Multiagent Systems

5. Reactive and Hybrid Agent Architectures

B. Nebel, C. Becker-Asano, S. Wölfl

Albert-Ludwigs-Universität Freiburg

May 21, 2014

Multiagent Systems

May 21, 2014 — 5. Reactive and Hybrid Agent Architectures

5.1 Background

5.2 Reactive Architectures

5.3 Hybrid Architectures

5.4 Summary

5.1 Background

Symbolic AI: A critical view I

Recall the "Symbol systems hypothesis":

- Is **inference on symbols representing the world** sufficient to solve real-world problems . . .
- . . . or are these **symbolic representations irrelevant** as long as the agent is successful in the physical world?
- "Elephants don't play chess" (Brooks, 1990)

$\Rightarrow$ "[..] it is unfair to claim that an elephant has no intelligence worth studying just because it does not play chess." (Brooks, 1990)

Chess has been called the "Drosophila of AI" (Kronrod, 1965), because for the symbolic AI proponents it served as a standard problem exemplifying the power of human intelligence.

Symbolic AI: A critical view II

Problems with "symbolic AI" (also called "Traditional AI" or Good Old-Fashioned AI (**GOFAI**)):

- **Computational complexity** of reasoning in real-world applications
- The **transduction/knowledge acquisition** bottleneck
- Logic-based approaches largely focus on **theoretical reasoning**
- In itself, **detached from interaction** with physical world

"**Nouvelle AI** tries to demonstrate less sophisticated tasks operating **robustly in noisy complex domains**. The **hope** is that the ideas used will **generalize to more sophisticated tasks**." (Brooks, 1990)
⇒ Central idea: **emergence** of more global behavior from the interaction of smaller behavioral units

Types of Agent Architectures

From this dispute a distinction between **reactive** (behavioural / situated) and **deliberative** agents evolved.

Distinction arises naturally from tension between:

- reactivity and
- proactiveness

as key aspects of intelligent behaviour.

Three broad categories:

- **Deliberative architectures**: focus on **planning** and symbolic reasoning
- **Reactive architectures**: focus on **reactivity** based on behavioral rules
- **Hybrid architectures**: attempt to **balance** proactiveness and reactivity

5.2 Reactive Architectures

- Braitenberg Vehicle
- Subsumption Architecture
- Discussion

Motivation for Reactive Architectures

BDI approach:

- most popular (e.g. Jason) for modeling **rational agency**
- **criticized** for heavily relying on **symbolic AI methods**

Some of the (unsolved) problems of symbolic AI have lead to research in **reactive architectures**.

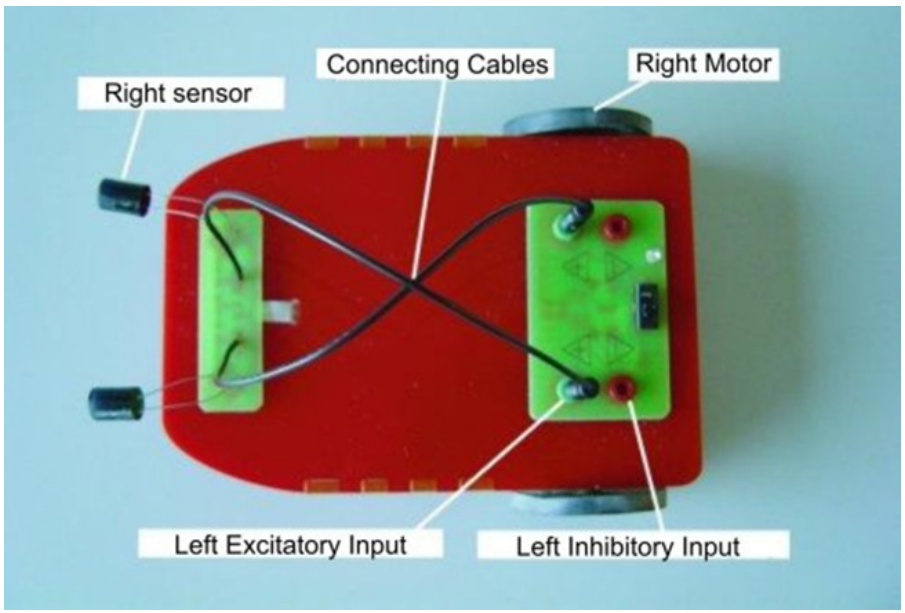
Rodney Brooks' three theses:

1. Intelligent behavior can be generated **without explicit representations** of the kind that symbolic AI proposes
2. Intelligent behavior can be generated **without explicit abstract reasoning** of the kind that symbolic AI proposes
3. Intelligence is an **emergent property** of certain complex systems

And finally:
$\Rightarrow$ Intelligence is 'in the eye of the beholder'.

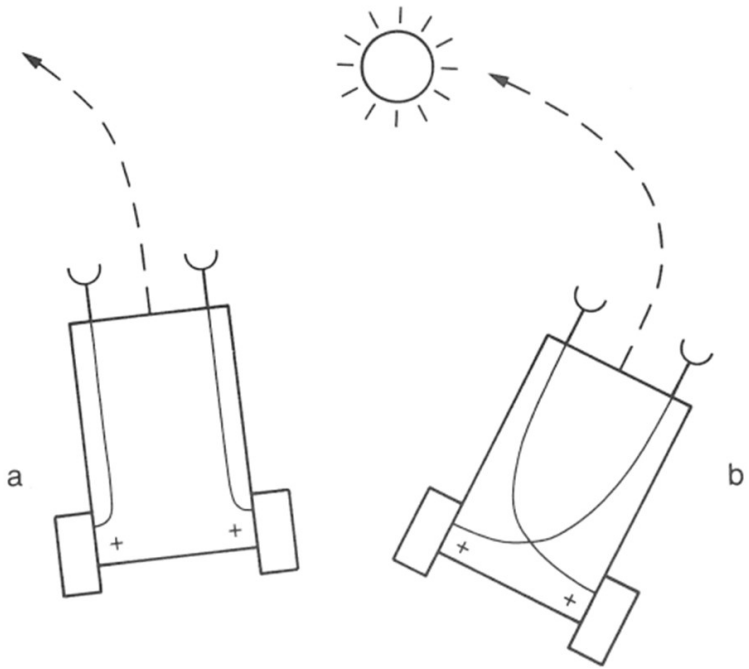
Example 1: Braitenberg Vehicle I

Valentino Braitenberg invented these vehicles as an example for how easily we ascribe "feelings" to moving artifacts (ETH Zürich, 1980s).



Example 1: Braitenberg Vehicle II

Braitenberg Vehicle either **fears** (a) or **loves** (b) the light:



Functionality: sensors **connected directly** to wheels, so that they turn when sensors are activated ⇒ absolutely **reactive**, no processing/reasoning possible!

Ideas behind "Subsumption Architecture"

Two key ideas behind Brooks' research:

1. Situatedness/embodiment: Real intelligence is situated in the world, not in disembodied systems such as theorem provers or expert systems
2. Intelligence and emergence: Intelligent behavior results from agent's interaction with its environment

Subsumption architecture illustrates these principles:

- Essentially a hierarchy of task-accomplishing **behaviors** (simple rules) competing for control over agent's behavior
- **Behaviors** (simple situation-action rules) can fire simultaneously $\Rightarrow$ need for meta-level control
- Lower layers correspond to "primitive" behaviors and have precedence over higher (more abstract) ones
- Extremely simple in computational terms $\Rightarrow$ can be implemented in hardware

Subsumption architecture

Formally: *see*() as before, *action*() = set of behaviors

- Set of all behaviors $Beh = \{(c, \alpha) \mid c \subseteq Per \text{ and } \alpha \in Ac\}$
- Behavior fires in state s iff $see(s) \in c$
- Agent's set of behaviors $R \subseteq Beh$
- **inhibition relation** $\prec \subseteq R \times R$ is a strict total ordering (transitive, reflexive, antisymmetric)
- If $b_1 \prec b_2$, b_1 will get priority over b_2

Action selection

Action selection in Subsumption Architecture:

```
1  function action(p : Per): returns an action α ∈ Ac
2      var fired : P(R); selected : A;
3      begin
4          fired ← {(c, α) | (c, α) ∈ R and p ∈ c};
5          foreach (c, α) ∈ fired do
6              if ¬(∃(c′, α′) ∈ fired with (c′, α′) ≺ (c, α) then
7                  return α;
8          end
9          return null
10     end
```

Example 2: Mars Explorer World I

Luc Steels' cooperative Mars explorer system

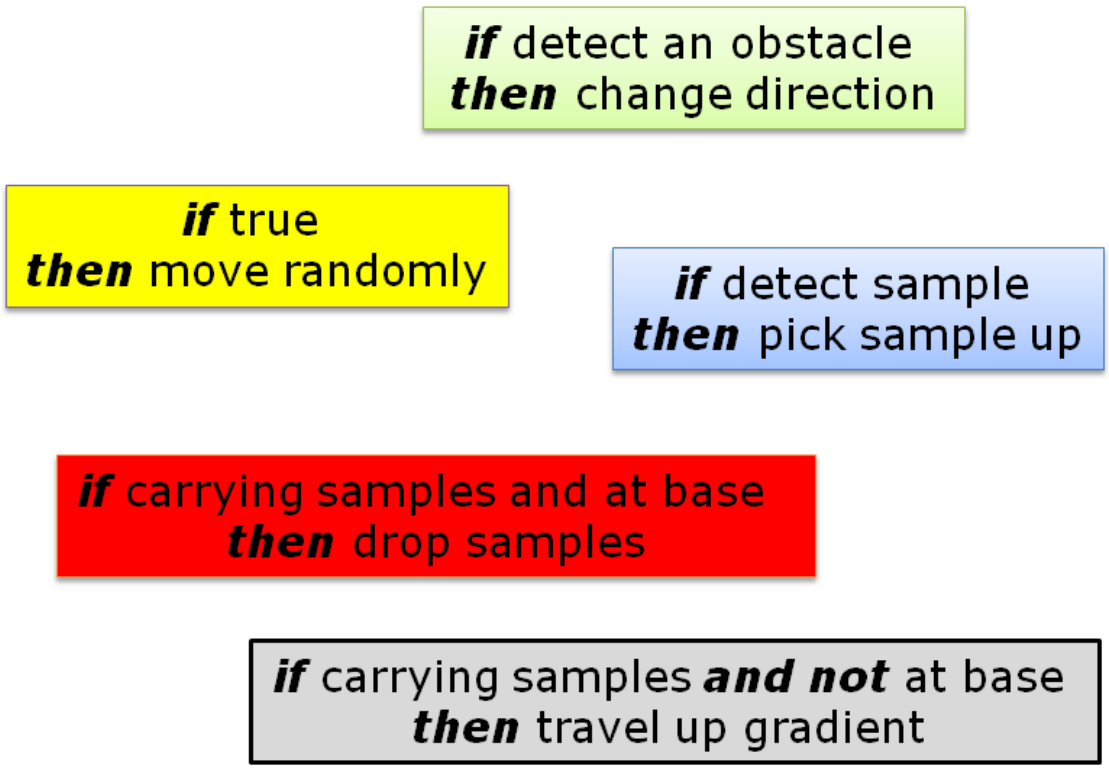
Domain:

- Robots attempt to gather rock samples on Mars
- Location of rocks unknown, but usually appear in clusters
- Obstacles block the path and prevent "direct communication"
- Global radio signal from mother ship to find way back

How can we program a single agent's behavior?

Example 2: Mars Explorer World II

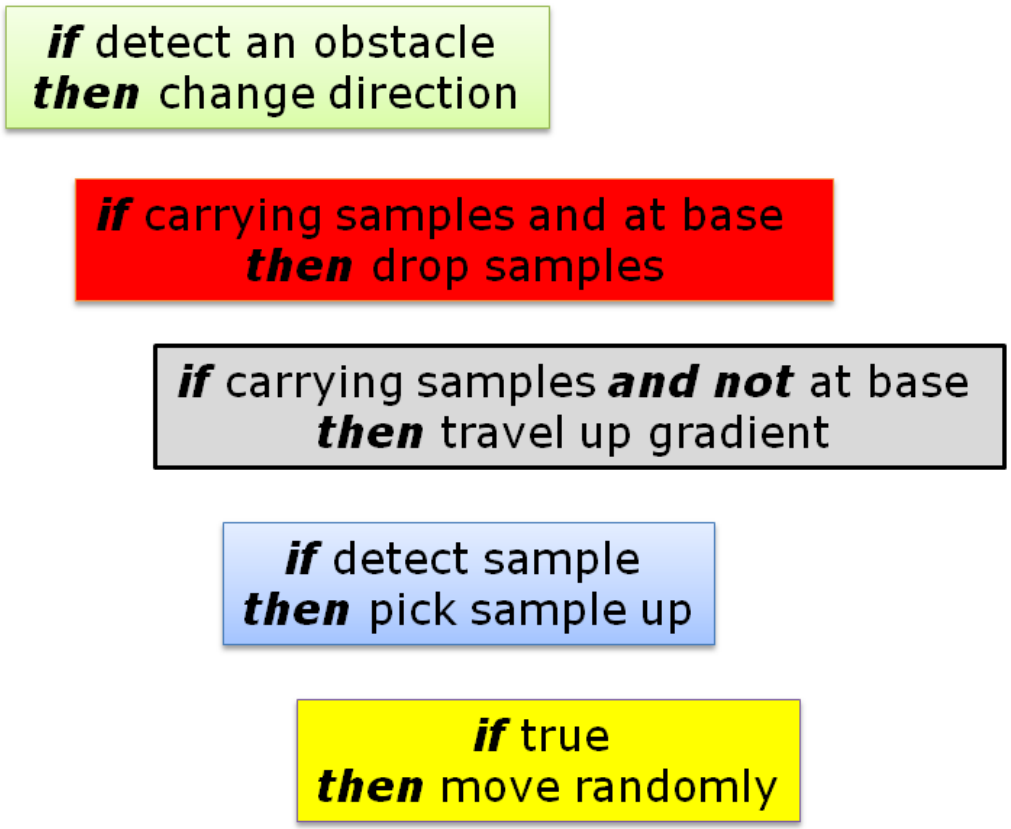
Consider the following five rules:



Define $color_1 \prec color_2 \prec color_3 \prec color_4 \prec color_5$ to assure successful roaming activity. Any ideas?

Example 2: Mars Explorer World III

It works with $green \prec red \prec grey \prec blue \prec yellow$:



. . . but how about clusters and team behavior?

Example 2: Mars Explorer World IV

Integrating indirect communication:

- When sample found, better tell other agents about it
- Direct communication not possible

Solution inspired from ants' foraging behavior:

- Agent creates **trail** by dropping crumbs on way back to base
- Other agents will pick crumbs up $\Rightarrow$ trail will faint away
- When not carrying sample and found a crumb/trail, then follow trail away from base
- If no more samples at end of trail, trail won't be reinforced

Example 2: Mars Explorer World V

Modified set of behaviors:

1. *If* detect obstacle *then* change direction
2. *If* carrying samples and at base *then* drop samples
3. *If* carrying samples and not at base *then* **drop 2 crumbs** and travel up gradient
4. *If* detect sample *then* pick sample up
5. *If* **sense crumbs** *then* **pick up 1 crumb** and **travel down gradient**
6. *If* true *then* move randomly

Discussion

Reactive architectures achieve tasks that would be considered very impressive using symbolic AI methods.
But they have some drawbacks:

- Agents must be able to map local knowledge to appropriate action
- Impossible to take non-local (or long-term) information into account
- If it works, how do we know why it works?
  $\Rightarrow$ departure from knowledge level means loss of transparency
- What if it doesn't work? $\Rightarrow$ difficult to debug
- Lack of clear design methodology
- Design becomes difficult with increasing number of rules
- How about communication with humans?

5.3 Hybrid Architectures

- TouringMachines
- InteRRaP

Hybrid Architectures

Idea:

- Neither completely deliberative nor completely reactive
- Combine both perspectives in one architecture

Most obvious approach: Construct agent with one (or more) reactive and one (or more) deliberative sub-component:

- Reactive sub-component(s): capable to respond to world changes quickly, without complex reasoning and decision-making
- Deliberative sub-component(s): responsible for abstract planning and decision-making using symbolic representations

Hybrid Architectures: meta-level control

Meta-level control of interactions between these components becomes a key issue in hybrid architectures.
⇒ Commonly used: **layered** approaches.

Horizontal layering:

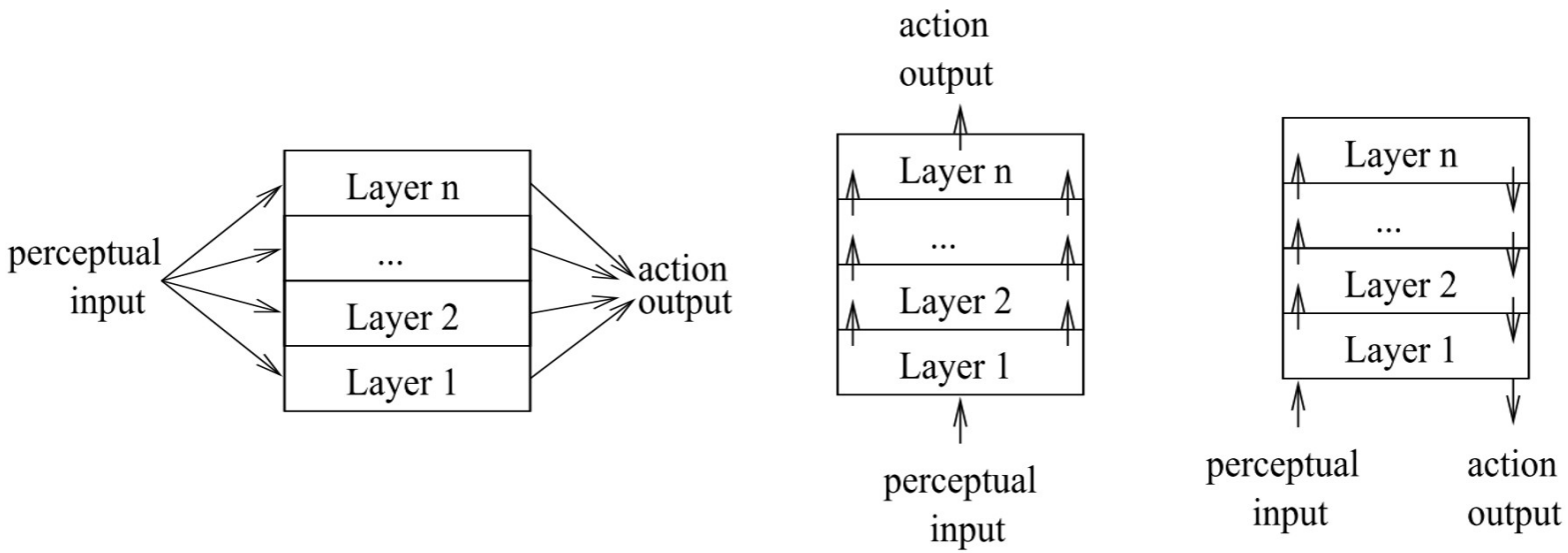
- All layers connected to sensory input/action output
- Each layer produces an action, different suggestions have to be arbitrated

Vertical layering:

- Only one layer connected to sensors/effectors
- Filtering approach (one-pass control): propagate intermediate decisions from one layer to another
- Abstraction layer approach (two-pass control): different layers make decisions at different levels of abstraction

Hybrid Architectures: layered approach

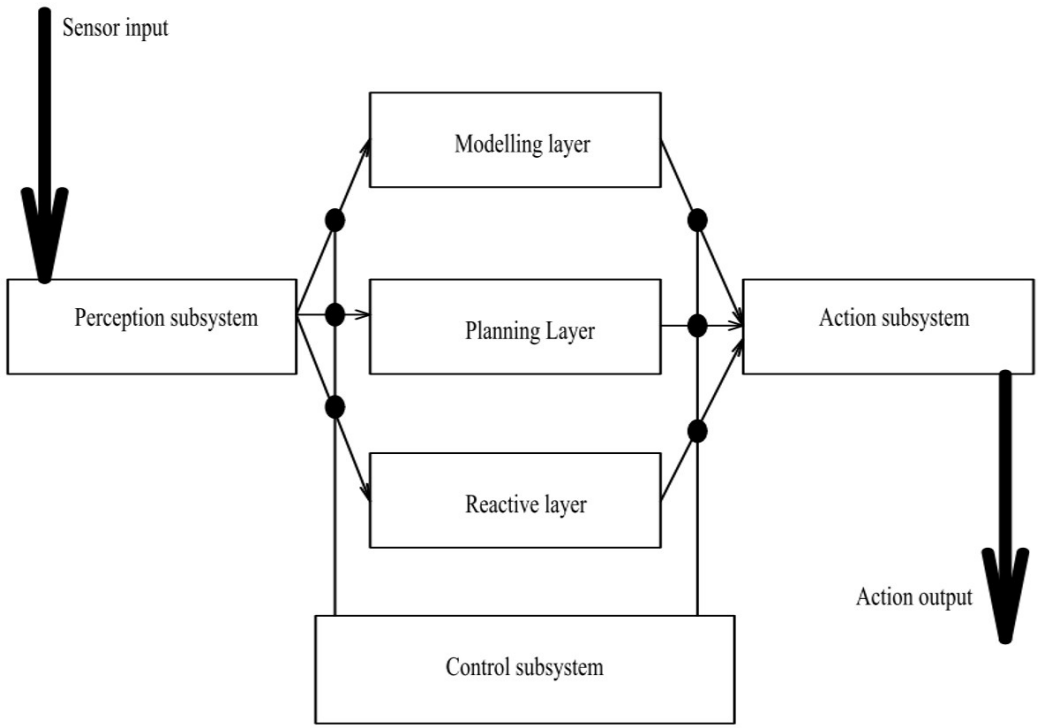
Horizontal layering (left), vertical layering one-pass (middle) and two-pass (right):



(from Wooldridge 2002, p. 98)

Horizontally layered agent architecture example

The **TouringMachines** architecture is an example for a horizontal layered architecture.



(after Wooldridge 2009, p. 95)

TouringMachines

Three sub-systems:

- Perception sub-system
- Control sub-system
- Action sub-system

Control sub-system consists of:

- Reactive layer: situation-action rules
- Planning layer: construction of plans and action selection
- Modelling layer: contains symbolic representations of mental states of other agents

The three layers communicate via explicit **control rules**.

Vertically layered agent architecture example

InteRRaP:

- Integration of rational planning and reactive behavior
- Vertical (two-pass) layering architecture
- Three layers:
  1. Behavior Layer: manages reactive behaviour of agent
  2. Local Planning Layer: individual planning capabilities
  3. Social Planning Layer: determining interaction/cooperation strategies
- Two-pass control flow:
  - Upward activation: when capabilities of lower layer are exceeded, higher layer obtains control
  - Downward commitment: higher layer uses operation primitives of lower layer to achieve objectives

InteRRap I

Every layer consists of two modules:

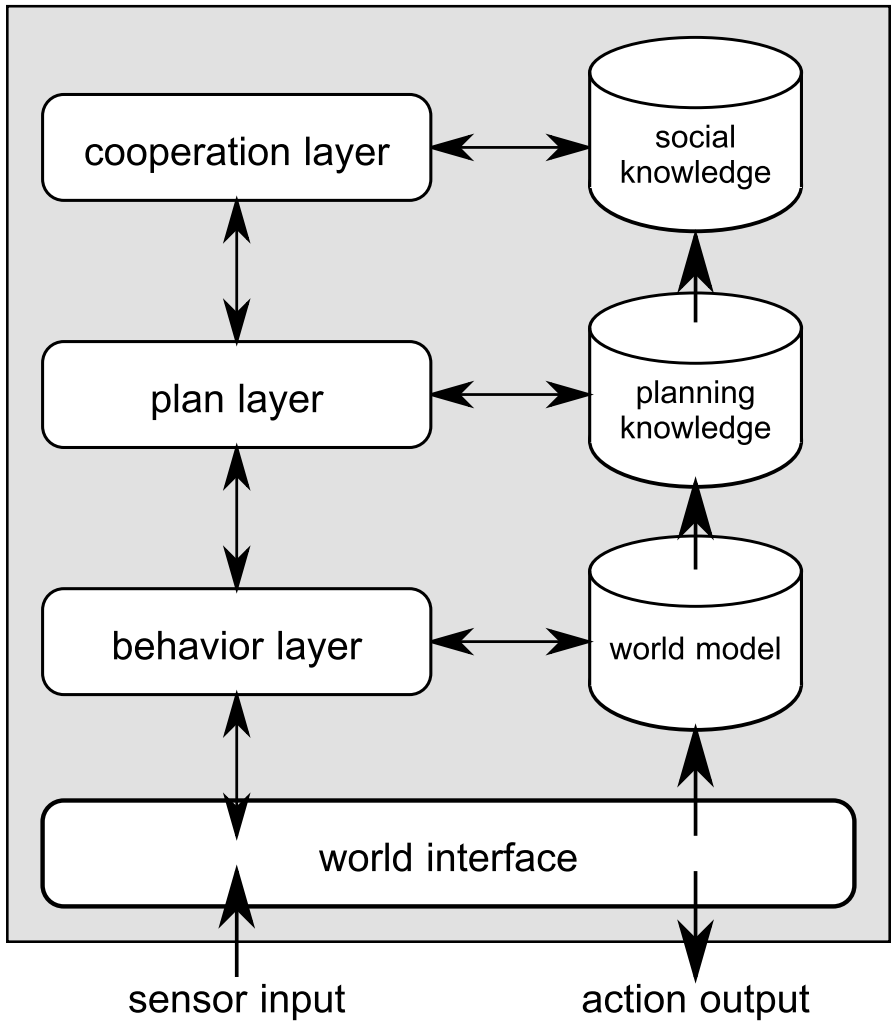
- situation recognition and goal activation module (SG)
- decision-making and execution module (DE)

Every layer contains a specific kind of knowledge base:

- World model
- Mental model
- Social model

Only knowledge bases of lower layers can be utilized by any one layer (nice principle for decomposition of large KB's)
$\Rightarrow$ very powerful and expressive, but highly complex!

InteRRaP II



(after Wooldridge 2009, p. 97)

5.4 Summary

- Thanks

Summary

- Agent architectures: deliberative, reactive and hybrid
- Tension between reactivity and proactiveness
- BDI architecture: "intentional stance", computationally heavy
- Subsumption architecture: effective, but reasons for success sometimes "obscure" ("black-box" character)
- Hybrid architecture: attempt to balance both aspects, but increased complexity (and lack of conceptual clarity)

$\Rightarrow$ Next time: **Agent Communication**

Acknowledgments

These lecture slides are based on the following resources:

- Dr. Michael Rovatsos, The University of Edinburgh
  `http://www.inf.ed.ac.uk/teaching/courses/abs/abs-timetable.html`
- Michael Wooldridge: **An Introduction to MultiAgent Systems**, John Wiley & Sons, 2nd edition 2009 and 1st edition 2002.
- Rodney A. Brooks: **Elephants Don't Play Chess**, Robotics and Autonomous Systems 6, 1990, p. 3–15