



This presentation is released under the terms of the
Creative Commons Attribution-Share Alike license.

You are free to reuse it and modify it as much as you want as long as:

- (1) you mention Séverin Lemaignan as being the original author,
- (2) you re-share your presentation under the same terms.

You can download the sources of this presentation here:

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

b
rl

UWE
Bristol

University
of the
West of
England



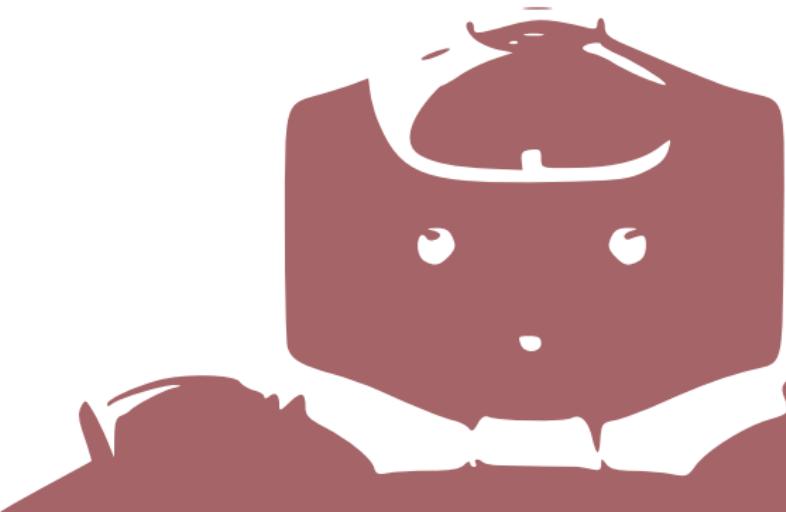
University of
BRISTOL

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:

- o a map with the important landmarks like **fridge**
- o the following modules:

speech_synthesis

open_fridge

pick_up_beer

detect_cat

follow_path_to

plan_to

detect_mum

process_verbal_commands

Can you draw a control architecture that achieves just that?

Control paradigms

oooooooooooooooooooo

From an architecture perspective

oooooo

A glimpse at cognitive architectures

oooooooooooooooooooo

IN THIS LECTURE

- Control paradigms

Control paradigms

oooooooooooooooooooo

From an architecture perspective

oooooo

A glimpse at cognitive architectures

oooooooooooooooooooo

IN THIS LECTURE

- Control paradigms
- One architecture instantiation

Control paradigms

oooooooooooooooooooo

From an architecture perspective

oooooo

A glimpse at cognitive architectures

oooooooooooooooooooo

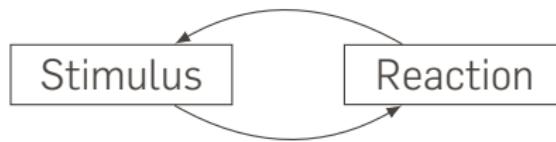
IN THIS LECTURE

- Control paradigms
- One architecture instantiation
- Cognitive architectures

CONTROL PARADIGMS

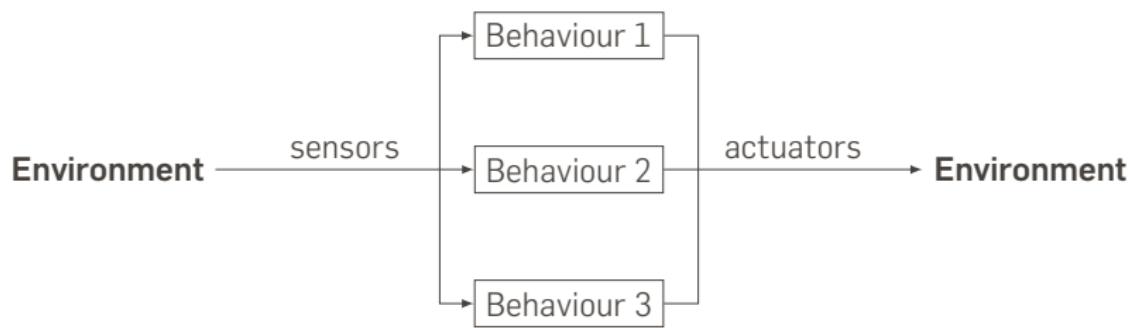
BEHAVIOURAL (OR REACTIVE) CONTROL

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

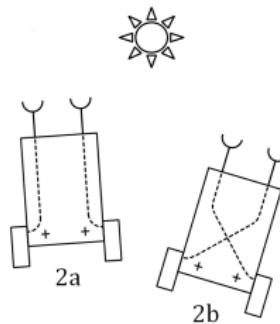
Behaviours are small programs that read sensors and control actuators. Each behaviour does **one simple thing**; it typically has access to all sensors/actuators.



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

EXTREME CASE: BRAITENBERG MACHINES

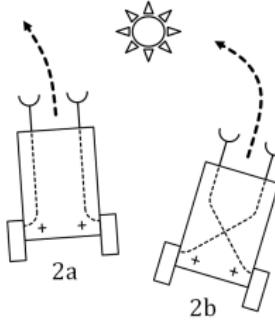
- Motion directly controlled by sensors (typically photocells)
- Yet the resulting behaviour may appear complex or even intelligent
- Can you guess the behaviours of vehicles 2a and 2b?



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?

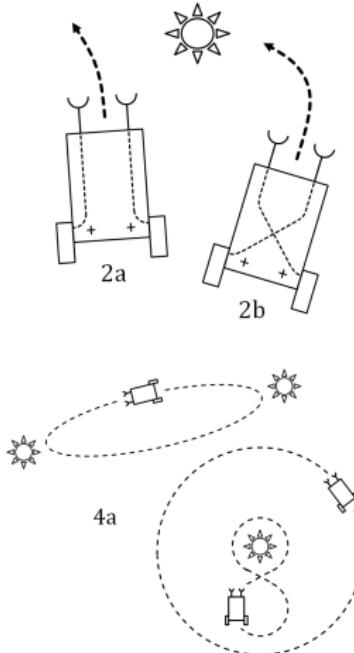


Source: Wikipedia

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 robot**

Sensor: IR communication

Behaviour: follow another robot

- **Avoid obstacle**

Sensors: bumpers

Behaviour: move away from the wall

- **Wander**

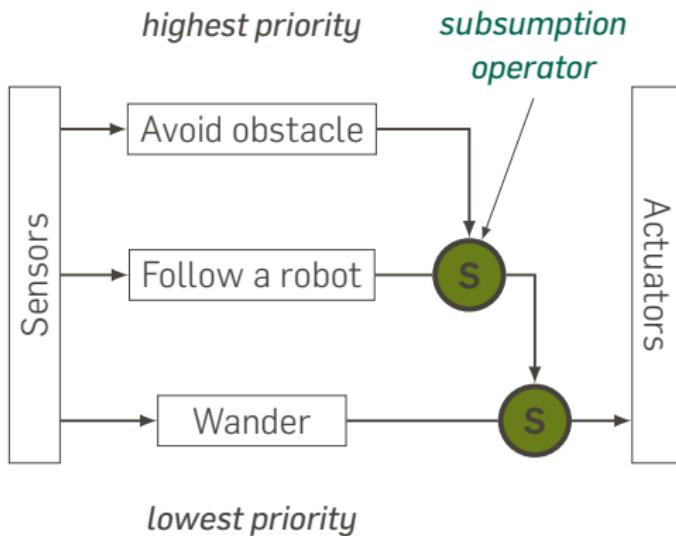
Sensors: encoders

Behaviour: move forward and turn

Which behaviour should have the highest priority? the lowest?

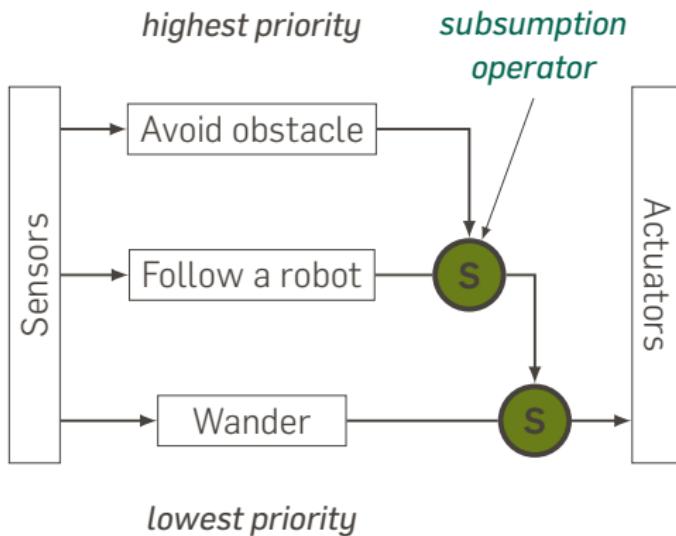
Source: *example borrowed from Rice University ENGI128*

COMBINING BEHAVIOURS: EXAMPLE



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

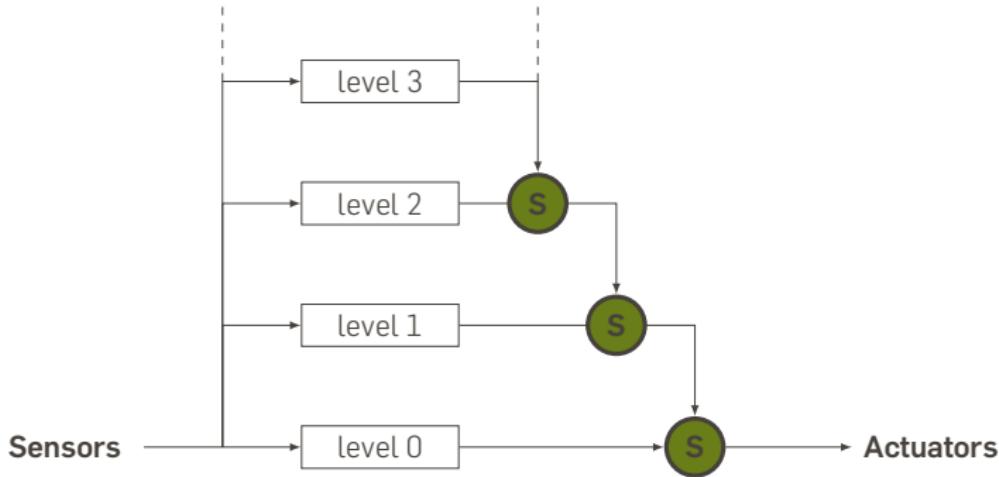
COMBINING BEHAVIOURS: EXAMPLE



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

→ **subsumption architecture**

SUBSUMPTION ARCHITECTURE



Control paradigms

oooooooo●ooooooooooooooo

From an architecture perspective

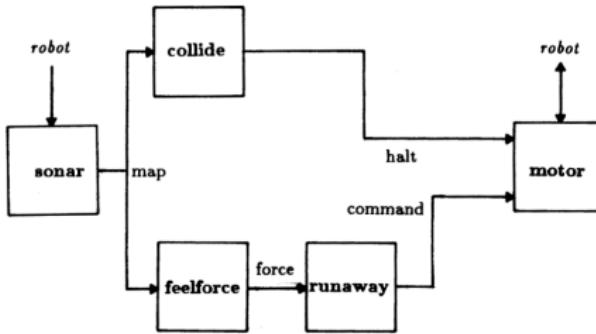
oooooo

A glimpse at cognitive architectures

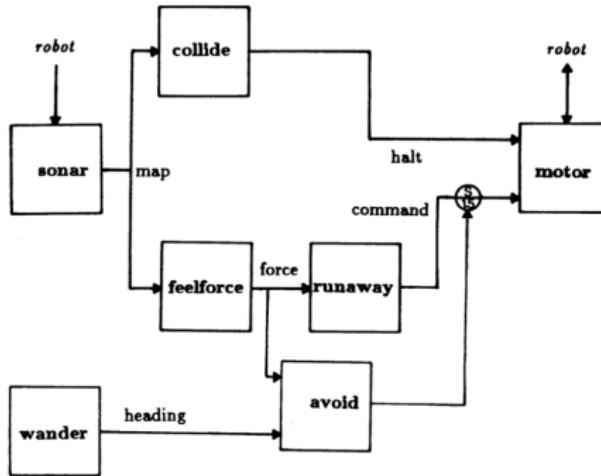
ooooooooooooooooooooooo

EXAMPLE: THE MIT GENGHIS ROBOT

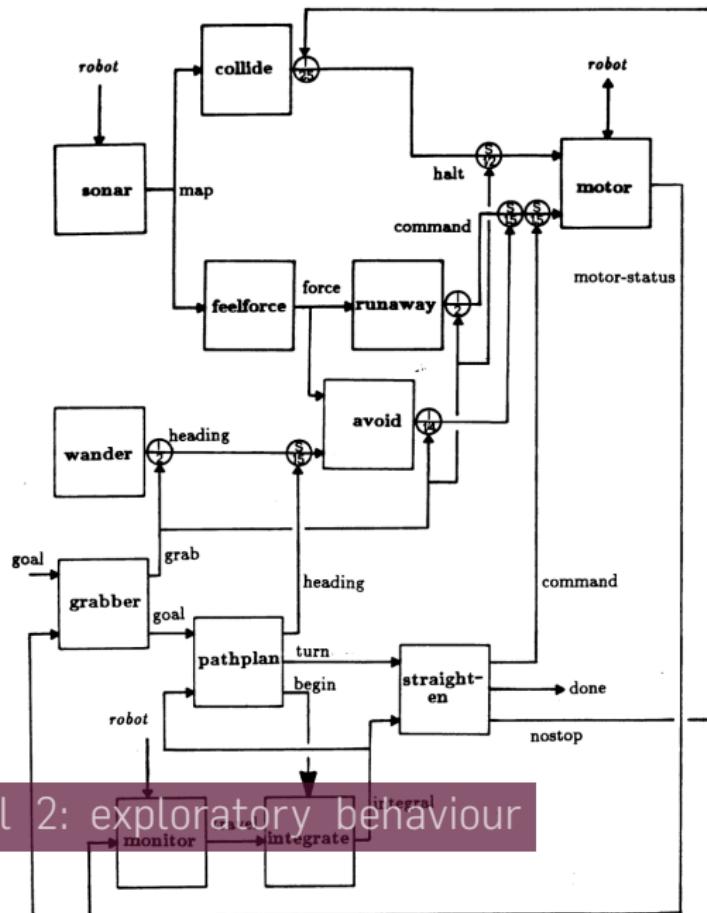




Level 0: obstacle avoidance



Level 1: wander around aimlessly



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

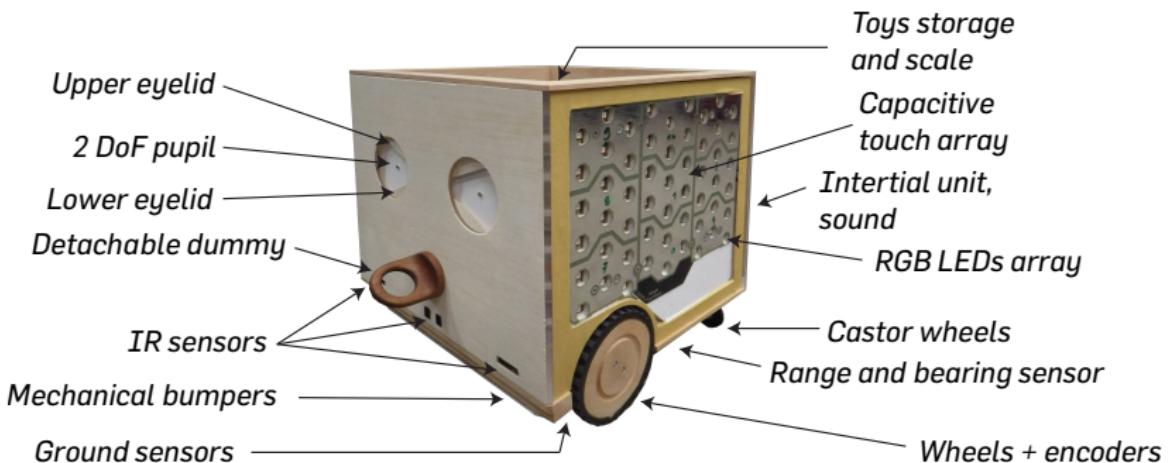
- **Incremental development**
- By definition **modular**
- Effective to react to events → well suited to **dynamic environments**

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.glance_at_touches()

    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

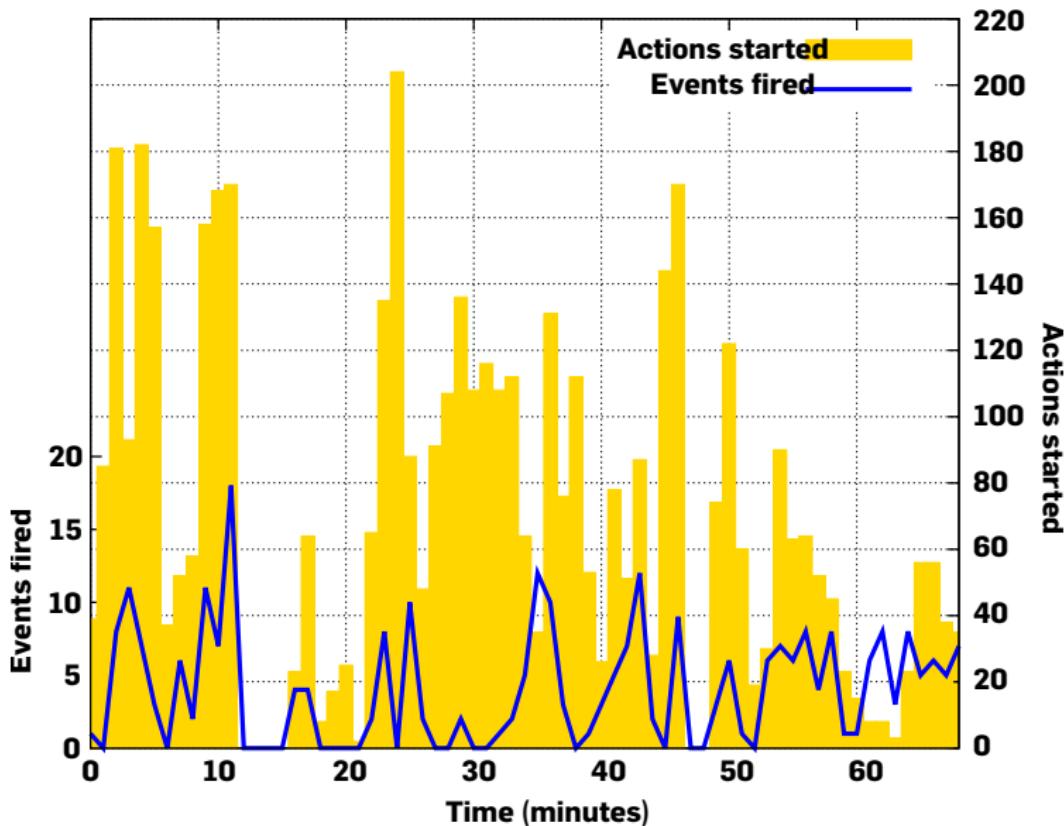
Event-oriented programming is a possible way of implementing a behavioural control paradigm:

```
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()
```



Control paradigms

oooooooooooo●oooooooo

From an architecture perspective

ooooooo

A glimpse at cognitive architectures

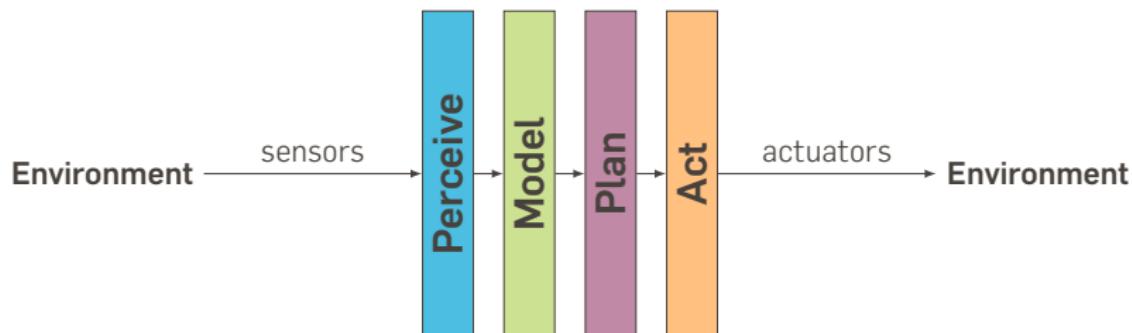
oooooooooooooooooooooooo



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 constraints to *decompose actions* into *primitive operators*.

TASK PLANNING

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

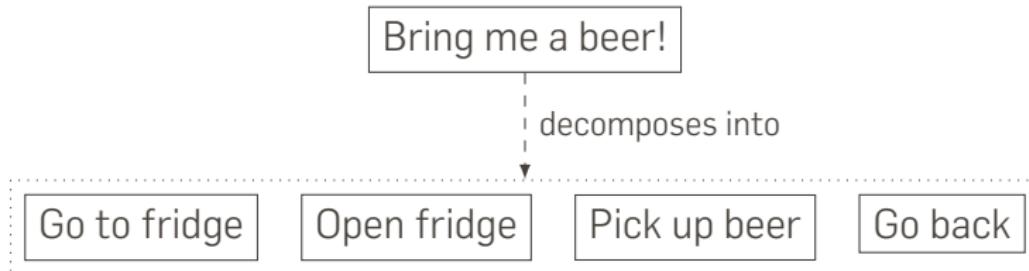
A standard approach relies on **Hierarchical task networks (HTN)**: uses partial-order constraints to *decompose actions* into *primitive operators*.

Bring me a beer!

TASK PLANNING

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

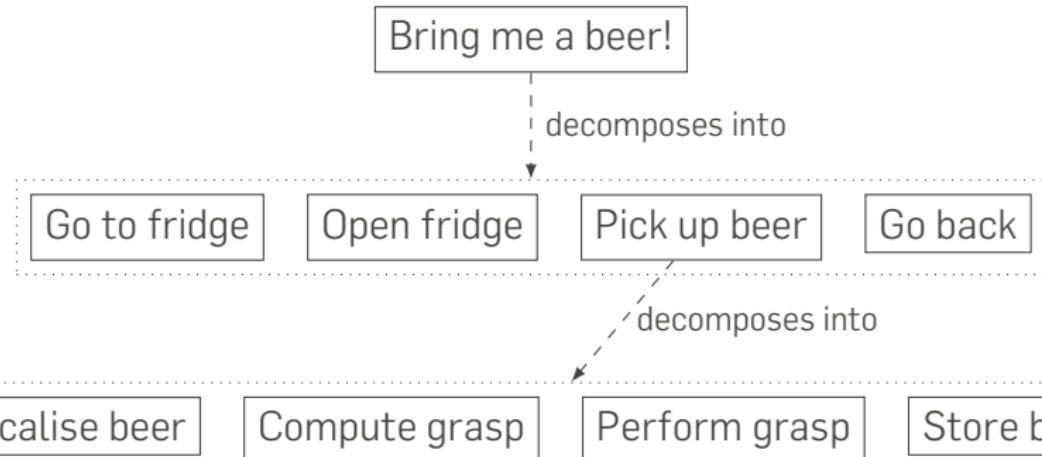
A standard approach relies on **Hierarchical task networks (HTN)**: uses partial-order constraints to *decompose actions* into *primitive operators*.



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 constraints 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?

Control paradigms

oooooooooooooooooooo●ooo

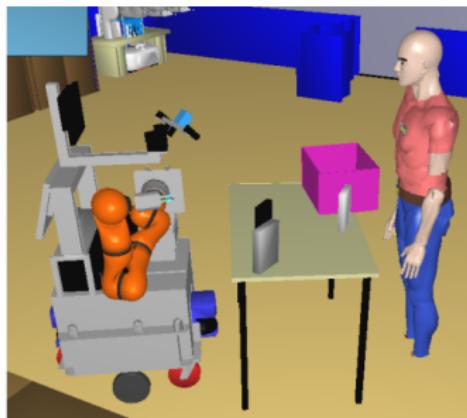
From an architecture perspective

oooooo

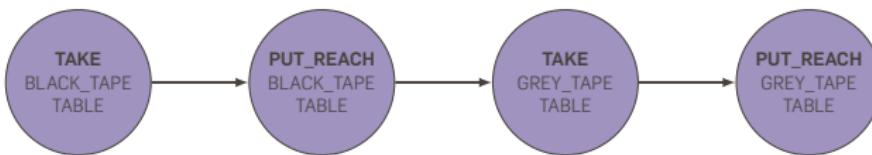
A glimpse at cognitive architectures

ooooooooooooooooooooooo

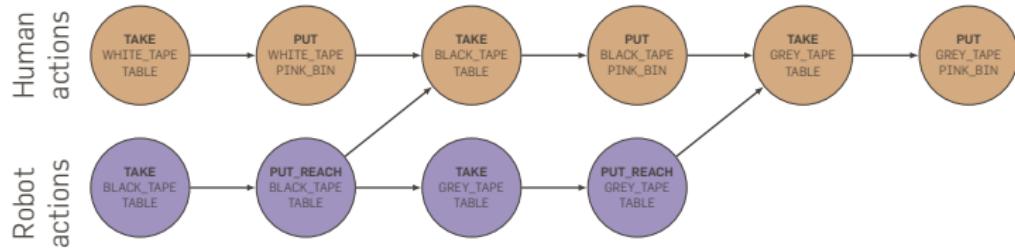
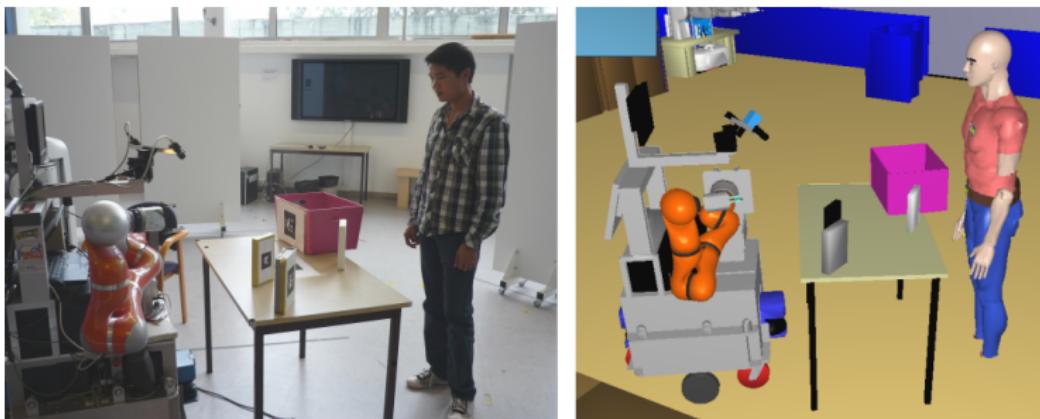
TASK PLANNING



Robot
actions



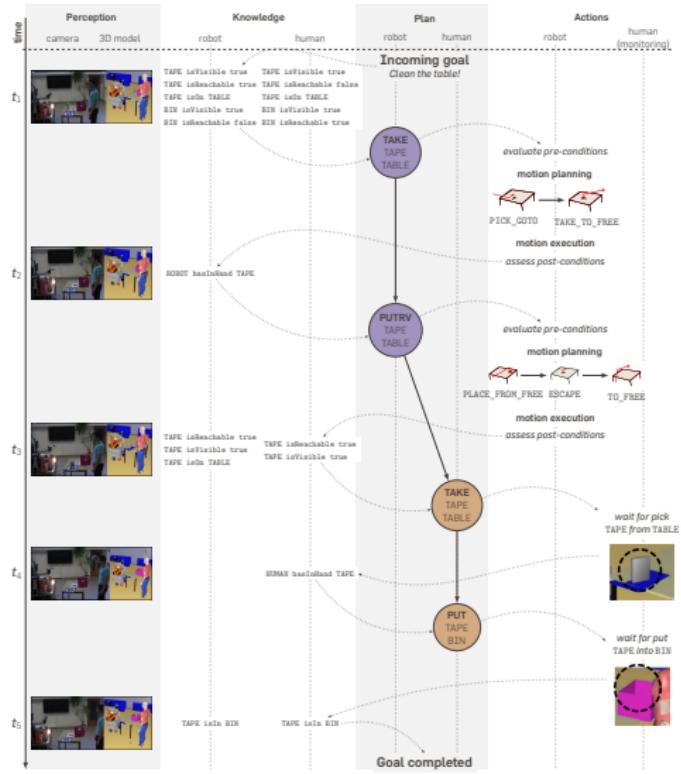
TASK PLANNING



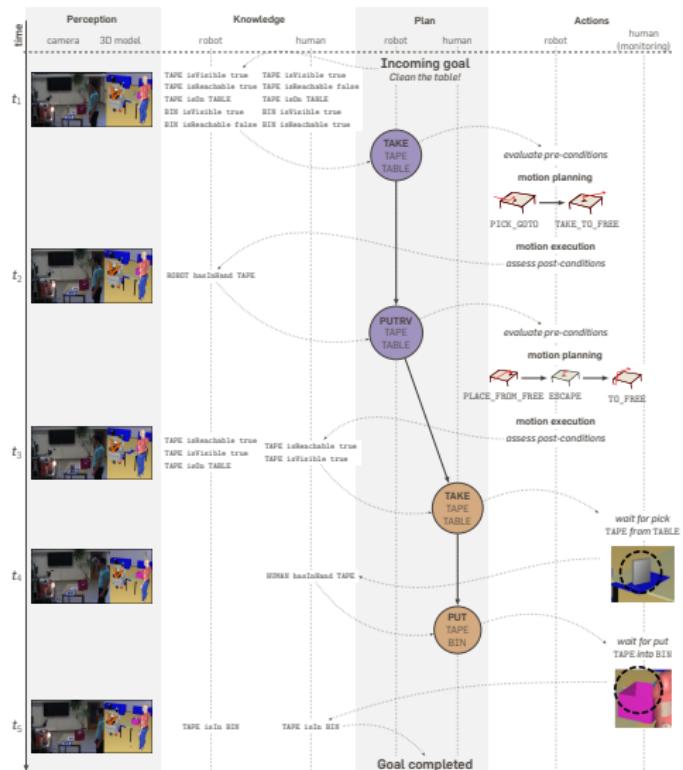


LAAS-CNRS

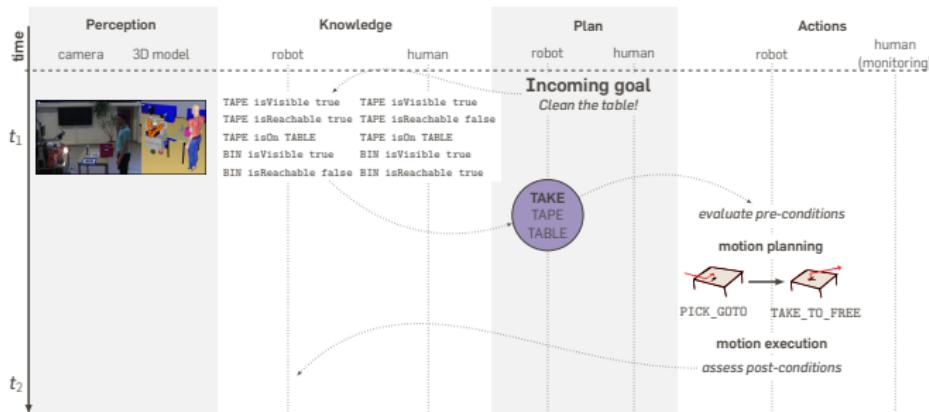
FULL SOCIAL & AUTONOMOUS INTERACTION: ONE EXAMPLE



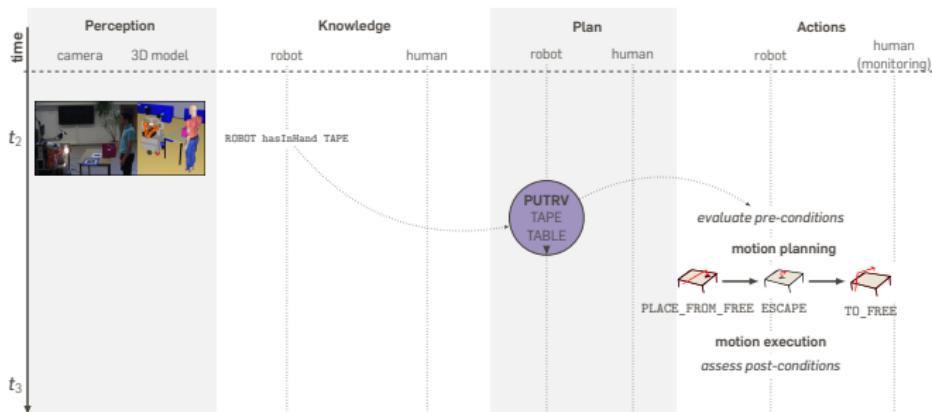
"CLEANING THE TABLE"...



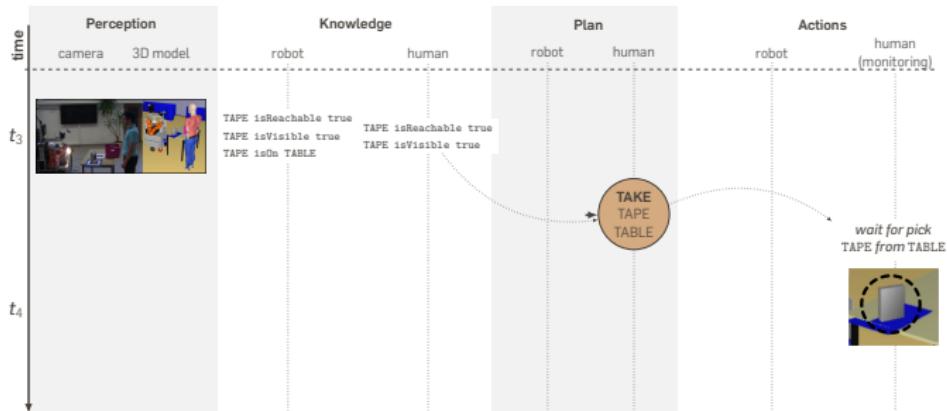
"CLEANING THE TABLE"...



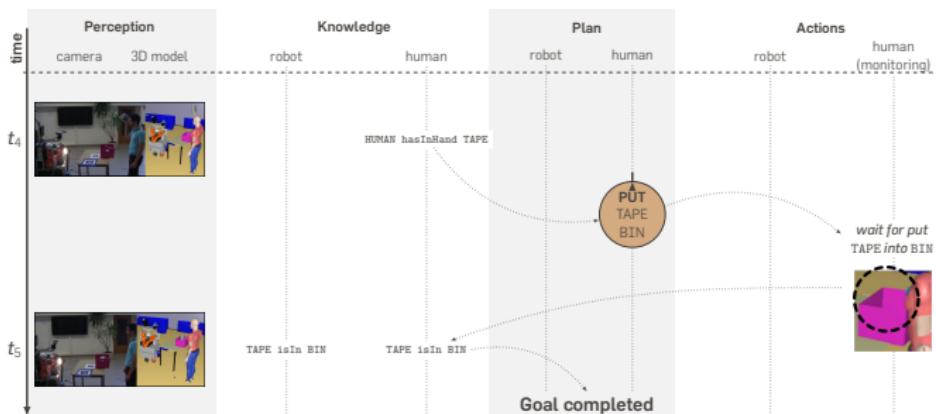
"CLEANING THE TABLE"...



"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

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

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

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

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

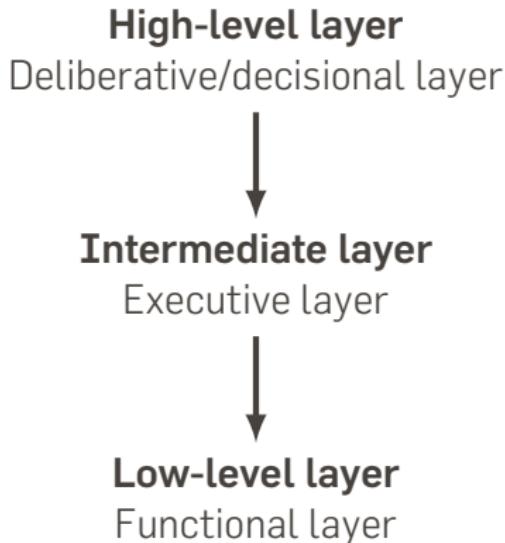
Time scale of a few seconds

- **High level control**

What is the plan for moving these boxes out of the room?

Discrete problems, long time scale (minutes)

LAYERED ARCHITECTURES



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

Control paradigms

oooooooooooooooooooo

From an architecture perspective

oooo●o

A glimpse at cognitive architectures

oooooooooooooooooooo

DELIBERATIVE ARCHITECTURE FOR INTERACTION

One example: the LAAS deliberative architecture for interaction



Control paradigms

oooooooooooooooooooo

From an architecture perspective

ooooo●

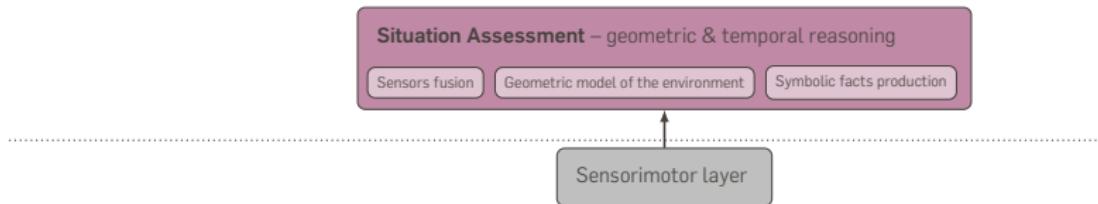
A glimpse at cognitive architectures

oooooooooooooooooooo

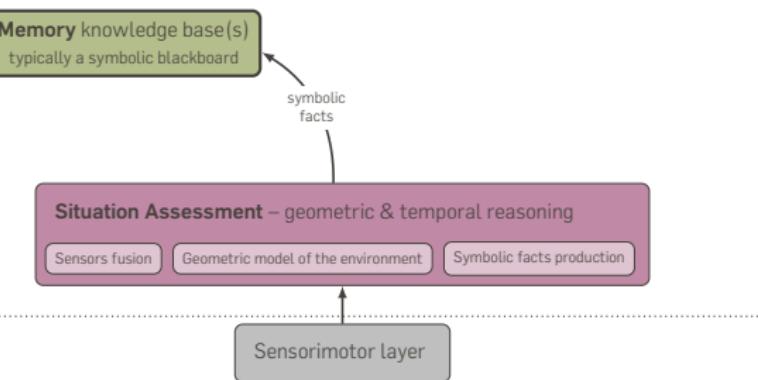
A SYMBOLIC CONTROL ARCHITECTURE

Sensorimotor layer

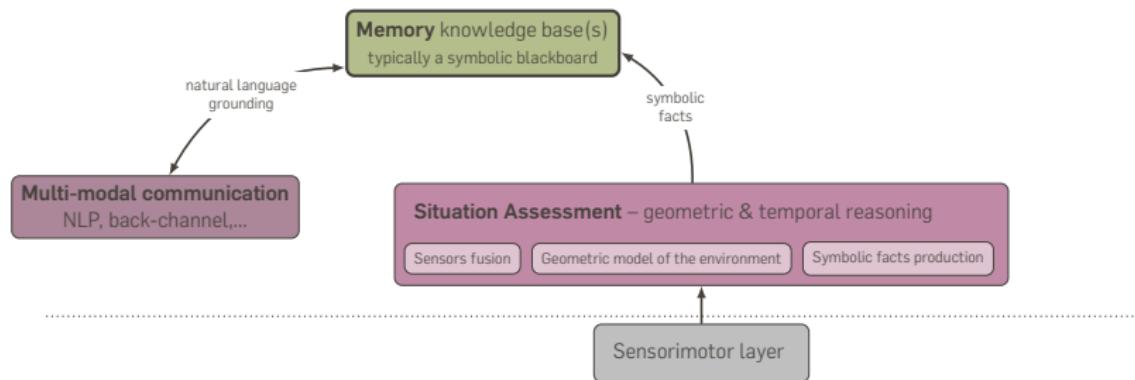
A SYMBOLIC CONTROL ARCHITECTURE



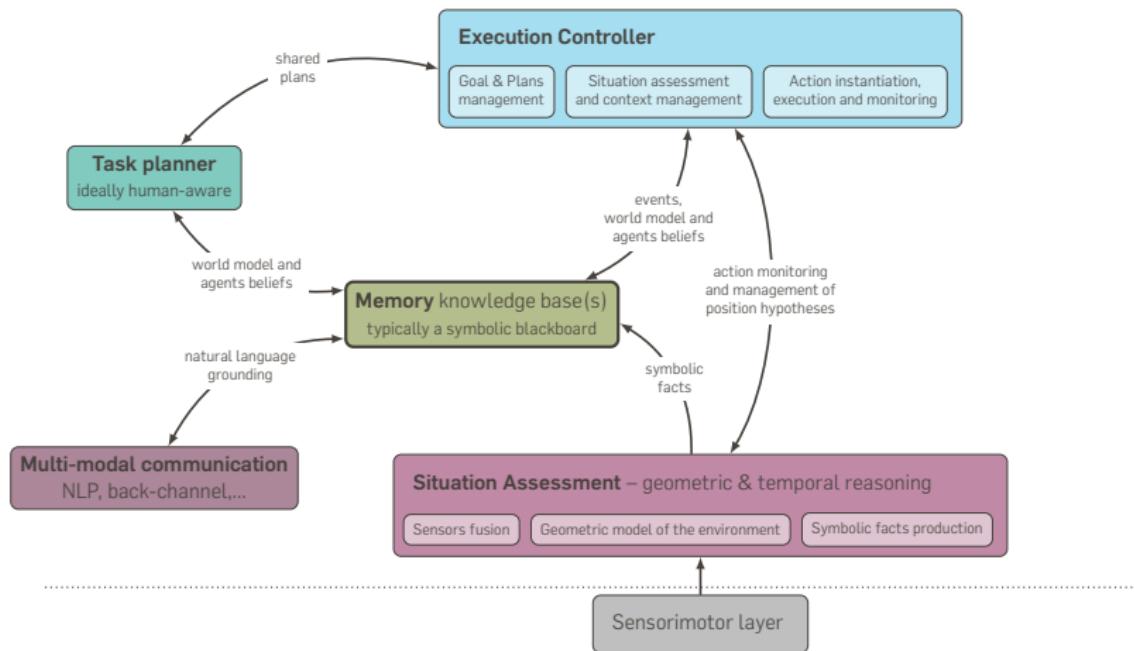
A SYMBOLIC CONTROL ARCHITECTURE



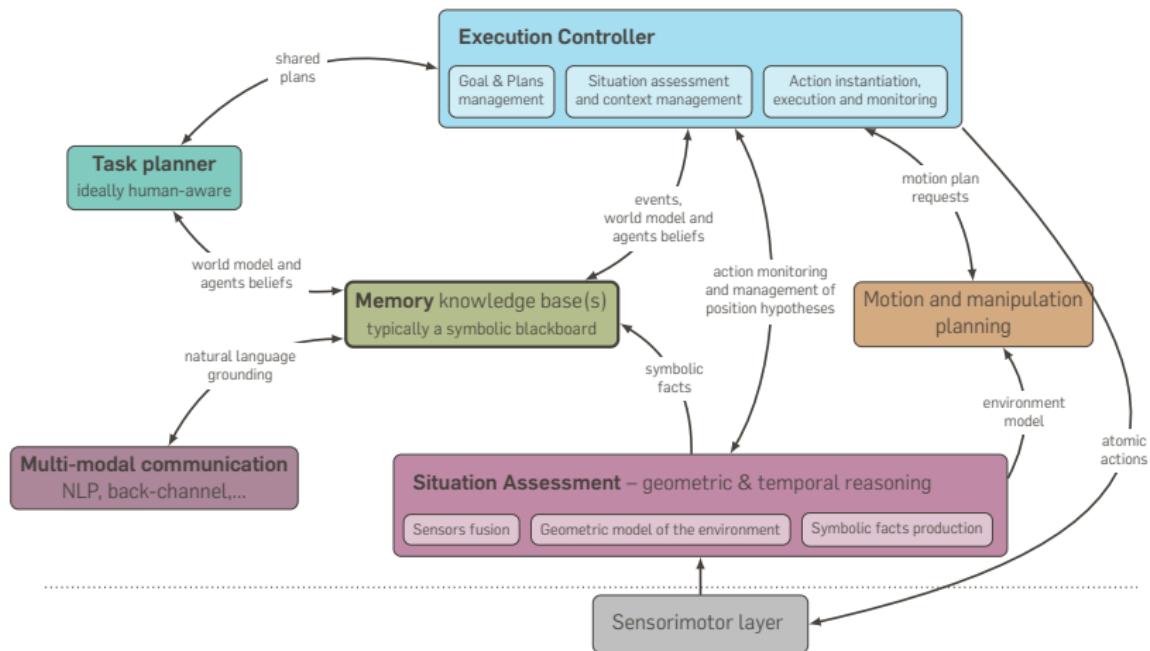
A SYMBOLIC CONTROL ARCHITECTURE



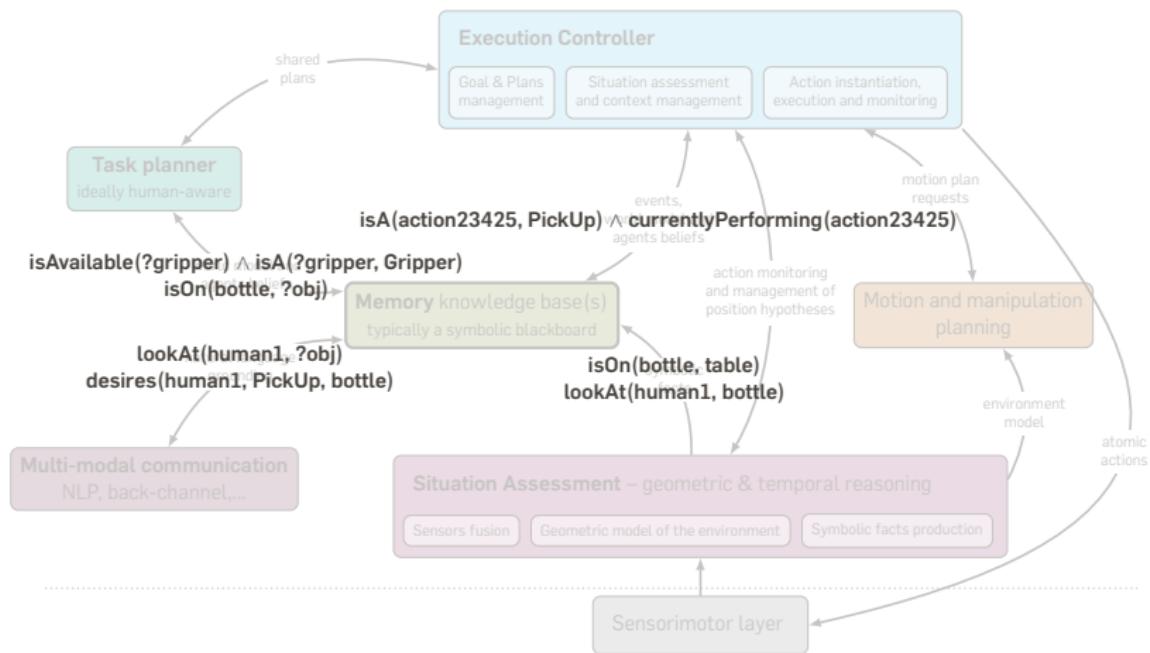
A SYMBOLIC CONTROL ARCHITECTURE



A SYMBOLIC CONTROL ARCHITECTURE



A SYMBOLIC CONTROL ARCHITECTURE



A GLIMPSE AT COGNITIVE ARCHITECTURES

COGNITION?



COGNITION?

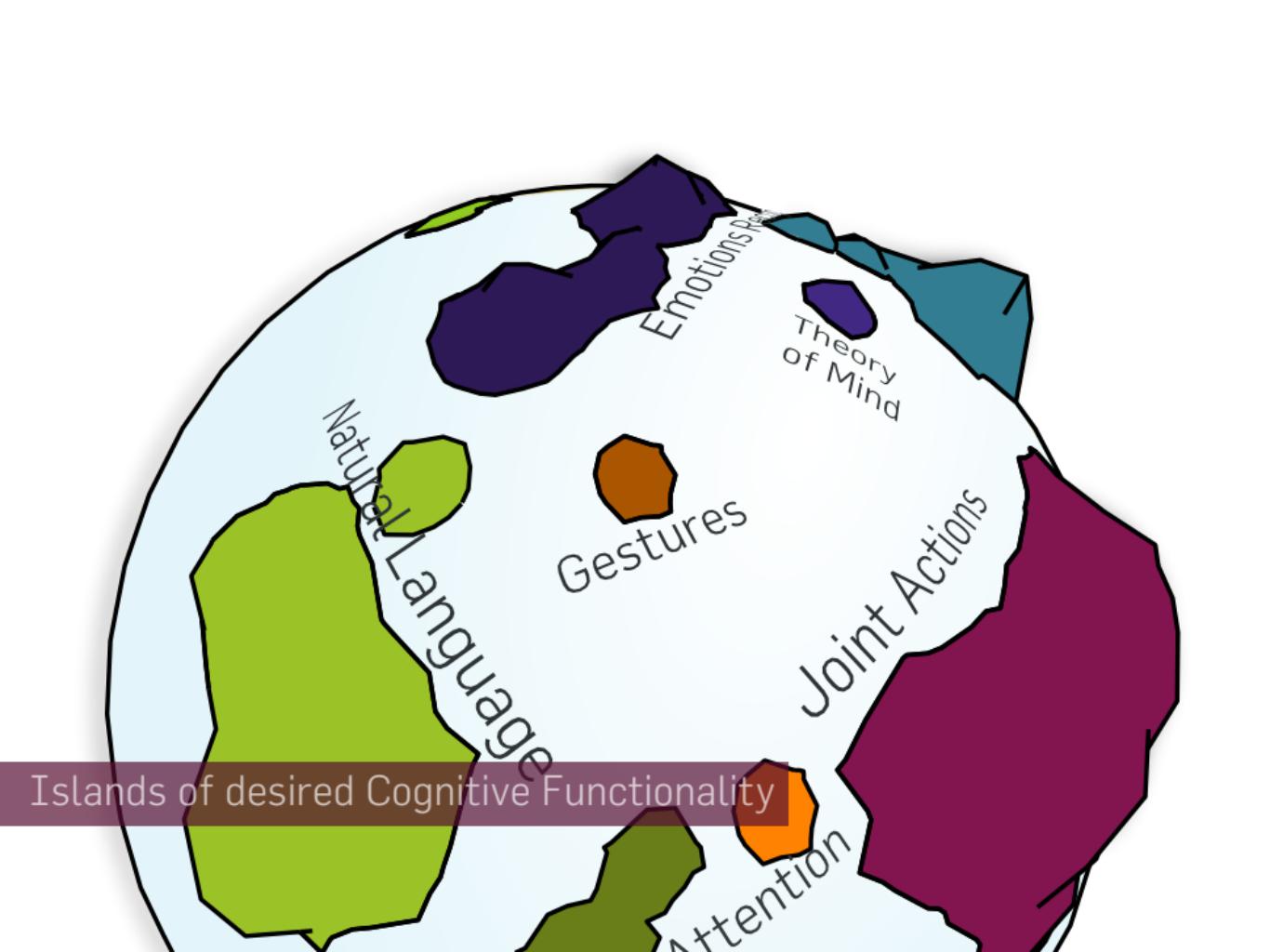


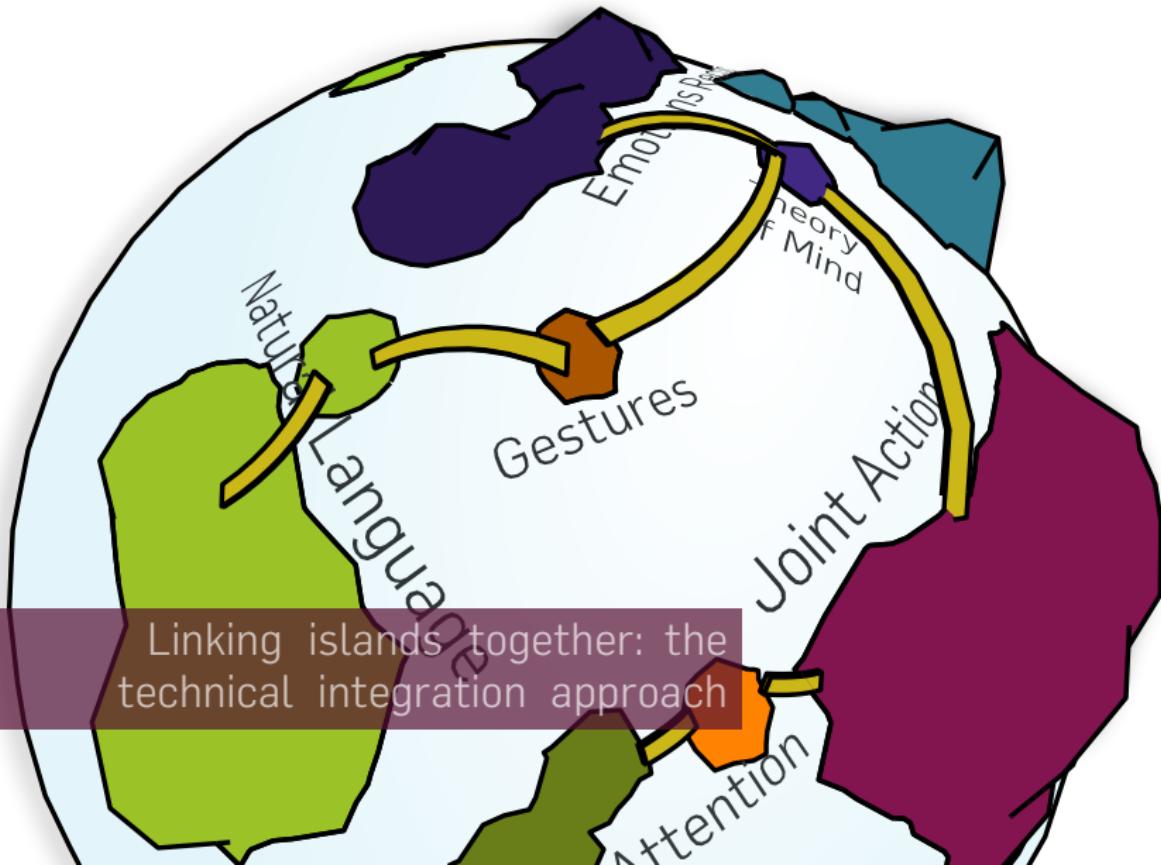
What is a cognitive architecture?

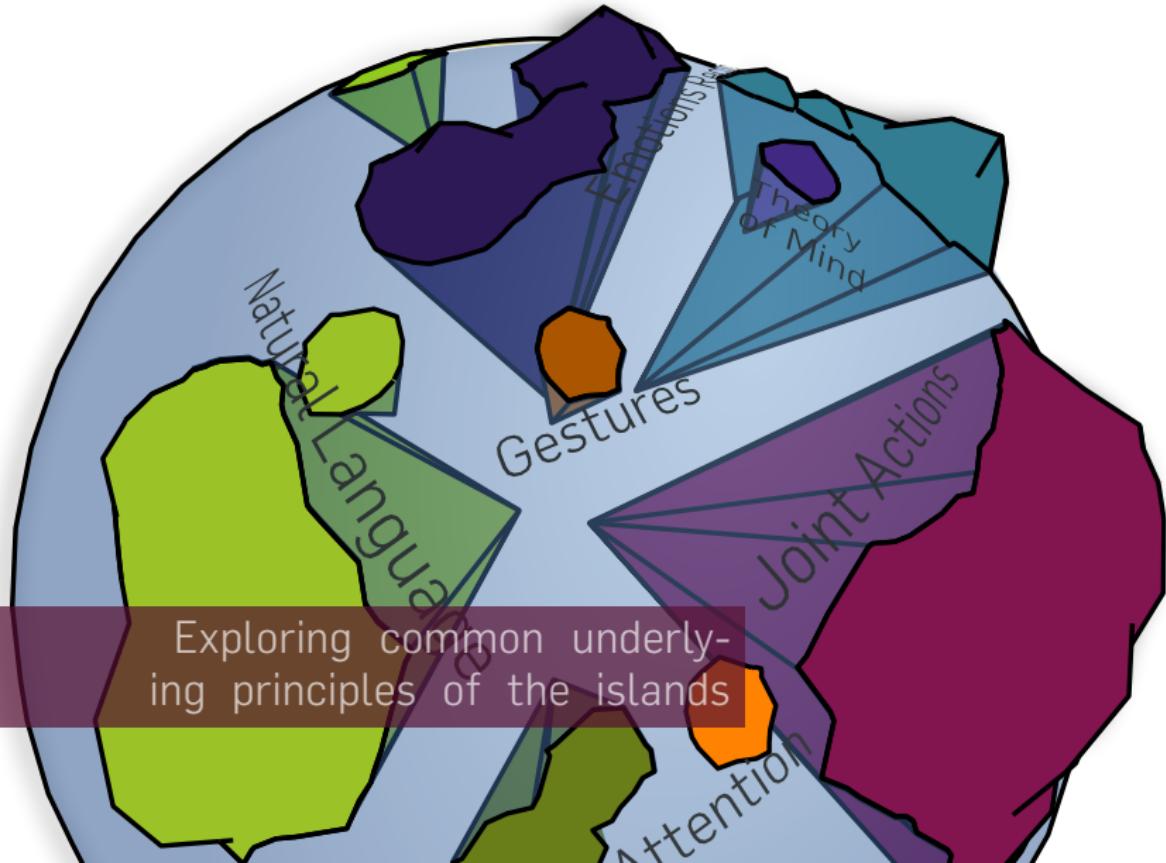
Support modelling (human) social cognition
(application to robots nice to have)

or...

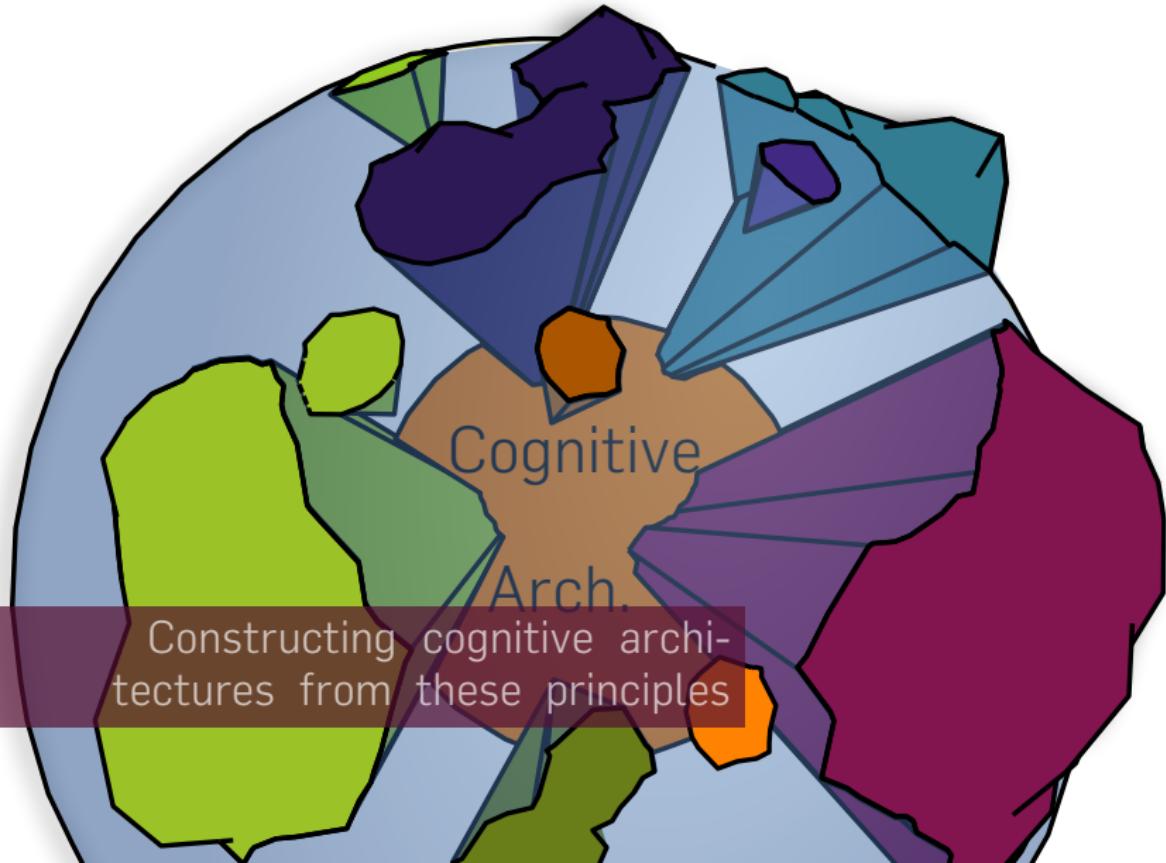
Support thinking about robot programming
in term of socio-cognitive skills







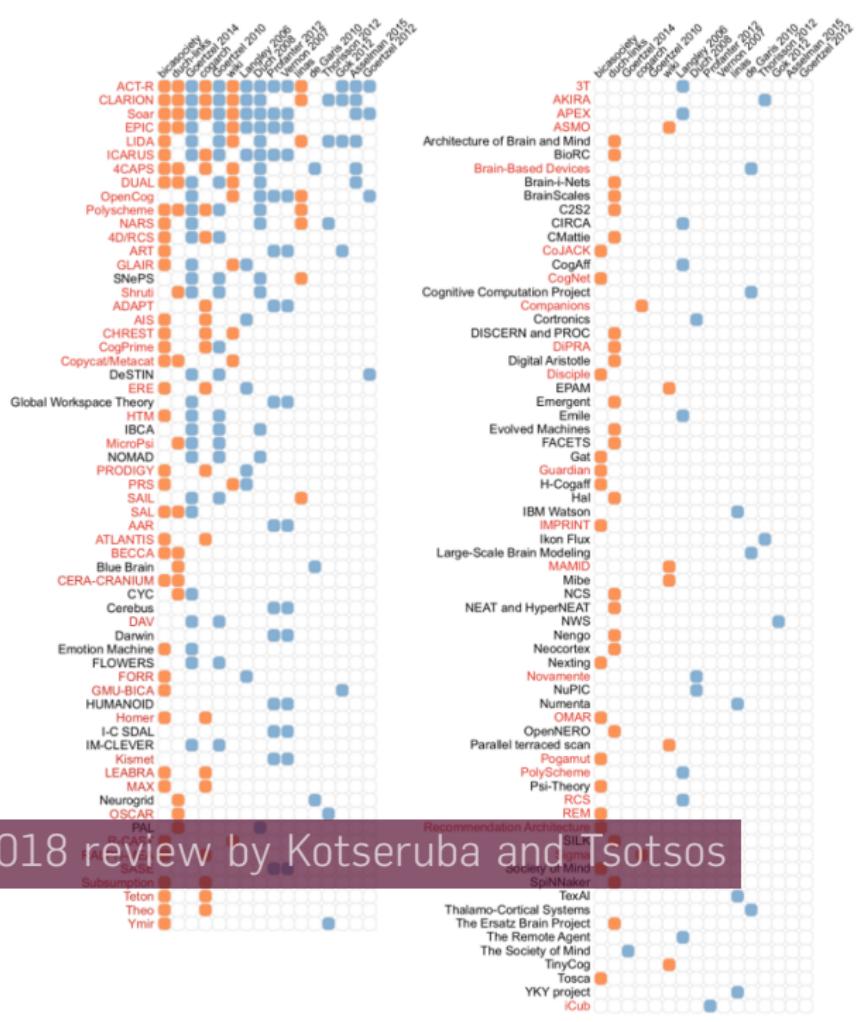
Exploring common underlying principles of the islands



Cognitive

Arch.

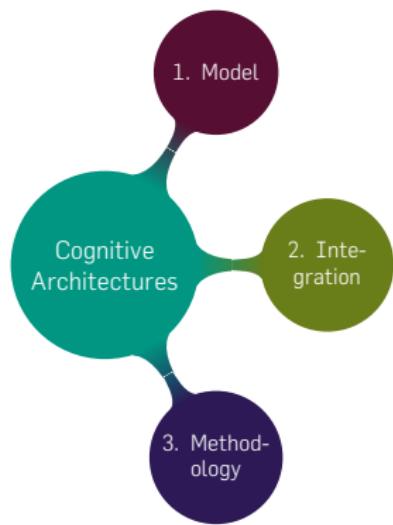
Constructing cognitive architectures from these principles



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
CELTIS
CRAM
Cognitive Symmetry Engine
Darwinian Neurodynamics
DIARC
EFAA
Feelix Growing
IAC
ISAC
JDE
MDB
Meta-AQUA
NEUCOGAR
Psikharpx
Roilo
SERAS
SIMA
SPA
The Cupboros Model
Juana 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

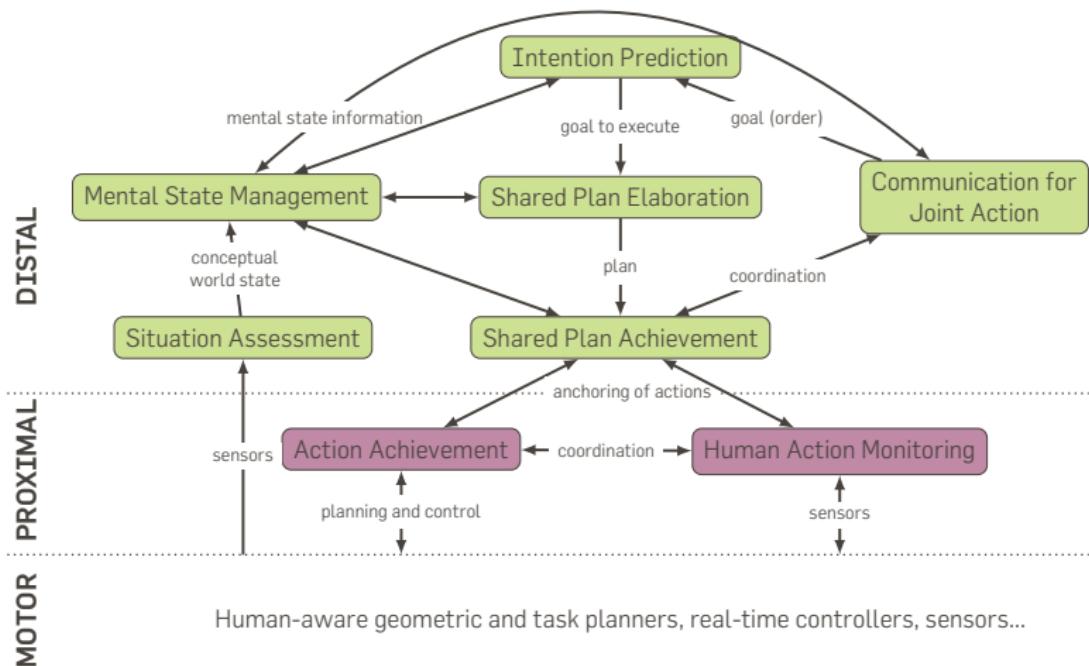
- Model some human cognitive phenomenon, or preferably set of phenomena (typically based on human behavioural data)
- Derive operating principles/algorithms that fulfil the requirements of the human behaviour
- Verify/validate resulting model by fitting to existing human data, or making predictions of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



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

Control paradigms

oooooooooooooooooooo

From an architecture perspective

oooooo

A glimpse at cognitive architectures

oooooooooooo●oooo

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.

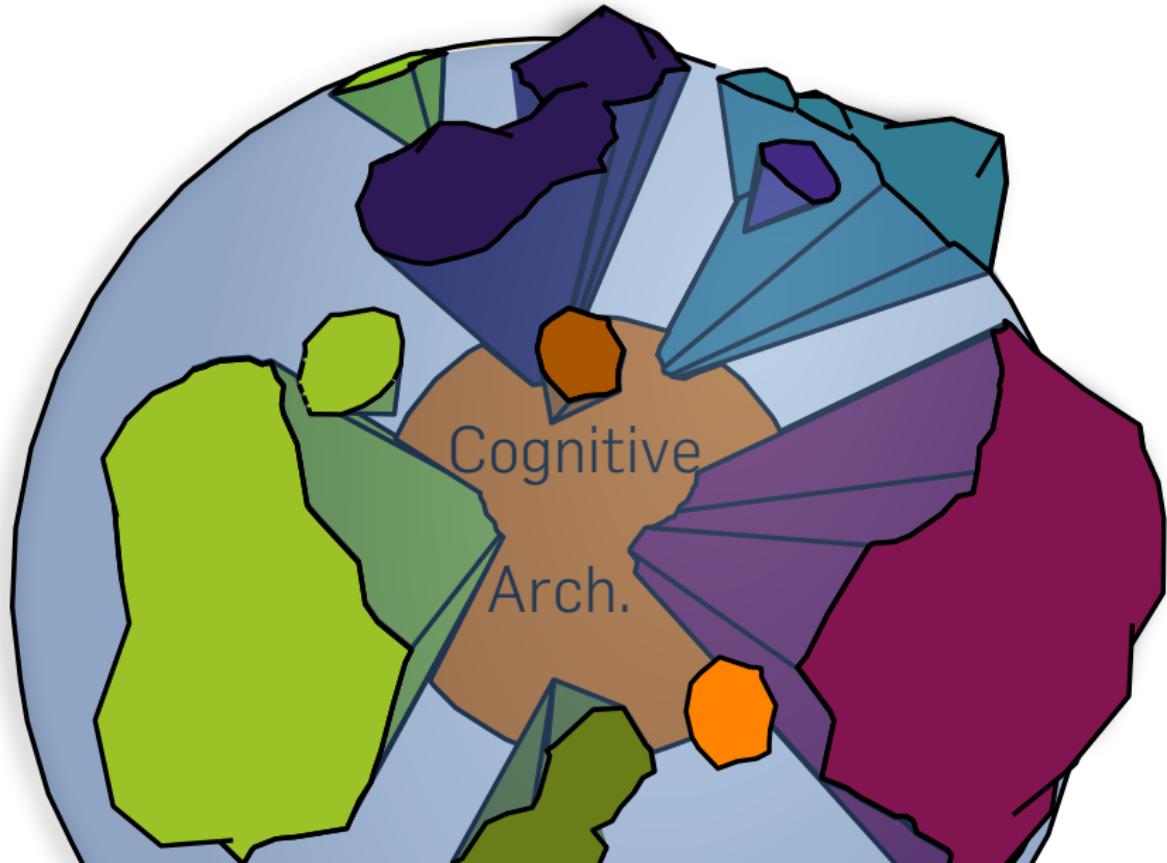
The underpinning mechanisms are generally hidden to the end-user.

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.



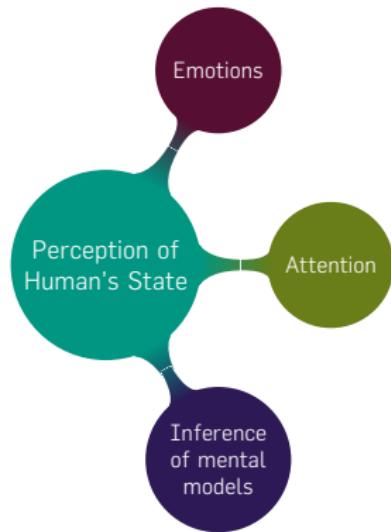
Cognitive

Arch.

SURFACE FUNCTIONS FOR SOCIAL COGNITION



PERCEPTION OF HUMAN'S COGNITIVE STATE



Emotions and expressions

Recognition facial, non-verbal acoustic,

verbal features / sentient analysis

Interpretation empathy,...

Attention

Visual attention

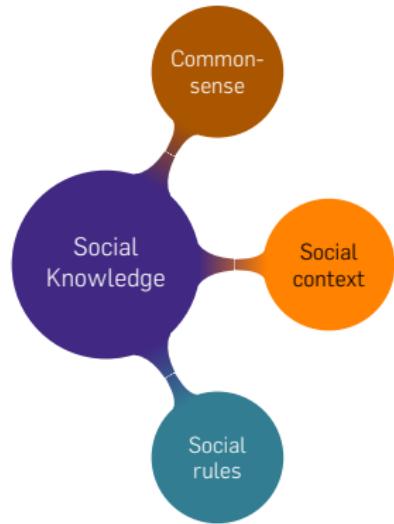
Joint attention

Mental Modelling

User modelling

(perceptual) theory of mind [2]

SOCIAL KNOWLEDGE



Common-sense

Hand-coded ontologies

Automated web-scraping [3]

Social context

Social Rules

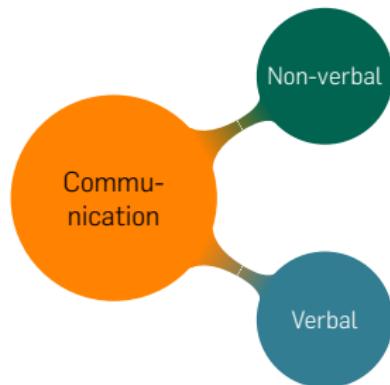
Proxemics

Social norms

Moral norms

Ethics

COMMUNICATION



Review [4]:

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

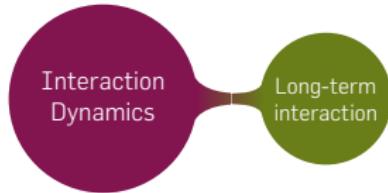
Non-verbal

Behavioural alignment

Verbal

Situated dialogue [5]

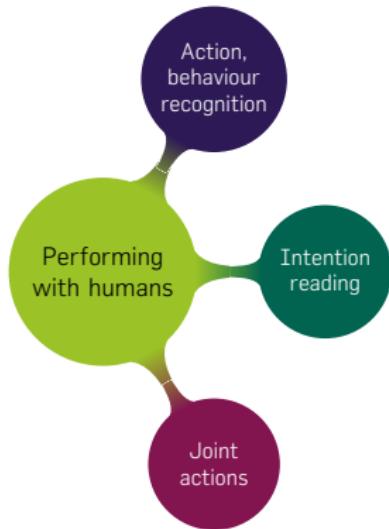
INTERACTION DYNAMICS



Dynamics Turn-taking
Mimicry
...

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

PERFORMING WITH HUMANS



Action & Behaviour recognition

Hidden Markov Models

Intention reading

Recognition of conventional situation ("the hitchhiker")

...

Joint action

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@birl.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,rx))
Deletions: AT(ROBOT,$1,$2), NEXTTO(ROBOT,$1)
Additions: *AT(ROBOT,x,y)

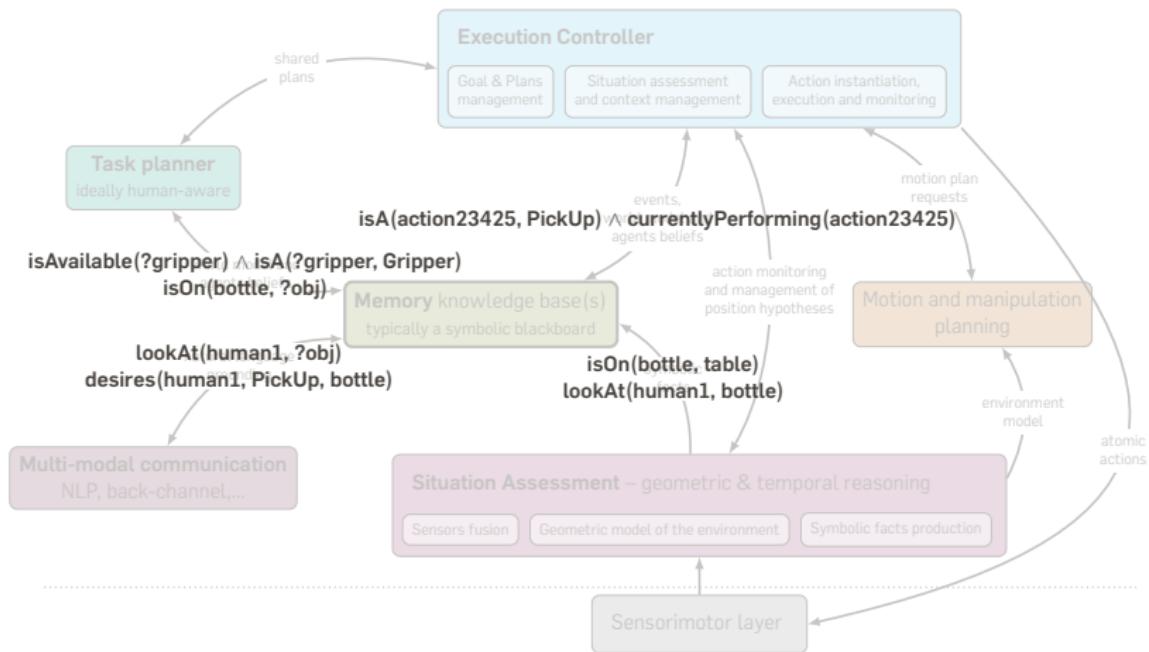
Go through door dx into room rx.
GOTHUDR(dx,rx)

Preconditions: TYPE(dx,DOOR), STATUS(dx,OPEN), TYPE(rx,ROOM),
NEXTTO(ROBOT,dx) (?rx)(INROOM(ROBOT,rx) ∧ CONNECTS(dx,rx,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,...)
- ⇒ **close to the human level** (both for the users and the developers)

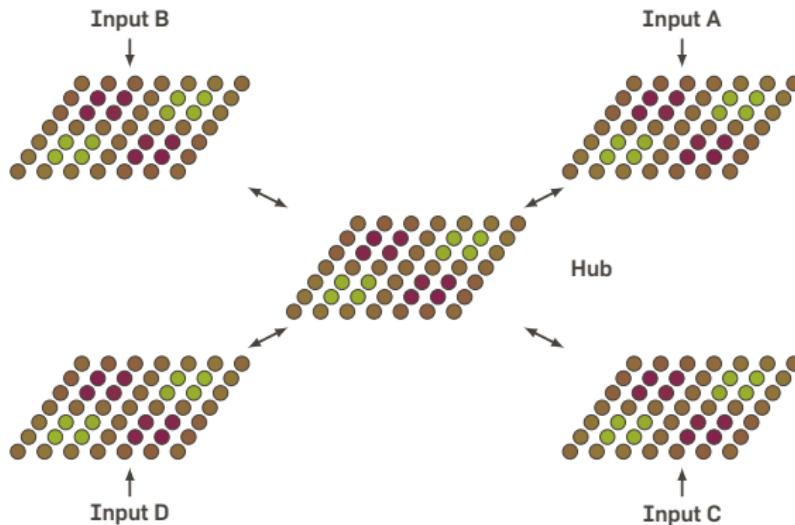
HIGHLIGHTS

- **well-established theoretical grounds** in formal logics
- **explicit, high-level semantics** (good for NLP for instance!)
- good engineering properties (modular, loose coupling,...)
- ⇒ **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,...?)

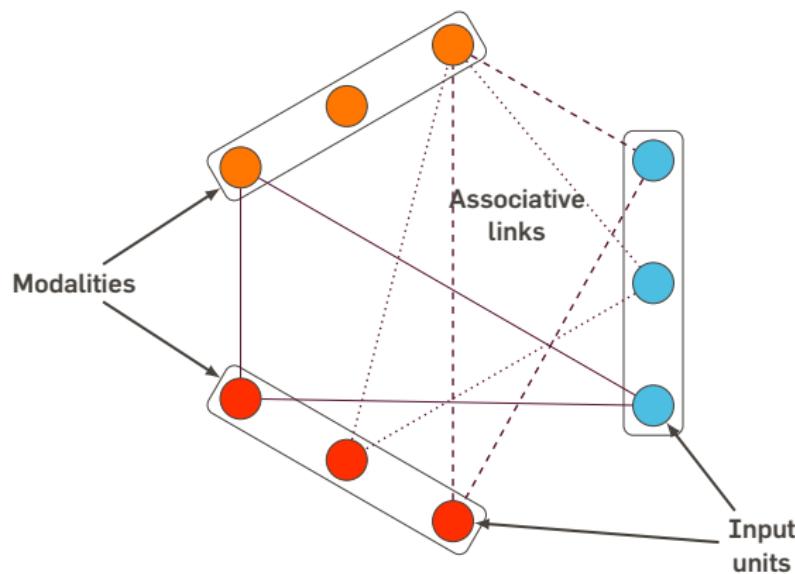
SUB-SYMBOLIC ARCHITECTURES

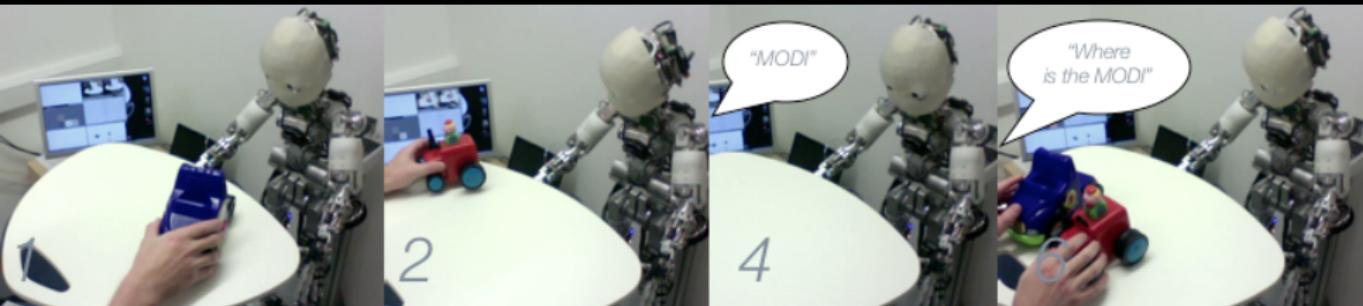
- (Hierarchical) Self-Organizing Maps [2]



SUB-SYMBOLIC ARCHITECTURES

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





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 apparatus
- applications to high-level socio-cognitive tasks only emerging
- **still hard to deal with time!**

Control paradigms

oooooooooooooooooooo

From an architecture perspective

oooooo

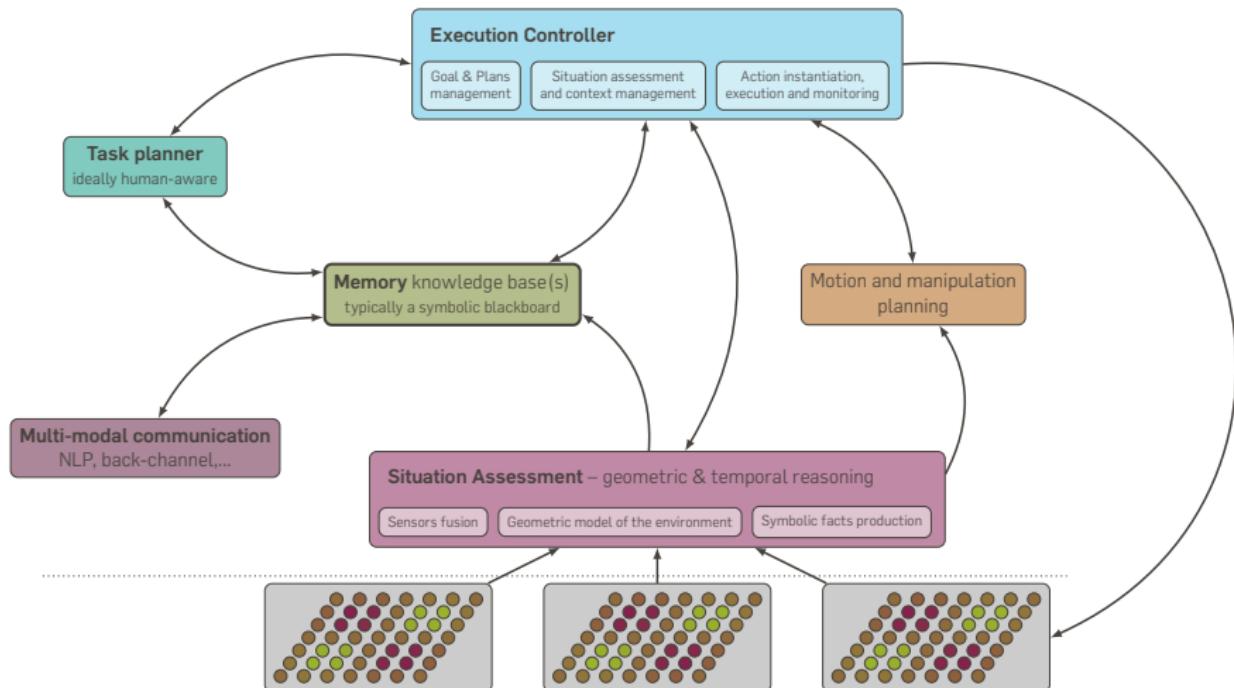
A glimpse at cognitive architectures

oooooooooooooooooooo

HYBRID ARCHITECTURES

What about...

HYBRID ARCHITECTURES



Control paradigms

oooooooooooooooooooo

From an architecture perspective

oooooo

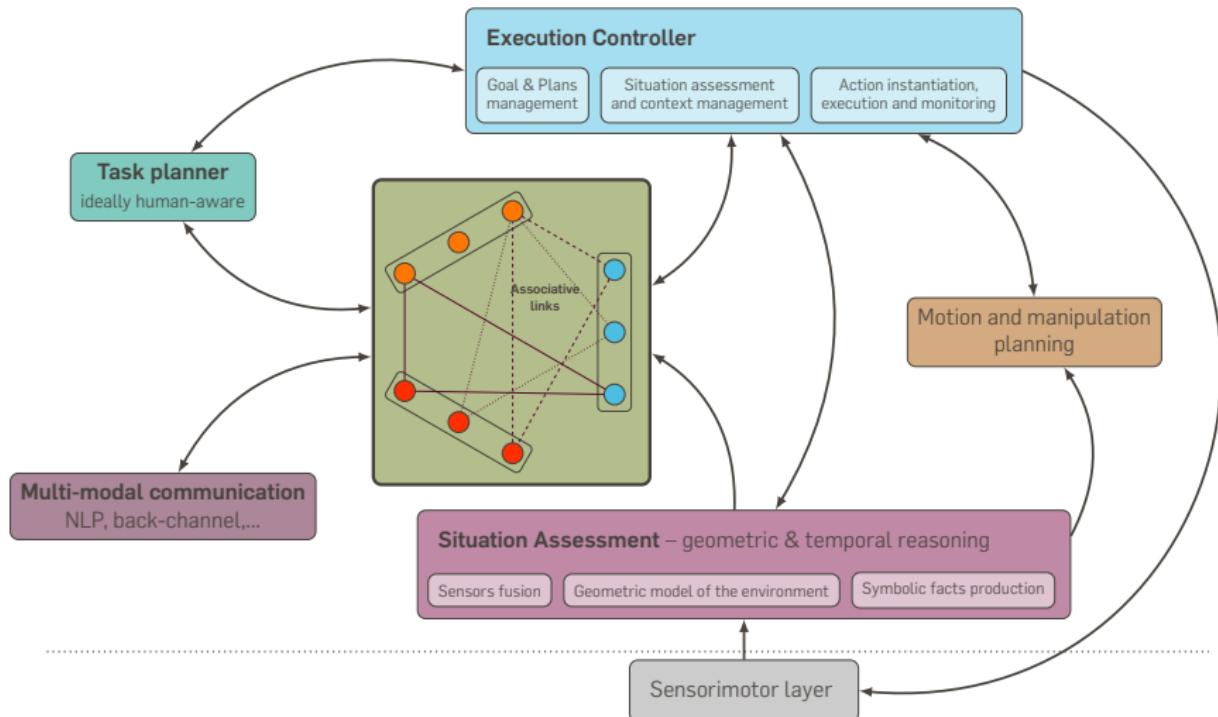
A glimpse at cognitive architectures

oooooooooooooooooooo

HYBRID ARCHITECTURES

or...

HYBRID ARCHITECTURES



Control paradigms

oooooooooooooooooooo

From an architecture perspective

oooooo

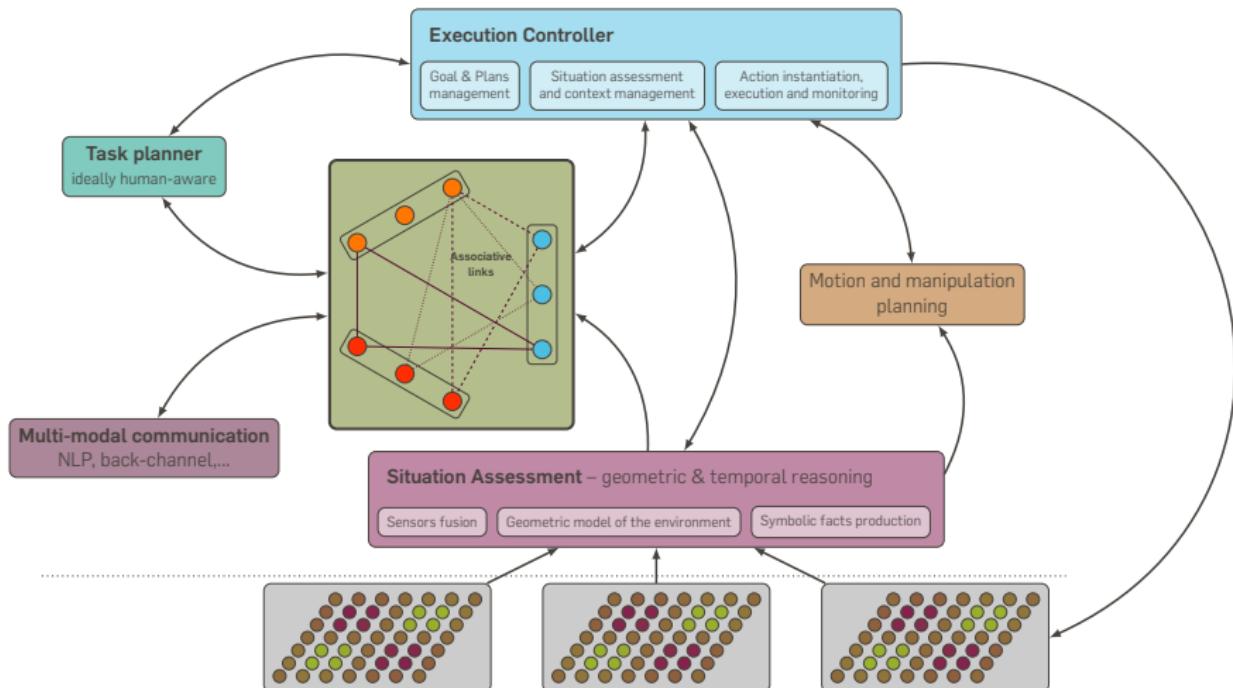
A glimpse at cognitive architectures

oooooooooooooooooooo

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)

⇒ “**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)

⇒ “**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!**