Cognitive Hierarchies: A Principled Approach to Cognitive Robotics

Abdallah Saffidine

Slide design: David Rajaratnam

Never Stand Still Faculty of Engineering Computer Science and Engineering

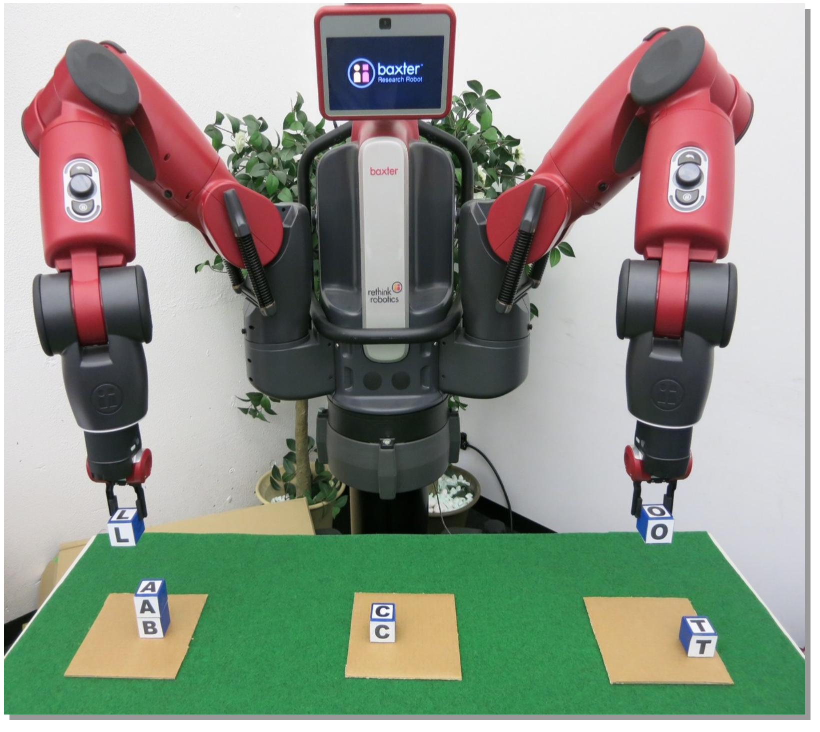
Motivation

Before we get started...

- **Cognition** is the process of acquiring knowledge through experience and reflection.
- Crucially for AI-robotics, it involves both perception and abstract reasoning.
- For robots that interact with humans it also requires abstract reasoning that match human concepts.

Building cognitive robots

- Robots that can operate and interact in human environments (e.g., domestic robots).
- Solving blocksworld problems with a Baxter robot provides a useful test-bed for cognitive robotics.



Baxter and blocksworld - video

Baxter and blocksworld - challenges

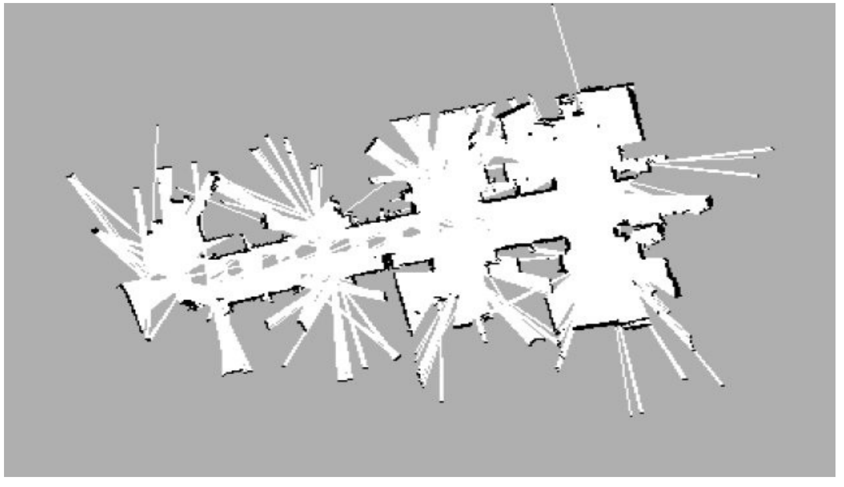
- Requires dealing with both abstract planning and low-level details (e.g., noisy sensors and partial observability).
- Can we give a principled account of such a robot's behaviour from the lowest to the highest levels of abstraction?
  - Lowest level: noisy cameras and actuators requiring visual servoing to pickup items.
  - Highest level: abstract logical representation to solve a planning problem.

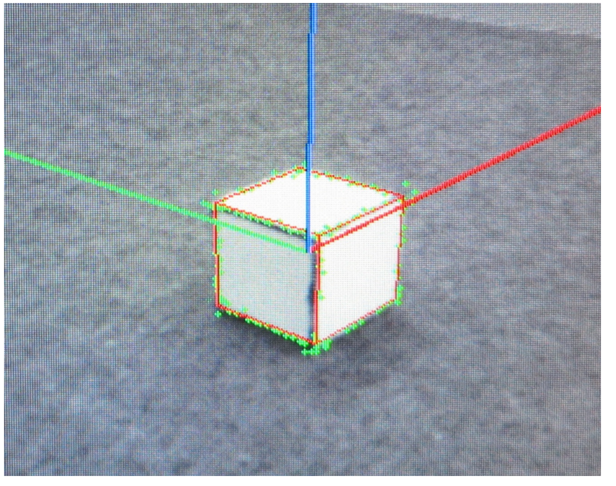
Background

The current approach

A lot of research on individual components and techniques:

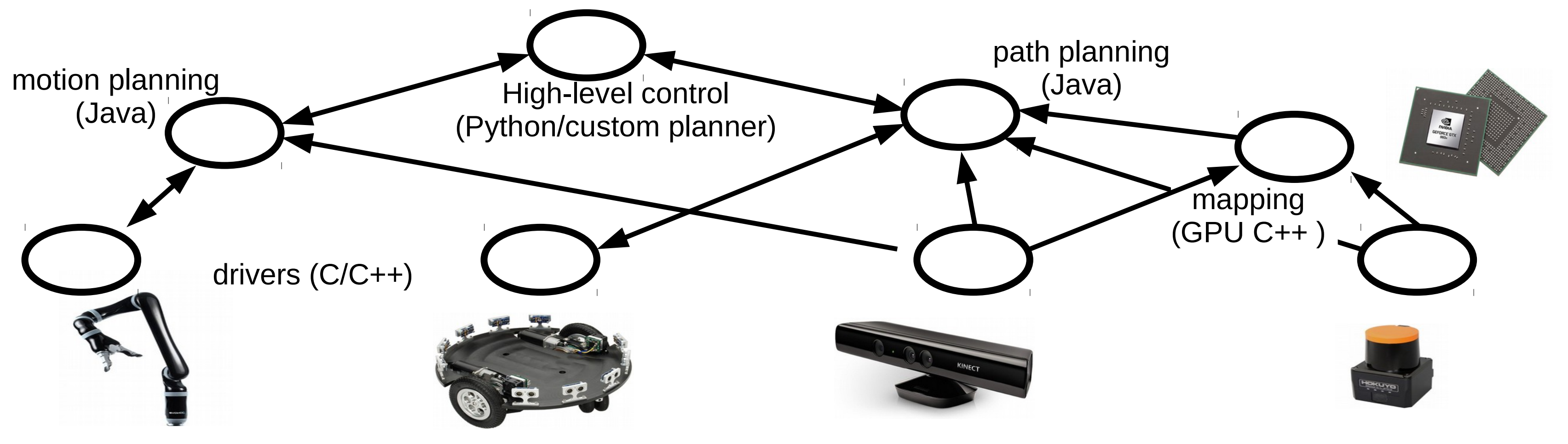
- Simultaneous Localisation and Mapping (SLAM) - builds map and localises robot based on laser scans (e.g., using Kalman filter or particle filters).
- Navigation between points on a map (e.g., A* or Dijkstra's algorithm).
- Motion planning algorithm to compute kinematic chains and control actuators.
- Robot vision uses many techniques from computer vision and machine learning.
- High-level control using reinforcement learning or classical planning techniques.



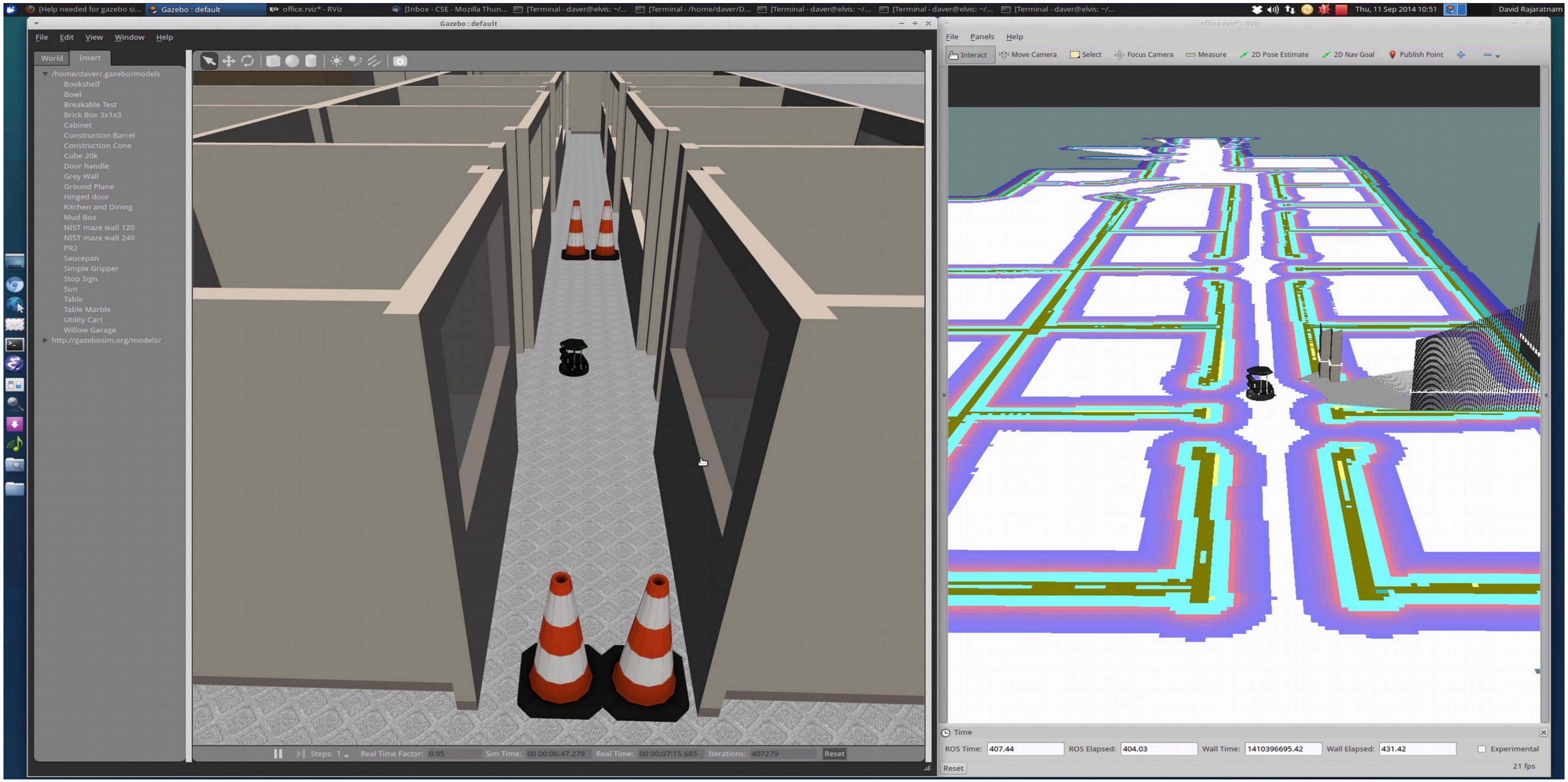


The current approach

- Combining the components is mostly viewed as a software development challenge.
- Software frameworks for building robots such as ROS (Robot Operating System).
- ROS provides communications infrastruture between *nodes*, and library of drivers, filters (e.g., mapping) and behaviours (e.g., A* for navigation).
- **Problem**: individual parts may have well-understood properties (e.g., kalman filter or A*) but system as a whole is poorly understood.



Navigation problems - video



- How should the robot behave? … trick question.

Work on robot architectures

- Orthogonal to software approach is cognitive/robot architectures.
- Two categories:
  - Fixed languages:
    - Cognitive architectures (e.g., SOAR, ACT-R) – symbolic representation.
    - Nilsson's triple-tower – symbolic representation.
    - Robot *dual dynamic* (DD) hierarchy – differential equations.
  - Informal architectures:
    - Real-time Control System (RCS) – text description and diagrams.
    - *Subsumption architecture* uses finite state machines to represent individual levels, but interaction between levels is informal.
    - Note: attempts to connect SOAR to real robots typically fall into this category.
- **The research gap**: fixed language approaches lack flexibility, and informal approaches cannot be used to establish properties of the system as a whole.

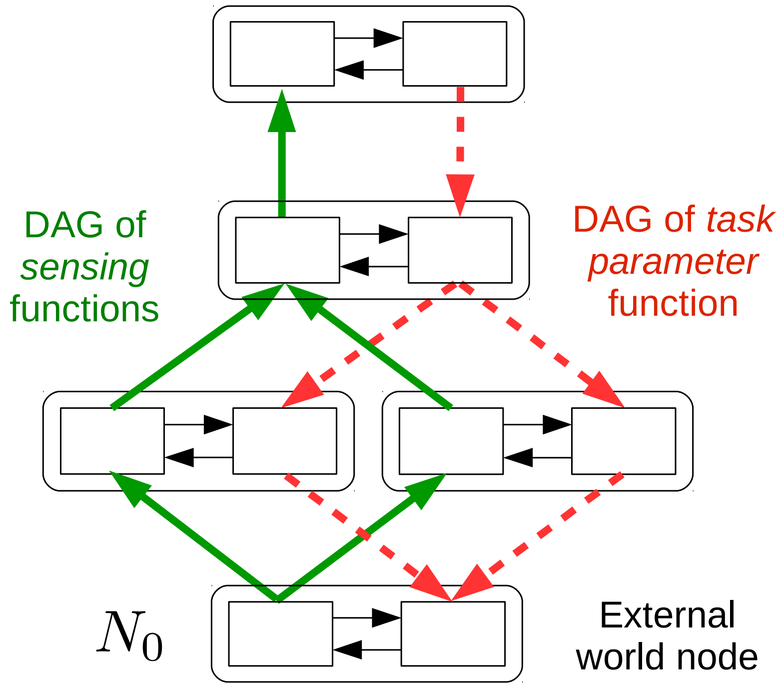
A better way

What are we want?

- A formal approach so we can establish system level properties.
- Flexible: does not commit to a particular representation.
- Solution:
  - Formalisation of disparate sub-systems as nodes in a hierarchy.
  - Within node: formalised in terms of functions over domains.
  - Between nodes: formalised in terms of the application of the node functions in some well-defined manner.

Cognitive hierarchy meta-theory

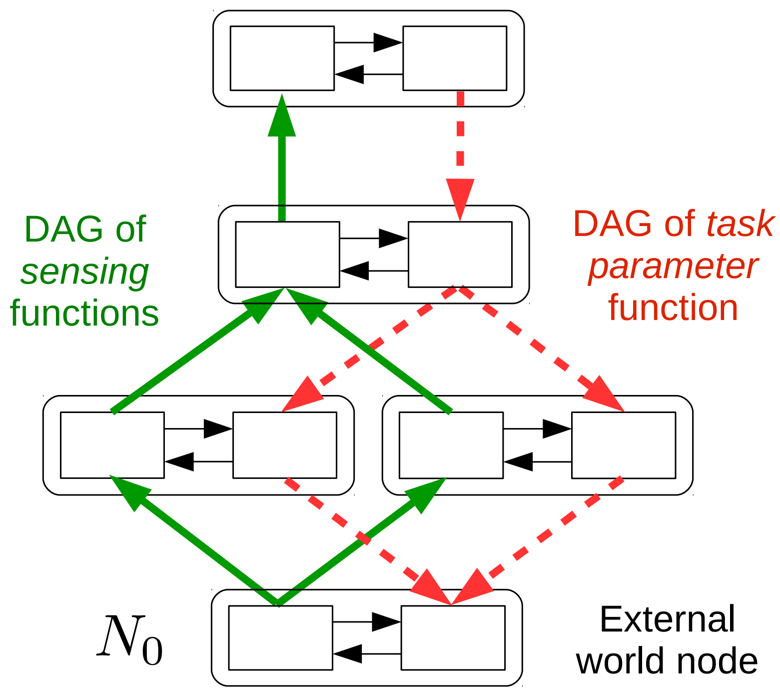
Formalises the interaction between nodes, using a *directed acyclic graph* (DAG), while leaving internal node representation details open.

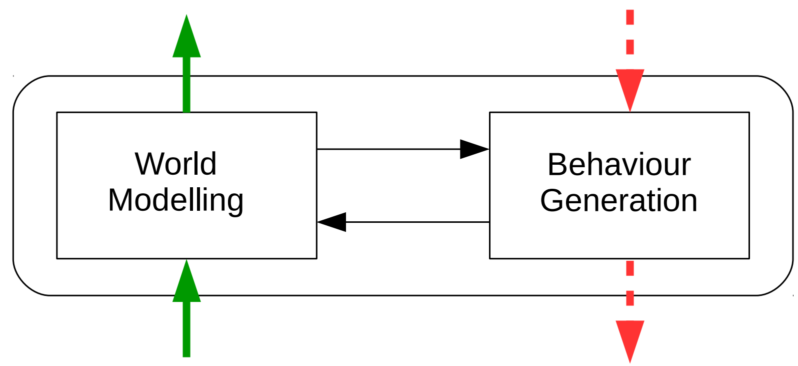


- Distinguished node 0 represents the external world:
  - Input/output to the node is specified, but
  - internal details (belief states and behaviour) remains opaque.

Cognitive hierarchy meta-theory

Formalises the interaction between nodes, using a *directed acyclic graph* (DAG), while leaving internal node representation details open.

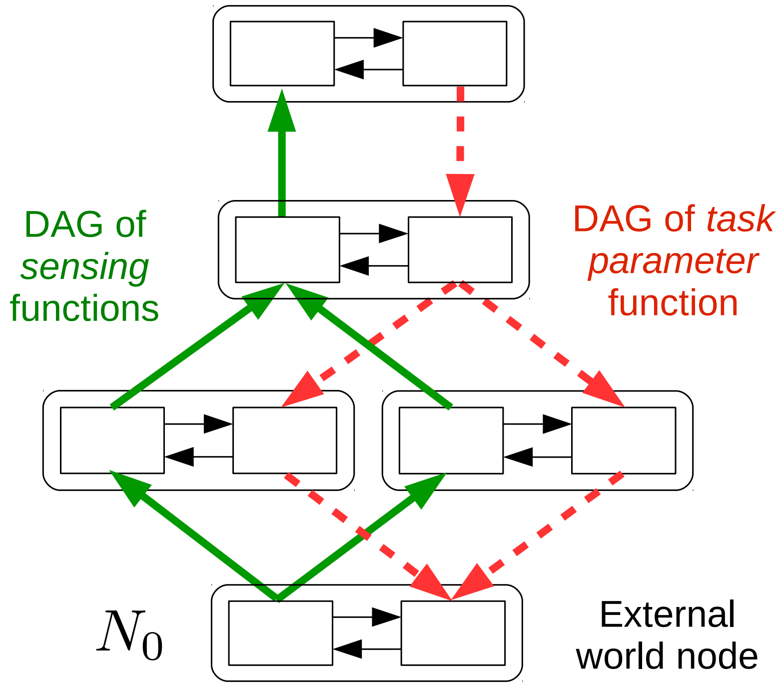


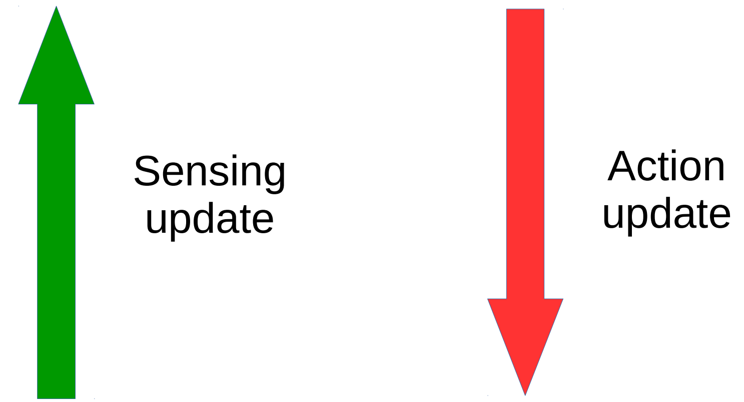


- Belief state updated by sensing or *expectation* update from actions.
- Actions from higher-level nodes determine the policy function for the current node.
- Policy function maps belief states to actions.

Cognitive hierarchy meta-theory

Formalises the interaction between nodes, using a *directed acyclic graph* (DAG), while leaving internal node representation details open.





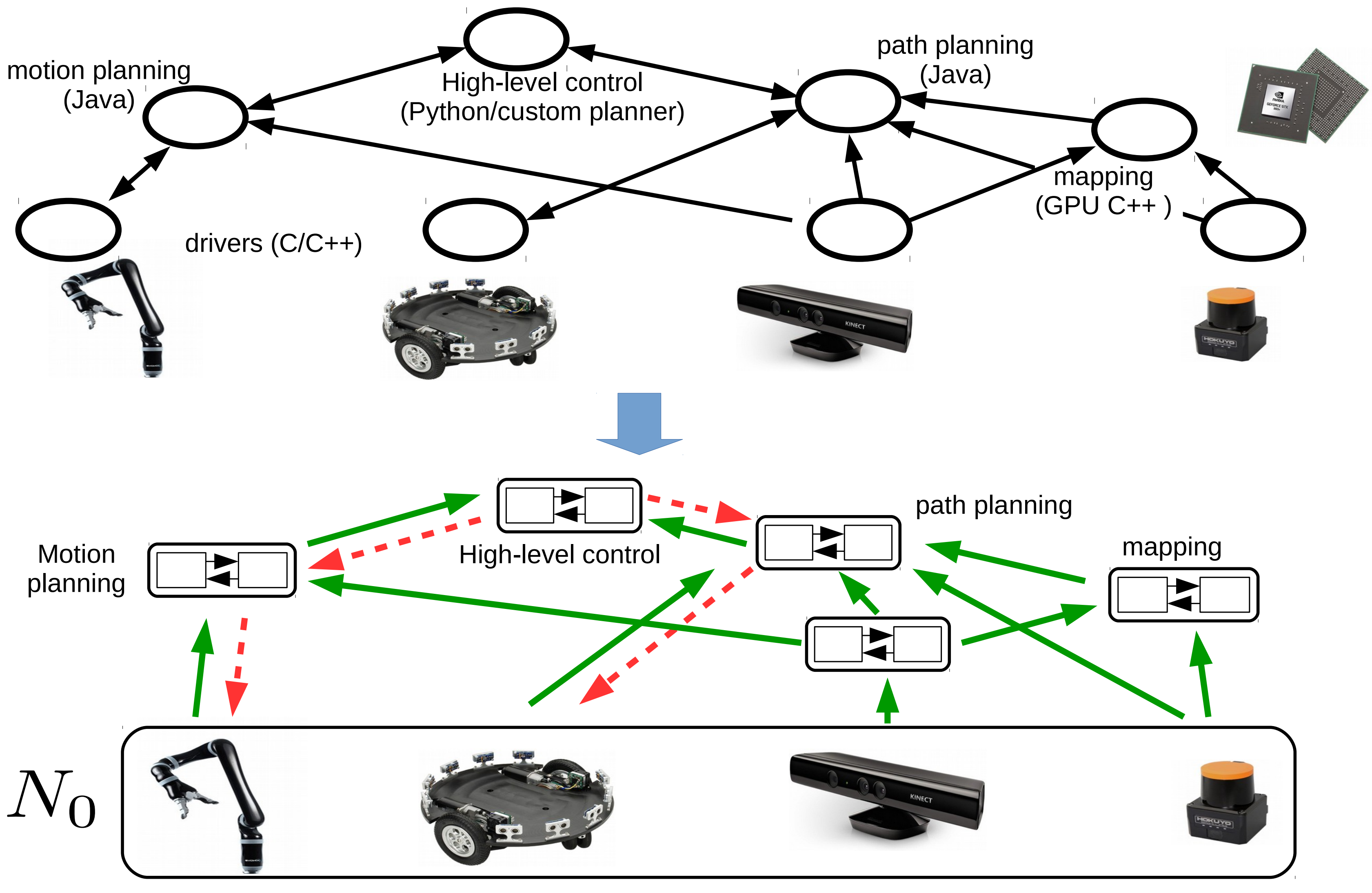
- *Process update model* to update the cognitive hierarchy
- Sensing update followed by action update.
- Update is formally well-defined.

Implementing cognitive hierarchies

Complex robot systems can be implemented based on the cognitive hierarchy meta-theory (with some effort):

- Not necessarily a one-to-one correspondence between ROS nodes and cognitive nodes.
- Deciding if a driver should be part of the external world node; e.g., if it maintains state for filtering then maybe better as a separate cognitive node.
- Providing theoretical structure to ROS nodes; separating belief modelling from behaviour generation.
- Enforcing DAG structures (usually easy) and process model synchronisation (less easy).

Back to the example



Why is this interesting?

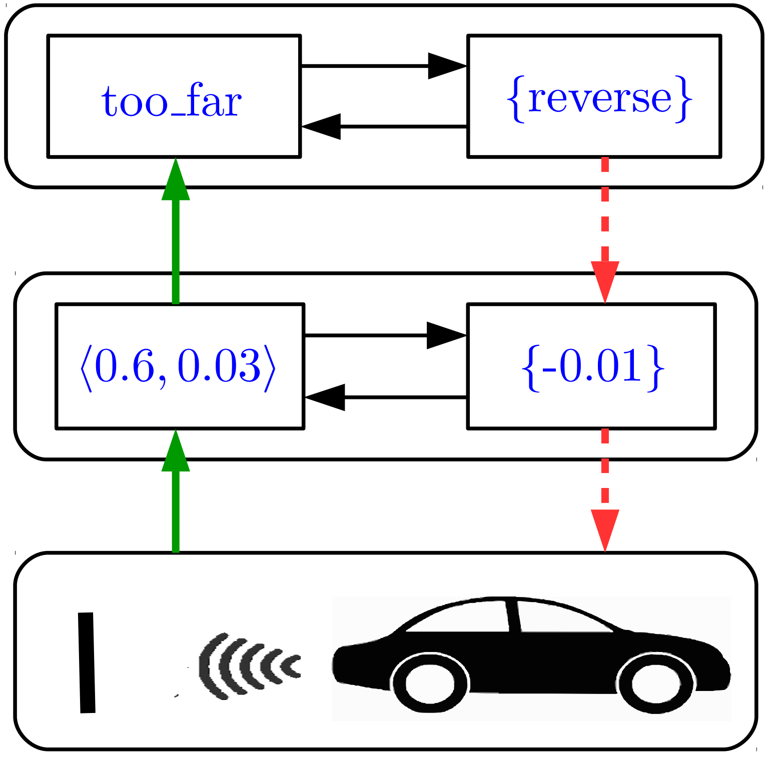
Guarantee system-level behaviour

- Could establish that an automated reversing system will never hit the wall.

Symbolic reasoning

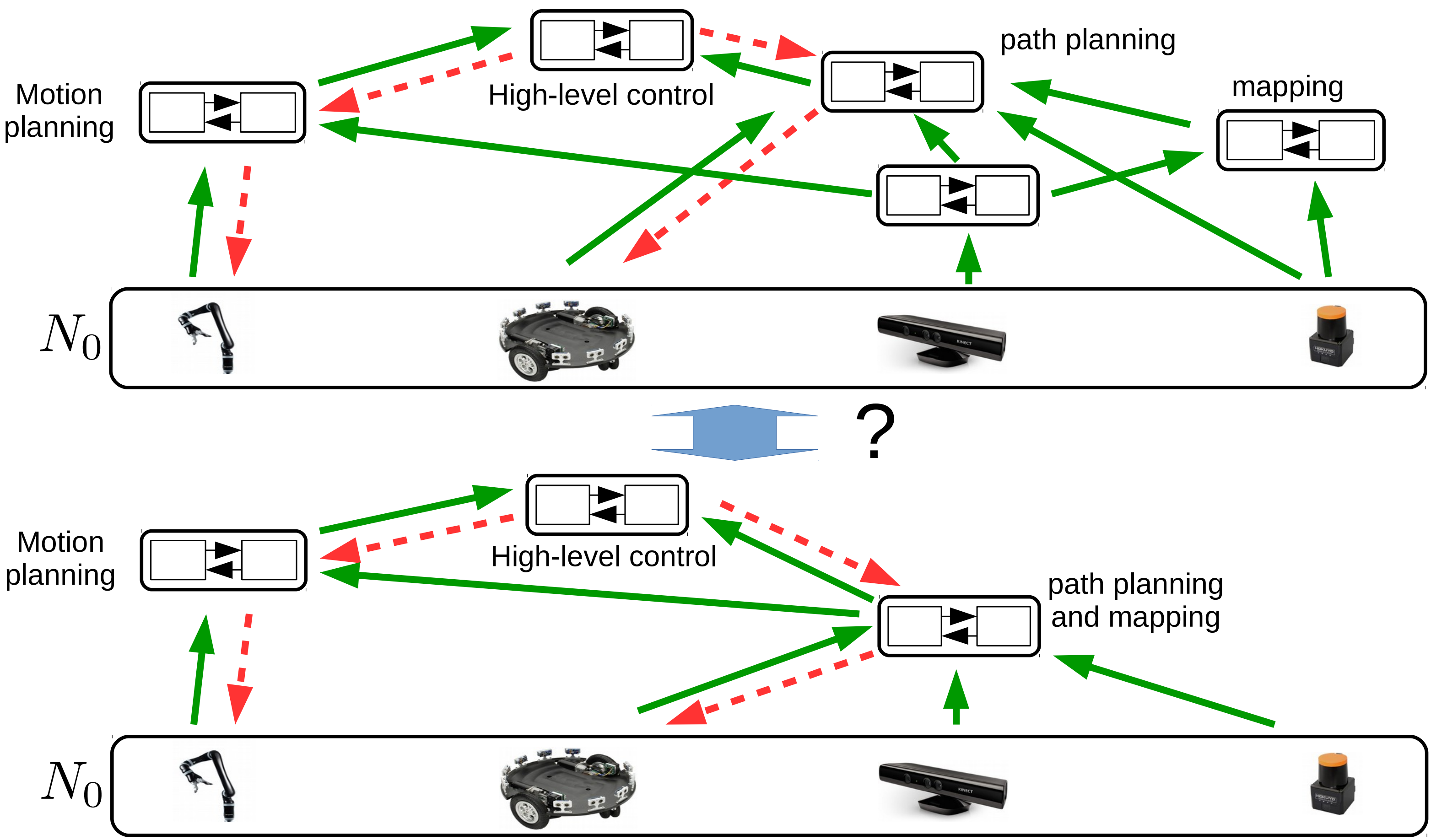
Probabilistic reasoning: combines readings from camera and laser sensor

External-world with camera and laser sensors and motor controller



Establish relationships

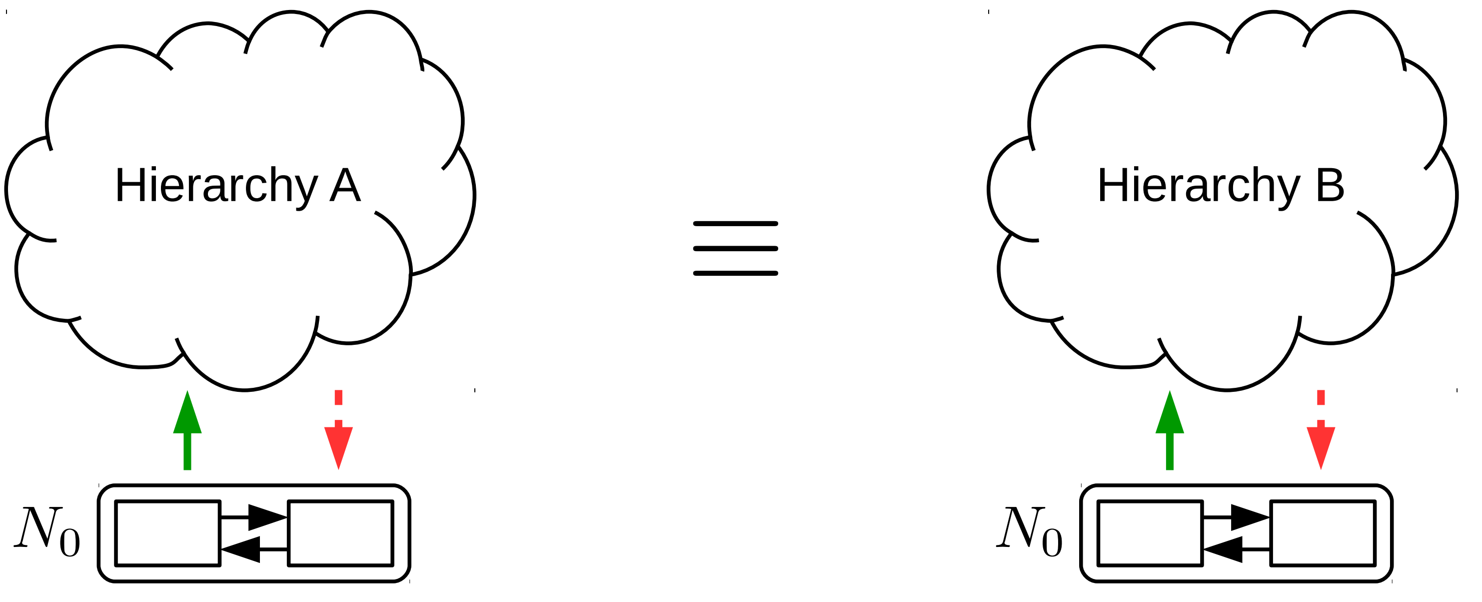
Can we say anything about the relationship between two cognitive hierarchies?



Behaviour equivalence

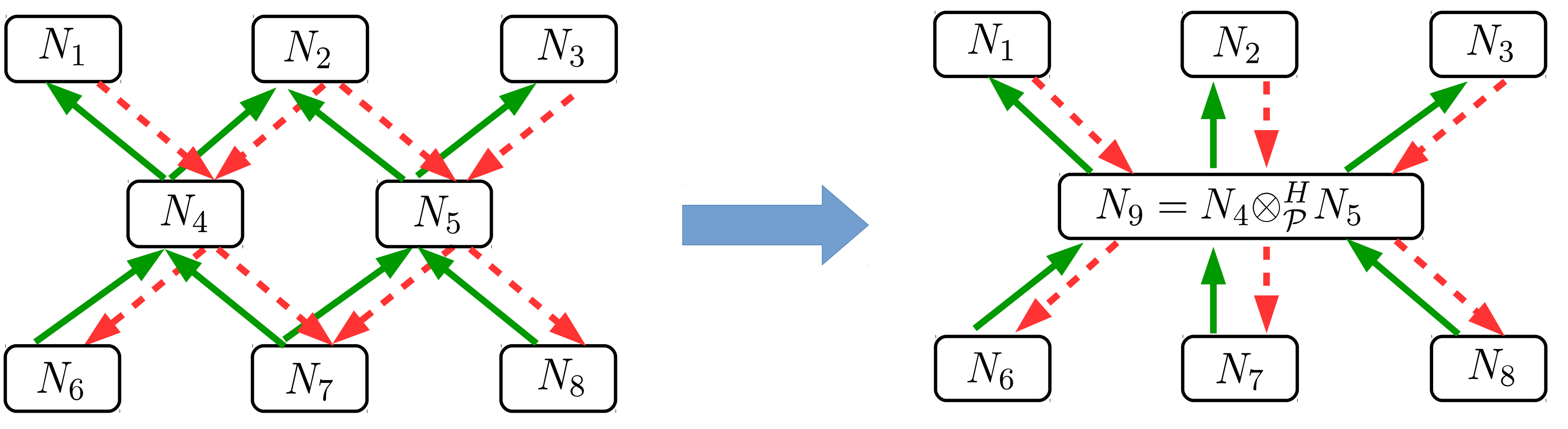
A formal definition of equivalence w.r.t. N_0 :

- Given the same sensing inputs the two hierarchies generate the same actions for N_0.
- Must produce identical behaviour over time.
- Theoretical concept: cannot be established experimentally.



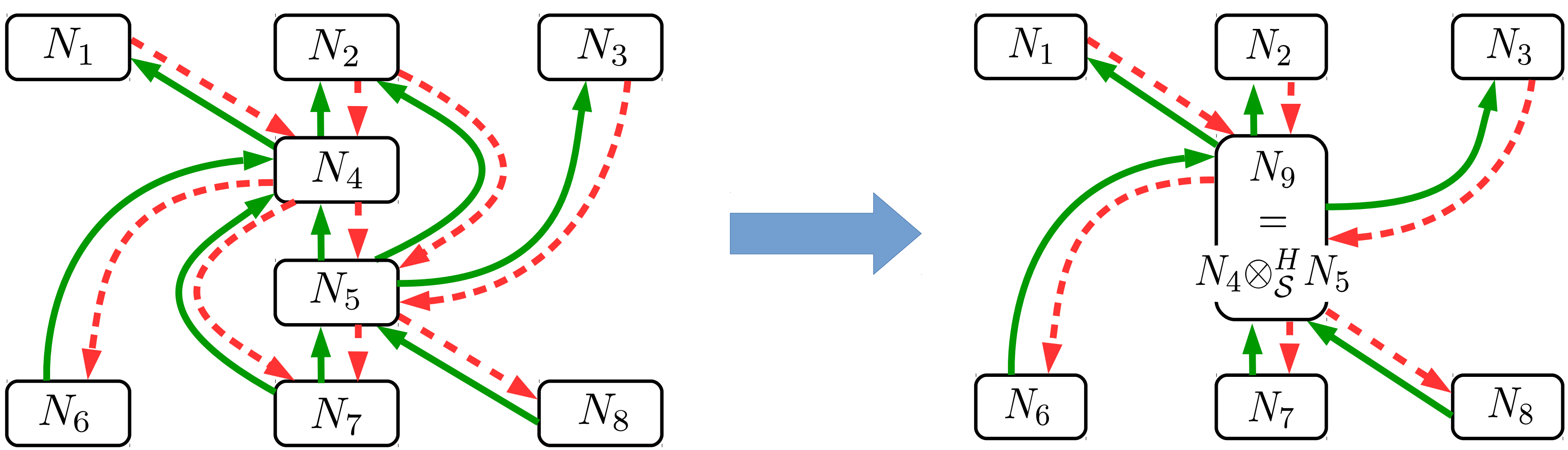
Parallel node composition

- Define a parallel node composition operator that guarantees behaviour equivalence of the resulting cognitive hierarchy (w.r.t. process update model).
- Condition: when two nodes are unrelated in the partial ordering induced by the dependency graph.



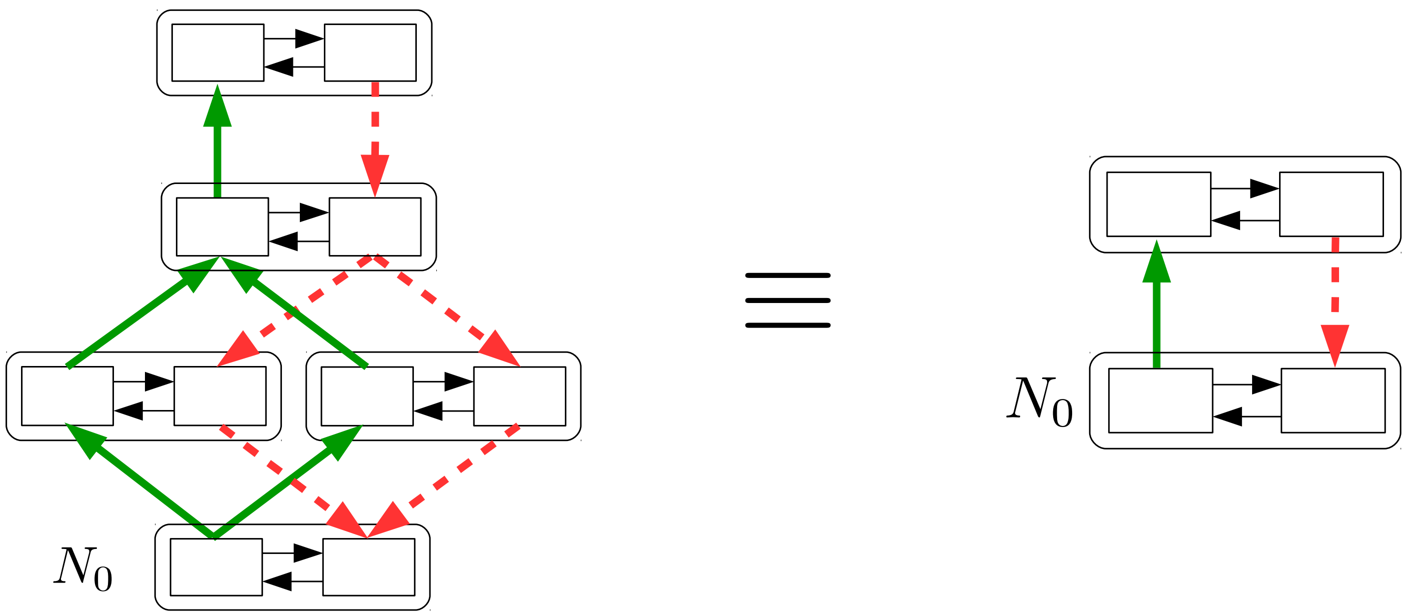
Sequential node composition

- Sequential node composition operator guarantees behaviour equivalence of the resulting cognitive hierarchy (w.r.t. process update model).
- Condition: when two nodes are related in the partial ordering induced by the dependency graph but there is no node between them.



Formal Properties

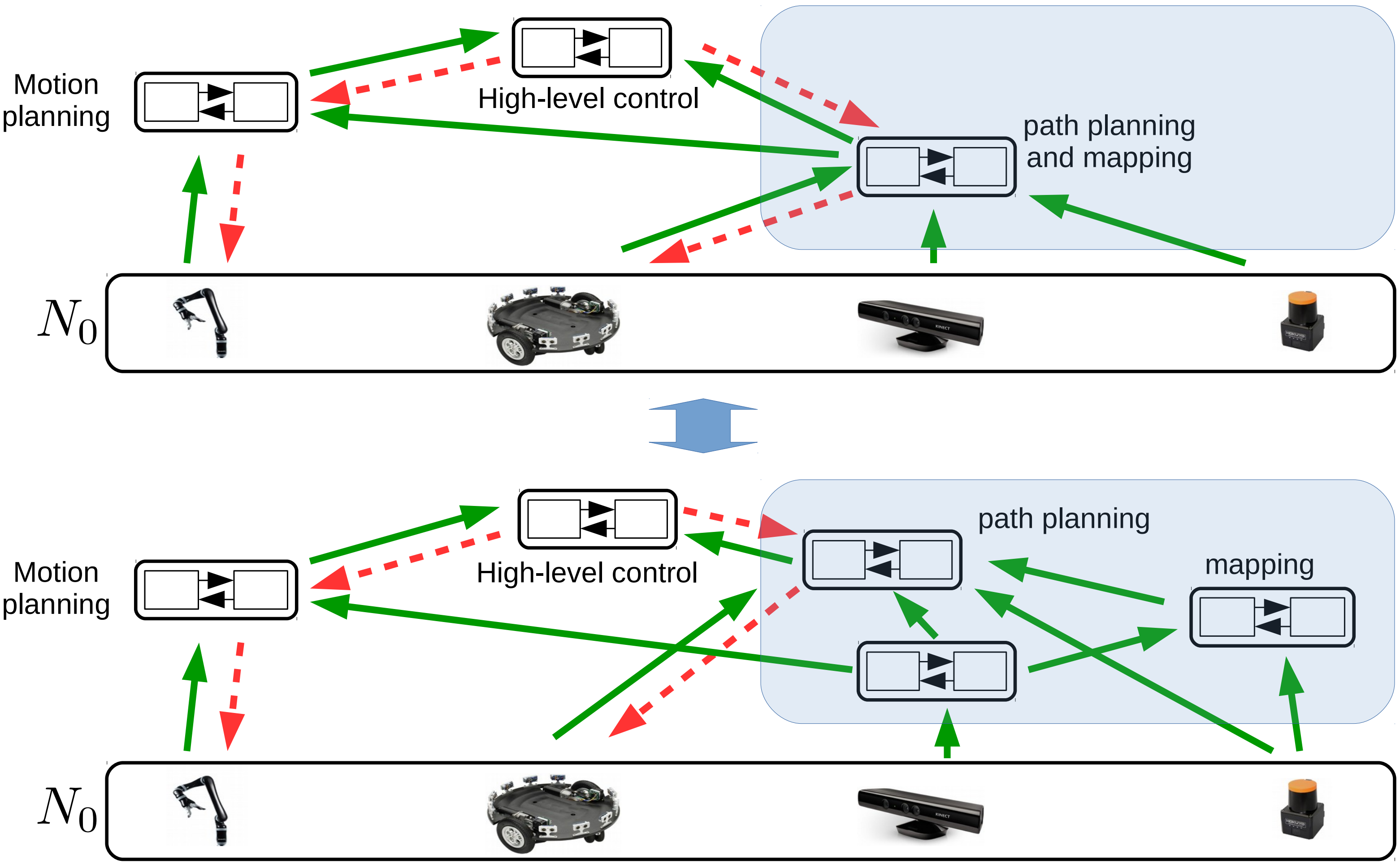
Every cognitive hierarchy is behaviourally equivalent to a hierarchy with two nodes (inclusive of N_0).



- Can be shown constructively by iteratively applying either the sequential or parallel composition operators where possible.
- The fact that this is possible reduces to the lack of cycles in the graph.
- The combination of sequential and parallel composition are general enough to capture all (provably) behaviour equivalent compositions.

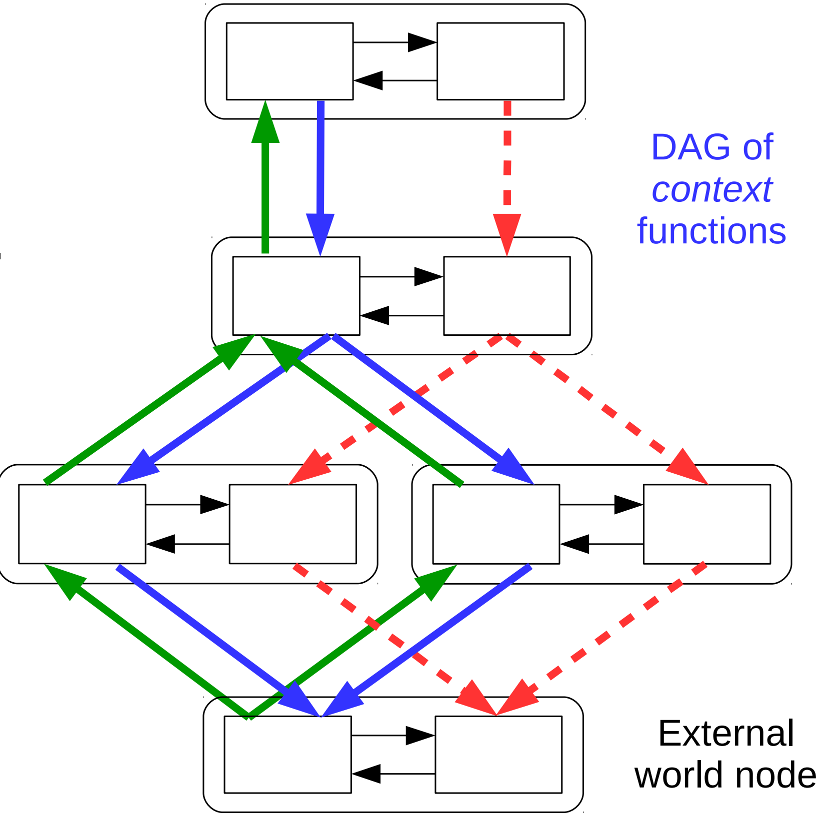
Back to the example

We can establish behaviour equivalence if we can show a composition relationship between the nodes.



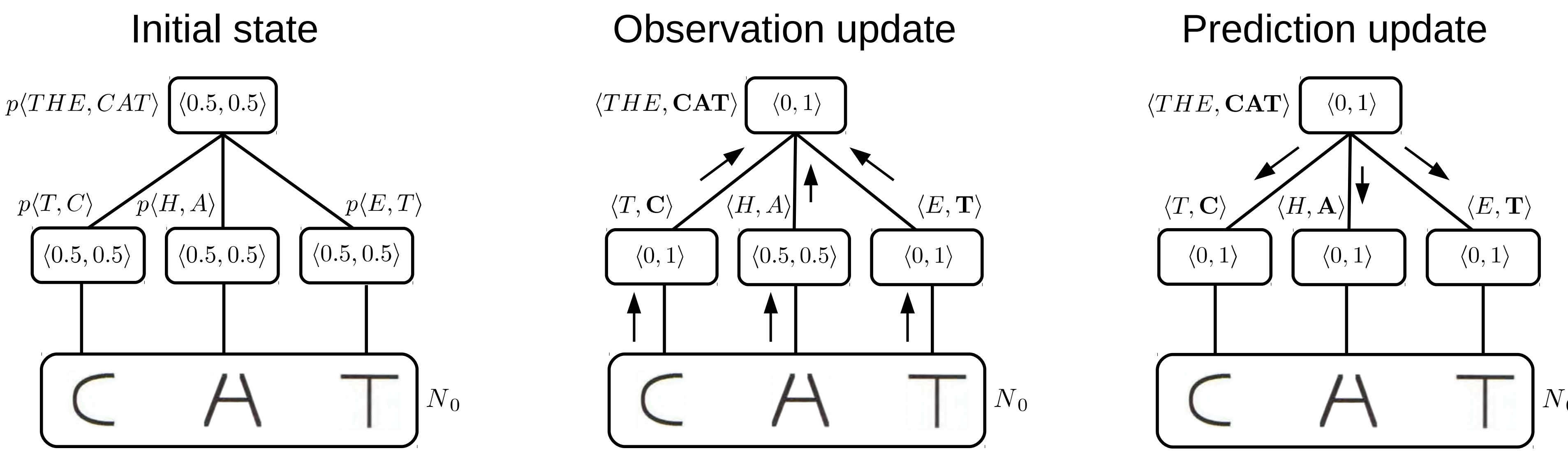
Adding context to the hierarchy

- **Intuition**: what we perceive about the world is both a product of what we expect to see as well as a what we actually see.
- **Observation update**: belief state is modified from below through sensing.
- **Prediction update**: Belief state is modified with context from above and calculated actions.



Capturing Bayesian networks

- A context enriched hierarchy adds more modelling flexibility.
- Can capture Pearl's causal trees (a form of Bayesian network).
- Example: disambiguating  in 

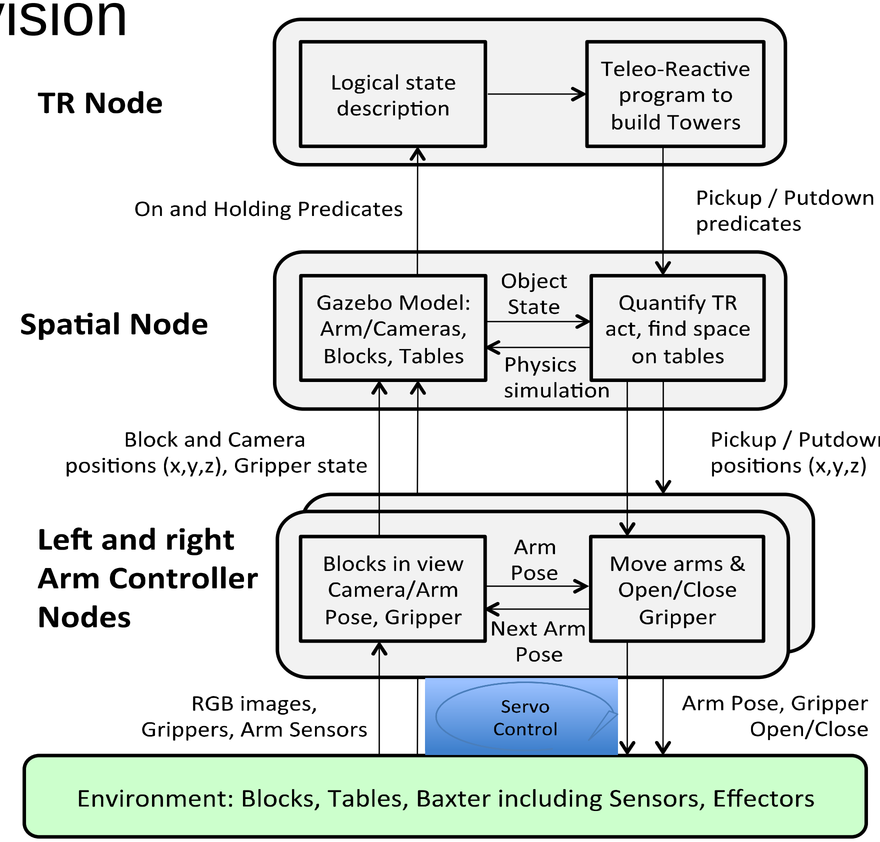


- Can add actions to Bayesian networks... in a principled way.
- Also allows integration with other formalisms.

How are we using this to solve blocksworld?

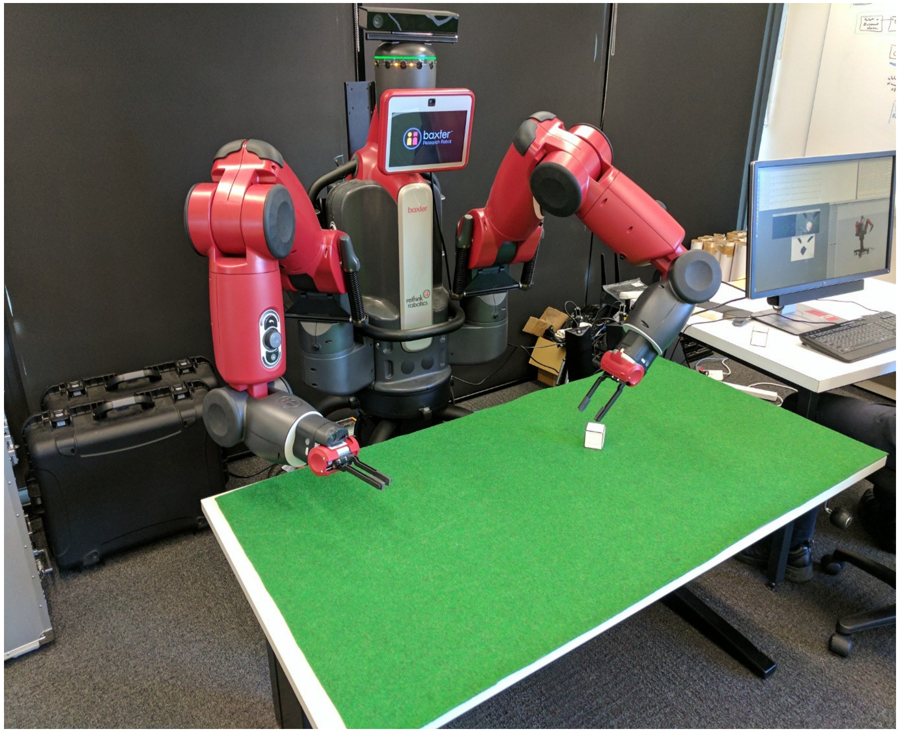
Integrating different components

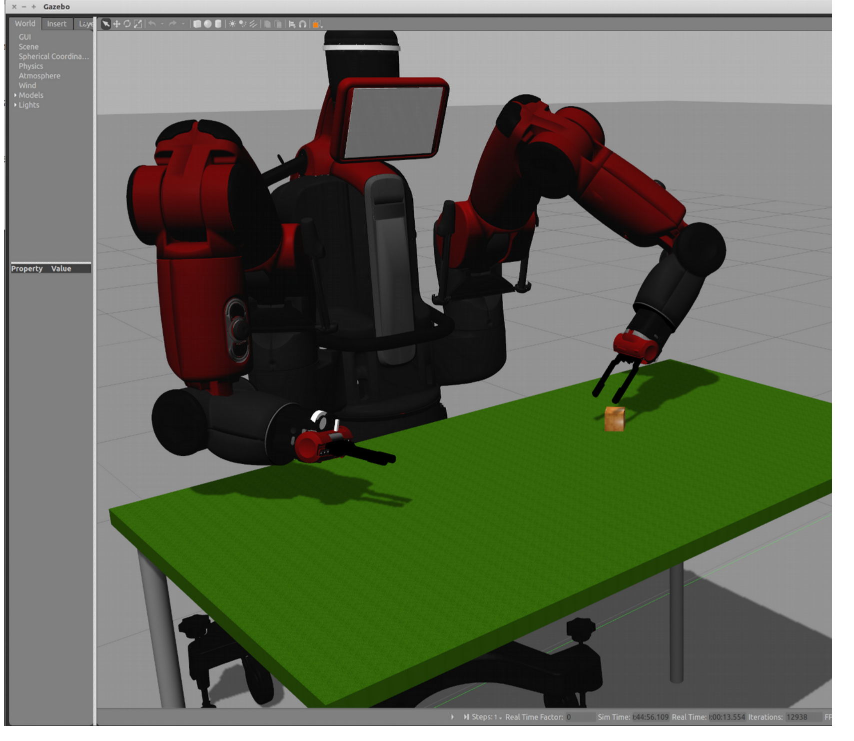
- Node for high-level actions using Teleo-reactive program or planner
- Node for spatial modelling using Gazebo physics engine
- Nodes to control arm actuation and vision



Physics engine integration

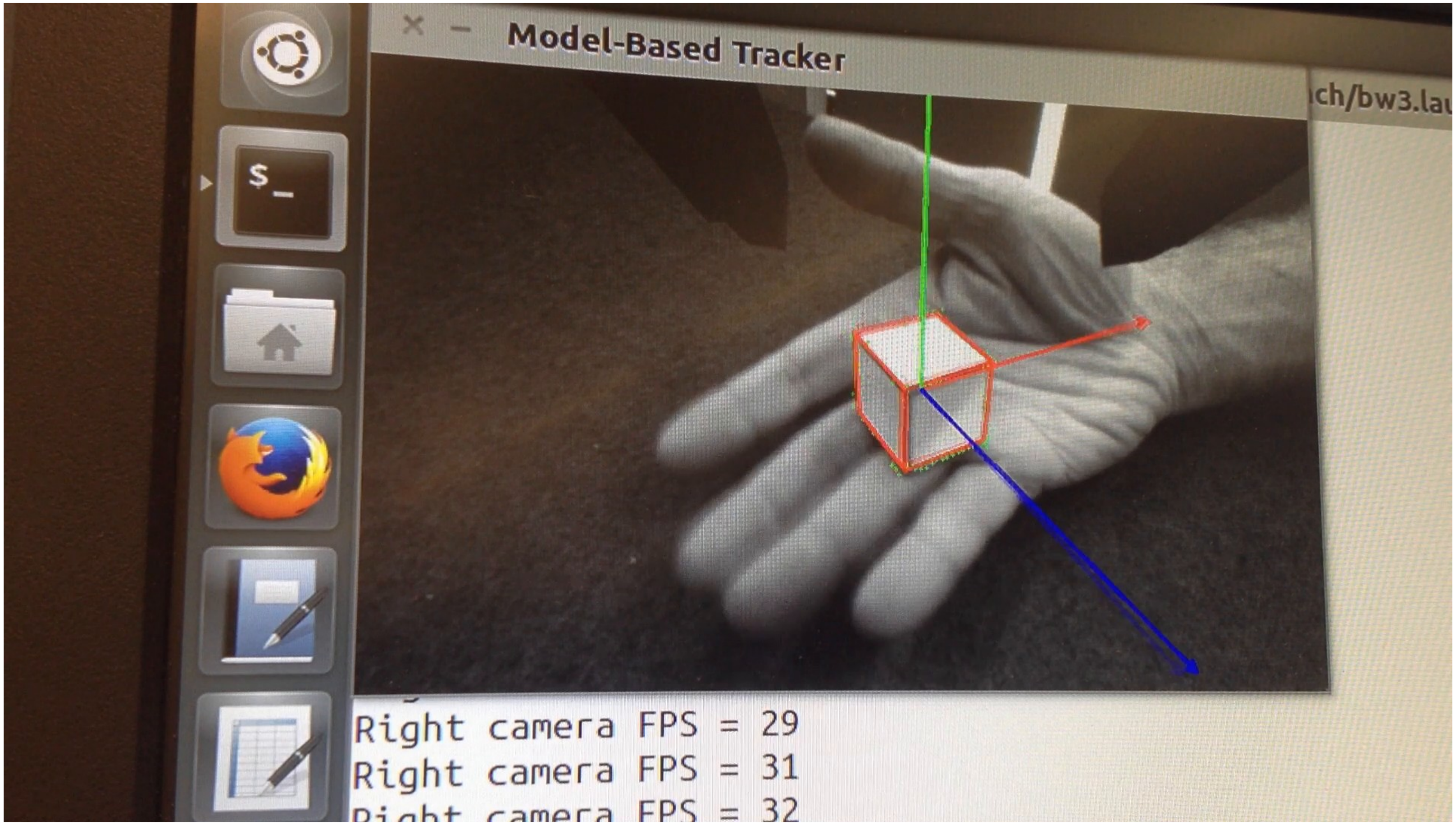
- Provides a “minds eye” view: spatial model of objects of interest as well as Baxter itself.
- Can extract high-level symbolic facts based on physics engine state.





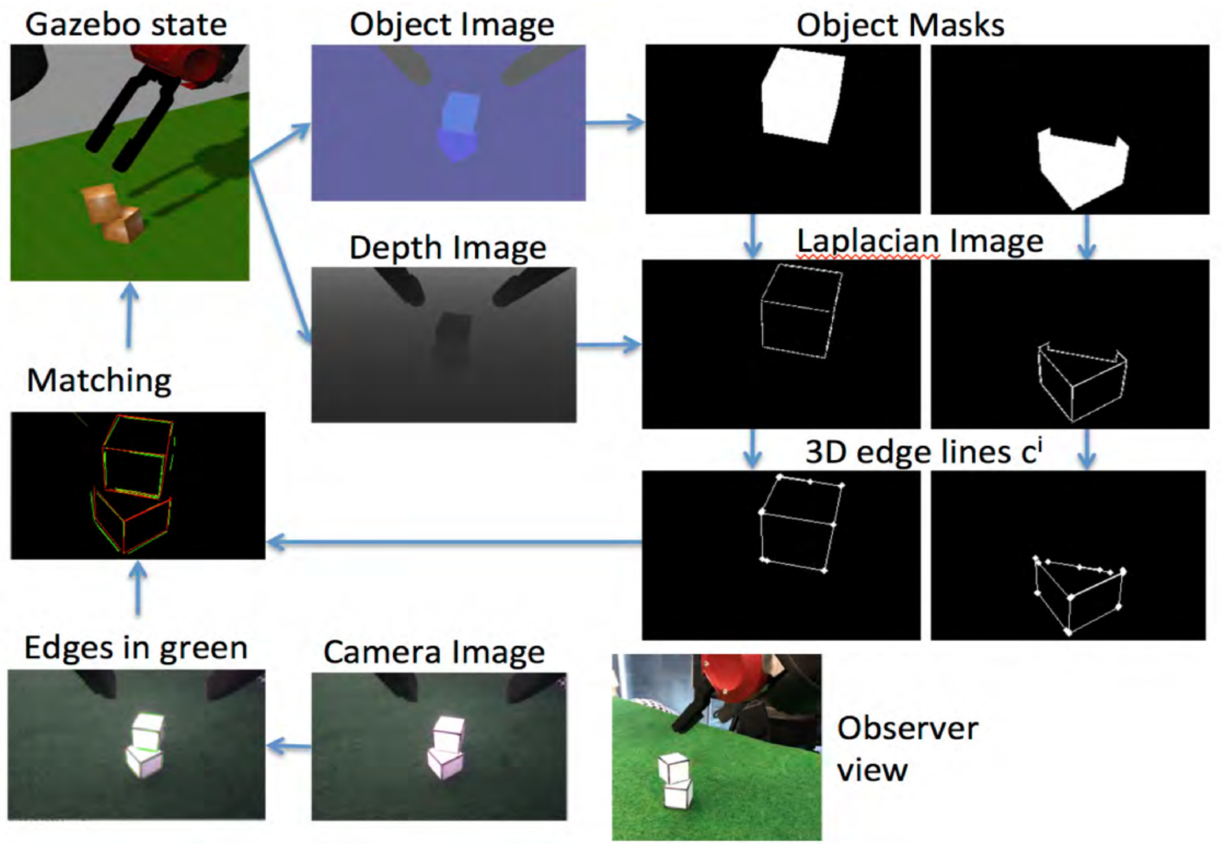
Going further

Better tracking



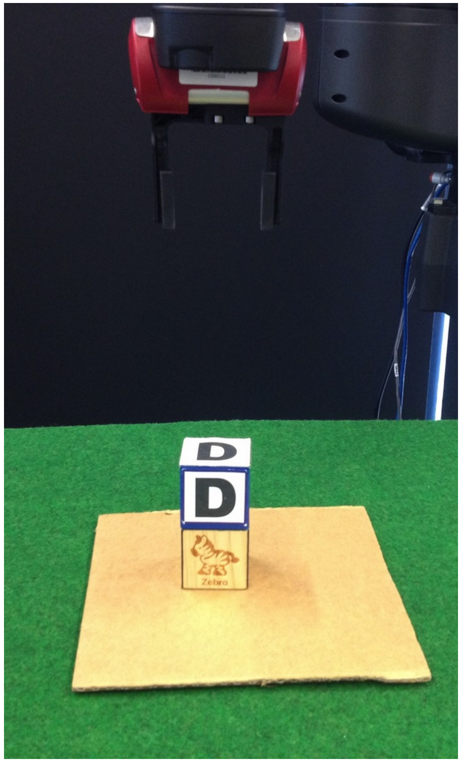
Richer physics engine integration

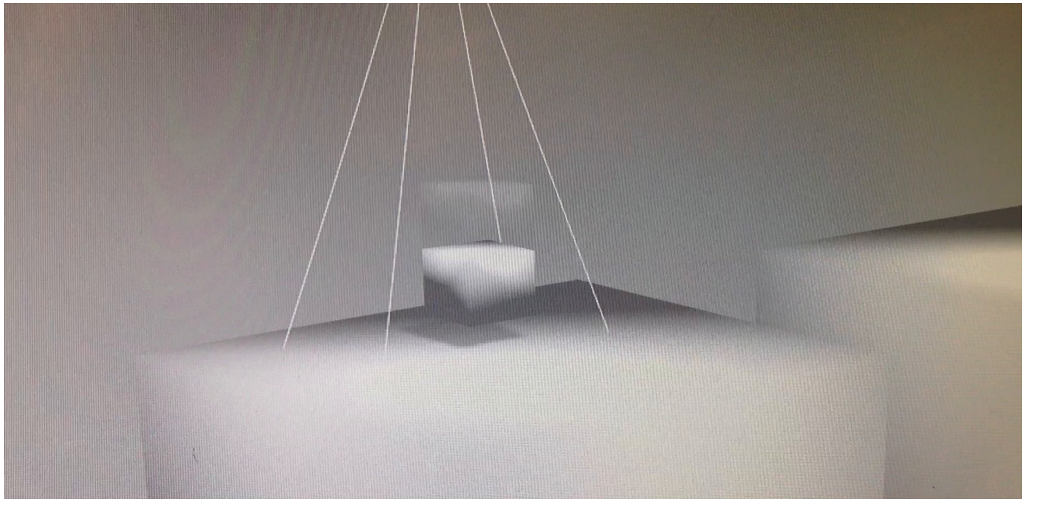
- Using gazebo state to provide context for vision system using an object-aware virtual depth camera.
- Goal: enable fast object tracking even with occlusion.



Richer physics engine integration

- Perform reasoning about partial observations.
- Something must be underneath the “D” block.





Richer physics engine integration

- Put the human into the gazebo model
- Currently using Microsoft Kinect skeleton tracking
- Potentially combine with other sensors

https://www.youtube.com/watch?v=PFAU96AKH-4&t=5s

Conclusion

- Formalised a meta-theory for constructing cognitive hierarchies.
  - Captures the interaction between nodes defined using arbitrary representations.
  - Flexible but also has well-defined properties.
- **Opinion**: this model (or something like it) is a crucial missing link for cognitive robotics – we need principled methods to deal with systems as they become more complicated.
- Lots of work to do both in terms of theory and practice.

David Rajaratnam

Bernhard Hengst

Maurice Pagnucco

Claude Sammut

Michael Thielscher

{daver,bernhardh,morri,claude,mit}@cse.unsw.edu.au