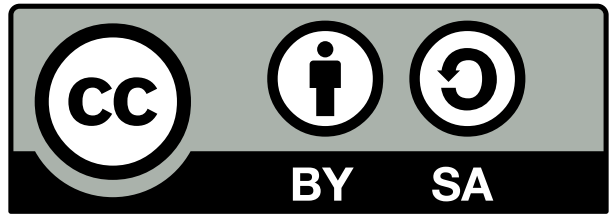
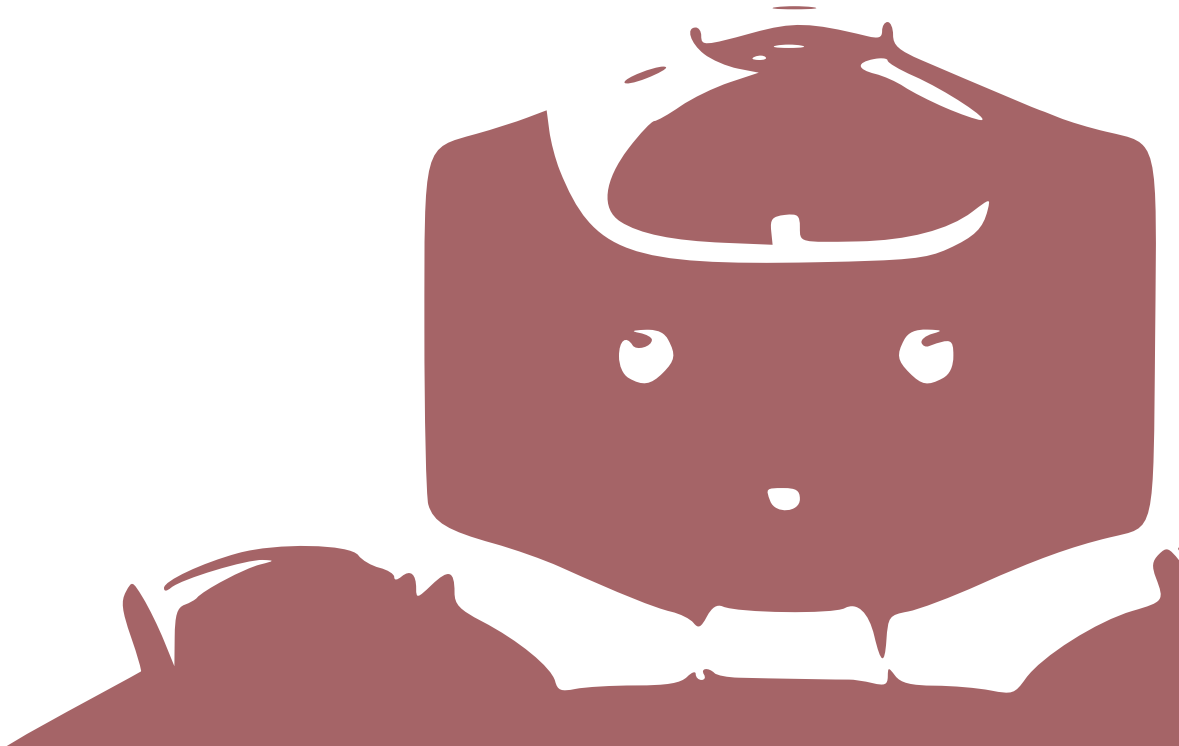


You can download the sources of this presentation here:

github.com/severin-lemaignan/lecture-hri-architectures

Software Architectures for HRI



Séverin Lemaignan

Bristol Robotics Lab
University of the West of England

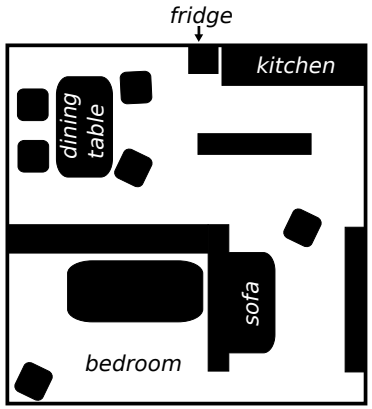
How hard might it be?

LET'S THINKER A LITTLE...

Let's imagine you want to build a robot that **fetches beers from the fridge** and bring them back to you whenever you ask. It should **not kill the cat**, and it should **politely greet your mum** whenever it sees her.

You have:

- a map with the important landmarks like `fridge`



- the following modules:

`speech_synthesis` `door_opening`

`object_picking` `cat_detector`

`robot_navigation`

`route_planner` `person_detector`

`natural_language_processor`

Can you draw a control architecture that achieves just that?

IN THIS LECTURE

- Control paradigms

IN THIS LECTURE

- Control paradigms
- One architecture instantiation

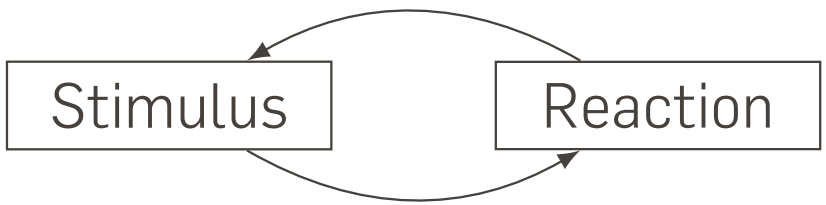
IN THIS LECTURE

- Control paradigms
- One architecture instantiation
- Cognitive architectures

CONTROL PARADIGMS

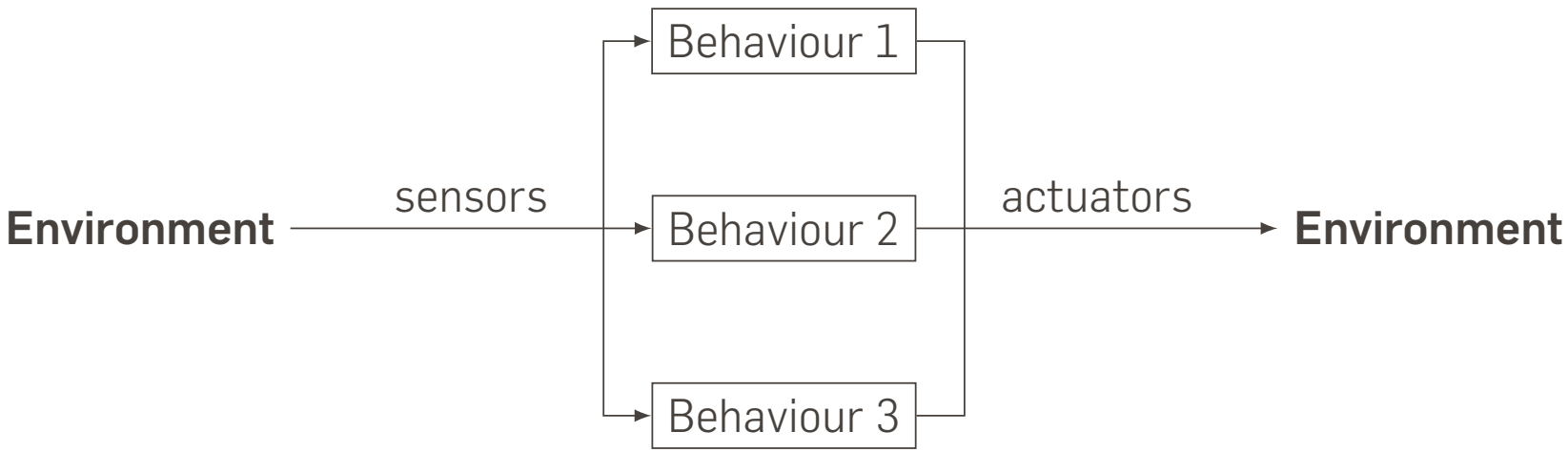
BEHAVIOURAL (OR REACTIVE) CONTROL

Behaviours are small programs that read sensors and control actuators. Each behaviour does **one simple thing**; it typically has access to all sensors/actuators.



BEHAVIOURAL (OR REACTIVE) CONTROL

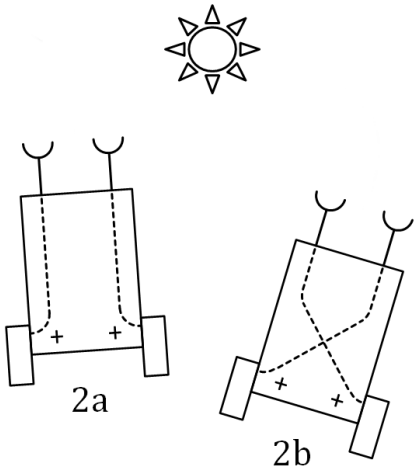
Behaviours are small programs that read sensors and control actuators. Each behaviour does **one simple thing**; it typically has access to all sensors/actuators.



We only want one behaviour at a time! Need to **prioritise**: behaviours can *override* or **subsume** less important ones.

EXTREME CASE: BRAITENBERG MACHINES

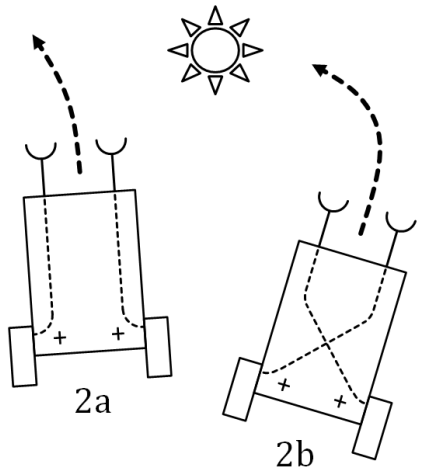
- Motion directly controlled by sensors (typically photocells)
- Yet the resulting behaviour may appear complex or even intelligent
- Can you guess the behaviours of vehicles 2a and 2b?



Source: Wikipedia

EXTREME CASE: BRAITENBERG MACHINES

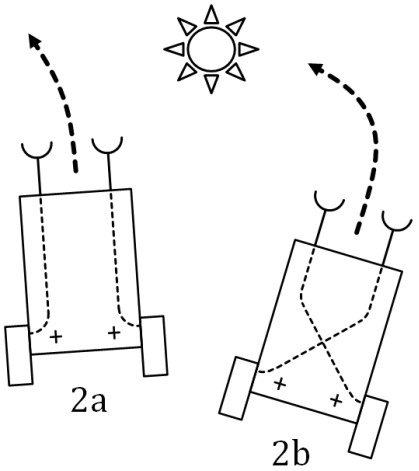
- Motion directly controlled by sensors (typically photocells)
- Yet the resulting behaviour may appear complex or even intelligent
- Can you guess the behaviours of vehicles 2a and 2b?

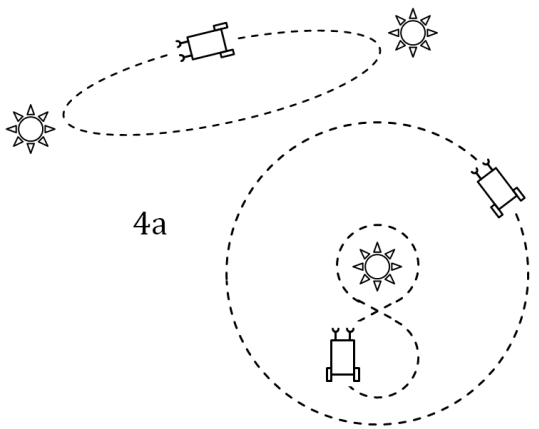


Source: Wikipedia

EXTREME CASE: BRAITENBERG MACHINES

- Motion directly controlled by sensors (typically photocells)
- Yet the resulting behaviour may appear complex or even intelligent





- Complex behaviours emerge (typically with non-linear control functions).

Source: Wikipedia

COMBINING BEHAVIOURS: EXAMPLE

Three behaviours:

- **Follow a human**

  Sensor: camera
  Behaviour: follow a human seen in camera

- **Avoid obstacle**

  Sensors: bumpers
  Behaviour: move away from the wall

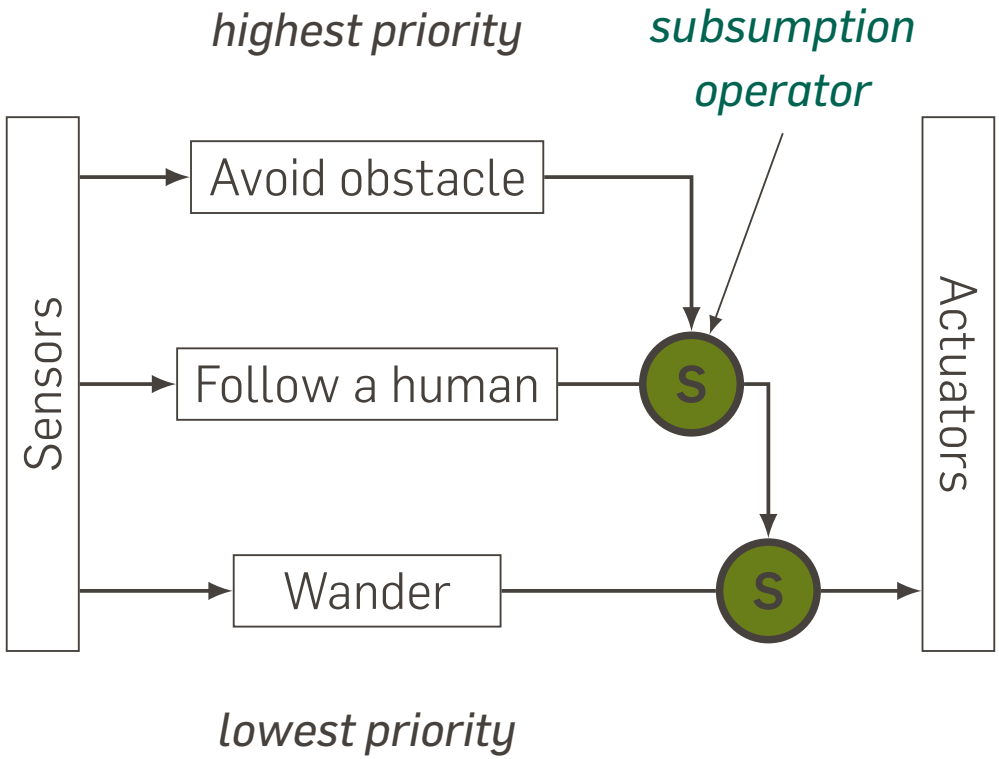
- **Wander**

  Sensors: encoders
  Behaviour: move forward and turn

Which behaviour should have the highest priority? the lowest?

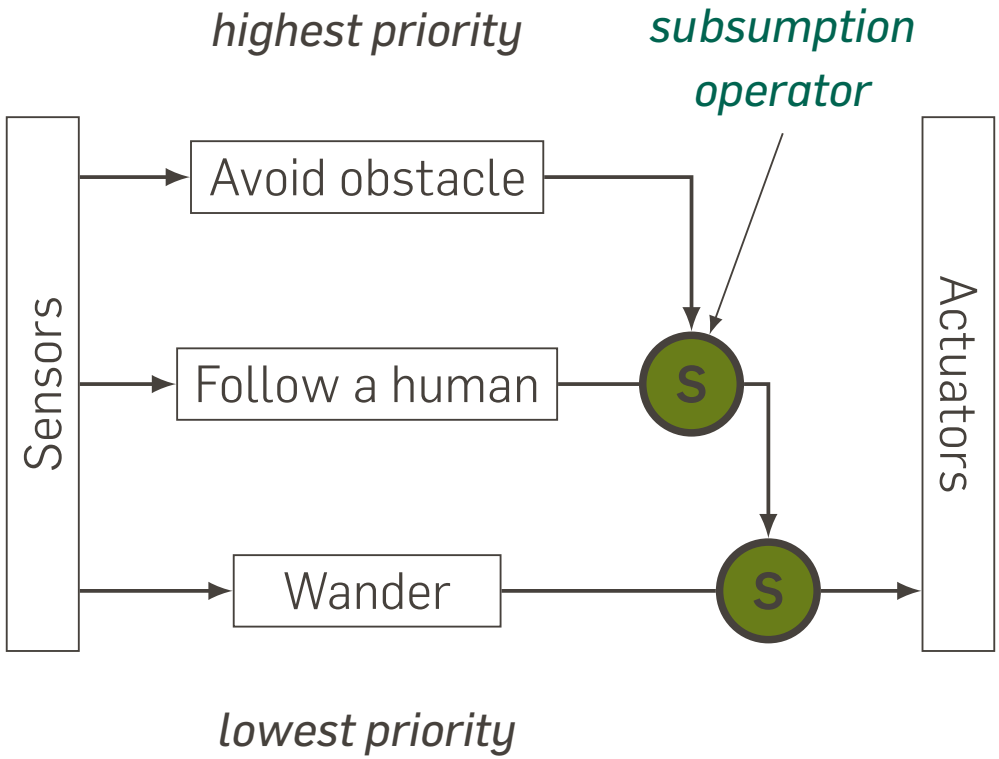
Source: example inspired by Rice University ENGI128

COMBINING BEHAVIOURS: EXAMPLE



We combine behaviours by **subsuming** lower-priority behaviours whenever a higher-priority behaviour becomes active.

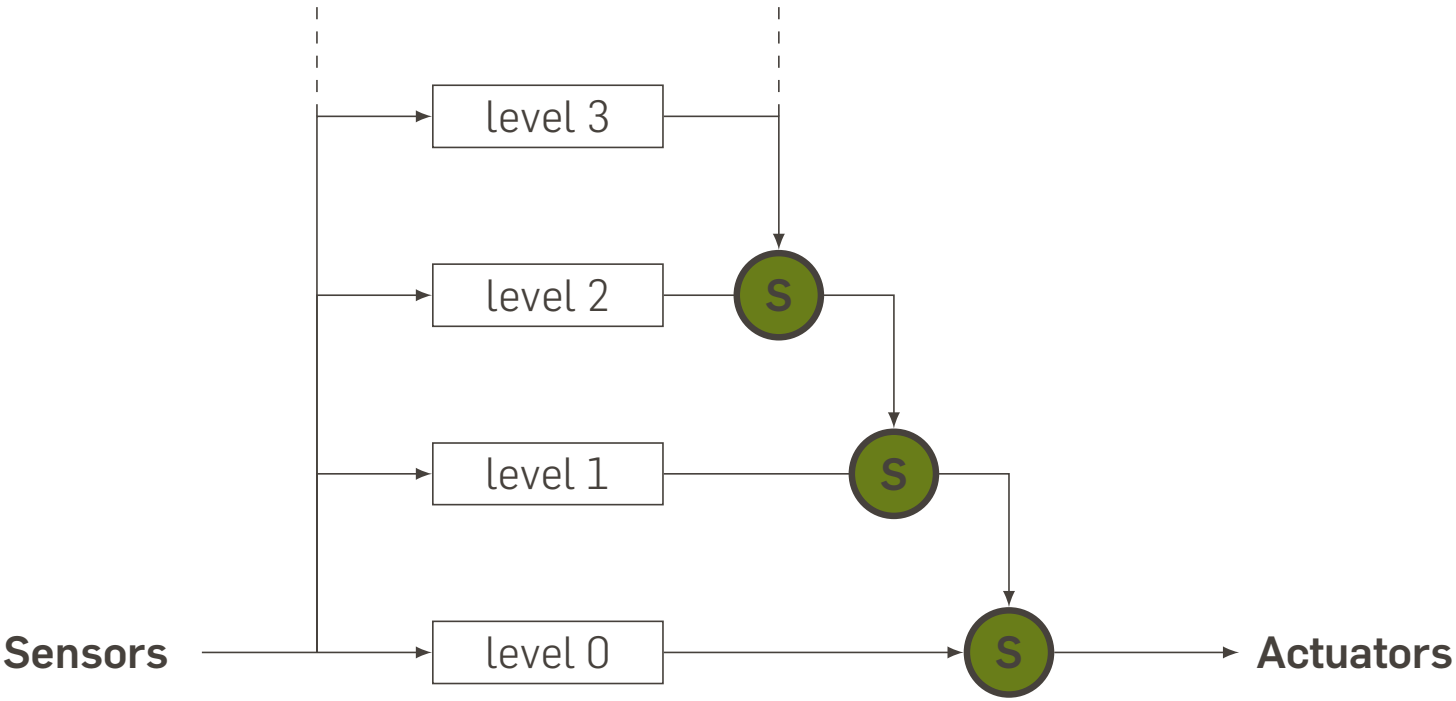
COMBINING BEHAVIOURS: EXAMPLE



We combine behaviours by **subsuming** lower-priority behaviours whenever a higher-priority behaviour becomes active.

→ **subsumption architecture**

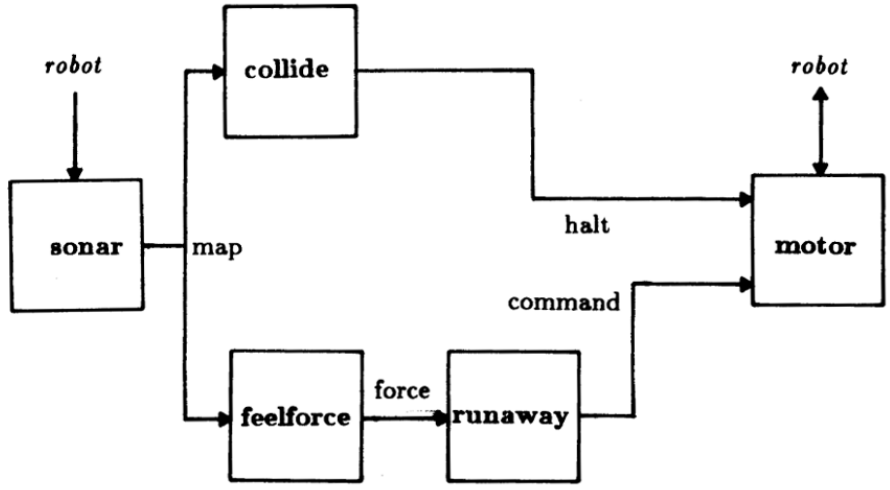
SUBSUMPTION ARCHITECTURE



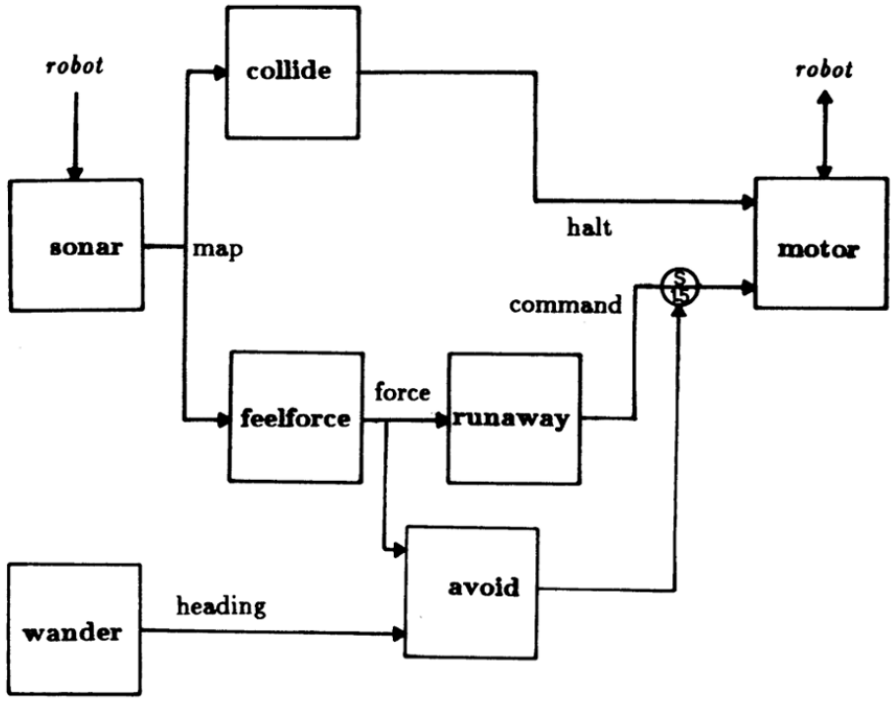
[Brooks **A robust layered control system for a mobile robot**, 1986]

EXAMPLE: THE MIT GENGHIS ROBOT

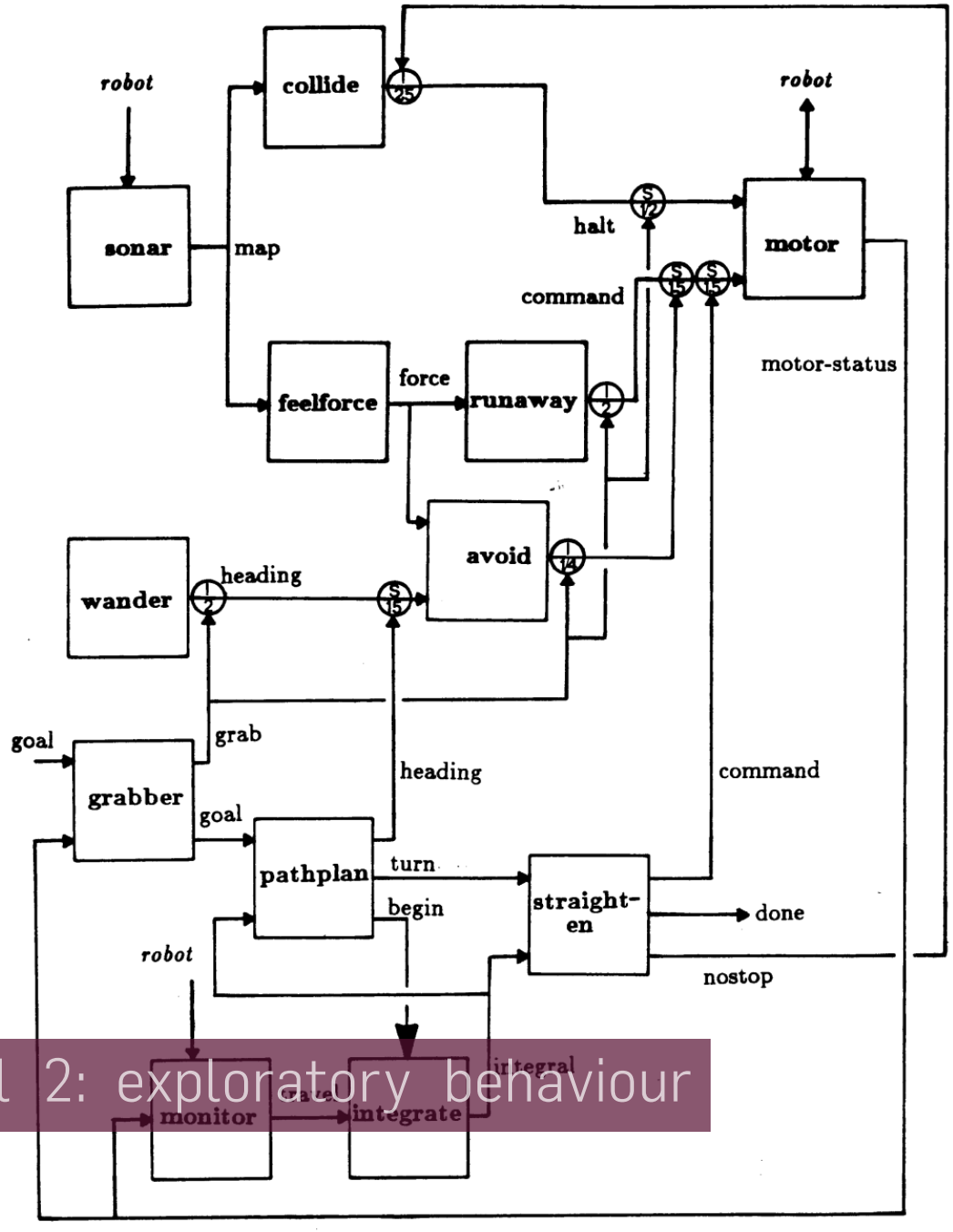
[Brooks **A robust layered control system for a mobile robot**, 1986]



Level 0: obstacle avoidance



Level 1: wander around aimlessly



Level 2: exploratory behaviour

BEHAVIOURAL CONTROL: STRENGTHS/WEAKNESSES

Strengths

- **Incremental development**
- By definition **modular**
- Effective to react to events → well suited to **dynamic environments**

BEHAVIOURAL CONTROL: STRENGTHS/WEAKNESSES

Strengths

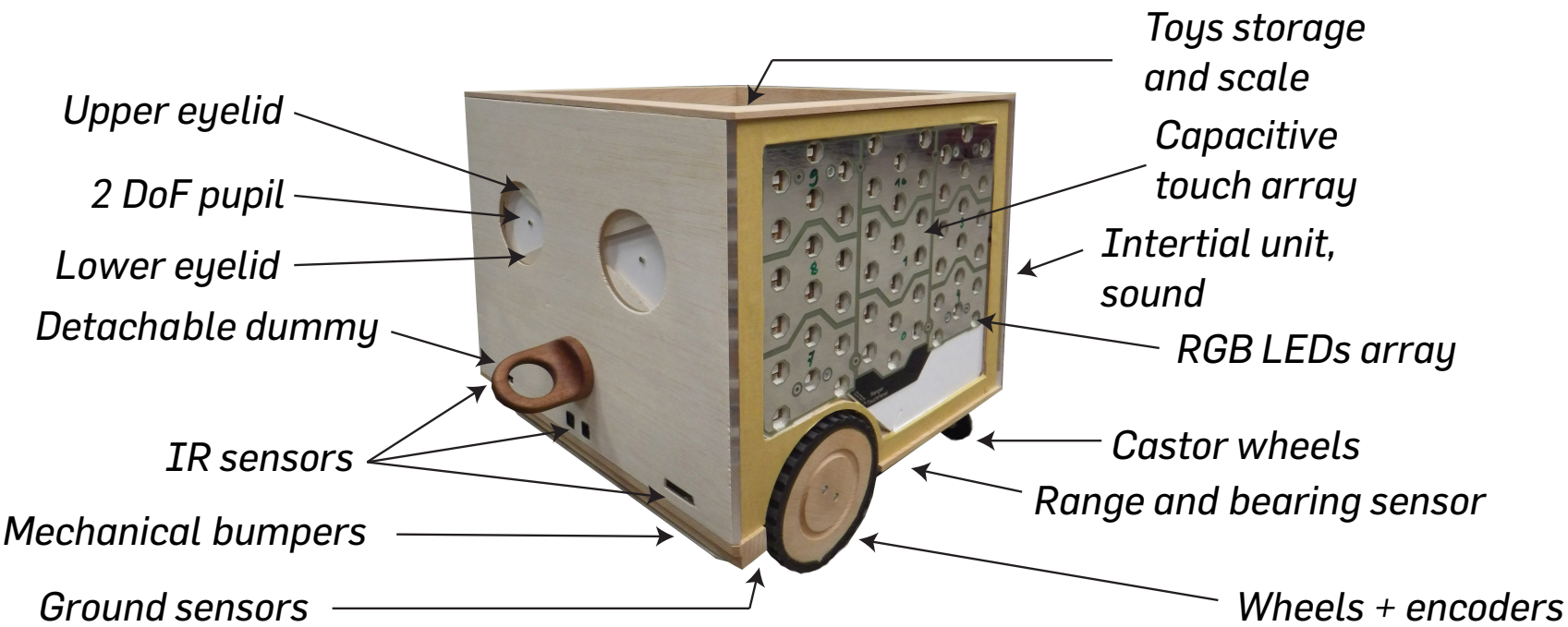
- **Incremental development**
- By definition **modular**
- Effective to react to events → well suited to **dynamic environments**

Weaknesses

- **goal-oriented behaviours hard to implement** ("what will my robot do?")
- **importance of the arbiter**: who inhibits (i.e. *subsumes*) who might be context-dependent
- debugging difficult (need to trace which behaviours are active)

EVENT-ORIENTED PROGRAMMING: EXAMPLE OF RANGER

Ranger is a ‘box on wheels’ developed at EPFL



EVENT-ORIENTED PROGRAMMING

Event-oriented programming is a possible way of implementing a behavioural control paradigm:

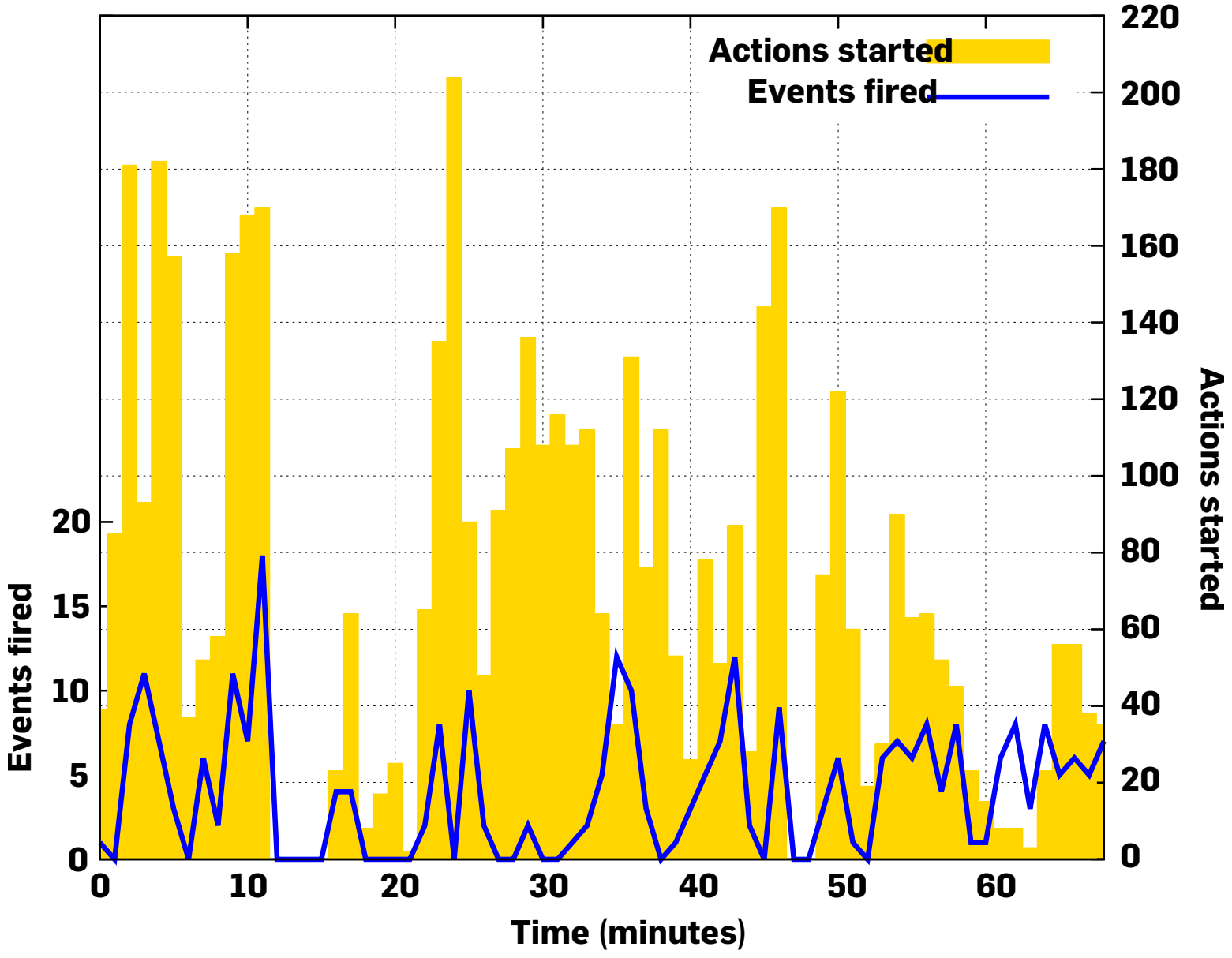
```
with Ranger() as robot:

  robot.background_blink()

  robot.whenever("dummy", becomes = True).do(on_dummy)
  robot.whenever("dummy", becomes = False).do(on_dummy_removed)
  robot.whenever("scale", increase = 0.3).do(on_toy)
  robot.whenever("bumper", becomes = True).do(on_bumper)

  while True:
      time.sleep(0.1)
```

EVENT-ORIENTED PROGRAMMING

Event-oriented programming is a possible way of implementing a behavioural control paradigm:

```
def on_dummy(robot):
    robot.look_at_dummy()
    robot.blink()
    sleep = robot.fall_asleep()
    robot.lightbar(RAINBOW).wait()
    sleep.wait()

def on_dummy_removed(robot):
    robot.light_bar(colors.rand())
    robot.wakeup().wait()
    robot.move(0.4, v = 0.8).wait()
    robot.idle().wait()
```

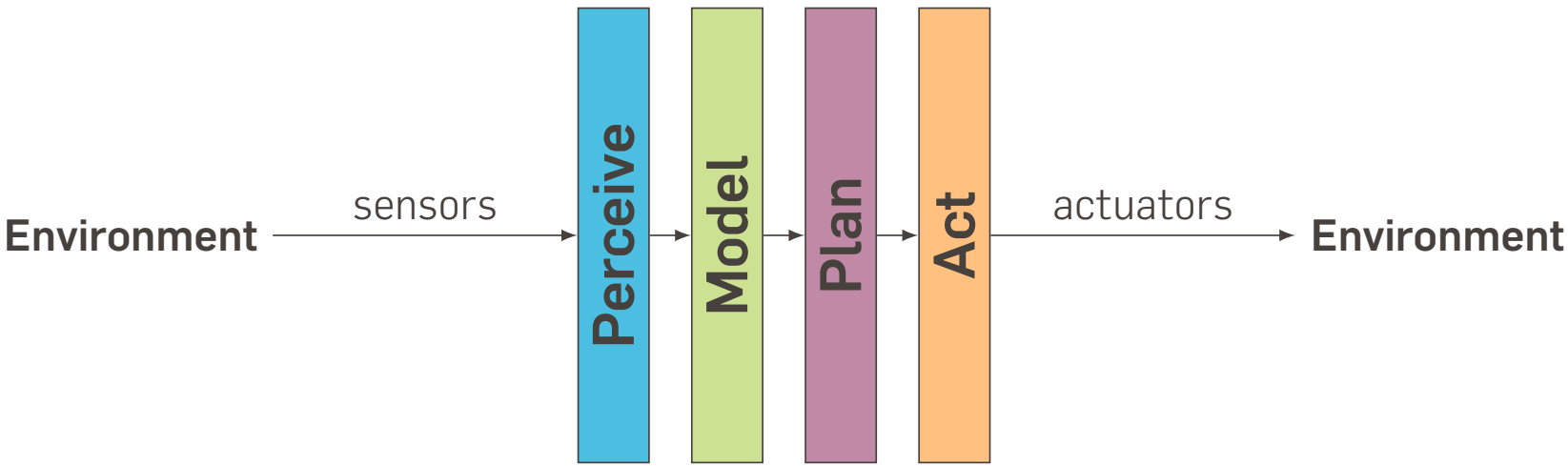
```
def on_bumper(robot):
    pulse = robot.pulse_row(0)
    while abs(robot.state.v) > 0.01:
        robot.sleep(0.2)
    pulse.cancel()

def on_toy(robot):
    robot.playsound(SOUNDS["toy_in"])
    robot.lightbar(RAINBOW).wait()
```


Actions started
Events fired
Events fired
Actions started
Time (minutes)
0
5
10
15
20
0
20
40
60
80
100
120
140
160
180
200
220
0
10
20
30
40
50
60

Note that this is a example of **aimless, purely reactive**, behaviour.

MODEL-PLAN-ACT

Basic paradigm for **deliberative architectures**.



TASK PLANNING

Turn a high-level goal (*bring me a beer!*) into 'simple' **primitive** actions.

A standard approach relies on **Hierarchical task networks** (**HTN**): uses partial-order contraints to *decompose actions* into *primitive operators*.

TASK PLANNING

Turn a high-level goal (*bring me a beer!*) into 'simple' **primitive** actions.

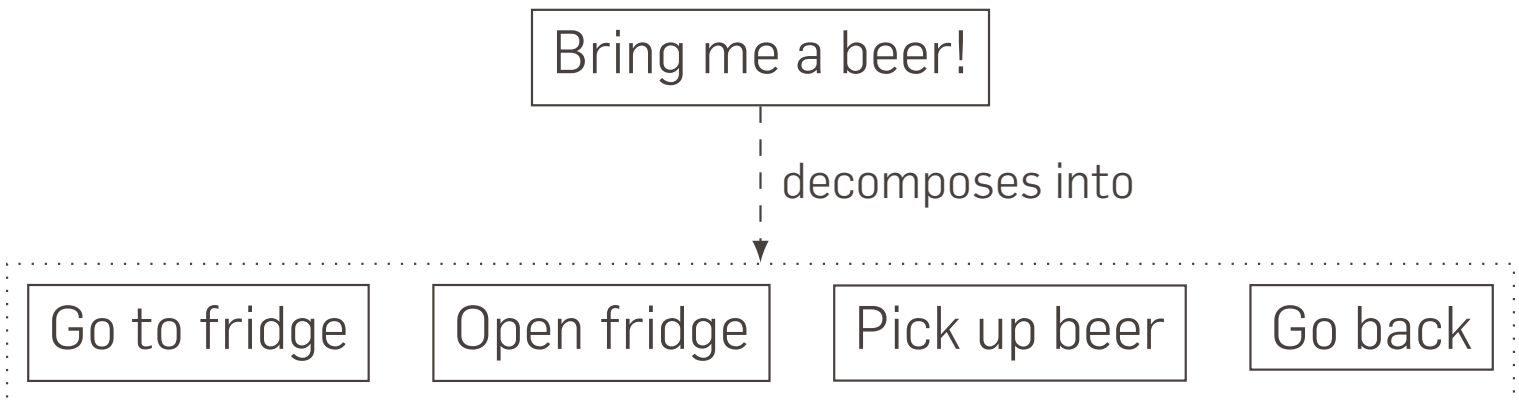
A standard approach relies on **Hierarchical task networks** (**HTN**): uses partial-order contraints to *decompose actions* into *primitive operators*.

Bring me a beer!

TASK PLANNING

Turn a high-level goal (*bring me a beer!*) into 'simple' **primitive** actions.

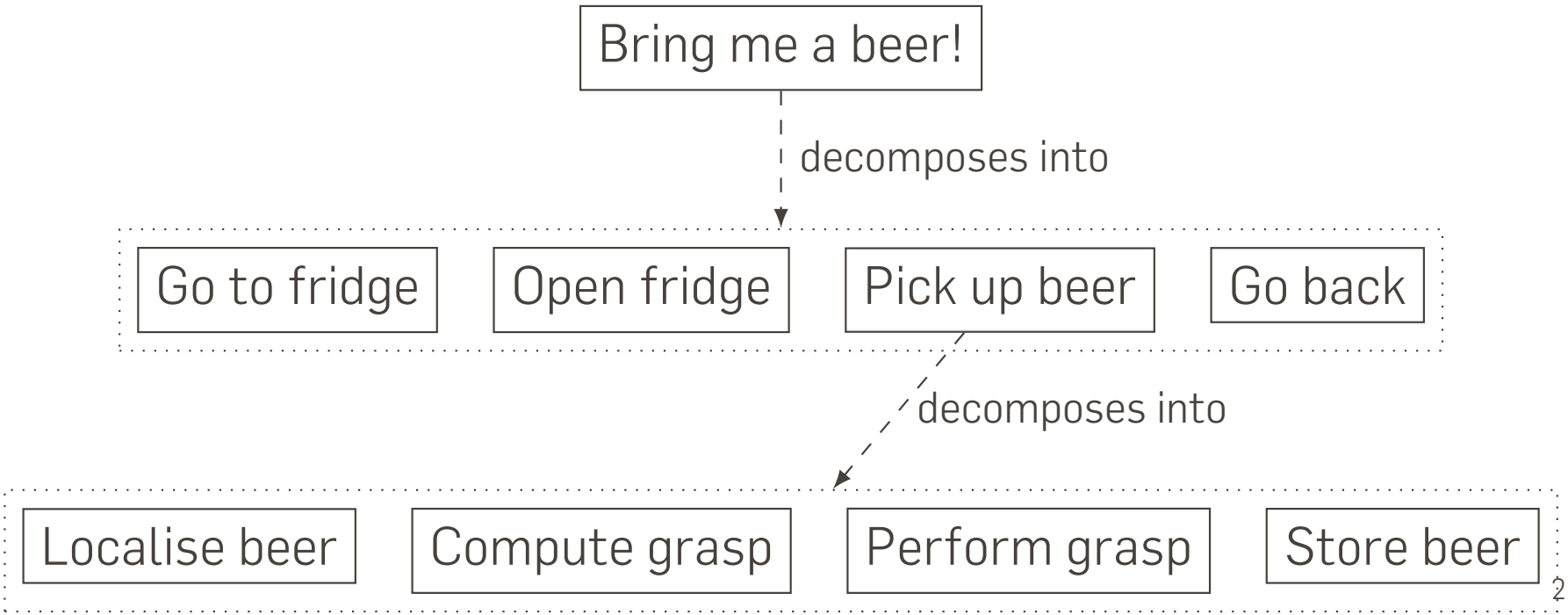
A standard approach relies on **Hierarchical task networks** (**HTN**): uses partial-order contraints to *decompose actions* into *primitive operators*.



TASK PLANNING

Turn a high-level goal (*bring me a beer!*) into 'simple' **primitive** actions.

A standard approach relies on **Hierarchical task networks** (**HTN**): uses partial-order contraints to *decompose actions* into *primitive operators*.

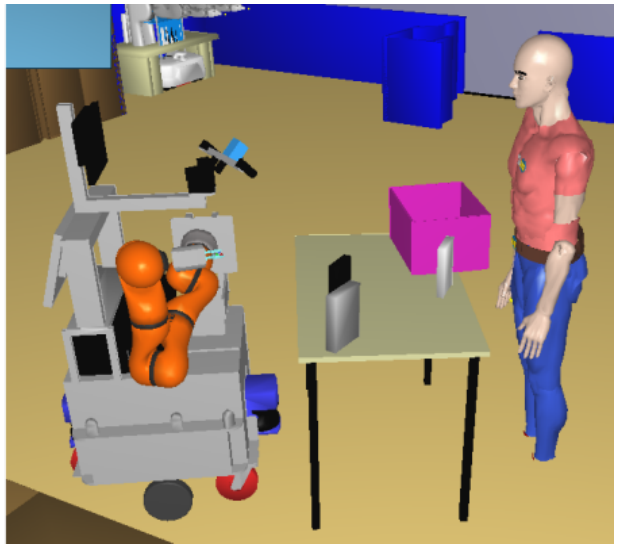


TASK PLANNING

What ***primitive action*** means is system-dependent: what an agent considers as primitive can be another agent's plans.

How would that look, if applied to HRI?

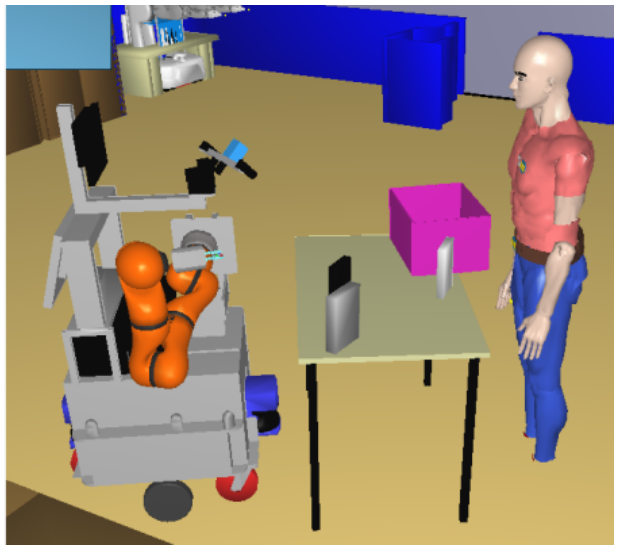
TASK PLANNING

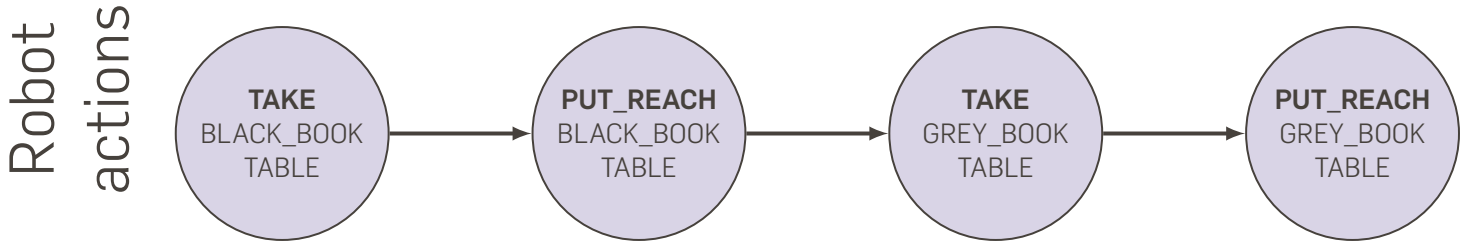


Goal: (collaboratively) put the books into the pink bin.

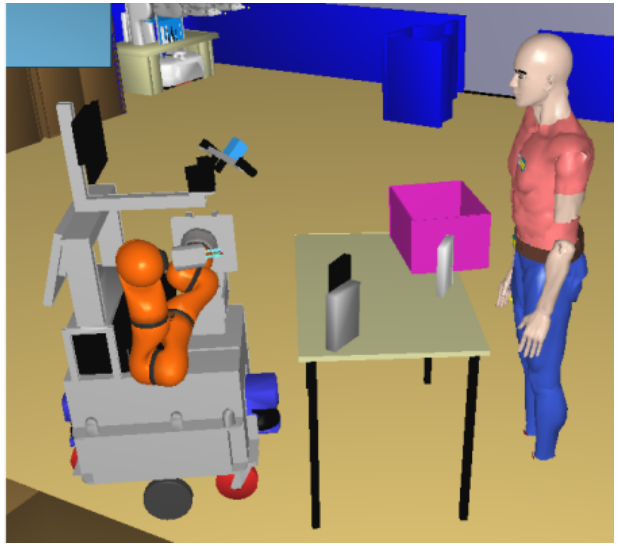
- The bin is only accessible to the human
- Three books: a black one and a grey one (only accessible to robot), a white one (only accessible to the human).

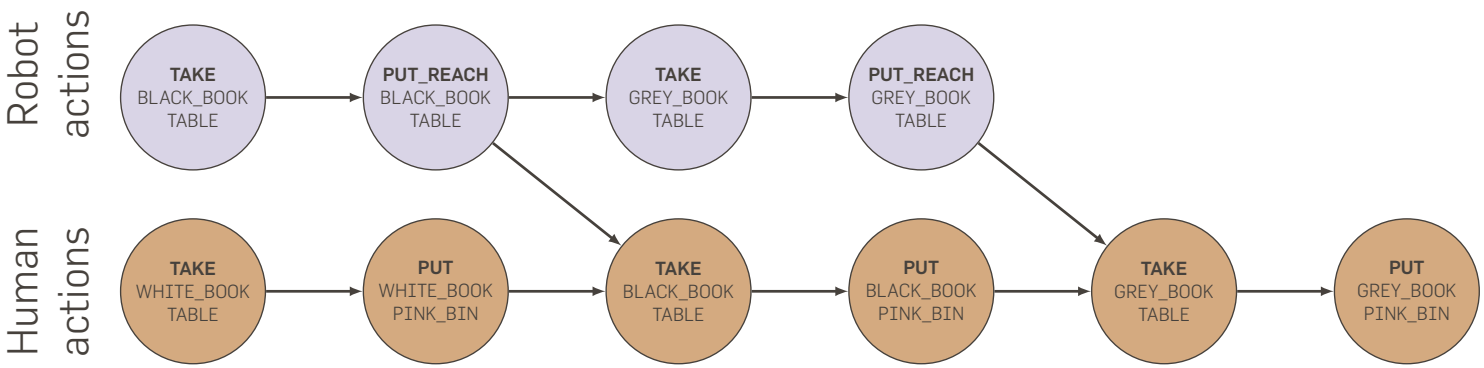
TASK PLANNING





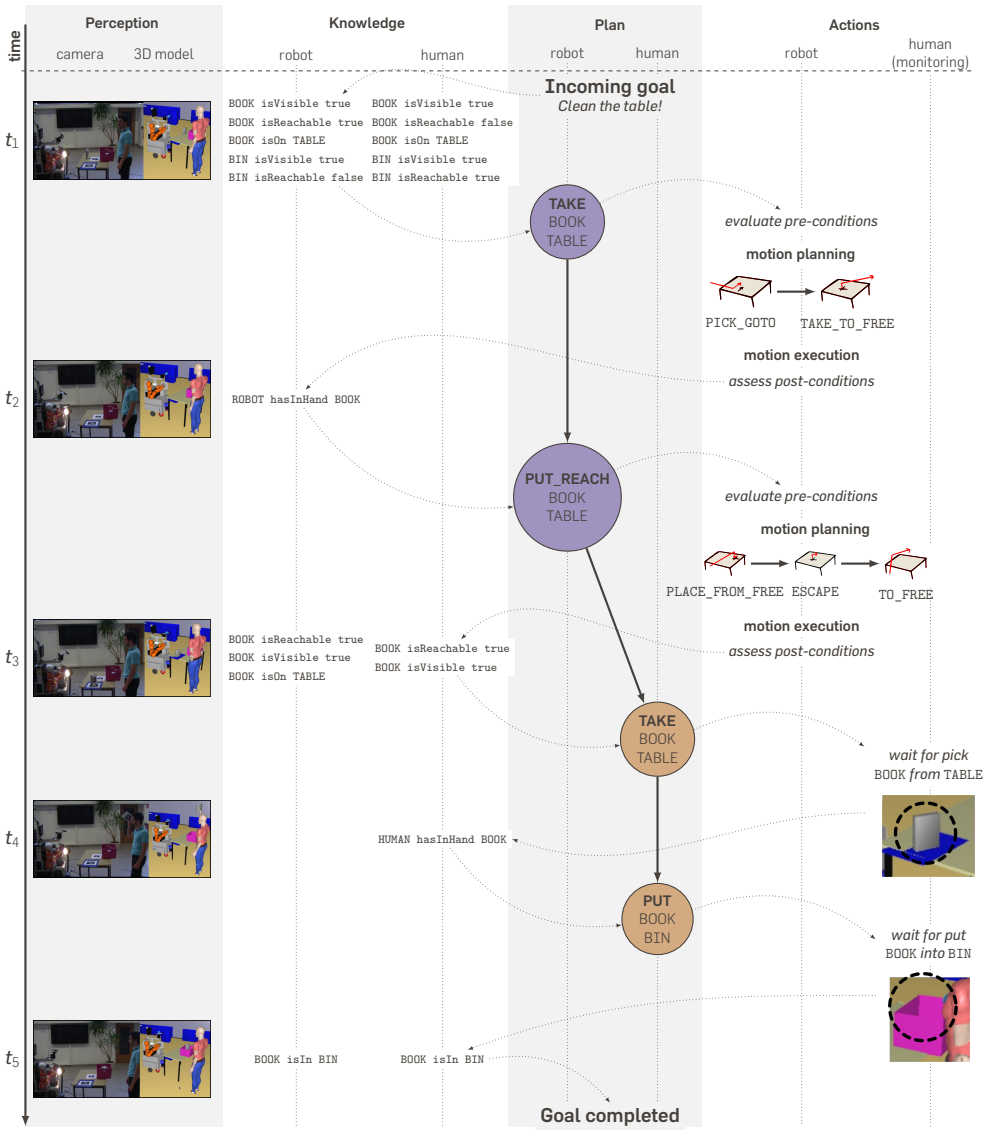
TASK PLANNING






LAAS-CNRS

FULL SOCIAL & AUTONOMOUS INTERACTION: ONE EXAMPLE

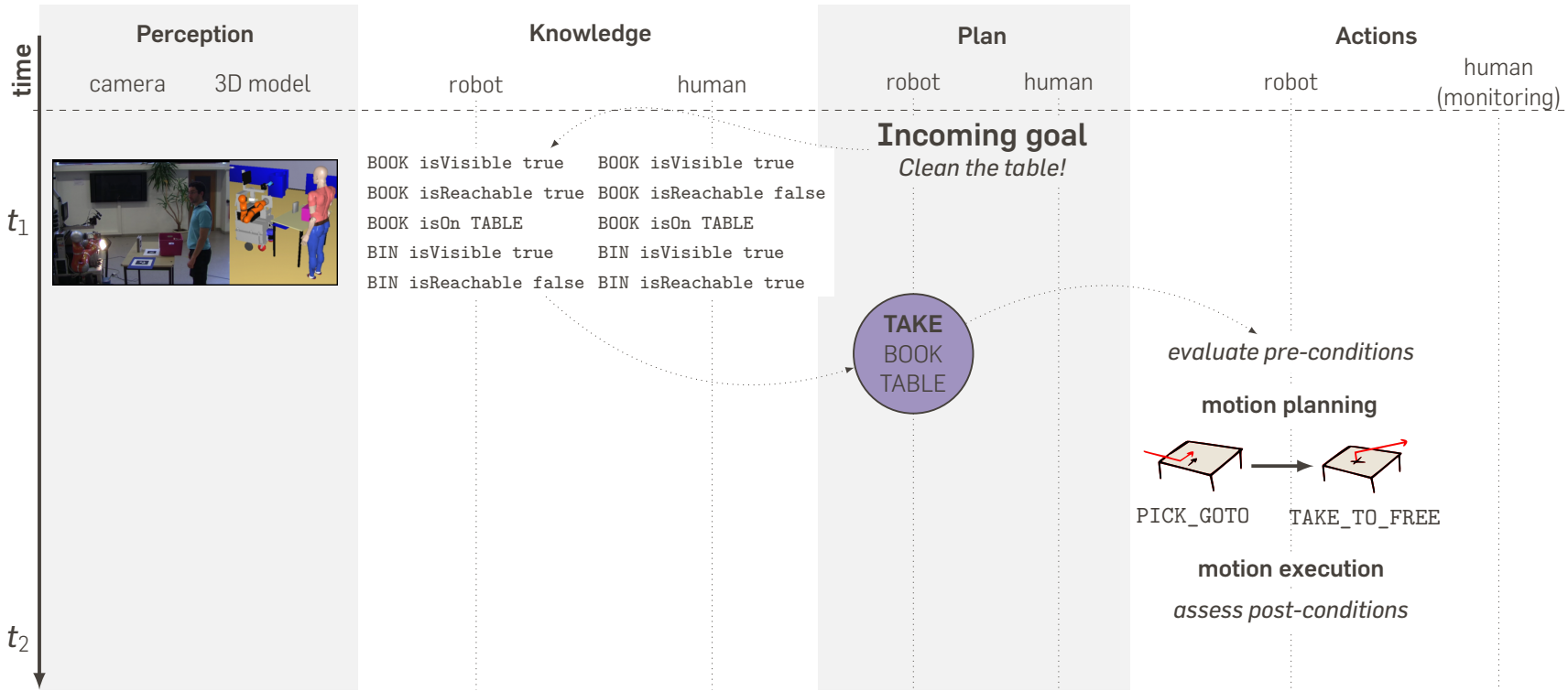


[Lemaignan et al. **Artificial Cognition for Social Human-Robot Interaction: An Implementation** Artifical Intelligence, 2017]

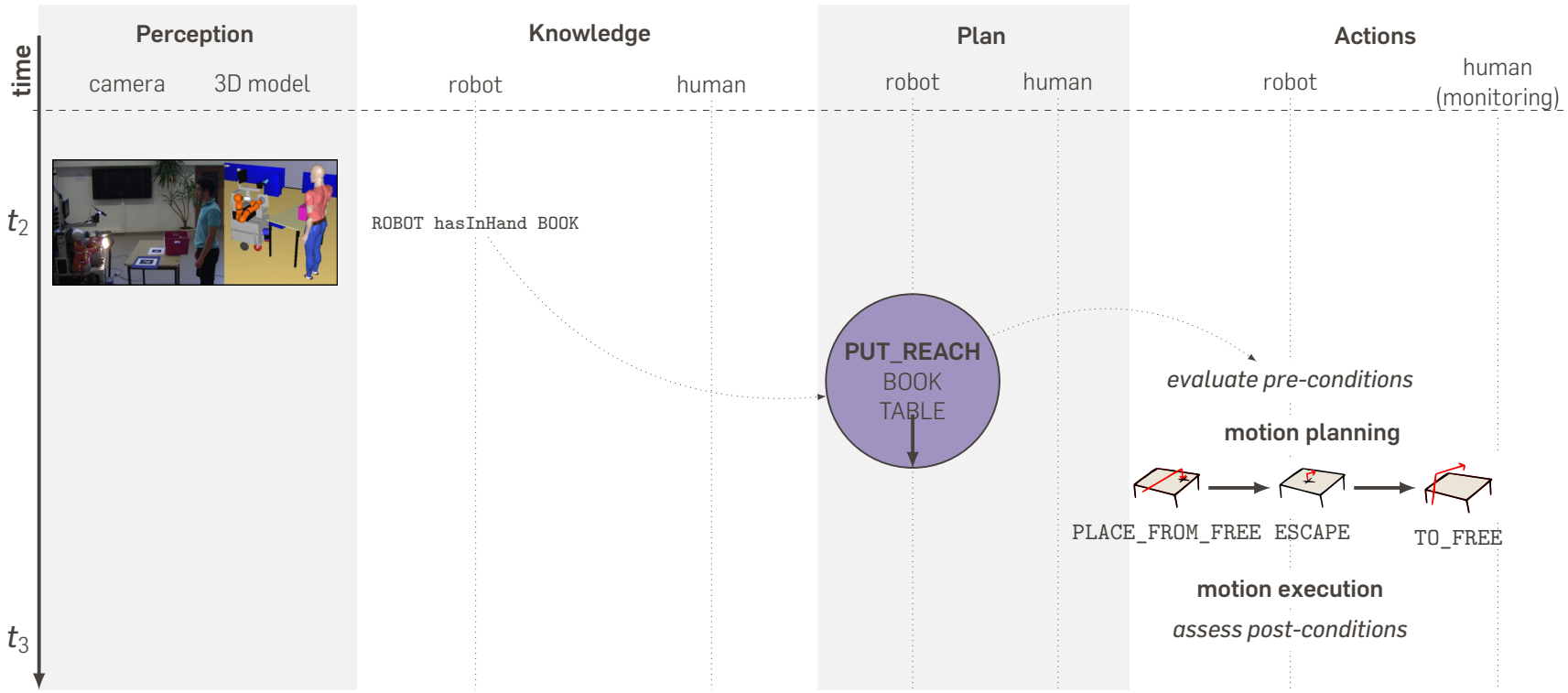
“CLEANING THE TABLE”...

Example of a subplan: one book, accessible to the robot, to be handed over to the human and discarded.

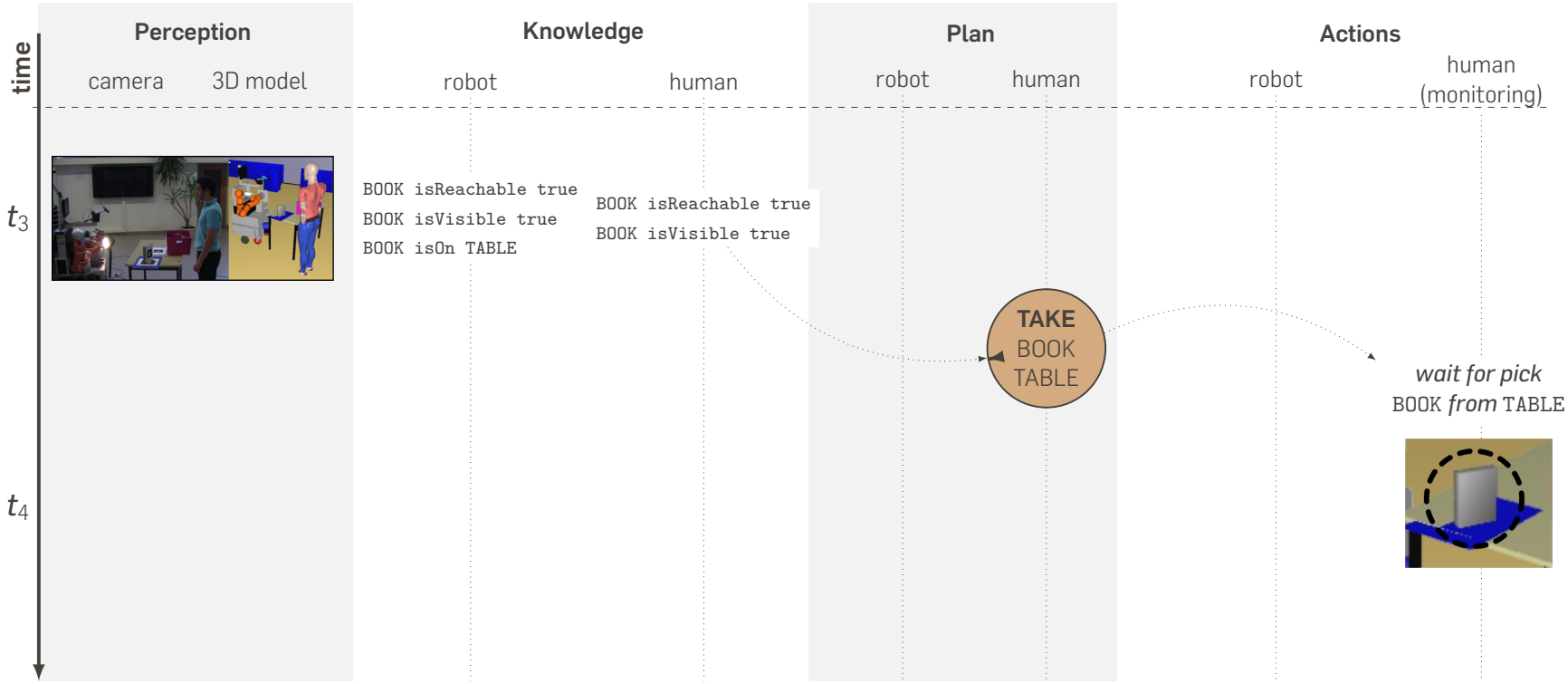
"CLEANING THE TABLE"...



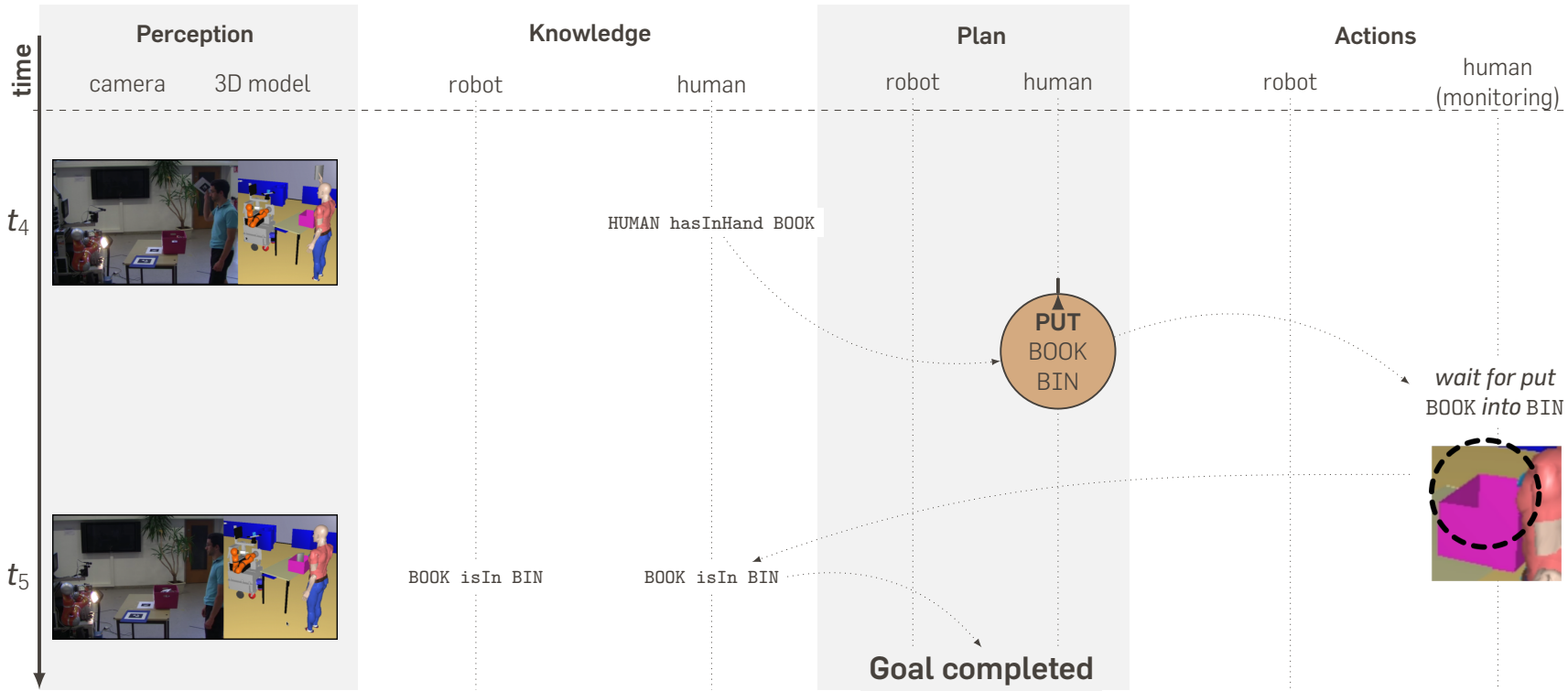
"CLEANING THE TABLE"...



"CLEANING THE TABLE"...



"CLEANING THE TABLE"...



FROM AN ARCHITECTURE PERSPECTIVE

CONTROL STRATEGIES

Behavioural or reactive

- *bottom-up* approach
- lots of independent modules executing concurrently, monitoring sensor values, triggering actions and possibly inhibiting each others
- hard to organize into complex behaviours; gets messy quickly

CONTROL STRATEGIES

Behavioural or reactive

- *bottom-up* approach
- lots of independent modules executing concurrently, monitoring sensor values, triggering actions and possibly inhibiting each others
- hard to organize into complex behaviours; gets messy quickly

Hierarchical: classic model/plan/act

- *top-down* approach
- starts with high-level goals, decompose into sub-tasks
- not very agile, reacting to unexpected events hard

CONTROL STRATEGIES

Behavioural or reactive

- *bottom-up* approach
- lots of independent modules executing concurrently, monitoring sensor values, triggering actions and possibly inhibiting each others
- hard to organize into complex behaviours; gets messy quickly

Hierarchical: classic model/plan/act

- *top-down* approach
- starts with high-level goals, decompose into sub-tasks
- not very agile, reacting to unexpected events hard

Hybrid approaches?

- Deliberative at high level, reactive at low level

LEVELS OF CONTROL

Control problems are usually split into three levels:

- **Low-level control**

  Example: where to place a leg as robot takes its next step
  Generally, continuous-valued problems
  Short time scale (under a second); high frequency loop

Source: CMU lecture on Cognitive Robotics

LEVELS OF CONTROL

Control problems are usually split into three levels:

- **Low-level control**

  Example: where to place a leg as robot takes its next step
  Generally, continuous-valued problems
  Short time scale (under a second); high frequency loop

- **Intermediate level control**

  Navigating to a destination, or picking up an object → *capabilities*
  Continuous or discrete valued problems
  Time scale of a few seconds

Source: CMU lecture on Cognitive Robotics

LEVELS OF CONTROL

Control problems are usually split into three levels:

- **Low-level control**

  Example: where to place a leg as robot takes its next step
  Generally, continuous-valued problems
  Short time scale (under a second); high frequency loop

- **Intermediate level control**

  Navigating to a destination, or picking up an object → *capabilities*
  Continuous or discrete valued problems
  Time scale of a few seconds

- **High level control**

  What is the plan for moving these boxes out of the room?
  Discrete problems, long time scale (minutes)

Source: CMU lecture on Cognitive Robotics

LAYERED ARCHITECTURES

High-level layer
Deliberative/decisional layer

Intermediate layer
Executive layer

Low-level layer
Functional layer

Each layer has different time constraints (from real-time to possibly 'slow'); they typically use different control paradigms

DELIBERATIVE ARCHITECTURE FOR INTERACTION

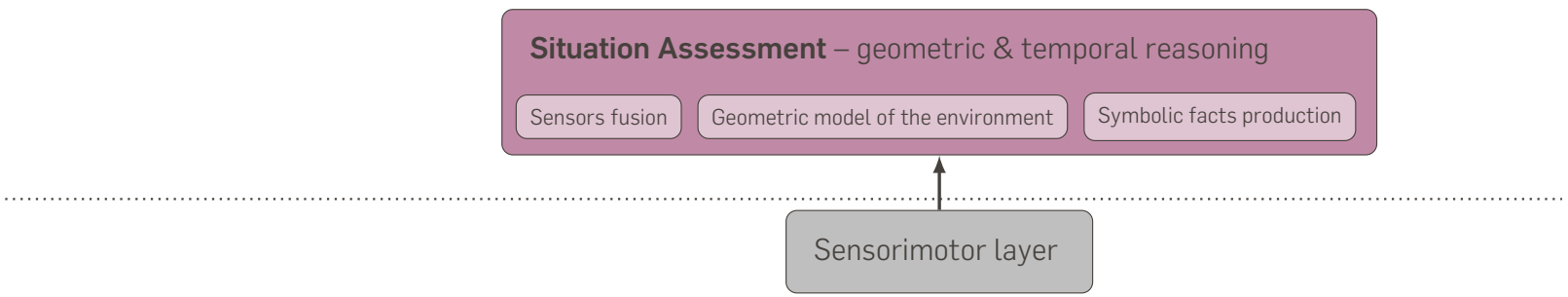
One example: the LAAS deliberative architecture for interaction

[Lemaignan et al., **Artificial Cognition for Social Human-Robot Interaction: An Implementation**, Artifical Intelligence, 2017]

A SYMBOLIC CONTROL ARCHITECTURE

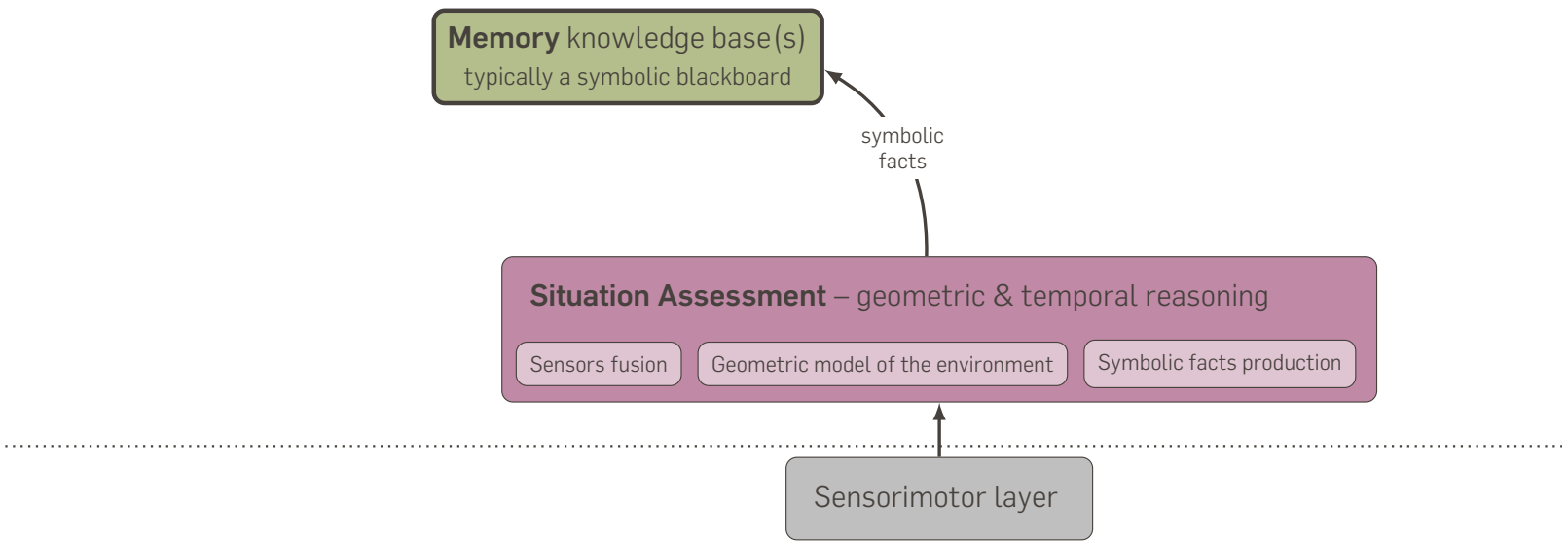
Sensorimotor layer

A SYMBOLIC CONTROL ARCHITECTURE



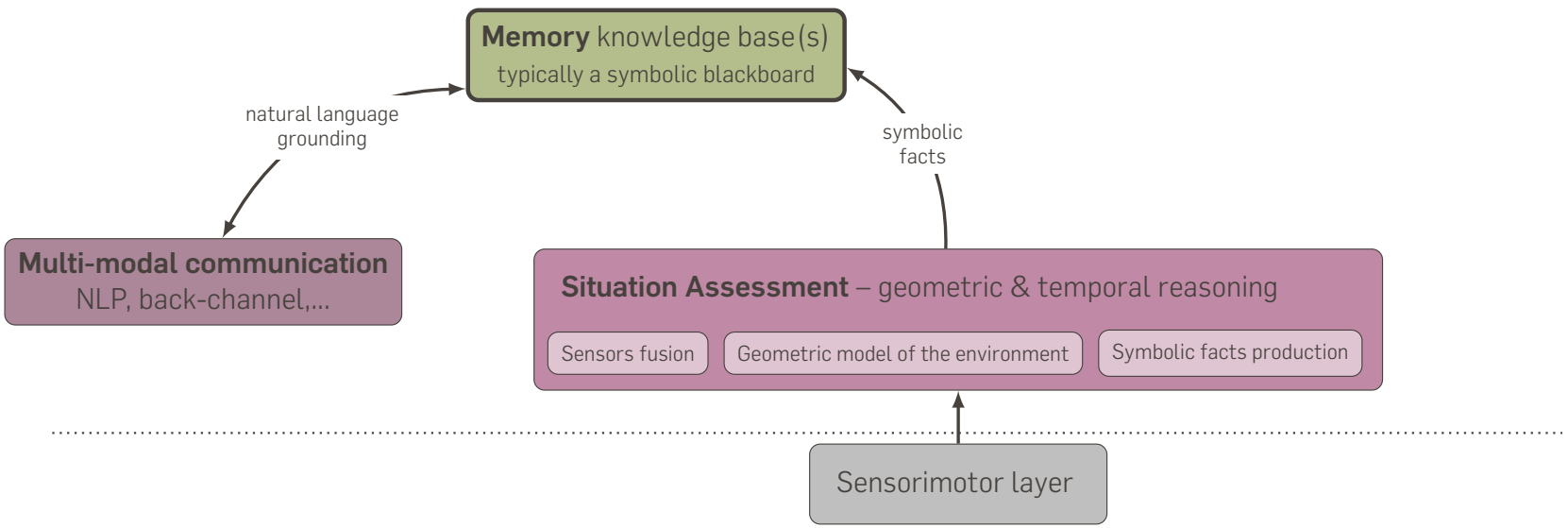
[Lemaignan et al., **Artificial Cognition for Social Human-Robot Interaction: An Implementation**, Artifical Intelligence, 2017]

A SYMBOLIC CONTROL ARCHITECTURE



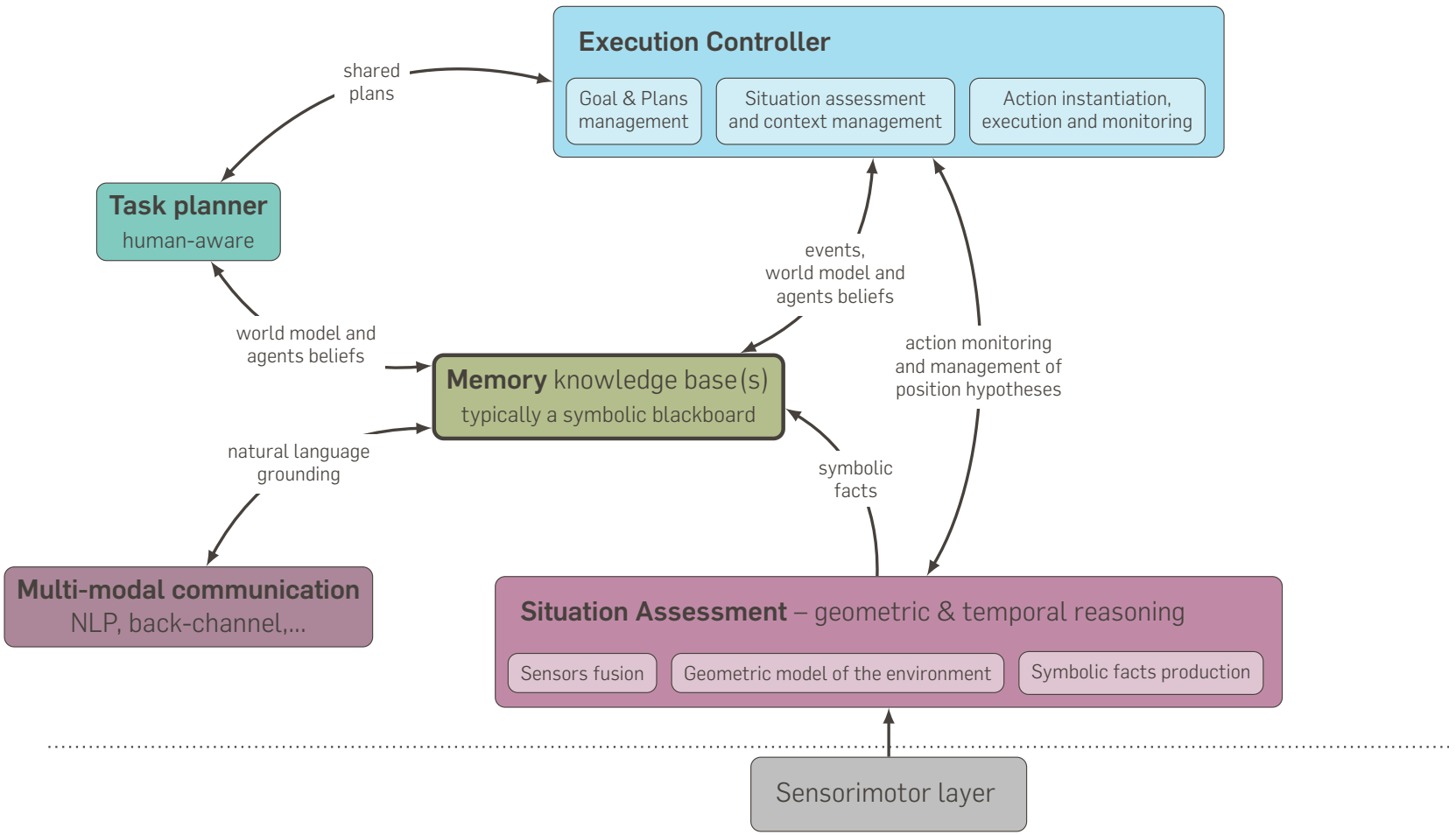
[Lemaignan et al., **Artificial Cognition for Social Human-Robot Interaction: An Implementation**, Artifical Intelligence, 2017]

A SYMBOLIC CONTROL ARCHITECTURE



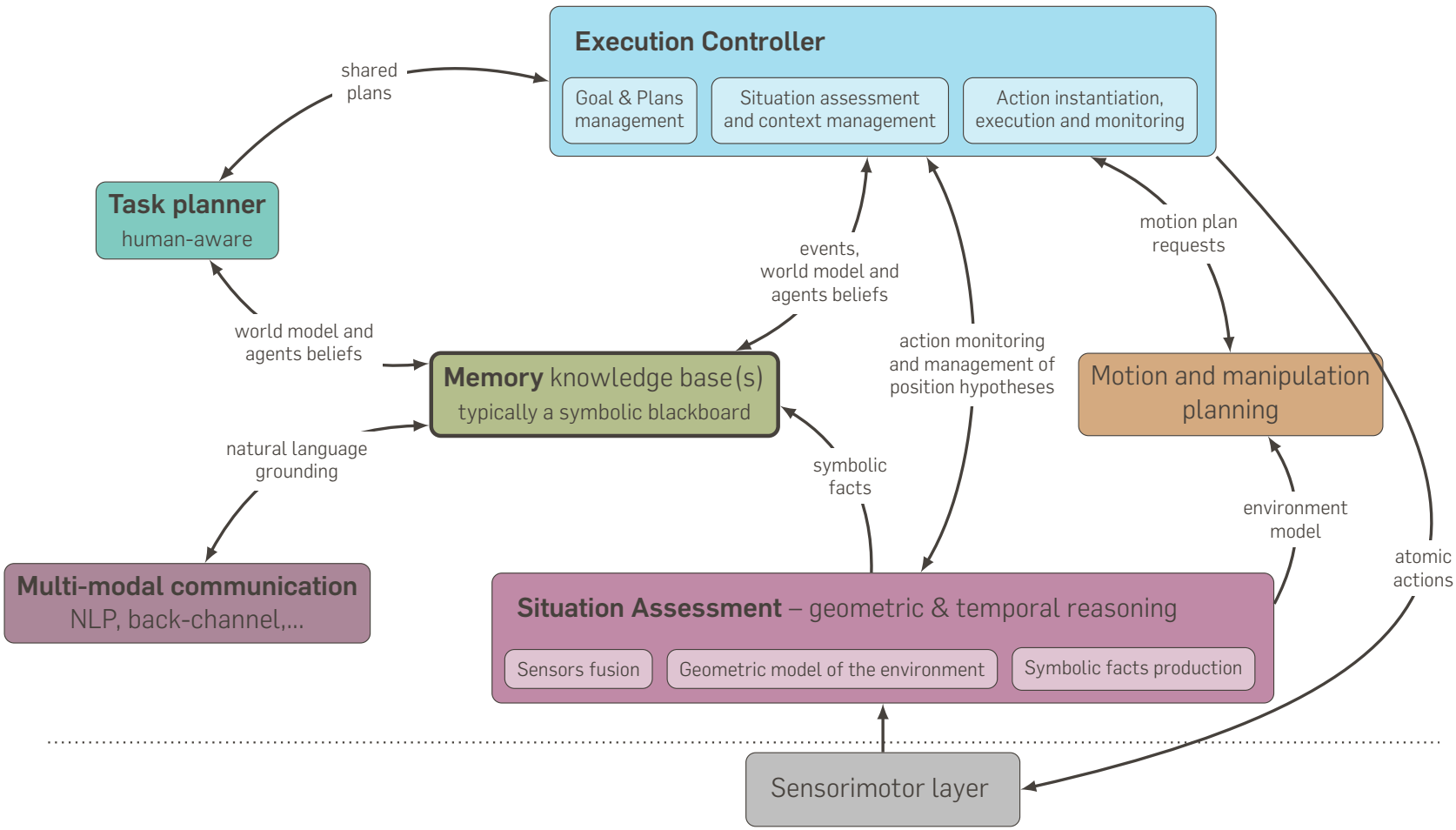
[Lemaignan et al., **Artificial Cognition for Social Human-Robot Interaction: An Implementation**, Artifical Intelligence, 2017]

A SYMBOLIC CONTROL ARCHITECTURE



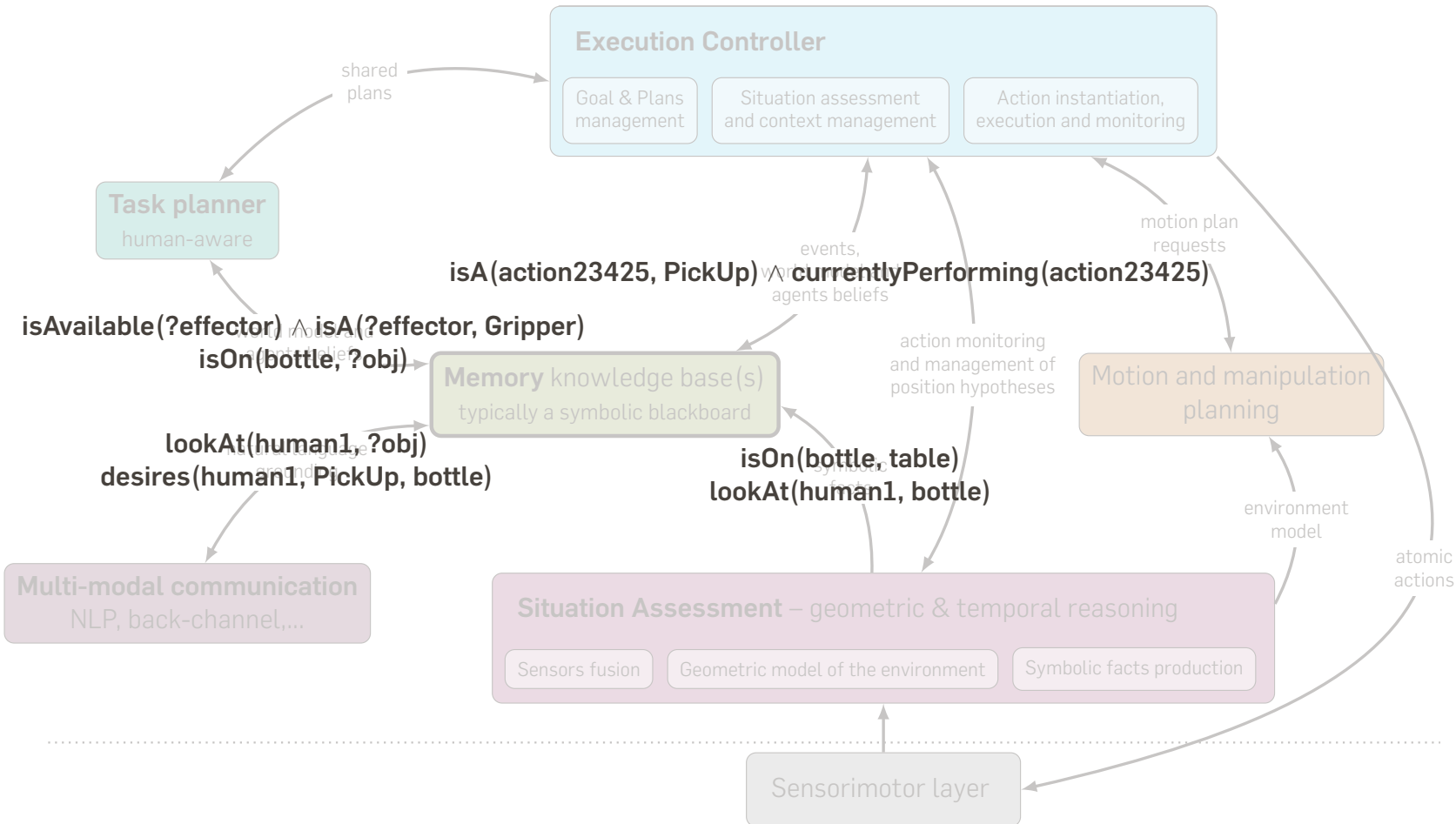
[Lemaignan et al., **Artificial Cognition for Social Human-Robot Interaction: An Implementation**, Artifical Intelligence, 2017]

A SYMBOLIC CONTROL ARCHITECTURE



[Lemaignan et al., **Artificial Cognition for Social Human-Robot Interaction: An Implementation**, Artifical Intelligence, 2017]

A SYMBOLIC CONTROL ARCHITECTURE



[Lemaignan et al., **Artificial Cognition for Social Human-Robot Interaction: An Implementation**, Artifical Intelligence, 2017]

A GLIMPSE AT COGNITIVE ARCHITECTURES

COGNITION?


Adaptation
Problem solving
Learning
Reasoning
Task Planning
Decision making
Action Execution
Cognition
Attention
Perception
Commu-nication
Knowledge Repre-sentation
Commu-nication
Theory of Mind
Memory

COGNITION?

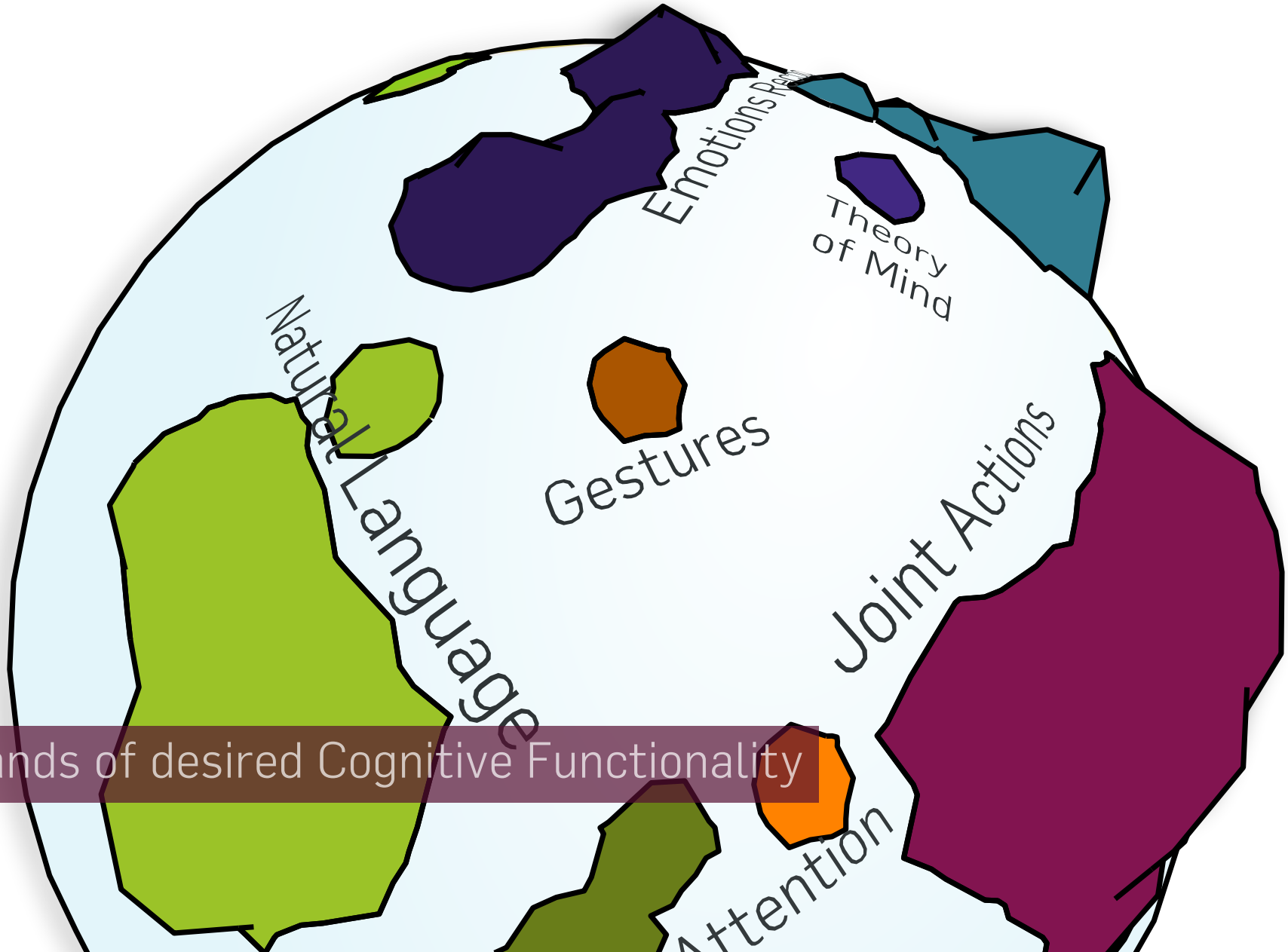

Adaptation
Problem solving
Learning
Task Planning
Reasoning
Decision making
Cognition
Action Execution
Attention
Perception
Commu-nication
Knowledge Repre-sentation
Theory of Mind
Memory
Commu-nication

What is a cognitive architecture?

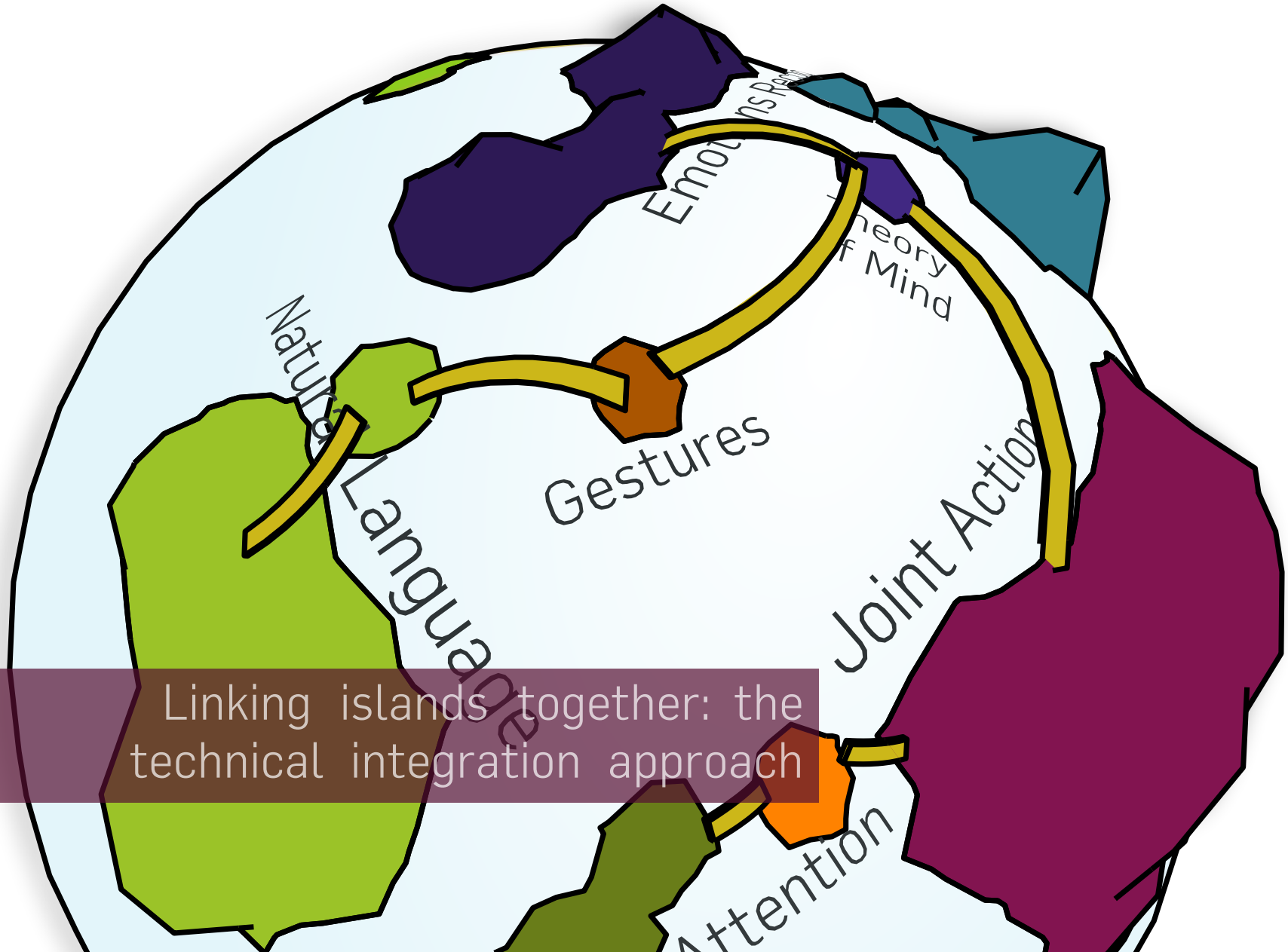
Support modelling (human) social cognition
(application to robots nice to have)

or...

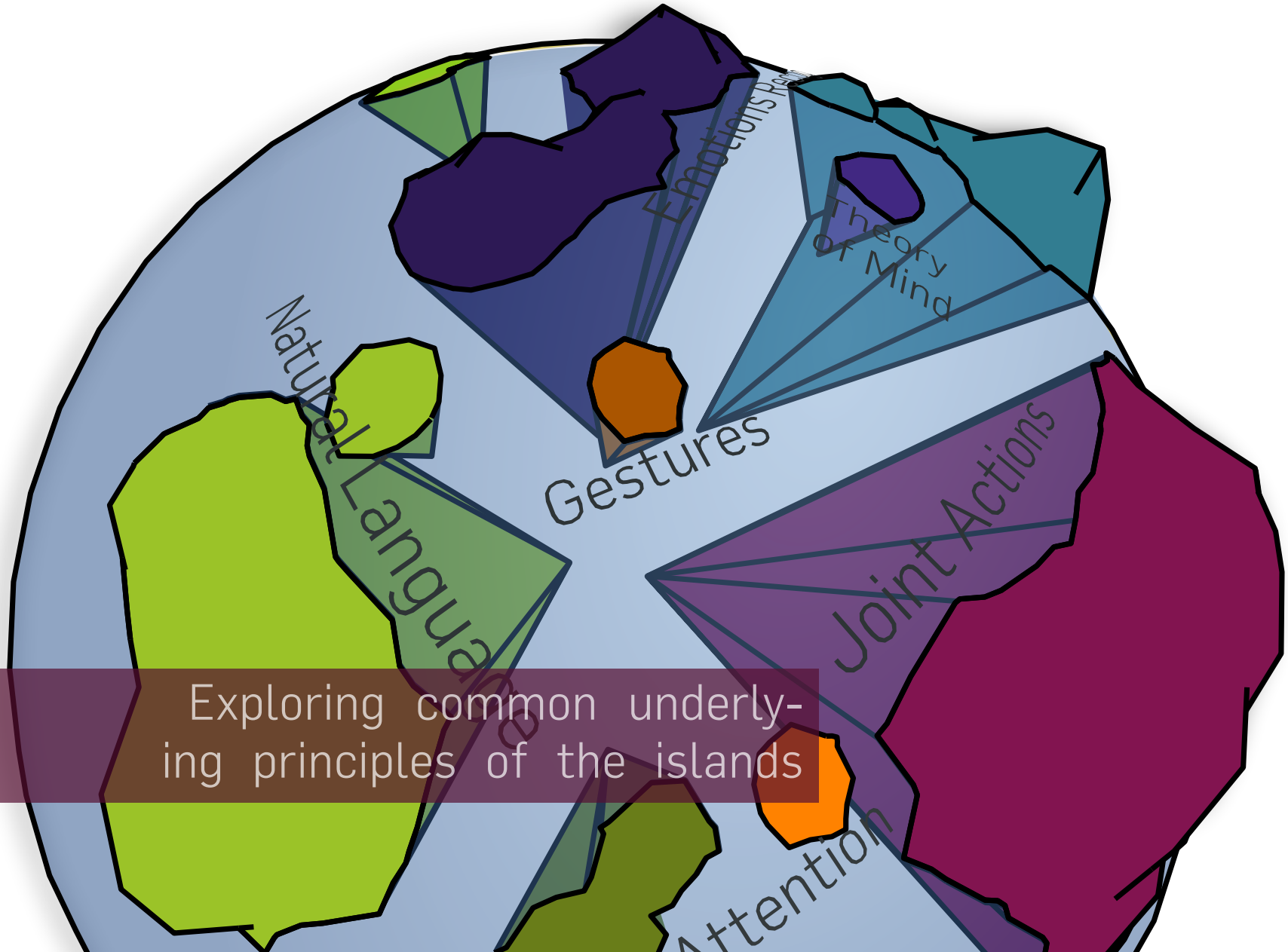
Support thinking about robot programming
in term of socio-cognitive skills



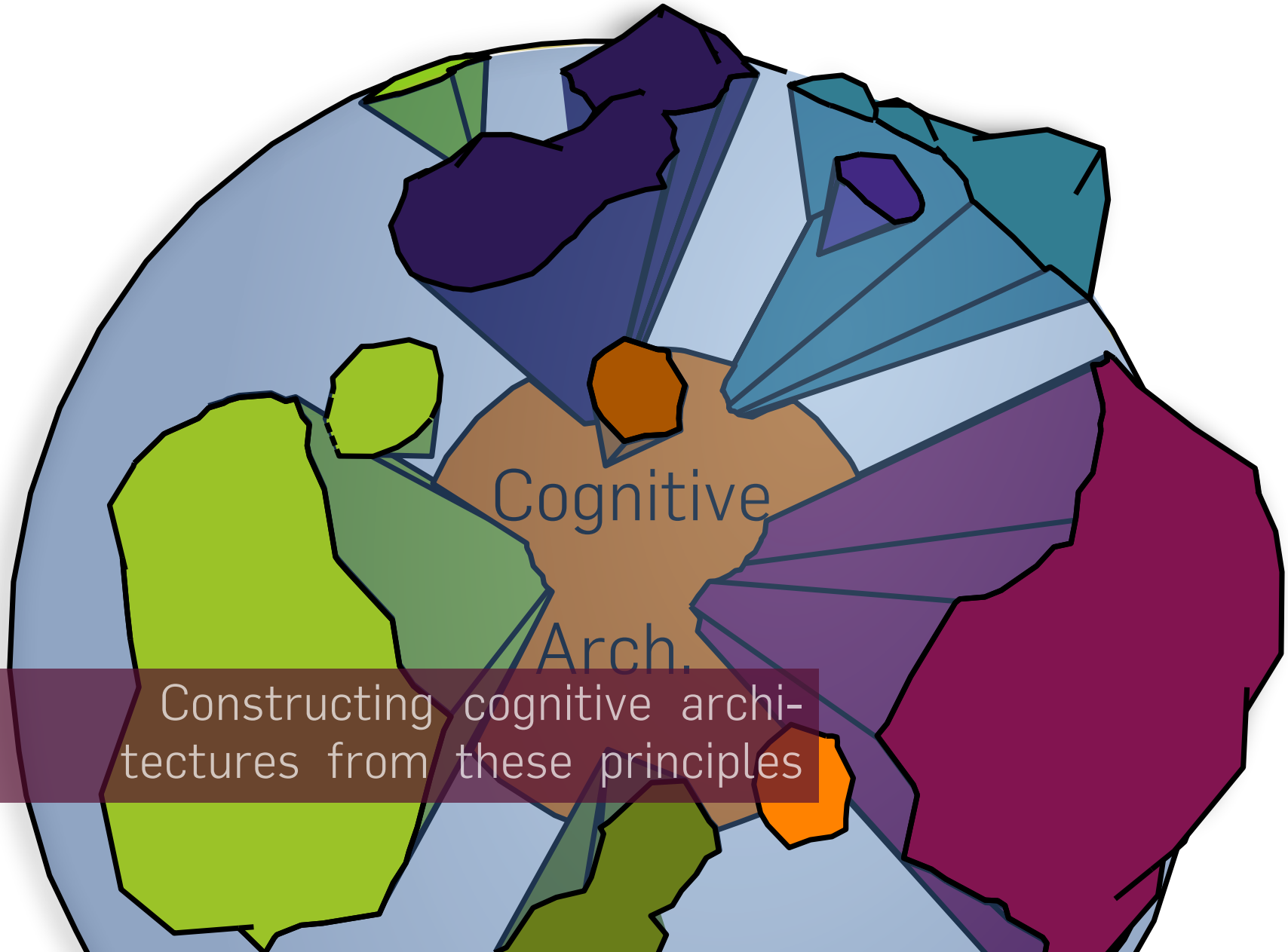
Islands of desired Cognitive Functionality



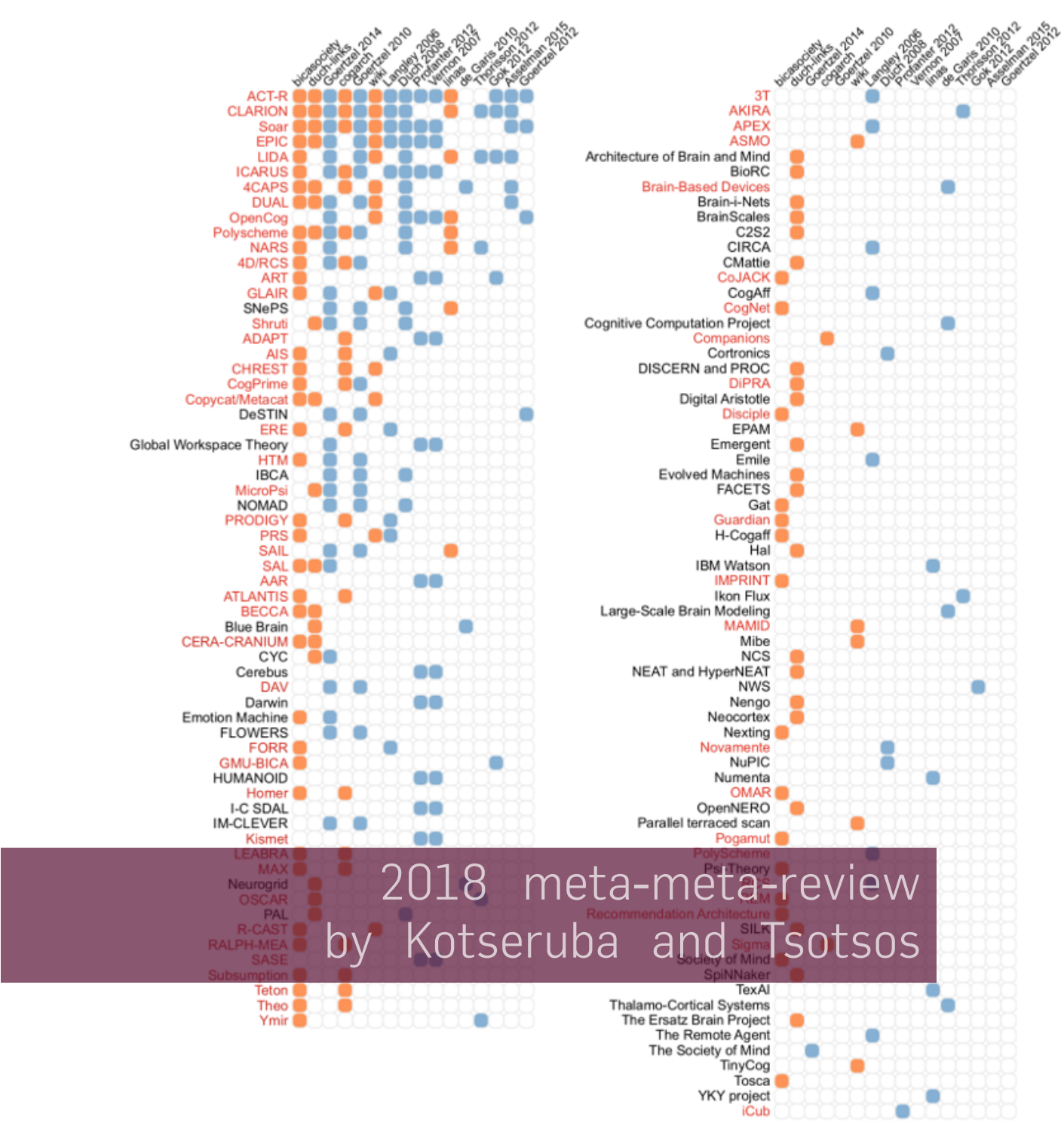
Linking islands together: the technical integration approach



Exploring common underlying principles of the islands



Constructing cognitive architectures from these principles

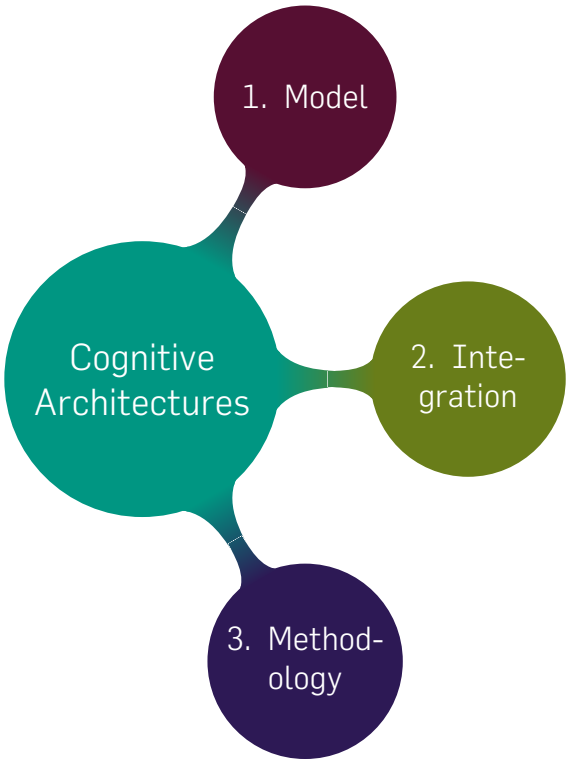


2018 meta-meta-review
by Kotseruba and Tsotsos

GOOGLE SCHOLAR:

ARCADIA
CECEP
CHRIS
Casimir
DAC
ECA
EmoCog
HiTECH
IOCA
Ikaros
MACSi
MUSICOG
MuRCA
PERSEED
RoboCog
SBR
SOIMA
The Meta-Cognitive Loop
robo-CAMAL
CARACaS
CHARISMA
CTRNN
CoSy
DSO
ERA
Haikonen CA
IMA
ITALK
STAR
ARDIS
CELTS
CRAM
Cognitive Symmetry Engine
Darwinian Neurodynamics
DIARC
EFAA
Feelix Growing
IAC
ISAC
JDE
MDB
Meta-AQUA
NEUCOGAR
Psikharpax
Roillo
SERA
SiMA
SPA
The Ouroboros Model
Joshua Blue
MLECOG
MoNAD
NIMBUS
ROSSI
SACA
SMRITI
Storm
VICAL
Xapagy

THREE INTERPRETATIONS OF COGNITIVE ARCHITECTURES



1. Models of Human Cognition

– Modelling (aspects of) human cognition

– Subsequent application to robots

2. Technical Integration

– Define required functionality of robots

– Implement algorithms (etc) necessary

3. Cognitive architectures as Methodology

– Formalising assumptions

– Integrating knowledge from multiple disciplines

– Iteratively updating architecture

1. MODELS OF HUMAN COGNITION

- Model some human cognitive phenomenon, or preferably set of phenomena (typically based on human behavioural data)
- Derive operating principles/algorithms that fulfil the requirements of the human behaviour
- Verify/validate resulting model by fitting to existing human data, or making predictions of human behaviour
- Apply model to robotics system

1. MODELS OF HUMAN COGNITION

- Model some human cognitive phenomenon, or preferably set of phenomena (typically based on human behavioural data)
- Derive operating principles/algorithms that fulfil the requirements of the human behaviour
- Verify/validate resulting model by fitting to existing human data, or making predictions of human behaviour
- Apply model to robotics system

bicasociety.org/cogarch/architectures.htm

Notable examples: **ACT-R** (Anderson and Lebiere), **SOAR** (Laird, Newell, Rosenbloom)

1. MODELS OF HUMAN COGNITION

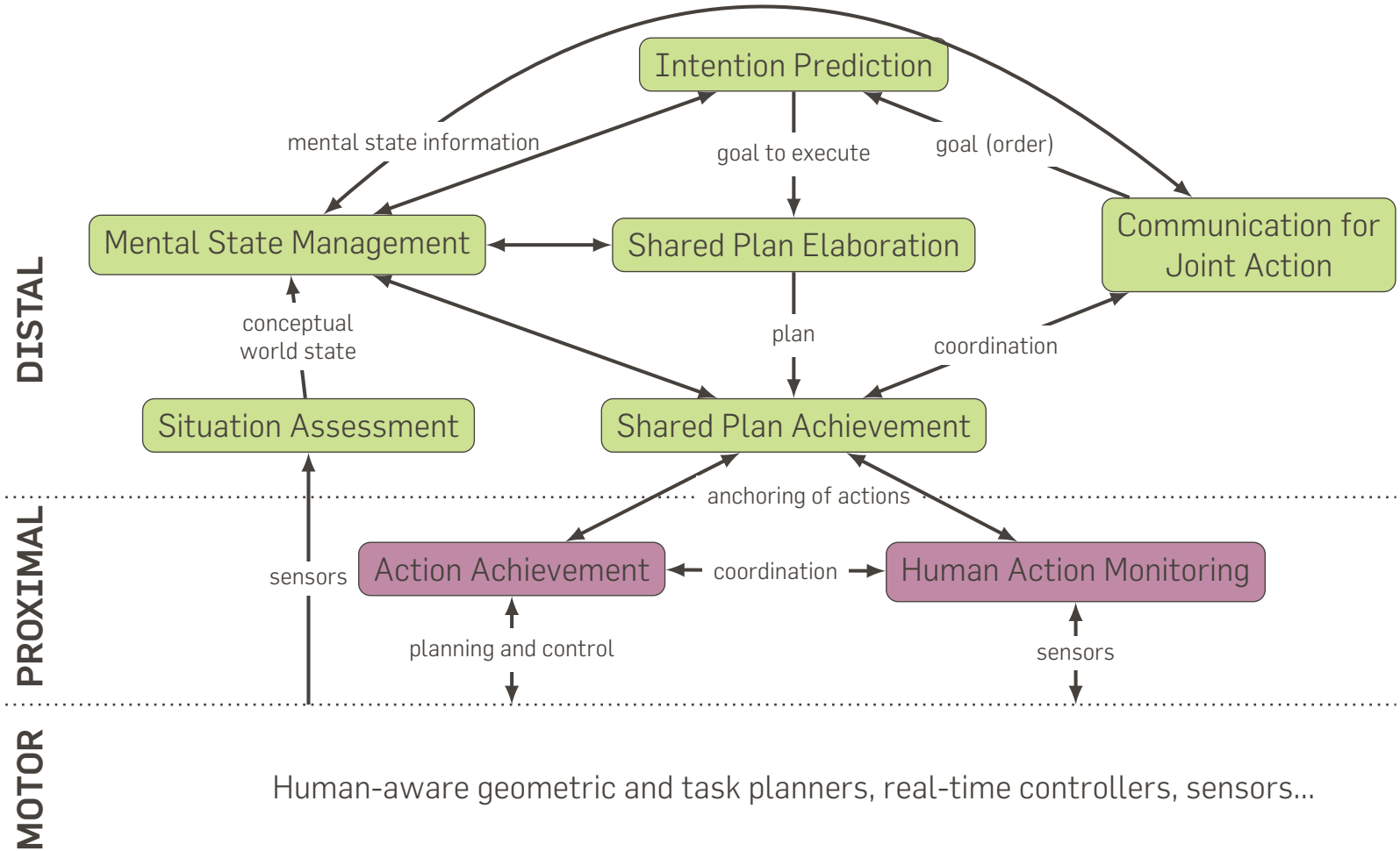
- Model some human cognitive phenomenon, or preferably set of phenomena (typically based on human behavioural data)
- Derive operating principles/algorithms that fulfil the requirements of the human behaviour
- Verify/validate resulting model by fitting to existing human data, or making predictions of human behaviour
- Apply model to robotics system

bicasociety.org/cogarch/architectures.htm

Notable examples: **ACT-R** (Anderson and Lebiere), **SOAR** (Laird, Newell, Rosenbloom)

ACT-R/E (Trafton) attempts to apply ACT-R to robotics.

ARCHITECTURES TO MODEL HUMAN COGNITION



[Devin, Alami **An implemented theory of mind to improve human-robot shared plans execution**, HRI, 2016]

2. TECHNICAL INTEGRATION

- Start with the application domain/problem to be solved
- Define set of algorithms required to fulfil task
- Implement architecture on robot
- Run experiment

3. COGNITIVE ARCHITECTURES AS METHODOLOGY

- Human-derived evidence (from perhaps multiple domains)
- Define set of principles and constraints
- Implement architecture on robot; generate predictions; run experiment(s)
- Feed back results into principles and constraints; repeat

THIS ALL LOOKS QUITE COMPLICATED

Useful to think in terms of surface functions, ie the cognitive functions visible and directly useful during the interaction.

THIS ALL LOOKS QUITE COMPLICATED

Useful to think in terms of surface functions, ie the cognitive functions visible and directly useful during the interaction.

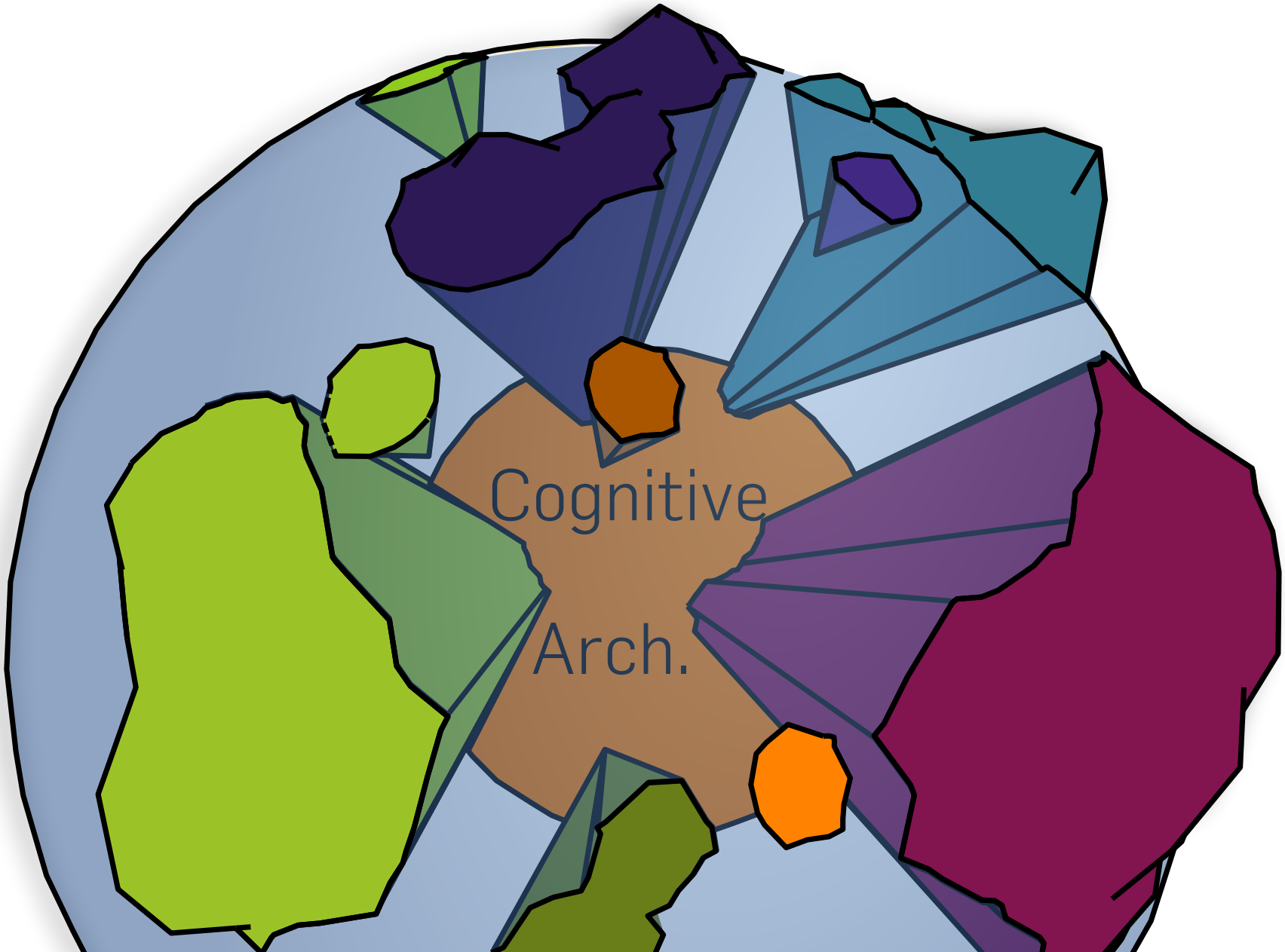
The underpinning mechanisms are generally hidden to the end-user.

THIS ALL LOOKS QUITE COMPLICATED

Useful to think in terms of surface functions, ie the cognitive functions visible and directly useful during the interaction.

The underpinning mechanisms are generally hidden to the end-user.

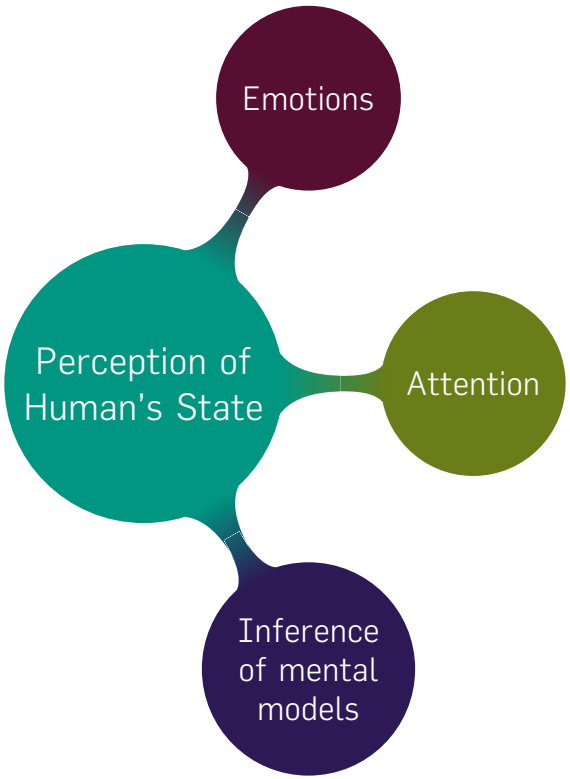
Note that discovering unifying cognitive principles is a key research problem.


Cognitive
Arch.

SURFACE FUNCTIONS FOR SOCIAL COGNITION

[Baxter, Lemaignan, Trafton **Workshop on Cognitive Architectures for Social HRI** – HRI 2016]

PERCEPTION OF HUMAN'S COGNITIVE STATE



Emotions and expressions

Recognition **facial, non-verbal acoustic, verbal features / sentient analysis**

Interpretation **empathy,...**

Attention

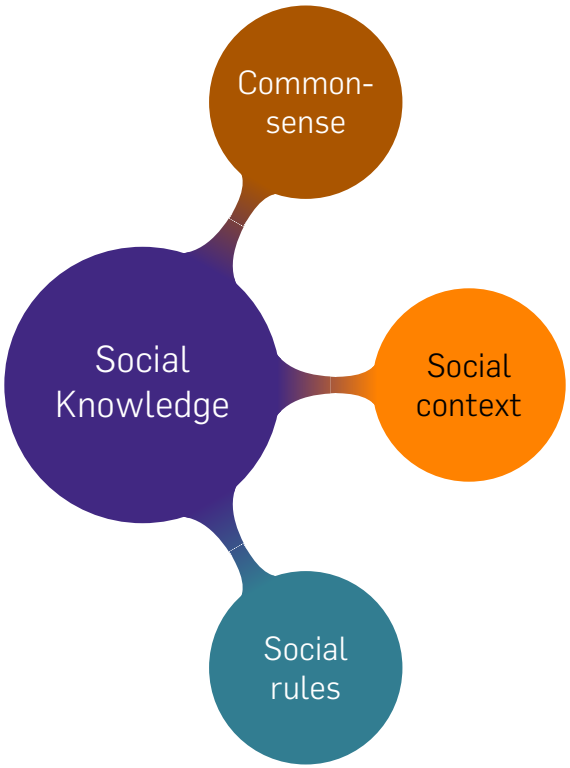
Visual attention
Joint attention

Mental Modelling

User modelling
(perceptual) theory of mind [2]

[Baxter, Lemaignan, Trafton **Workshop on Cognitive Architectures for Social HRI** – HRI 2016]

SOCIAL KNOWLEDGE



Common-sense

Hand-coded ontologies
Automated web-scrapping [3]

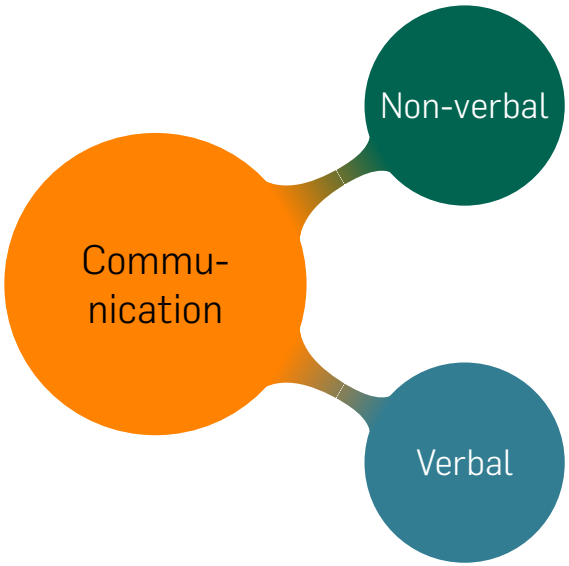
Social context

Social Rules

Proxemics
Social norms
Moral norms
Ethics

[Baxter, Lemaignan, Trafton **Workshop on Cognitive Architectures for Social HRI** – HRI 2016]

COMMUNICATION



Review [4]:

- Breaking the “simple commands only” barrier;
- Mixed initiative dialogue;
- Symbol grounding problem;
- Motor correlates
- ...

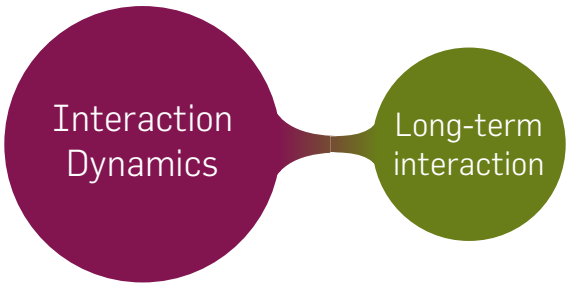
Non-verbal

Behavioural alignment

Verbal

Situated dialogue [5]

INTERACTION DYNAMICS

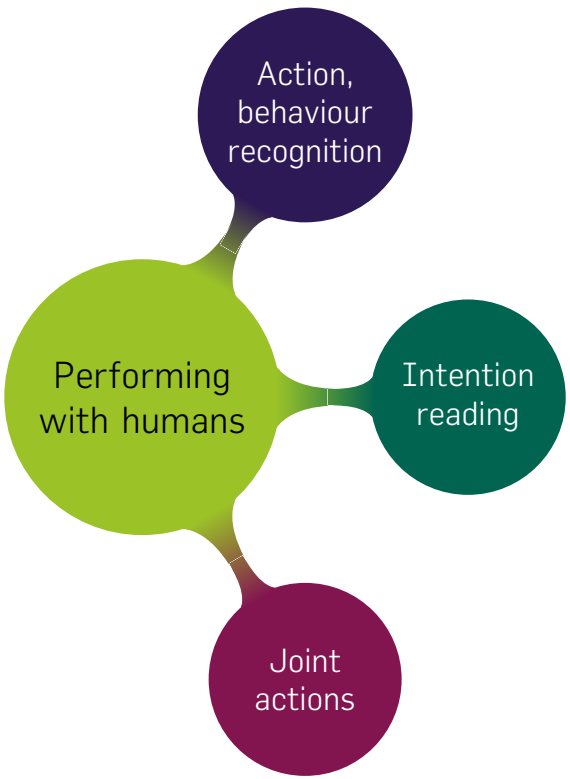


Dynamics Turn-taking
Mimicry
...

Long-term interaction
Managing post-novelty effect
Lifelong learning
Personalisation
...

[Baxter, Lemaignan, Trafton **Workshop on Cognitive Architectures for Social HRI** – HRI 2016]

PERFORMING WITH HUMANS



Action & Behaviour recognition

Hidden Markov Models

Intention reading

Recognition of conventional situation (“the hitchhiker”)

...

Joint action

[Baxter, Lemaignan, Trafton **Workshop on Cognitive Architectures for Social HRI** – HRI 2016]

Your turn, now!

ADDITIONAL BIBLIOGRAPHY


I. Kotseruba, J. K. Tsotsos
»A Review of 40 Years in Cognitive Architecture ResearchCore Cognitive Abilities and Practical Applications«
2018


S. Lemaignan, P. Dillenbourg
»Mutual Modelling: Inspiration for the Next Steps«
2015


M. Tenorth, D. Nyga, M. Beetz
»Understanding and Executing Instructions for Everyday Manipulation Tasks from the World Wide Web«
2010


N. Mavridis
»A review of verbal and non-verbal human–robot interactive communication«
2015


G.-J. M. Kruijff ***et al.***
»Situated dialogue processing for human-robot interaction«
2010

ADDITIONAL BIBLIOGRAPHY


P. Baxter, J. de Greeff, T. Belpaeme
»Cognitive Architecture for HRI: Towards behavioural alignment«
2013


A. Morse, J. de Greeff, T. Belpaeme, A. Cangelosi
»Epigenetic Robotics Architecture«
2010


A. Rueckert, D. Kappel, D. Tanneberg, D. Pecevski, J. Peters
»Recurrent Spiking Networks Solve Planning Tasks«
2016


D. Vernon
»Artificial Cognitive Systems«
2013

That's all for today, folks!

Questions:
severin.lemaignan@brl.ac.uk

Slides:
github.com/severin-lemaignan/lecture-hri-architectures

MORE ON TASK PLANNING

TASK PLANNING: ACTIONS

An action has **pre-conditions** and **post-conditions** (or *effects*).

```
Action <action name>
{
    preconditions {...};
    effects{...};
    cost{<cost_function_name>};
    duration{<duration_function_name>};
}
```

TASK PLANNING: ACTIONS

An action has **pre-conditions** and **post-conditions** (or *effects*).

```
Action <action name>
{
    preconditions {...};
    effects{...};
    cost{<cost_function_name>};
    duration{<duration_function_name>};
}

Action open_fridge
{
    preconditions {facing_fridge AND fridge_door.closed};
    effects {facing_fridge AND fridge_door.open};
}
```

EXAMPLE: SHAKEY THE ROBOT

Shakey the robot (Stanford, 1968), using the **STRIPS** planner

Go ...

Go to object bx

GOTOB(bx)

Preconditions: TYPE(bx,OBJECT),(∃rx)[INROOM(bx,rx) ∧ INROOM(ROBOT,rx)]
Deletions: AT(ROBOT,$1,$2), NEXTTO(ROBOT,$1)
Additions: *NEXTTO(ROBOT,bx)

Go to door dx.

GOTOD(dx)

Preconditions: TYPE(dx,DOOR),(∃rx)(∃ry)[INROOM(ROBOT,rx) ∧ CONNECTS(dx,rx,ry)]
Deletions: AT(ROBOT,$1,$2), NEXTTO(ROBOT,$1)
Additions: *NEXTTO(ROBOT,dx)

Go to coordinate location (x,y).

GOTOL(x,y)

Preconditions: (∃rx)[INROOM(ROBOT,rx) ∧ LOCINROOM(x,y,rx)]
Deletions: AT(ROBOT,$1,$2), NEXTTO(ROBOT,$1)
Additions: *AT(ROBOT,x,y)

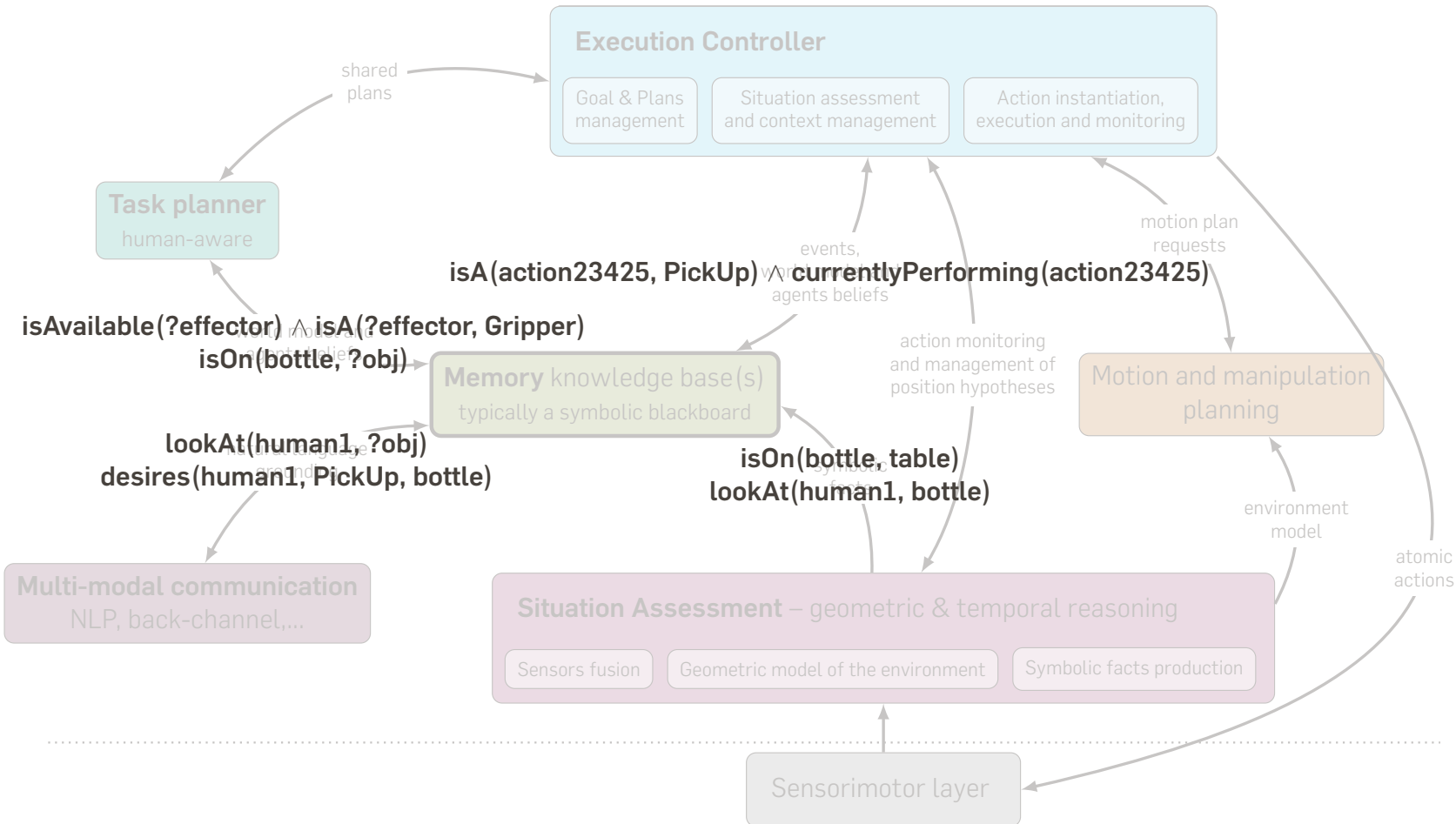
Go through door dx into room rx.

GOTHRUDR(dx,rx)

Preconditions: TYPE(dx,DOOR), STATUS(dx,OPEN), TYPE(rx,ROOM),
NEXTTO(ROBOT,dx) (∃rx)[INROOM(ROBOT,ry) ∧ CONNECTS(dx,ry,rx)]
Deletions: AT(ROBOT,$1,$2), NEXTTO(ROBOT$1), INROOM(ROBOT,$1)
Additions: *INROOM(ROBOT,rx)

SYMBOLIC VS SUB-SYMBOLIC ARCHITECTURES

A SYMBOLIC CONTROL ARCHITECTURE



HIGHLIGHTS

- **well-established theoretical grounds** in formal logics
- **explicit**, **high-level semantics** (good for NLP for instance!)
- good engineering properties (modular, loose coupling,...)
- $\Rightarrow$ **close to the human level** (both for the users and the developers)

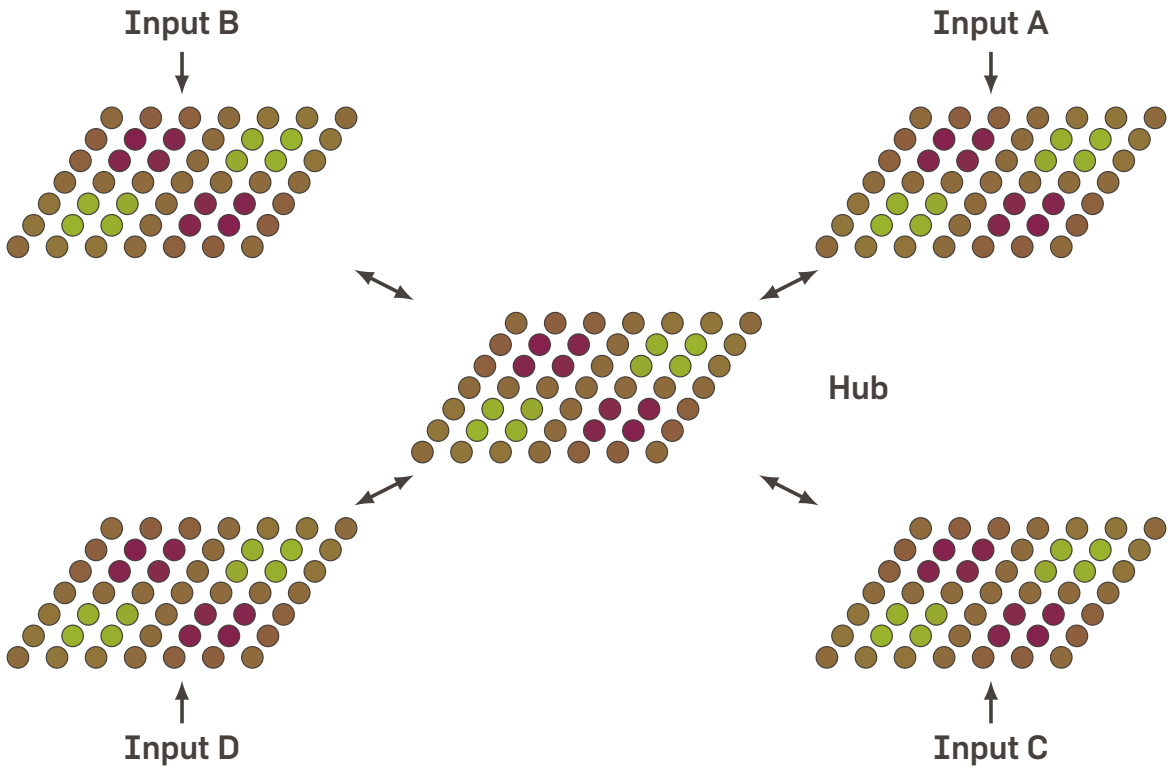
HIGHLIGHTS

- **well-established theoretical grounds** in formal logics
- **explicit**, **high-level semantics** (good for NLP for instance!)
- good engineering properties (modular, loose coupling,...)
- $\Rightarrow$ **close to the human level** (both for the users and the developers)

- **very successful** so far
- **fundamentally top-down** approach $\Rightarrow$ somewhat difficult to match with the constructive approach of developmental psychology
- **learning** is an **after-thought**
- representation of **uncertainty not natural**
- **hard to deal with time** (chronicles, fluents,...?)

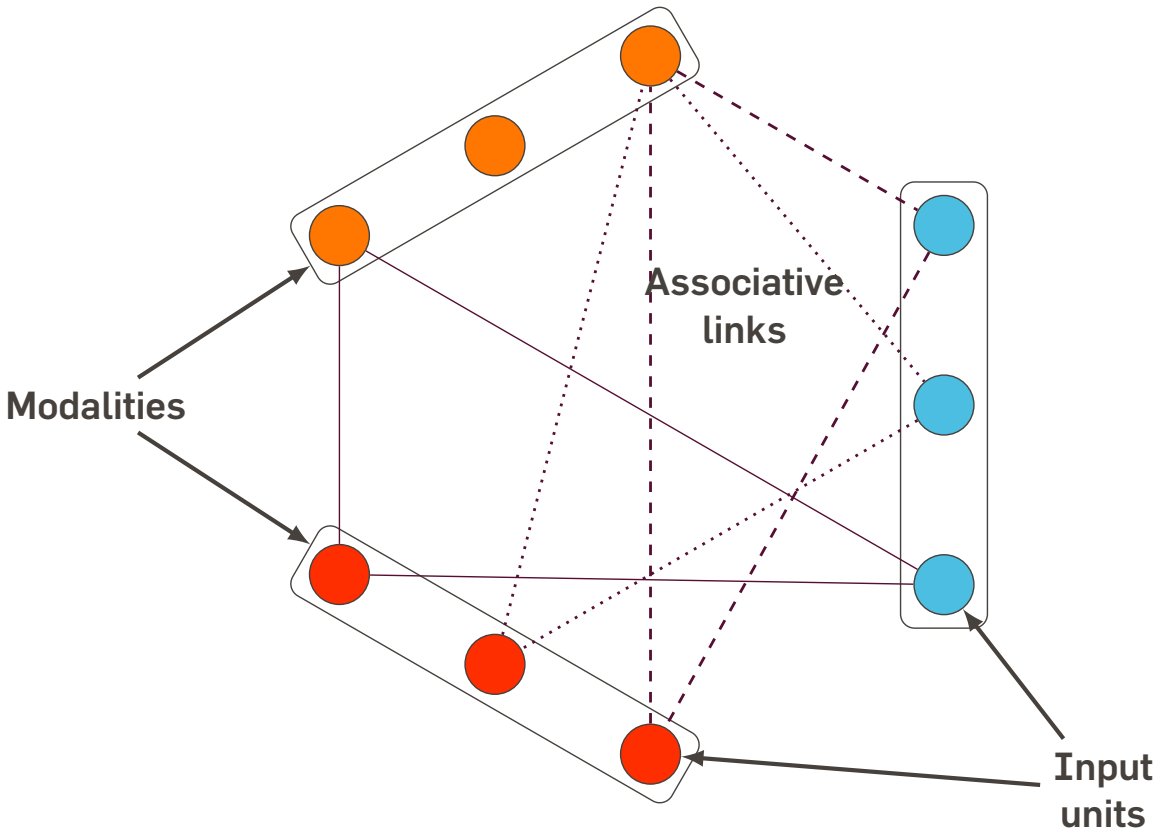
SUB-SYMBOLIC ARCHITECTURES

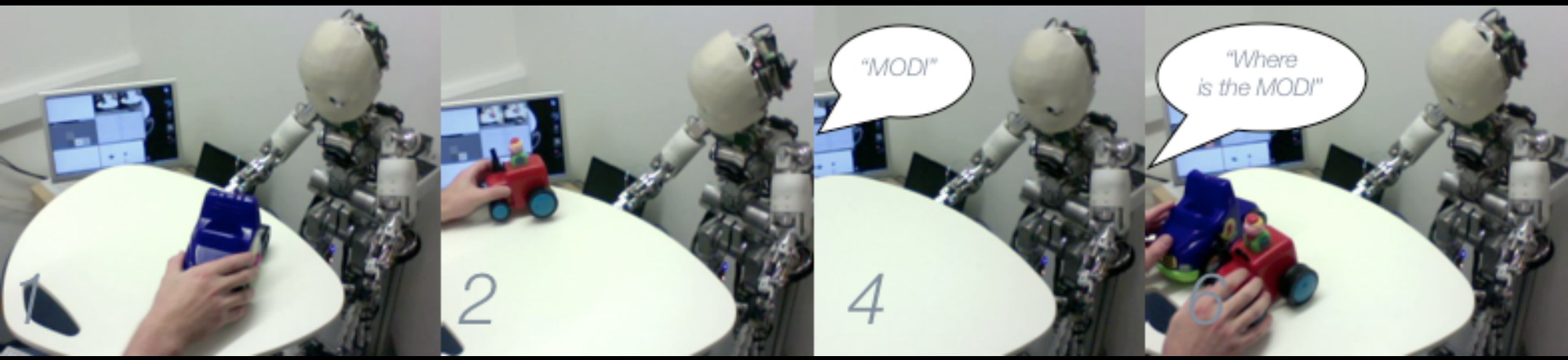
- **(Hierarchical) Self-Organizing Maps** [2]



SUB-SYMBOLIC ARCHITECTURES

- **(Hierarchical) Self-Organizing Maps** [2]
- **Memory network** [1]




1
2
"MODI"
4
"Where
is the MODI"
6

HIGHLIGHTS

- often **general principled approaches**
- **learning** is intrinsic and fundamental
- **emergent features** (eg concept priming)

HIGHLIGHTS

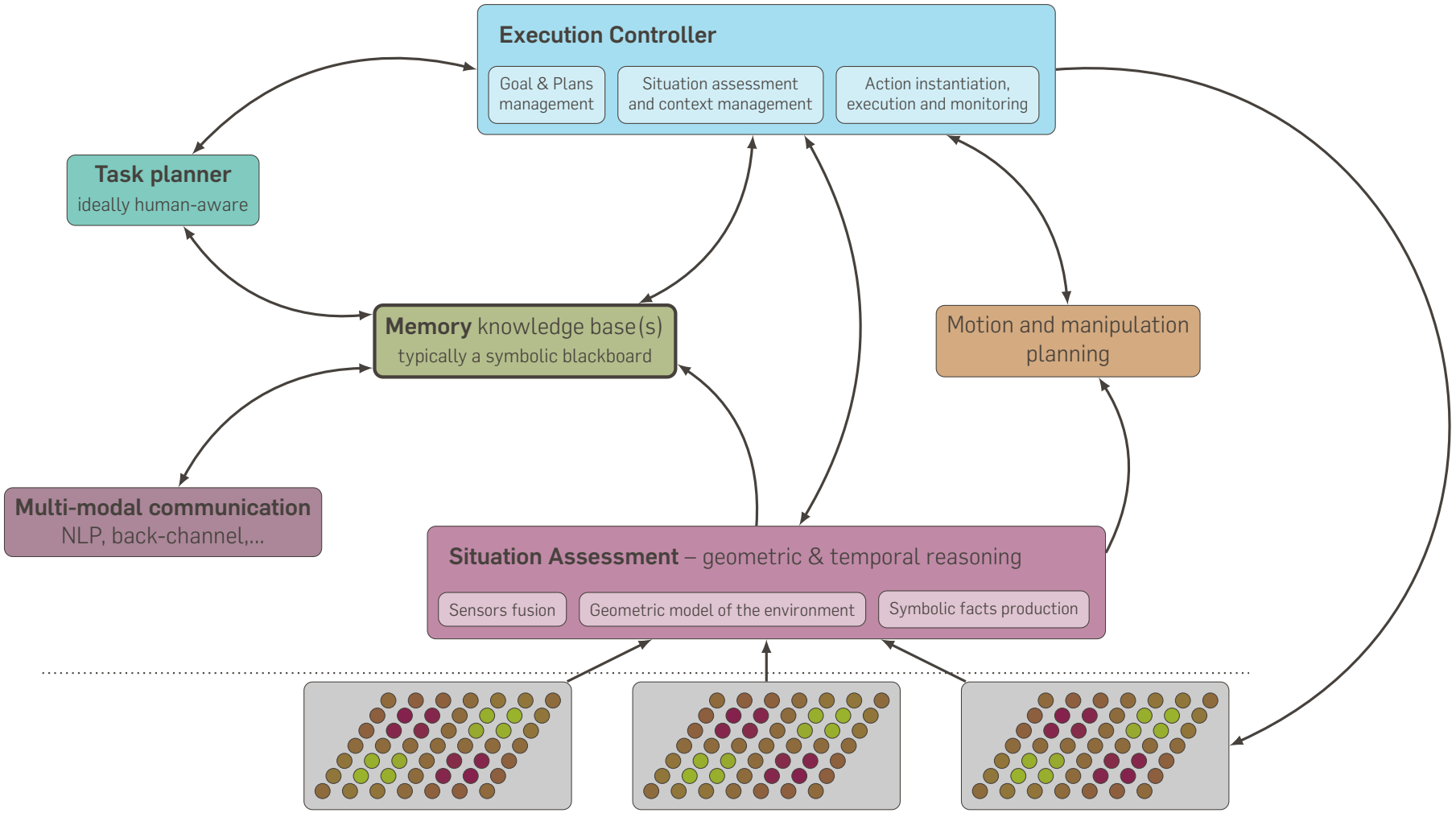
- often **general principled approaches**
- **learning** is intrinsic and fundamental
- **emergent features** (eg concept priming)

- more directly comparable to human cognitive appartus
- applications to high-level socio-cognitive tasks only emerging
- **still hard to deal with time!**

HYBRID ARCHITECTURES

What about...

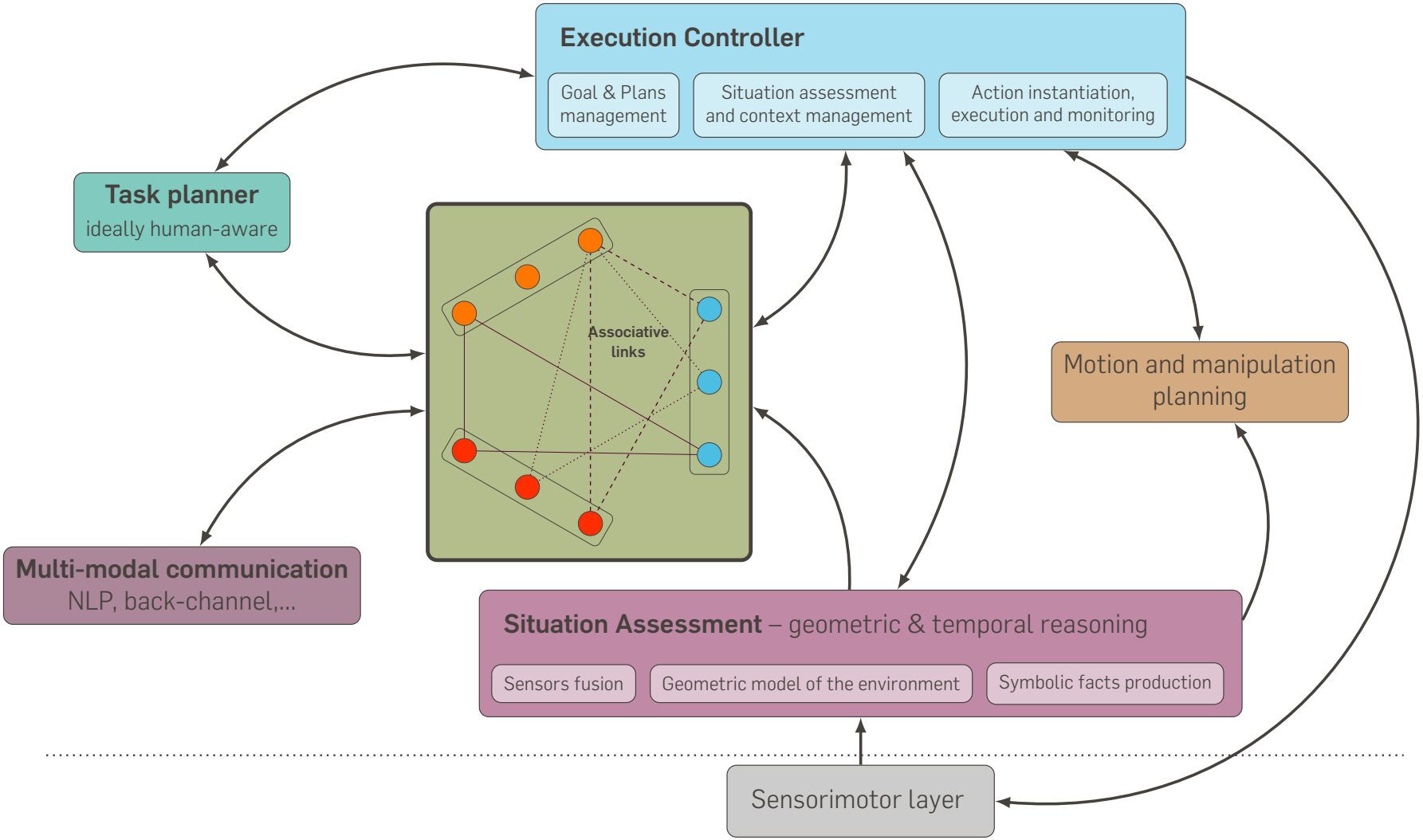
HYBRID ARCHITECTURES



HYBRID ARCHITECTURES

or...

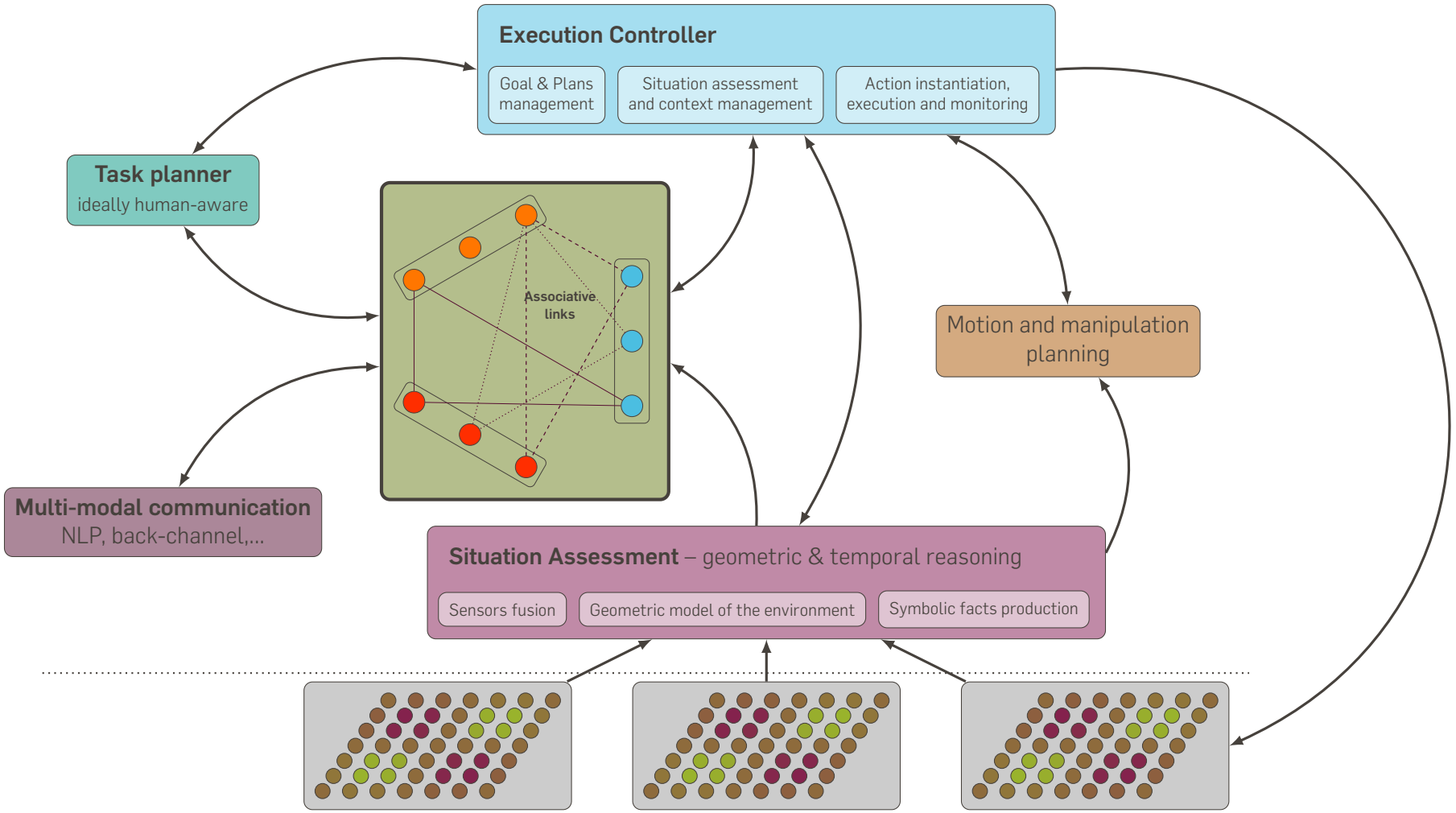
HYBRID ARCHITECTURES



HYBRID ARCHITECTURES

or even...

HYBRID ARCHITECTURES



HYBRID ARCHITECTURES: EPISTEMIC ARGUMENTS

The **computational functionalism** argument [4]:

»the physical realization of the computational model is inconsequential to the model«

HYBRID ARCHITECTURES: EPISTEMIC ARGUMENTS

The **computational functionalism** argument [4]:

> *»the physical realization of the computational model is inconsequential to the model«*

Embodied cognition takes the opposite view.

Some see the symbolic/sub-symbolic questions as an instantiation of this argument.

CROSS-DISCIPLINARY “IMPEDANCE MATCHING”

The choice of a computational model impacts how **legible** the architecture is to other disciplines:

- **psycholinguistics** will be more comfortable manipulating symbolic architectures;
- **neurosciences** may be more comfortable with certain sub-symbolic approaches (but not all of them: deep learning is hardly a biologically inspired neural technique)

$\Rightarrow$ **“epistemic impedance”**

CROSS-DISCIPLINARY "IMPEDANCE MATCHING"

The choice of a computational model impacts how **legible** the architecture is to other disciplines:

- **psycholinguistics** will be more comfortable manipulating symbolic architectures;
- **neurosciences** may be more comfortable with certain sub-symbolic approaches (but not all of them: deep learning is hardly a biologically inspired neural technique)

$\Rightarrow$ **"epistemic impedance"**

Some disciplines are however mostly **agnostic**
cognitive psy., developmental psy.,...

CROSS-DISCIPLINARY "IMPEDANCE MATCHING"

The choice of a computational model impacts how **legible** the architecture is to other disciplines:

- **psycholinguistics** will be more comfortable manipulating symbolic architectures;
- **neurosciences** may be more comfortable with certain sub-symbolic approaches (but not all of them: deep learning is hardly a biologically inspired neural technique)

⇒ **"epistemic impedance"**

Also relevant is the nature of **predictions** you want to generate or test (eg, reaction time vs language competencies).

A FEW INTERNAL COGNITIVE FUNCTIONS

	Symbolic	**Sub-symbolic**
Concepts	taxonomies, external knowledge integration	conceptualization
Memory	facts storage	associative memory, priming
Planning	Plenty (prototypically: HTN)	recurrent networks [3]
...		

SIDE-TO-SIDE

Symbolic

- **well-established theoretical grounds** in formal logics
- **explicit**, **high-level semantics**
- good engineering properties (modular, loose coupling,...)
- ⇒ **close to the human level** (both for the users and the developers)
- **very successful** so far
- **fundamentally top-down** approach ⇒ somewhat difficult to match with the constructive approach of developmental psychology
- **learning** is an **after-thought**
- representation of **uncertainty not natural**
- **hard to deal with time** (chronicles, fluents,...?)

Sub-symbolic

- often **principled approaches**
- **learning** is intrinsic and fundamental
- **interesting features** come for free (eg concept priming)
- principled approach brings **predictive power**
- applications to high-level socio-cognitive tasks only emerging
- **still hard to deal with time!**