Intelligent Control and Cognitive Systems

brings you...

Cognitive Architectures

Joanna J. Bryson

University of Bath, United Kingdom

From Last Week

- Combinatorics is the problem, search is the only solution.
- The task of intelligence is to focus search.
  - Called bias (learning) or constraint (planning).
  - Most `intelligent' behavior has no or little real-time search (non-cognitive) (c.f. Brooks IJCAI91).
- For artificial intelligence, most focus from design.

Architectures

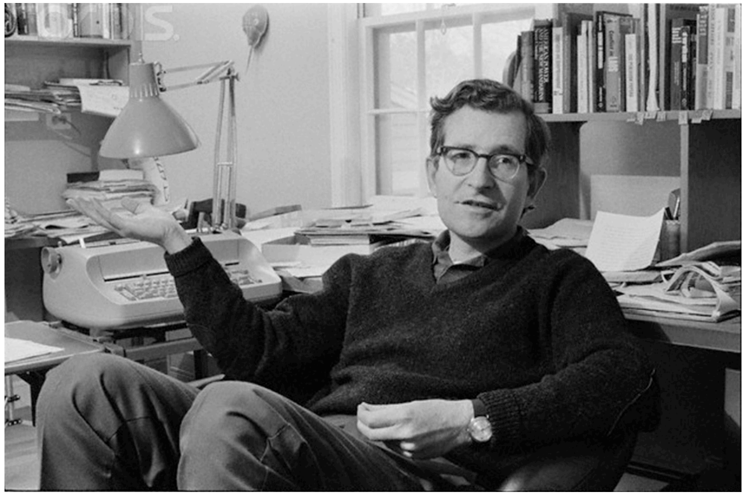
- What kinds of parts does the system need?
  - Ontology
- How should those parts be put together?
  - Development methodology
- How exactly is the whole thing arranged?
  - Architecture

“Architectures?”

- Like *reactive planning*, the term *cognitive architecture* doesn’t quite mean what its component words do.
- People have been looking for a generic plan for building “real” (human-like) AI.
- This used to be a popular area of research, now gets fewer publications.
- Nevertheless, evolutionary history tells us something about what worked & didn’t.

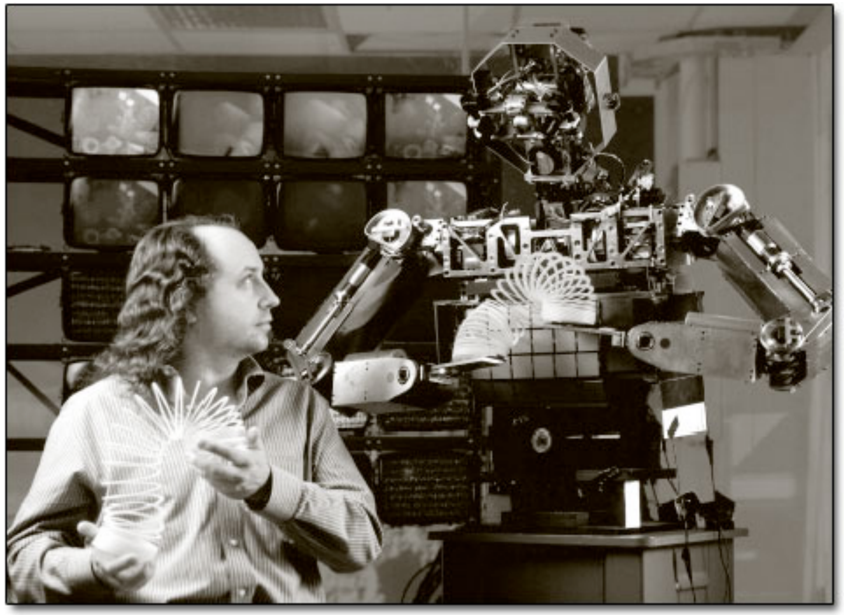
What Worked

- The past does not necessarily predict the future, particularly in AI.
- Changes in hardware and other tech change what is possible.



Cognitive Architecture

- Where do you put the cognition?
- Really: How do you bias / constrain / focus cognition (learning, search) so it works?



Basic Unit– Production

- From sensing to action (c.f. Skinner; conditioning; Witkowski 2007.)
- These work -- basic component of intelligence.
- The problem is choice (search).
  - Require an arbitration mechanism.

Production-Based Architectures

*arbitration mechanisms

- Expert Systems: allow choice of policies, e.g. recency, utility, random.
- SOAR: problem spaces (from GPS), impasses, chunk learning.
- ACT-R: (Bayesian) utility, problem spaces (reluctantly, from SOAR/GPS.)

Expert Systems

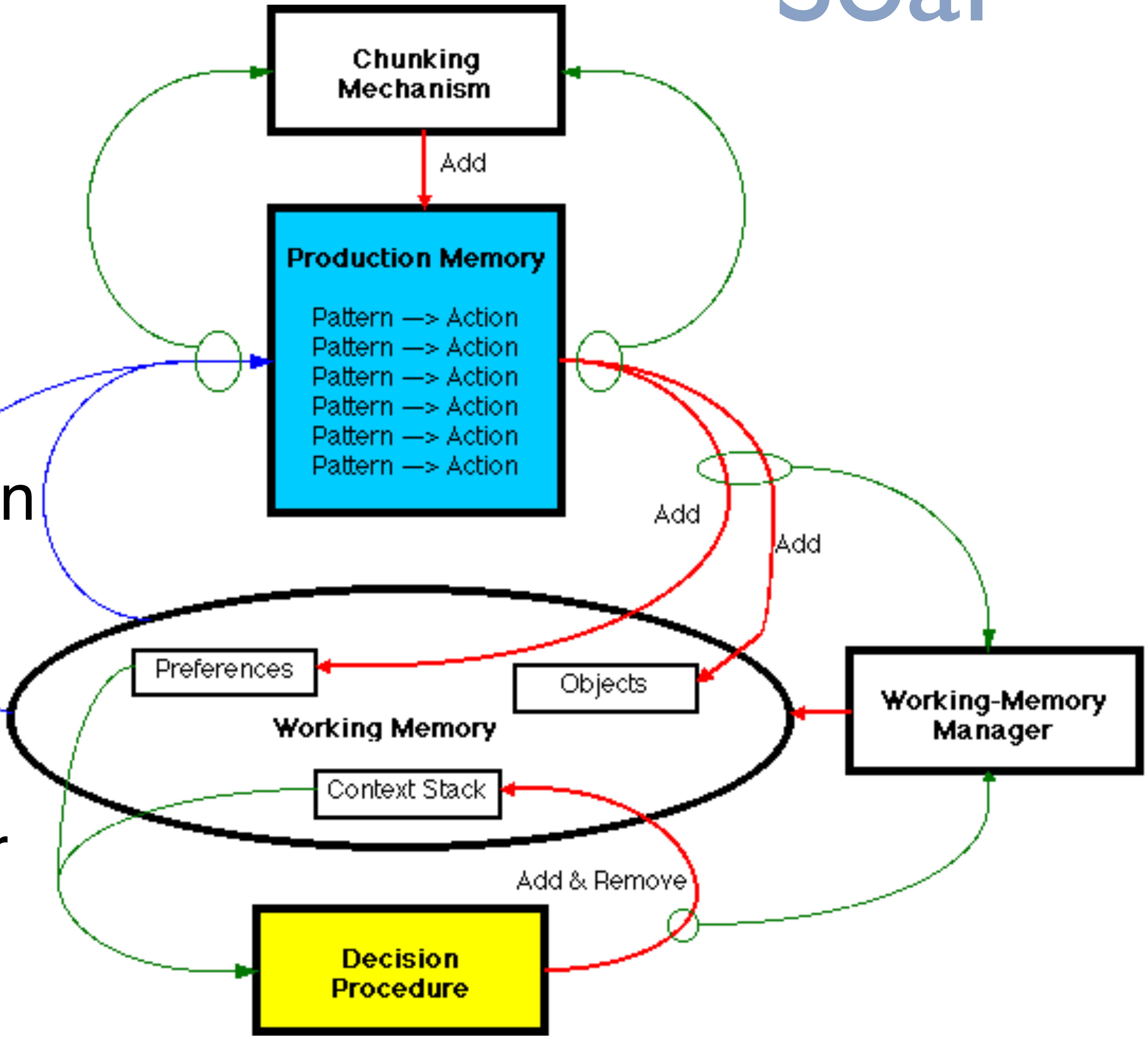
- Idea: Encode the knowledge of a domain expert as productions, replace them with AI.
- Big hype in 1980s, do still exist e.g. for checking circuit boards, credit / fraud detection, device driver code.
- Problem: Experts don't know why they do what they do, tend to report novice knowledge (last explicit rules learned.)

General Problem Solver

- GPS, written by Newell, Shaw & Simon (1959, CMU), first program that separated specific problem (coded as productions) from reasoning system.
- Cool early AI, but suffered from both combinatorial explosion and the Markov assumption.
- Soar was Newell's next try.

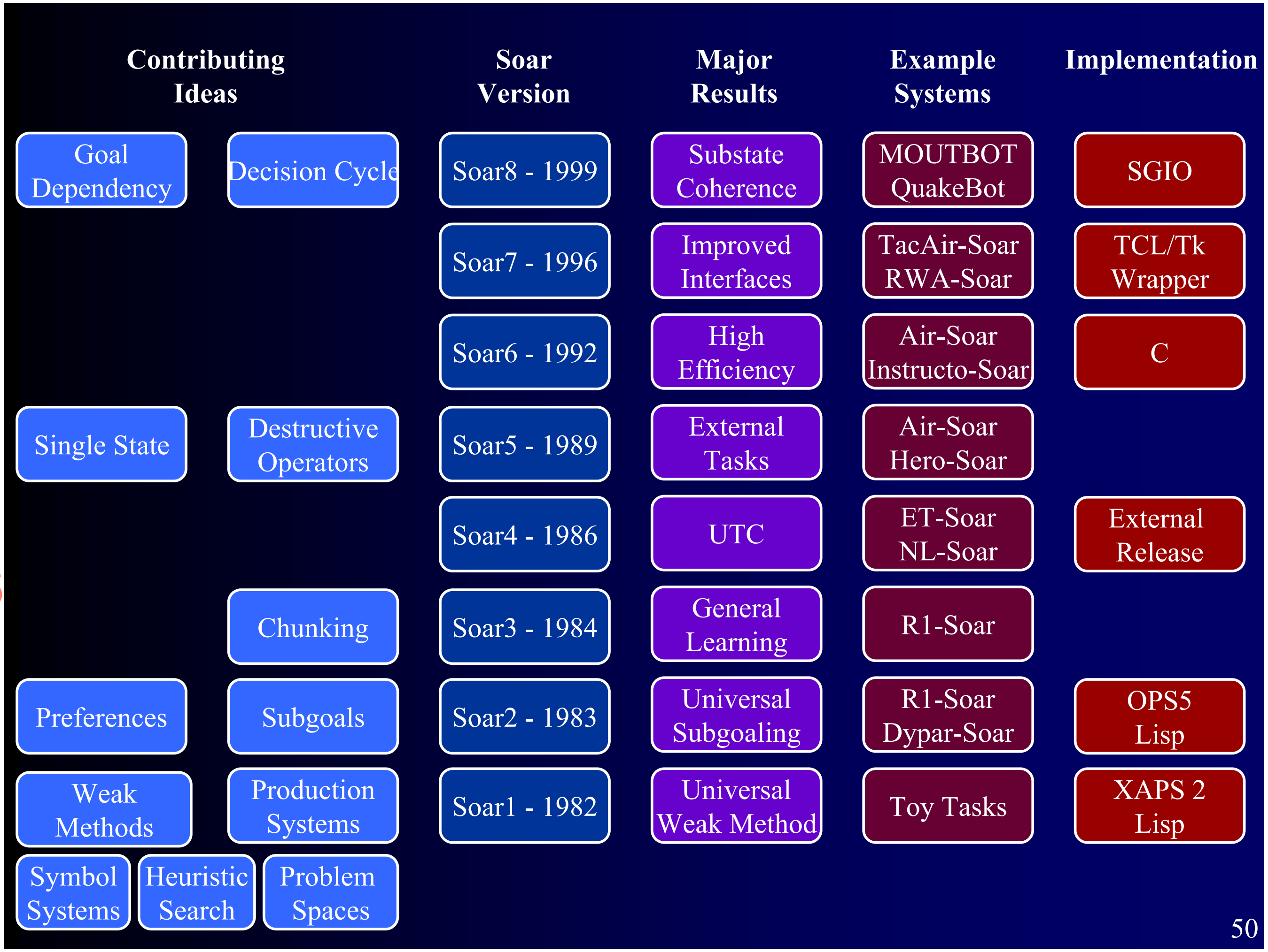
Soar

- Productions operate on a predicate database.
- If conflict, declare impasse, then reason (search harder).
  - Remember resolution: chunk



Soar

- Soar has serious engineering.
- "Evolution of Soar" is a favourite AI paper (Laird & Rosenbloom 1996) – admits problems & mistakes!
  - Not enough applications for human-like AI



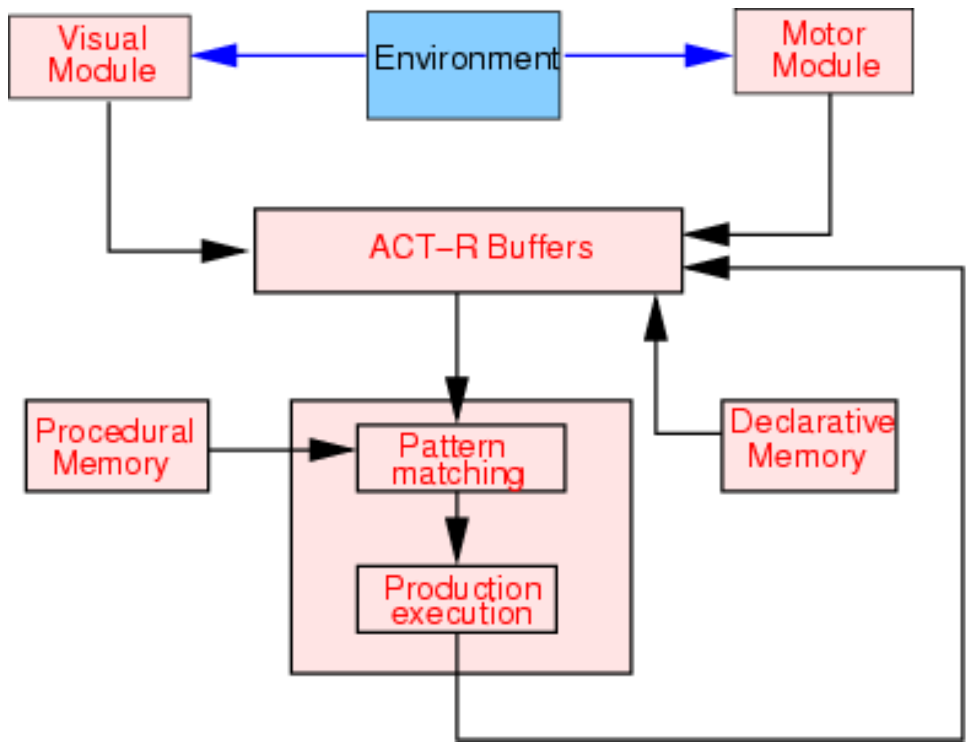
← One problem: main ap / funding is war games for US military.

Architecture Lessons (from CMU➢Michigan)

- An architecture needs:
  - action from perception, **and**
  - further structure to combat combinatorics.
- Dealing with time is hard (Soar 5).

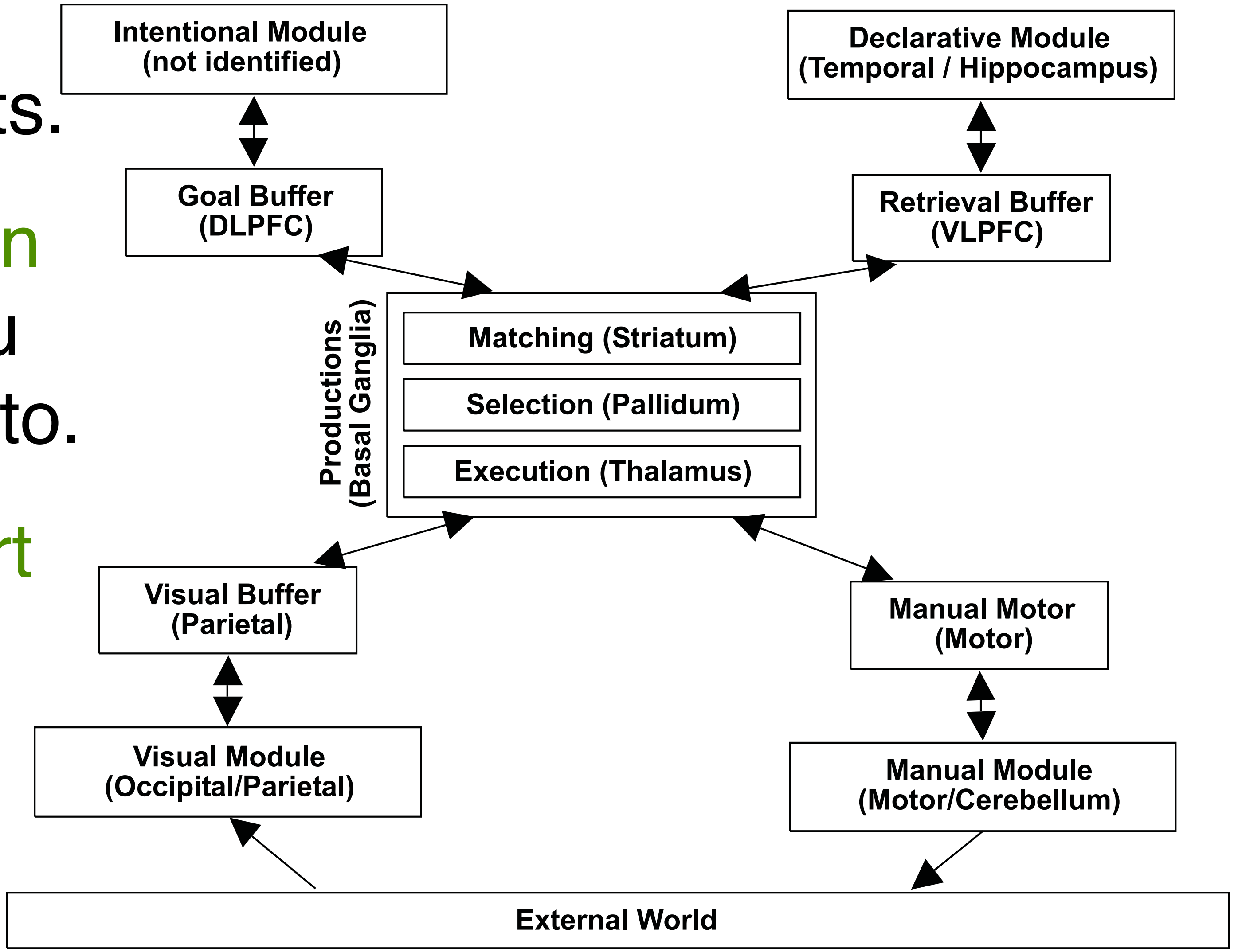
ACT-R

- Learns (& executes) productions.
- For arbitration, relies on (Bayesian probabilistic) utility.
  - Call utility "implicit knowledge".



ACT-R Research Programme

- Replicate lots of Cognitive Science results.
- See if the brain does what you think it needs to.
- Win Rumelhart Prize (John Anderson, 2000).

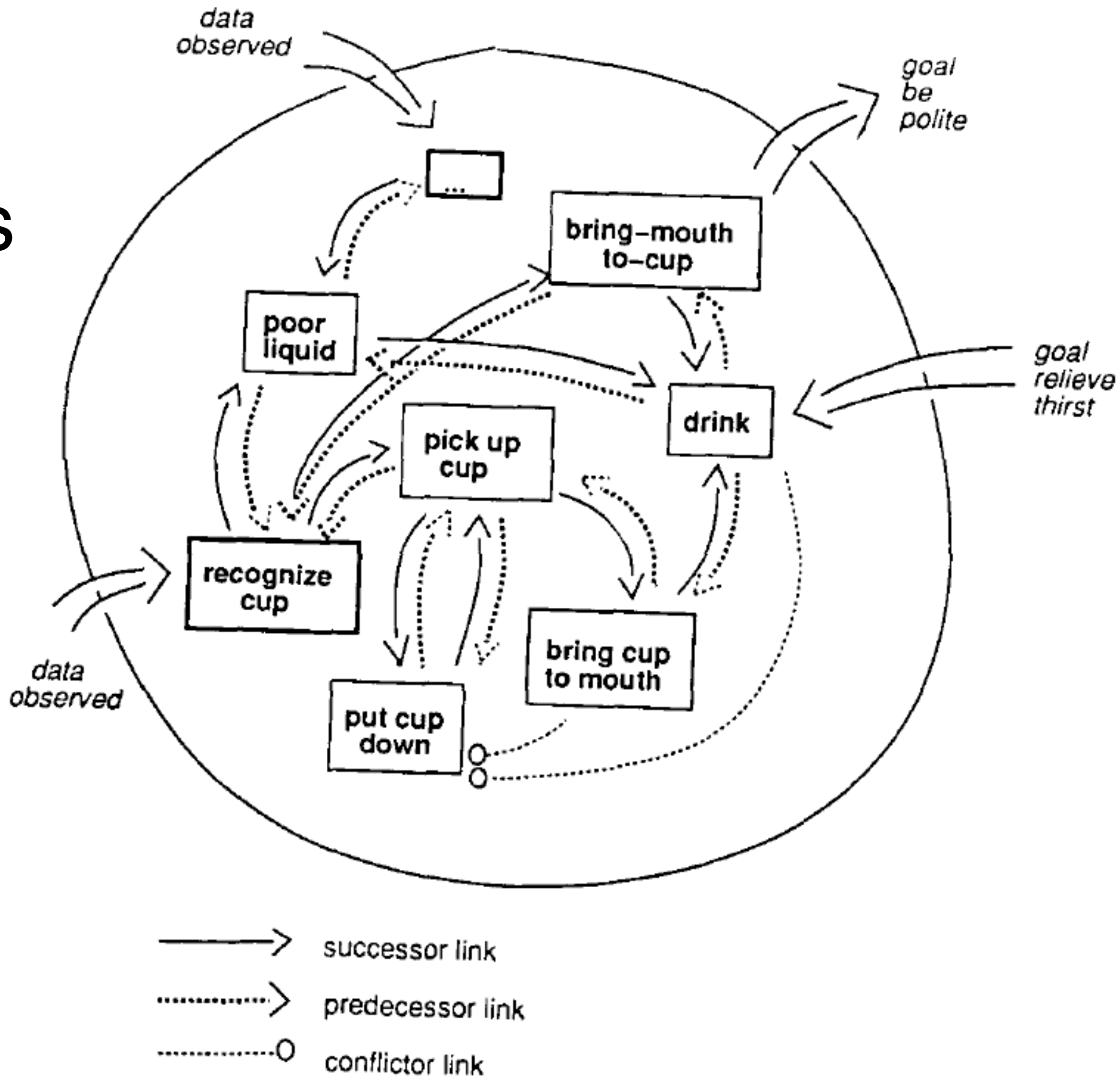


Architecture Lessons (from CMU Ψ)

- Architectures need productions and problem spaces.
- Real-time is hard.
- Grounding in biology is good PR, may be good science too.
- Being easy to use can be a win.

Spreading Activation Networks

- "Maes Nets" (Adaptive Neural Arch.; Maes 1989, VUB)
- Activation spreads from senses and from goals through net of actions.
- Highest activated action acts.



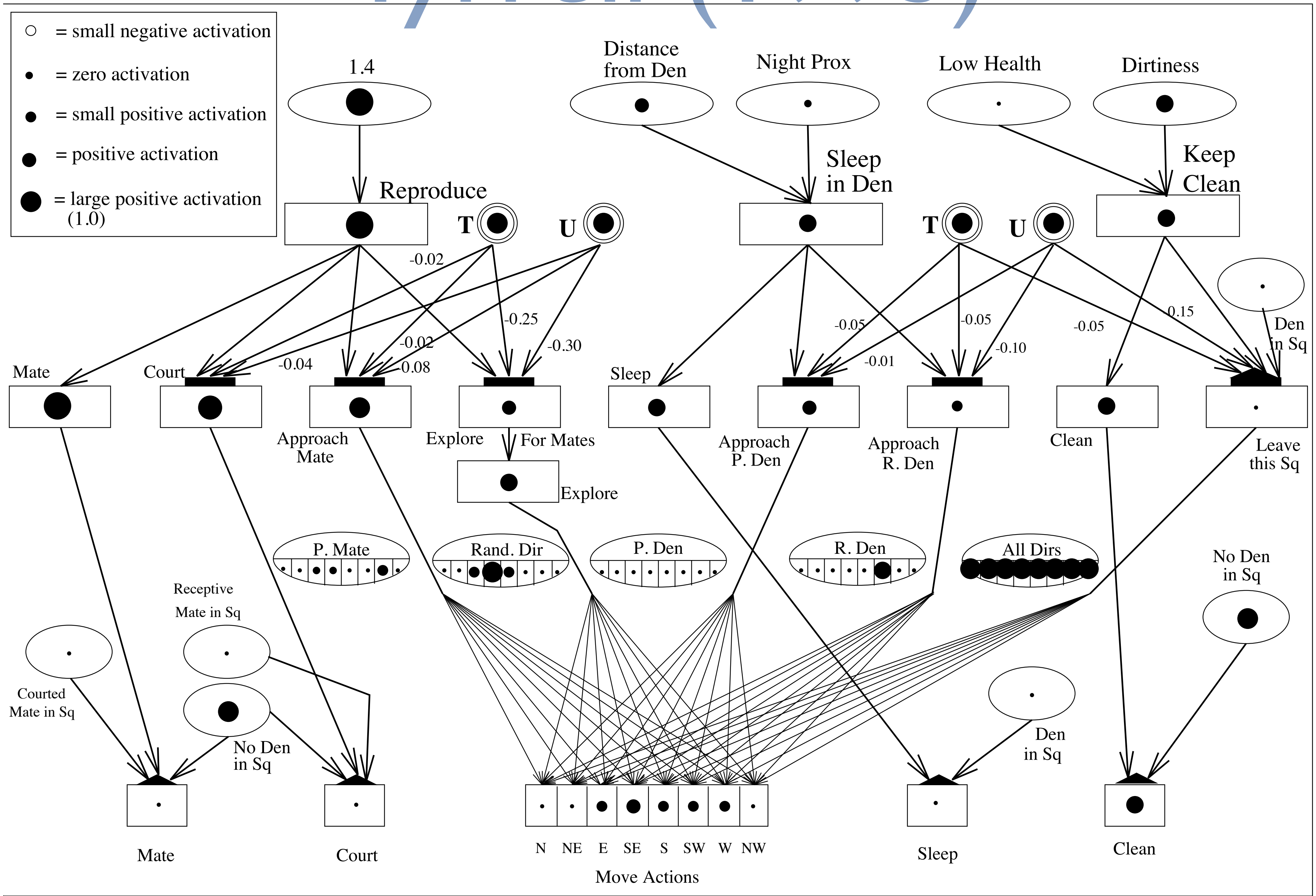
Spreading Activation Networks

- Sound good:
  - easy
  - brain-like (priming, action potential).
  - Still influential (Franklin & Baars 2010, Shanahan 2010).
- Can't do full action selection:
  - Don't scale; don't converge on comsumatory acts (Tyrrell 1993).

Tyrrell's Extended Rosenblatt & Payton Networks

- Consider all information & all possible actions at all times.
- Favour consumatory actions by system of weighting.
- Also weight uncertainty (e.g. of memory, temporal discounting).

Tyrrell (1993)



Extended Rosenblatt and Payton Free-Flow Hierarchy

Tyrrell's Analysis

- Compared all leading architectures.
- Discovered many weren't practical.
- Hoped to be "fair" by having parameters learned with a GA.
- Discovered this wasn't tractable.
- Went into oceonagraphy after PhD.

National Oceanography Centre, Southampton
UNIVERSITY OF SOUTHAMPTON AND NATURAL ENVIRONMENT RESEARCH COUNCIL

discovering the unknown

Contact us | Find us | For media | NOCSnet | Webmail | Disclaimer

People | Search:

- NOCS home
- About us
- Support us
- Recent News & archive
- Research
- Education
- Schools & public
- National Marine Facilities
- Commercial
- Data
- Library & Information
- Policy & coordination

Ocean Biogeochemistry and Ecosystems Staff

Dr Toby Tyrrell

Telephone
External: +44 (0) 23 80596110
Internal: 26110

Send email
Publications

Ocean Biogeoc Ecosyst
OBE ho
People
Resear
Publica
Faciliti
PhD. P
Project
Other N groups
Select

Specialism

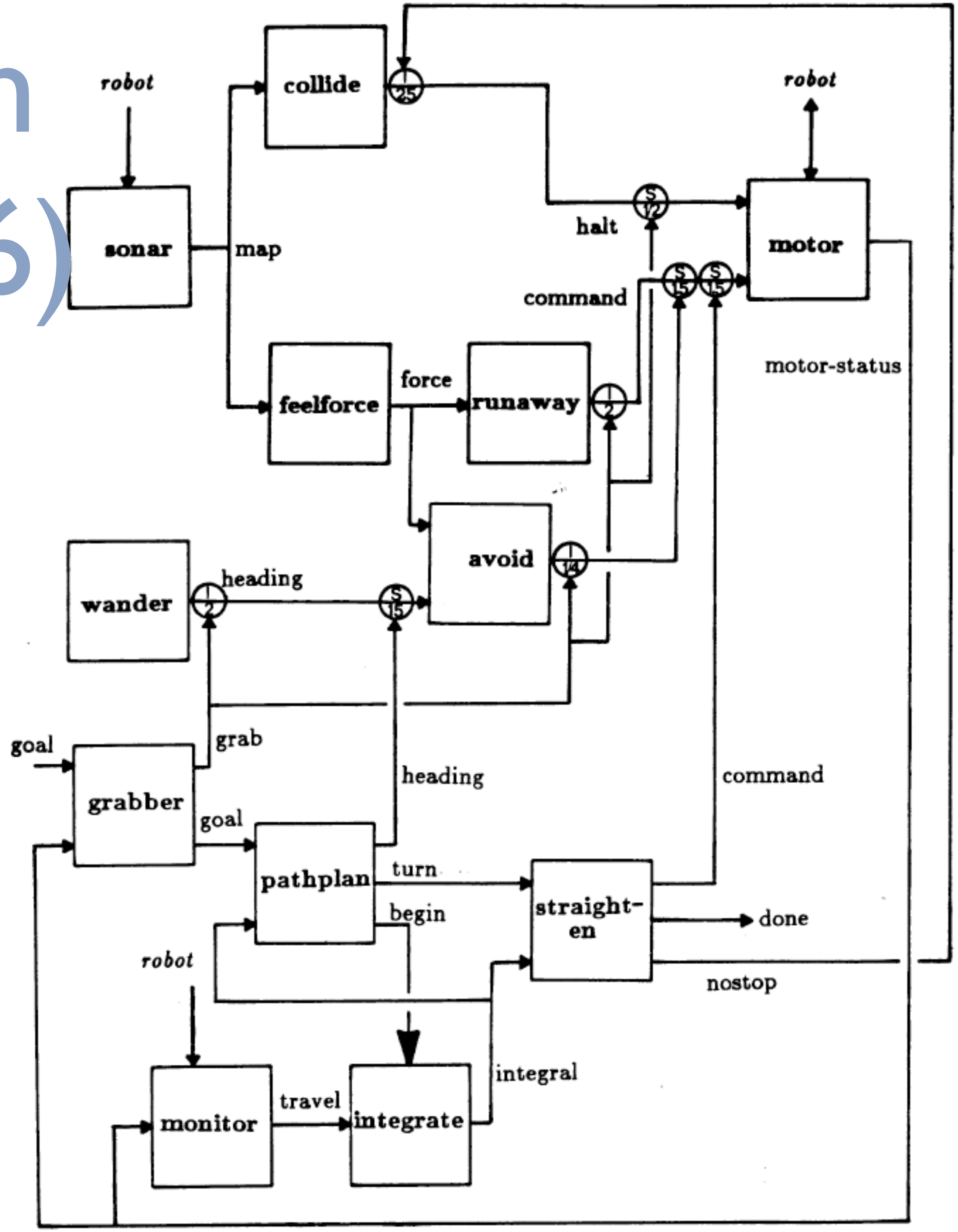
Ocean acidification

Research Interests

How organisms interact with their environments. Ecology of phytoplankton, coccolithophores in particular. Ocean acidification. Ocean biogeochemistry, including during extreme events in Earth' ancient past such as the E/O and K/T boundaries. Ocean carbon cycle and its effect on future atmospheric CO2 levels. Marine cycles of N, P, C, Si. The control of biogenic element concentrations in the sea as a function of ecological competition between different functional groups of phytoplankton. Modelling of all of the above.

Subsumption (Brooks 1986)

- Emphasis on sensing to action (via Augmented FSM).
- Very complicated, distributed arbitration.
- No learning.
- Worked.

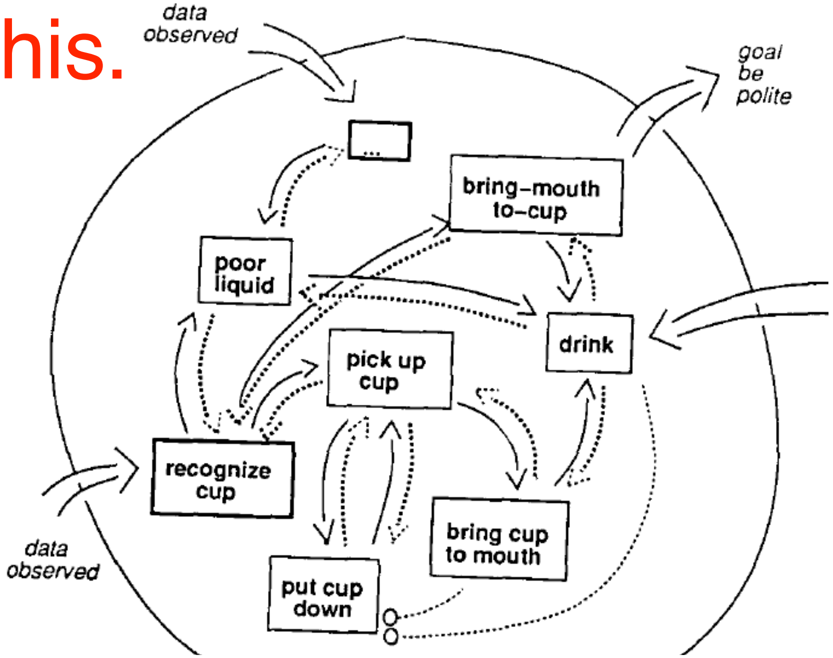


Architecture Lessons (Subsumption)

- Action from perception can provide the further structure – modules (behaviors).
  - Modules also support iterative development / continuous integration.
- Real time should be a core organising principle – start in the real world.
- Good ideas can carry bad ideas a long way (no learning, hard action selection).

Architecture Lesson?

- Goals ordering needs to be flexible.
- Maybe spreading activation is good for this.



A Robust Layered Control System for a Mobile Robot

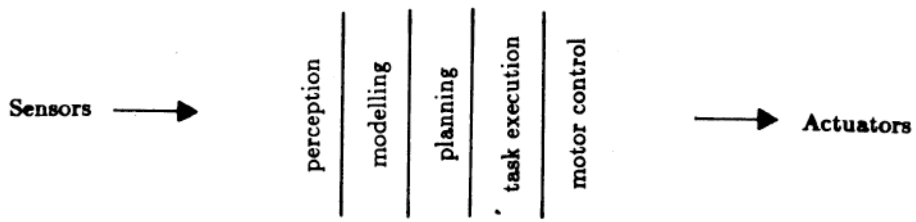
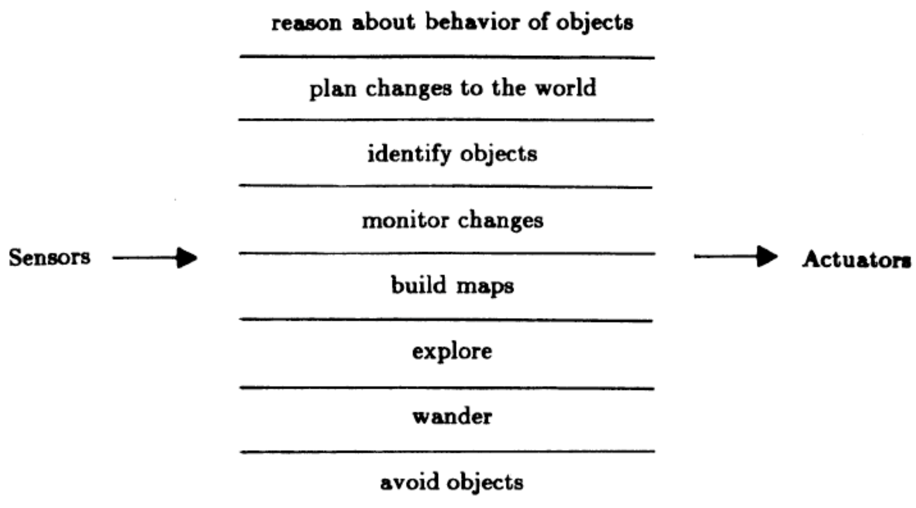
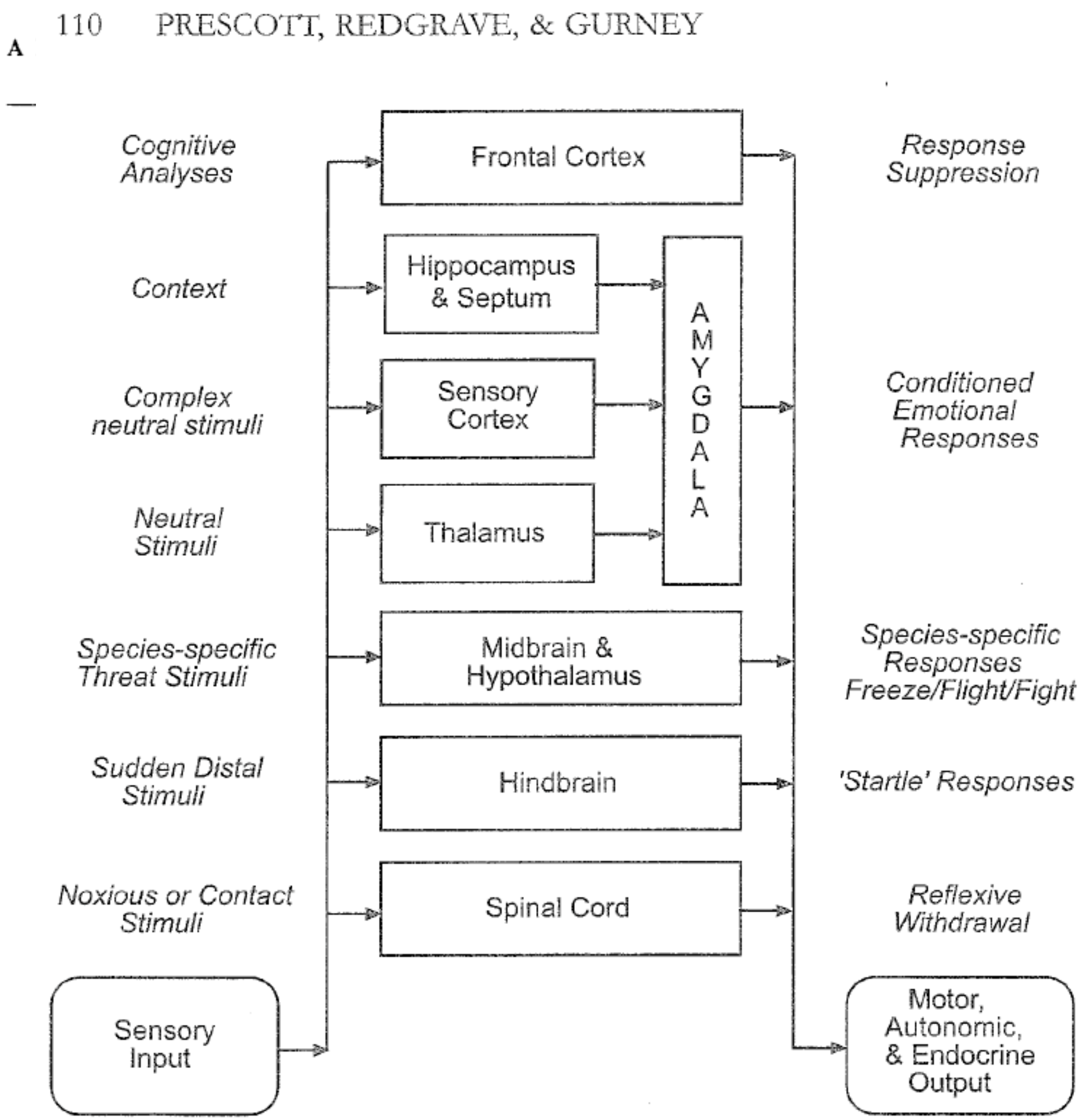
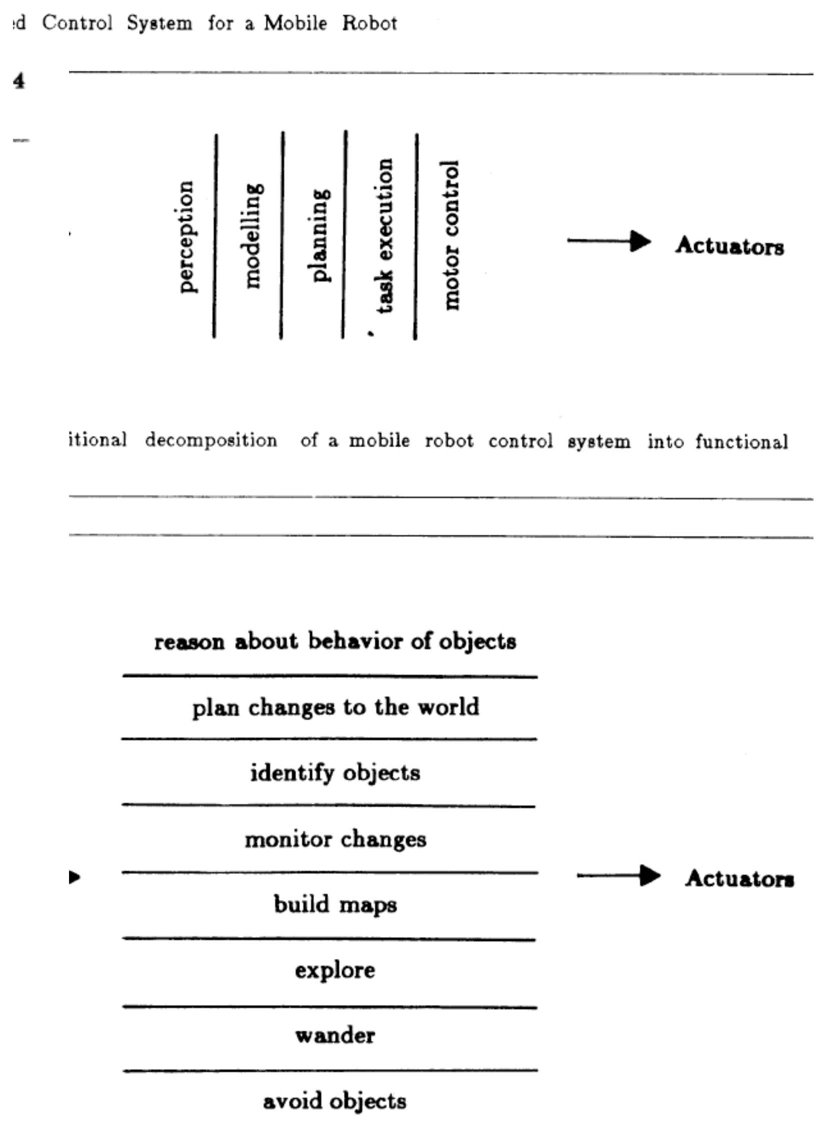


Figure 1. A traditional decomposition of a mobile robot control system into functional modules.



SA: Layers vs. Behaviours





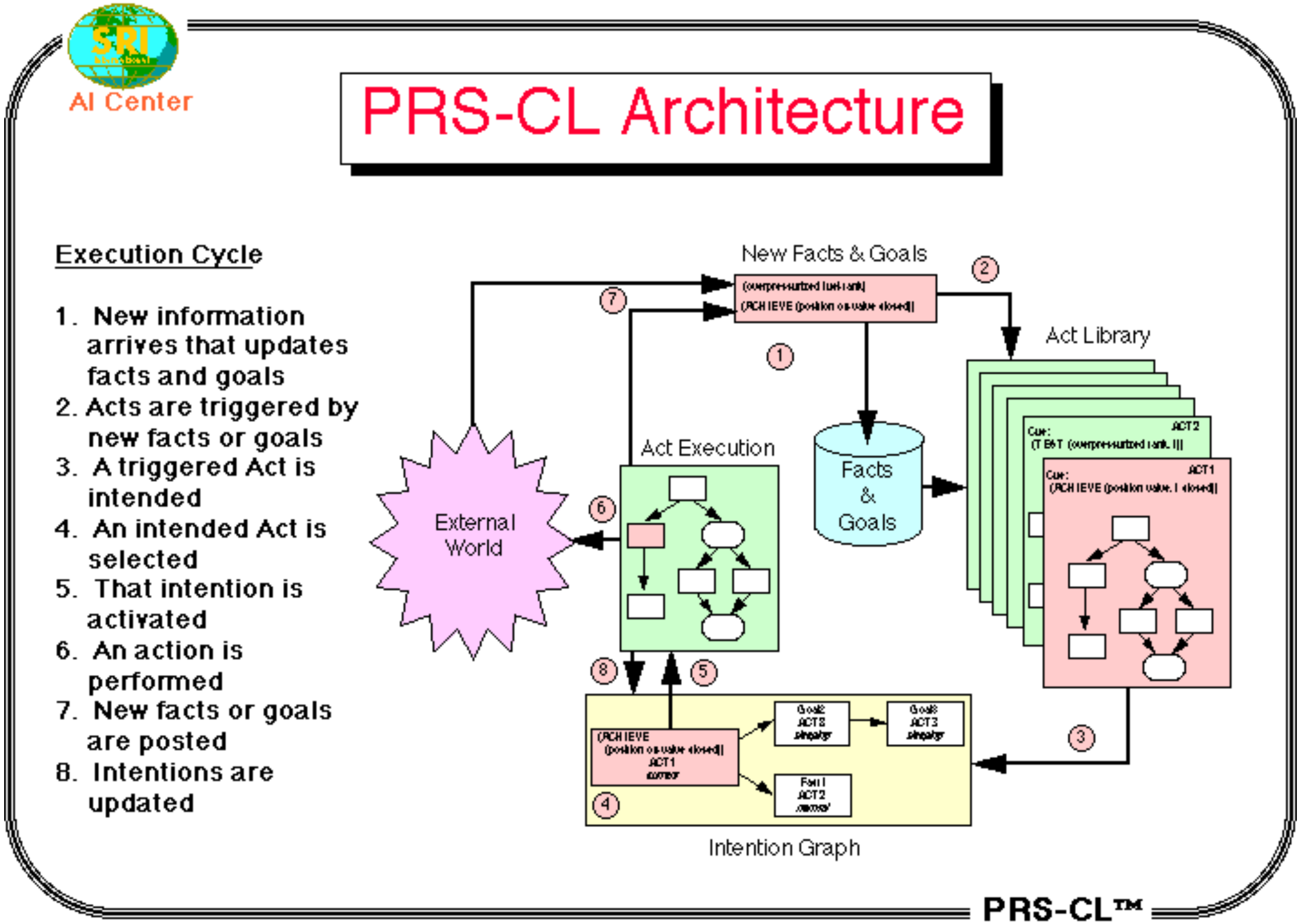
Layered *or* Hybrid Architectures

1. Incorporate behaviors/modules (action from sensing) as “smart” primitives.
2. Use hierarchical dynamic plans for behavior sequencing.
3. (Allegedly) some have automated planner to make plans for layer 2.

- Examples: Firby/RAPS/3T (‘97); PRS (1992-2000); Hexmoore ‘95; Gat ‘91-98

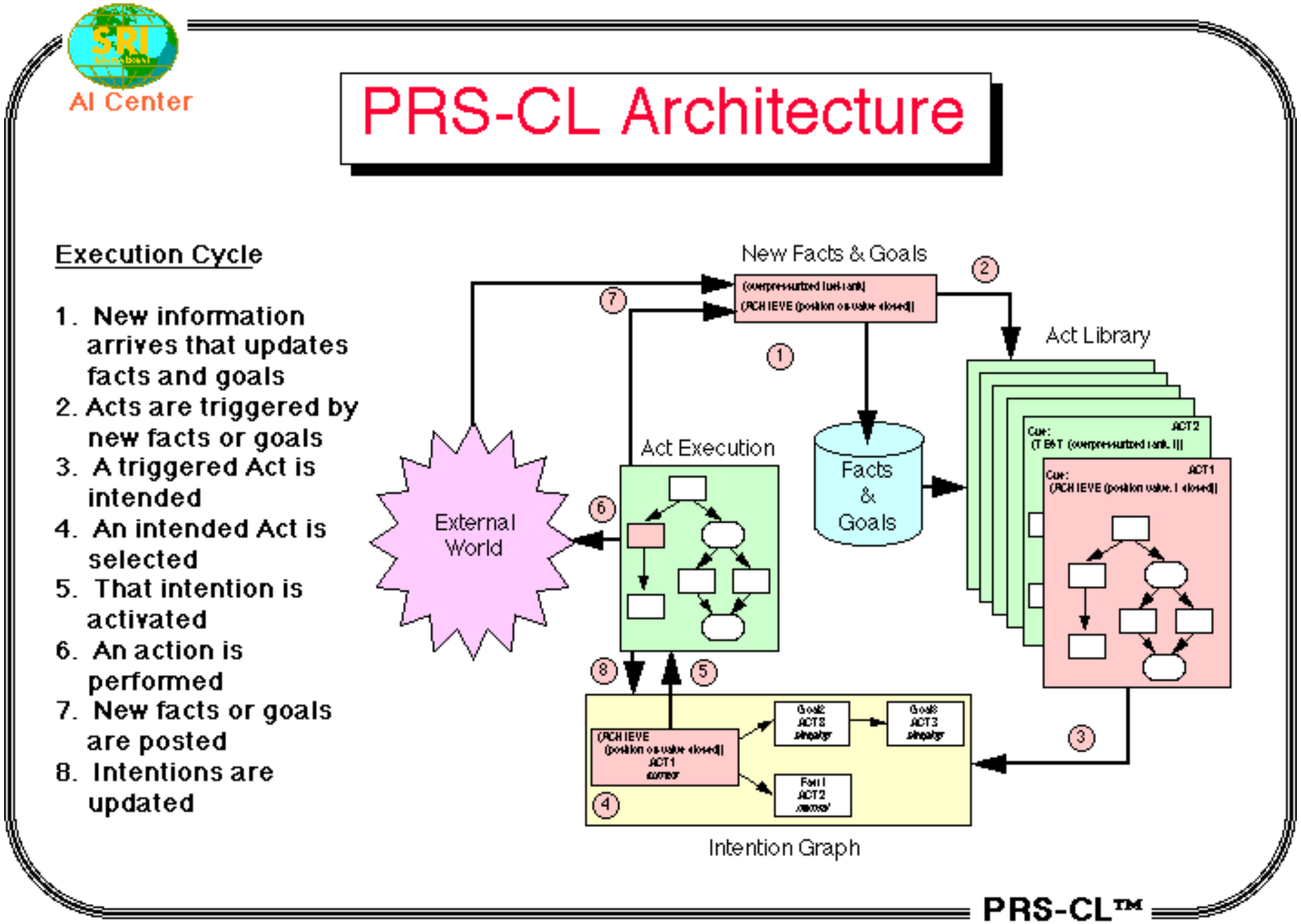
Belief, Desires, Intentions (BDI)

- *Beliefs*: Predicates
- *Desires*: goals & related dynamic plans
- *Intentions*: current goal



Procedural Reasoning System

- BDI
- And reactive (responds to emergencies by changing intentions.)
- Er... once or twice (Bryson ATAL 2000).



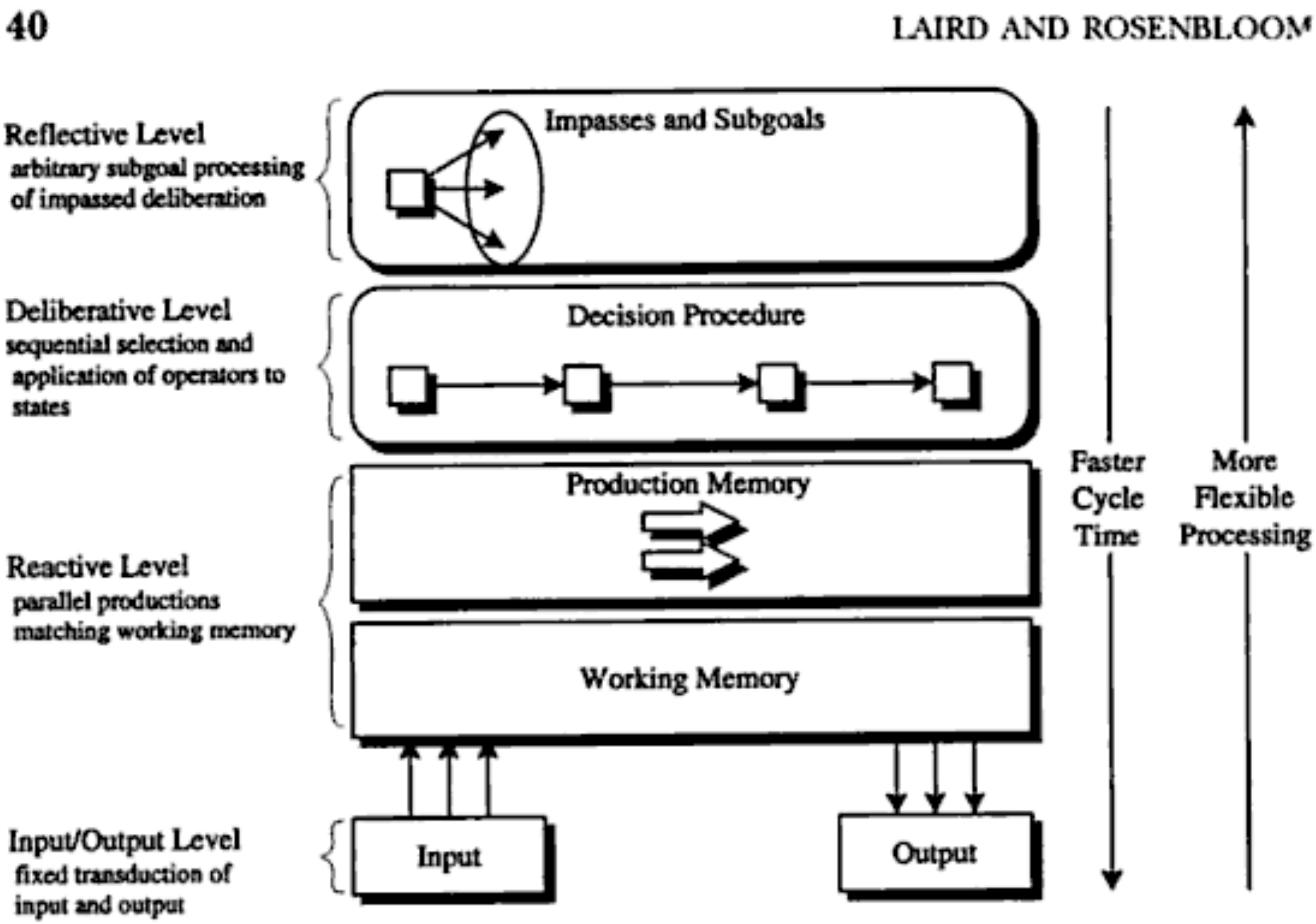
Architecture Lessons

- Structured dynamic plans make it easier to get your robot to do complicated stuff.
- Automated planning (or for Soar, chunking/learning) is seldom actually used.
- To facilitate that automated planning, modularity is often compromised.

(Bryson JETAI 2000)

Soar as a 3LA

J. Laird & P. Rosenbloom, “The Evolution of the Soar Cognitive Architecture”, *Mind Matters*, D. Steier and T. Mitchell eds., 1996.



Architecture Lessons

- Structured dynamic plans make it easier to get your robot to do complicated stuff.
- Automated planning (or for Soar, chunking/learning) is seldom actually used.
- Military turns chunking off because more productions slow down the system.
- “Teaching by brain surgery” / programming, not learning in real, installed systems.

CogAff

- Reflection on Top.
- Sense & Action separated!
- (Davis & Sloman 1995)

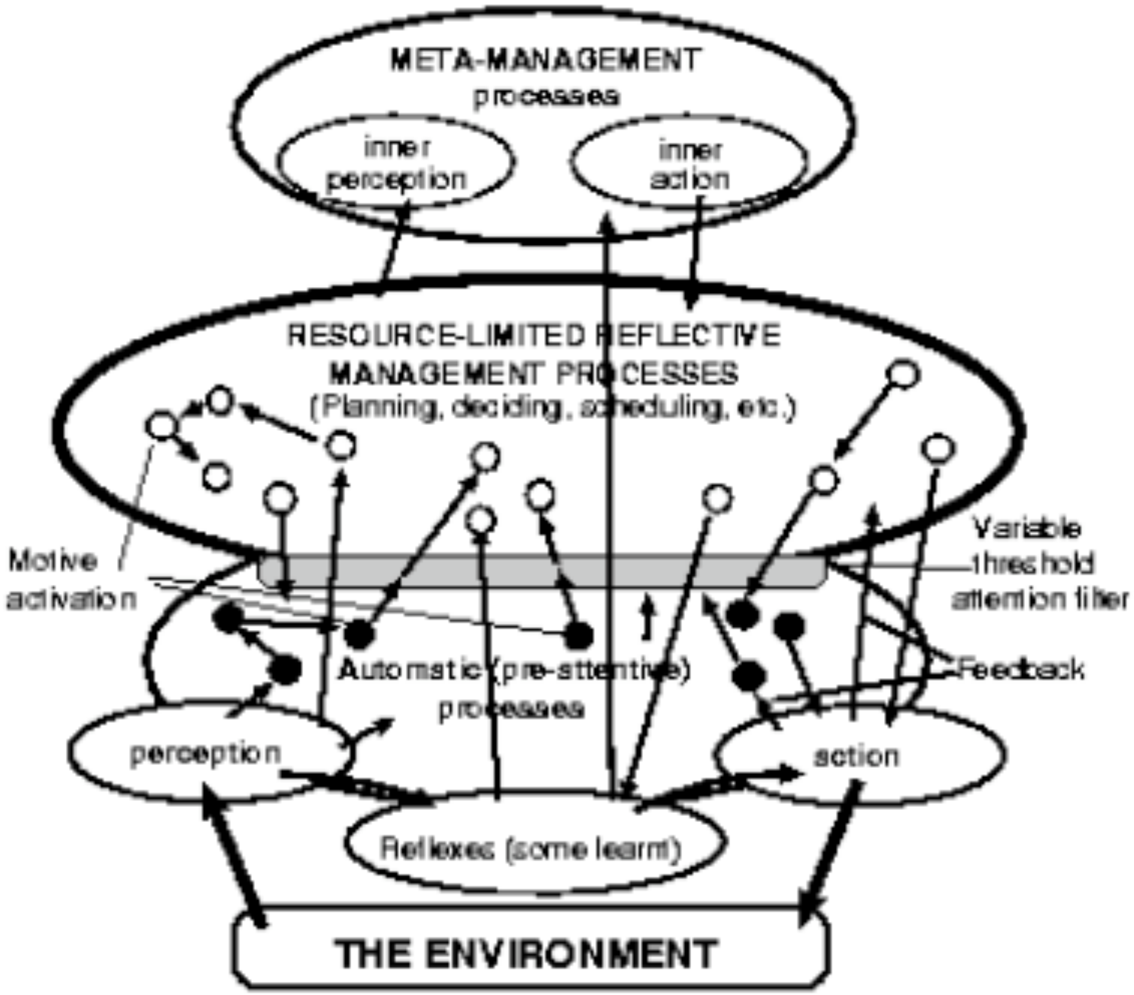
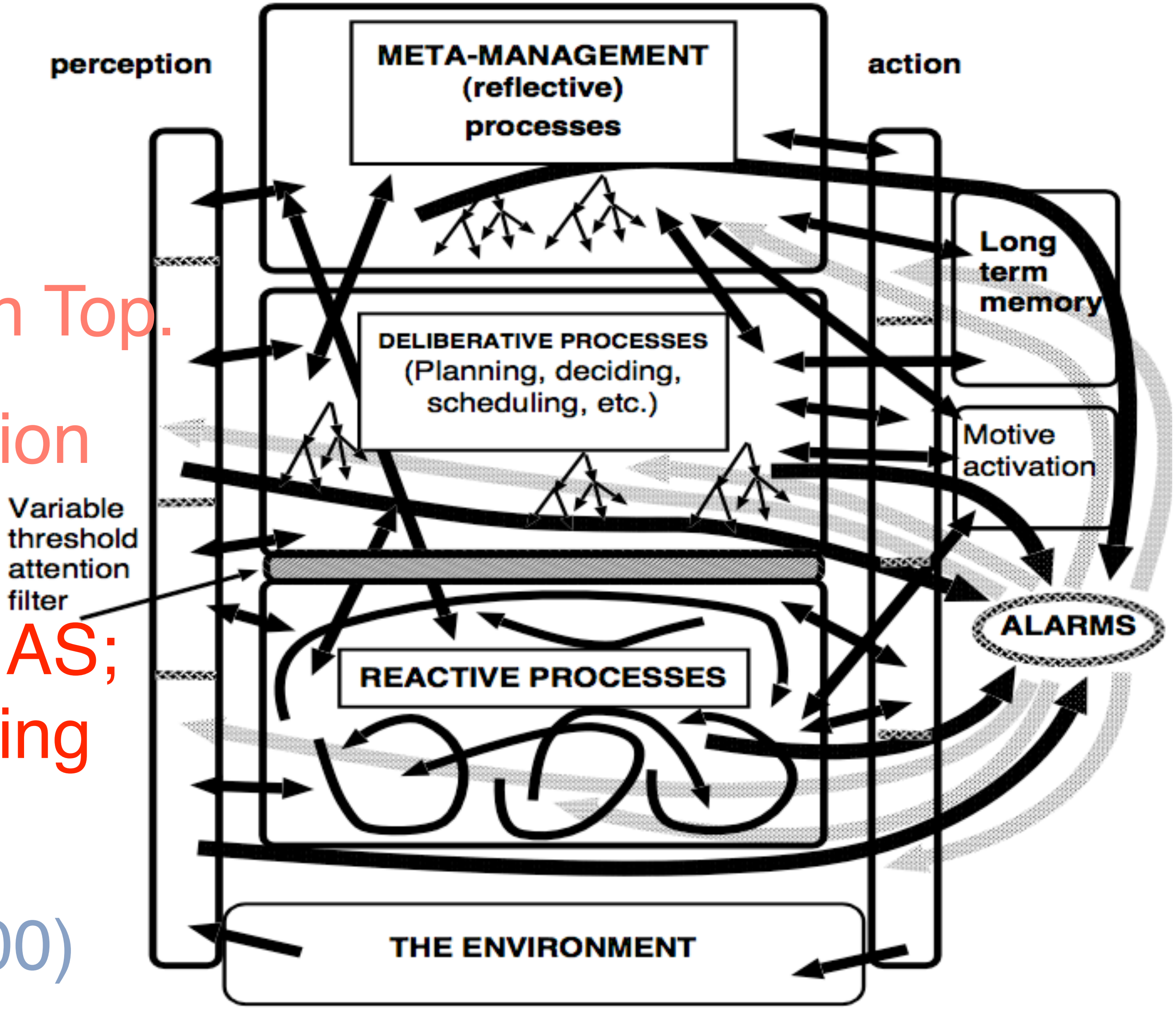


Figure 1 Towards an Intelligent Agent Architecture

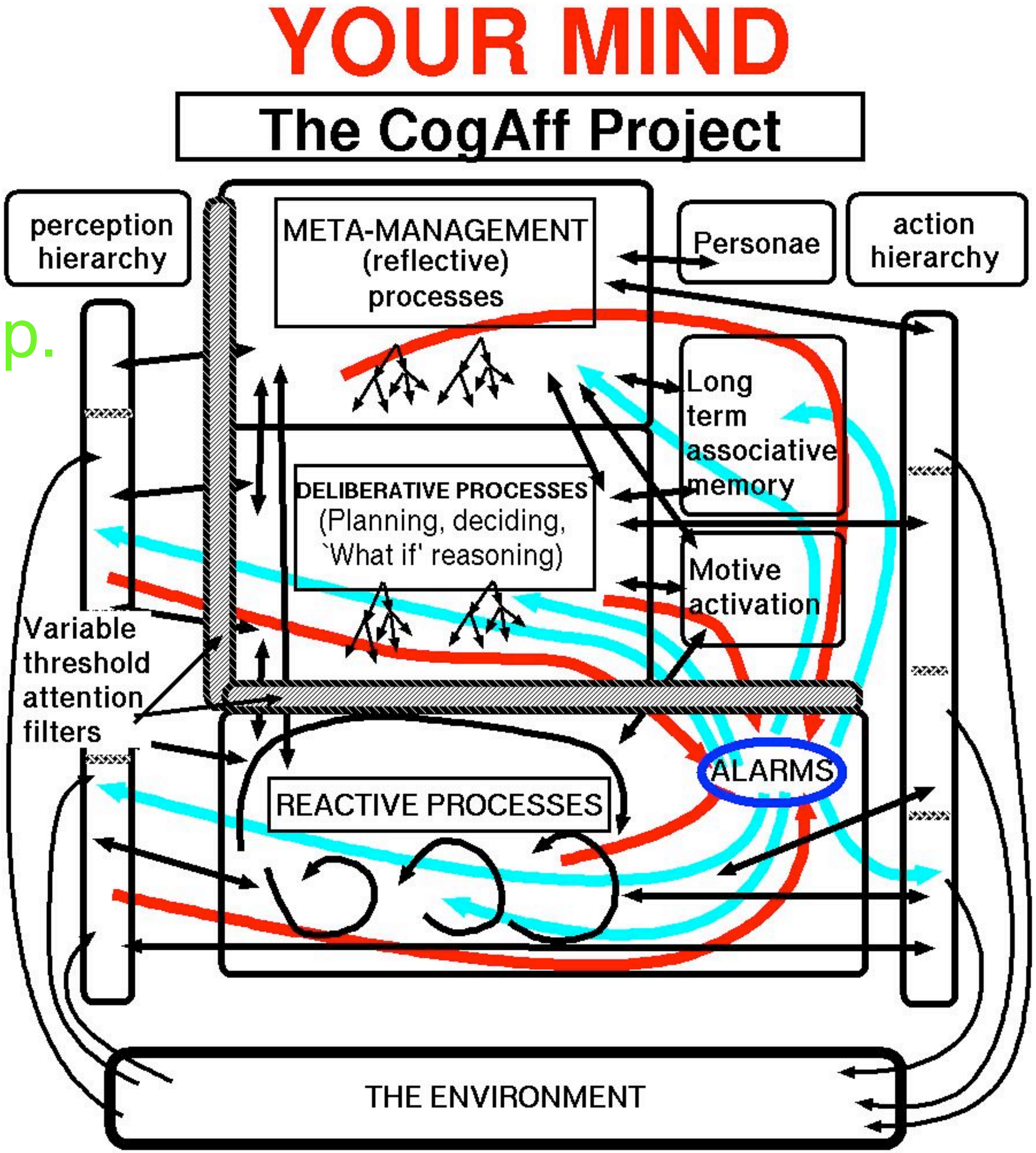
CogAff

- Reflection on Top.
- Sense & Action separated!
- Hierarchy in AS; Goal Swapping (Alarms).
- (Sloman 2000)

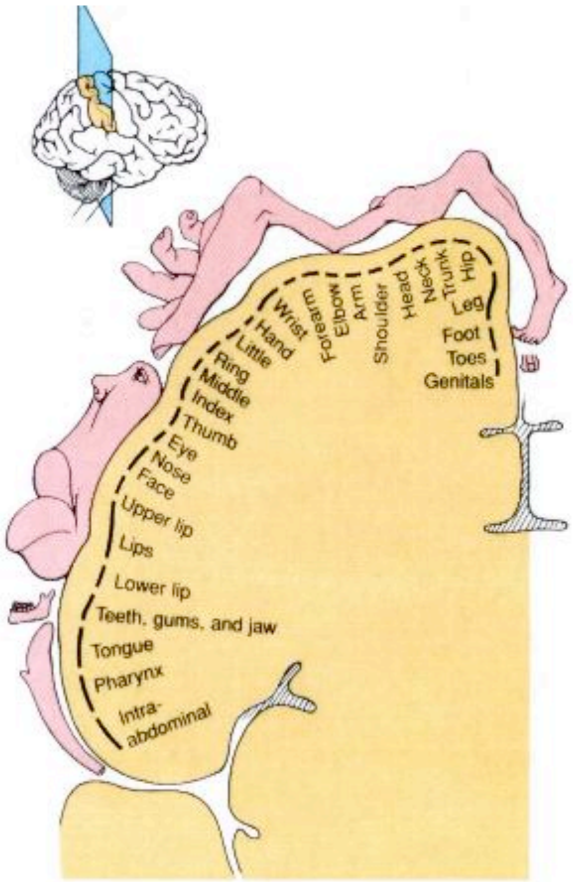


CogAff

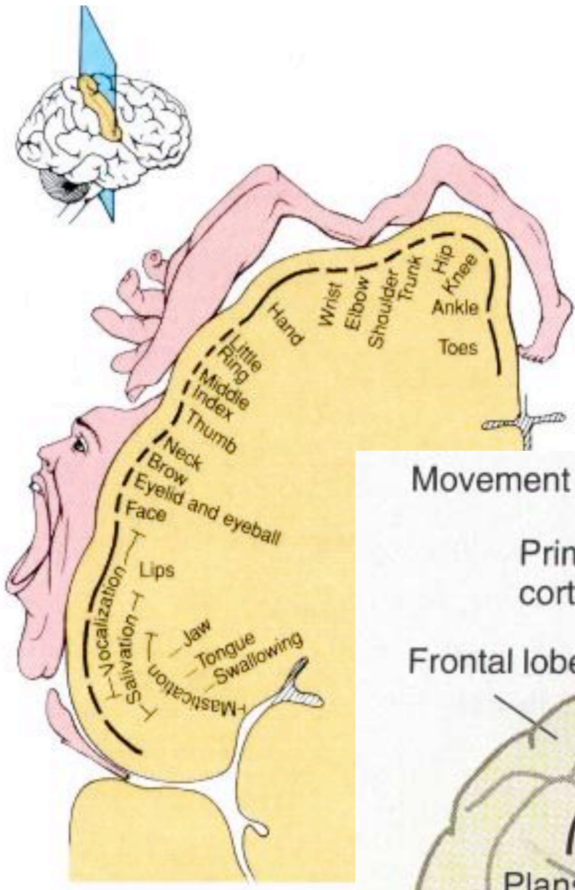
- Reflection on Top.
- Sense & Action separated!
- Hierarchy in AS, Goal Swapping (now reactive).
- Current Web



Separate Sense & Action

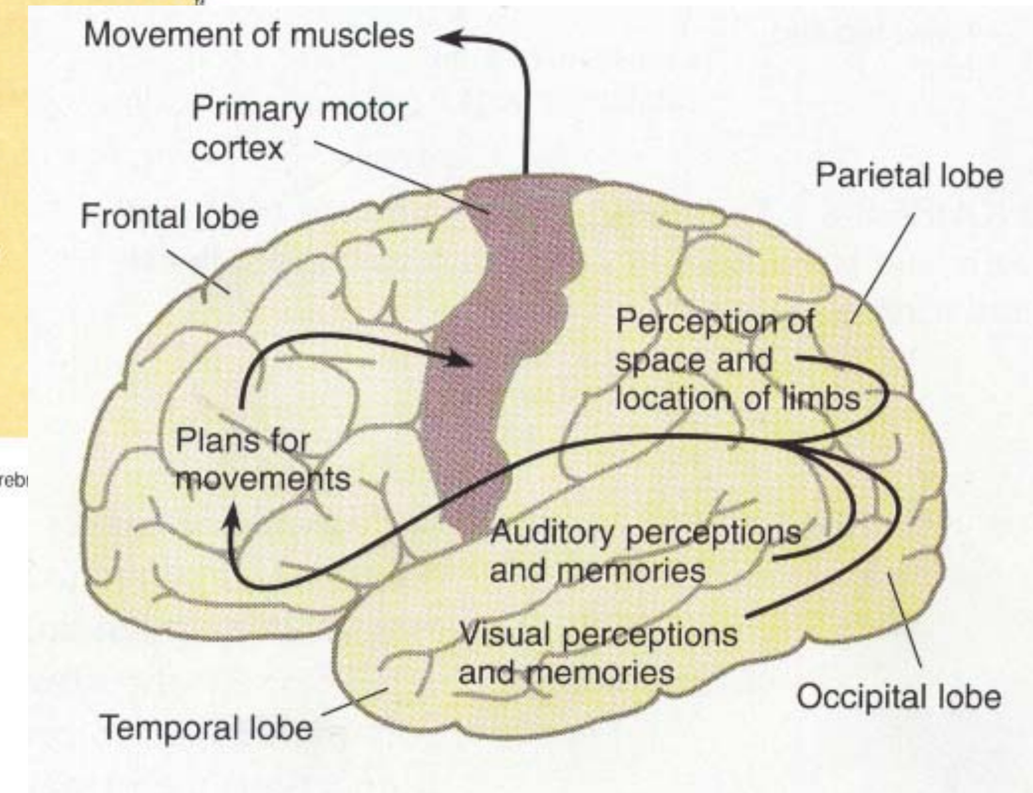


(a) Somatosensory cortex in right cerebral hemisphere



(b) Motor cortex in right cerebr

- Something we higher mammals do.
- Central Sulcus



Chance for Cognition?

(pictures from Carlson)

Architecture Lessons (CogAff)

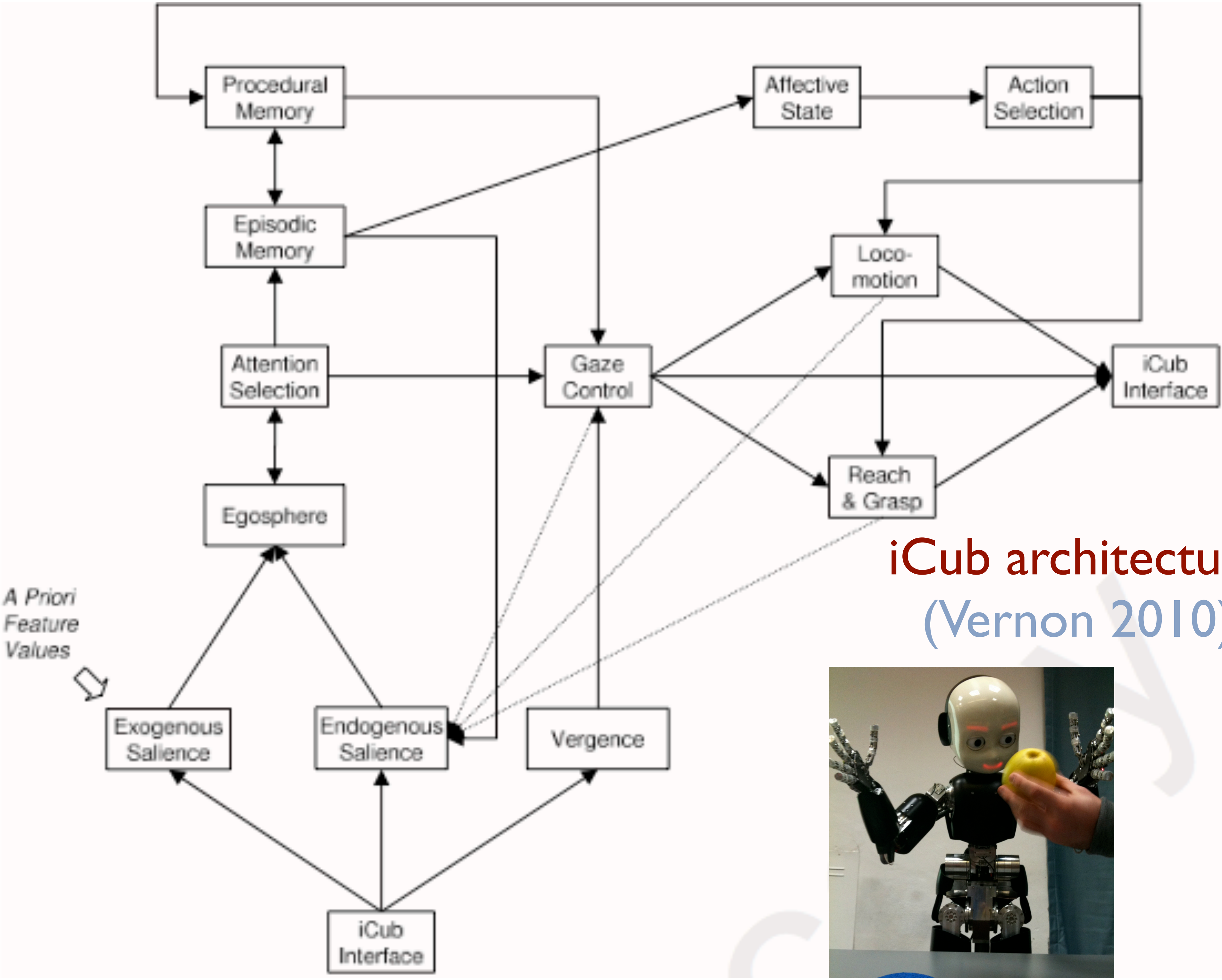
- Maybe you don't really want productions as your basic representation – you may want to come between a sense and an act sometimes.
- Your architecture looks very different if you really worry about adult human linguistic / literature-level behaviour rather than just making something work.

Contemporary Architectures?

- Currently people talk more about an architecture for a system, not an "architecture" meaning a generic development methodology + ontology.
- But the topic may come back again.
- And the ontologies and histories are still useful.

iCub architecture

(Vernon 2010)



Contemporary Architectures?

- Currently people talk more about an architecture for a system, not an "architecture" meaning a generic development methodology + ontology.
- But the topic may come back again.
- And the ontologies and histories are still useful.

Summary

- Architectures assume an ontology of what intelligence needs, and a development methodology.
- Architectures describe how the necessary parts should be connected.
- Cognitive architectures are often identified with working code – action selection systems.