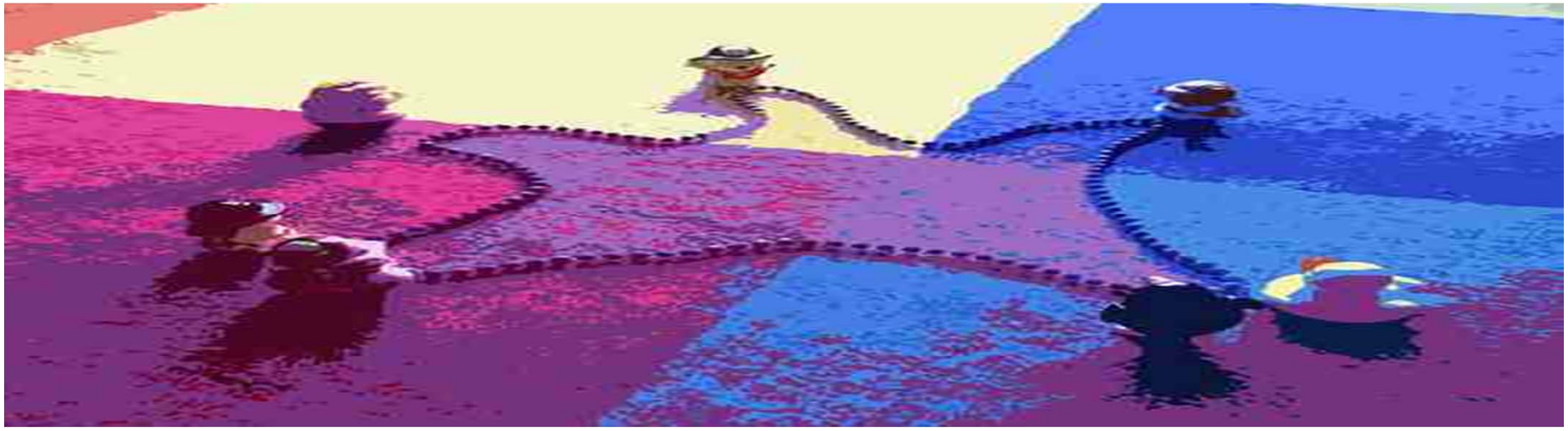
10th Colloquium SPP OC
Observation and Control of Collaborative Systems (OCCS)

Learning Architectures for Collaborative Systems



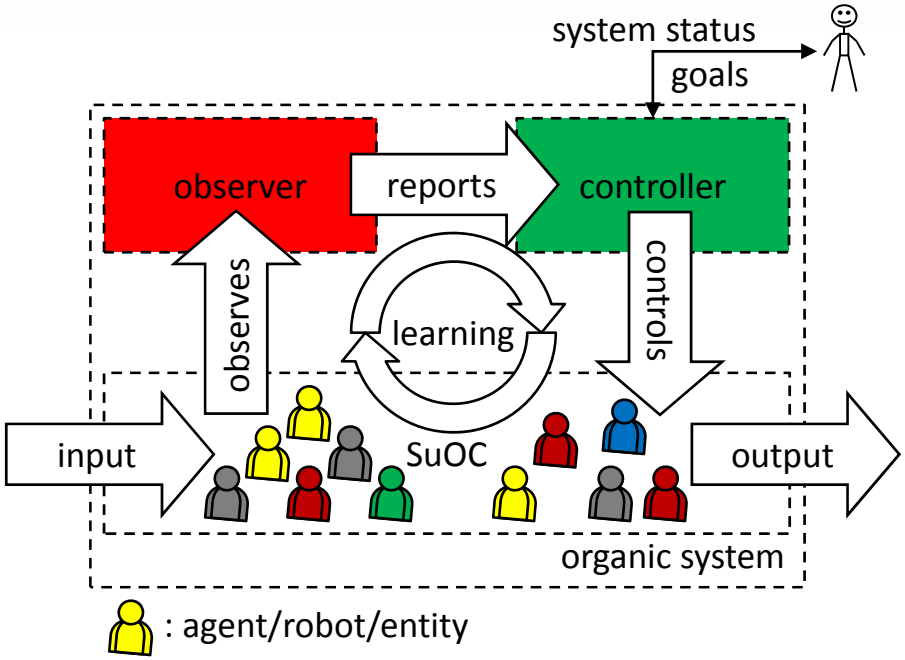
Jürgen Branke, Nugroho Fredivianus, Urban Richter, Hartmut Schmeck
Karlsruhe Institute of Technology

Emre Cakar, Jörg Hähner, Christian Müller-Schloer
Leibniz Universität Hannover

© C. Müller-Schloer, H. Schmeck, J. Branke, J. Hähner, M. Mnif, U. Richter, E. Cakar, N. Fredivianus 2005-2010

Generic O/C Architecture

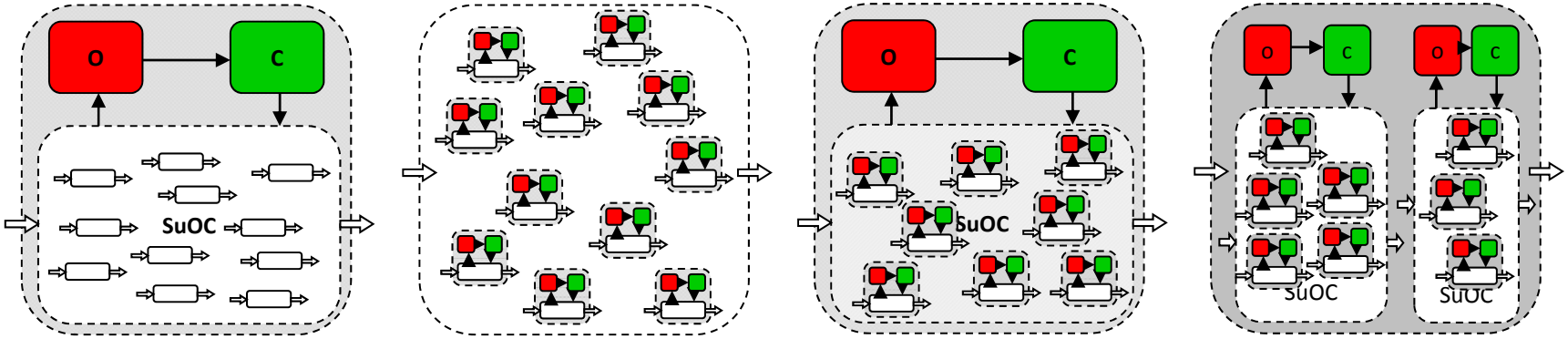
- From the Phase I
  - Development of the generic observer/controller architecture
  - **Observer**: Monitors system state and dynamics and quantifies them
  - **Controller**: Influences the system under observation and control (SuOC)



© C. Müller-Schloer, H. Schmeck, J. Branke, J. Hähner, M. Mnif, U. Richter, E. Cakar, N. Fredivianus 2005-2010

O/C Architectural Variants

- From the Phase II

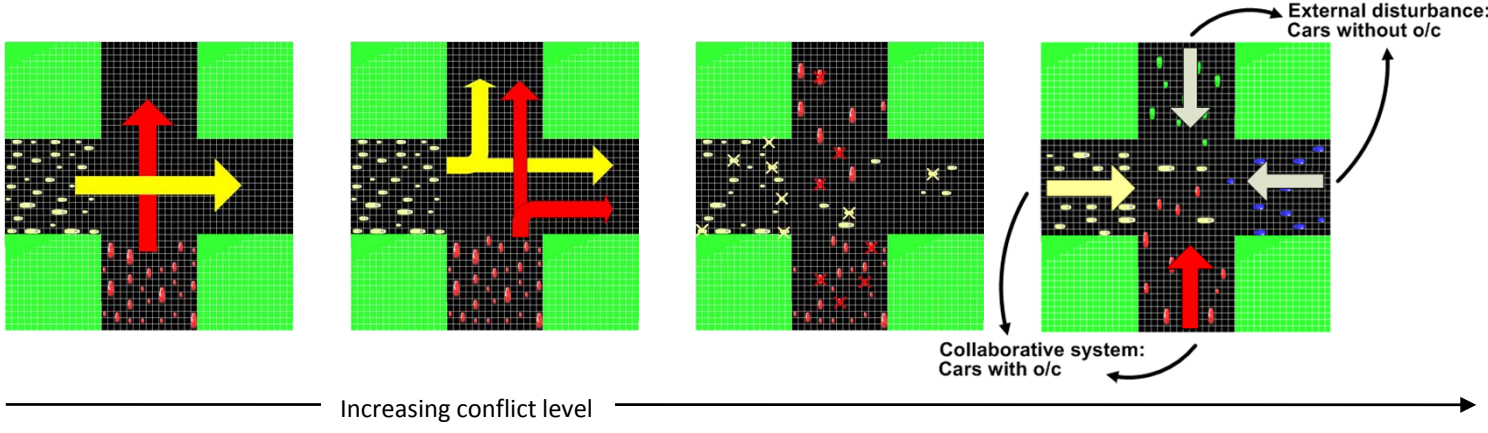


  - **Central**: One observer/controller for the whole system
    - *Example scenarios: chicken, elevator, or off-highway machines*
  - **Distributed**: An observer/controller on each agent
    - *Example scenarios: Indian junction, predator/prey pursuit, or cleaning robots*
  - **Hierarchical or multi-level**: Multi-observer/controllers on each technical unit and at the higher level (or levels)
    - *Example: Regional manager for organic traffic control*

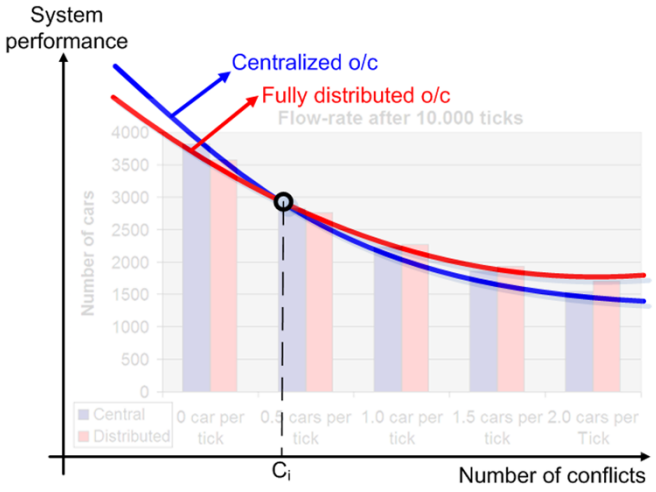
© C. Müller-Schloer, H. Schmeck, J. Branke, J. Hähner, M. Mnif, U. Richter, E. Cakar, N. Fredivianus 2005-2010

Centralised vs. Distributed

- System performance is measured by traffic-flow rate.



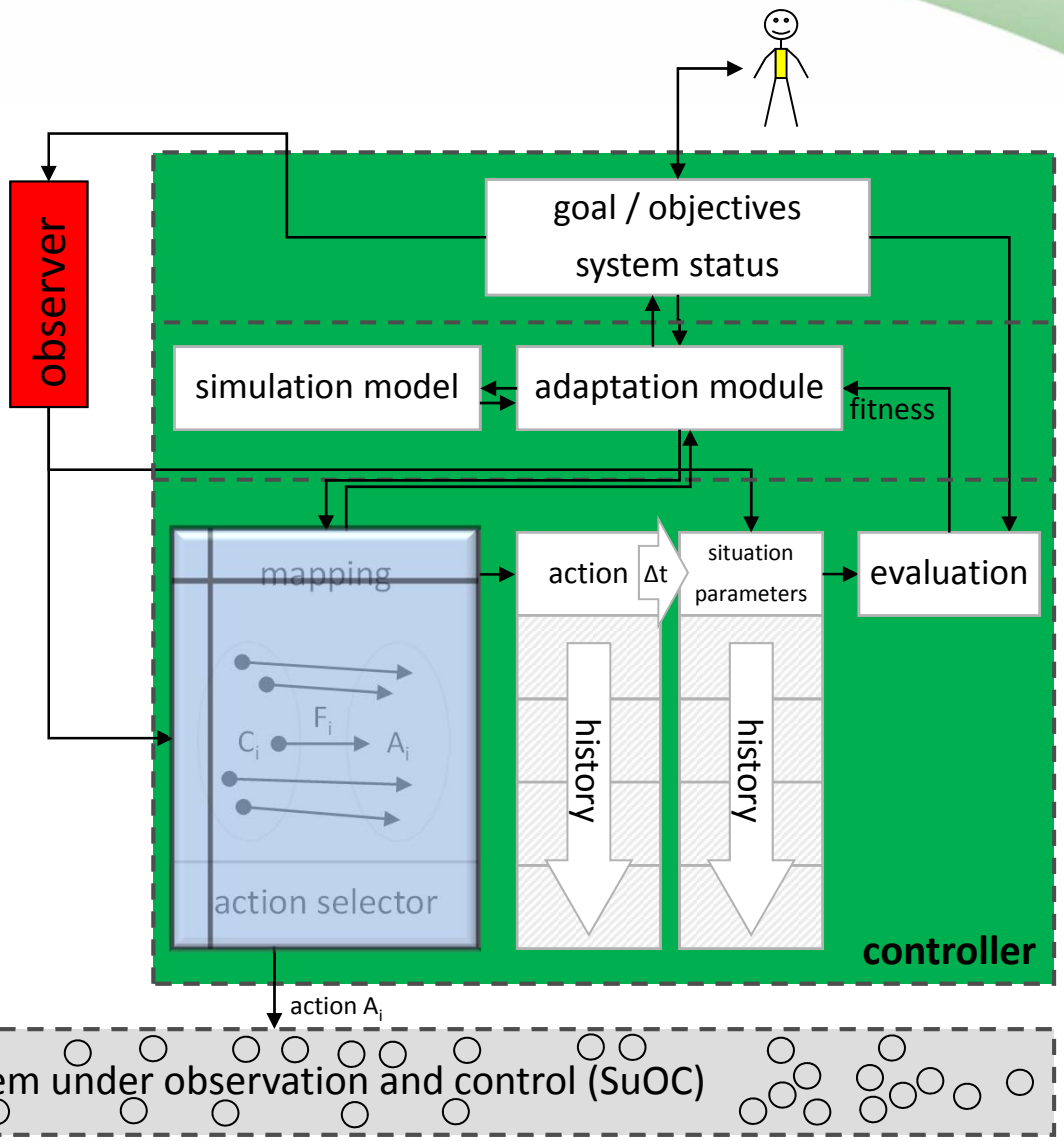
- Neither the centralised nor on the distributed architecture leads to optimal collaboration.
- Adaptive architecture that switches between the two architectures depending on the conflict level.



© C. Müller-Schloer, H. Schmeck, J. Branke, J. Hähner, M. Mnif, U. Richter, E. Cakar, N. Fredivianus 2005-2010

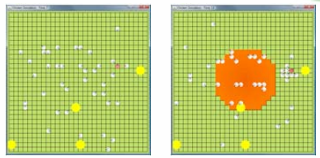
Learning

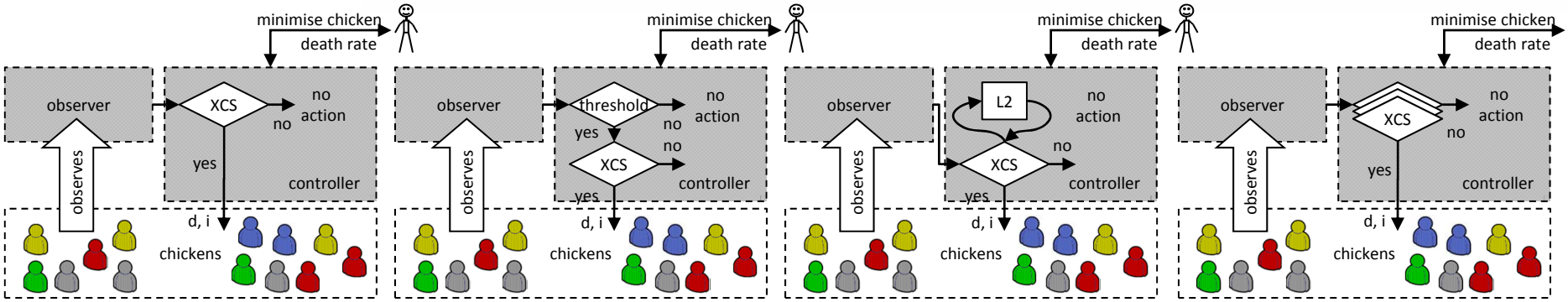
- Focus on learning as part of the generic controller
- Two-level learning approach (on-line vs. off-line)
- Learning architecture as action selector in form of a rule base



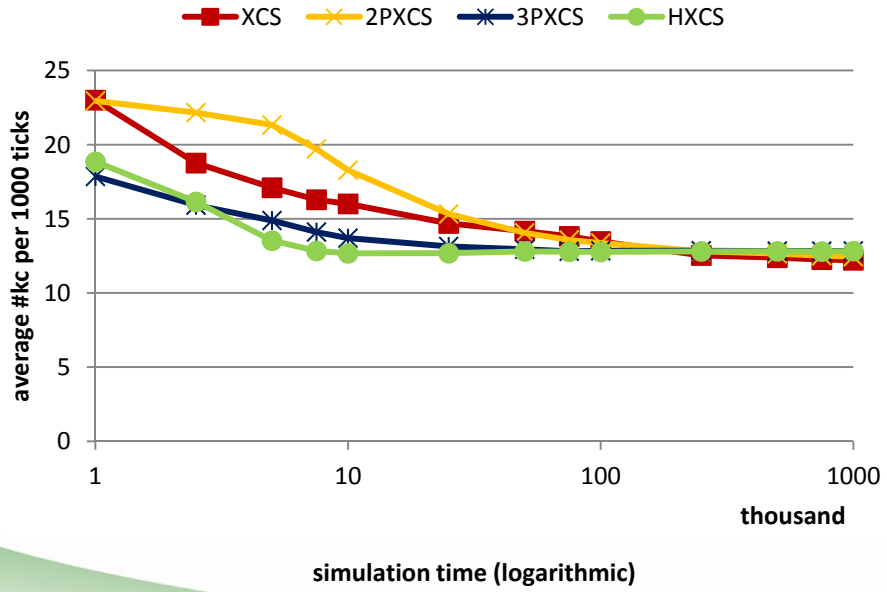
© C. Müller-Schloer, H. Schmeck, J. Branke, J. Hähner, M. Mnif, U. Richter, E. Cakar, N. Fredivianus 2005-2010

Single-Agent Learning





All situations | Only critical situations | On-line learning and off-line planning | Parallelism

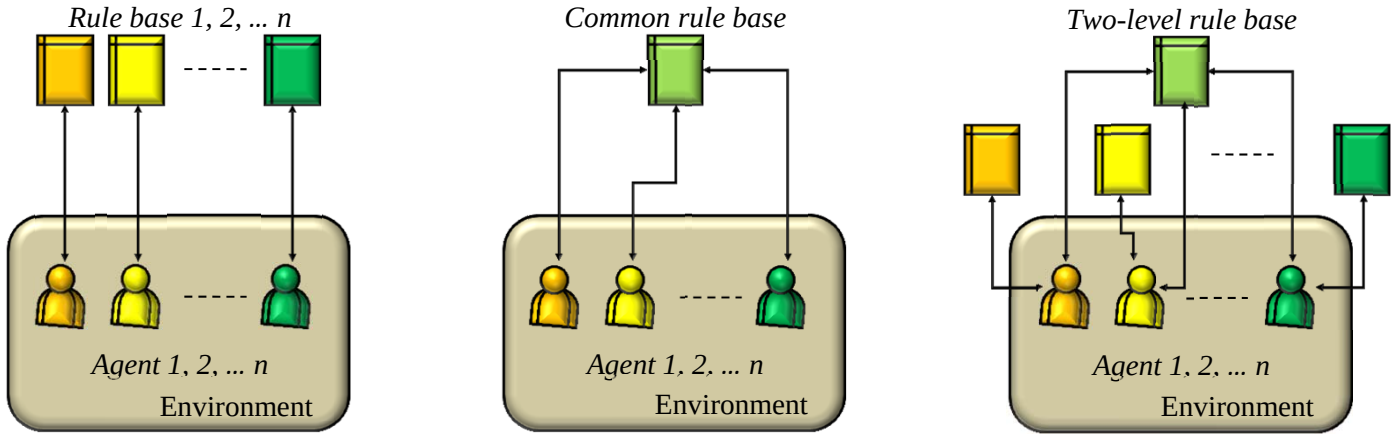


+ Significant improvements in performance and learning speed
– Parallelism is limited by the size of the problem.
– The task of decomposing a task into subtasks is static and predefined.

© C. Müller-Schloer, H. Schmeck, J. Branke, J. Hähner, M. Mnif, U. Richter, E. Cakar, N. Fredivianus 2005-2010

Multi-Agent Learning

- Three different architectures
  - ***Individual rule base approach:*** Every agent stores its learned knowledge in an individual rule base.
  - ***Common rule-base approach:*** Agents store their experience in a centralized rule base and get advantages from the shared knowledge.
  - ***Two-level rule-base approach:*** A combination of ***common*** and ***individual rule base*** which both are available to be used.

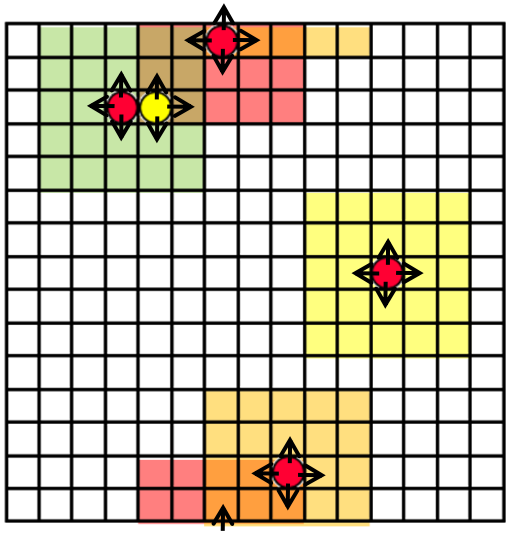


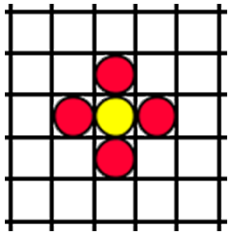
© C. Müller-Schloer, H. Schmeck, J. Branke, J. Hähner, M. Mnif, U. Richter, E. Cakar, N. Fredivianus 2005-2010

Testbed: Predator/Prey Pursuit Scenario

(Benda et. al., 1986)

- Scenario
  - Two-dimensional borderless (torus) grid-world
  - Four predators (red dots) and a prey (yellow dot)
  - The predators' team goal: Capture the prey
- Predators
  - Observe local environment
  - Communicate each other
  - Make decisions autonomously
  - Move themselves closer to the prey
- Simulations
  - Limited to 1 million ticks
  - A new prey appears at random location of the grid, after the previous one is captured and eliminated
  - The average number of capture times denotes the system performance.
- Simulations are tested using two different prey settings: *static* (not moving) and *moving*.



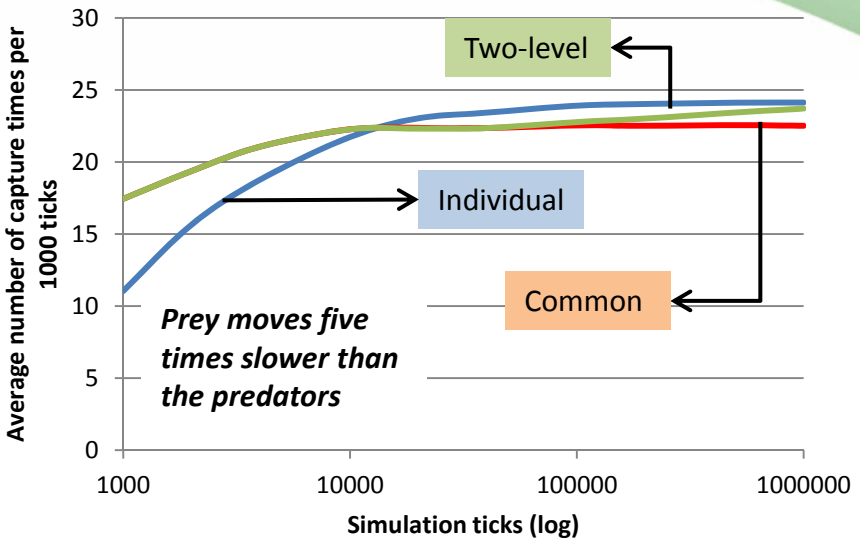


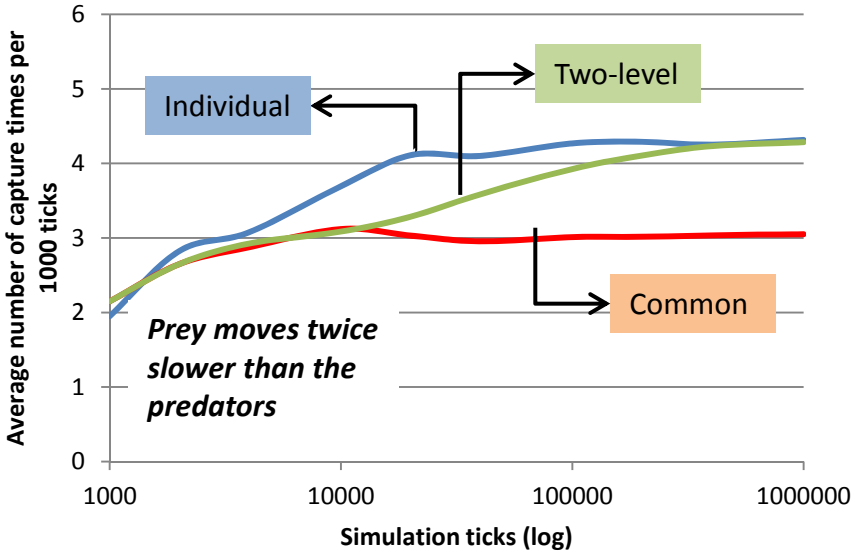
© C. Müller-Schloer, H. Schmeck, J. Branke, J. Hähner, M. Mnif, U. Richter, E. Cakar, N. Fredivianus 2005-2010

Simulation Results

+ The ***common rule base*** learning architecture succeeds in increasing the learning speed at the beginning of the simulations, although it is inferior to the ***individual rule base*** afterwards.

+ The ***two-level rule base*** combines both advantages of quicker learning as shown by the ***common rule base*** and better results in the long run as ***individual rule base*** gives.

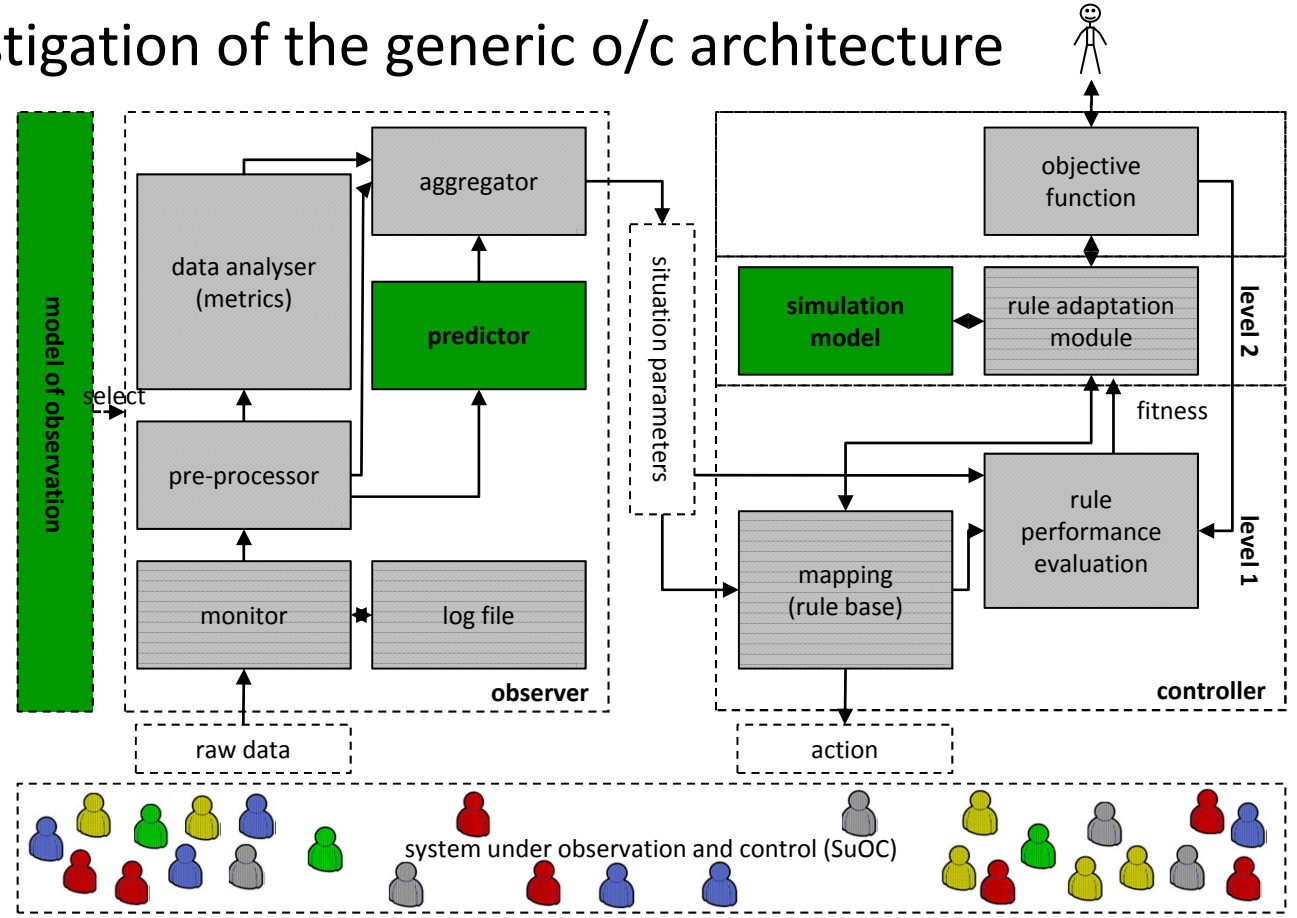




© C. Müller-Schloer, H. Schmeck, J. Branke, J. Hähner, M. Mnif, U. Richter, E. Cakar, N. Fredivianus 2005-2010

Outlook on the O/C Architecture

- Extended investigation of the generic o/c architecture
- Quantitative definitions of robustness and flexibility
- Complex and adaptive cooperation and collaboration mechanisms

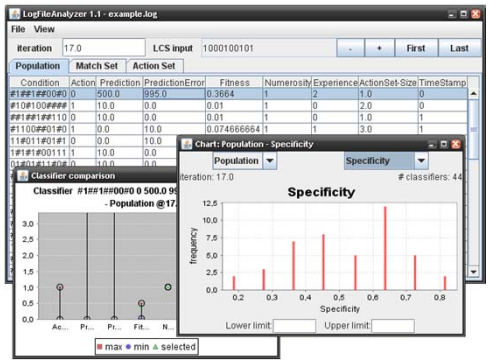


- Demonstration of the achievements (OTC^3)

© C. Müller-Schloer, H. Schmeck, J. Branke, J. Hähner, M. Mnif, U. Richter, E. Cakar, N. Fredivianus 2005-2010

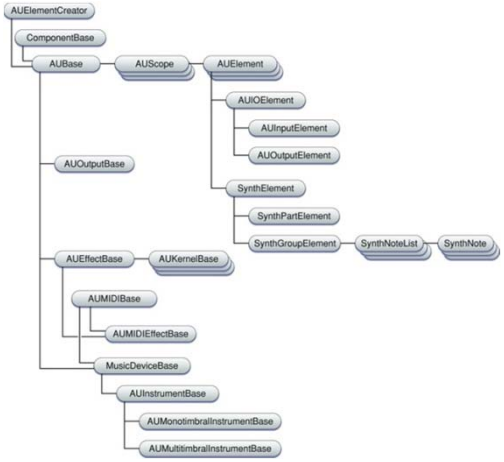
Tools

LogFileAnalyzer



- Supports the analysis of Learning Classifier System (LCS) experiments
- Visualisation of classifier populations
- Written in JAVA
- Available at http://www.aifb.uni-karlsruhe.de/EffAlg/Projekt/otcqe/publ/software.htm

Reference implementation of the abstract OTC architecture



- Written in JAVA
- Defining interfaces and (abstract) classes
- Prototypical implementation

© C. Müller-Schloer, H. Schmeck, J. Branke, J. Hähner, M. Mnif, U. Richter, E. Cakar, N. Fredivianus 2005-2010

10th Colloquium SPP OC
Observation and Control of Collaborative Systems (OCCS)

Learning Architectures for Collaborative Systems

Jürgen Branke, Nugroho Fredivianus, Urban Richter, Hartmut Schmeck
Karlsruhe Institute of Technology

Emre Cakar, Jörg Hähner, Christian Müller-Schloer
Leibniz Universität Hannover

© C. Müller-Schloer, H. Schmeck, J. Branke, J. Hähner, M. Mnif, U. Richter, E. Cakar, N. Fredivianus 2005-2010