IAT 888: Metacreation
Machines endowed with creative behavior

Philippe Pasquier
Office 565 (floor 14)
pasquier@sfu.ca

SFU

Assignments: for this Week

- Thoughts about the final project:
  - One page about your final project and how it relates to your research (PDF)

➡ **In-class discussion**

Assignments: for this Week

- Let's talk about the scheduling of the presentations:
    - Nathan and Automatic Architecture Design (Week 6)
    - Parjad and Digital Puppets (Week 8)
    - Alireza and Agent Programing (Week 7)
    - James and David Cope (Week 9)
    - Andrew and the Continuator (Week 10)
- The idea is to get an independent, readable document in a portable format (PDF).
- I would like (with your permission) to put these on-line (after correction based on the feedback you get in class) in order to:
    - Archive you work
    - Avoid redundancy in future runs
    - ...

Assignments: for this Week

- **Readings:**
  - Brooks, R. A. "A Robust Layered Control System for a Mobile Robot" , IEEE Journal of Robotics and Automation, Vol. 2, No. 1, March 1986, pp. 14–23;
  - OPTIONAL, Brooks, R. A., "Elephants Don't Play Chess" , Robotics and Autonomous Systems (6), 1990, pp. 3–15. (Available from the authors Web Page).
  - OPTIONAL, Luc Steels: “Fifty Years of AI: From Symbols to Embodiment - and Back”, in 50 Years of Artificial Intelligence, Lecture Notes in Computer Science, Volume 4850, 2006: 18-28
  - Hutzler, G., Gortais, B. From Computer Art to Ambient Displays, Machine Graphics and Vision, Vol. 13, nr 1/2, s. 181--191, 2004. (available online).

In-class Discussion

- **Brooks, R. A. "A Robust Layered Control System for a Mobile Robot" , IEEE Journal of Robotics and Automation, Vol. 2, No. 1, March 1986, pp. 14–23;**
  - OPTIONAL, Brooks, R. A., "Elephants Don't Play Chess" , Robotics and Autonomous Systems (6), 1990, pp. 3–15. (Available from the authors Web Page).
  - OPTIONAL, Luc Steels: “Fifty Years of AI: From Symbols to Embodiment - and Back”, in 50 Years of Artificial Intelligence, Lecture Notes in Computer Science, Volume 4850, 2006: 18-28

In-class Discussion

- Hutzler, G., Gortais, B. From Computer Art to Ambient Displays, Machine Graphics and Vision, Vol. 13, nr 1/2, s. 181--191, 2004.
  - Presents the work
  - Argue that the work is also an ambient display (art integrated in life, ...)
  - Present some validation experiments:
    - Test if the representation is accurate/expressive enough
    - Test if inference can be made on the base of the representation
    - Compare with other representations

?

Reading is to the mind what exercise is to the body.

Joseph Addison

So far, … so good

- So far we went through:
  - Introduction to metacreation
  - Elements on creativity
  - Introduction to agents
  - Cognitive agents: the BDI model
  - Case study: The Garden of Chance
  - Case study: MAMA
  - Elements of theories of communication
  - Agent communication
- Today, we will focus on:
  - Reactive agents and hybrid architectures

Autonomous Agents (Part 2)

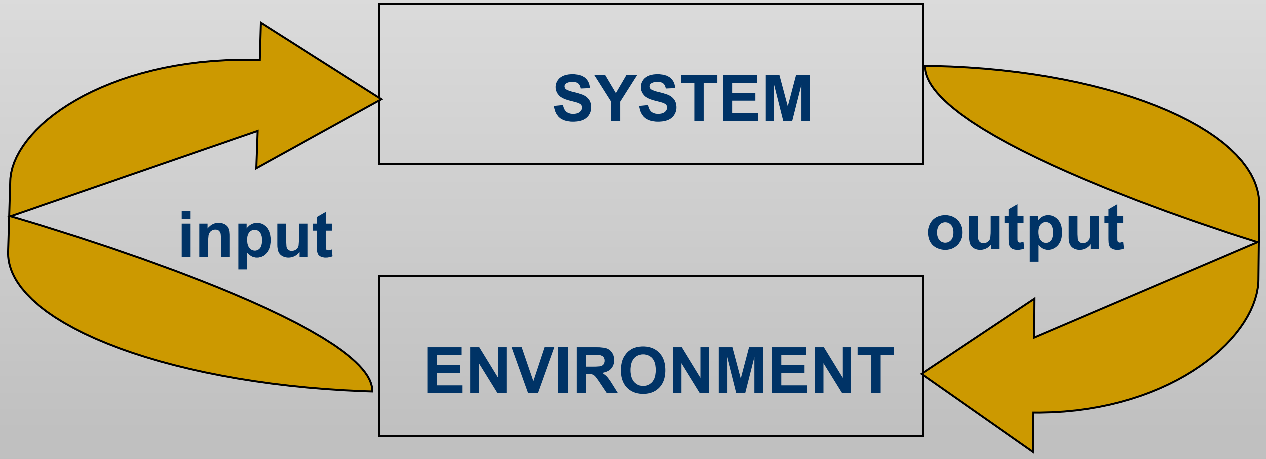
Based on, and inspired by slides from:
Michael Wooldridge, Jeff Rosenshein and Michael Rovatsos

Philippe Pasquier
Office 565 (floor 14)
pasquier@sfu.ca

SFU

Autonomous agents

- An agent is a computer system capable of autonomous flexible action in some environment in order to meet its design objectives



Summary

- An agent is a computer system capable of **flexible autonomous** action in some **environment**.
- **Situatedness**: perceiving the environment via sensors and being able to affect the environment via effectors
- **Autonomy**: a capability of action without intervention, and control over internal state
- **Flexibility**:
  - Responsiveness: respond in a timely fashion to change in the environment
  - Pro-activity: actions which go beyond simple response to stimulus
  - **Sociability**: ability to interact with other agents and humans for mutual benefit

Types of Agents

- Three types of agent architectures:
  - Cognitive: Also called deliberative architectures

    ➡ Focus on symbolic reasoning and planing
  - Reactive: no representation of the environment
    - Reflex: no internal states (just input->output rules)
    - Reactive: with internal states (but not cognitive)

    ➡ Focus on reactivity (based on behavioral rules)
  - Hybrid: mixing reactive and cognitive components

    ➡ Attempt to balance reactiveness and deliberativeness

Reactive Architectures

- There are many unsolved (some would say insoluble) problems associated with symbolic AI (based on knowledge representation)
- These problems have led some researchers to question the viability of the whole paradigm, and to the development of reactive architectures
- Although, united by a belief that the assumptions underpinning mainstream AI are in some sense wrong, reactive agent researchers use many different techniques
- In this presentation, we start by reviewing the work of one of the most vocal critics of mainstream AI: Rodney Brooks
- This will also remind us that agent technologies are applied both in software and robotics

Brooks – behavior languages

"Elephants don't play chess"

Brooks has put forward three theses:

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

Brooks – behavior languages

He identifies three key ideas that have informed his research:

1. **Situatedness and embodiment**: 'Real' intelligence is situated in the world, not in disembodied systems such as theorem provers or expert systems.

   → This can be captured (to some extent) by the idea of agent (cognitive agents included)

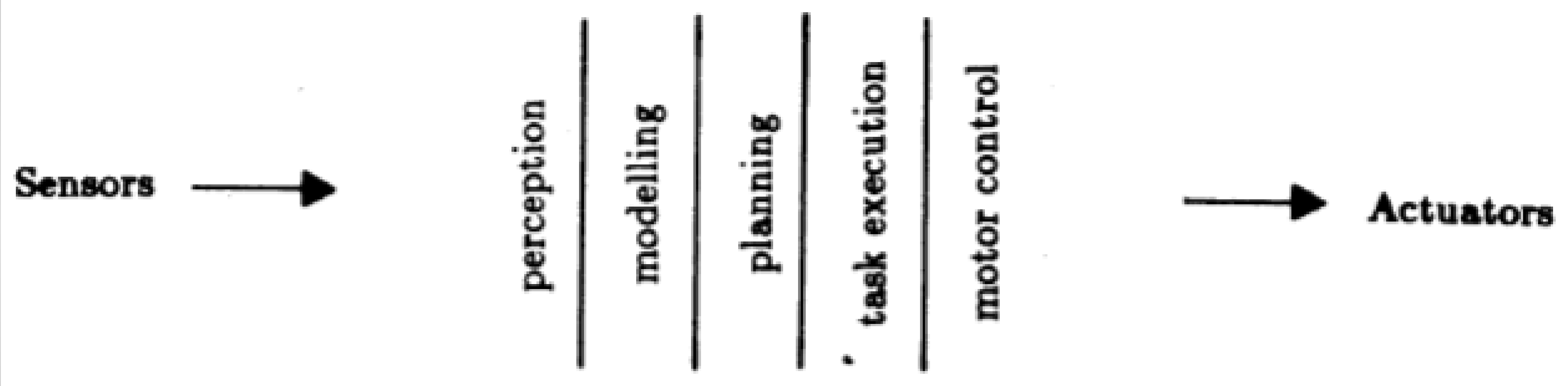
2. Intelligence is 'in the eye of the beholder'; it is not an innate, isolated property.
3. **Intelligence and emergence**: 'Intelligent' behavior arises as a result of an agent's interaction with its environment.

Brooks – behavior languages

- To illustrate his ideas, Brooks built some systems based on his subsumption architecture
- A subsumption architecture is a hierarchy of task-accomplishing behaviors
- Each behavior is a rather simple rule-like structure
- Each behavior 'competes' with others to exercise control over the agent
- Lower layers represent more primitive kinds of behavior (thought to be critical, such as avoiding obstacles), and have precedence over layers further up the hierarchy
- The resulting systems are, in terms of the amount of computation they do, extremely simple

From cognitive agents

A traditional decomposition of a mobile robot control system into functional modules:

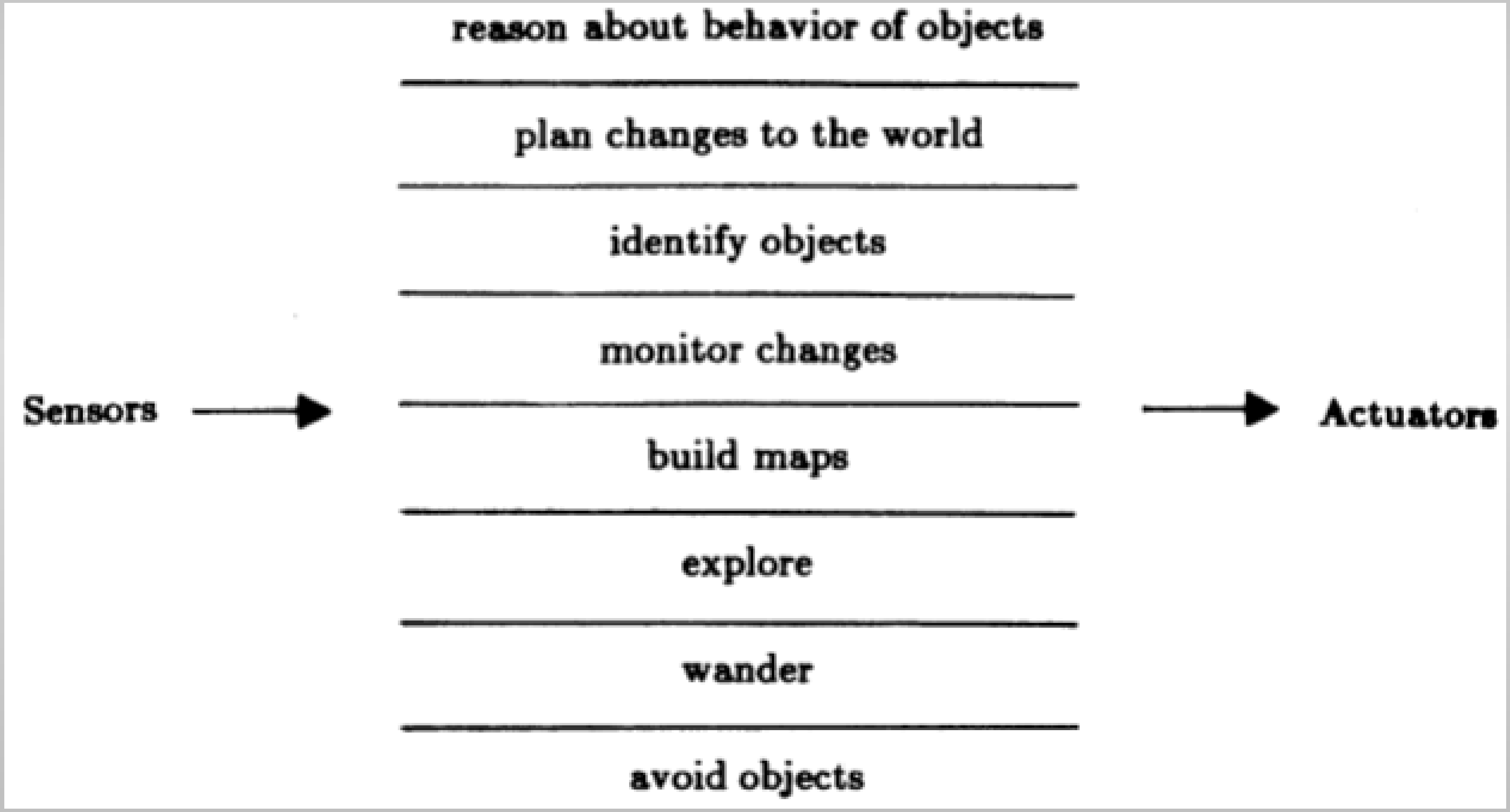


Note the similarity/analogy with the BDI architecture

From Brooks, "A Robust Layered Control System for a Mobile Robot", 1985

To Reactive Agents

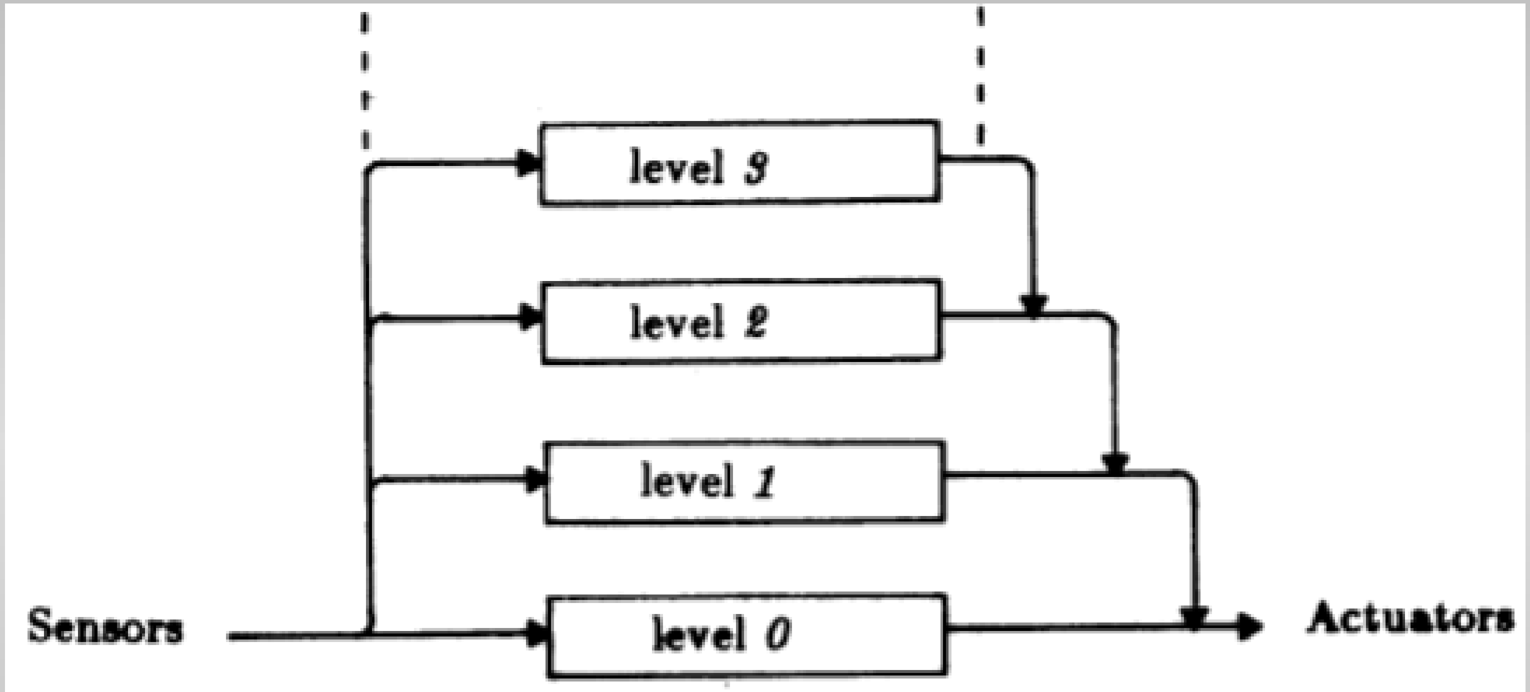
A decomposition of a mobile robot control system based on tasks achieving behaviors:



From Brooks, “A Robust Layered Control System for a Mobile Robot”, 1985

Subsumption Architecture

Layered control in the subsumption architecture:



- **The system can be partitioned at any level, the layers bellow will form an operational system**

From Brooks, “A Robust Layered Control System for a Mobile Robot”, 1985

Example: Steels' Mars Explorer

- **Luc Steels' Mars explorer system, using the subsumption architecture, achieves near-optimal cooperative performance in simulated 'rock gathering on Mars' domain:**

  The objective is to explore a distant planet, and in particular, to collect sample of a precious rock. The location of the samples is not known in advance, but it is known that they tend to be clustered. There is a radio signal from the mother ship to find the way back.

Steels' Mars Explorer Rules

- Each agent is made of five behavioral rules:
- For individual (non-cooperative) agents, the lowest-level behavior, (and hence the behavior with the highest "priority") is obstacle avoidance:
  - *if* detect an obstacle *then* change direction (1)
- Any samples carried by agents are dropped back at the mother-ship:
  - *if* carrying samples *and* at the base *then* drop samples (2)
- Agents carrying samples will return to the mother-ship:
  - *if* carrying samples and *not* at the base *then* follow the radio signal (3)

Steels' Mars Explorer Rules

- Agents will collect samples they find:
  - *if* detect a sample *then* pick sample up (4)
- An agent with "nothing better to do" will explore randomly:
  - *if* true *then* move randomly (5)

⇨ **Works well but does not take into account the fact that the samples are clustured. There is a potential for cooperation (through coordination and thus interaction).**

Steels' Mars Explorer Rules

- When finding a sample, it would be helpful to tell others, but direct communication is not available
- Inspiration from ants' foraging behaviour:
  - Agent will create trails by dropping crumbs of rock on way back to base, other agents will pick these up (making the trail fainter)
  - If agents find that a trail didn't lead to more samples, they won't reinforce trail
- Modified set of behaviours:
  1. *If* detect an obstacle *then* change direction
  2. *If* carrying samples and at the base *then* drop samples
  3. *If* carrying samples and not at the base *then* drop 2 crumbs and follow the radio signal
  4. *If* detect a sample *then* pick sample up
  5. *If* sense crumbs *then* pick up 1 crumb and travel down gradient
  6. *If* true *then* move randomly

Advantages of Reactive Architecture

- Watch the result (simulated in Starlogo):
  - http://www.youtube.com/watch?v=kN0M49iqFRc
- Reactive architectures achieve tasks that would be considered very impressive using symbolic AI methods (why not use them for metacreation?)
- Advantages of reactive architectures include:
  - Simplicity
  - Economy
  - Computational tractability
  - Robustness against failure
  - Elegance

Limitations of reactive architectures

- But also 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? The departure from “knowledge level” implies a loss of transparency and readability
  - What if it doesn’t work? Purely reactive systems are typically hard to debug
  - Engineering emergence: lack of clear design methodology
  - Design becomes difficult with more than a few rules (dynamic interactions between rules become quickly too complex)
  - How about communication with humans?

Hybrid Architectures

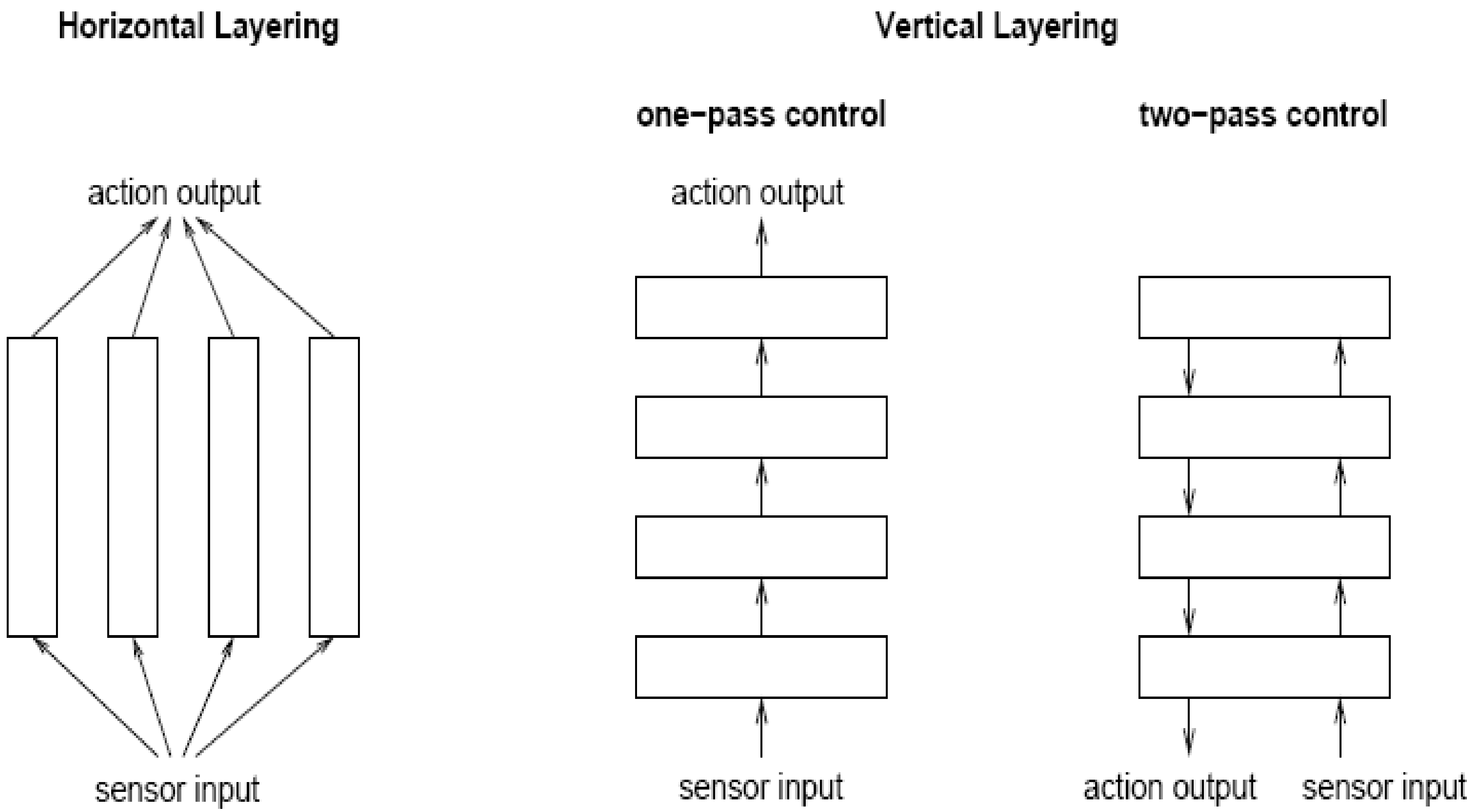
- Often, neither completely deliberative nor completely reactive architectures are suitable
- Hybrid architectures combine both perspectives in one agent/system/architecture
- Most obvious approach: construct an agent that consists of one (or more) reactive and one (or more) deliberative sub-components
  - Reactive sub-components would be capable of responding to world changes without any complex reasoning and decision-making
  - Deliberative sub-system would be responsible for abstract planning and decision-making using symbolic representations

→ **Meta-level control of interactions between these components is a key issue in hybrid architectures**

Hybrid Architectures

- Often, the reactive component is given some kind of precedence over the deliberative one
- This kind of structuring leads naturally to the idea of layered architecture:
  - Horizontal layering:
    - All layers are connected to sensory input/action output
    - Each layer produces an action, different suggestions have to be reconciled
  - 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

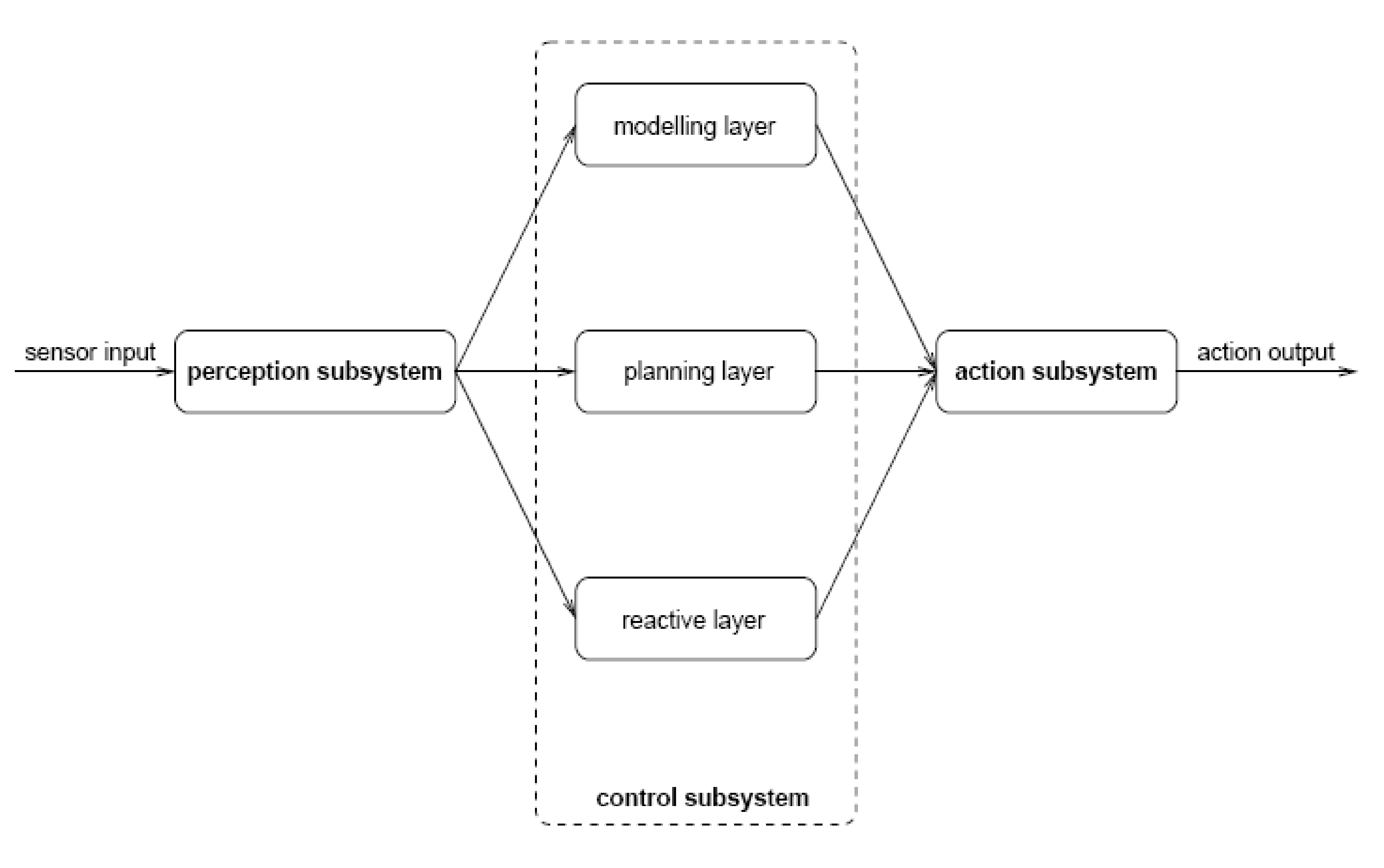


Example – TouringMachines

I. A. Ferguson, TouringMachines: Autonomous Agents with Attitudes, IEEE Computer, 25, 5, p.51-55, 1992.

- **Horizontal layering architecture**
- **Three sub-systems: (1) perception sub-system, (2) control sub-system and (3) action sub-system**
- **Control sub-system consists of:**
  - **Reactive layer: situation-action rules (*a la* subsumption architecture)**
  - **Planning layer: construction of plans and action selection**
  - **Modelling layer: contains symbolic representations of the environment (including mental states of other agents)**
- **The three layers communicate via explicit control rules**

Ferguson – TOURINGMACHINES



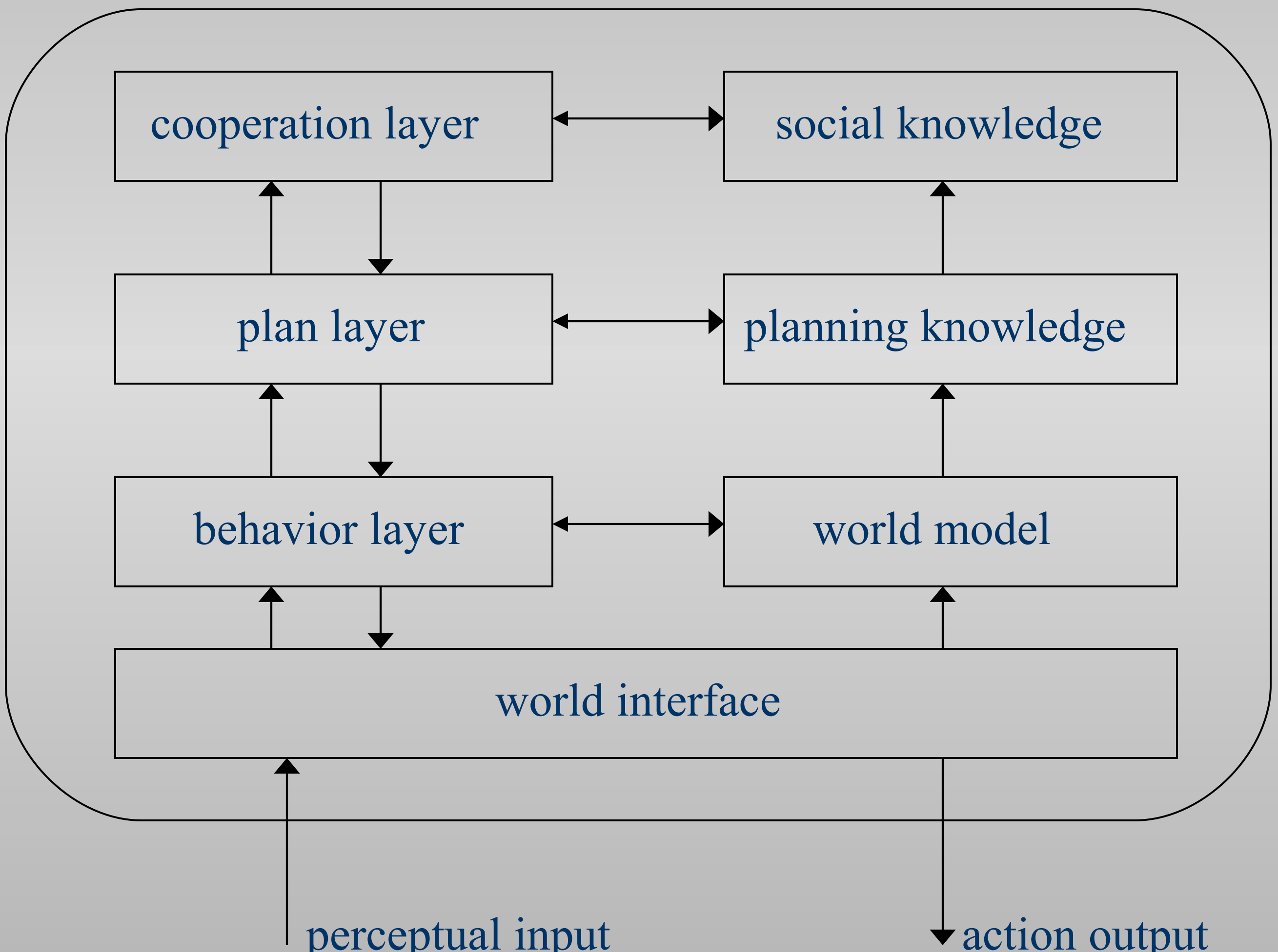
Example: InteRRaP

J. Muller and M. Pischel, The Agent Architecture InteRRaP: Concept and Application, Technical Report RR-93-26, DFKI Saarbrucken, 1993.

- InteRRaP: Integration reactive behaviour and rational planning
- Vertical (two-pass) layering architecture
- Three layers:
  - Behaviour-Based Layer: manages reactive behaviour of agent
  - Local Planning Layer: individual planning capabilities
  - 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

Example: InteRRaP

- **Vertically layered, two-pass architecture:**



Implementing Reactive Agents

- **Netlogo**: cross-platform multi-agent programmable modeling environment.
  - http://www.ccl.sesp.northwestern.edu/netlogo/
  - See dozens of applications here: http://www.ccl.sesp.northwestern.edu/netlogo/models/community/
- **Startlogo** (MIT version):
  - http://education.mit.edu/starlogo/
  - Example: http://www.youtube.com/watch?v=kN0M49iqFRc
- **RePast**:
  - http://repast.sourceforge.net/
- **Madkit**: built upon the AGR (Agent/Group/Role) organizational model.
  - http://www.madkit.org/
- Note that there are strong links between reactive agents and some A-life systems.

Other types of hybrid agent architectures

- Cognitive architectures: developed in the aim to render believable agents (rather than purely rational ones)
  - Try not to address isolated cognitive components
  - Closer to cognitive science and useful for simulation (rather than problem solving)
- Famous examples include:
  - CLARION
  - ACT-R
  - Soar
  - Cougaar
  - PRODIGY

Cognitive architectures are often hybrid architectures

Conclusion on Agents and MAS

- Agents and MAS are the new paradigm of AI:
  - Agents design: allow addressing cognition and individual behavior
  - Society Design: MAS (ex-Distributed Artificial Intelligence) allow addressing decentralisation and the gap between local and global behavior (emergence)
  - Agents and MAS bridge computer science with: cognitive sciences, economy, social sciences, ...
  - Can be applied to metacreation (perfect candidate!)

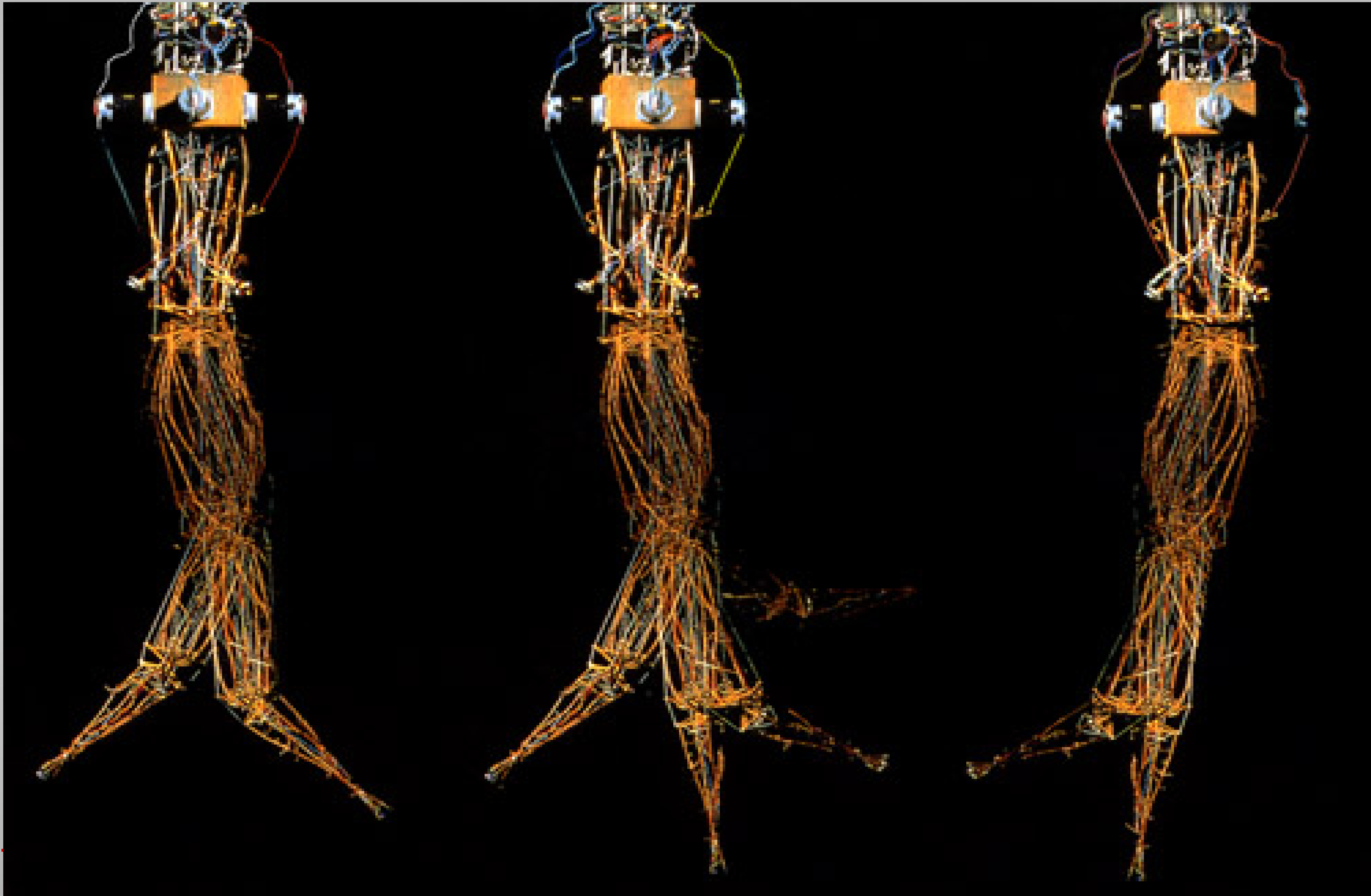
Examples of Metacreations

- **While our foci is on software machines, some robotic art pieces are applying Brooksian model for robotics (acknowledged explicitly):**
  - **Kenneth Rinaldo: *The Flock* involves distinct behavioral rules running in parallel.**
  - **Bill Vorn and Louis-Philippe Demers: various robotic ecosystems (*Espave Vectoriel*)**
  - **Simon Penny: Works and writings (*Petit Mal)***
- **But as Penny puts it: *"Although the principle of Brooksian subsumption architecture was influential, the software architecture of Petit Mal is a hybrid of various techniques ..."***

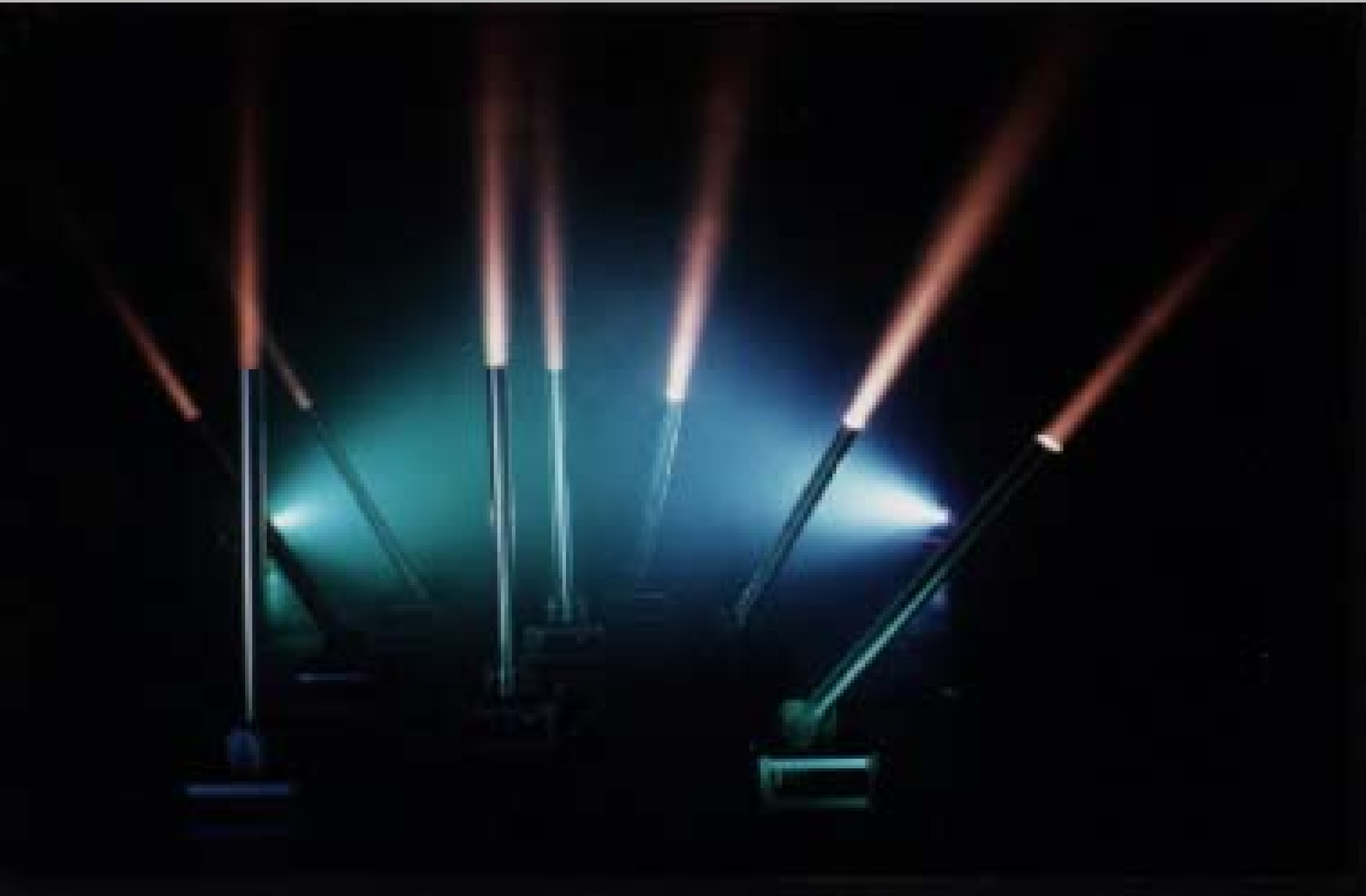
→ **Here, the absence of a strict methodology turns out to be an advantage!**

- ***Other artists using bottom-up emergent robotics: Ulrike Gabriel, Erwin Driessens and Maria Verstappen, Yves Amu Klein***

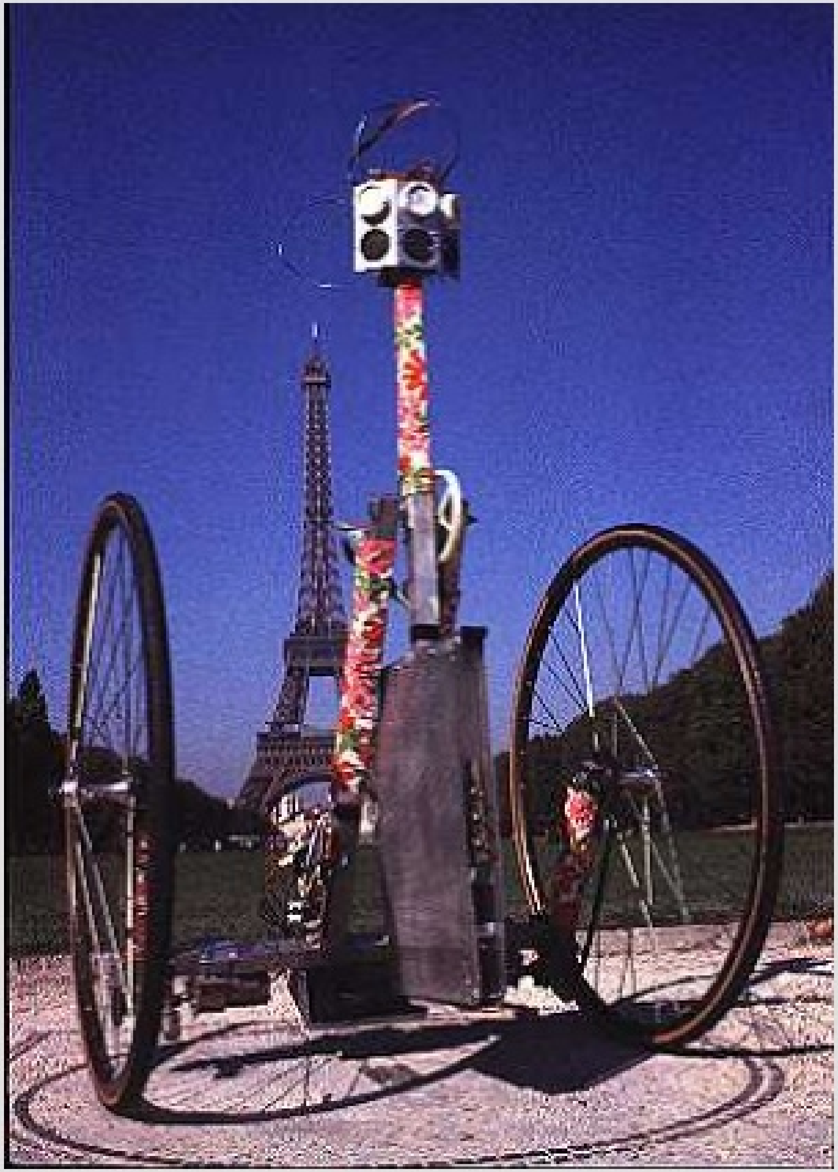
The Flock (Kenneth Rinaldo, Siggraph 1993.)



Espace Vectoriel (Vorn and Demers, 1994)



Petit Mal (Simon Penny, 1995)



Applications in Video Games, …

- Video games (aka Interactive Entertainment):
  - A version of Tetris using (behavior based) reactive agents
    - Wavish, P., Graham, M. (1996) A situated action approach to implementing characters in computer games. *Int. Journal of Applied Artificial Intelligence*, 10(1), 53-74.
  - Using cognitive agents for believable character (in video games and virtual theatre):
    - Hayes-Roth, B., Brownston, L., Sincoff, E., and van Gent, R. Directed improvisation by computer characters. Stanford Knowledge Systems Laboratory Report KSL-95-04, 1995.
  - While entertainment was not taken seriously by early AI researcher, it is now the reverse. Why? (17.9 billion dollars in 2007, in the U.S. only)

Examples of Metacreations

- We have been looking at:
  - The Garden of Chances (Guillaume Hutzler, Paris) – Visual - Agent/MAS
  - MAMA (David Murray-Rust, University of Edinburgh) – Sound – Speech Acts, Agent/MAS
- Other software machines exploiting agent and MAS concepts have also been proposed:
  - Bob (Belinda Thom, Carnegy Melon University) – Sound – Agent/Unsupervised learning
  - Kinetic Engine (Arne Engelfeld, SFU, Vancouver) – Sound – MAS
  - VMMAS (Whulfhorst, UFRGS, Rio) – Sound – MAS
  - Virtualatin (David Murray-Rust, University of Edinburgh) – sound – Agent/MAS
  - ANDANTE (Leo Ueda, University of Sao Paolo) – sound – Agent
  - Eden (Mc Cormack, CEMA, Monash University, Melbourne) – visual – MAS/A-life
  - Continuator (Francois Pachet, Sony lab, Paris) – sound – Agent + HMM
  - O-MAX Brothers (Assayag, Ircam, Paris) – sound – statistical learning + MAS
  - An interactive MIDI accompanist. (Toiviainen, P.) – sound – agent
  - SPAA or AALIVENET (Michael Spicer, Singapore*)* – sound – agent based

?

“When many work together for a goal, great things may be accomplished. It is said a lion cub was killed by a single colony of ants.”

Saskya Pandita (1182-1251)

Assignments: for next Week

- Readings:
  - Lippmann, R. An introduction to computing with neural nets. IEEE Signal Processing Magazine, Volume: 4, Issue: 2 , Part 1, page(s): 4- 22, 1987 (Available via IEEE Xplore system, accessible via SFU library electronic collection)
  - Dolson, M., Machine Tongues XII: Neural Networks, Computer Music Journal, Vol. 13, No. 3., pp. 28-40, 1989. (Available through JSTOR, accessible via SFU library electronic collection)
  - Todd, P.M. Neural networks for applications in the arts. In M. Scott (Ed.), Proceedings of the Eleventh Annual Symposium on Small Computers in the Arts (pp. 3-8). Philadelphia, PA: Small Computers in the Arts Network, Inc. 1991. (Available on the author's Web page).
  - OPTIONAL, Lewis, J.P. Shape and Texture Generation by Neural Network Creation Paradigm, Proceedings of Graphics Interface '91, pages 129-134, 1991. (Available online)