# **Modularity** in **A**rtificial and Natural **I**ntelligence

Joanna J. Bryson

Artificial Intelligence Group, Bath Computer Science

`jjb@cs.bath.ac.uk`

`http://www.cs.bath.ac.uk/∼jjb`

## Who am I and What have I Done?

| Education | Work |
|---|---|
| B.A. Behavioral Sci. | System Analyst, Financial Industry |
| M.Sc. AI | Object-Oriented Reengineering |
| M.Phil Psychology | Research Scientist, AI for VR (LEGO) |
| Ph.D. Computer Sci. | |

Dialog Tutoring Systems, Characters, Modelling Primates

## My Chunk of Artificial Intelligence

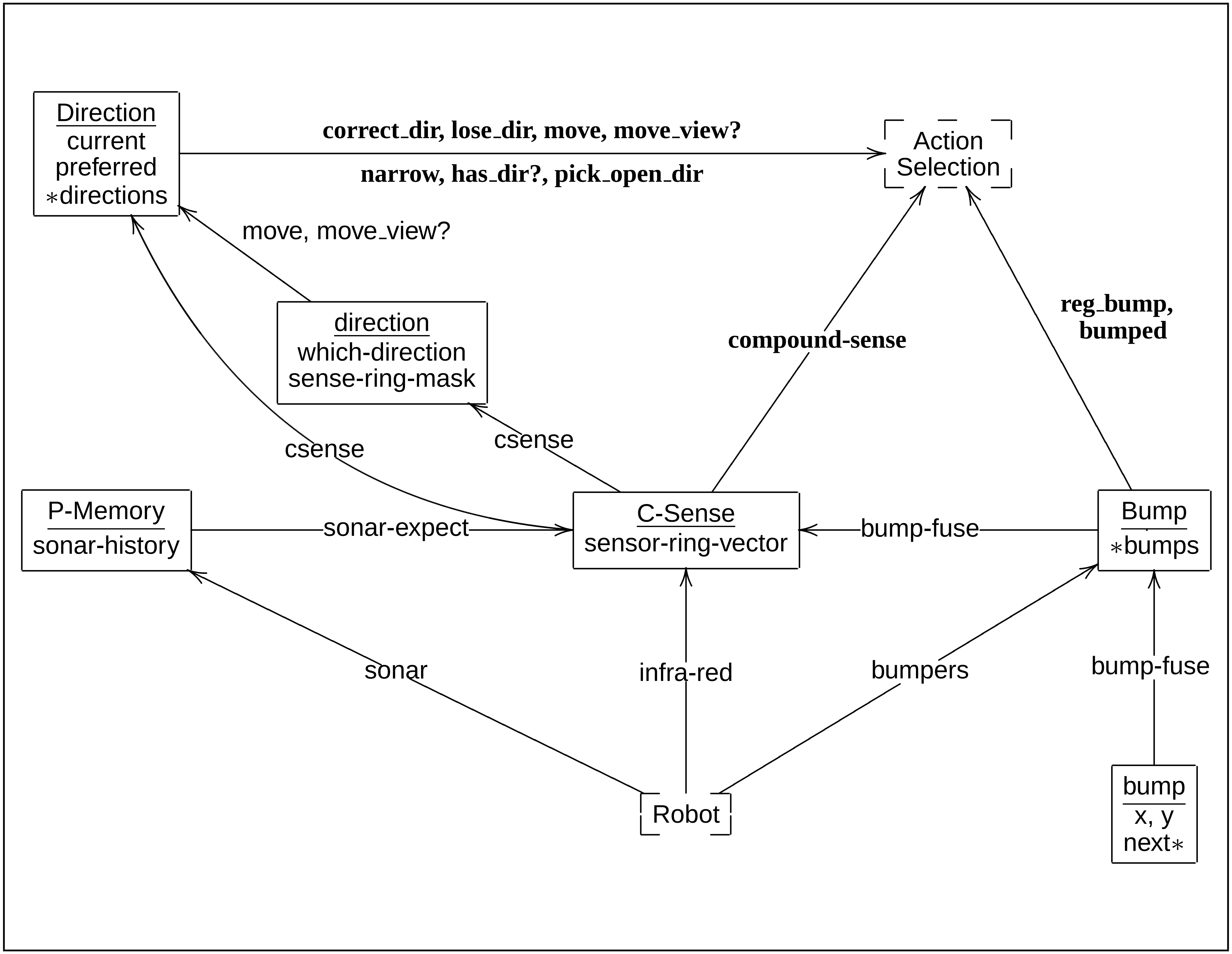
- Designing Intelligent Systems.
  - Modules to encapsulate learning, action & perception.
  - Reactive Plans to arbitrate between modules.
- Understanding Natural Intelligence.
  - Animals moving in space, integrating information & goals.
  - Individuals learning new tasks.
  - Behavior and structure of societies emerging from individual intelligence.

## Types of Projects

- Building AI systems with existing tools.
- Running experiments on existing AI systems.
- Making AI tools easier to use.

## Example: A Mobile Robot

(Bryson ATAL97)


Direction
current
preferred
∗directions
correct_dir, lose_dir, move, move_view?
narrow, has_dir?, pick_open_dir
Action
Selection
move, move_view?
direction
which-direction
sense-ring-mask
compound-sense
reg_bump,
bumped
csense
csense
P-Memory
sonar-history
sonar-expect
C-Sense
sensor-ring-vector
bump-fuse
Bump
∗bumps
sonar
infra-red
bumpers
bump-fuse
Robot
bump
x, y
next∗

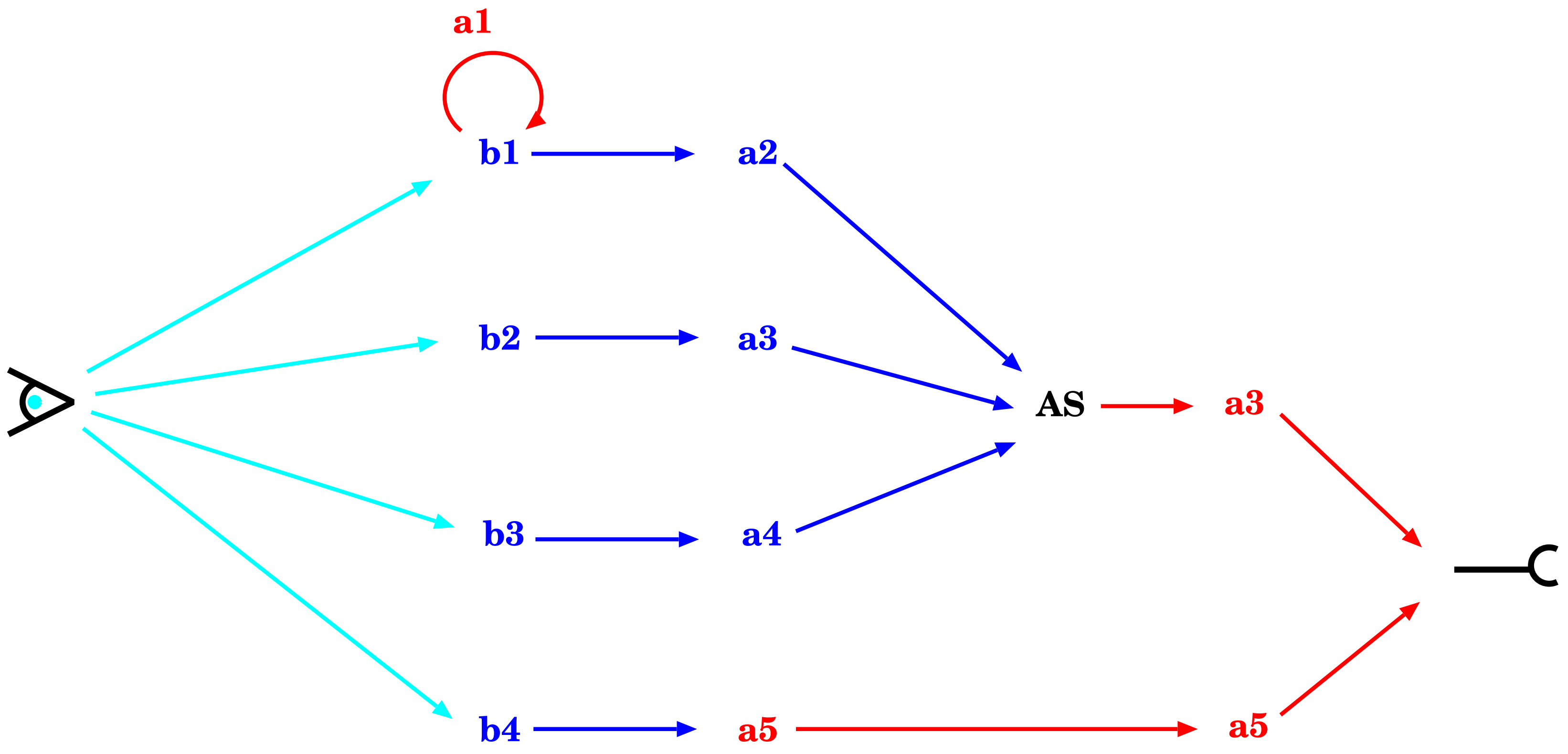
## Example: VR

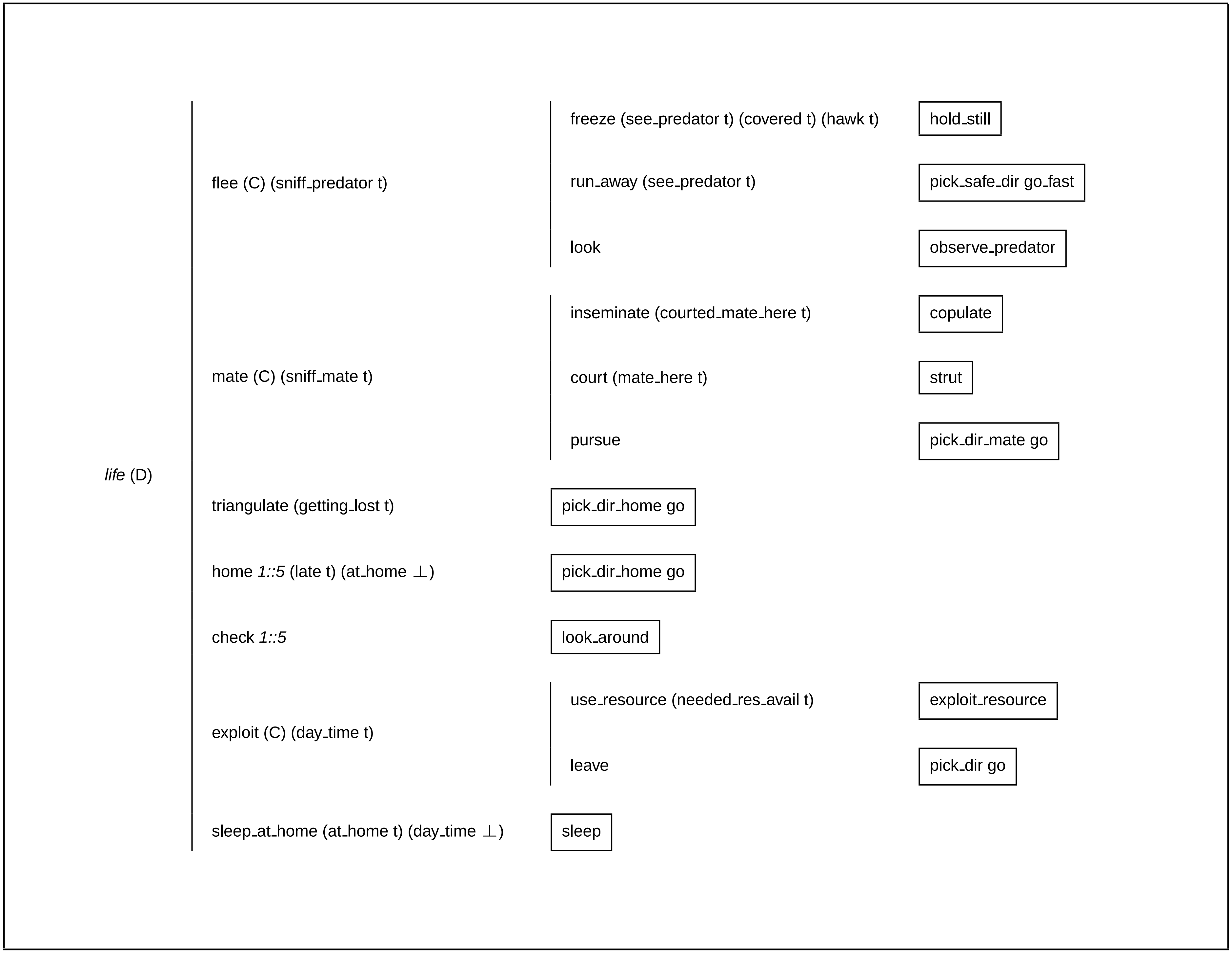
(Bryson & Thórisson 2000)

## Reactive Planning

- Modularity leads to coordination problems.
- Reactive plans are engineered solutions.
  - Planning
  - Reactive Planning
  - Reactive Plans
- Plans (and memory) are what create the individual personality of agents.

# The Place of Reactive Planning in an Agent




*life* (D)
flee (C) (sniff_predator t)
freeze (see_predator t) (covered t) (hawk t)
hold_still
run_away (see_predator t)
pick_safe_dir go_fast
look
observe_predator
mate (C) (sniff_mate t)
inseminate (courted_mate_here t)
copulate
court (mate_here t)
strut
pursue
pick_dir_mate go
triangulate (getting_lost t)
pick_dir_home go
home *1::5* (late t) (at_home ⊥)
pick_dir_home go
check *1::5*
look_around
exploit (C) (day_time t)
use_resource (needed_res_avail t)
exploit_resource
leave
pick_dir go
sleep_at_home (at_home t) (day_time ⊥)
sleep

## Project 1: Make Writing Plans Easier

1. Write a GUI application.
2. Not really AI in itself, but get experience with AI systems.
3. If it works, will be used by *lots* of people.
4. Mostly about programming, some HCI.

## Project 2: Rewrite my Action Selection mechanism in Python

1. Write a lot of good code, preferably with documentation.
2. Platform independent, connect to AI middleware system (work with other engineers), GUI.
3. If it works, will be used by *lots* of people. Immediately for VR, maybe on a robot within a year or two.
4. Again, mostly programming, good credential for industry or academia.

## Macaque Social Order

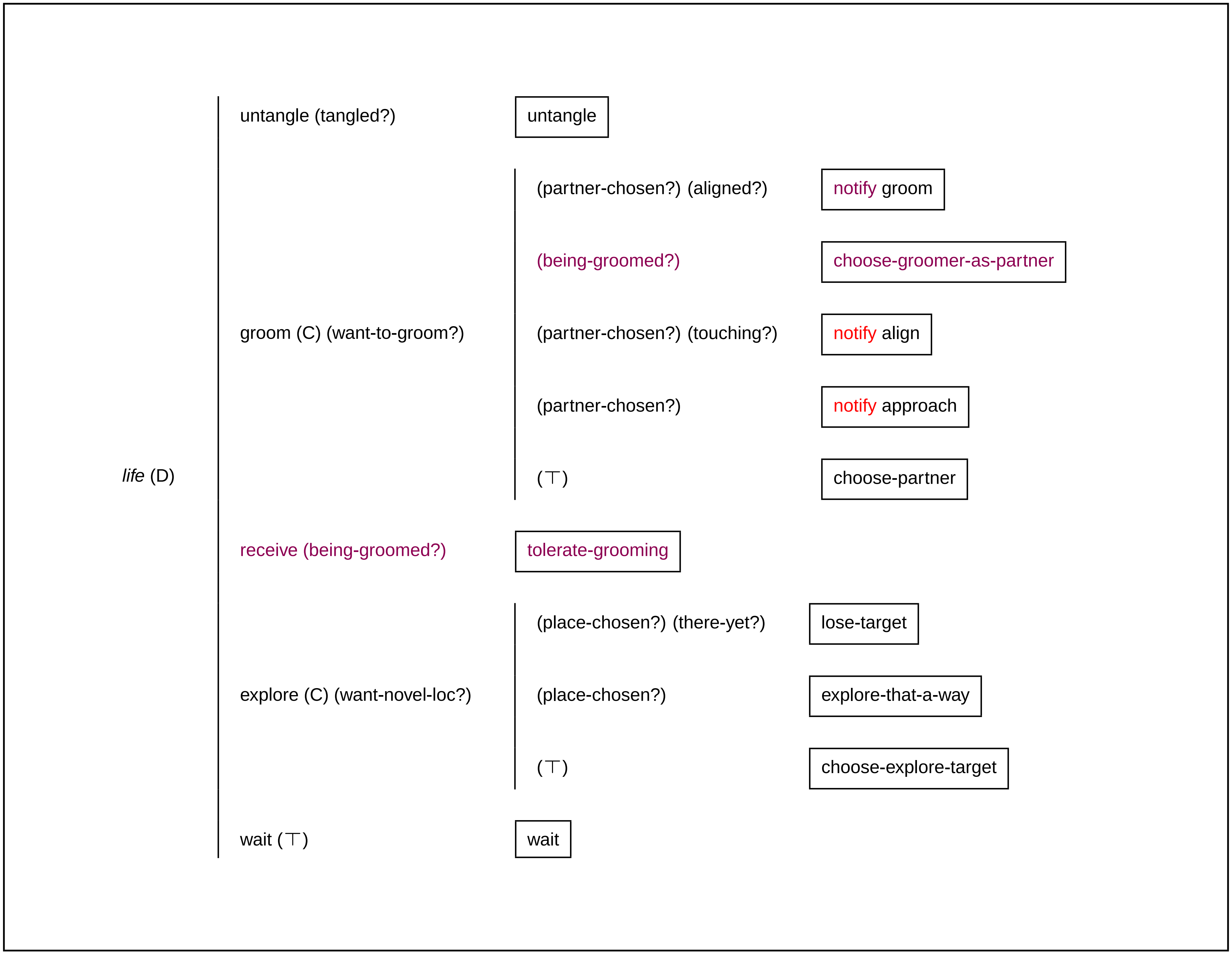
- Some (e.g. Rhesus) show strict, hierarchical order, also violent but infrequent conflict.
- Some (e.g. Stumptail) show egalitarian social order, more frequent but less violent conflict.

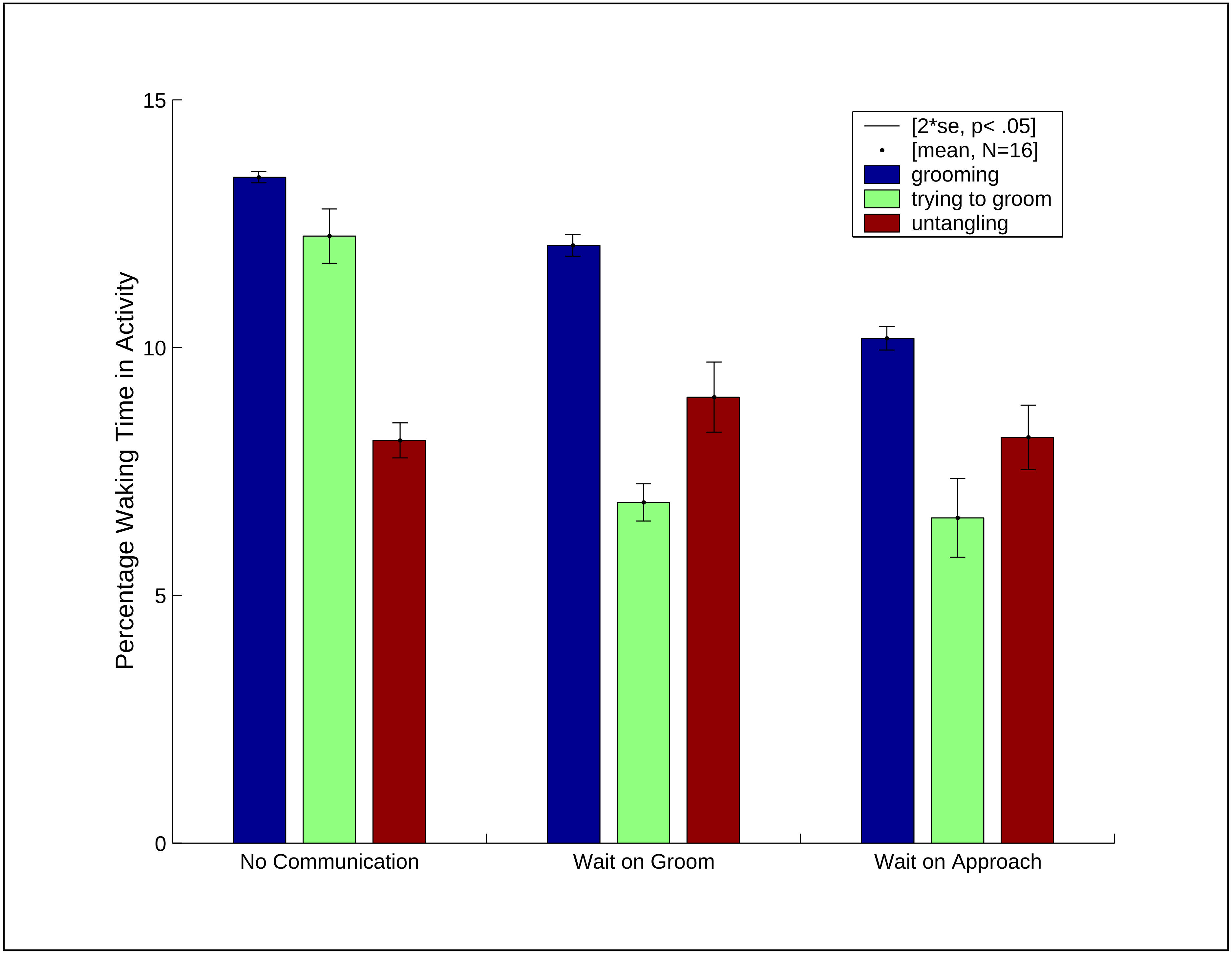
## Hypotheses of Macaque Social Order

- Less resources (e.g. food) $\Rightarrow$ more violence $\Rightarrow$ selective pressure for social structure (Hemelrijk 2001, 2002)
- New conflict resolution behavior $\Rightarrow$ less violence $\Rightarrow$ less pressure for social structure (de Waal 2001, Flack *in prep.*)

## Basic Social Behaviors

| | **Navigate** | **Groom** | **Explore** |
|---|---|---|---|
| *state* | x, y, size, name<br>focus-of-attn | drive-level, partner<br>groomed-when,<br>being-groomed? | drive-level<br>direction-of-interest |
| *actions* | approach<br>wait, align<br>untangle | groom, choose-partner<br>partner-chosen?<br>tolerate, notify | choose-new-location<br>lose-target, explore<br>want-novel-loc? |


life (D)
untangle (tangled?)
untangle
groom (C) (want-to-groom?)
(partner-chosen?) (aligned?)
notify groom
(being-groomed?)
choose-groomer-as-partner
(partner-chosen?) (touching?)
notify align
(partner-chosen?)
notify approach
(⊤)
choose-partner
receive (being-groomed?)
tolerate-grooming
explore (C) (want-novel-loc?)
(place-chosen?) (there-yet?)
lose-target
(place-chosen?)
explore-that-a-way
(⊤)
choose-explore-target
wait (⊤)
wait



15
10
5
0
Percentage Waking Time in Activity
[2*se, p< .05]
[mean, N=16]
grooming
trying to groom
untangling
No Communication
Wait on Groom
Wait on Approach

## Project 3: Reimplement Hemelrijk's Work in Repast or Netlogo

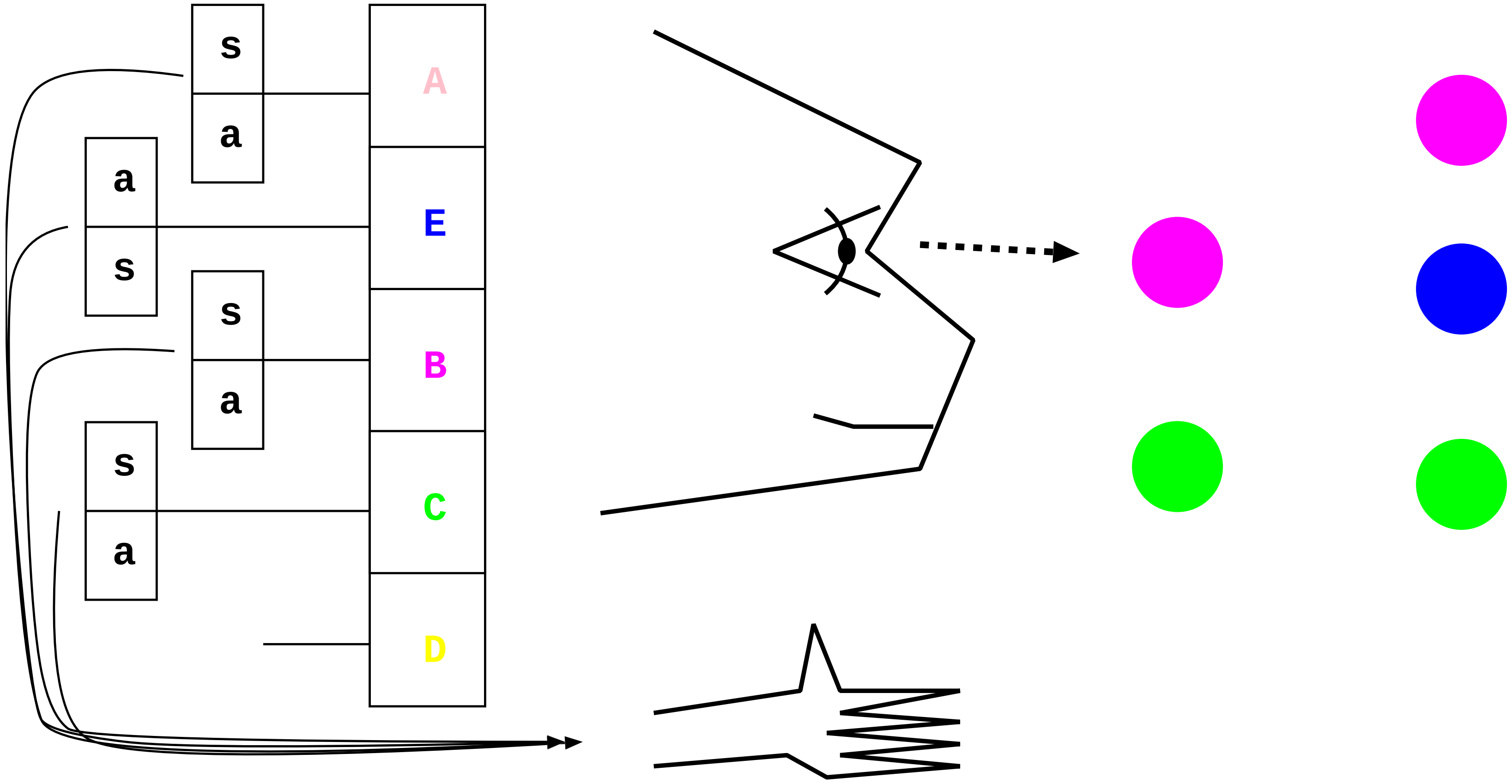
1. Use an existing AI platform, some coding in Java (probably).
2. Reading papers, running experiments, compiling results.
3. If it works, would probably get papers published, code would go on repositories, future students would use it.
4. Useful for getting into a science career, PhD.

## Project 4: Connect to Decent VR

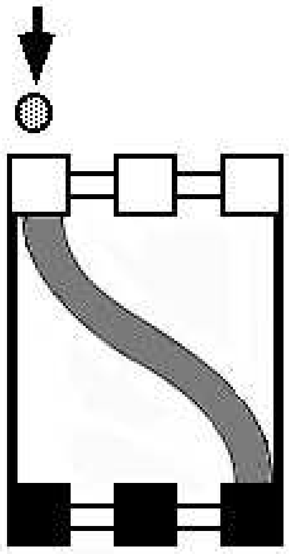
1. Find opensource game engines (I have a list).
2. Evaluate for usability, reliability.
3. Connect at least one to at least one to the AI system (I’ll do the lisp.)
4. Will learn about animation, AI in games.
5. Writeup will be an evaluation — good for industry jobs.

## Harris & McGonigle's (1994) Squirrel Monkeys

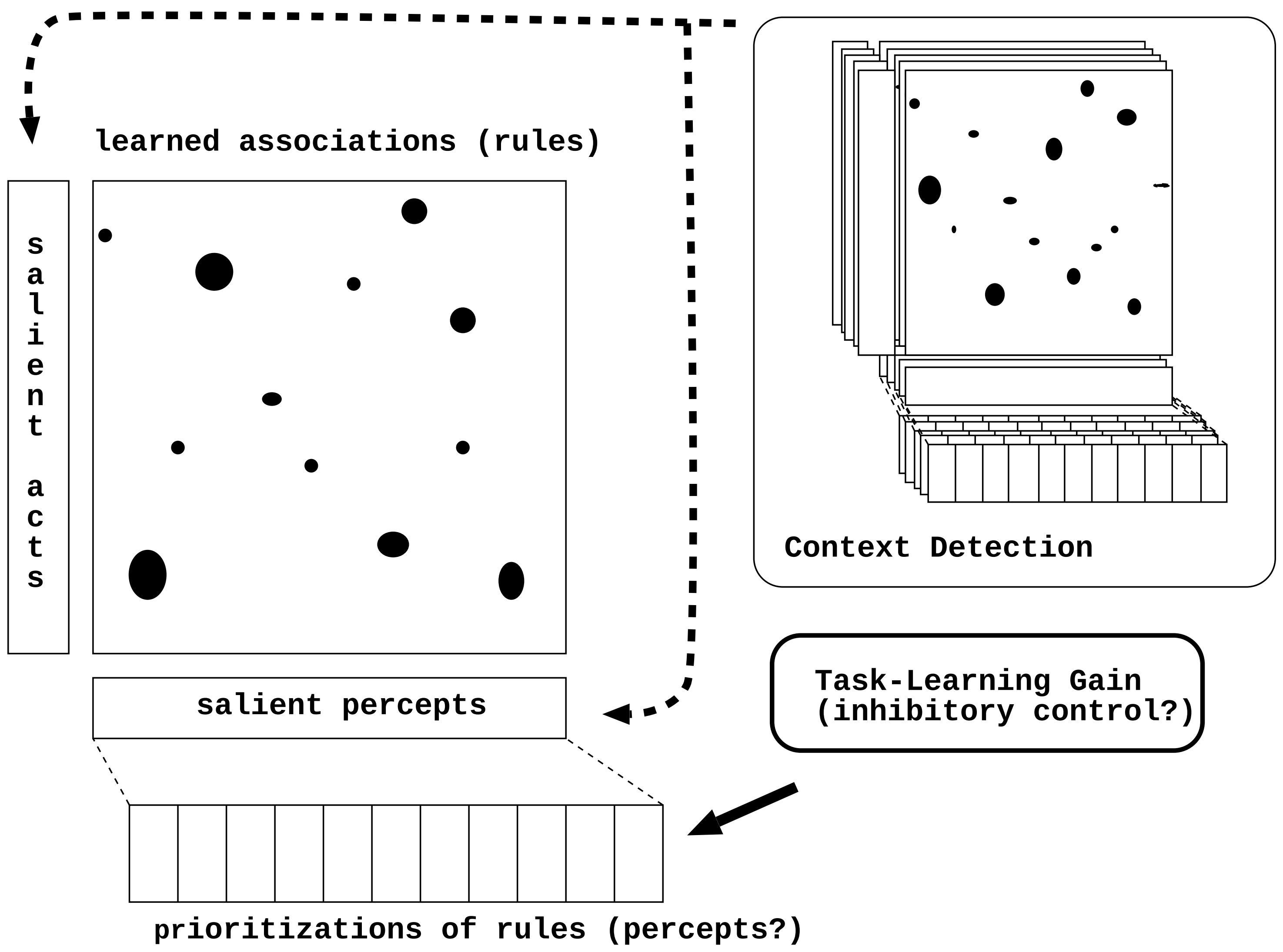
# Two-Tiered Transitive Inference Model



## Hauser’s Cotton-Top Tamarins



# Two-Tiered Task-Learning Model



## Project 5 (unlikely): Program New Task-Learning Model

1. Must be in Lisp, *possibly* ACT-R.
2. Takes serious coding, running experiments, reading literature, understanding representations.

## Project 6 (unlikely): Program Dialog Agent from Web Email Archive

1. Again, lots of coding, but can use any language.
2. Requires downloading lots of computational linguistics tools, getting a system working, integrating with AI.
3. Probably really a PhD, but let me know if you want to try to do a piece.

## Project $N$: Propose Something Along these Lines

1. All my students will be expected to attend group meetings, and to contribute to AI projects in some way.
2. Most (but not all) projects involve working with existing code — just like the real world.
3. AI is really cool & you can tell your friends about it.

Joanna Bryson

jjb@cs.bath.ac.uk
http://www.cs.bath.ac.uk/∼jjb

UNIVERSITY OF BATH