What can Organic Computing learn from Multi Agent Systems?

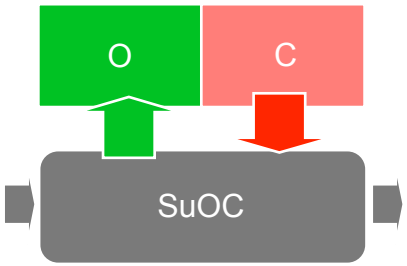
Prof. Dr.-Ing. C. Müller-Schloer

Leibniz Universität Hannover
Institut für Systems Engineering – System- und Rechnerarchitektur
Appelstraße 4
30159 Hannover
cms@sra.uni-hannover.de
+49 (0)511 762 19730

www.sra.uni-hannover.de

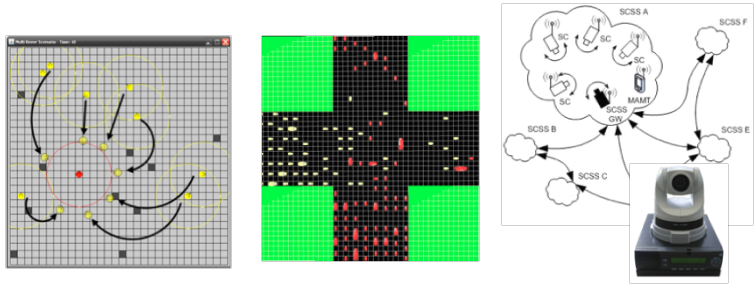
- Build technical systems with the capability to self-adapt and self-organize.
- Two-faced character of OC
  - internal (self)
  - external

Self-observation
Self-control (self-configuration)



Tasks of O/C:
- State observation
- Aggregation
- Action selection
- Real-time learning

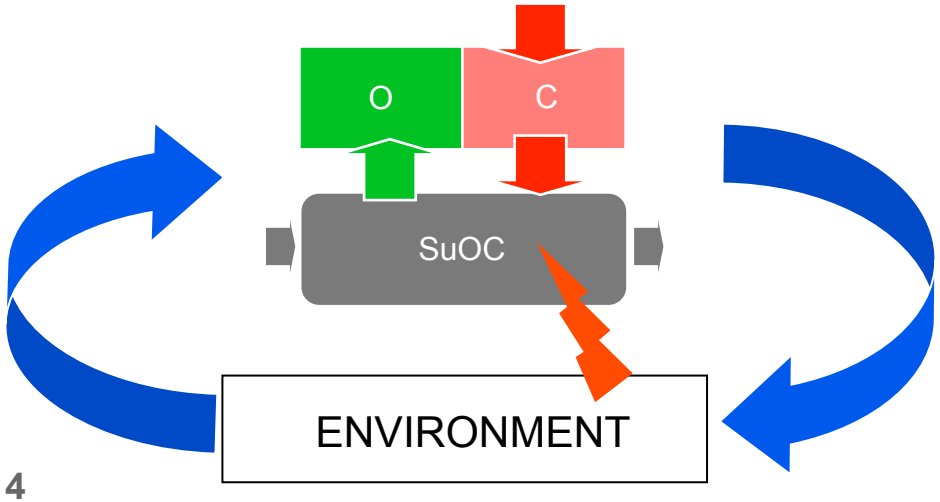
Collections
Societies

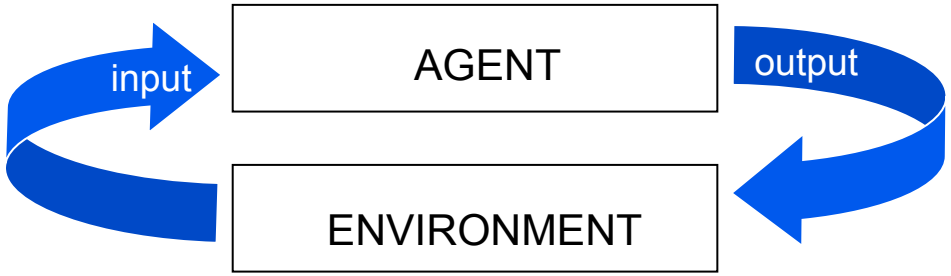


Problems:
- Interaction
- Communication
- Negotiation
- Collective learning
- ...

- Definition “agent” (Wooldridge)
  - An agent is a computer system that is capable of *independent* action on behalf of its user or owner.
- Definition “multiagent system”
  - A multiagent system is one that consists of a number of agents, which *interact* with one-another.
  - In the most general case, agents will be acting on behalf of users with *different* goals and motivations.
  - To successfully interact, they will require the ability to *cooperate*, *coordinate*, and *negotiate* with each other, much as people do

OC system	Agents
embodied: software + hardware	mainly software
hard real-time requirements	soft real-time constraints
limited resources	“unlimited” resources
reactive (so far)	reasoning and planning
OC systems are autonomous but always subject to higher-level directives.	An agent is a computer system capable of autonomous action situated in some environment in order to meet its design objectives (BDI).



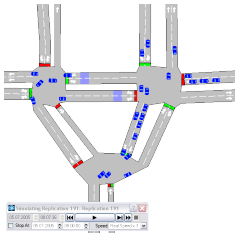


1. Coordination and cooperation
2. Norms and institutions
3. Agent architectures
4. Methodologies and tools

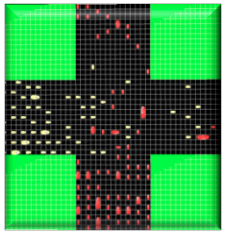
1. Coordination and cooperation
2. Norms and institutions
3. Agent architectures
4. Methodologies and tools

- Agents are self-motivated. → Cooperation or conflict?
- Examples

Cooperation: Traffic light controllers

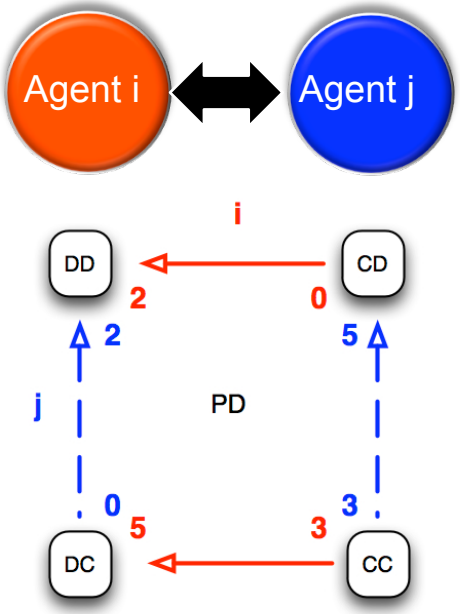


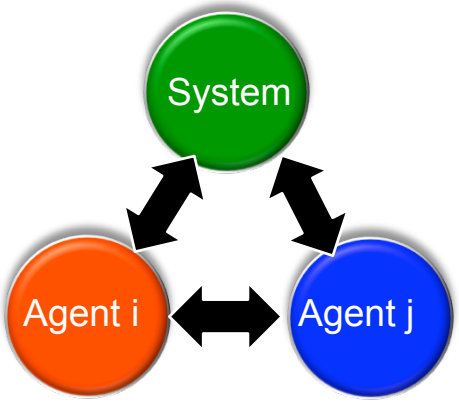
Resource conflict: Self-organized intersection

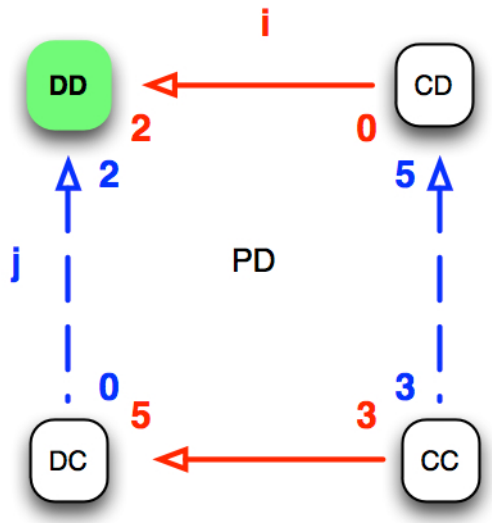


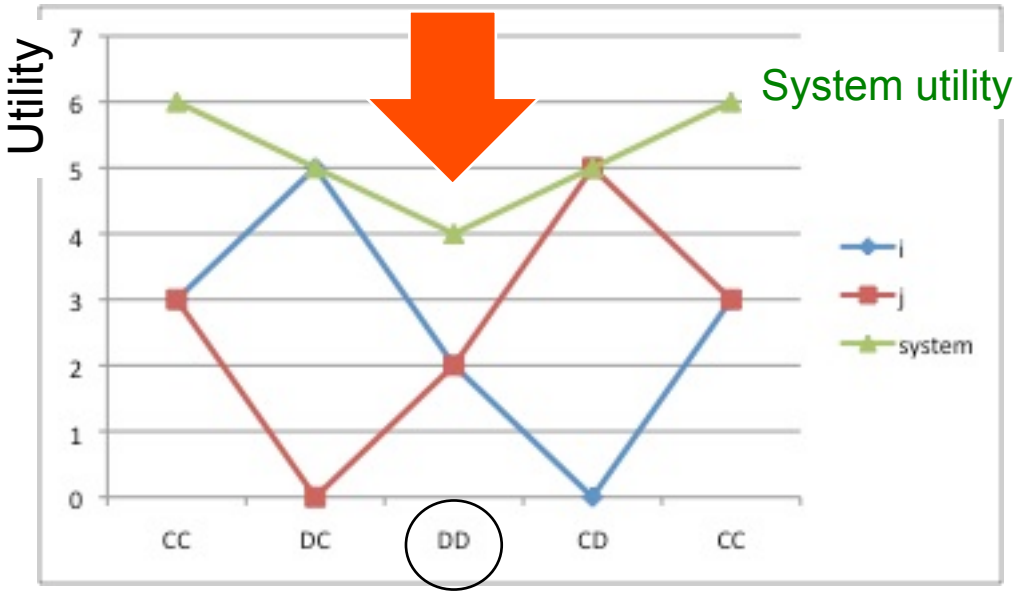
Conflict and cooperation:
Area coverage of smart cameras

- A game is characterized by its payoff matrix.
- Utility: Payoff for an individual agent depending on the outcome of the game.
- Rational agent acts such that his payoff is maximized.
- System utility: Cumulative payoff



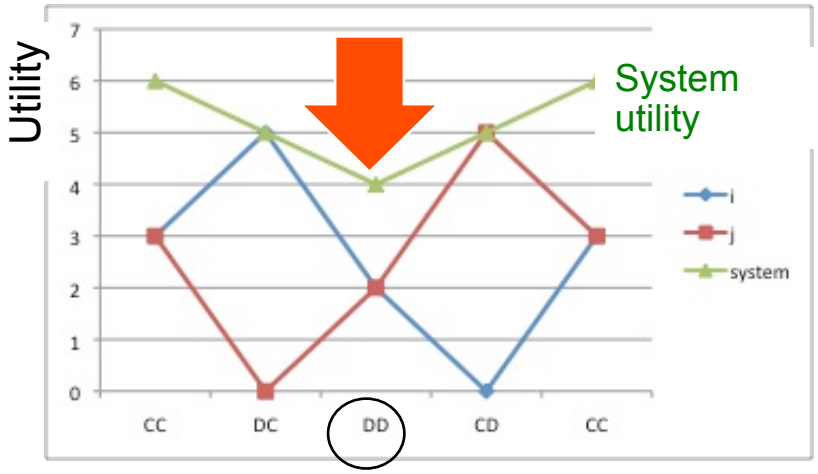






- Rational Agents
  - Agent i: **D**C ≥ CC ≥ **D**D ≥ CD
  - Agent j: C**D** ≥ CC ≥ D**D** ≥ DC
- Nash equilibrium at DD: Neither agent has an incentive to deviate from a Nash equilibrium.

- The rational choice can lead to a sub-optimal system utility.

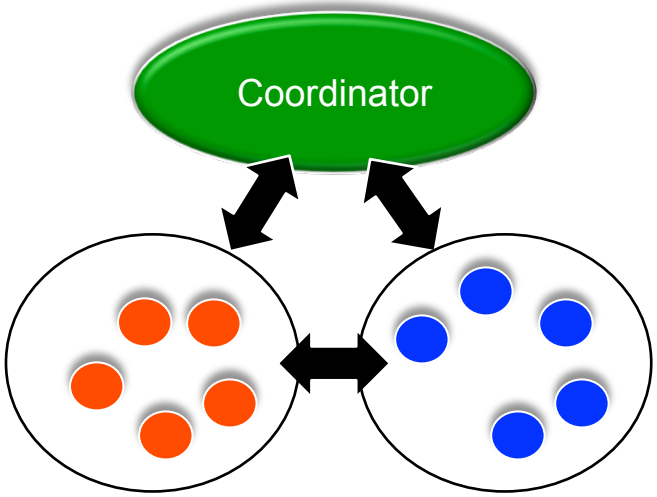


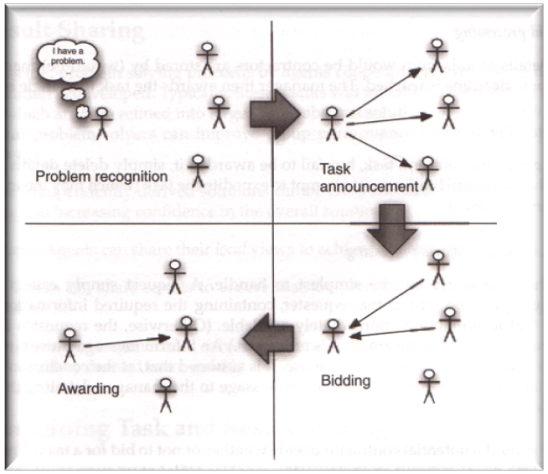
- How can we optimize the system utility and the individual utility?

1. Extend the reasoning process to include the system utility (avoid the local optimum!).
   - The "Rational Agent" is not really rational.
2. Enable the agents to negotiate state transition sequences.
3. This requires binding commitments → trust!

→ Cooperative Game Theory: Forming coalitions

- Forming coalitions (cooperative games)
  - e.g. Peleg and Sudholter, 2002
  - Cooperation lifecycle
  - Representation of games
- Negotiation: Contract Nets
  - Smith 1977, 1980
  - Auction mechanism (problem recognition, task announcement, bidding, awarding)
- Bargaining
- Arguing



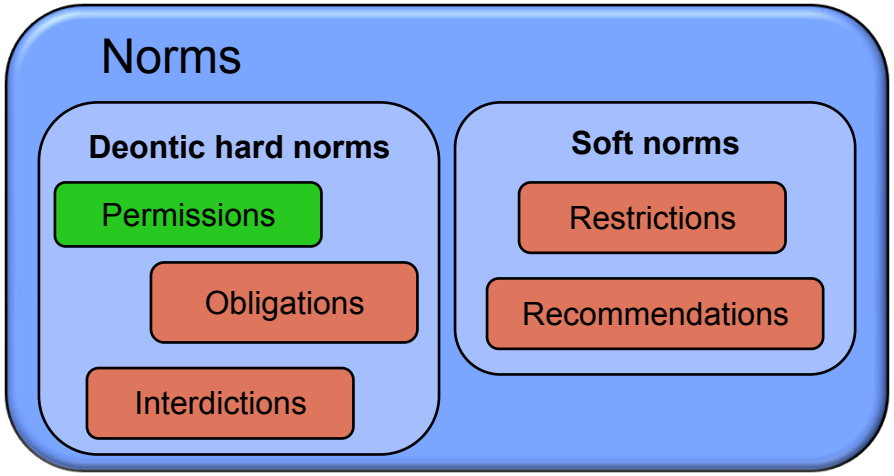


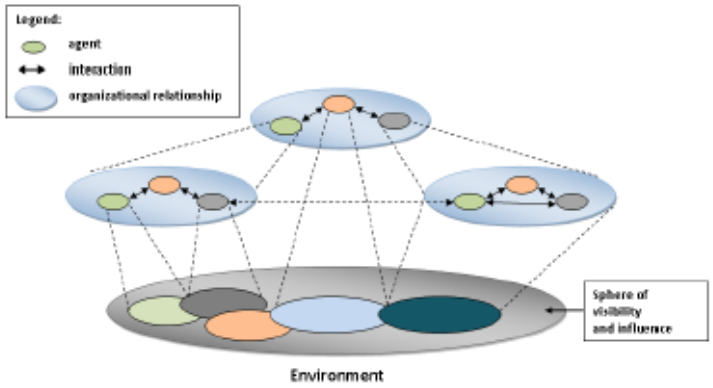
Contract Net Protocol

➡ Make Game Theory operational (at run time)!

1. Coordination and cooperation
2. Norms and institutions
3. Agent architectures
4. Methodologies and tools

- Complex (agent) societies need rules (or norms).
- Norms can be
  - Permissions, obligations, constraints, conventions, commitments…
  - Hard/soft, global/domain-specific
  - hierarchical
- Norm implementation
  - Formalization (logic, fitness functions…)
- Norms require organizational structures: Institutions and organizations

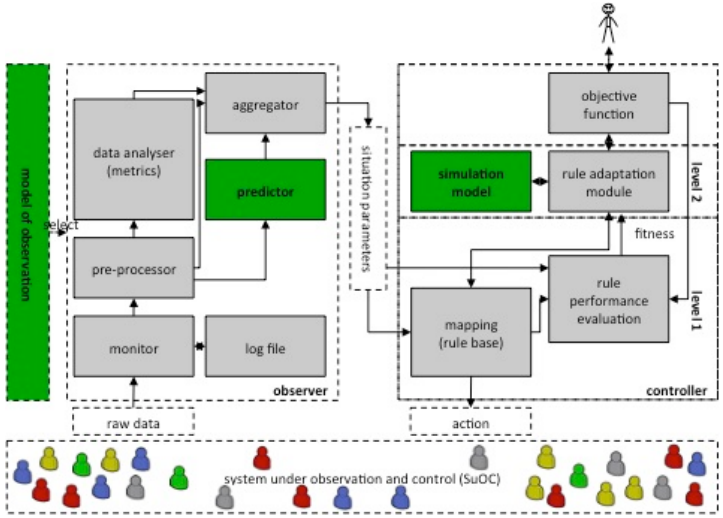




- Norm utilization
  - Agent obedience
  - Enforcement (sanctions)
  - Learning: Occasional violation might be beneficial
- Norm generation
  - Deontic norms: Top-down
  - Emergent norms: Bottom-up generation/modification
  - Conflicting norms?

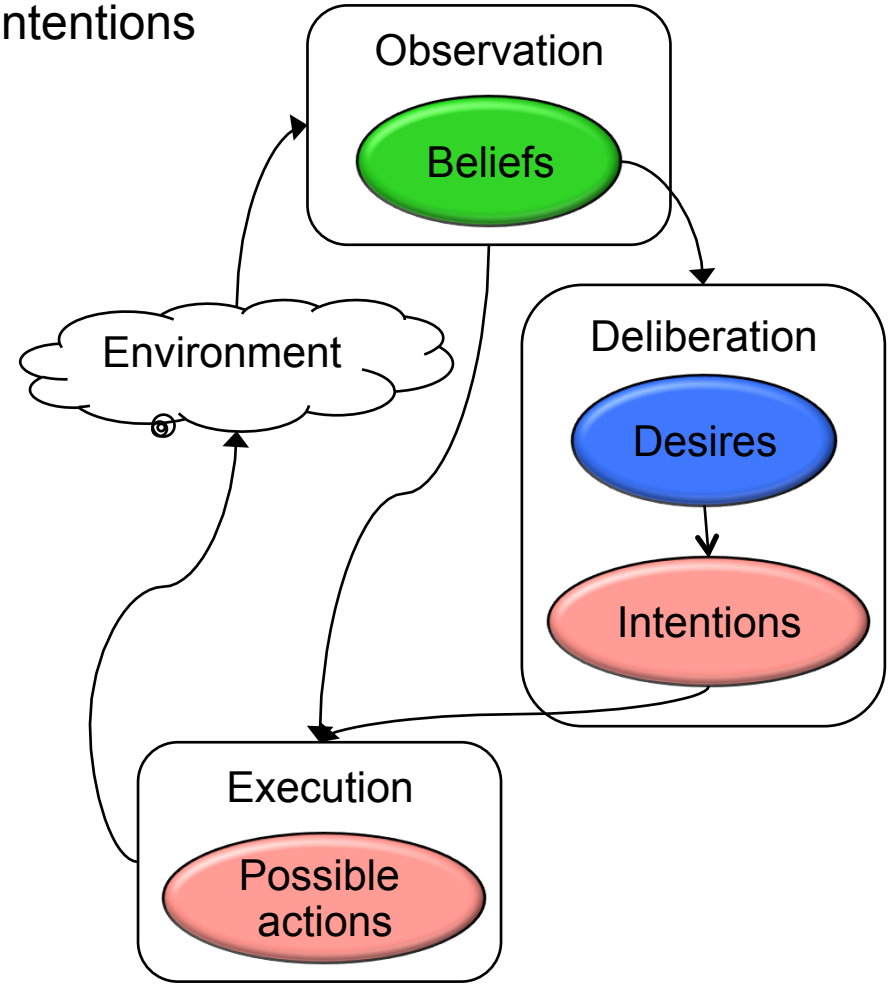
1. Coordination and cooperation
2. Norms and institutions
3. Agent architectures
4. Methodologies and tools

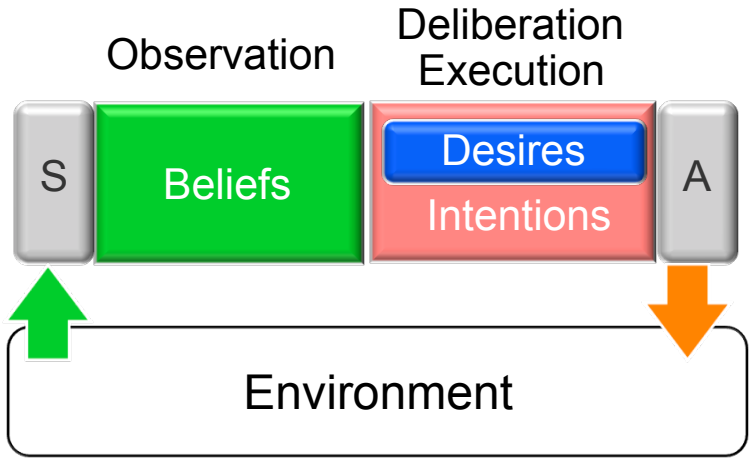
- OC has developed the Observer/Controller (O/C) architecture
  - So far implemented:
    - Stimulus-response system
    - Fast online-learning, extensive off-line optimization
    - Rudimentary history mechanism

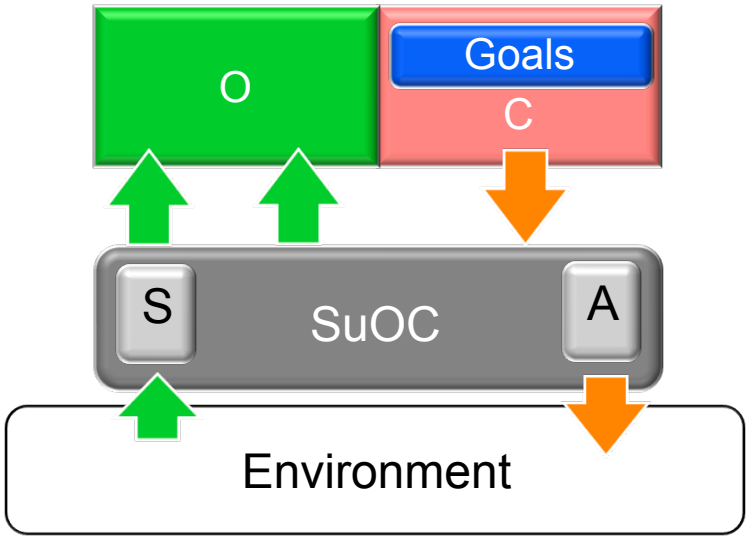


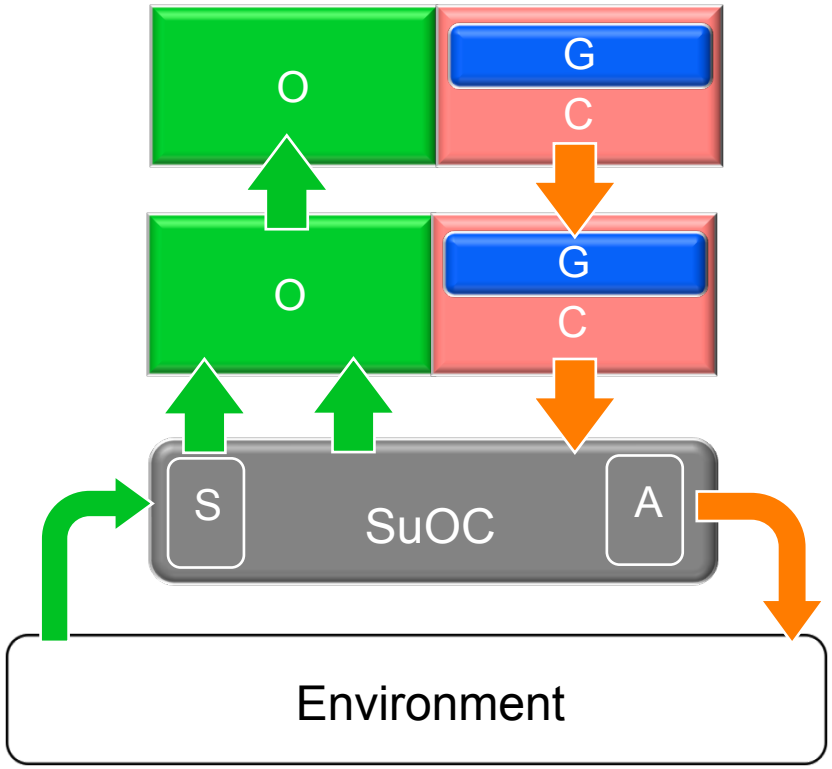
- MAS
  - Deductive reasoning: not practical
  - Subsumption architecture (R. Brooks)
  - Belief – Desire – Intention architecture (BDI - psychological model)

- Beliefs
- Desires
- Intentions





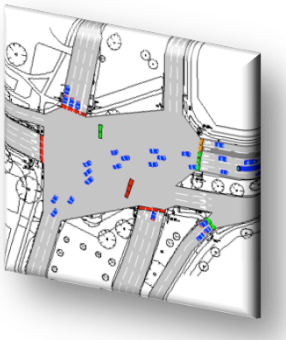


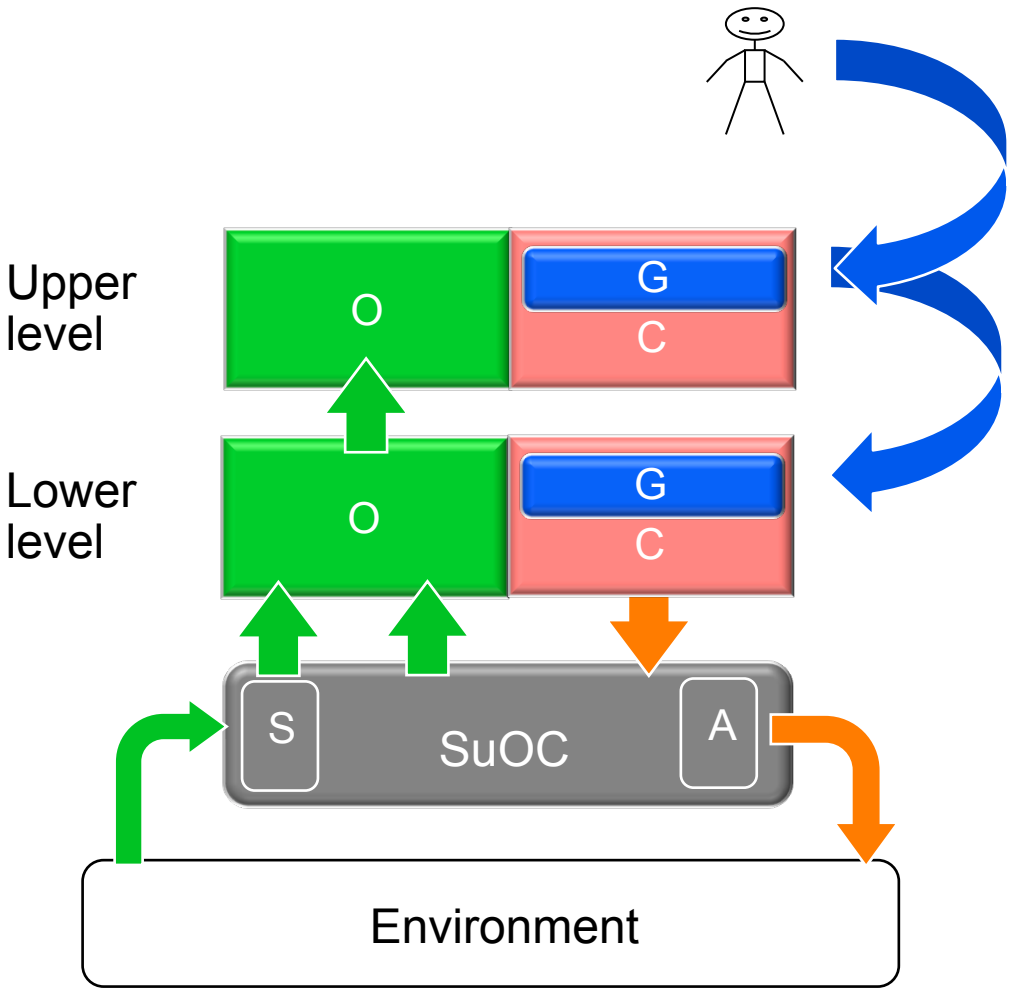


Level 2 Planning (e.g. GA)

Level 1 Action selection (e.g. LCS)

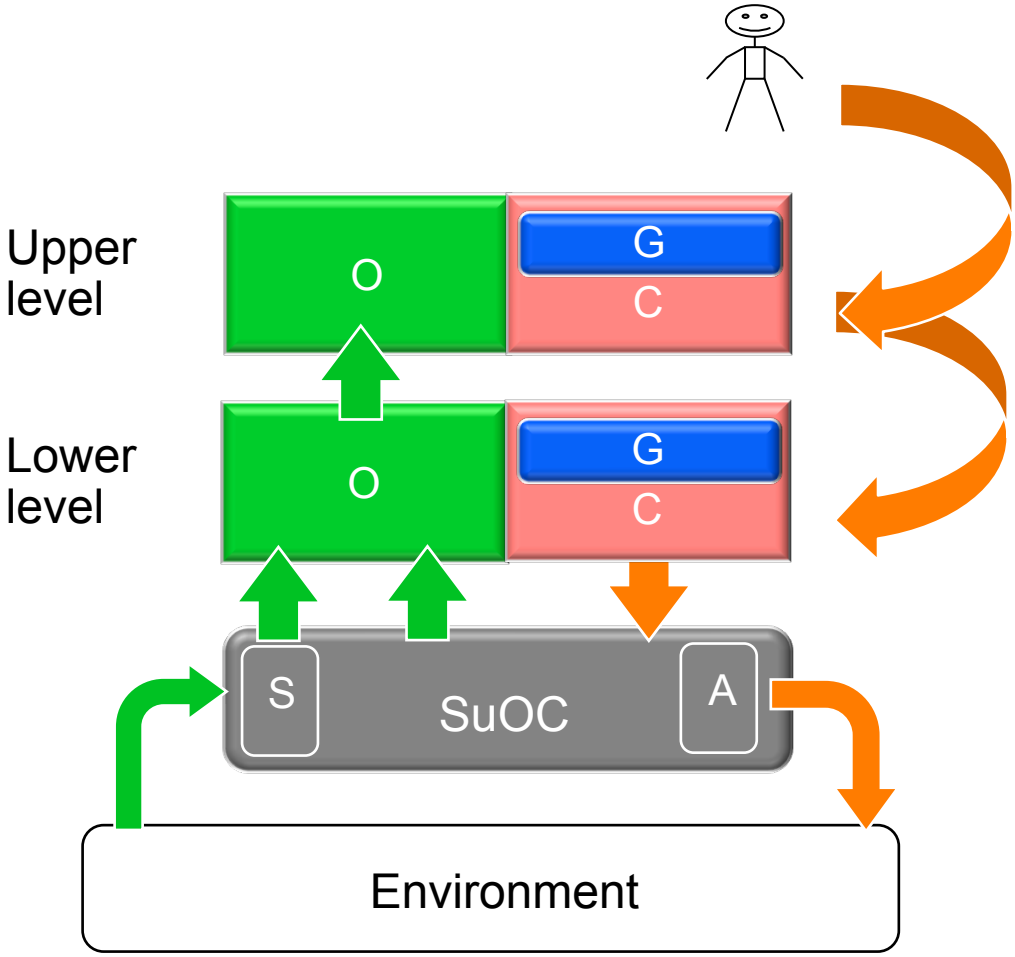
Level 0 Execution (physical)





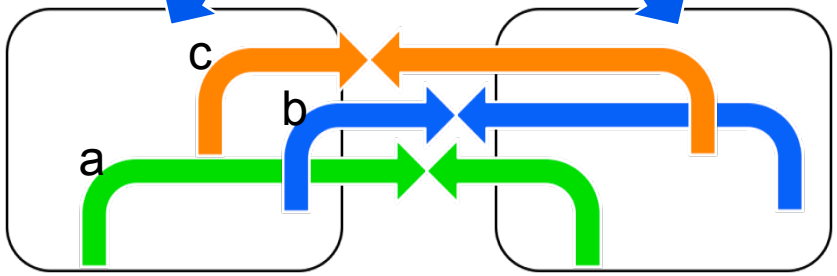
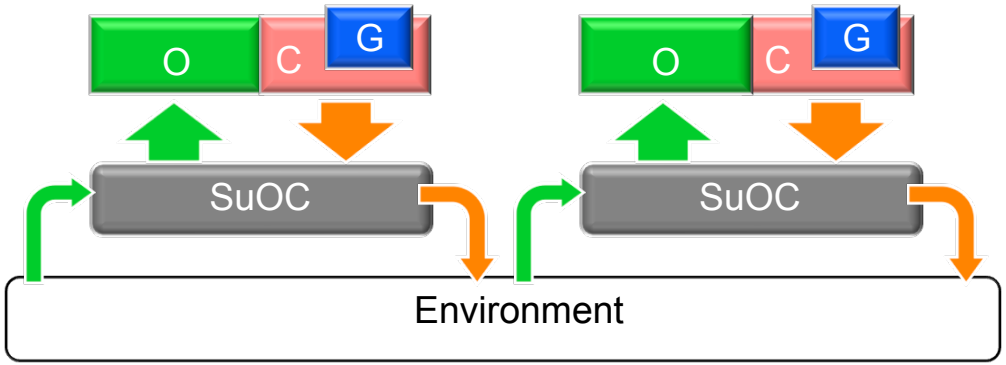
Goal flow

- Top level: designer or user
- Simple case: Top-down
- Different goal representations
- Different abstraction levels

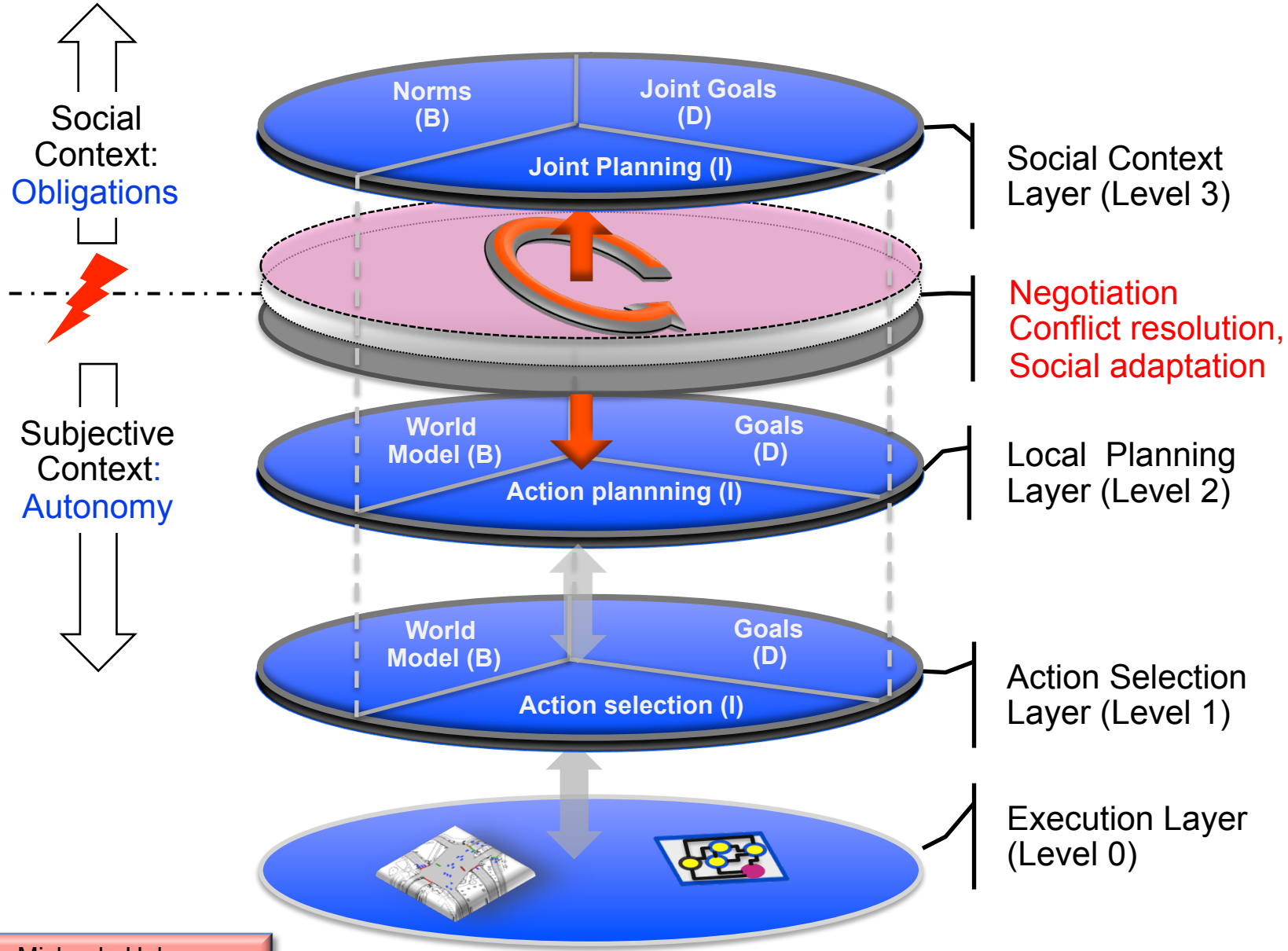


Action/plan flow

- Top level: designer
- Simple case: Top-down
- Decreasing degrees of freedom
- Different action representations

Extended Interaction Architecture	Level	Interaction
	4 Institutional/ normative level	- Soft/hard constraints - Norm enforcement - Norm adaptation (legislation) - Coordination (conflict resolution)
	3 Social level (P2P)	a) Observation exchange b) Goal reconciliation c) Cooperative action planning
	0 – 2 Execution levels	Indirect through environment

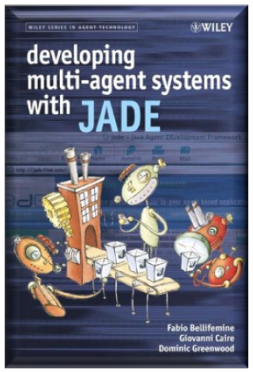
Architectures (6/6): Negotiation and conflict resolution



Source: Michaela Huhn, Jörg Müller et al.: AIM Project

1. Coordination and cooperation
2. Norms and institutions
3. Agent architectures
4. Methodologies and tools

- Agent communication
  - Speech act theory (formalized dialogues)
  - Agent communication language (FIPA ACL[1])
- Tools
  - Frameworks (JADE[2], Jadex[3], …)
  - Simulators (RePast…)
- MAS system development
  - Agent development methodology[4]



FIPA Performatives:

- `accept-proposal`
- `agree`
- `cancel`
- `confirm`
- `disconfirm`
- `inform`
- `request`
- `...`

[1] FIPA: Foundation for Intelligent Physical Agents

[2] Developing Multi-Agent Systems with JADE: Fabio Luigi Bellifemine, Giovanni Caire, Dominic Greenwood, Wiley 2007

[3] http://jadex.informatik.uni-hamburg.de/xwiki/bin/view/About/Overview

[4] Michael Winikoff; Developing Intelligent Agent Systems (Wiley 2004)

- Organic Computing is developing towards collections (or societies) of embodied agents.
- OC is more than just MAS but …
- The MAS community has broad experience or is currently active in a variety of research fields relevant also for OC.
  - Coordination and cooperation
  - Norms and institutions
  - Agent architectures
  - Methodologies and tools
- OC should use this experience!

Thank you for your attention!

NATURAM OBSERVANTES
VIVERE DISCIMUS