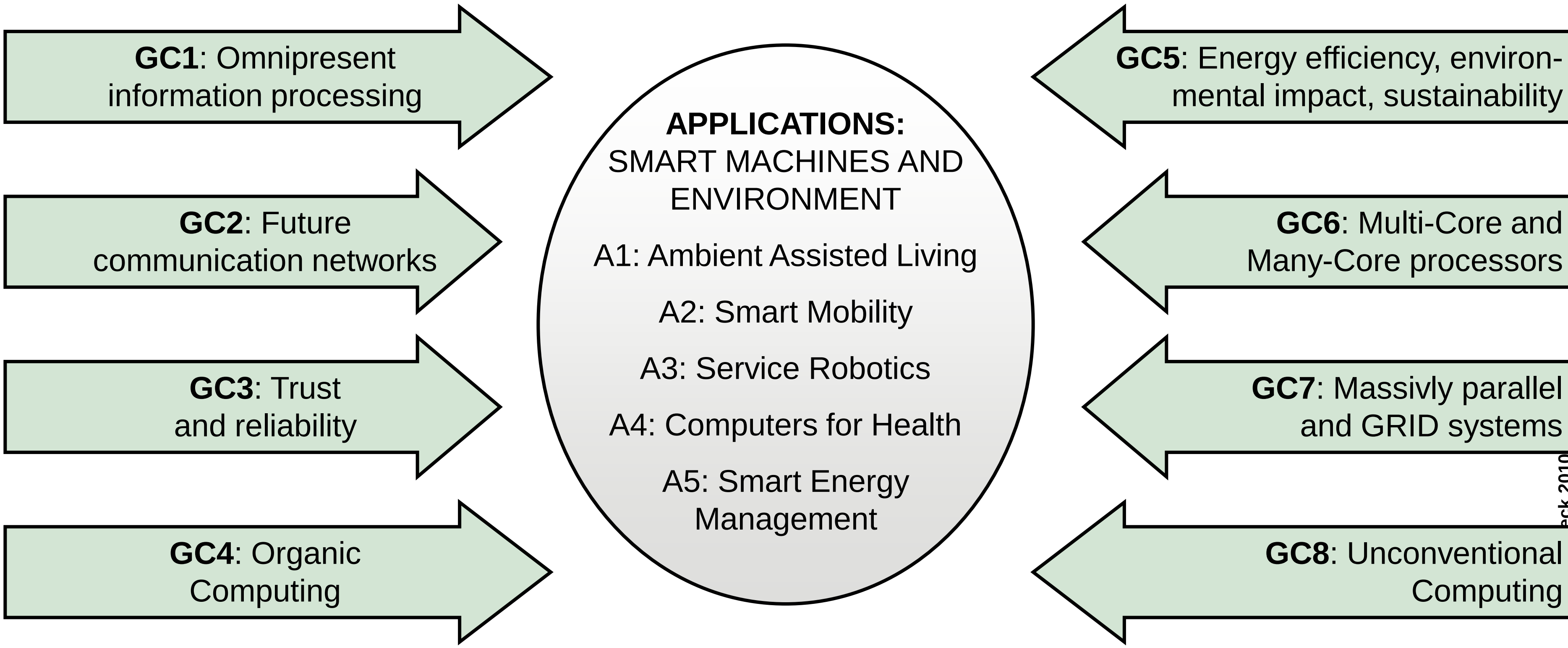
Computer Engineering is everywhere

Autonomous Cars

Smart Labels

Artifical Senses

Grand Challenges and Application Scenarios



National Roadmap for Embedded Systems (ZVEI)

- Presented at German IT Summit 2009
- Steering Committee:
  - Reinhold Achatz, Klaus Beetz, Siemens AG
  - Manfred Broy, Technische Universität München
  - Heinrich Dämbkes, EADS Deutschland GmbH
  - Werner Damm, OFFIS (Chair)
  - Klaus Grimm, Daimler AG
  - Peter Liggesmeyer, Fraunhofer IESE and Univ.Kaiserslautern
- 10 Theses
- 6 Research Focuses („Forschungsschwerpunkte“)

National Roadmap for Embedded Systems - Focuses

F1 Seamless Interaction:
Get needed information at the right time
(health care, crisis managment, intermodal logistics, smart shops)

F2 Autonomic Systems:
Providing critical functionality without human intervention (deep sea resource exploration, space missions)

F3 Distributed Realtime Systems:
Coordinated situation evaluation, problem solving in areas like crisis management, health care, car driving etc.

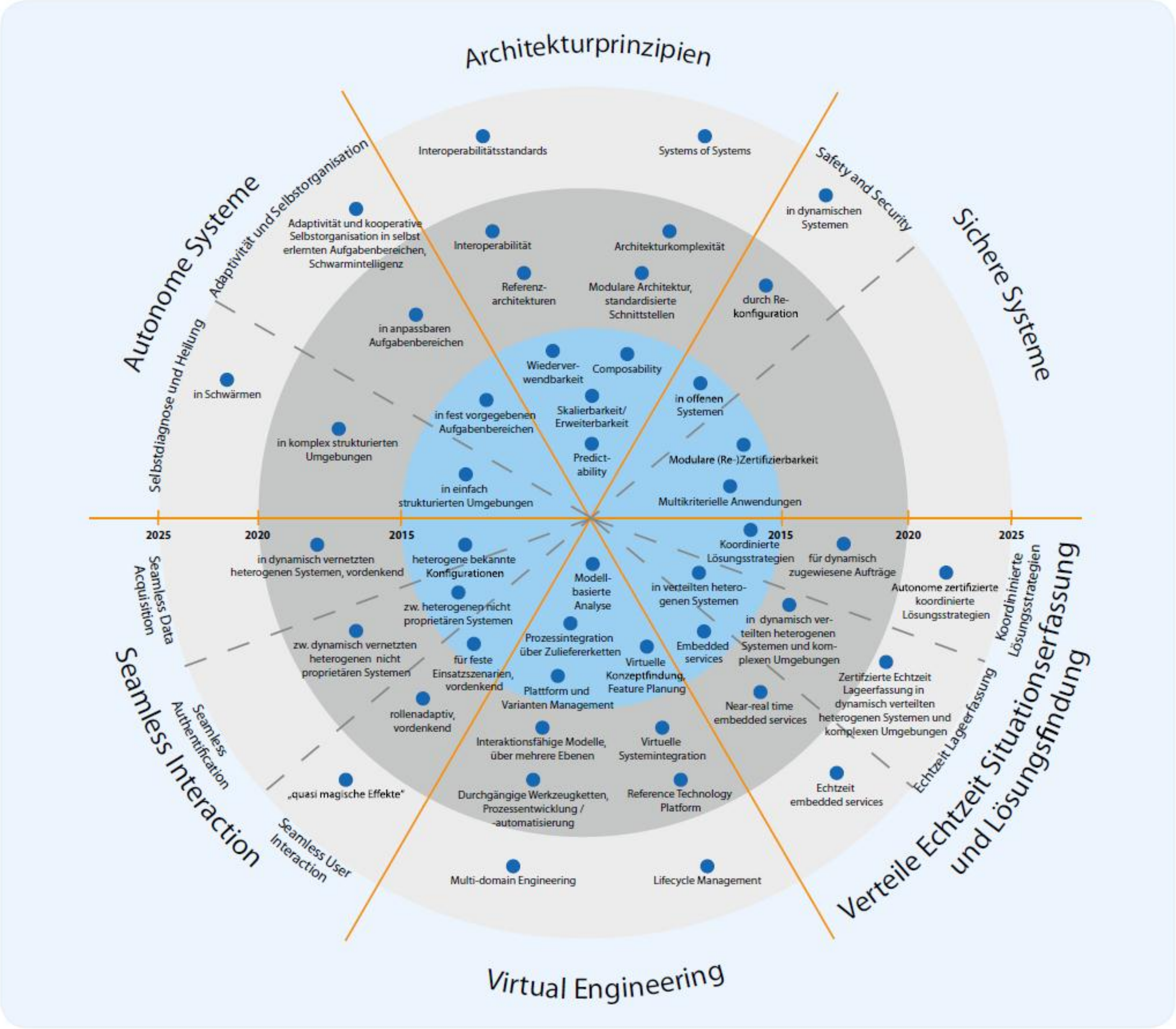
F4 Secure Systems:
Trust in embedded systems as prerequisite for the acceptance of complex, networked, embedded systems.

F5 Architectural Principles:
Standardized, manageable, generic architectures

F6 Virtual Engineering:
Improved design processes

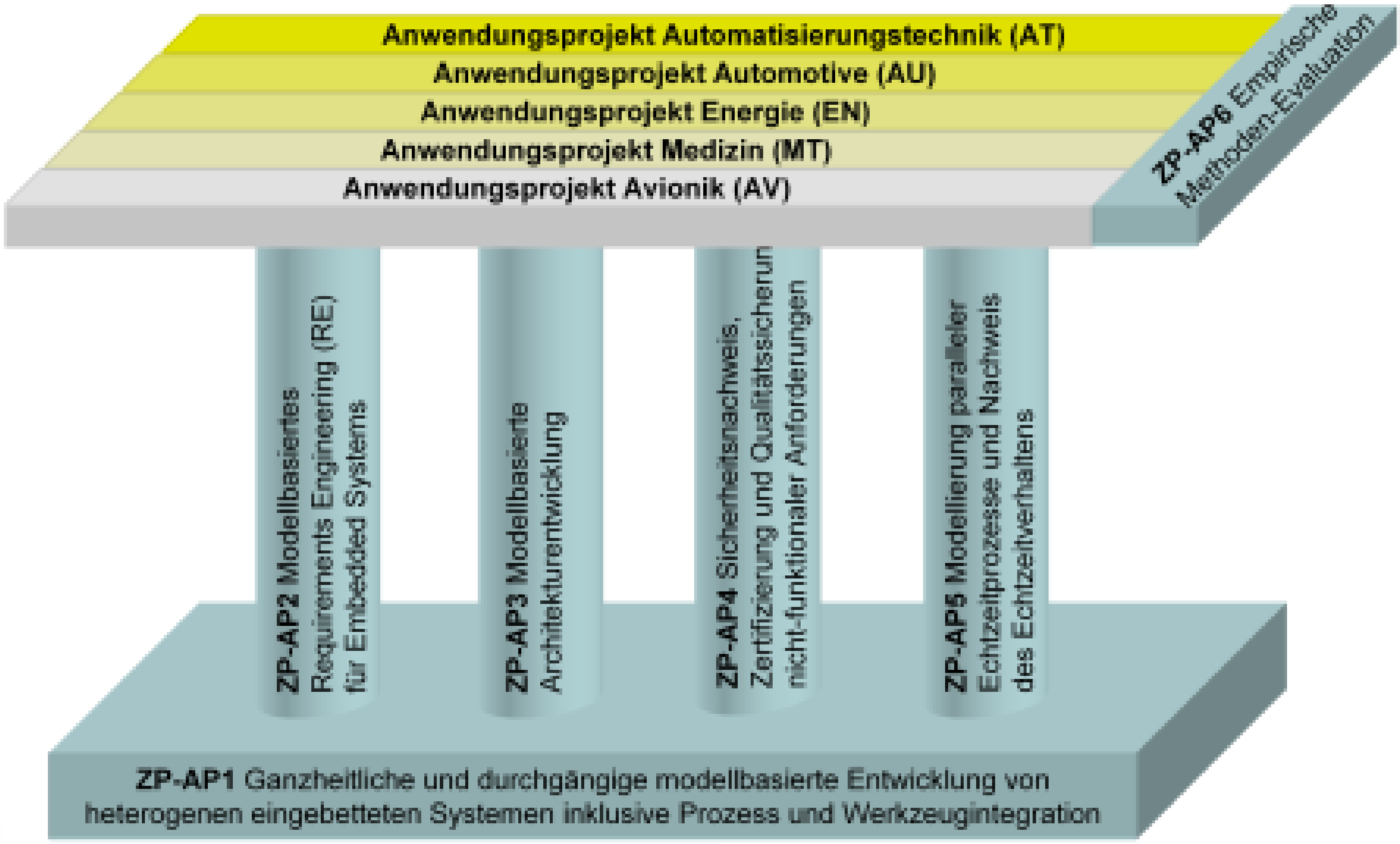
Requested Funds: 2,5 Bill. € within 10 years

National Roadmap for Embedded Systems



Software Platform for Embedded Systems – SPES 2020 (Broy et al.)

- Cooperative projects, 23 Mio € for a period of three years

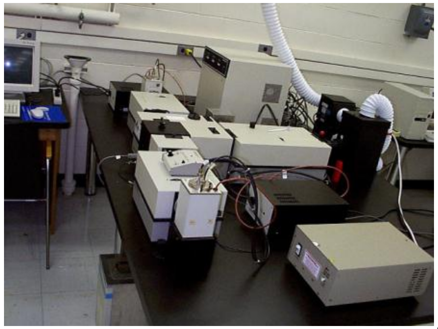


Cyber Physical Systems
(based on a poster of E.A. Lee, UCBerkeley
https://chess.eecs.berkeley.edu/pubs/653.html)

Cyber-Physical Systems (CPS): Orchestrating networked computational resources with physical systems.

"Networked computers have already changed the way humans communicate and manage information.

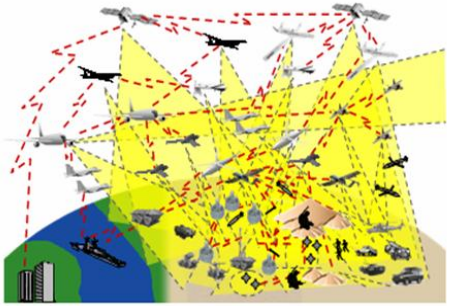
The change we envision is to the way humans manage their physical environment, including for example transportation, energy, health, and environmental quality.



This change requires computing and networking technologies to embrace not just information, but also physical dynamics. The impact of this change could well dwarf that of the information revolution."

The Vision:

Reliable and Evolvable Networked Time-Sensitive Computational Systems, Integrated with Physical Processes



Where CPS Differs from General-Purpose Software Systems

(based on a poster of E.A. Lee, UCBerkeley)

The software systems problem:

- Software systems are sets of interacting sequences of state transformations with the end objective of transforming data.

The CPS problem:

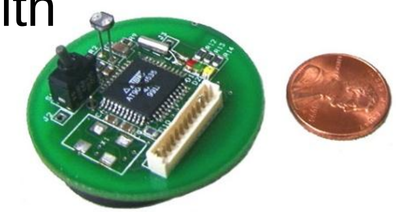
- CPS has the end objective of orchestrating physical processes. Timeliness, safety, reliability, security, privacy, and adaptability all take on a different character.

Where CPS Differs from Embedded Software

(based on a poster of E.A. Lee, UCBerkeley)

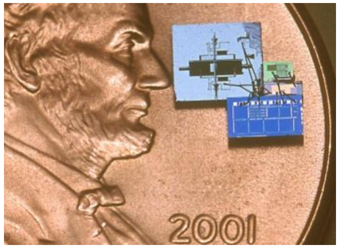
The embedded software problem:

- Embedded software is software on small computers. The technical problem is one of optimization (coping with limited resources).

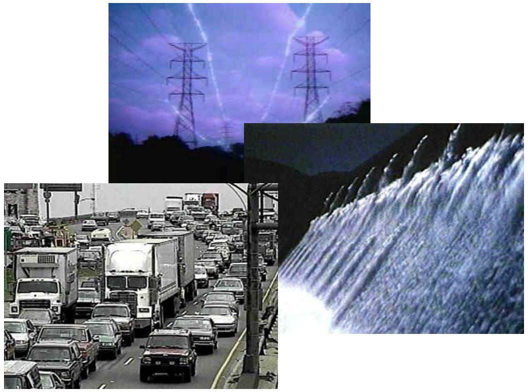


The CPS problem:

- Computation and networking integrated with physical processes.



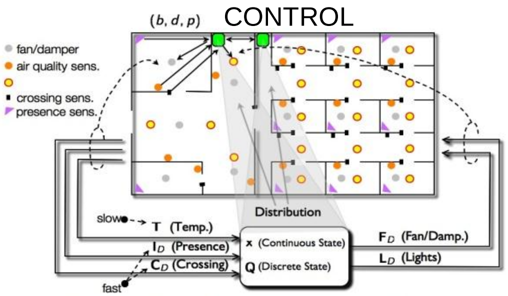
The technical problem is managing time and concurrency in networked computational systems.



Applications of Cyber-Physical Systems (CPS)

- medical devices and systems
- assisted living
- traffic control and safety
- advanced automotive systems,
- process control
- energy conservation
- environmental control
- avionics and aviation software
- instrumentation
- critical infrastructure (power, water)
- distributed robotics (telepresence)
- weapons systems
- distributed sensing
- command and control
- manufacturing
- smart structures
- biosystems (morphogenesis,…)
- communications systems

- integrated medical systems
- distributed micro power generation
- safe/efficient transportation
- military dominance
- economic dominance
- disaster recovery
- energy efficient buildings
- alternative energy
- pervasive adaptive communications
- more efficient financial networks
- distributed service delivery
- social networking and games
- …



Multi-scale, Multi-Agent, Potentially Distributed Hybrid Control System

Events on Cyber-Physical Systems (CPS)

- NSF Workshop on "Cyber-Physical Systems", October 16-17, 2006, Austin, TX.
- National Workshop on "High-Confidence Software Platforms for Cyber-Physical Systems (HCSP-CPS)", November 30 – December 1, 2006, Alexandria, VA.
- NSF Industry Round-Table on Cyber-Physical Systems, May 17, 2007, Arlington, VA.
- International Workshop, "From Embedded Systems to Cyber-Physical Systems: A Review of the State-of-the-Art and Research Needs", 4/08, St. Louis, MO.
- CPSWeek: San Francisco, April 13-16, 2009: 3 Conferences (RTAS/ISPN/HSCC)
- **NSF CPS Program, $30 million over 5 years, started September, 2009**
- CPS Week: Stockholm, April 12-16, 2010: 5 Conferences:
  - Real-Time and Embedded Technology and Applications Symposium (RTAS)
  - International Conference on Information Processing in Sensor Networks (IPSN)
  - International Conference on Hybrid Systems: Computation and Control (HSCC)
  - Languages, Compilers and Tools for Embedded Systems (LCTES)
  - **ACM/IEEE First International Conference on Cyber-Physical Systems (ICCPS)**

Conclusion

- SPP OC has produced a solid basis for further research and development projects.
- OC technologies are applicable to a range of interesting application areas (including those of CPS)
- Research programs may be focused on application scenarios where OC is an enabling technology
  - Traffic systems
  - Energy system
  - Health
  - Virtualisation (Cloud Computing,…)
  - …
- New research initiatives (CPS, NRMES, SPES) could benefit from Organic Computing concepts.
- New Dagstuhl Seminar on OC: Feb. 6-11, 2011

→ See Posters during coffee break!

Thanks for your attention!

Questions?

Challenges for research on OC systems

- Learning:
  - Potential of online- and offline learning
  - Collaborative learning
- Coordination and collaboration
  - Typical patterns of c & c in OC systems
  - Benefits, necessity of c & c
- Design
  - Finding the right balance between explicit design and degrees of freedom
  - Finding the right separation of concerns in hierarchical OC systems
- Cognition
  - Finding out “the needs of human users” (or, of the environment).
  - Detecting anomalies, distinguishing the “good” from the “bad”.

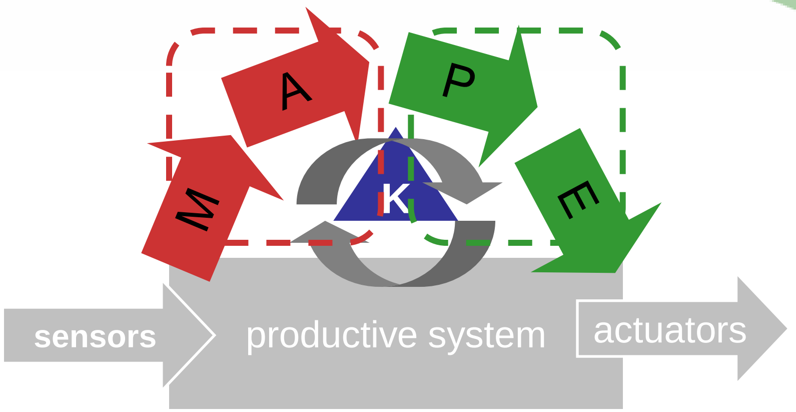
Challenges for research on OC systems (2)

- Control
  - Finding the right balance between “SO” and “control” phases.
  - Does “control by objectives” work?
- Trust
  - Can OC-systems be trustworthy?
  - Trust engineering?
- Assessment
  - Can there be “service level agreements” with guaranteed performance for OC-systems?
  - Benefits of OC versus “standard” designs?
  - Benchmark applications for OC
- …

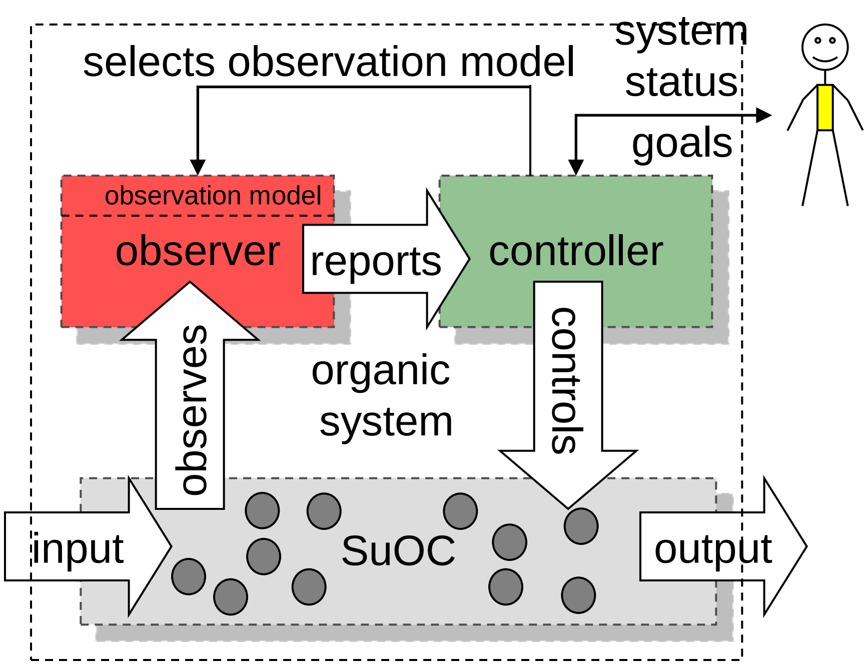
(Generic) Concepts for Control of SO-Systems

- IBM's **MAPE cycle** for autonomic computing
  - – Monitor
  - – Analyze
  - – Plan
  - – Execute
  - – Knowledge

  (called "autonomic element")



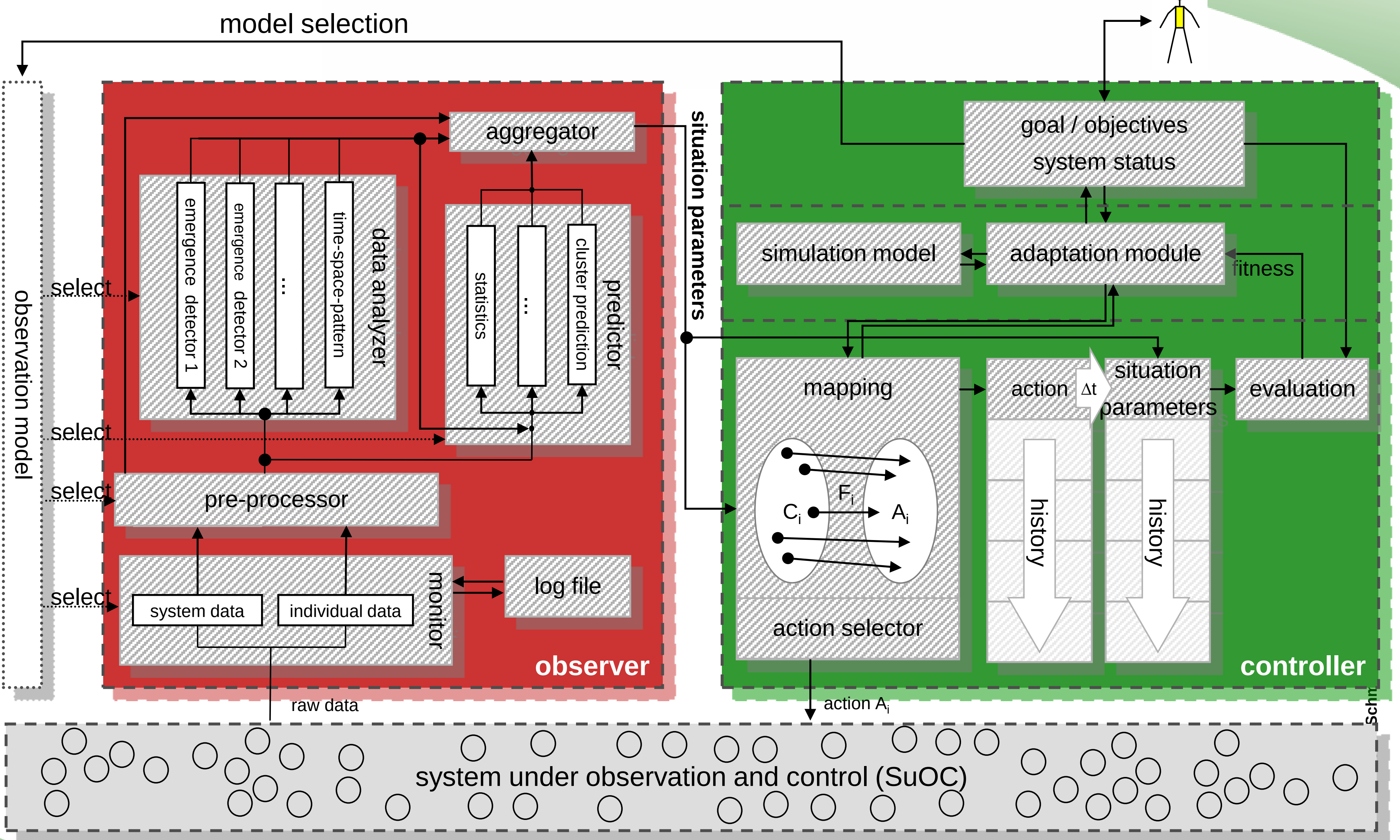
- System under observation and control (SuOC)
  - – A set of interacting elements/agents.
  - – Does not depend on the existence of observer/controller.
- Distributed and/or central **observer/controller-architecture**
  - – Driven by external goals
- Multilevel organization



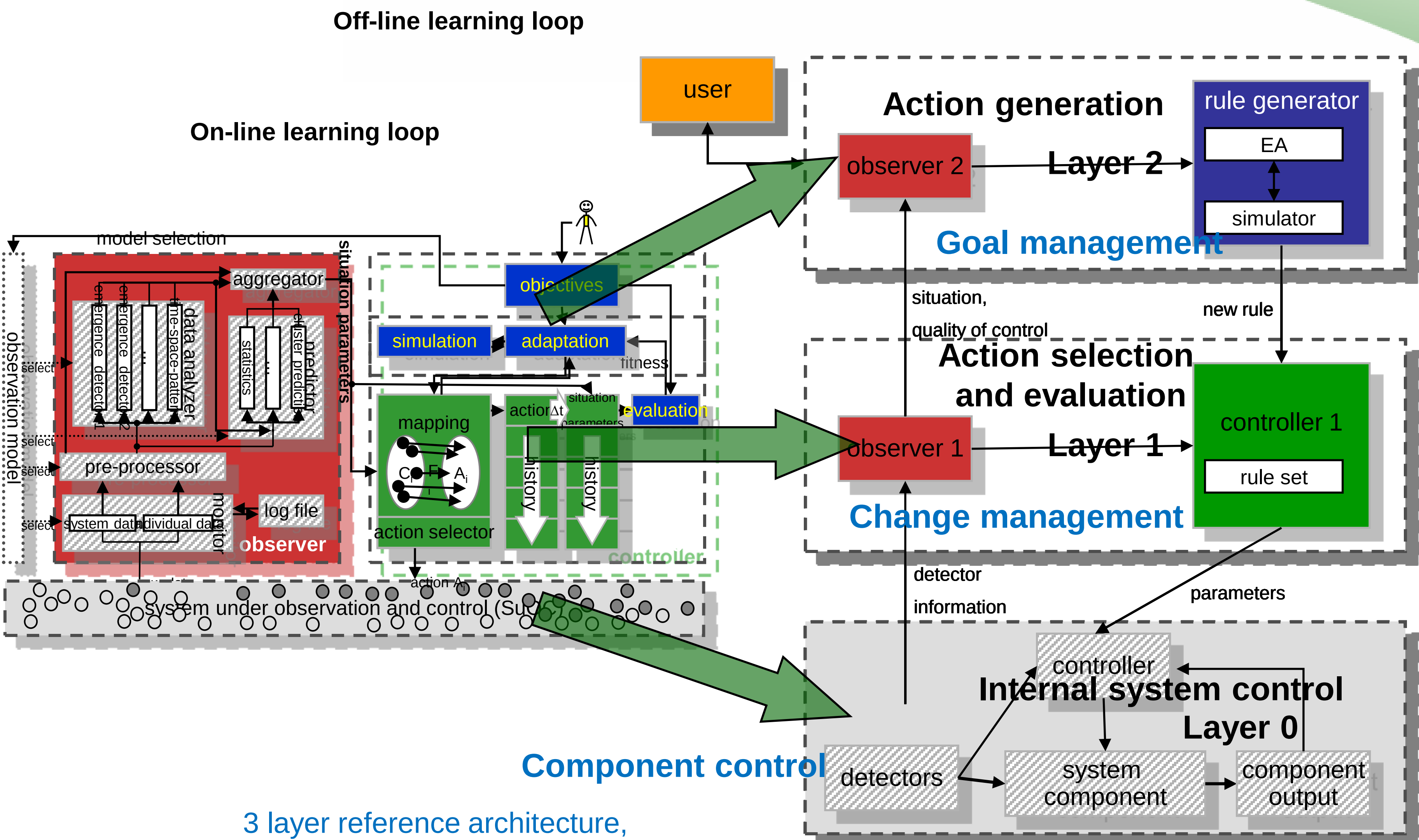
© H. Schmeck 2010

Generic O/C-Architecture

J.Branke, M.Mnif, C. Müller-Schloer, U. Richter, H. Schmeck 2006



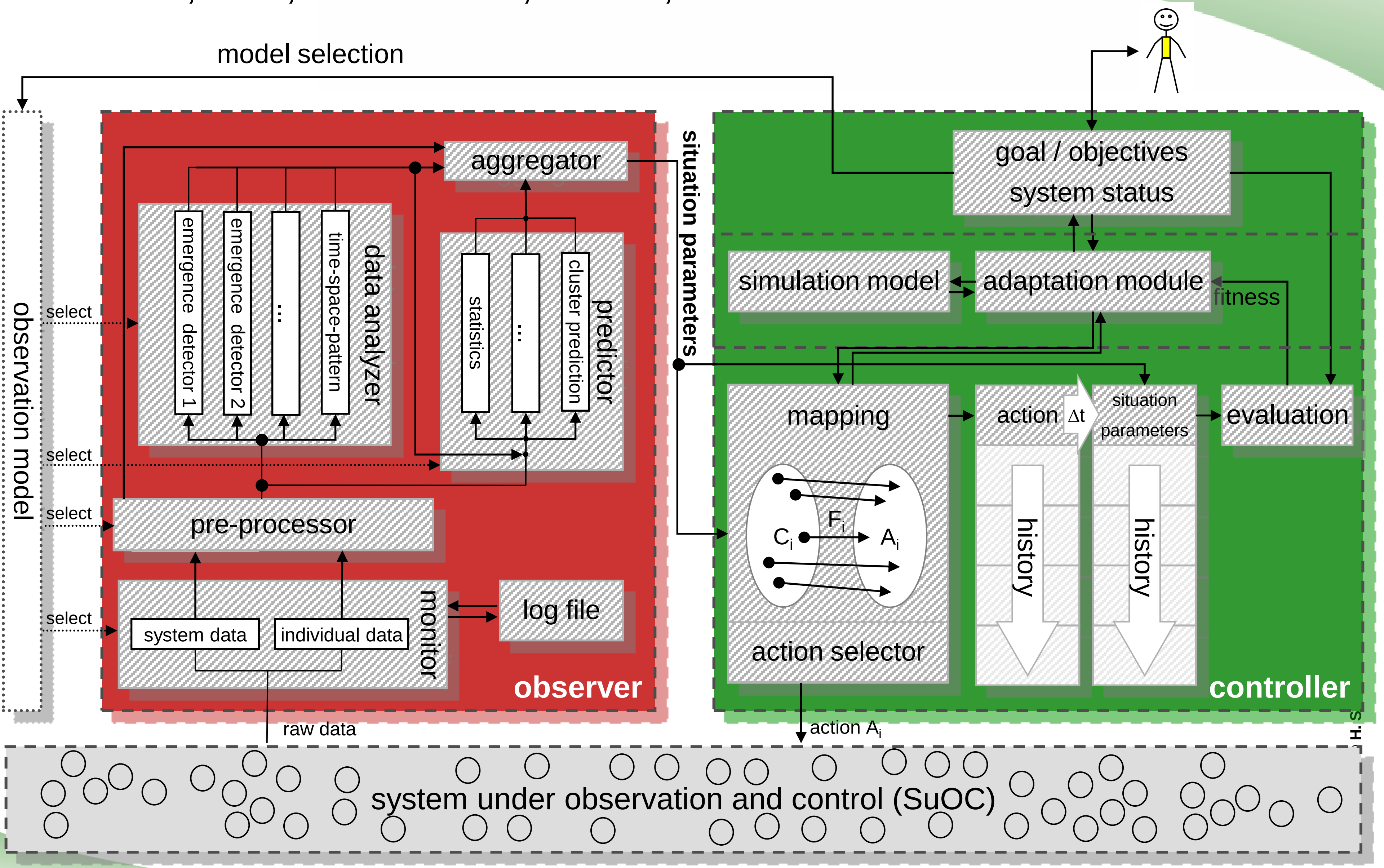
Different view on O/C-architecture



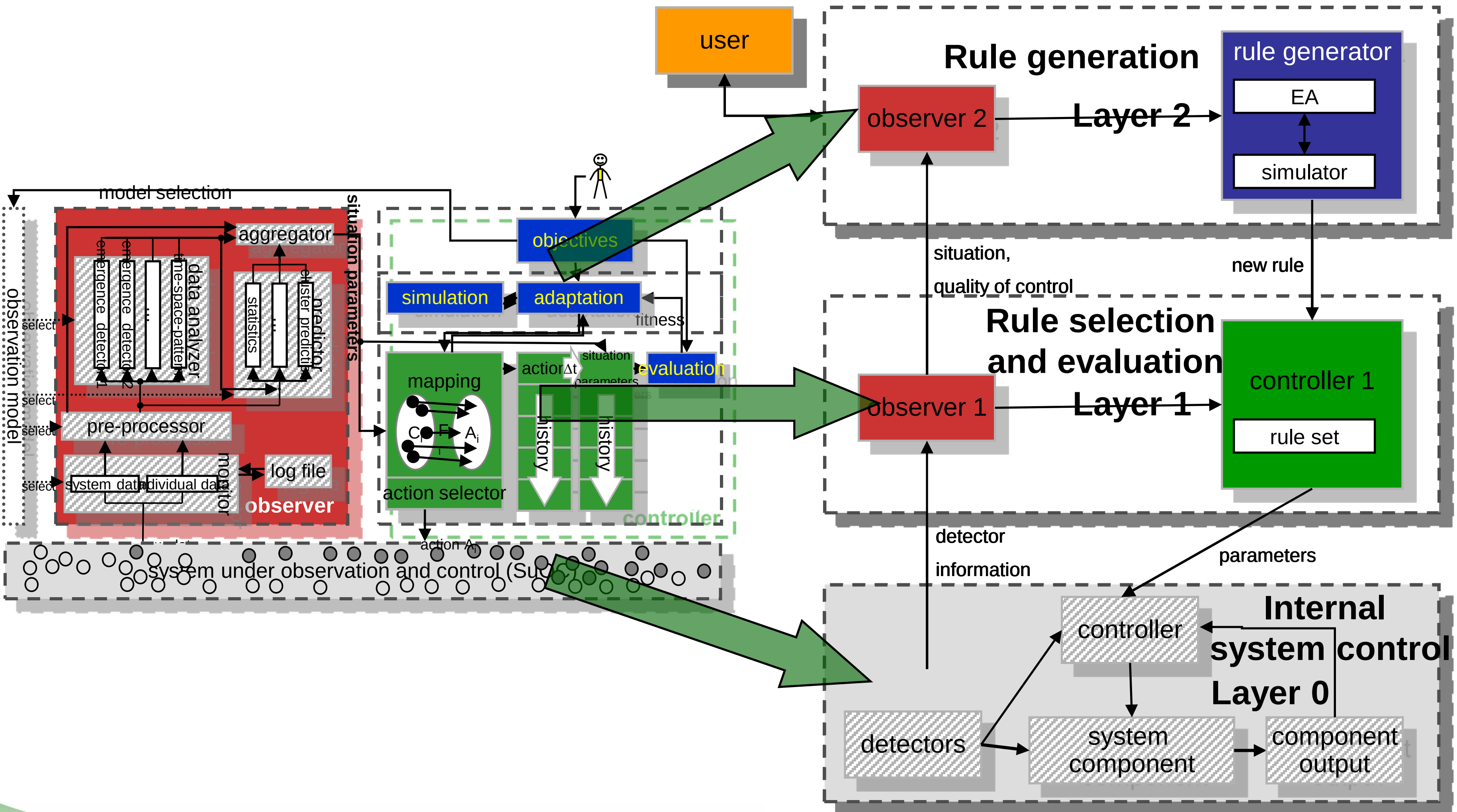
3 layer reference architecture,
Jeff Magee, (DagSEfSAS 2008)

Generic O/C-Architecture

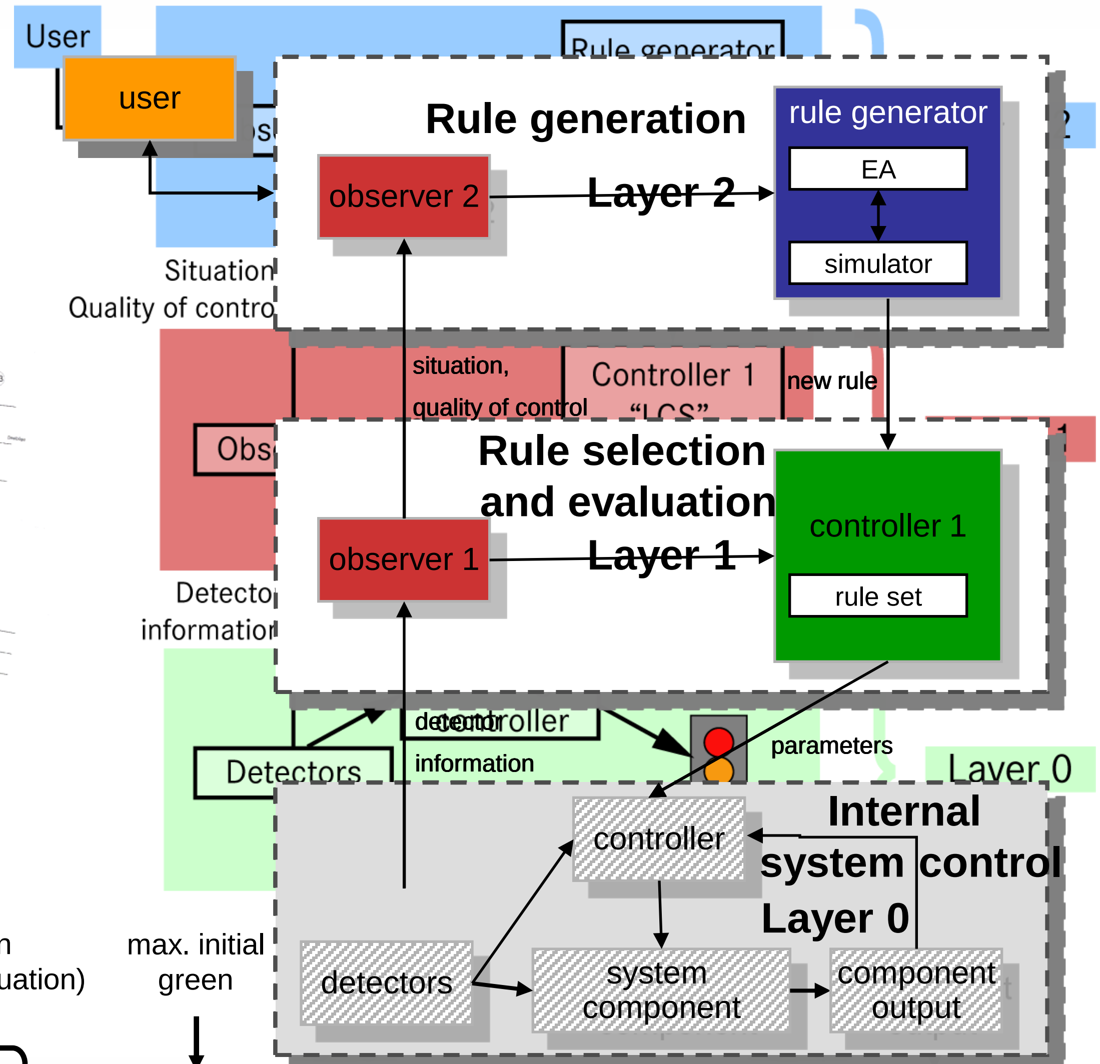
J.Branke, M.Mnif, C. Müller-Schloer, U. Richter, H. Schmeck 2006



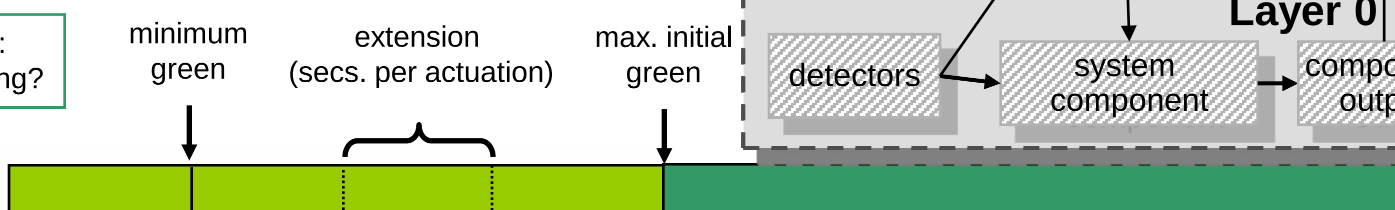
Different view on O/C-architecture



Example Application: Organic Traffic Control



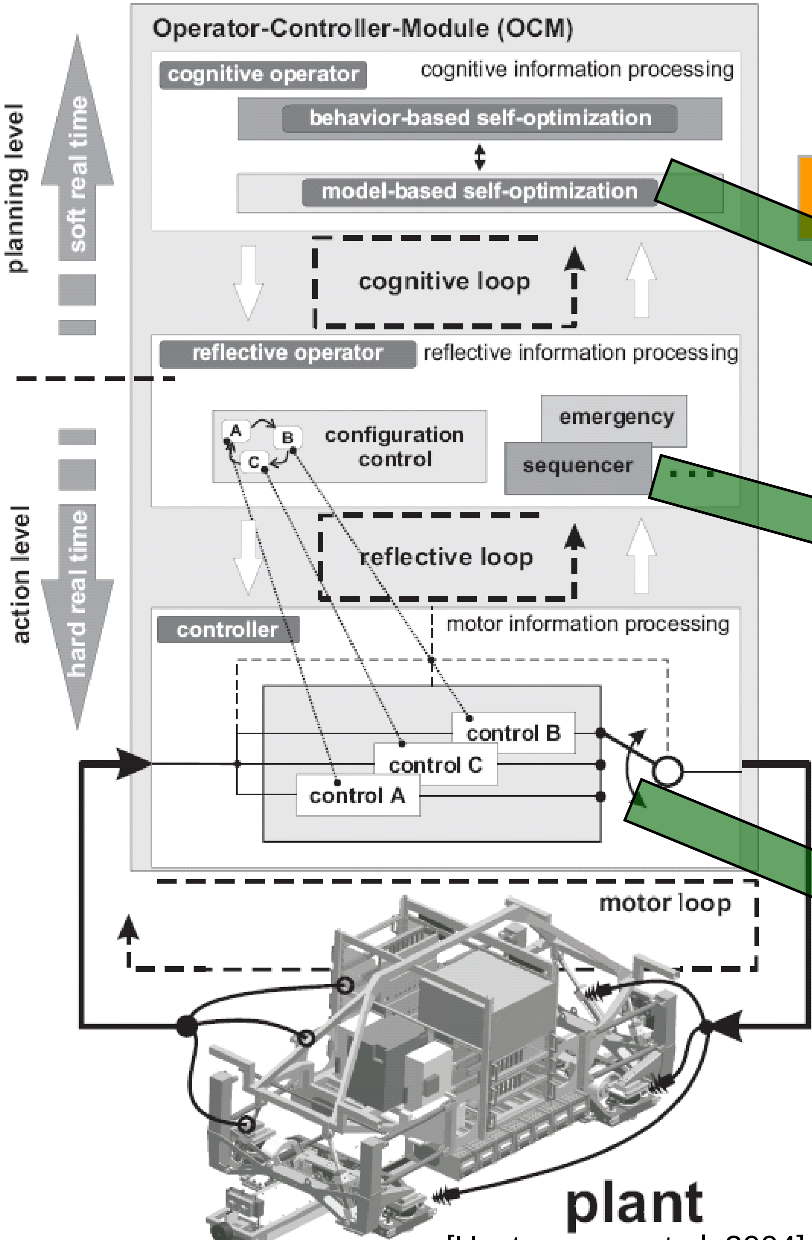
Recall:
car waiting?



© H. Schmeck 2010

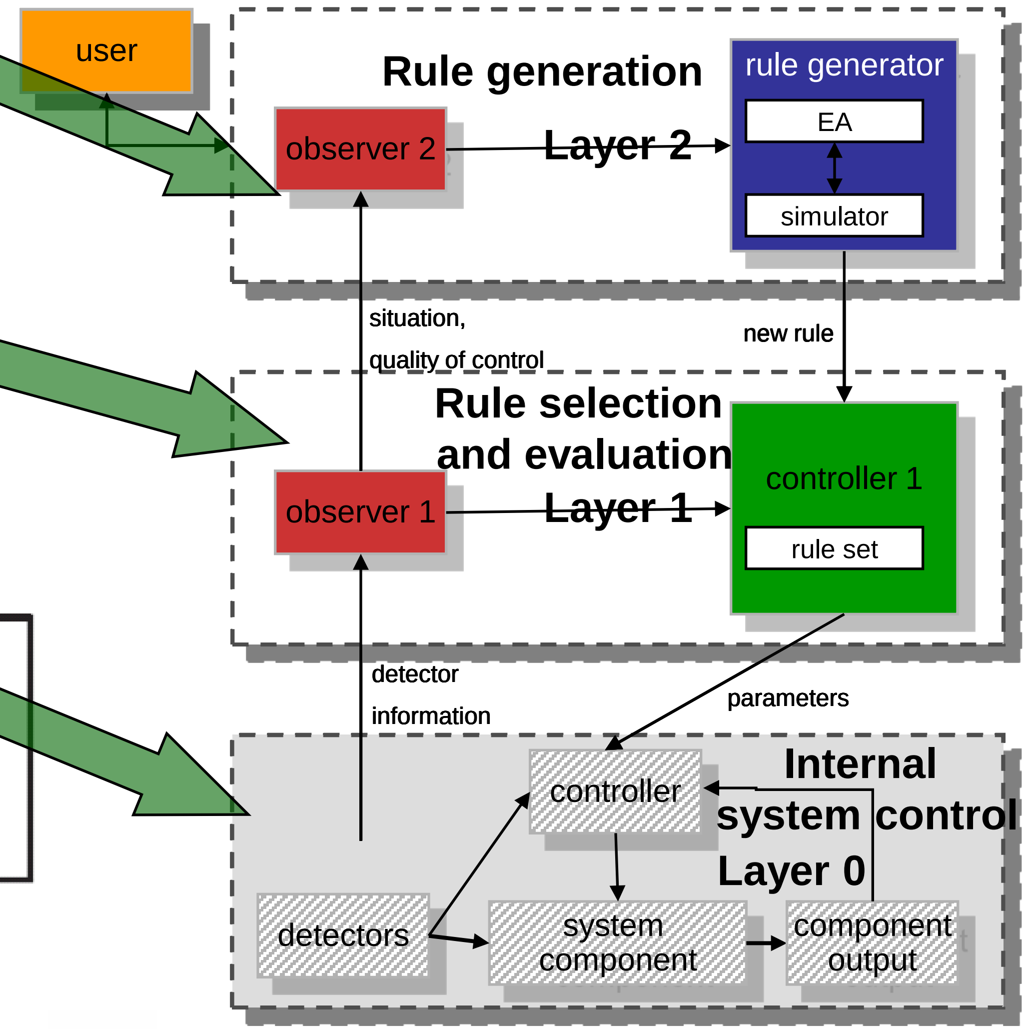
Operator-Controller Module

SFB 614 Paderborn

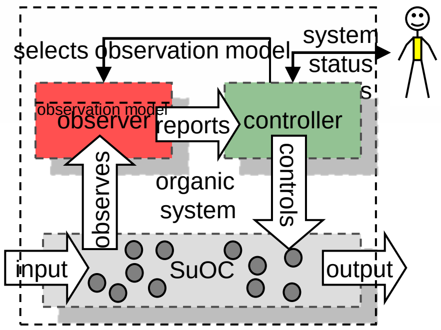


[Hestermeyer et al. 2004]

O/C-architecture



Remarks on O/C architecture



- Flexible template for observing, analysing, and influencing system behaviour.
- The potential of online and offline learning (Layers 1 and 2) should be explored in more application scenarios.
- The potential of modifying the model of observation and its consequences for controlling the system should be explored.
- There will be a roadmap of O/C architectures with growing capabilities, so far, only rather simple versions have been investigated.
- There is a need for more explicit support of communication and cooperation on the O/C-level(s).

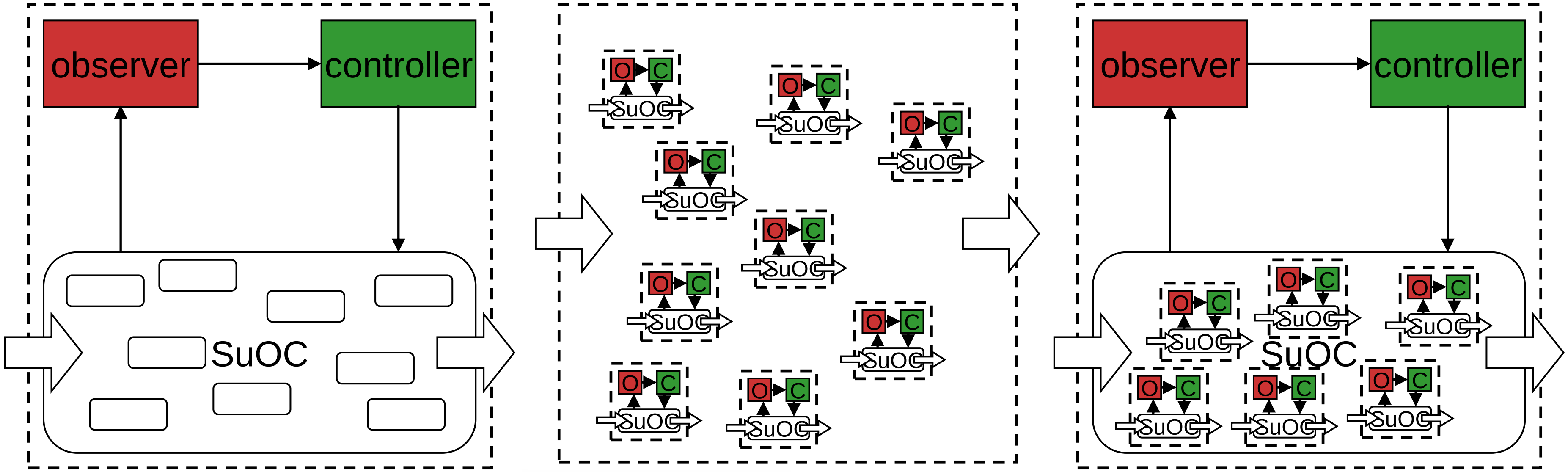
Realisation of OC systems

1. **Central**: One observer/controller for the whole system.
2. **Distributed**: An observer/controller on each system component.
3. **Multi-level**: An observer/controller on each system element as well as one for the whole system.

- adaptive
- top-down

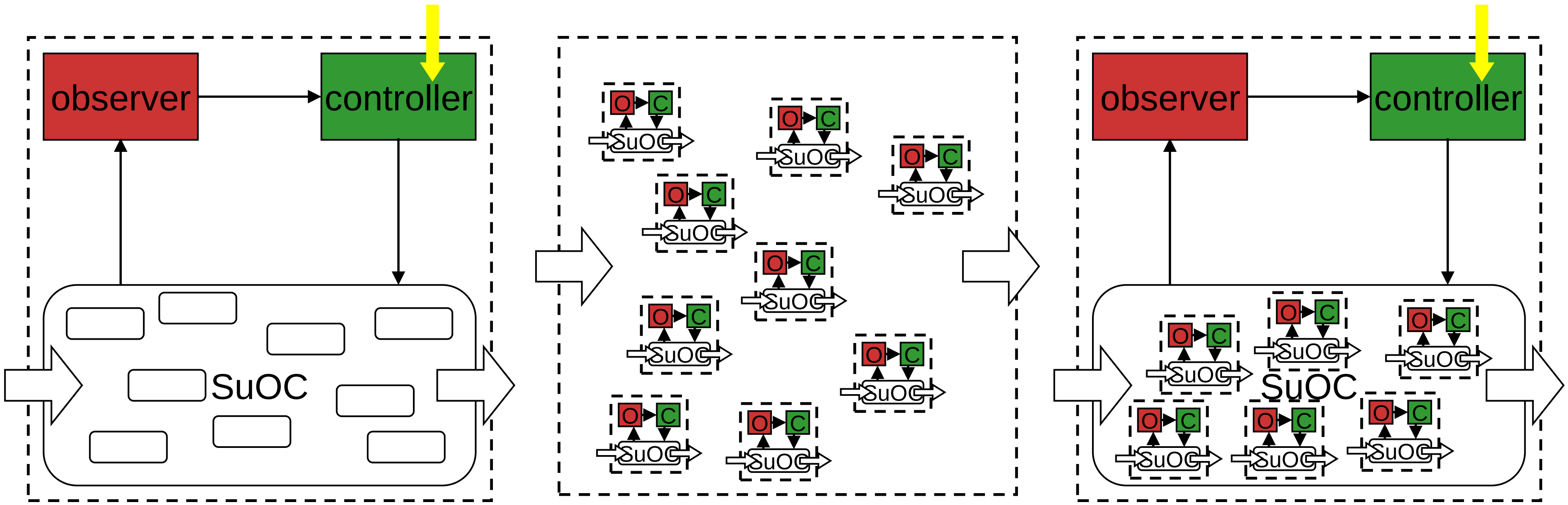
- self-organising
- bottom-up
- emergent control

- controlled self-organising
- bottom-up / top-down



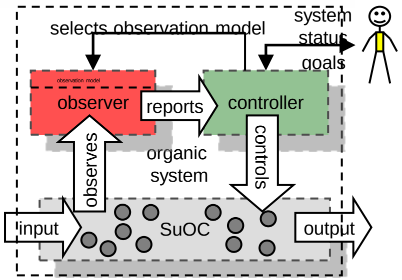
Types of control actions

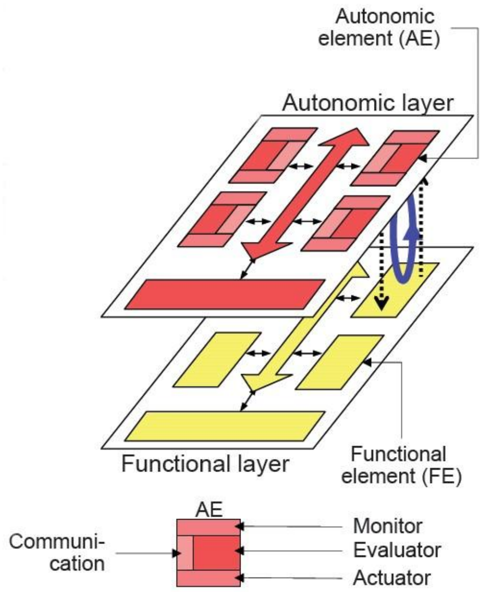
1. **Control the environment** (e.g. speed limit in traffic)
2. **Control the communication** (messages, addresses, neighborhoods,…)
3. **Control the local behavior of components** (reconfigure HW, update software, modify skills, set new local objectives,..)



Status:
What did we achieve so far?

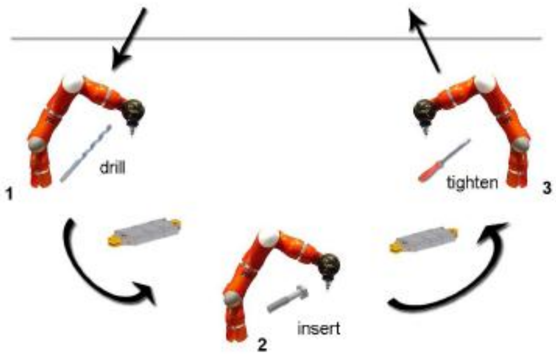
- Characterization and classification of emergence in SO-Systems (entropy-based)
  (Mnif, Müller-Schloer, Sick)
- Characterization of basic properties of OC systems (autonomy, controlled self-organisation, adaptivity, robustness, flexibility,…)
  (Mühl, Müller-Schloer, Ortmeier, Schmeck, …)
- Design of a generic O/C-architecture
  (Branke, Mnif, Müller-Schloer, Richter, Schmeck,…)
- Design of generic OC-middleware components (OCμ, Artificial Hormone System – AHS)
  (Ungerer et al., Becker, Brinkschulte, Henkel, Karl)
- Autonomic/Organic Systems on Chip
  (Herkersdorf, Rosenstiel et al.)





What did we achieve so far?

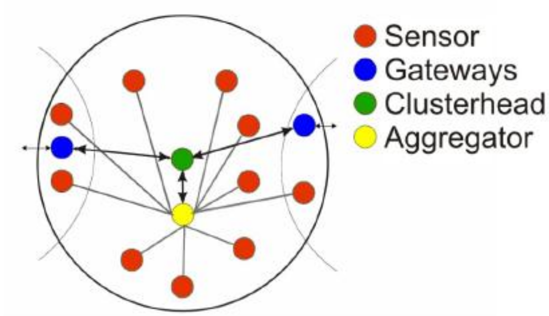
- Safety analysis and verification of OC-Applications
  (Reif et al.)



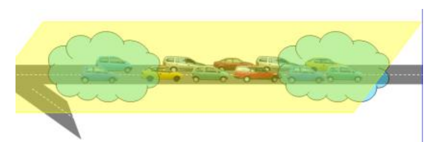
- Formal definition and modelling of self-x systems; self-organizing algorithms and development tools for OC applications.
  (Heiß, Mühl, Weis)

- Design of a self-configuring and self-healing robot,
  (Brauckmann, Maehle et al.)

What did we achieve so far?



- Embedded performance analysis of OC-systems using a variant of the O/C-architecture *(Ernst)*
- Energy and resource aware self-organized communication and cooperation (smart teams, sensor networks) *(Timmermann et al., Meyer auf der Heide et al.)*
- Artificial immune systems, self-protection, self-healing *(Trumler et al., Timmermann)*
- Systematic investigation of self-organizing systems in nature (swarms of bees, bio-chemical information processing) *(Middendorf, Dittrich)*
- Application of OC principles in traffic (traffic light control, progressive signal chains, hovering data clouds, organic information complexes, Adaptable Distributed Strategies) *(Branke et al.; Fekete, Fischer)*



What did we achieve so far?

- Concept of Marching Pixels (MP), emergent algorithms, emergent computing *(Fey)*
- Principles, methods, and architectures for evolutionary self-adaptation to real-world problems *(Platzner)*
- Principles of collaboration, coordination and learning in multi-agent systems *(Branke, Hähner, Müller-Schloer, Schmeck,…)*
- Use of Classifier Systems for online learning *(Prothmann, Richter, Rochner, Zeppenfeld)*
- Analysis of emerging behaviour in evolving societies of learning autonomous systems *(Rammig et al.)*
- Autonomic learning of functional knowledge in sensor systems *(Sick)*
- Learning to find and track humans in video sequences and to recognize individuals *(Würtz)*