LECTURE 5: REACTIVE AND HYBRID ARCHITECTURES

An Introduction to MultiAgent Systems
http://www.csc.liv.ac.uk/~mjw/pubs/imas

Reactive Architectures

- There are many unsolved (some would say insoluble) problems associated with symbolic AI
- 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

Brooks – behavior languages

- Brooks has put forward 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

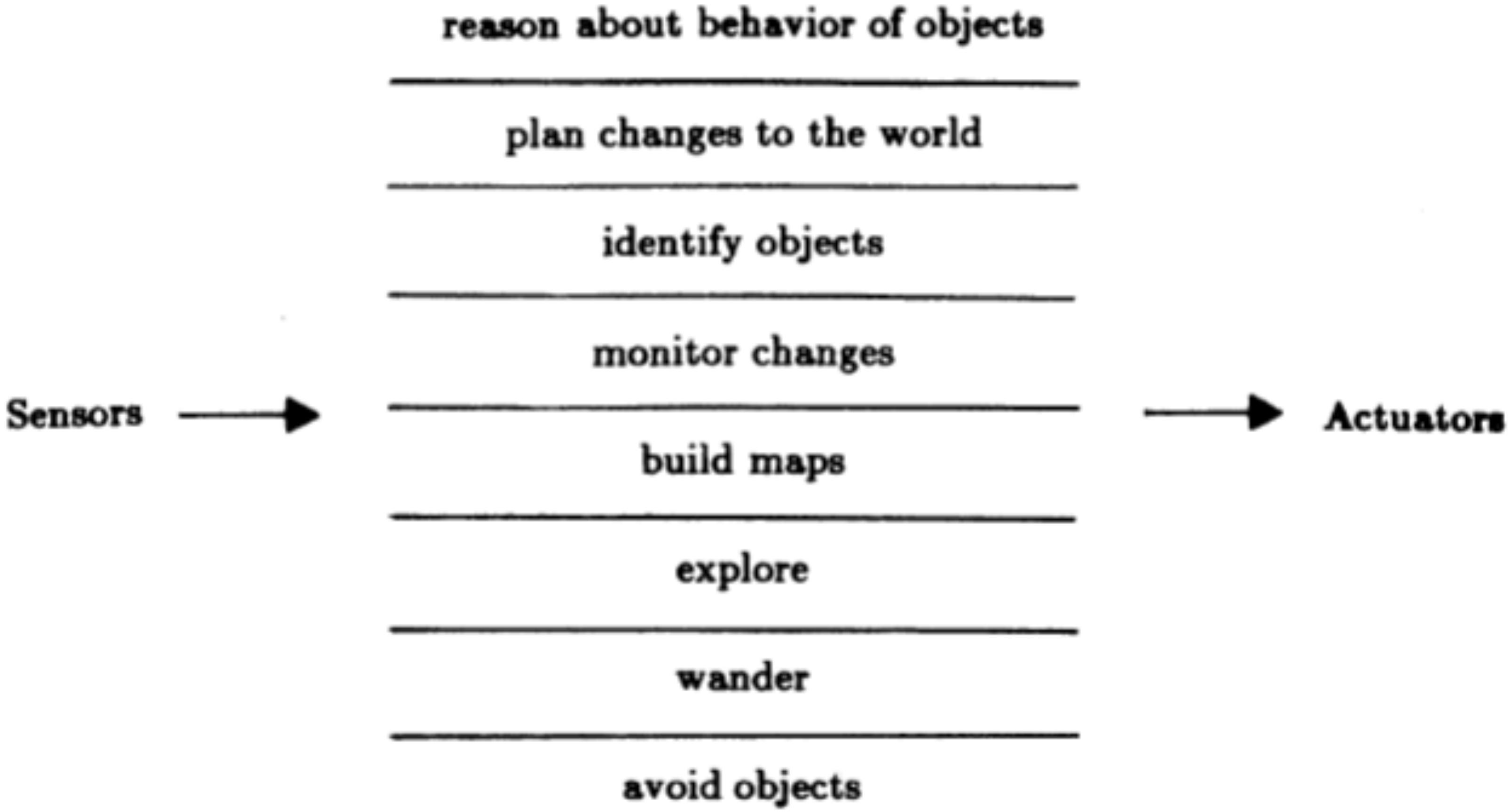
Brooks – behavior languages

- He identifies two 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
  2. Intelligence and emergence: ‘Intelligent’ behavior arises as a result of an agent’s interaction with its environment. Also, intelligence is ‘in the eye of the beholder’; it is not an innate, isolated property

Brooks – behavior languages

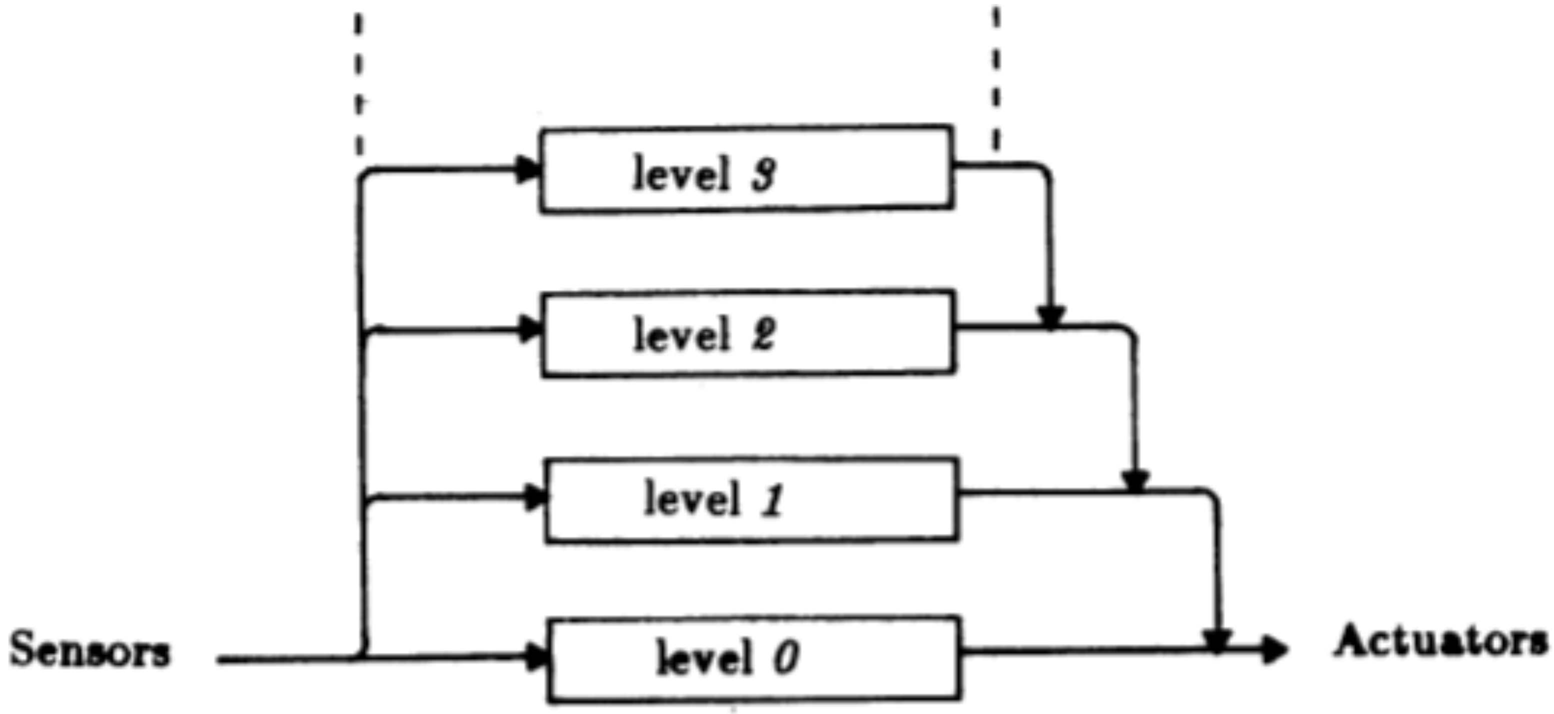
- To illustrate his ideas, Brooks built some based on his *subsumption architecture*
- A subsumption architecture is a hierarchy of task-accomplishing *behaviors*
- Each behavior is a rather simple rule-like structure
- Each behavior ‘competes’ with others to exercise control over the agent
- Lower layers represent more primitive kinds of behavior (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
- Some of the robots do tasks that would be impressive if they were accomplished by symbolic AI systems

A Decomposition of a Mobile Robot Control System Based on Task Achieving Behaviors



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

Layered Control in the Subsumption Architecture



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

Steels' Mars Explorer

- 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.*

Steels' Mars Explorer 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 travel up gradient (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)

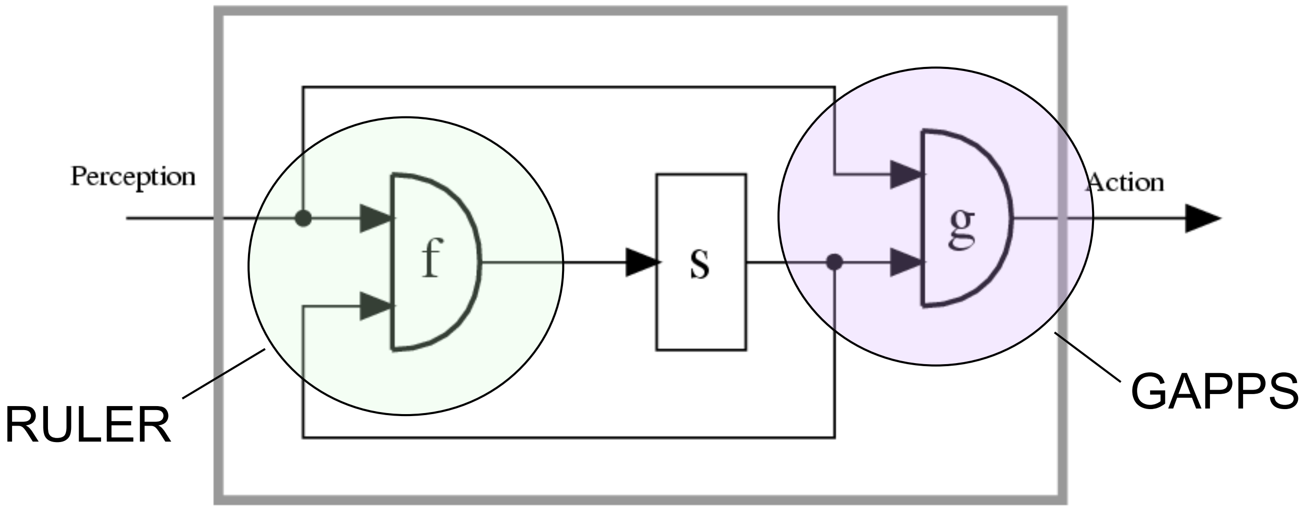
Situated Automata

- A sophisticated approach is that of Rosenschein and Kaelbling
- In their *situated automata* paradigm, an agent is specified in a rule-like (declarative) language, and this specification is then compiled down to a digital machine, which satisfies the declarative specification
- This digital machine can operate in a *provable time bound*
- Reasoning is done *off line*, at *compile time*, rather than *online* at *run time*

Situated Automata

- An agent is specified in terms of two components: perception and action
- Two programs are then used to synthesize agents
  - RULER is used to specify the perception component of an agent
  - GAPPS is used to specify the action component

Circuit Model of a Finite-State Machine



“The key lies in understanding how a process can naturally mirror in its states subtle conditions in its environment and how these mirroring states ripple out to overt actions that eventually achieve goals.”

From Rosenschein and Kaelbling,
“A Situated View of Representation and Control”, 1994

Situated Automata

- The theoretical limitations of the approach are not well understood
- Compilation (with propositional specifications) is equivalent to an NP-complete problem
- The more expressive the agent specification language, the harder it is to compile it
- (There are some deep theoretical results which say that after a certain expressiveness, the compilation simply can't be done.)

Advantages of Reactive Agents

- Simplicity
- Economy
- Computational tractability
- Robustness against failure
- Elegance

Limitations of Reactive Agents

- Agents without environment models must have sufficient information available from local environment
- If decisions are based on *local* environment, how does it take into account *non-local* information (i.e., it has a "short-term" view)
- Difficult to make reactive agents that learn
- Since behavior emerges from component interactions plus environment, it is hard to see how to *engineer* specific agents (no principled methodology exists)
- It is hard to engineer agents with large numbers of behaviors (dynamics of interactions become too complex to understand)

Hybrid Architectures

- Many researchers have argued that neither a completely deliberative nor completely reactive approach is suitable for building agents
- They have suggested using *hybrid* systems, which attempt to marry classical and alternative approaches
- An obvious approach is to build an agent out of two (or more) subsystems:
  - a *deliberative* one, containing a symbolic world model, which develops plans and makes decisions in the way proposed by symbolic AI
  - a *reactive* one, which is capable of reacting to events without complex reasoning

Hybrid Architectures

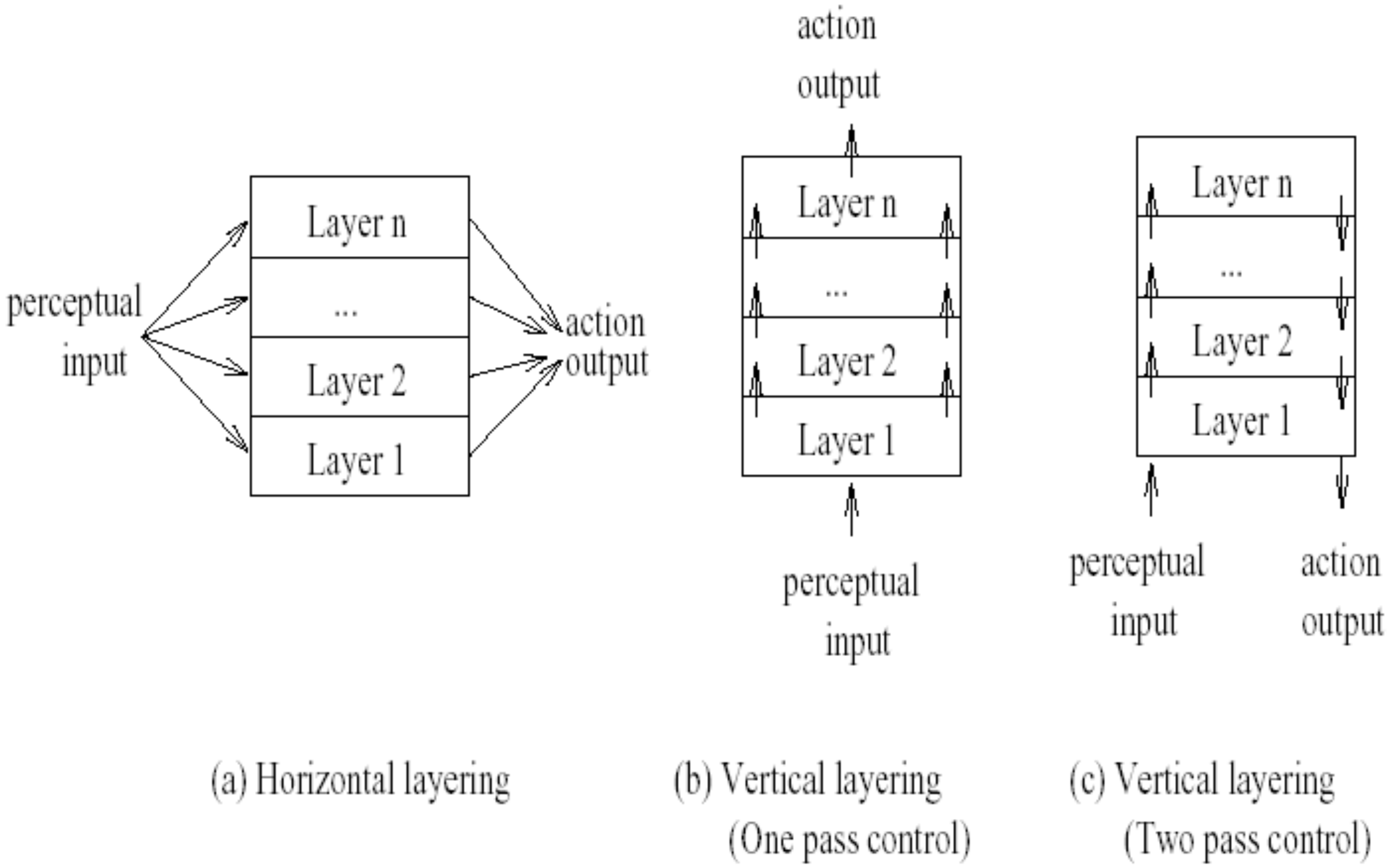
- Often, the reactive component is given some kind of precedence over the deliberative one
- This kind of structuring leads naturally to the idea of a *layered* architecture, of which TOURINGMACHINES and INTERRAP are examples
- In such an architecture, an agent's control subsystems are arranged into a hierarchy, with higher layers dealing with information at increasing levels of abstraction

Hybrid Architectures

- A key problem in such architectures is what kind of control framework to embed the agent's subsystems in, to manage the interactions between the various layers
- *Horizontal layering*
  Layers are each directly connected to the sensory input and action output.
  In effect, each layer itself acts like an agent, producing suggestions as to what action to perform.
- *Vertical layering*
  Sensory input and action output are each dealt with by at most one layer each

Hybrid Architectures

m possible actions suggested by each layer, *n* layers



m^n interactions

$m^2(n-1)$ interactions

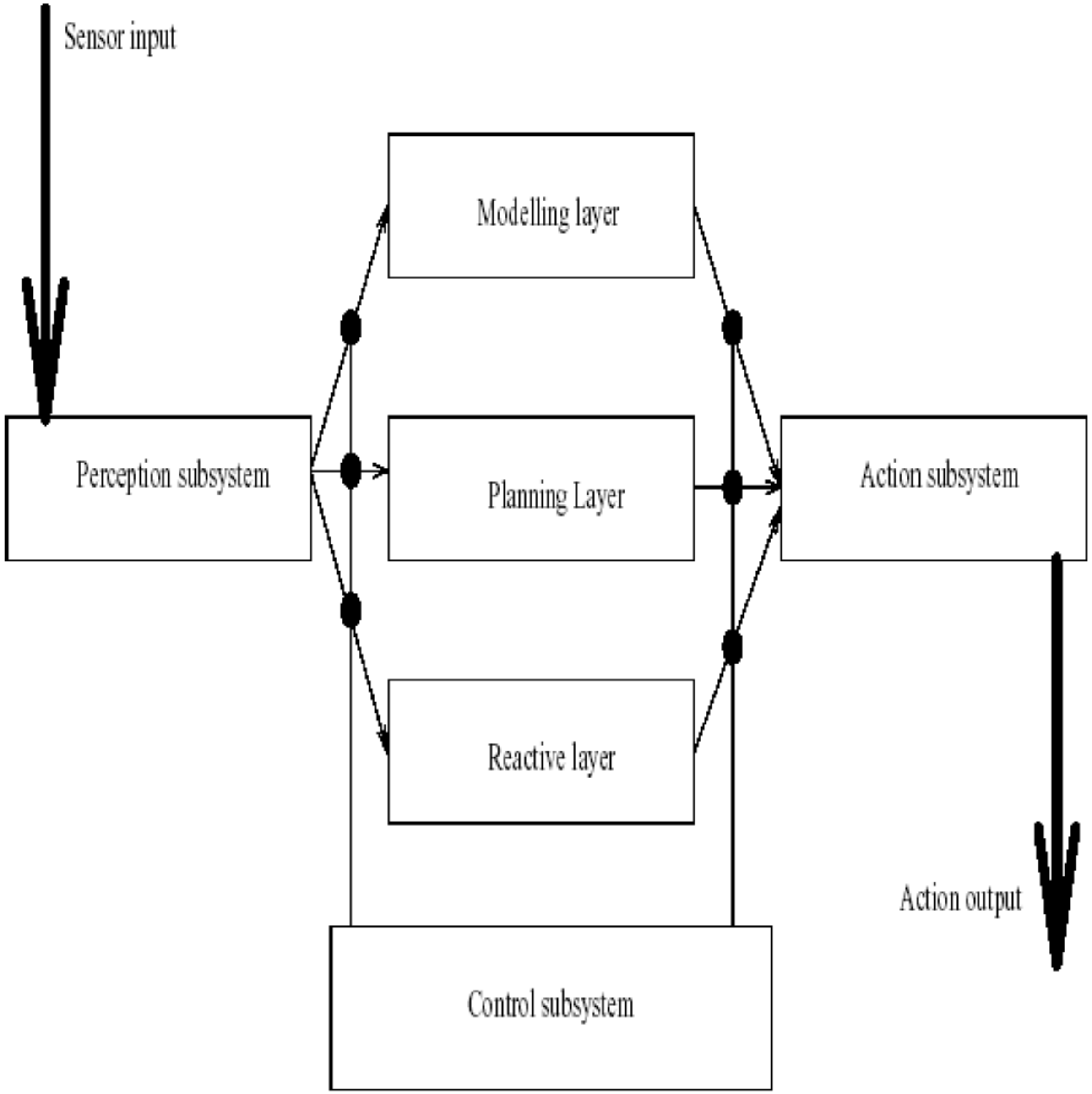
Introduces bottleneck
in central control system

Not fault tolerant to
layer failure

Ferguson – TOURINGMACHINES

- The TOURINGMACHINES architecture consists of *perception* and *action* subsystems, which interface directly with the agent's environment, and three *control layers*, embedded in a *control framework*, which mediates between the layers

Ferguson – TOURINGMACHINES



Müller –InteRRaP

- Vertically layered, two-pass architecture