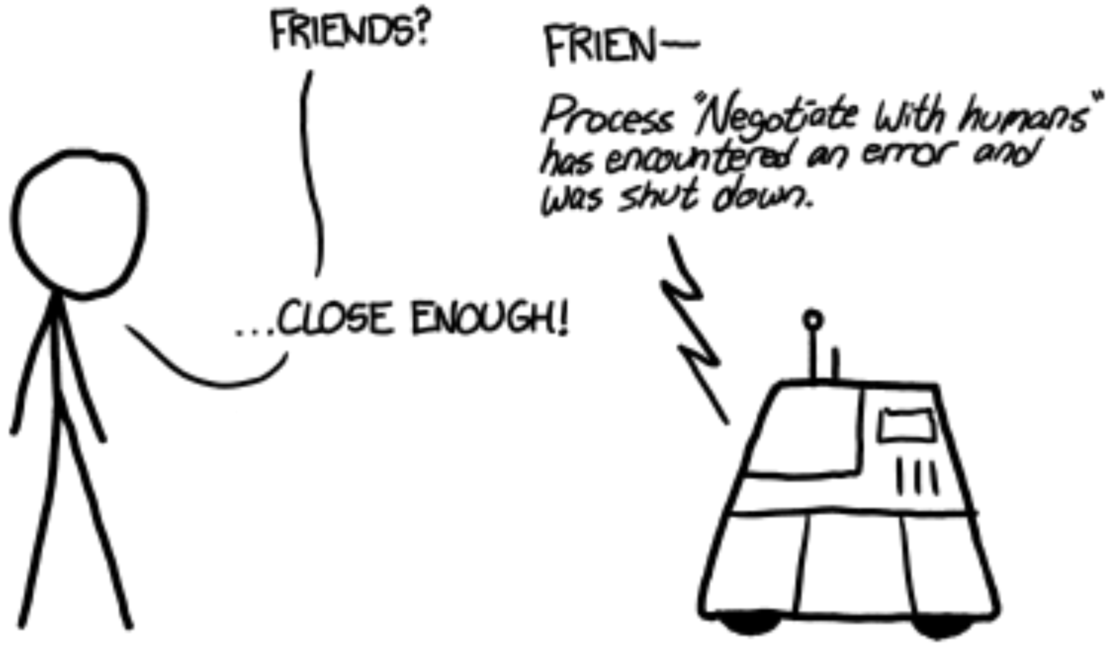


Lecture 9
"Teach me something."
Cognitive Skills for a Robot

Kim Baraka
Assistant Professor
Social AI group

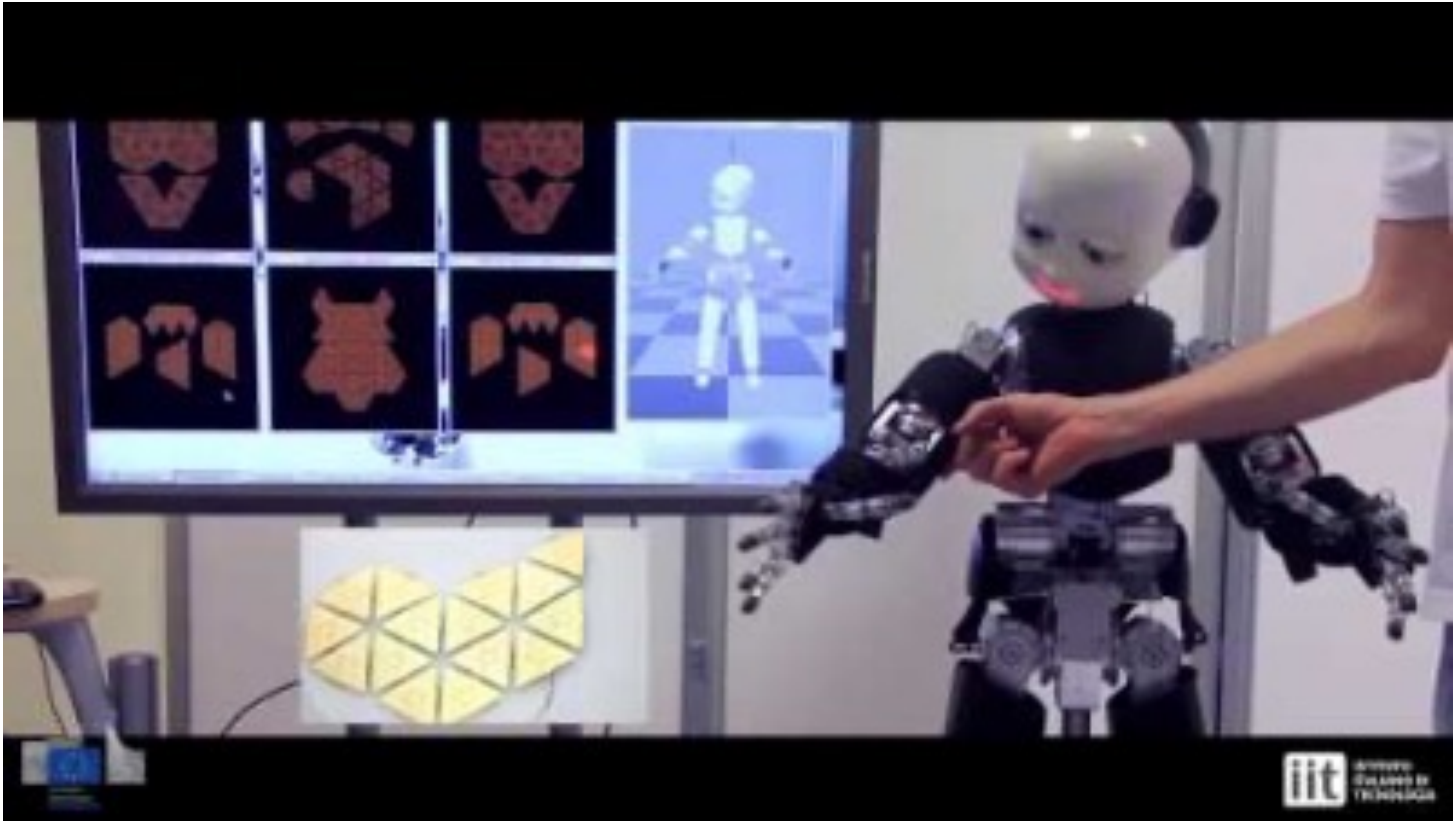
Cognitive skills in humans?

Non-cognitive skills in humans?

“Cognition” is a loaded term with varying definitions

- Traditionally, “social skills” are generally considered non-cognitive
- In fact, social skills heavily rely on cognitive processes
- Social cognition: “Mental processes involved in perceiving, attending to, remembering, thinking about, and making sense of the people in our social world.” (G. B. Moskowitz)

In this lecture, we take a broad view on what counts as a cognitive skill in a robot: basically any entity or process that allows a robot to process information and communicate with other similar entities, the environment, or other agents



https://www.youtube.com/watch?v=ErgfgF0uwUo

Learning Objectives:

- Able to explain key challenges of integrating robot cognitive skills and various robot architecture proposals
- Able to identify/disentangle challenges in robot learning in the context of an interaction with a human
- Able to explain different human-interactive robot learning solutions
- Able to brainstorm about novel forms of human-interactive robot learning

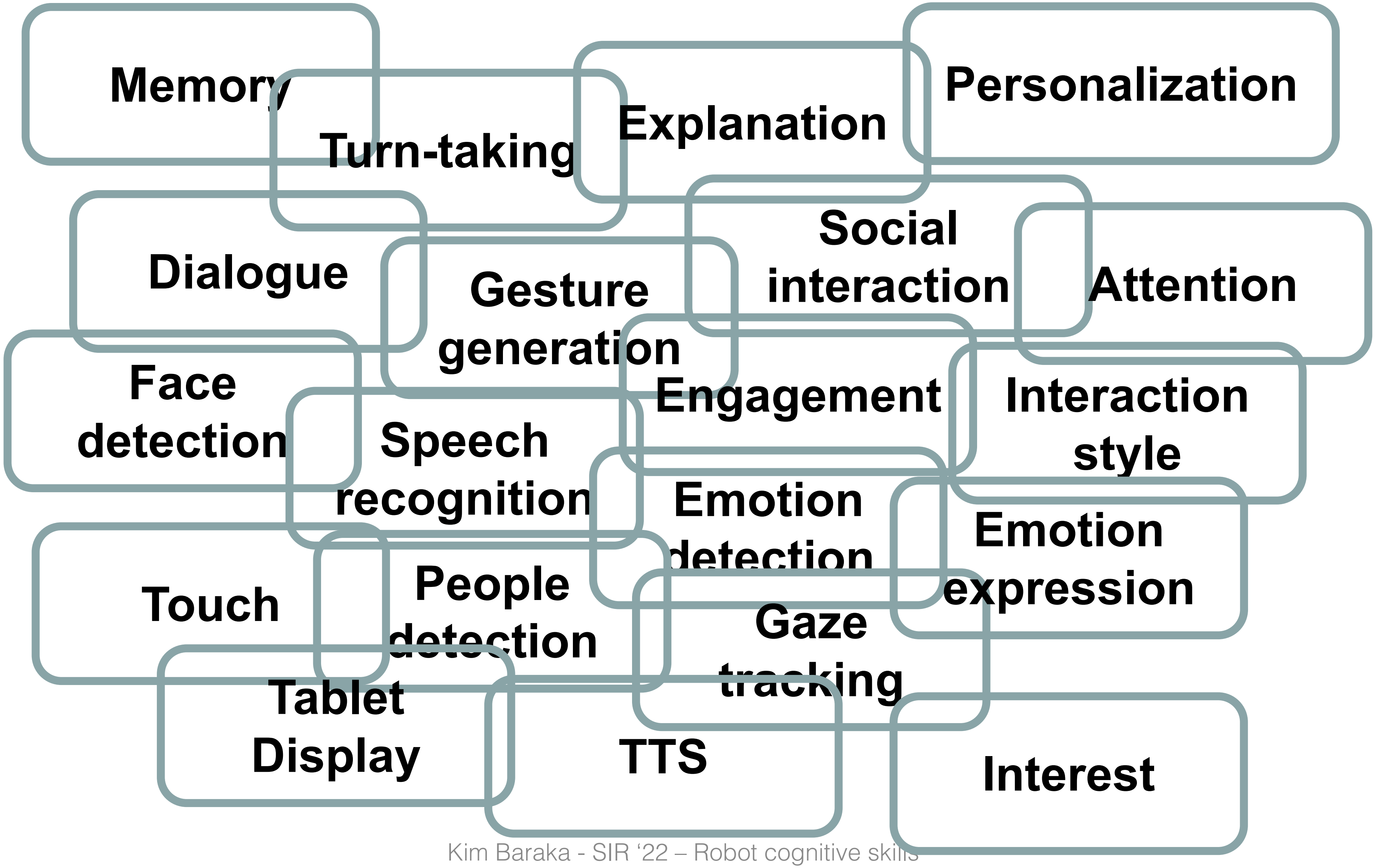
PART I: ARCHITECTURES FOR “SOCIAL INTELLIGENCE” IN ROBOTS

Motivating example: social greeting

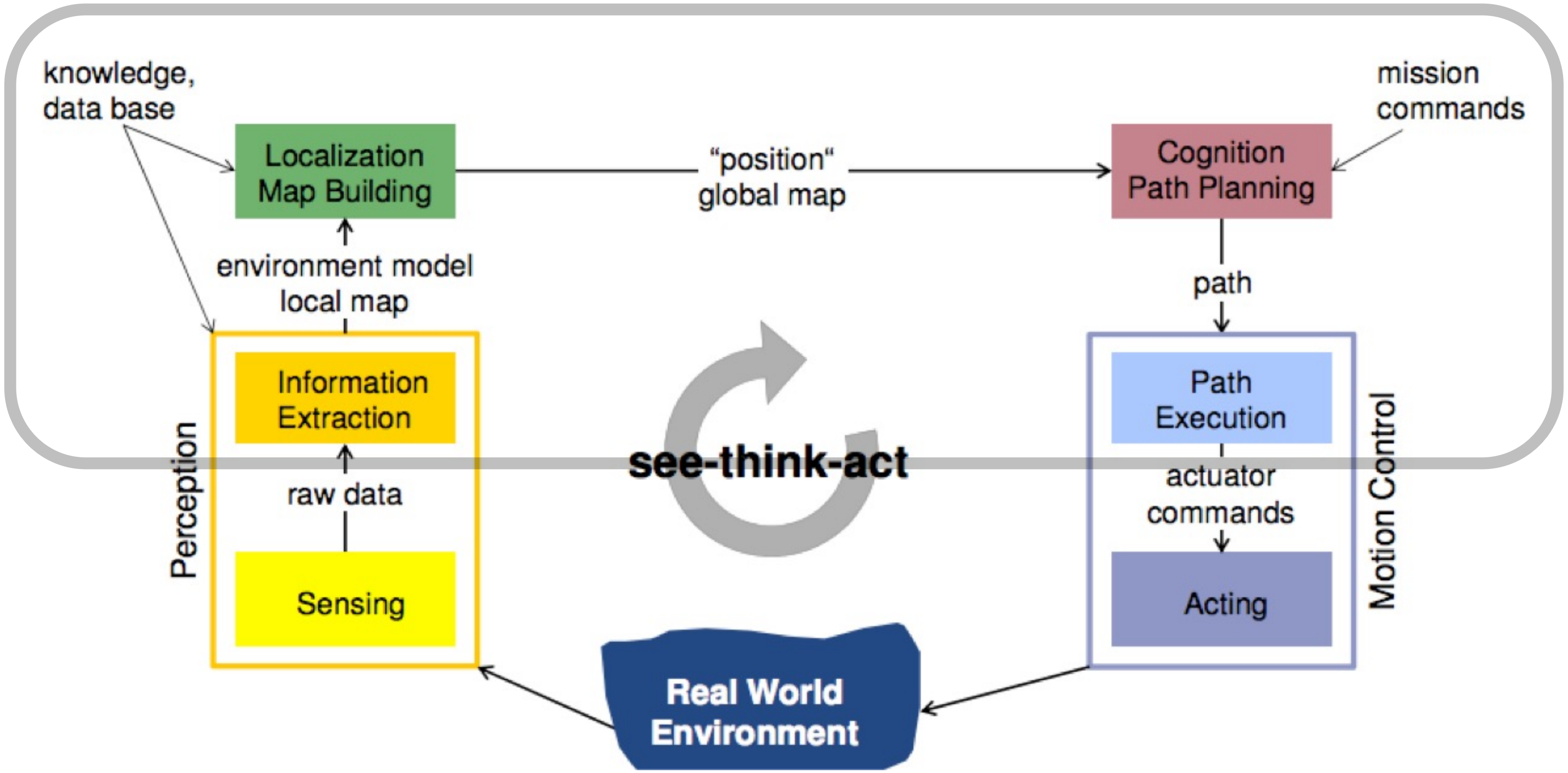
https://www.youtube.com/watch?v=0e3pEtECwLg

https://www.youtube.com/watch?v=0e3pEtECwLg

Creating Order from Chaos: Integrating HRI

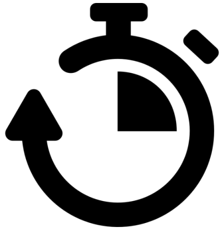


Robot Control in See-Think-Act Cycle

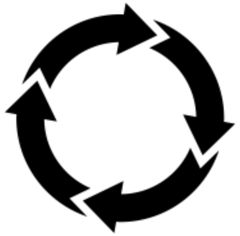


Integrating high-level (symbolic) planning with low-level control.

Software Engineering Robot Control



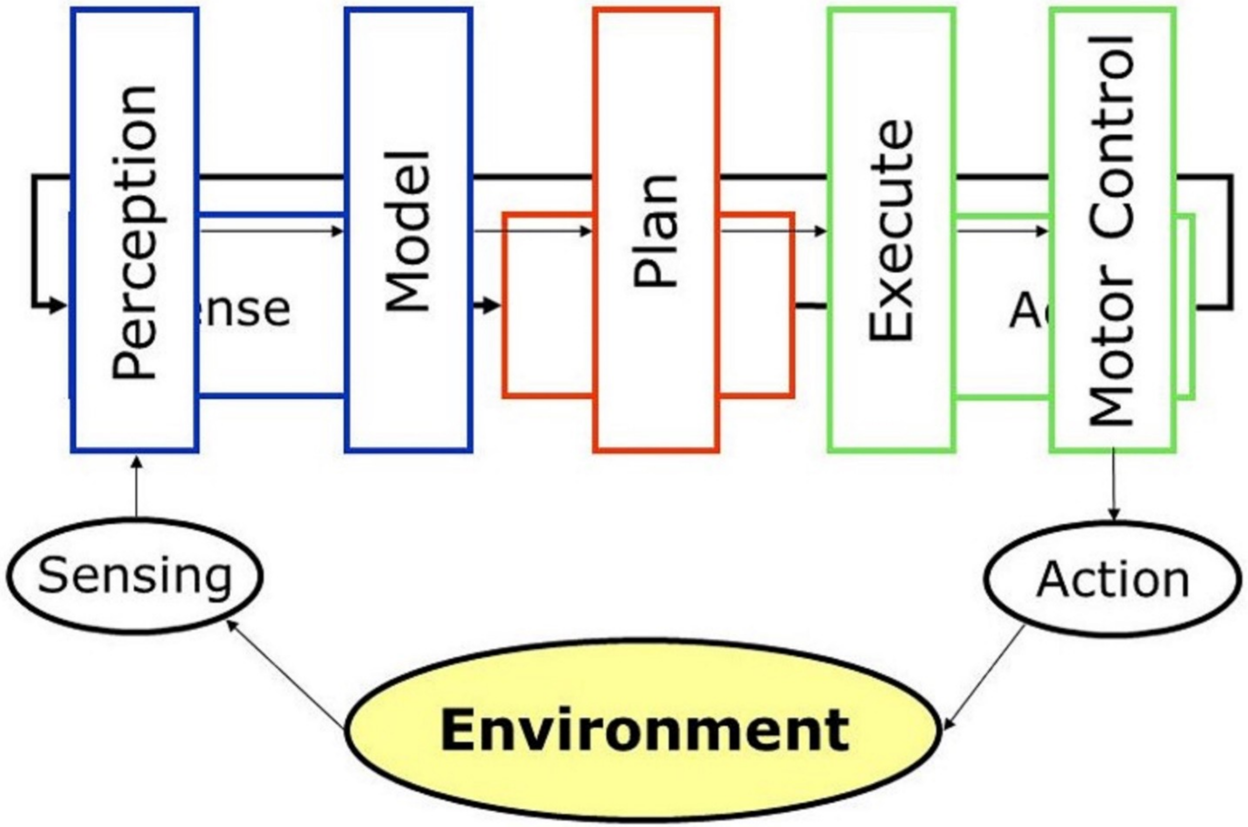
- **Real-time control**: supports event-based, reactive, and distributed interactions between sensors, motors and algorithms.



- **Reuse**: architecture is generic and abstracts from specific robot platforms for reuse.

- **Robustness**: ensures robust robot behavior and graceful degradation of task performance in case of failures.

Sense-Plan-Act (SPA)



Classic version of a "pipeline" architecture.

Classic SPA Architecture: Benefits & Issues

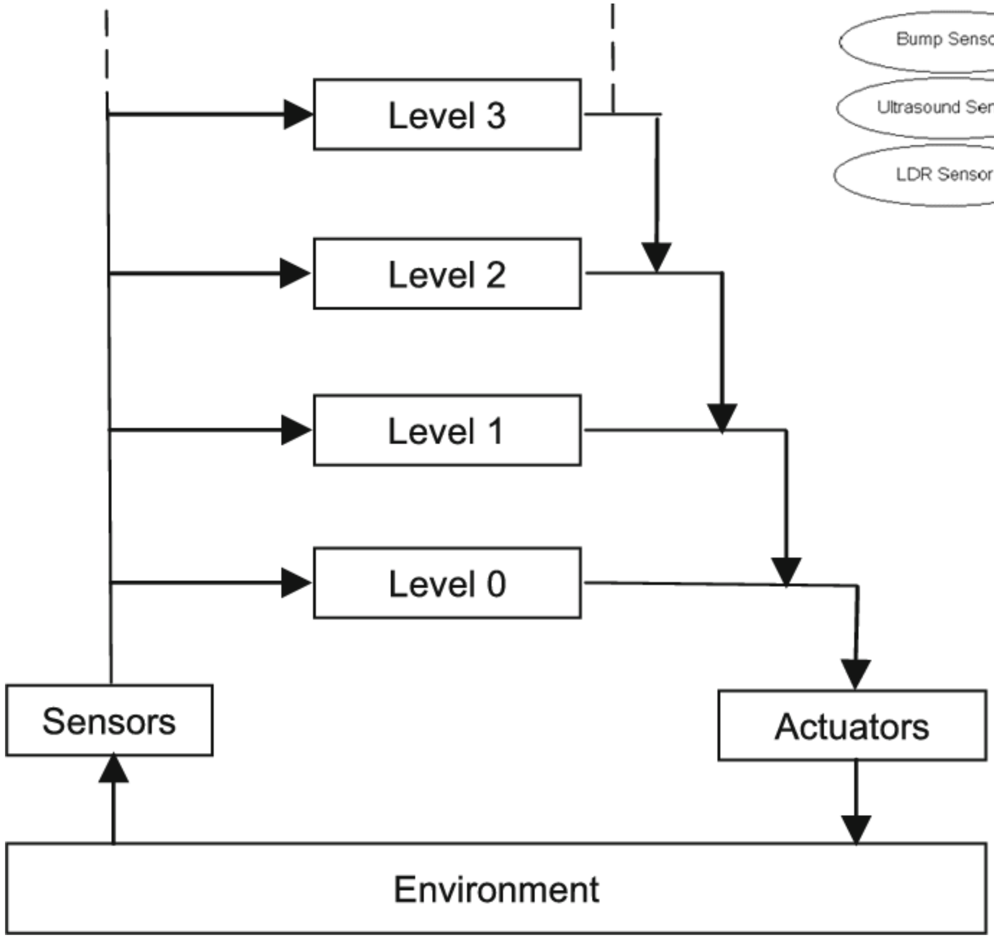
<u>Benefit</u>:

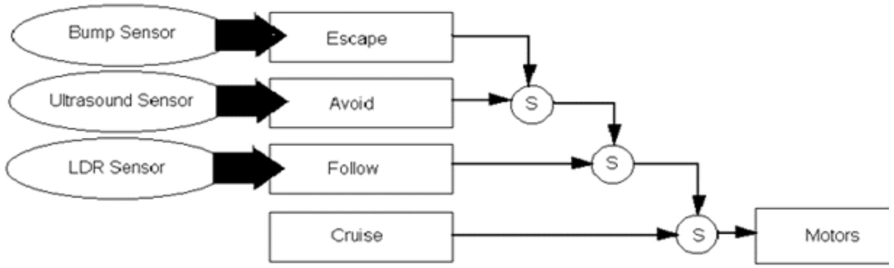
- Integrates symbolic and non-symbolic techniques.

<u>Issues</u>:

- Robot control *slow* due to “extensive deliberation”.
- Not very *robust* (no monitoring of task execution).

Subsumption Architecture (1985)

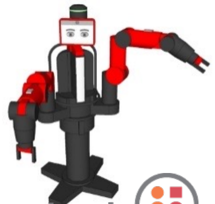





Rodney Brooks


Robot Genghis

iRobot



rethink robotics

Subsumption Architecture: Benefits & Issues

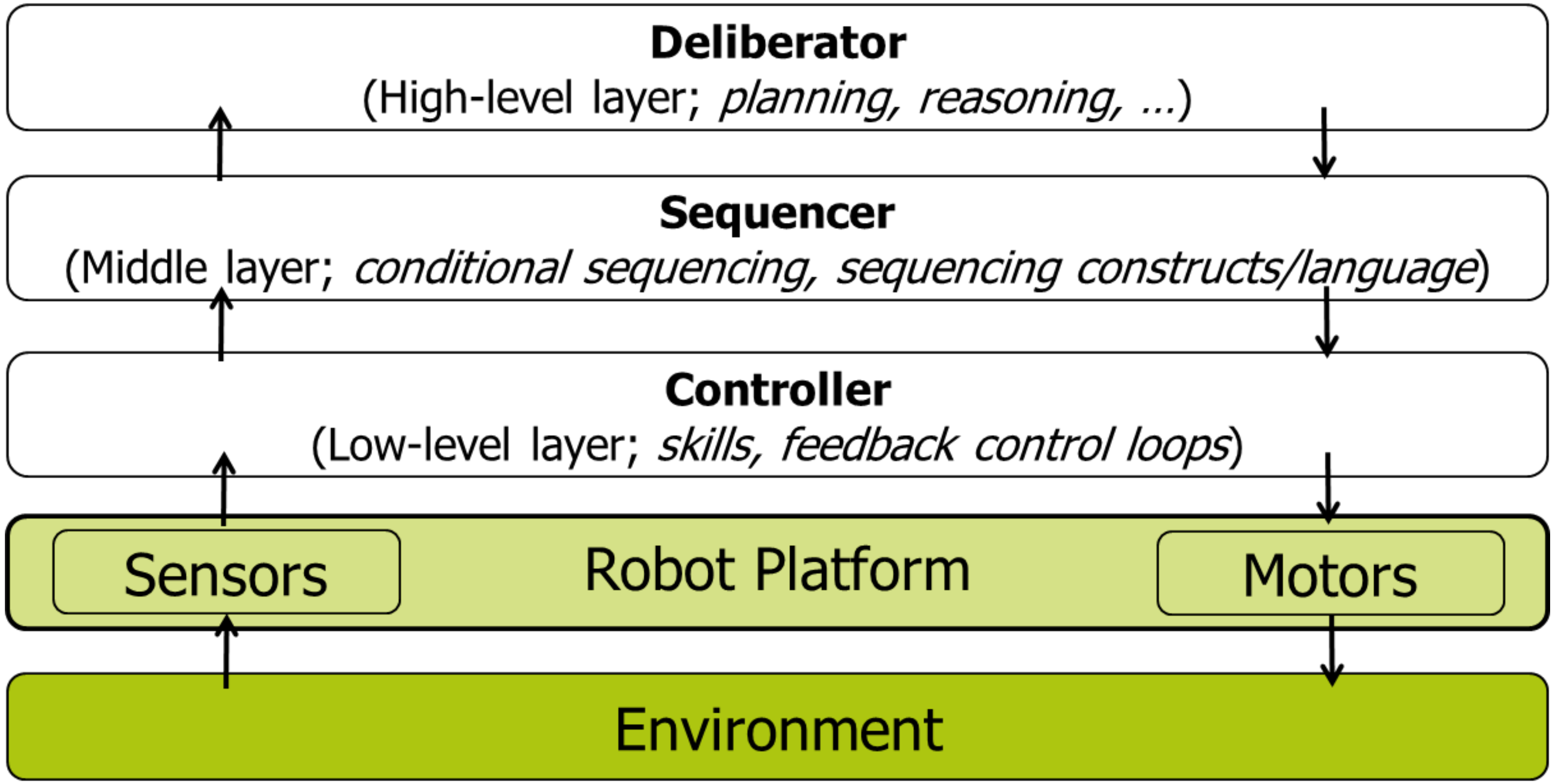
<u>Benefits</u>:

- Fast due to focus on behavior and tight sensor-behavior coupling (gave rise to behavior-based paradigm).
- Reactive, able to handle dynamic world due to constant sensing of the world.

<u>Issues</u>:

- difficult to compose behaviors to achieve long-range goals.
- almost impossible to optimize robot behavior.

3T or Layered Architectures



Classic examples: SSS (Connell 1991), ATLANTIS (Gat 1991), 3T (Bonasso 1991)

A Functional Perspective

3T

- (highest) *deliberative layer* responsible for task-planning and achieving long-term goals of the robot within resource constraints.

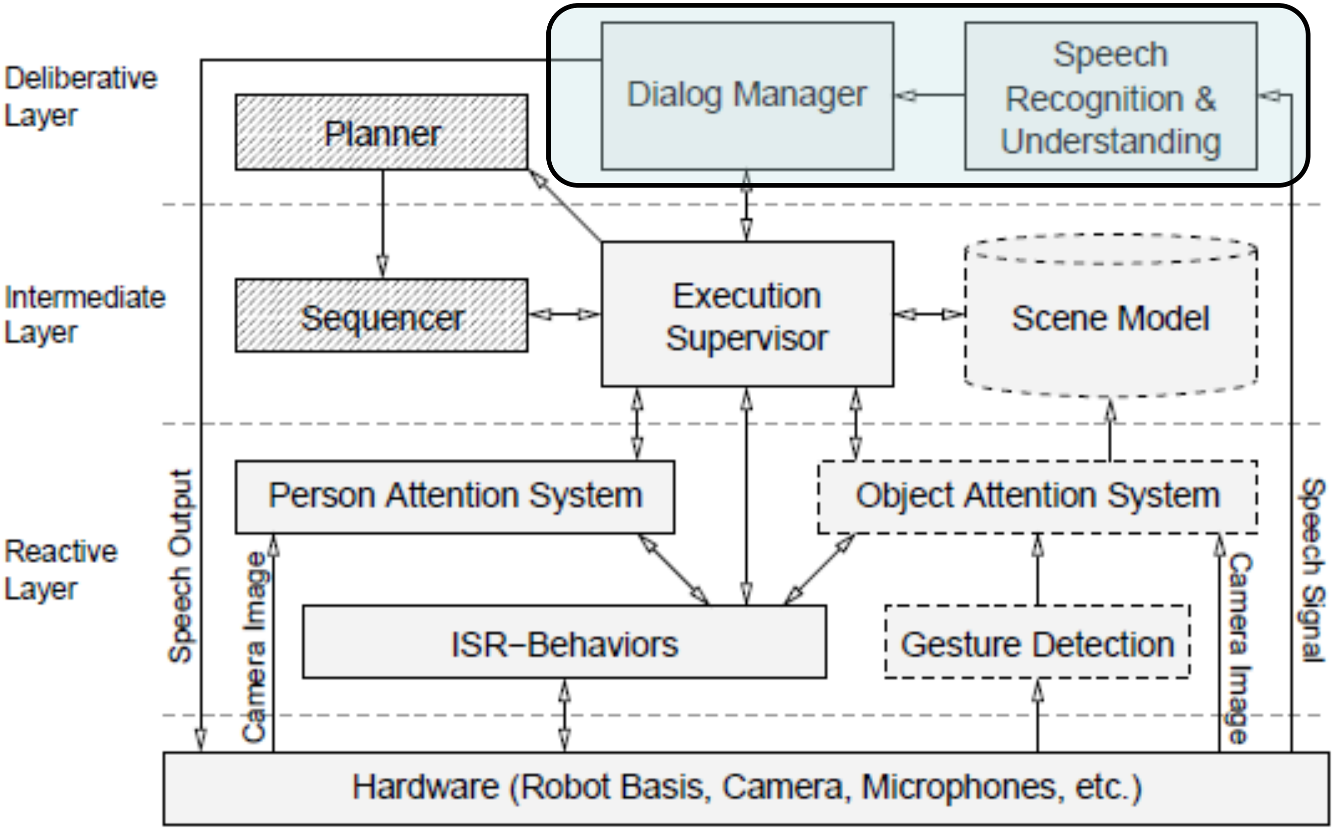
2T

- (middle) *executive or sequencer layer* responsible for choosing the current behaviors of the robot to achieve a task.

1T

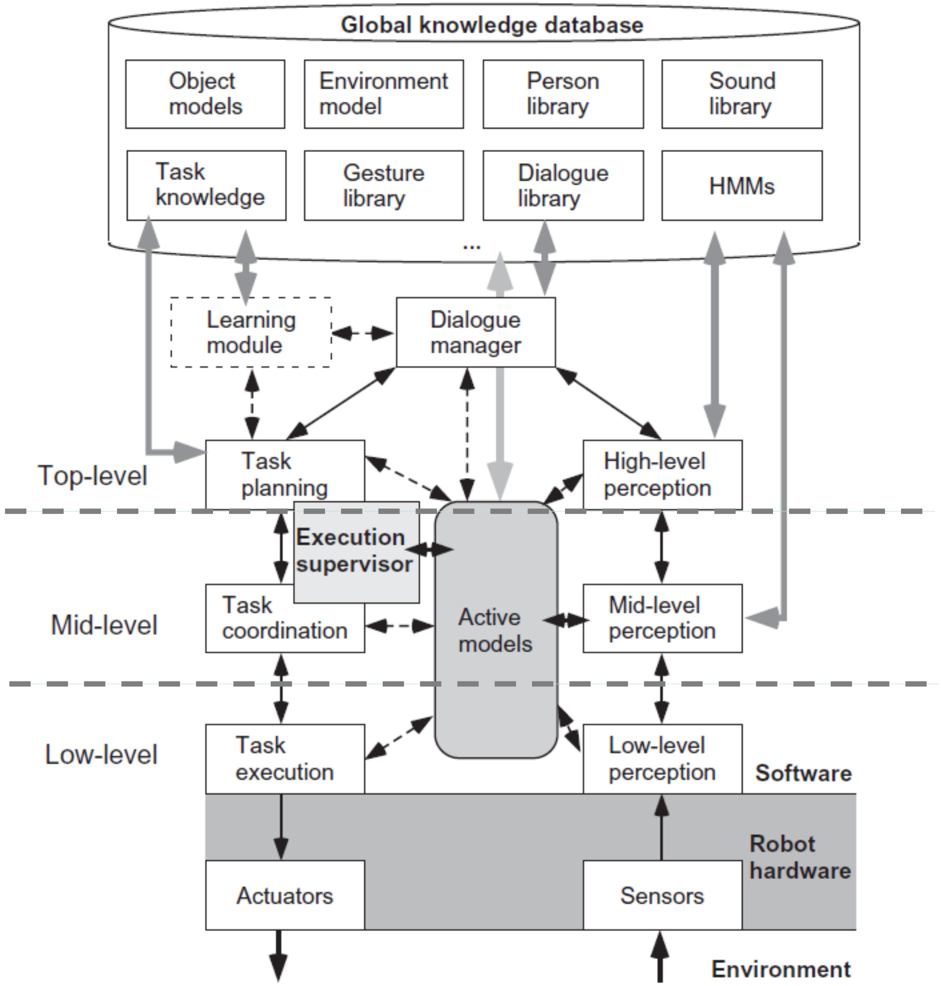
- (lowest) *behavioral control or skills layer* responsible for controlling *sensors* and *actuators*.

BIRON (2004)



The Bielefeld Robot Companion

Armar (Univ. of Karlsruhe)



3T Architecture: Benefits & Issues

Benefits:

- Rich architecture, with different levels of abstraction and clear “roles”: planning, execution (control and monitoring), and basic control layer (behaviors, …)

Issues:

- Complex: many ways to instantiate 3T, what is best?
- How many layers: >3 layers? Perhaps 2 layers?

  → Where to plan? Path planning at middle or highest layer?

Social Interaction Architecture

Support for:

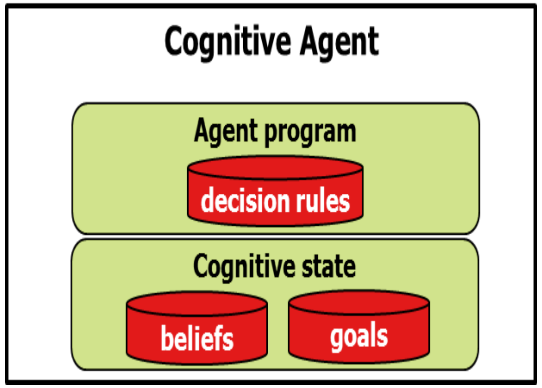
- Express emotions
- Perceive emotions
- Communicate with high-level dialog
- Model other agents
- Use social cues naturally
- Explain behavior
- Learn from interaction
- …

Finding the Middle Ground

Options available for developing an interaction architecture

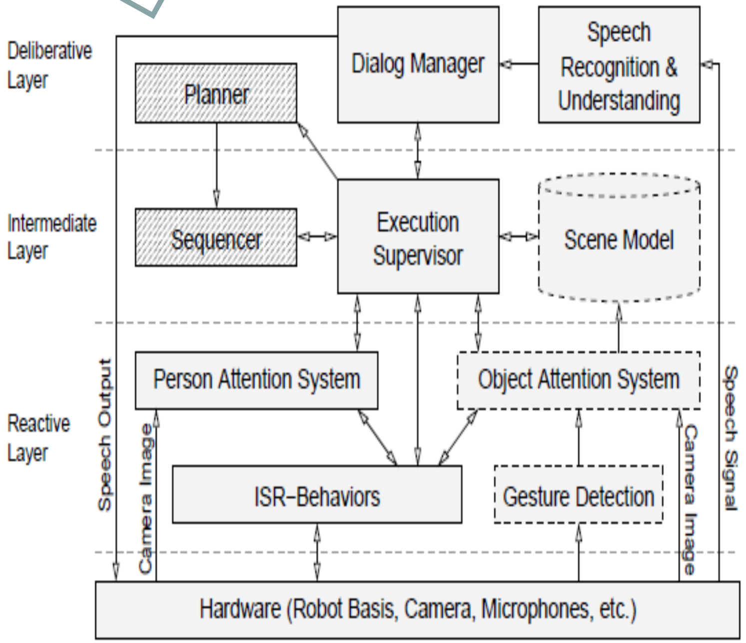
1 — General-purpose

3 — Hindriks (2009)



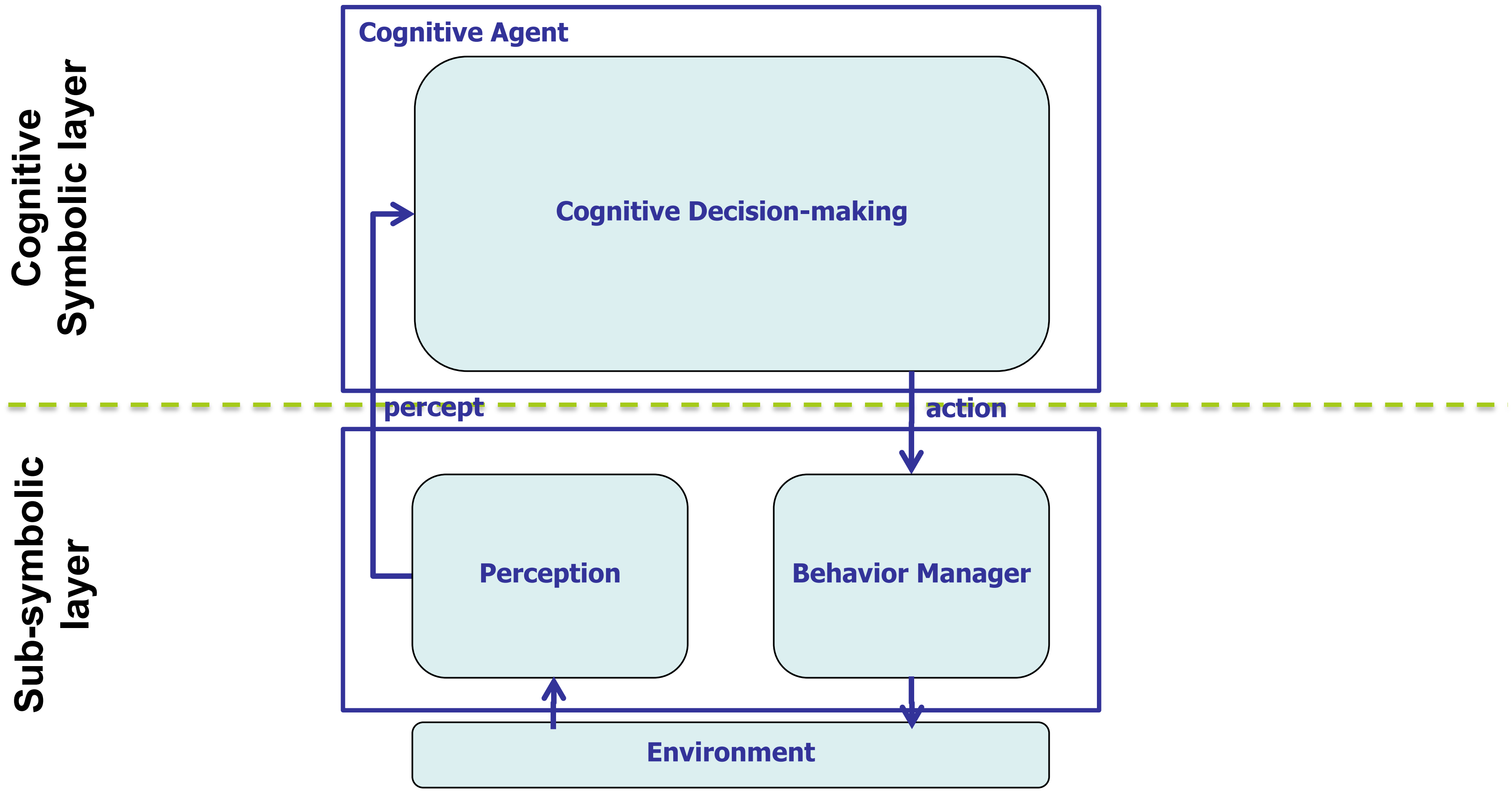
DSL for declarative decision-making

2

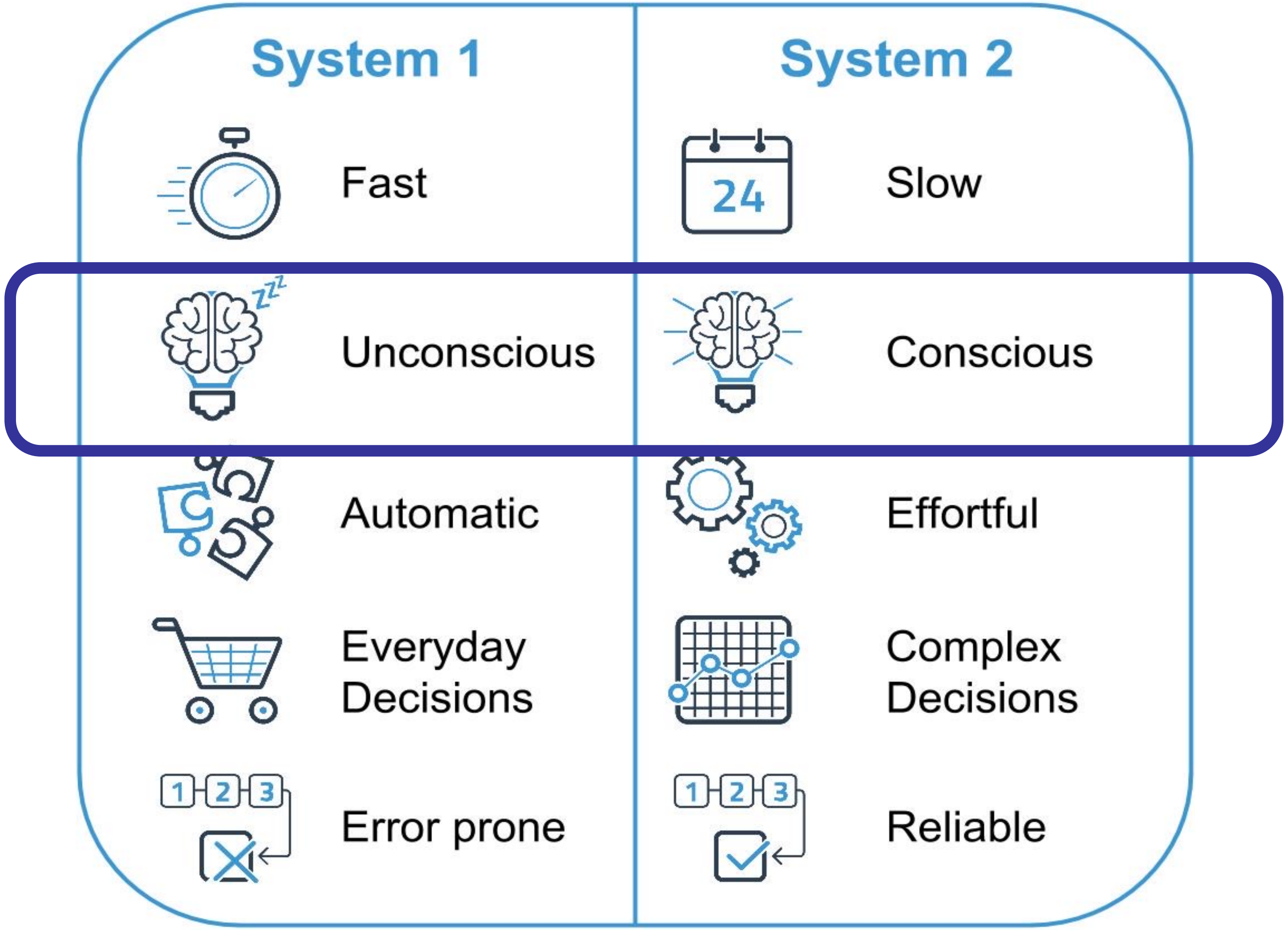


Typical 3-layer architecture

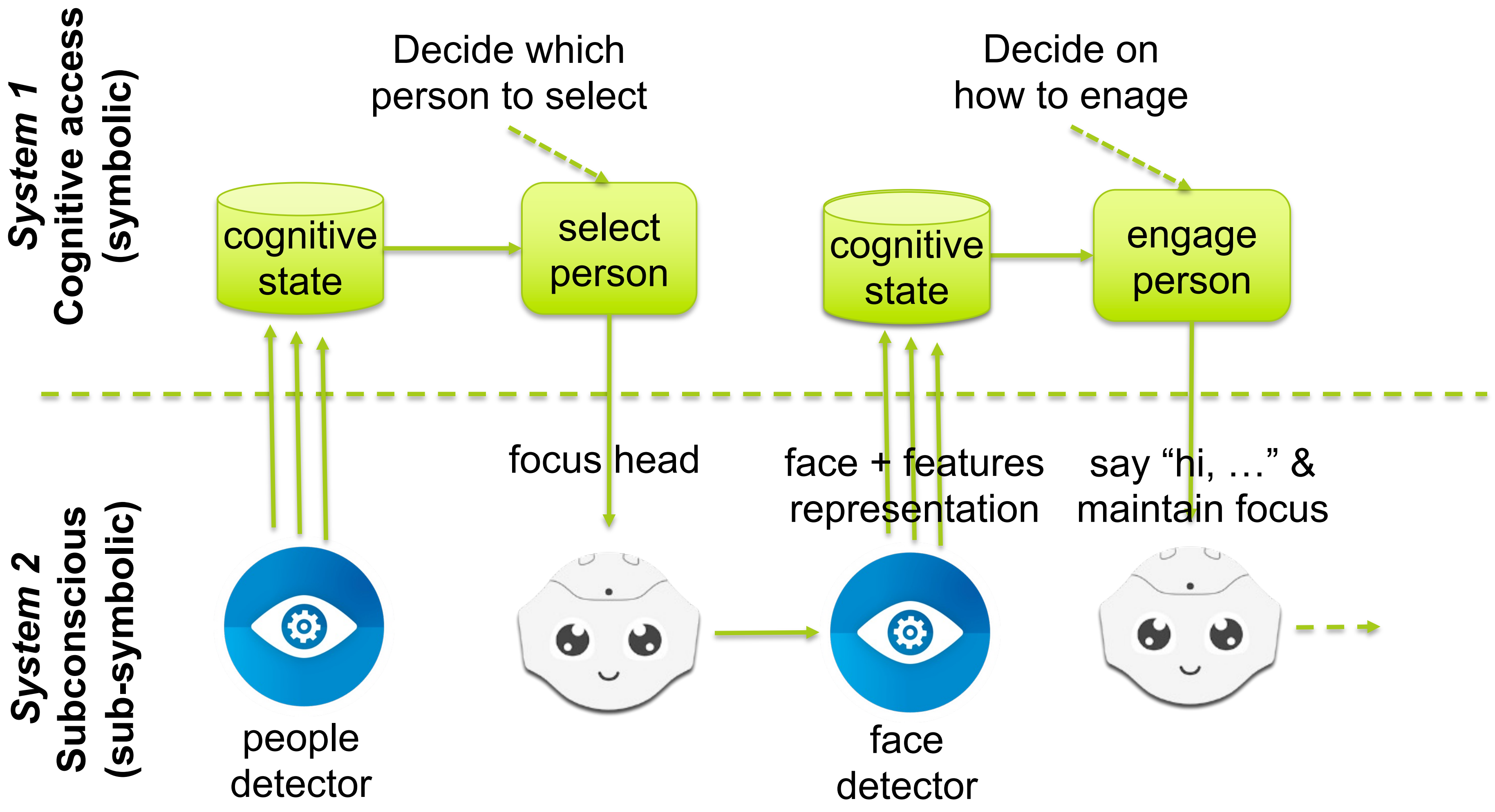
2-Layered Interaction Architecture



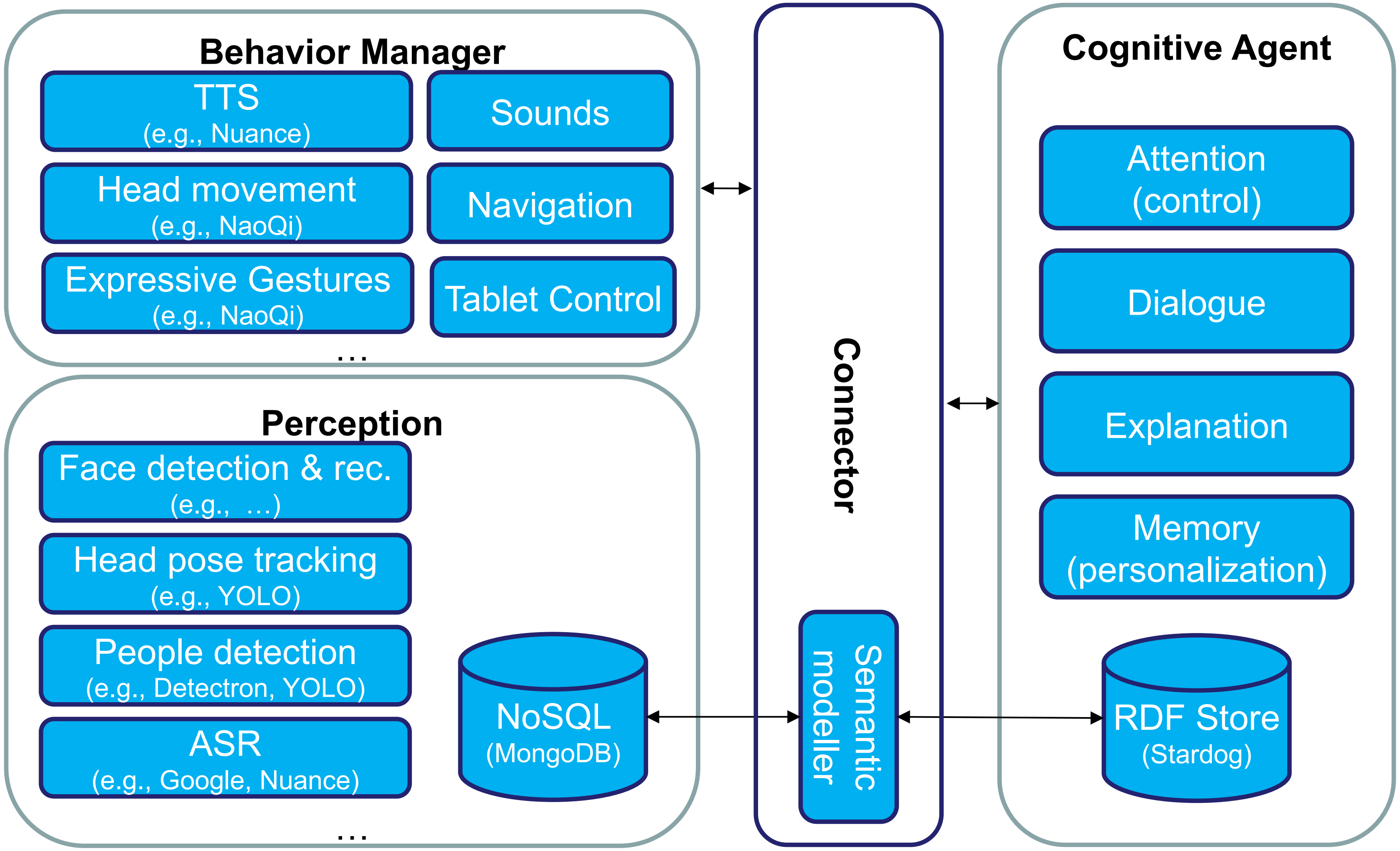
System 1 & 2 Architecture Design



Example: Attention/Engagement



Technical Architecture

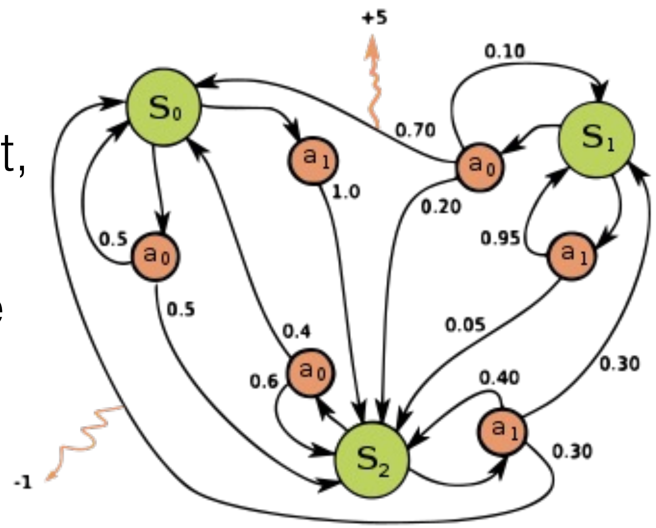


PART II: HUMAN-INTERACTIVE ROBOT LEARNING

Formulating a (sequential) robot learning problem: MDP

A Markov Decision Process is defined as:

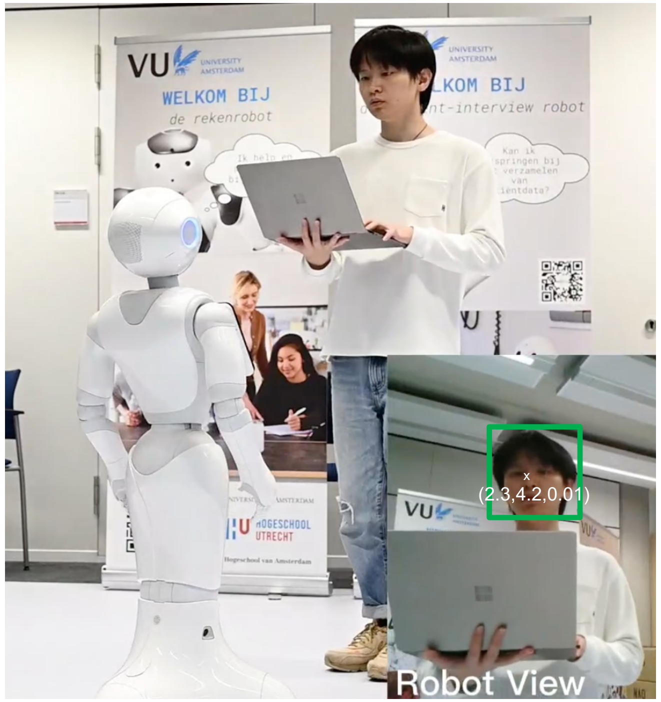
- State space (S) (state variables can include features of the robot, environment, human, or a combination of the above)
- Action space (A) (actions have to be directly controllable by the robot, and assumed to be the same regardless of the state)
- Transition function T(s,s',a) = P(s'|s,a) (probabilistic)
- Reward function R(s,s',a) (sometimes R(s',a) or simply R(s') – basically defines the task to optimize for)



A policy is a full mapping from states to actions $\pi(a|s)$. A policy can be deterministic or probabilistic. Solving an MDP requires finding an optimal policy π^* that maximizes a measure of utility, e.g., total or discounted sum of rewards

Example: Pepper faces human at fixed distance

- State space (S) Position of human face in robot camera stream (x,y,z)
- Action space (A) Velocity vector of robot base (v_x, v_y, v_θ)
- Transition function T(s,s',a) Robot dynamics
- Reward function R(s'), e.g., $1/(1-(s' - s_{target})^2)$
- Utility: $R(s1) + \gamma R(s2) + \gamma^2 R(s3) + \gamma^3 R(s4) + \ldots$
  where $\gamma < 1$ is called discount factor



Problems with MDP formulations

- State might not be fully observable → Partially Observable MDPs (POMDP)
- It is often intractable to solve an MDP analytically through dynamic programming only

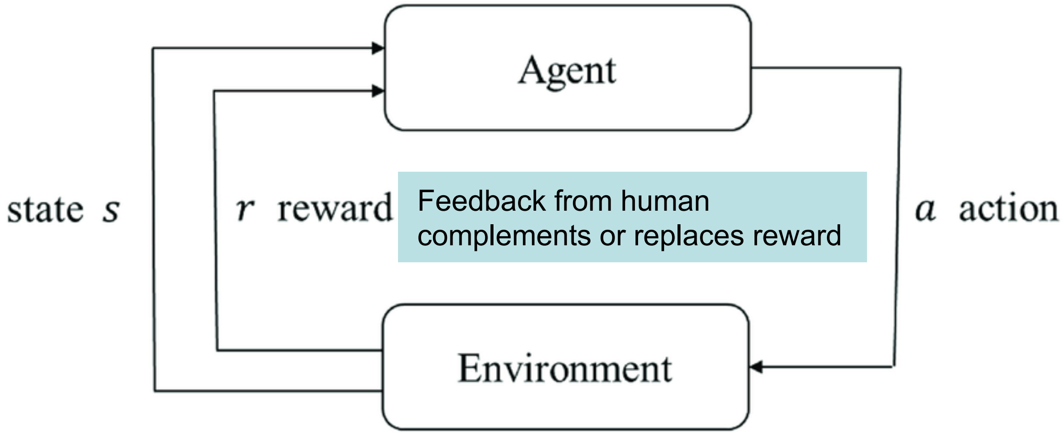
  → Reinforcement Learning (doesn't assume a known transition function)

  For RL approaches to Social Robotics, check out this paper
- There is no "correct" reward function → Reward design is an open field of research

  Alternative: Inverse reinforcement learning or Learning from demonstrations (recovering a reward function or learning an optimal policy directly from demonstrated trajectories)

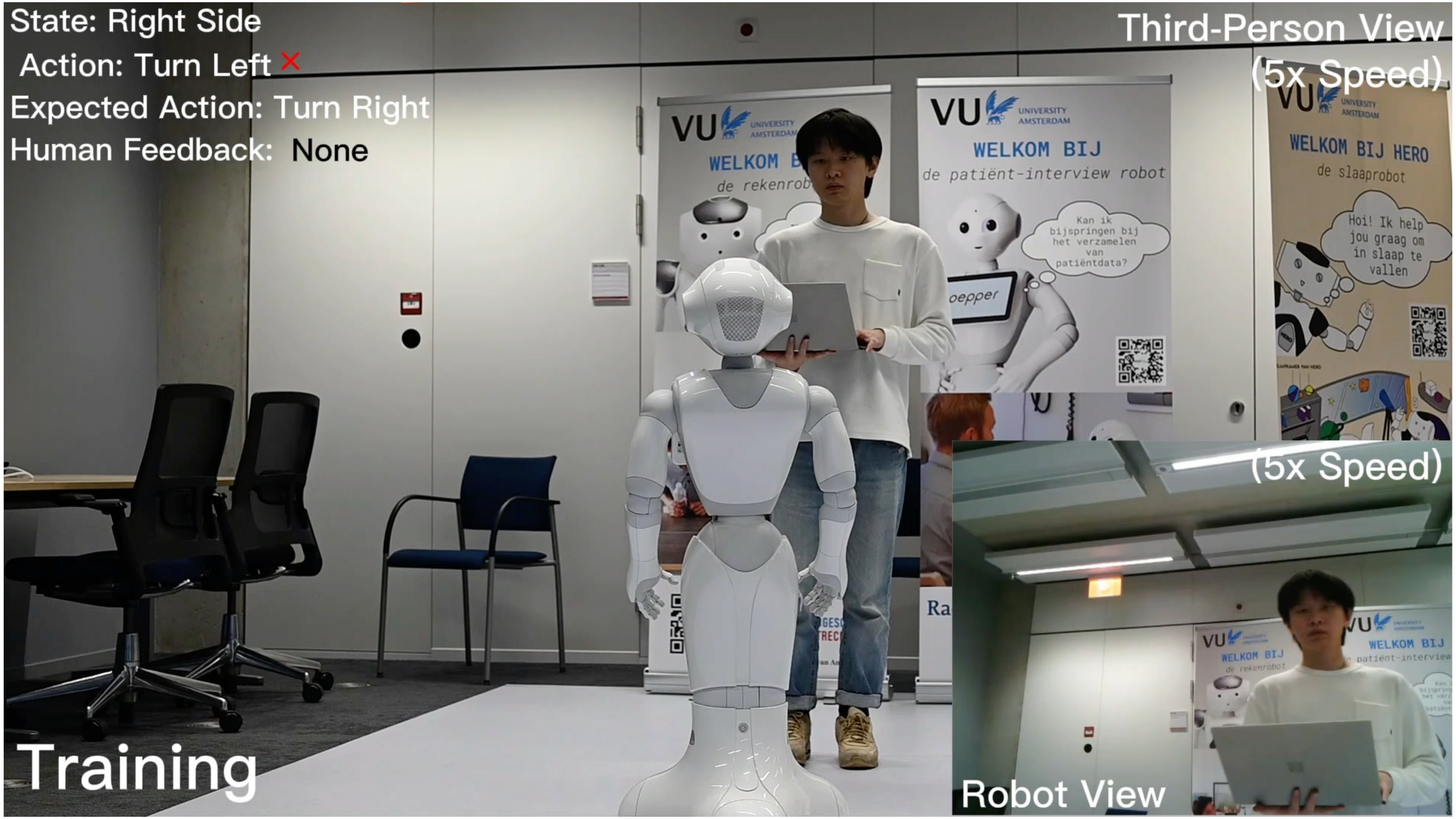
Interactive reinforcement learning

- Interactive RL is an RL problem where the reward function is partially or fully provided through human (evaluative) feedback
- Advantages: more personalized robot behaviors – helps leverage human knowledge in sparse reward scenarios
- Disadvantages: requires lots of training samples → can be tedious on the user
- For more information on interactive RL, check out this video



Interactive RL in action (credit Muhan Hou)

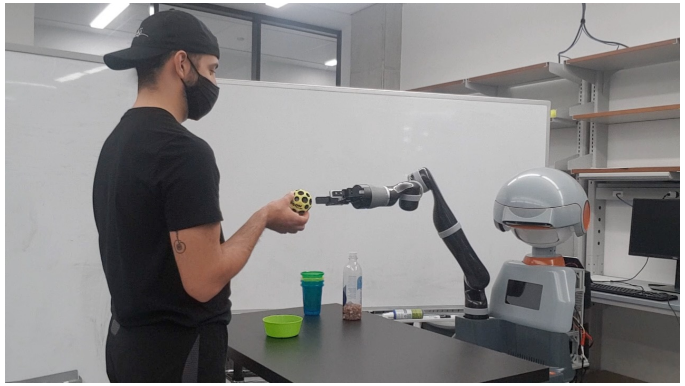
https://drive.google.com/file/d/1GnJCx1eQFPNOltJ5OuWxZXfUvQI_38Td/view?usp=sharing



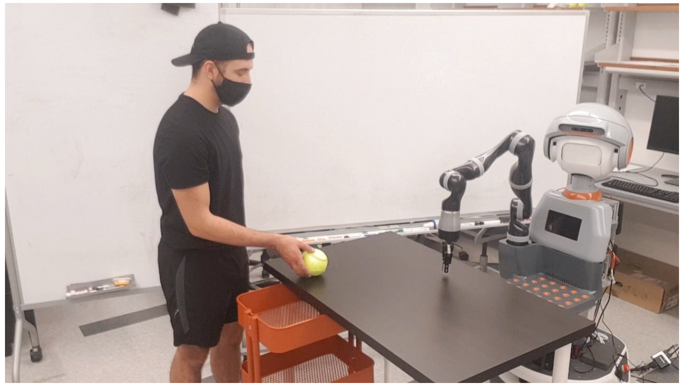
Case study: Interactive Task Learning (ITL)

Interactive task learning is the problem of learning a task purely from (social) interaction with human teachers, including the goal (or reward function), an optimal plan or policy, human preferences, etc.

Examples of tasks for ITL


Handover


Pouring


Ball shooting


Brushstrokes


Creative painting


Expressive motion

Discussion: what components are required to make ITL happen?

- Interpreting instructions
  - Pointing
  - Gaze
  - Demonstration (visual, kinesthetic, …)
  - Evaluative feedback
  - Corrective feedback
  - …
- Learning
  - Learning by imitation / from demonstrations
  - Interactive RL
  - Learning from action advice
  - Evolutionary approaches
- Planning
  - Active learning (asking questions)
  - Human-robot collaboration
  - Hierarchical skill acquisition

Inspiration from human-animal interaction?

Exploratory Design Process

Observations

Observations at a dogschool, observing three training sessions. Focus on both recognized methods from literature as new insights.

Interview

Interview with certified dog trainer on findings from observations as well as literature. Brainstorming about the possibilities and different uses.

Codesign

Codesign with two dog trainers and a dog shelter volunteer to try out the first design and explore the possibilities. Also brainstorm session on development.

Experimental study

Comparing the designed method based on dog training to a visual programming method with 18 participants.

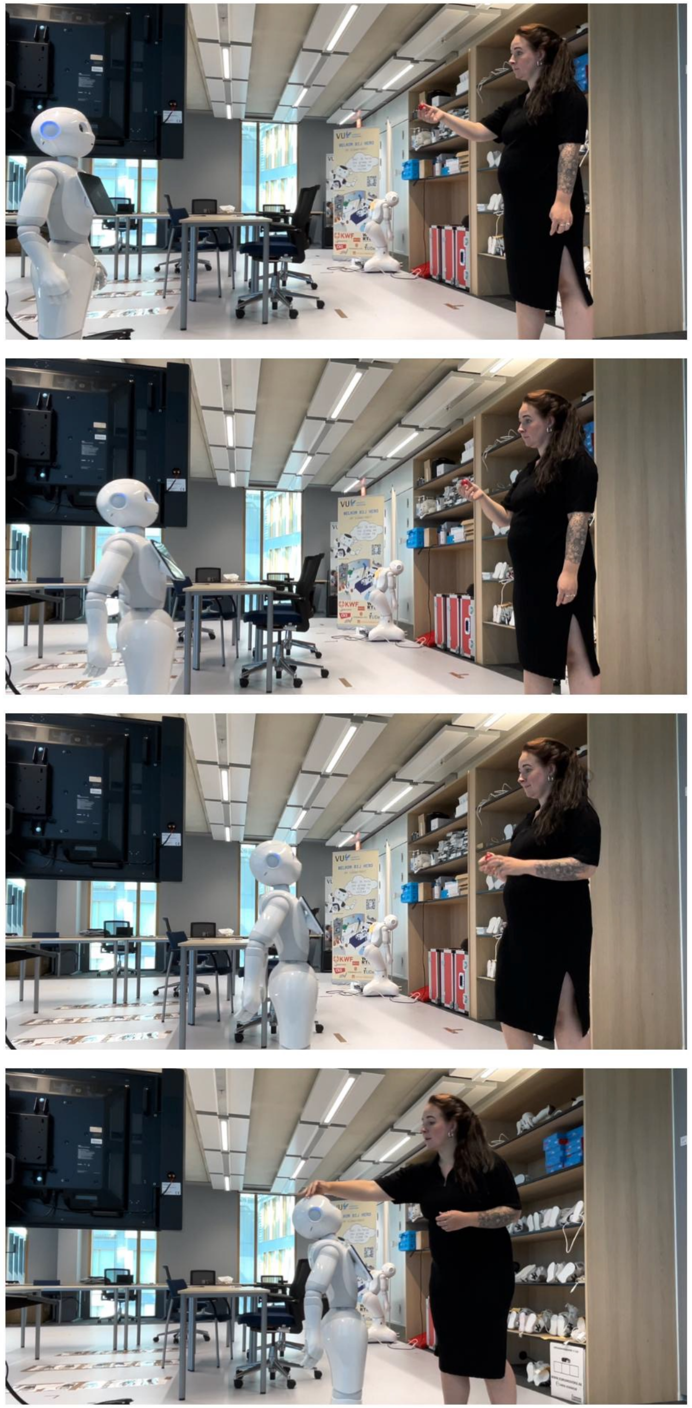
First version of design

Change of design based on codesign session

Nienke Schrage-Prent

“Follow” task

“Come” task



Discussion:

Which AI methods you expect to be useful for further developing cognitive skills of socially intelligent robots and under what assumptions?

Socially Intelligent Robotics Project (period 3)

- Apply AI techniques to endow the Pepper robot with new social interaction skills
- Choose your own project – either social cue detection or interactive robot learning
- Show a demo at the end – no user studies
- Have fun!