Intuitive robotic programming using string diagrams

Blake S. Pollard
NIST

Ongoing work with: Angeline Aguinaldo (UMD), Spencer Breiner, and Eswaran Subrahmanian

Motivation

- An intuitive high-level robot behavior specification language
- Interoperability among multiple robot types and manufacturers

Motivation

- An intuitive high-level robot behavior specification language
- Interoperability among multiple robot types and manufacturers

Elements enabling existing approaches:

Motivation

- An intuitive high-level robot behavior specification language
- Interoperability among multiple robot types and manufacturers

Elements enabling existing approaches:

- Canonical Robot Command Language (CRCL)
- Core Ontology for Robotics and Automation (CORA)
- Robot Operating System (ROS)

Motivation

- An intuitive high-level robot behavior specification language
- Interoperability among multiple robot types and manufacturers

Elements enabling existing approaches:

- Canonical Robot Command Language (CRCL)
- Core Ontology for Robotics and Automation (CORA)
- Robot Operating System (ROS)

The role of category theory?

Motivation

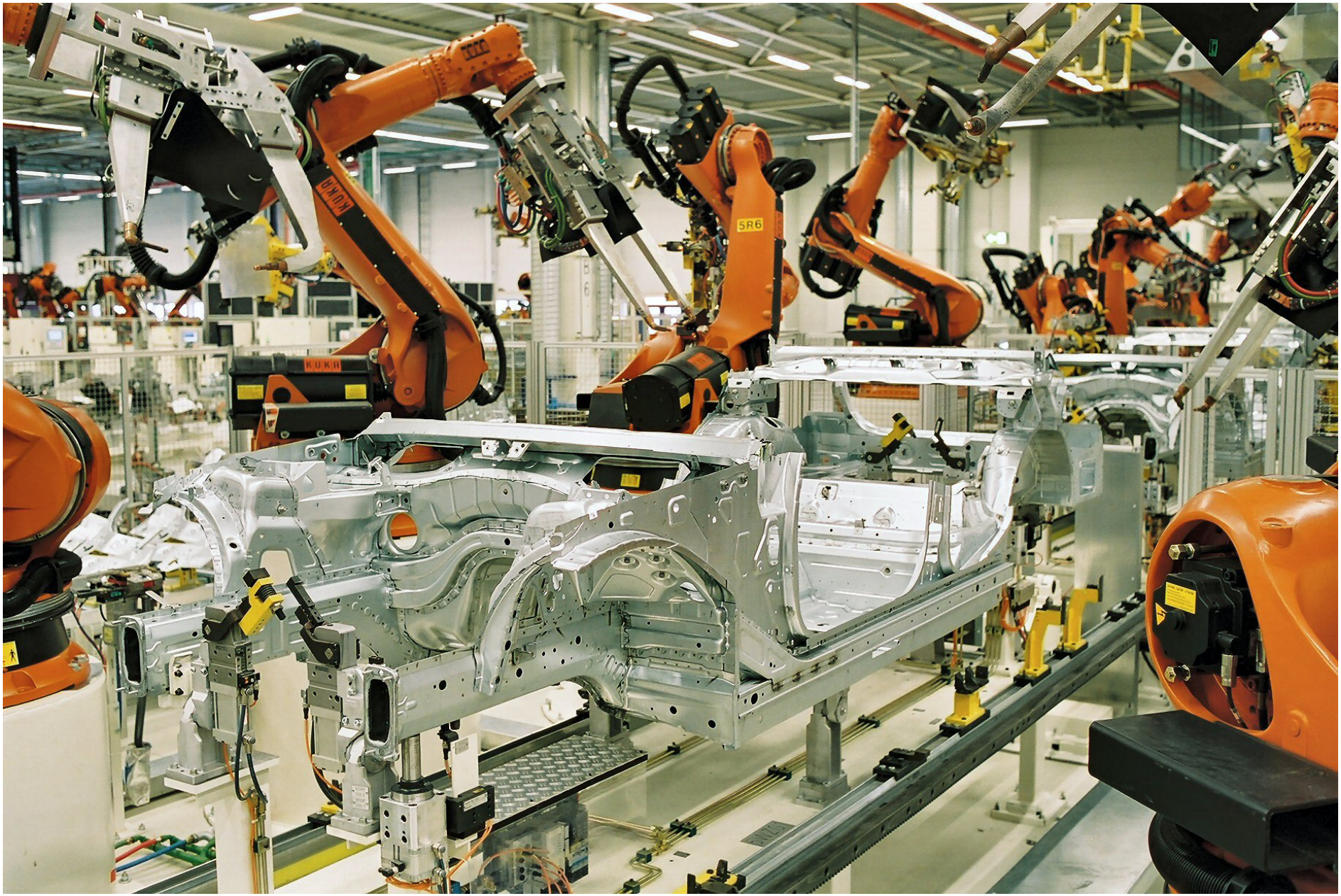
- An intuitive high-level robot behavior specification language
- Interoperability among multiple robot types and manufacturers

Elements enabling existing approaches:

- Canonical Robot Command Language (CRCL)
- Core Ontology for Robotics and Automation (CORA)
- Robot Operating System (ROS)

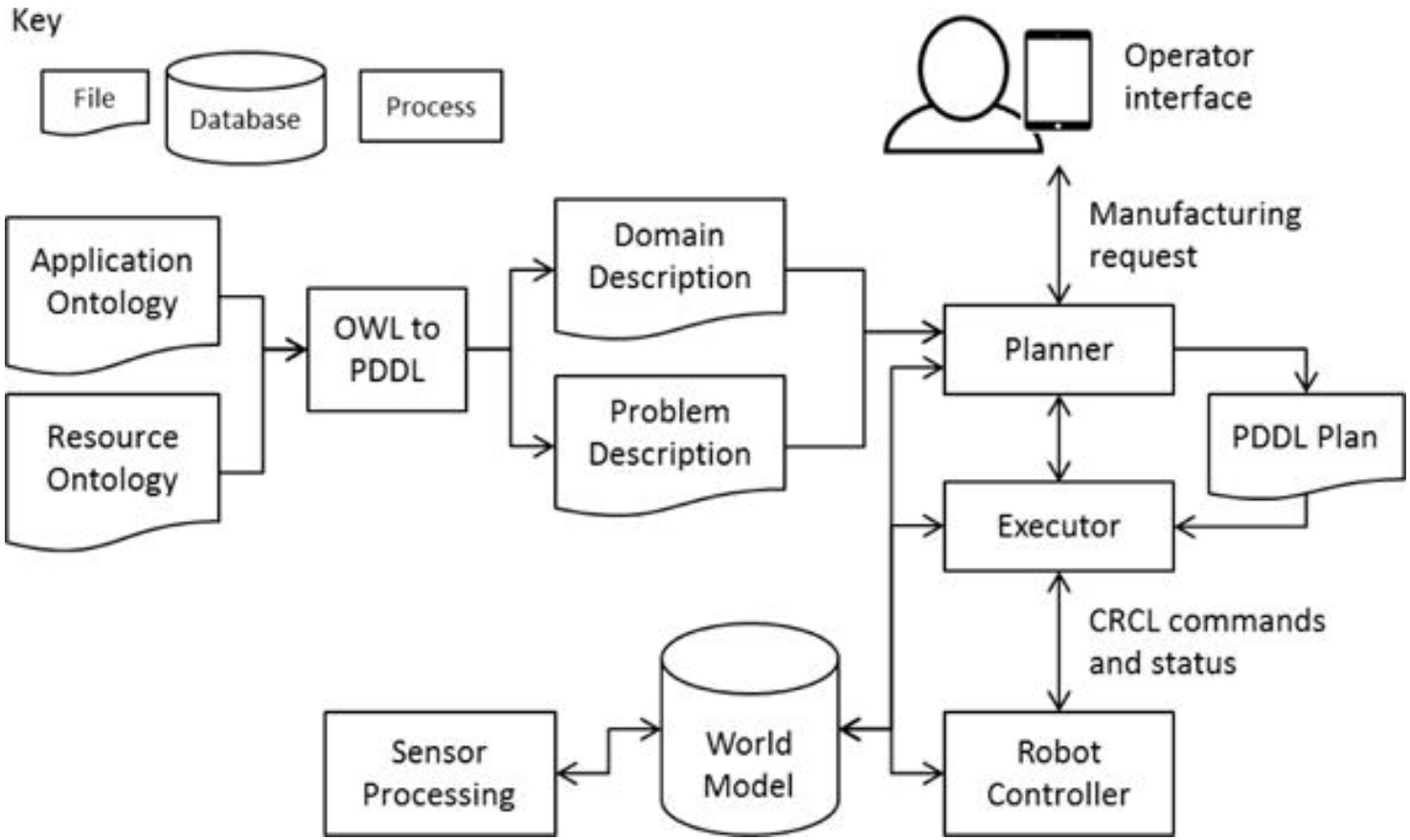
The role of category theory?

- String diagrams for specifying robot behaviors
- Compilers to different low-level languages
- Connecting components of an architecture



Wikipedia

One architecture



Proctor, F. M., Balakirsky, S. B., Kootbally, Z., Kramer, T. R., Schlenoff, C. I., & Shackleford, W. P. (2016). The Canonical Robot Command Language (CRCL). The Industrial robot, 43(5), 495502. doi:10.1108/IR-01-2016-0037

Planning Domain Description Language (PDDL)

domain
objects: things in the world
predicates: boolean-valued statements specifying state of the world
actions: state updates

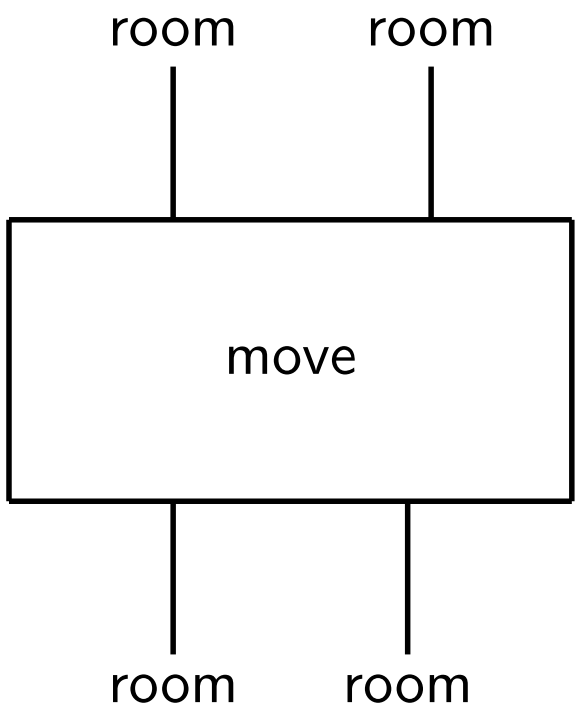
problem
initial state
goal state

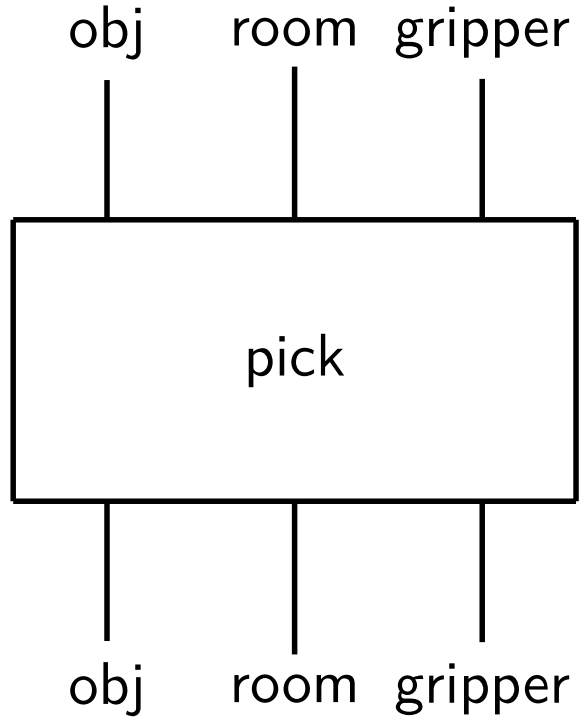
solution
chain of actions from initial state to goal state

Planning Domain Description Language (PDDL)

```
(define (domain gripper-strips)
  (:predicates (room ?r) (ball ?b) (gripper ?g) (at-robby ?r)
               (at ?b ?r) (free ?g) (carry ?o ?g))
  (:action move
   :parameters (?from ?to)
   :precondition (and (room ?from)
                      (room ?to)
                      (at-robby ?from))
   :effect (and (at-robby ?to)
                (not (at-robby ?from))))
  (:action pick
   :parameters (?obj ?room ?gripper)
   :precondition (and (ball ?obj)
                      (room ?room)
                      (gripper ?gripper)
                      (at ?obj ?room)
                      (at-robby ?room)
                      (free ?gripper))
   :effect (and (carry ?obj ?gripper)
                (not (at ?obj ?room))
                (not (free ?gripper))))
```

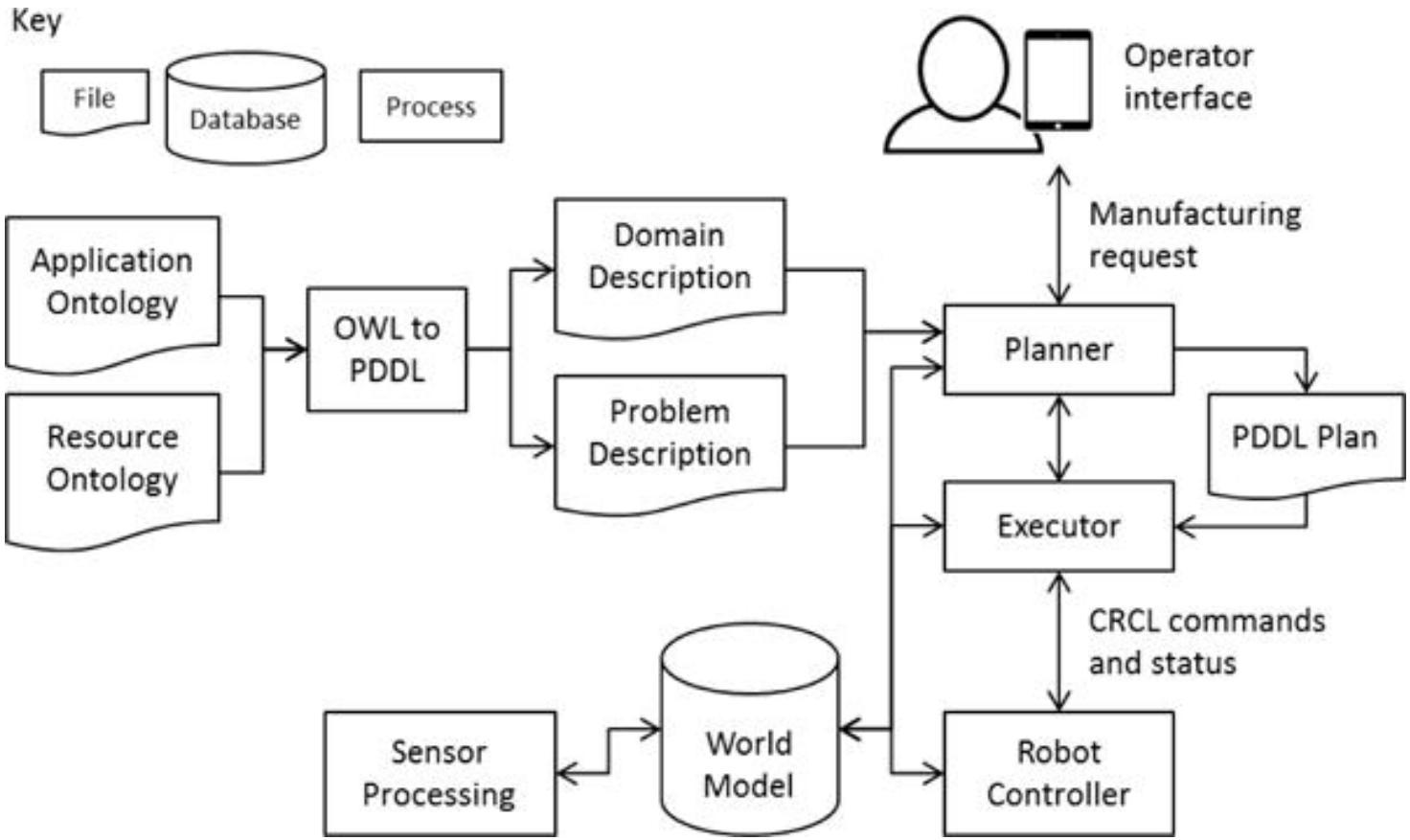
Wikipedia





... and a similar ‘drop’ action.

One architecture



Proctor, F. M., Balakirsky, S. B., Kootbally, Z., Kramer, T. R., Schlenoff, C. I., & Shackleford, W. P. (2016). The Canonical Robot Command Language (CRCL). The Industrial robot, 43(5), 495502. doi:10.1108/IR-01-2016-0037

Problem:

```
(define (problem strips-gripper2)
    (:domain gripper-strips)
    (:objects rooma roomb ball1 ball2 left right)
    (:init (room rooma)
           (room roomb)
           (ball ball1)
           (ball ball2)
           (gripper left)
           (gripper right)
           (at-robby rooma)
           (free left)
           (free right)
           (at ball1 rooma)
           (at ball2 rooma))
    (:goal (at ball1 roomb)))
```

Initial state:

$$at\_robby(rooma) = T$$

$$at(ball1, rooma) = T$$

$$at(ball2, rooma) = T$$

Goal state:

$$at(ball1, roomb) = T$$

Enabled actions:

$$pick(ball1/2, rooma, left/right)$$

$$pick(ball1, rooma, left)$$

$$move(rooma, roomb)$$

Initial state:

$$at\_robby(rooma) = T$$

$$at(ball1, rooma) = T$$

$$at(ball2, rooma) = T$$

Goal state:

$$at(ball1, roomb) = T$$

Enabled actions:

$$pick(ball1/2, rooma, left/right)$$

$$pick(ball1, rooma, left)$$

$$move(rooma, roomb)$$

But not, e.g.:

$$pick(ball1, roomb, left)$$

Problem:

```
(define (problem strips-gripper2)
    (:domain gripper-strips)
    (:objects rooma roomb ball1 ball2 left right)
    (:init (room rooma)
           (room roomb)
           (ball ball1)
           (ball ball2)
           (gripper left)
           (gripper right)
           (at-robby rooma)
           (free left)
           (free right)
           (at ball1 rooma)
           (at ball2 rooma))
    (:goal (at ball1 roomb)))
```

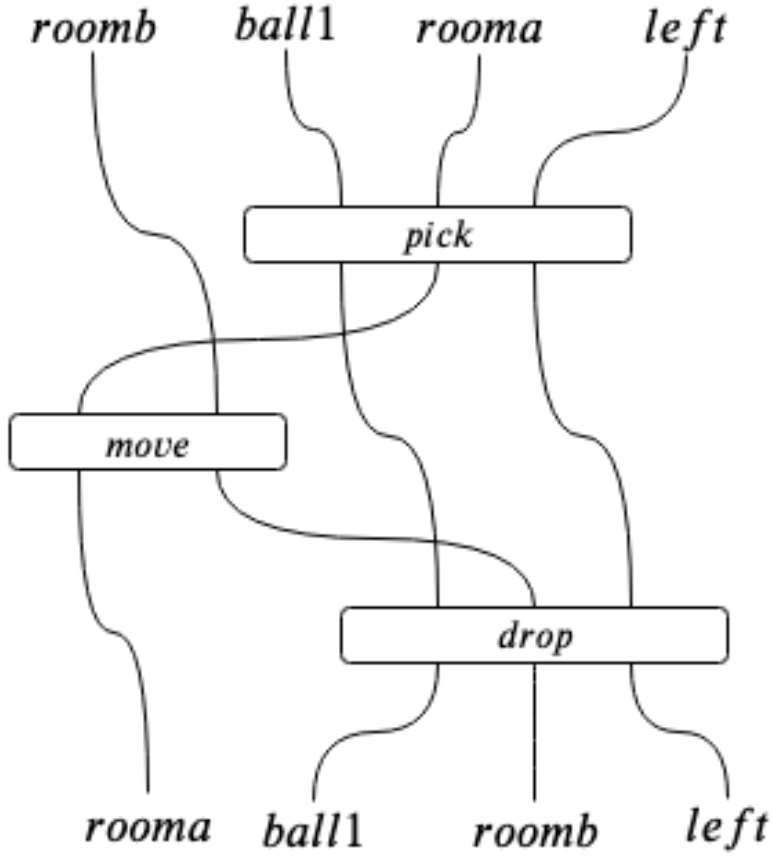
Problem:

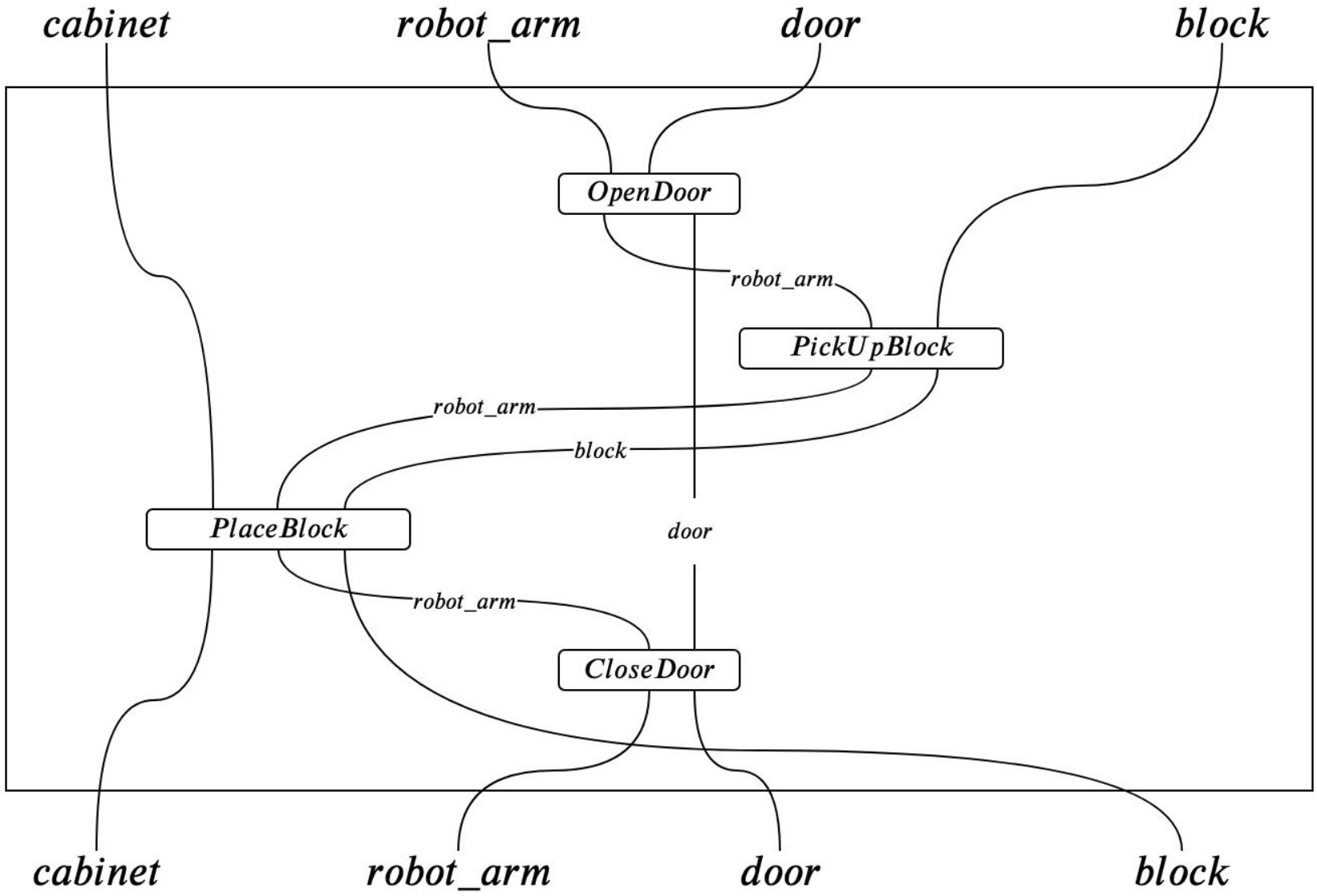
```
(define (problem strips-gripper2)
    (:domain gripper-strips)
    (:objects rooma roomb ball1 ball2 left right)
    (:init (room rooma)
           (room roomb)
           (ball ball1)
           (ball ball2)
           (gripper left)
           (gripper right)
           (at-robby rooma)
           (free left)
           (free right)
           (at ball1 rooma)
           (at ball2 rooma))
    (:goal (at ball1 roomb)))
```

Solution:

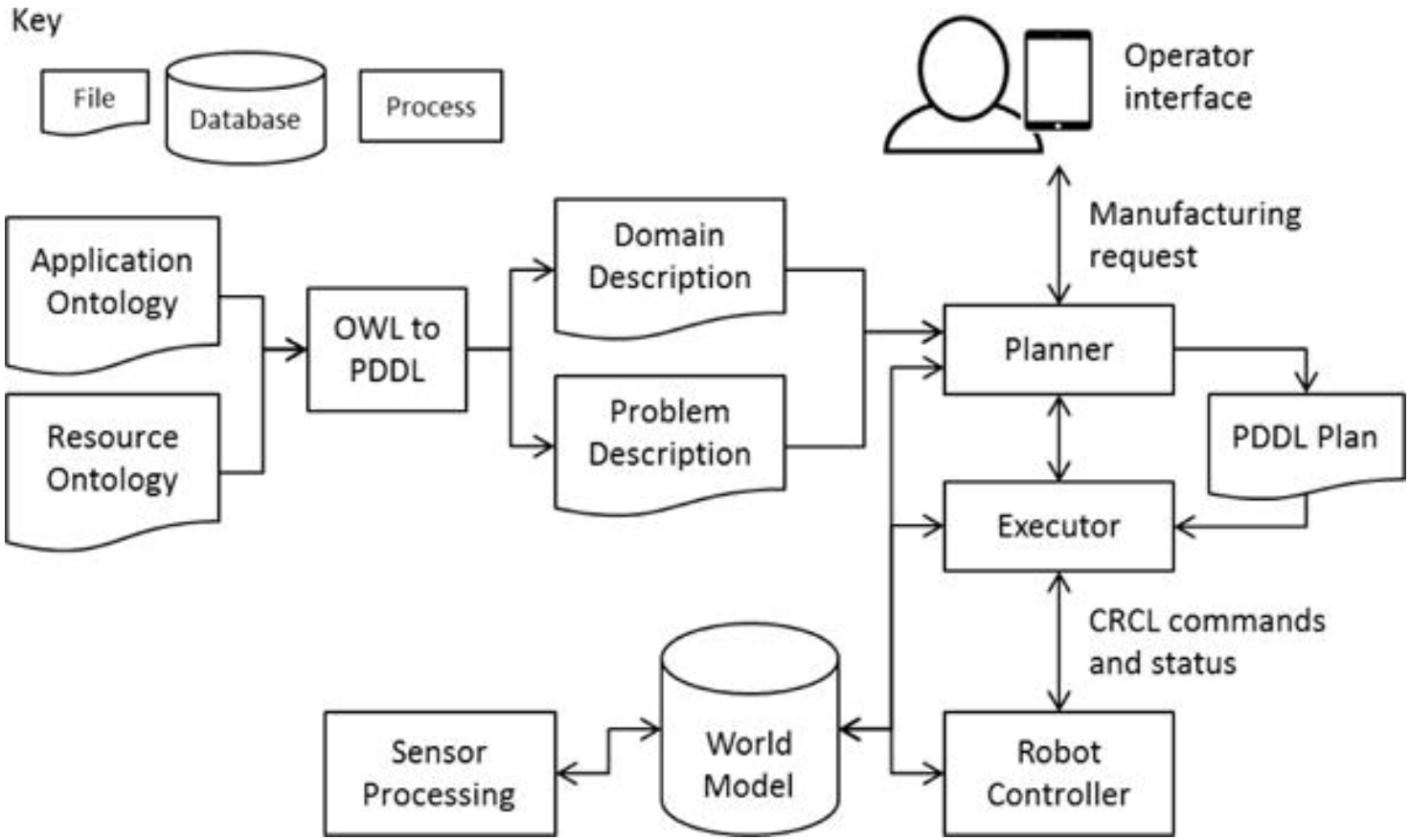
(pick ball1 rooma left)
(move rooma roomb)
(drop ball1 roomb left)

Given a PDDL domain and problem file, there is a symmetric monoidal category `PDDL` where objects are PDDL objects and morphisms are PDDL actions.




cabinet
robot_arm
door
block
OpenDoor
robot_arm
PickUpBlock
robot_arm
block
PlaceBlock
door
robot_arm
CloseDoor
cabinet
robot_arm
door
block

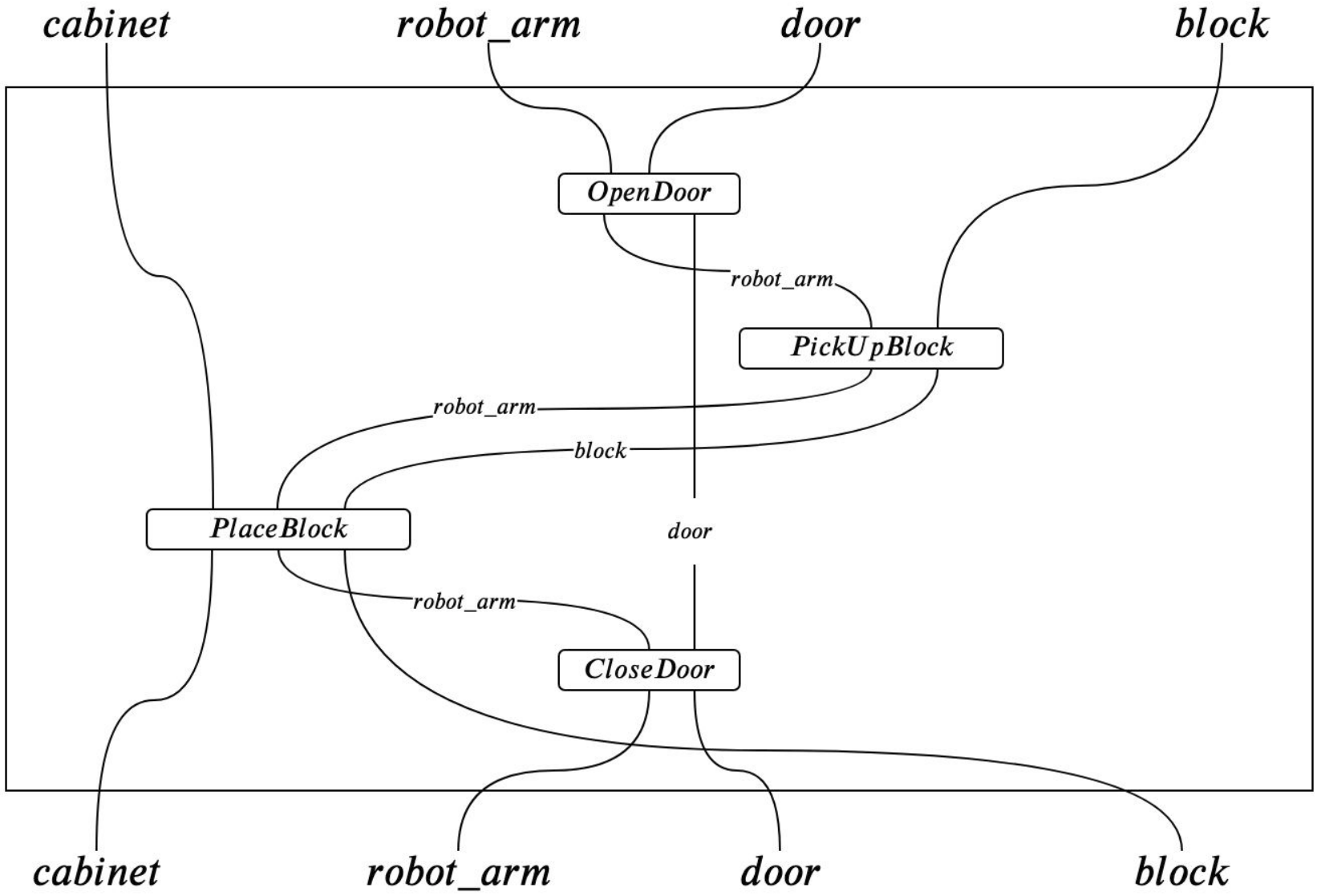
One architecture



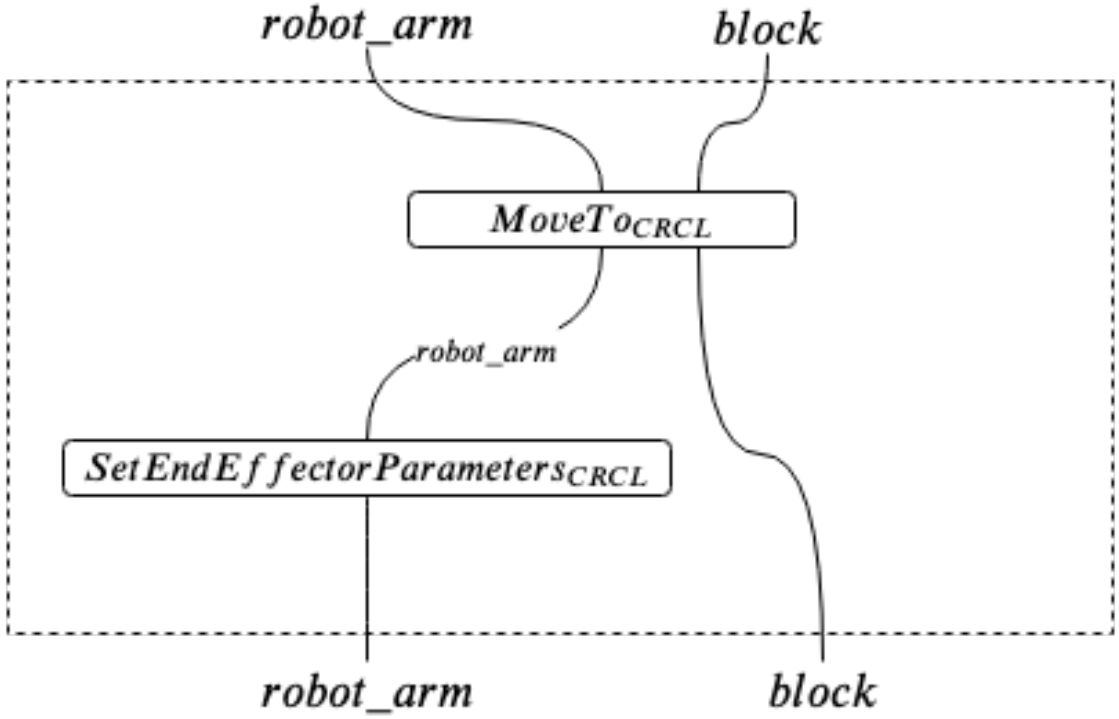
Proctor, F. M., Balakirsky, S. B., Kootbally, Z., Kramer, T. R., Schlenoff, C. I., & Shackleford, W. P. (2016). The Canonical Robot Command Language (CRCL). The Industrial robot, 43(5), 495502. doi:10.1108/IR-01-2016-0037

Canonical Robot Control Language (CRCL)

```
<?xml version="1.0" encoding="UTF-8"?>
<CRCLCommandInstance
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="../xmlSchemas/CRCLCommandInstance.xsd">
  <CRCLCommand xsi:type="MoveToType">
    <CommandID>2</CommandID>
    <MoveStraight>false</MoveStraight>
    <EndPosition>
      <Point>
        <X>2.5</X> <Y>1</Y> <Z>1</Z>
      </Point>
      <XAxis>
        <I>1</I> <J>0</J> <K>0</K>
      </XAxis>
      <ZAxis>
        <I>0</I> <J>0</J> <K>-1</K>
      </ZAxis>
    </EndPosition>
  </CRCLCommand>
</CRCLCommandInstance>
```


cabinet
robot_arm
door
block
OpenDoor
robot_arm
PickUpBlock
robot_arm
block
PlaceBlock
door
robot_arm
CloseDoor
cabinet
robot_arm
door
block

PickUpBlock



Hierarchical Tasking

Binding combinations of CRCL commands to PDDL actions by hand induces a functor on the syntax

$$I : \texttt{PDDL} \rightarrow \texttt{CRCL}$$

Hierarchical Tasking

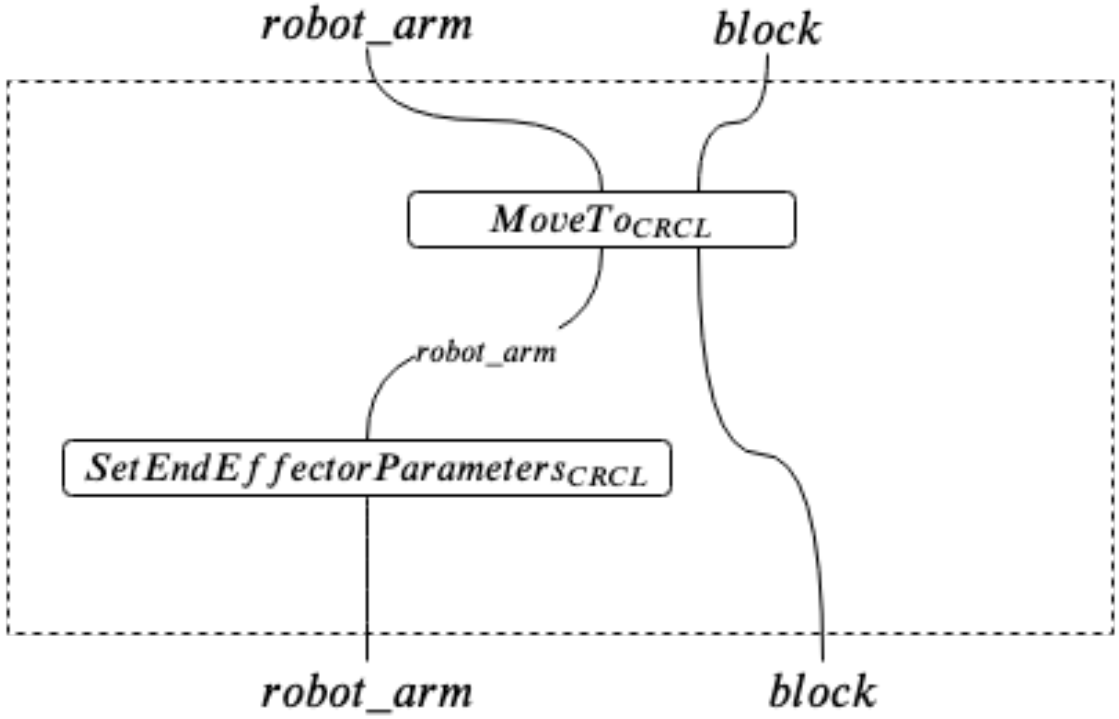
Binding combinations of CRCL commands to PDDL actions by hand induces a functor on the syntax

$$I : \texttt{PDDL} \to \texttt{CRCL}$$

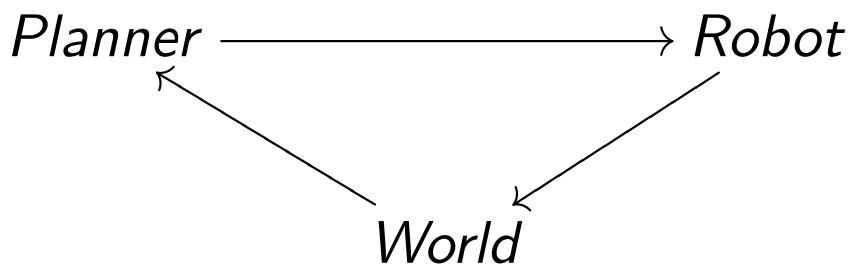
Assigning semantics leads to the following diagram:

$$\begin{array}{ccc} \texttt{PDDL} & \longrightarrow & \texttt{Bool} \\ \downarrow & & \\ \texttt{CRCL} & \xrightarrow{?} & \texttt{Geom} \end{array}$$

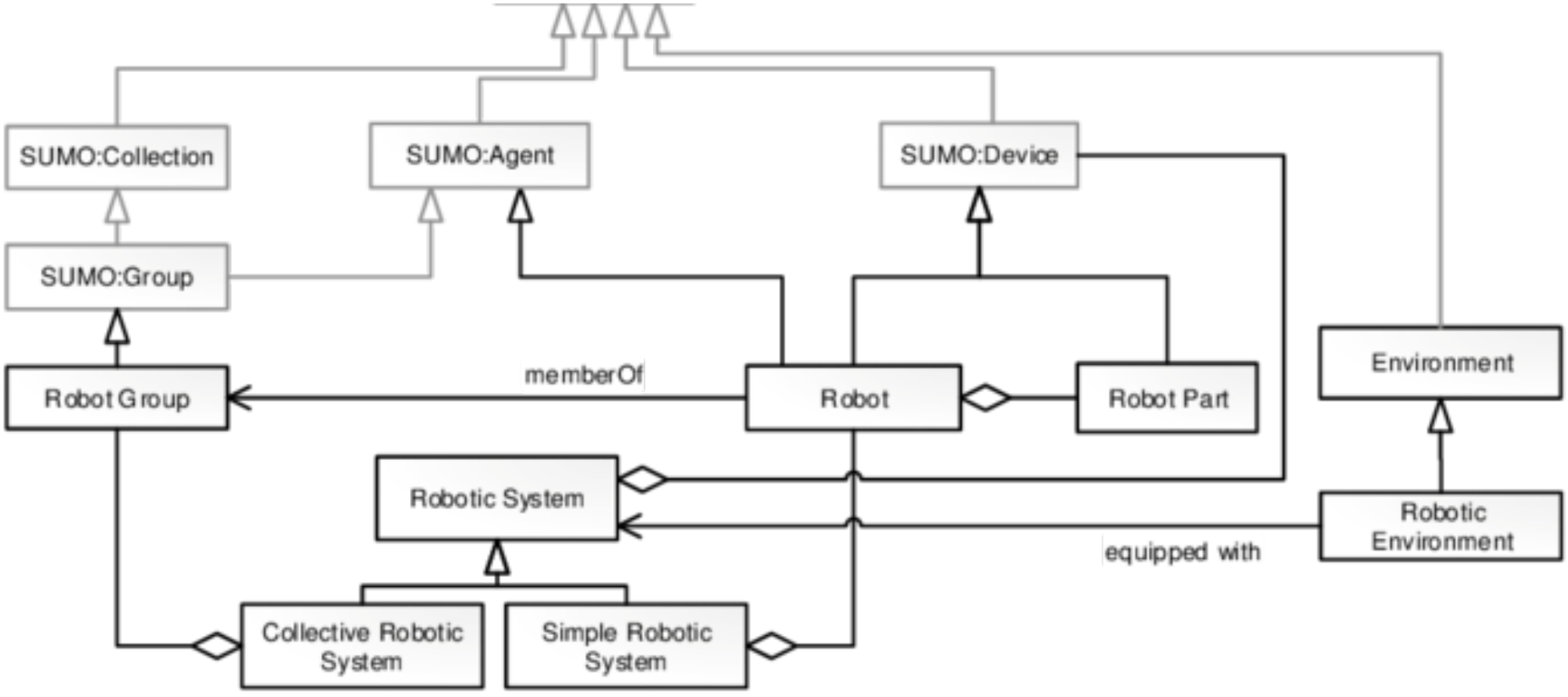
PickUpBlock



Closing the loop



Core Ontology for Robotics and Automation (CORA)



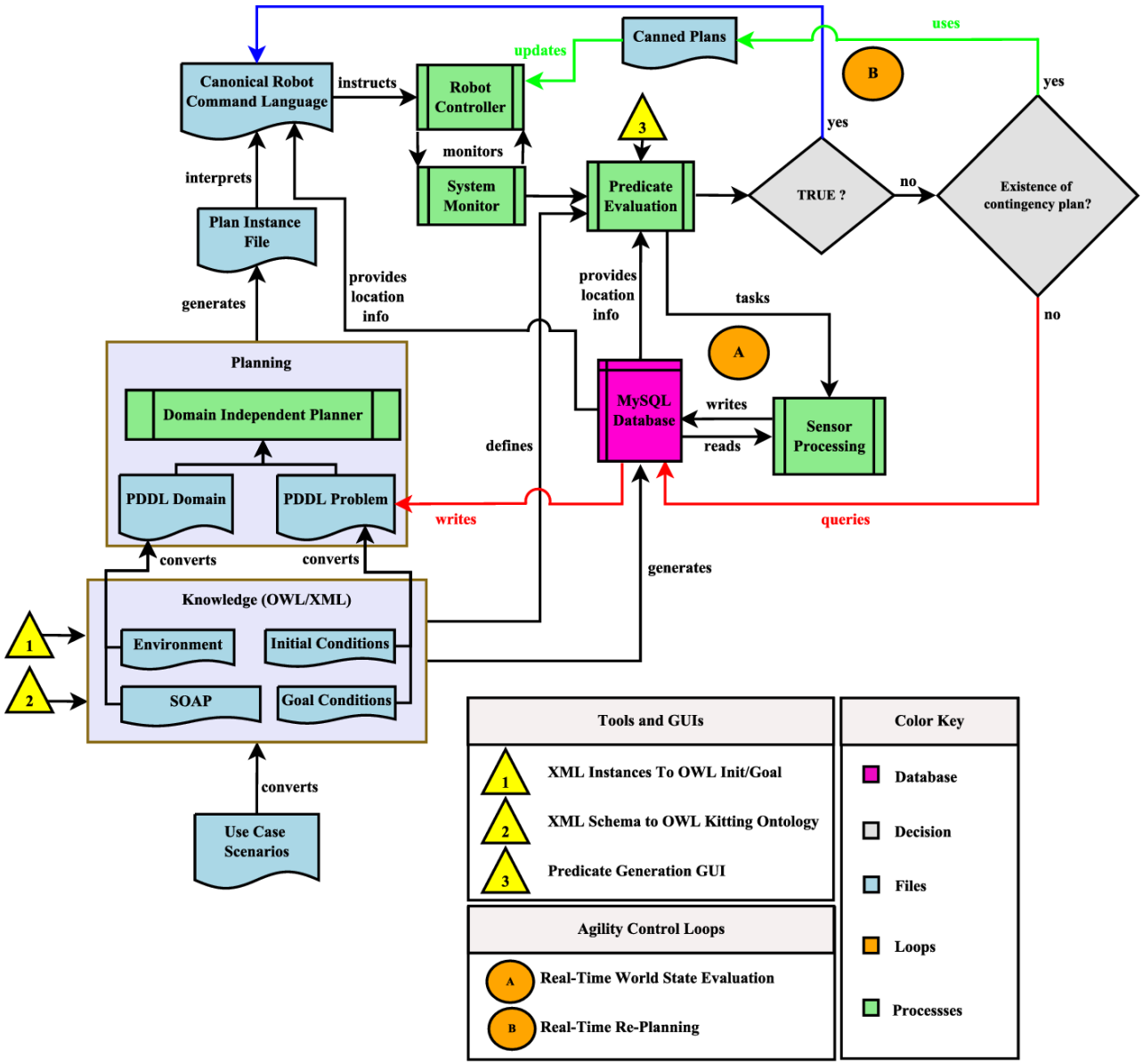


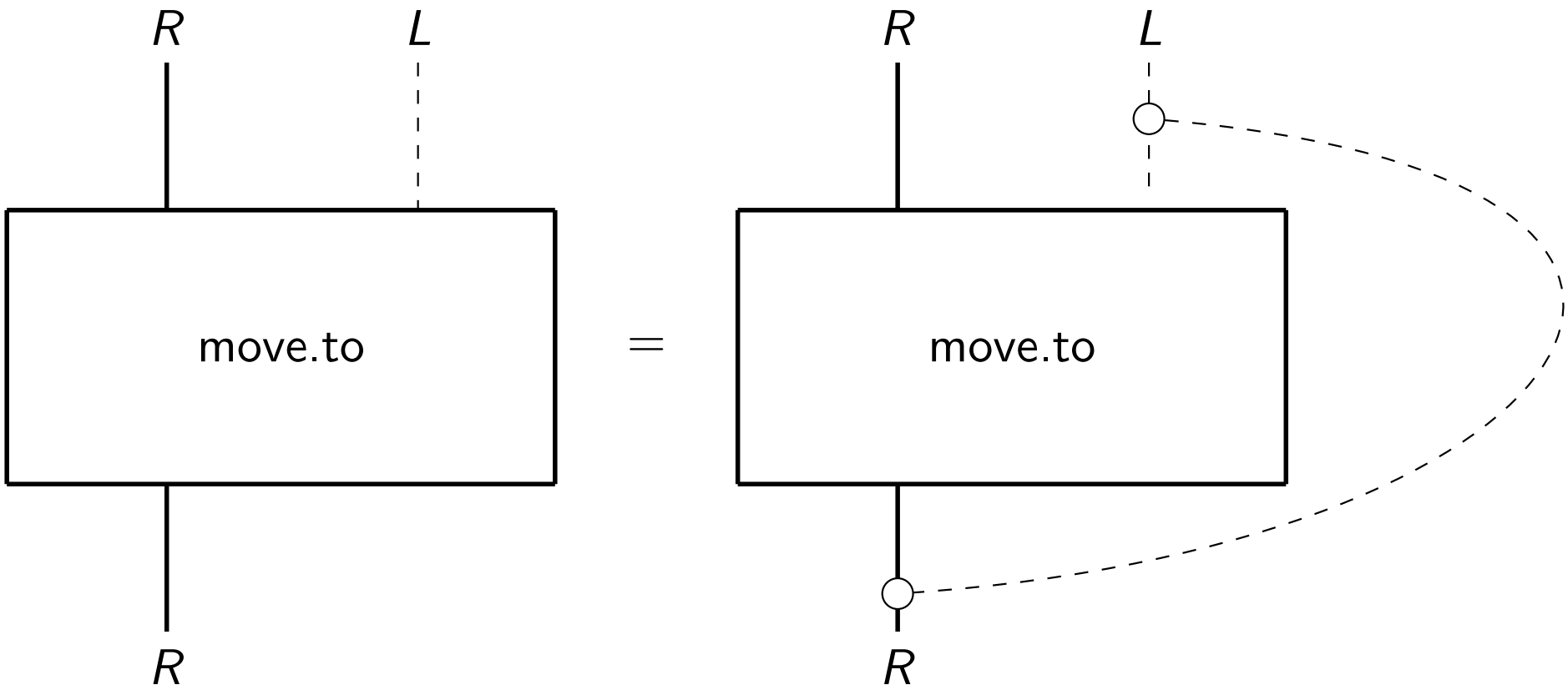
Fig. 1. Knowledge driven design extensions.

Z. Kootbally, C. Schlenoff, C. Lawler, T. Kramer, S.K. Gupta, Towards robust assembly with knowledge representation for the planning domain definition language (PDDL), Robotics and Computer-Integrated Manufacturing, (33),2015.

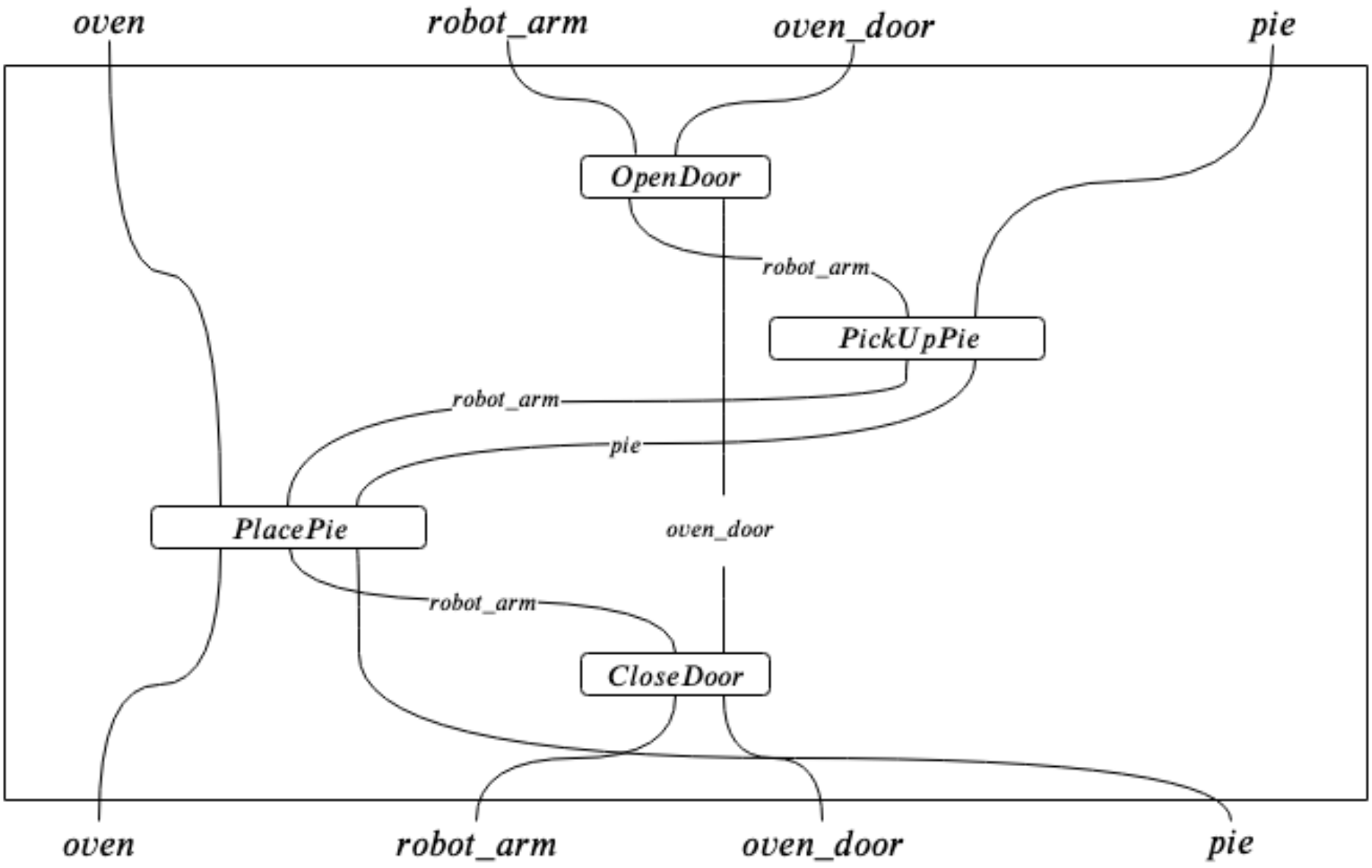
Where are we going?

- PDDL to Catlab (and vice-versa) parser
- XSDL/OWL to CQL
- Parallelization
- Collaboration
- Gazebo testing
- Souped up string diagrams

move



Lemon Meringue Pie



Thanks and Disclaimer

Special thanks to Fred Proctor and the rest of the Intelligent Systems Group at NIST.

Funding for this project provided by the U.S. Department of Defense through Advanced Robotics for Manufacturing (ARM), a Manufacturing USA Institute. ARM is the nations leading collaborative in robotics and workforce innovation. Structured as a public-private partnership, ARM accelerates the advancement of transformative robotic technologies and education to increase U.S. global manufacturing competitiveness.