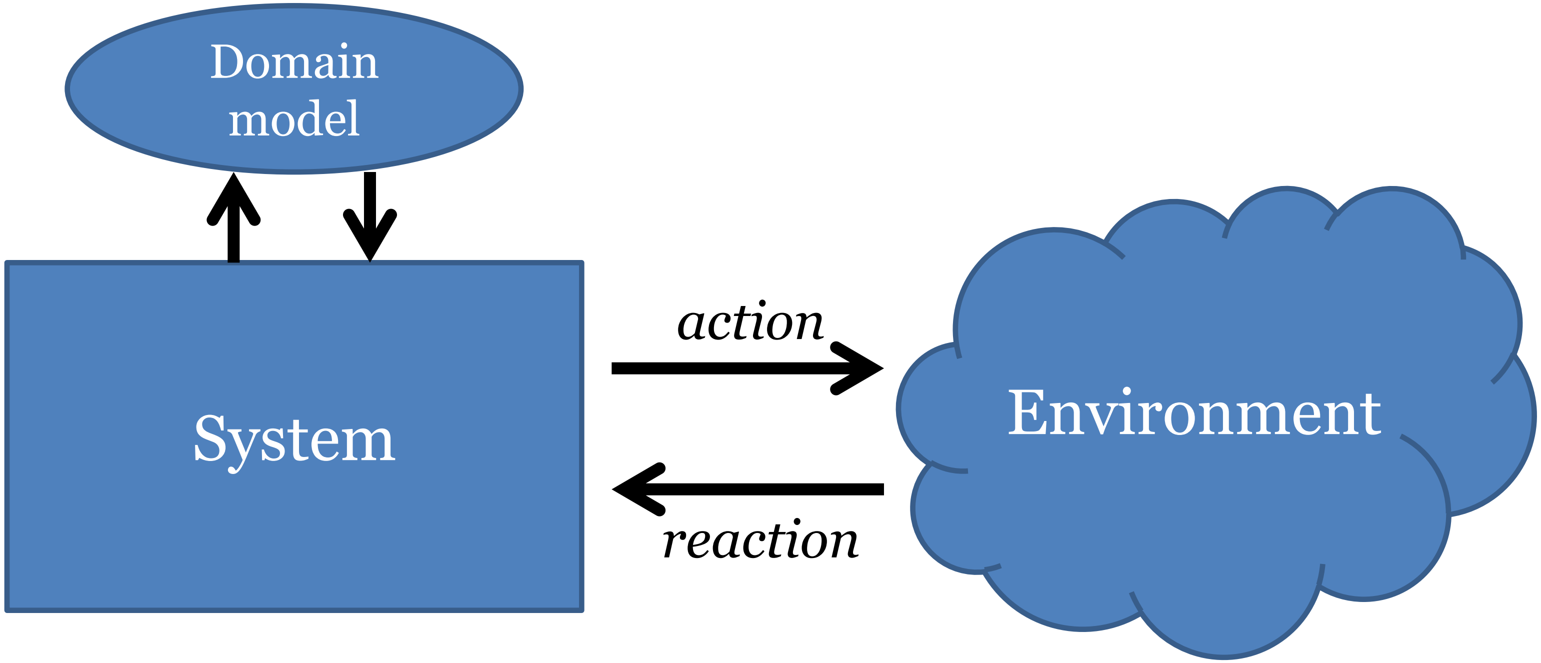
# *Learning Revised Models For Planning In Adaptive Systems*

**Daniel Sykes**, Domenico Corapi, Jeff Magee,
Jeff Kramer, Alessandra Russo, and Katsumi Inoue

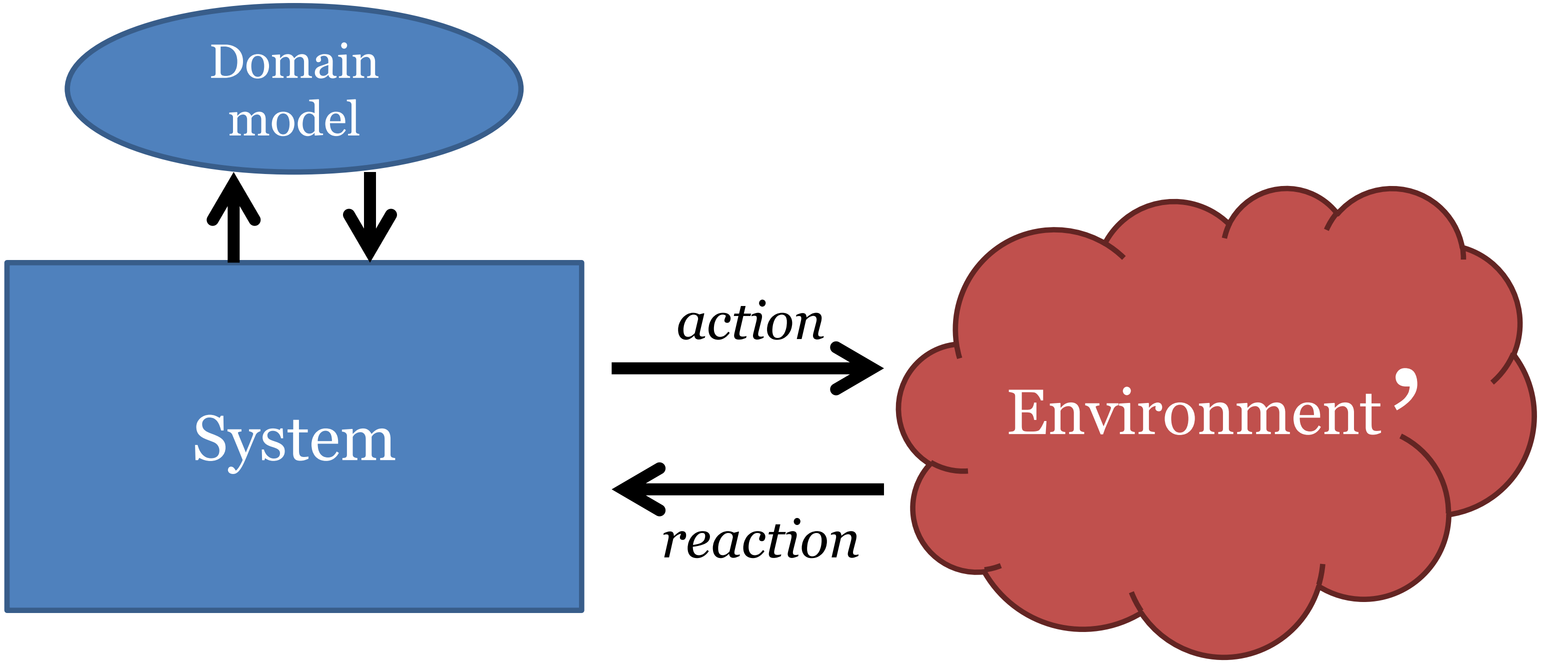
Imperial College, London

ICSE, 22nd May 2013

# Adaptive systems



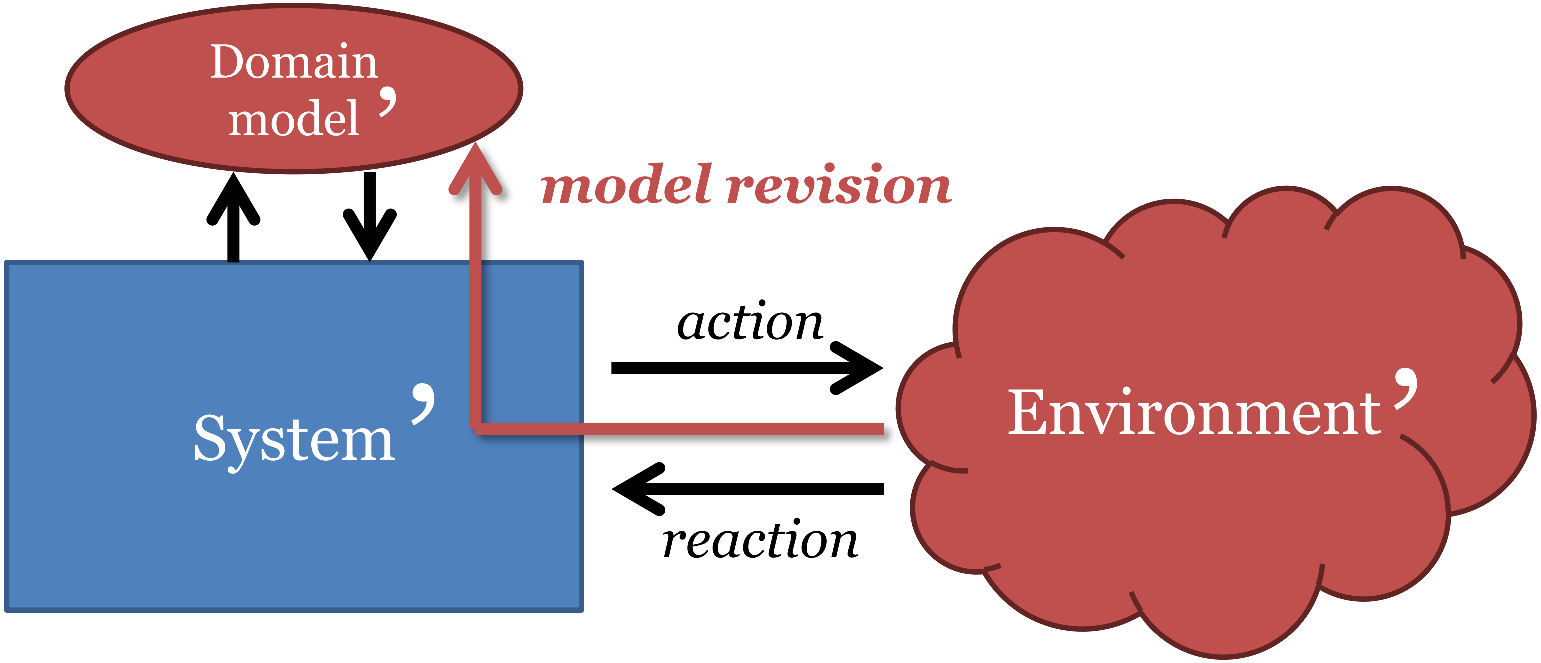
# Adaptive systems



**System** should adapt to **System**’, but domain model out of date

daniel.sykes@imperial.ac.uk

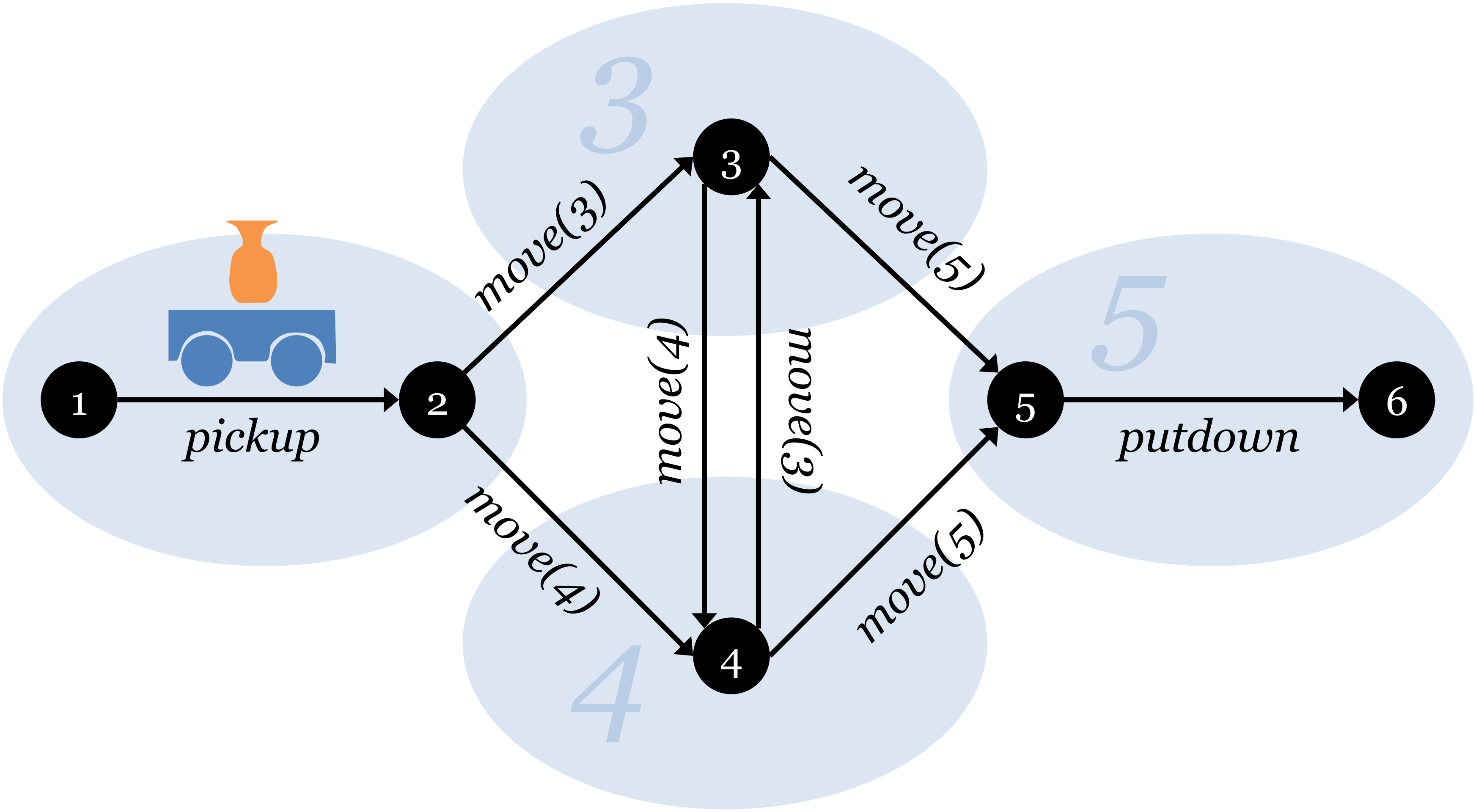
# Adaptive systems



Behavioural model revision through **probabilistic rule learning**

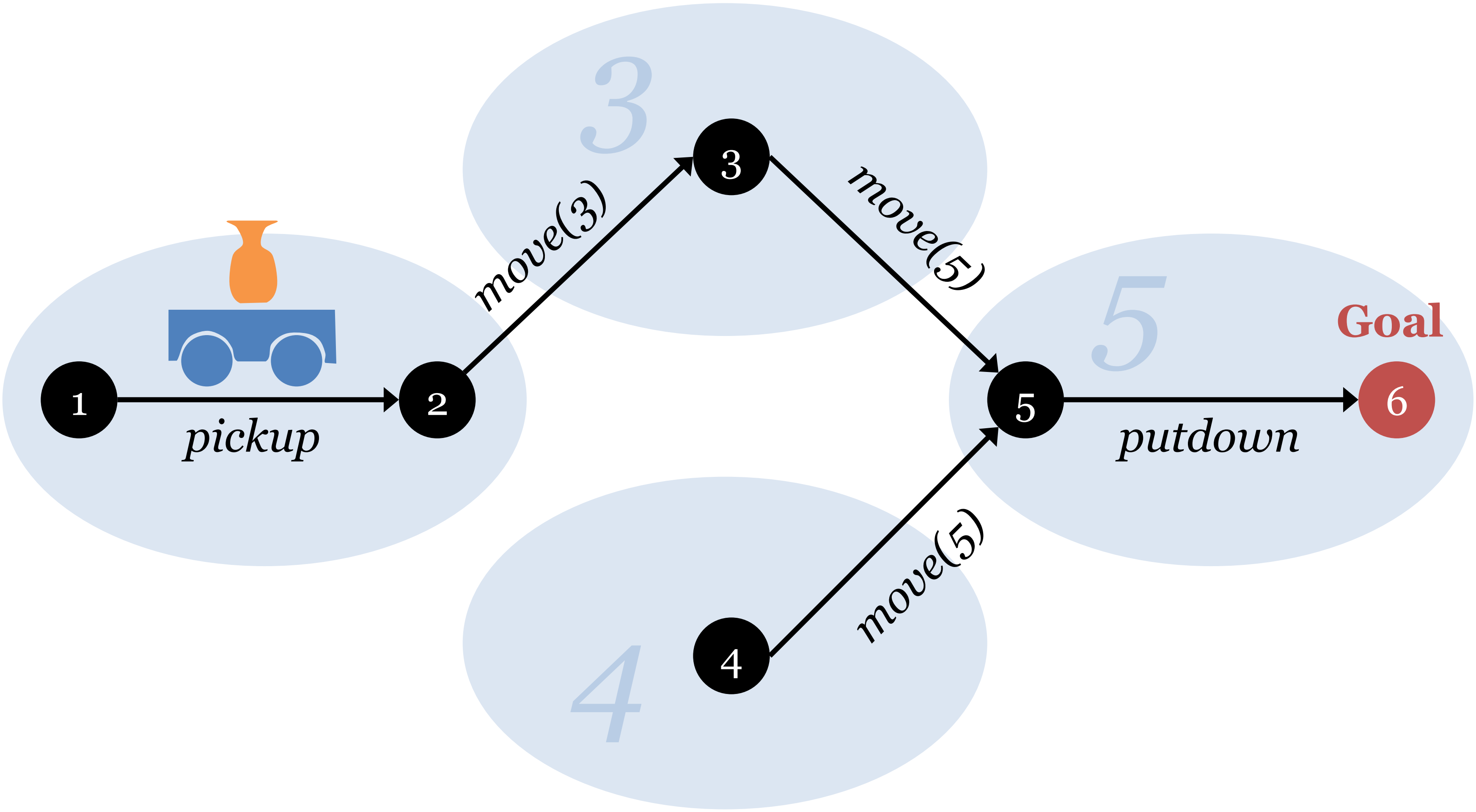
daniel.sykes@imperial.ac.uk

# Factory floor model



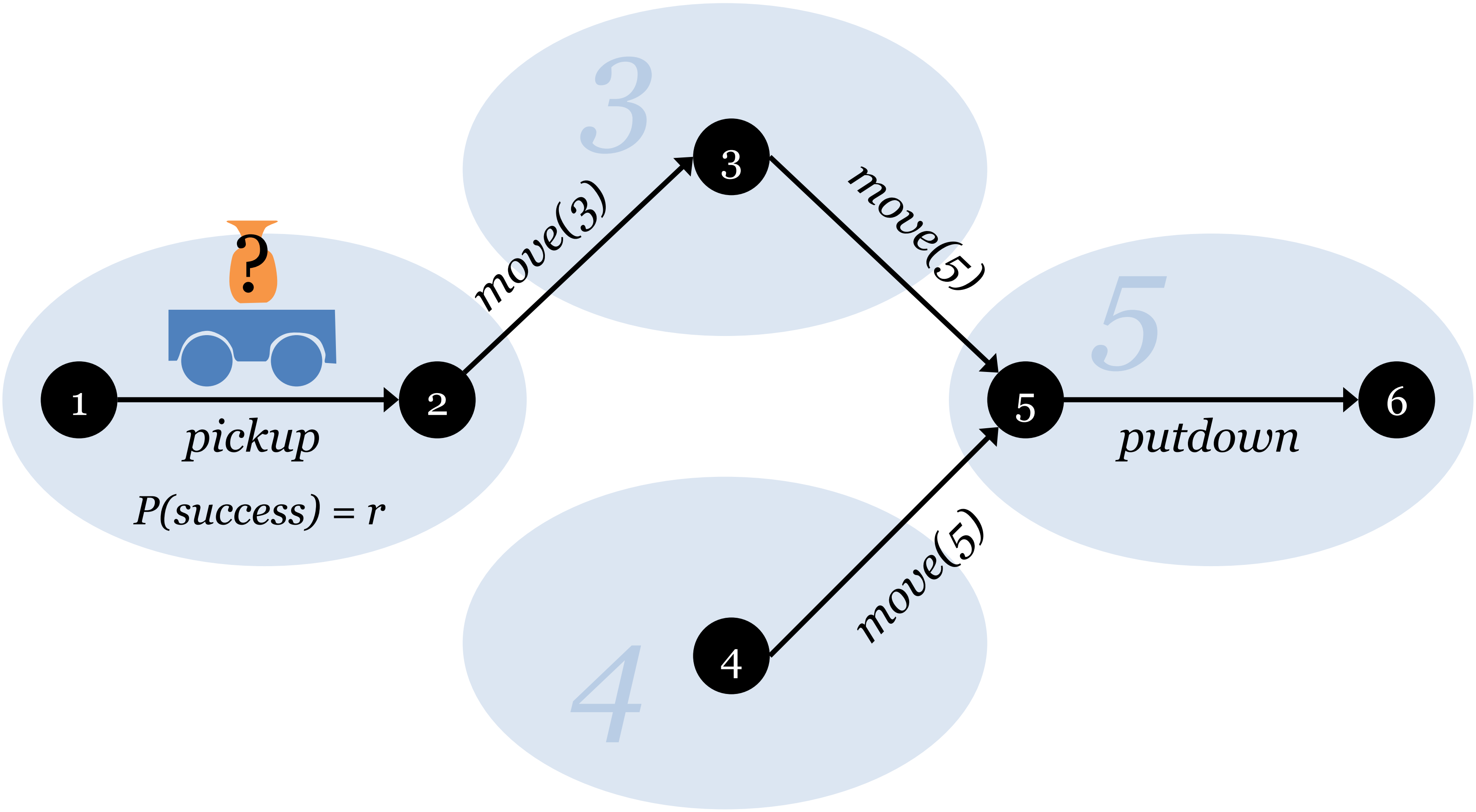
daniel.sykes@imperial.ac.uk

# Factory floor reactive plan



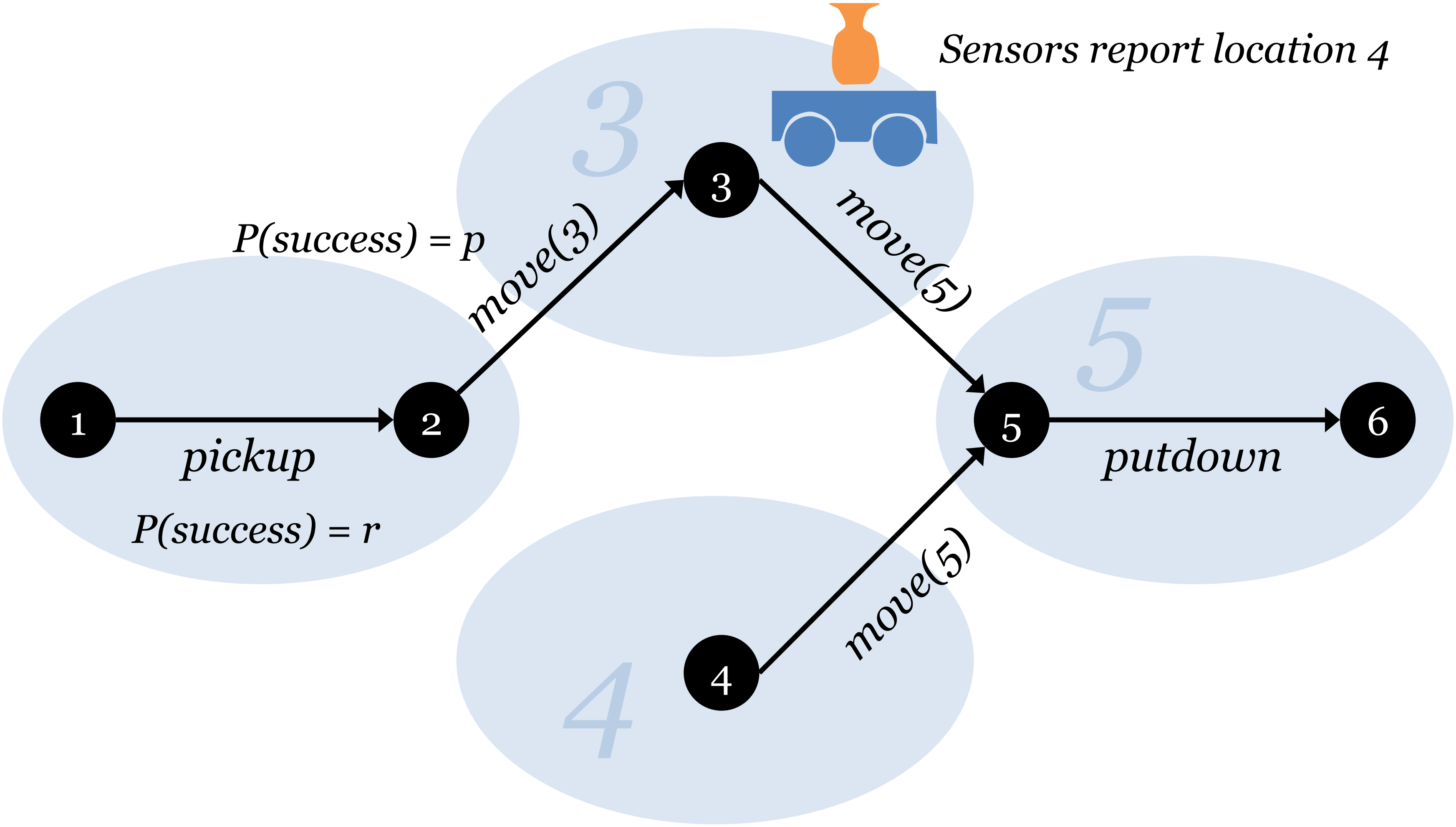
Sykes et al., SEAMS 2008

# Factory floor at runtime



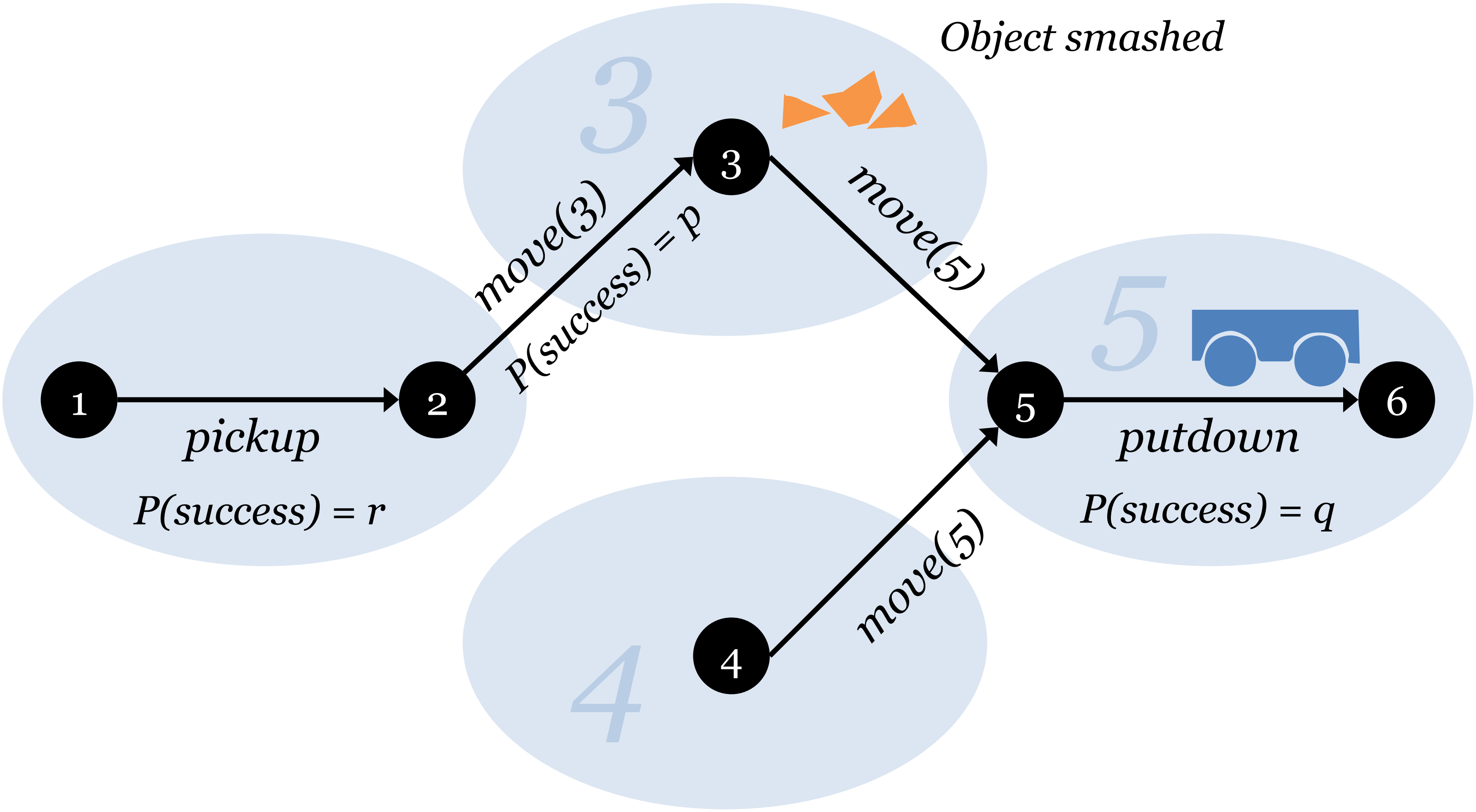
daniel.sykes@imperial.ac.uk

# Factory floor at runtime



daniel.sykes@imperial.ac.uk

# Factory floor at runtime



*Probability of overall path success = p.q.r*

daniel.sykes@imperial.ac.uk

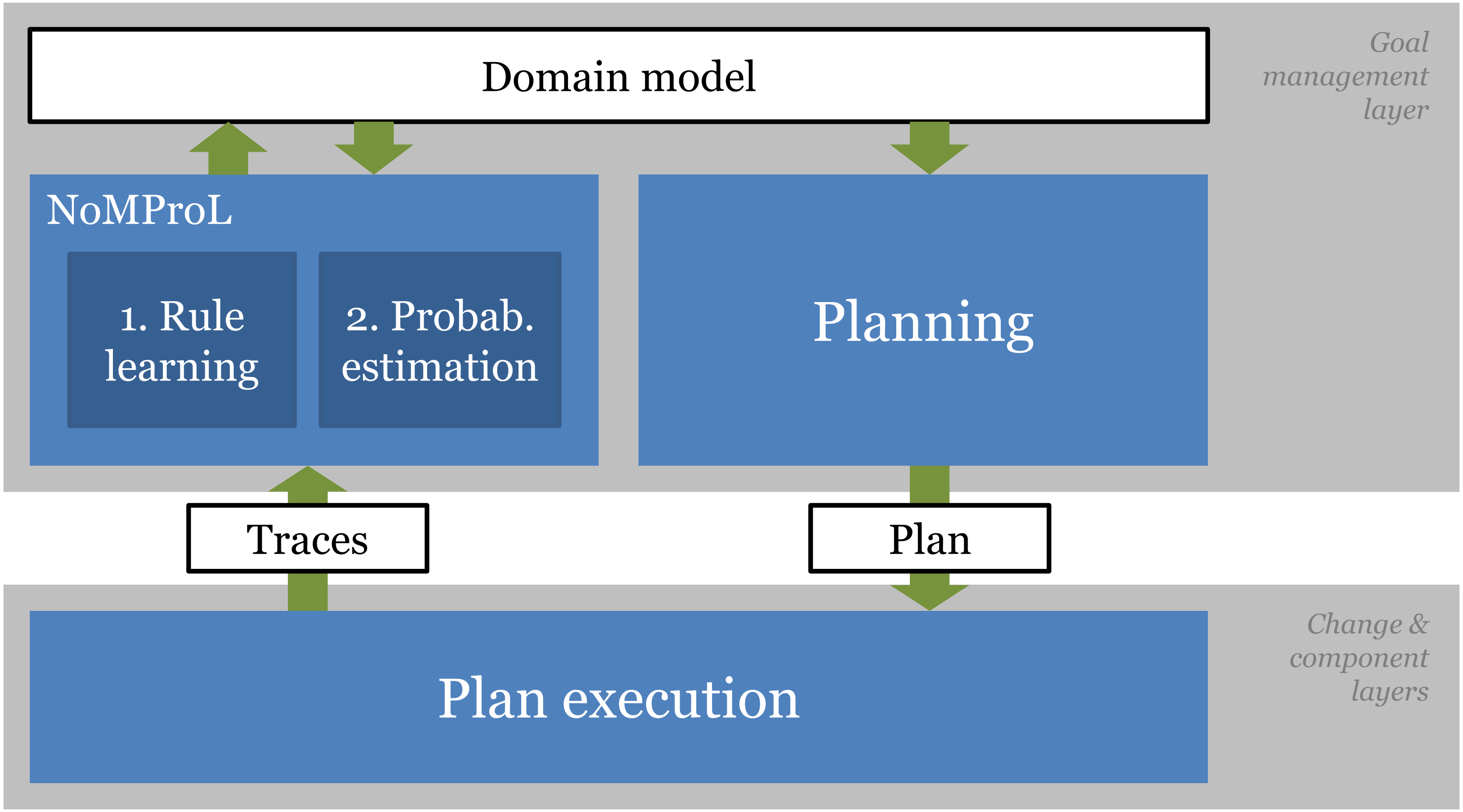
# Model revision

- Model does not reflect real environment
  - Unmodelled states or transitions
  - Original model not probabilistic
  - Difficult to estimate probabilities without testing

**Task** update model according to observed environment

daniel.sykes@imperial.ac.uk

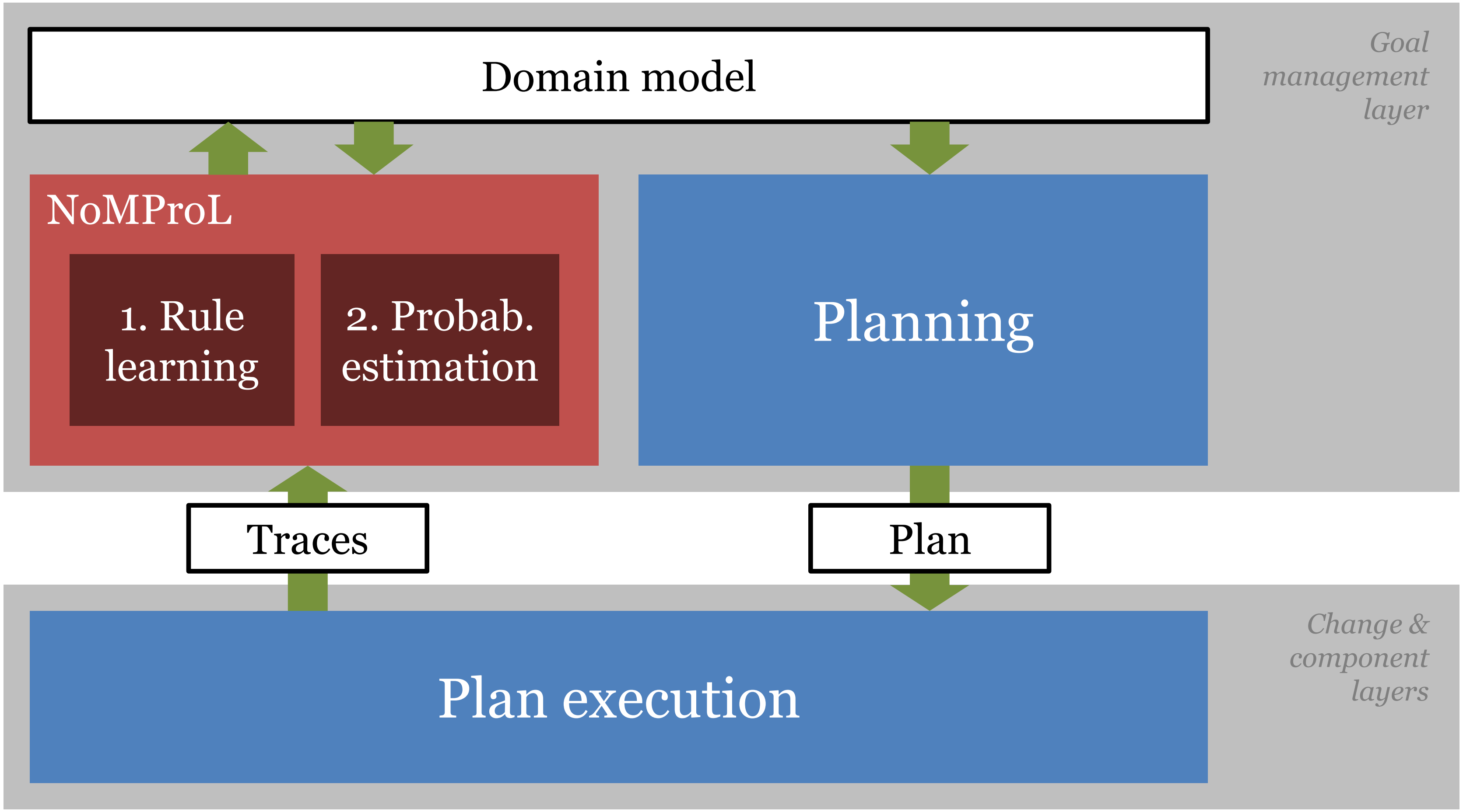
# Probabilistic rule learning



Kramer & Magee, FOSE 2007
Corapi et al., CLIMA 2011

daniel.sykes@imperial.ac.uk

# Probabilistic rule learning



Domain model

NoMProL

1. Rule learning

2. Probab. estimation

Planning

Traces

Plan

Plan execution

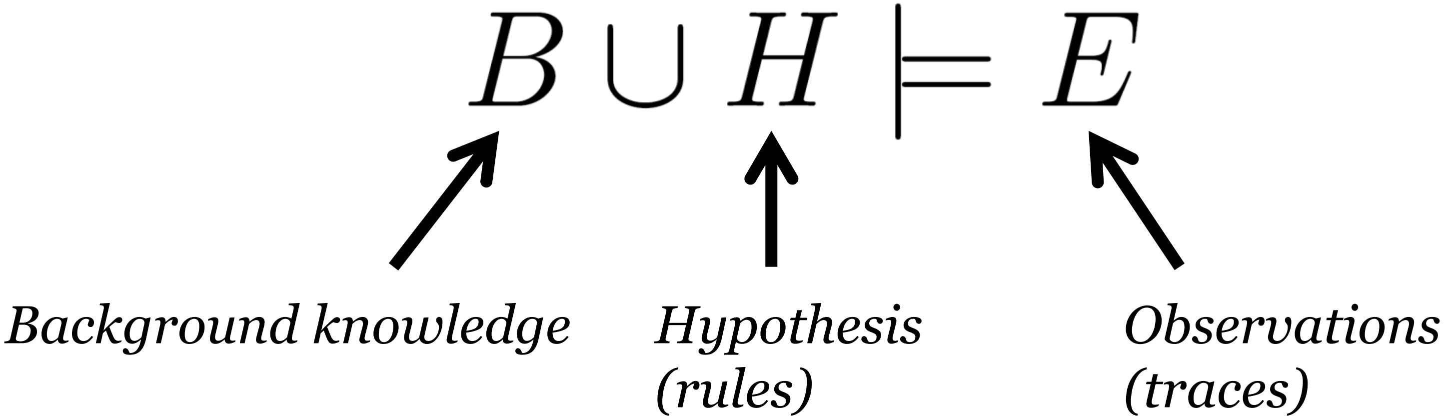
*Goal management layer*

*Change & component layers*

Kramer & Magee, FOSE 2007
Corapi et al., CLIMA 2011

daniel.sykes@imperial.ac.uk

# Inductive logic programming



$$B \cup H \models E$$

*Background knowledge*

*Hypothesis (rules)*

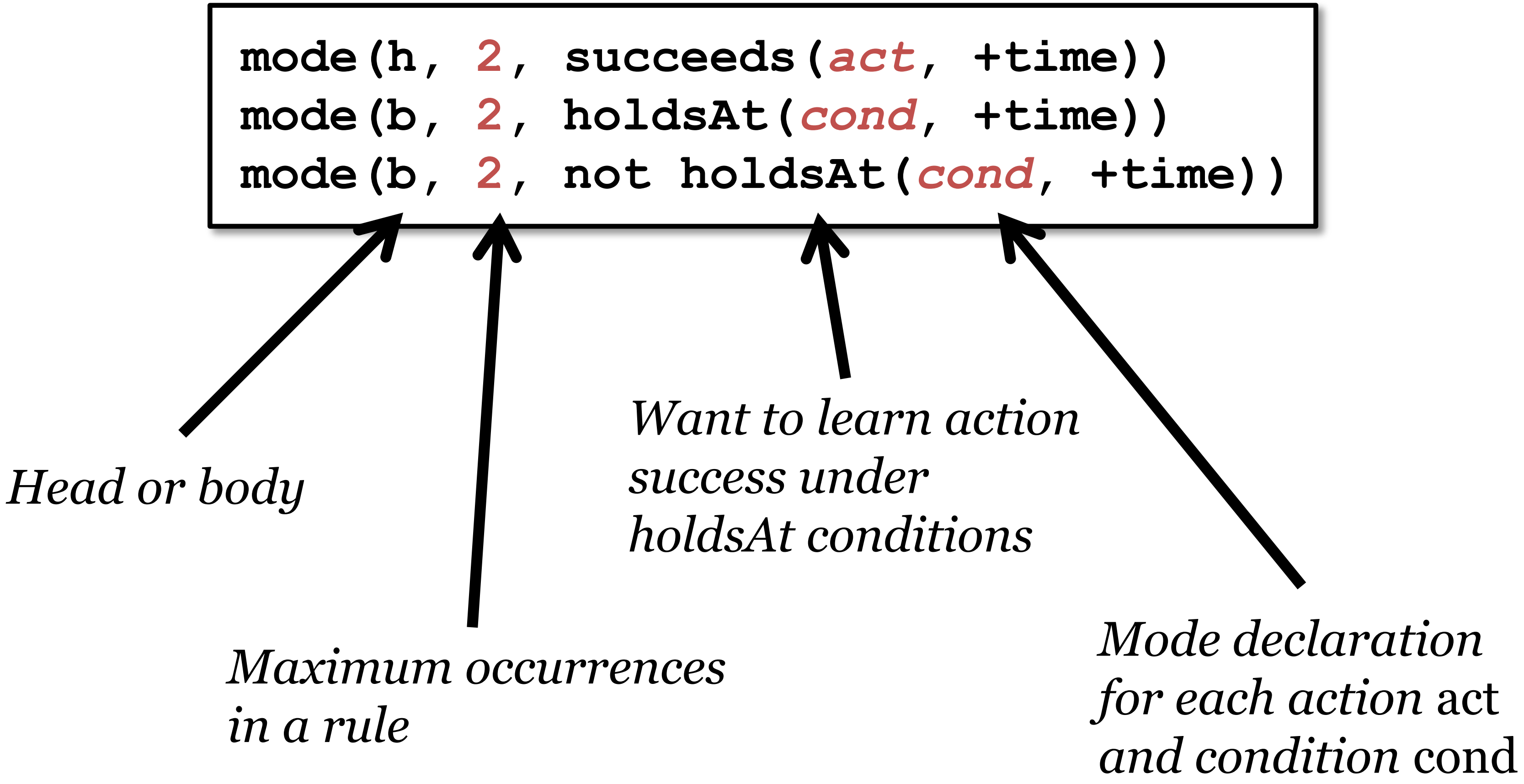
*Observations (traces)*

Many possible hypotheses, some very specific, some more general

Muggleton 1995

daniel.sykes@imperial.ac.uk

# Mode declarations



```
mode(h, 2, succeeds(act, +time))
mode(b, 2, holdsAt(cond, +time))
mode(b, 2, not holdsAt(cond, +time))
```

*Head or body*

*Maximum occurrences in a rule*

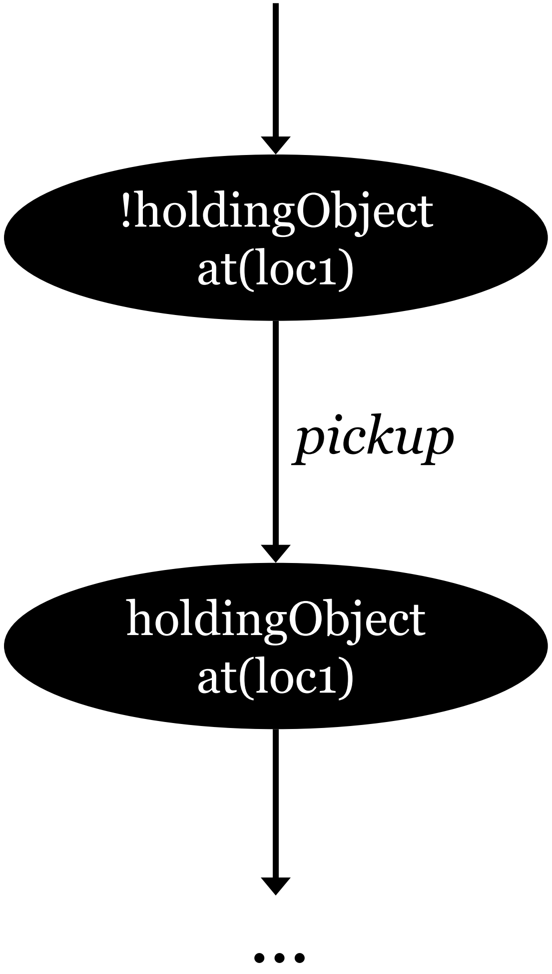
*Want to learn action success under holdsAt conditions*

*Mode declaration for each action* act *and condition* cond

daniel.sykes@imperial.ac.uk

# Domain modelling

```
possible(pickup, T) :-
       not holdsAt(holdingObject, T),
       holdsAt(at(loc1), T).
possible(putdown, T) :-
       holdsAt(holdingObject, T),
       holdsAt(at(loc5), T).
possible(move(L1, L2), T) :-
       holdsAt(at(L1), T),
       connected(L1, L2).
...
initiates(pickup, holdingObject, T).
terminates(putdown, holdingObject, T).
initiates(move(L1, L2), at(L2), T).
terminates(move(L1, L2), at(L1), T).
```



daniel.sykes@imperial.ac.uk

# Step 1: Rule learning

*Observations (traces)*

```
holdsAt(at(loc1), 0).
do(pickup, 0).

holdsAt(at(loc1), 1).
holdsAt(holdingObj, 1).
do(move(loc1, loc3), 1).

holdsAt(at(loc3), 2).
holdsAt(holdingObj, 2).
do(move(loc3, loc5), 2).

holdsAt(at(loc5), 3).
holdsAt(holdingObj, 3).
do(putdown, 3).
```

Learned rules result in new transitions in the domain model

*Explanatory rules (hypothesis)*

```
succ(move(loc3, loc5), T) :-
        holdsAt(at(loc3), T),
        holdsAt(holdingObj, T).
```

daniel.sykes@imperial.ac.uk

# Many traces, many hypotheses

- Traces may exhibit inconsistent behaviour

Maximum likelihood hypotheses has greatest probability of explaining observations

```
holdsAt(at(loc1), 0).
do(pickup, 0).
holdsAt(at(loc1), 1).
holdsAt(holdingObject, 1).
do(move(loc1, loc3), 1).
holdsAt(at(loc3), 2).
holdsAt(holdingObject, 2).
do(move(loc3, loc5), 2).
holdsAt(at(loc5), 3).
holdsAt(holdingObject, 3).
do(putdown, 3).
```

```
succeeds(move(loc3, loc5), T) :-
        holdsAt(at(loc3), T),
        holdsAt(holdingObject, T).
```

daniel.sykes@imperial.ac.uk

# Step 2: Probability estimation

*Probability of a hypothesis* h

$$P_0^\theta(h) = \prod_{a\in h}\theta_a \prod_{a\in A\setminus h}(1-\theta_a)$$

| | *Explains trace – increase probability* | *Does not explain trace – decrease probability* | |
|---|---|---|---|
| | Rule 1 | Rule 2 | … |
| **`holdsAt(holdingObject, T)`** | $\theta 11$ | $\theta 21$ | … |
| **`holdsAt(at(loc3), T)`** | $\theta 12$ | $\theta 22$ | … |
| … | … | … | … |

daniel.sykes@imperial.ac.uk

# Step 2: Probability estimation

*Minimise (by gradient descent)*

*MSE of current estimates θ*

$$MSE(\bar{\theta}) = \frac{1}{|X|}\sum_i (1 - P^\theta(x_i|\Delta \cup \chi_i))^2$$

*i.e. maximise prob. of hyp. predicting observations*

*Predictive ratio for observation x*

$$P^\theta(x_i|\Delta \cup \chi_i) = \frac{\boxed{\sum_{\{h\in\Delta, h\cup\chi_i \models x_i\}} P_0^\theta(h)}}{\sum_{\{h\in\Delta\}} P_0^\theta(h)}$$

# Applying learned rules

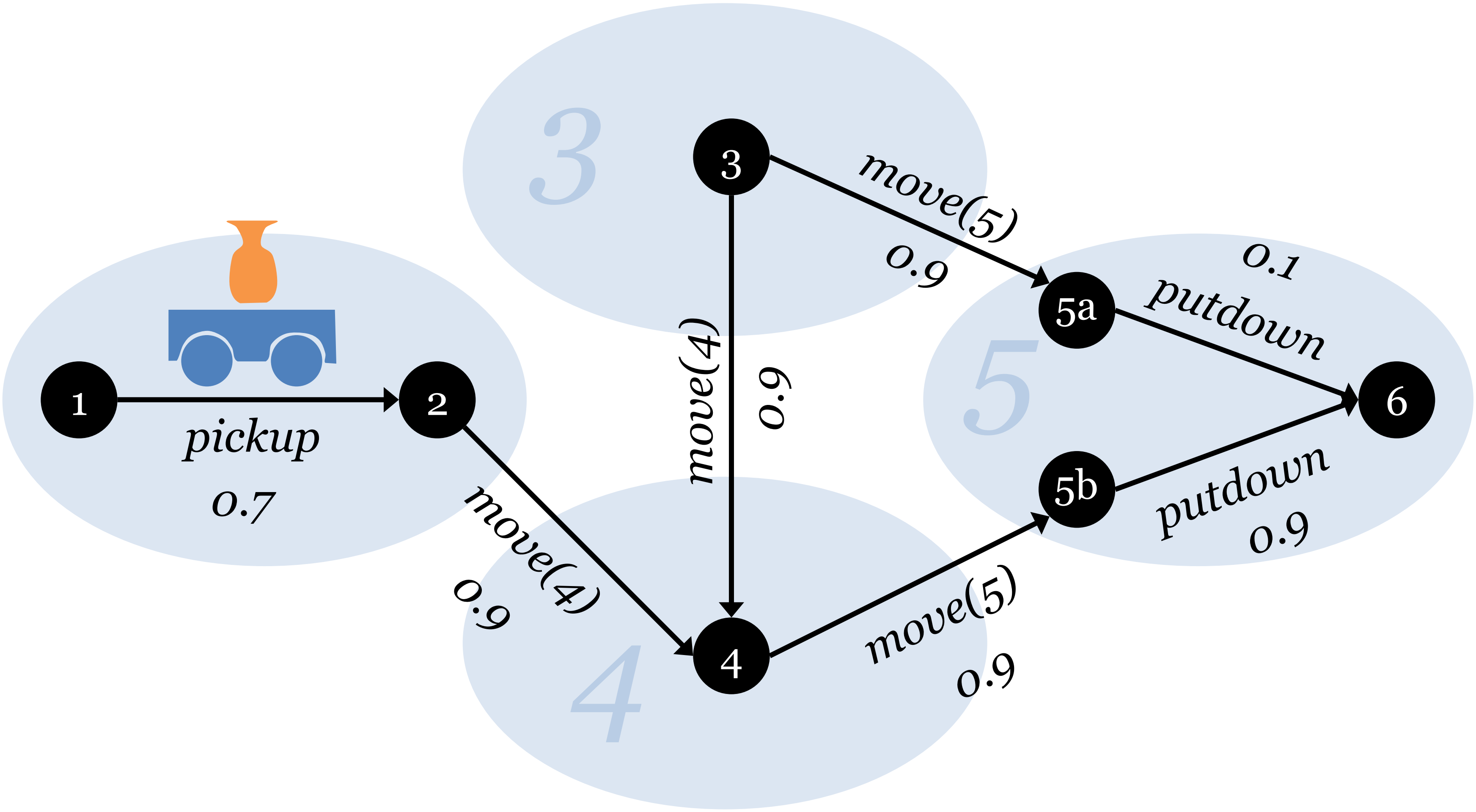
```
r1:0.7 : succeeds(pickup, T).
r2:0.9 : succeeds(move(L1, L2), T) :-
              holdsAt(at(L1), T),
              connected(L1, L2),
              L2 != loc3.
r3:0.9 : succeeds(putdown, T) :-
              not happened(move(loc2, loc3), T-2).
r4:0.1 : succeeds(putdown, T) :-
              happened(move(loc2, loc3), T-2).
```

*Rule probabilities calculated from condition probabilities*

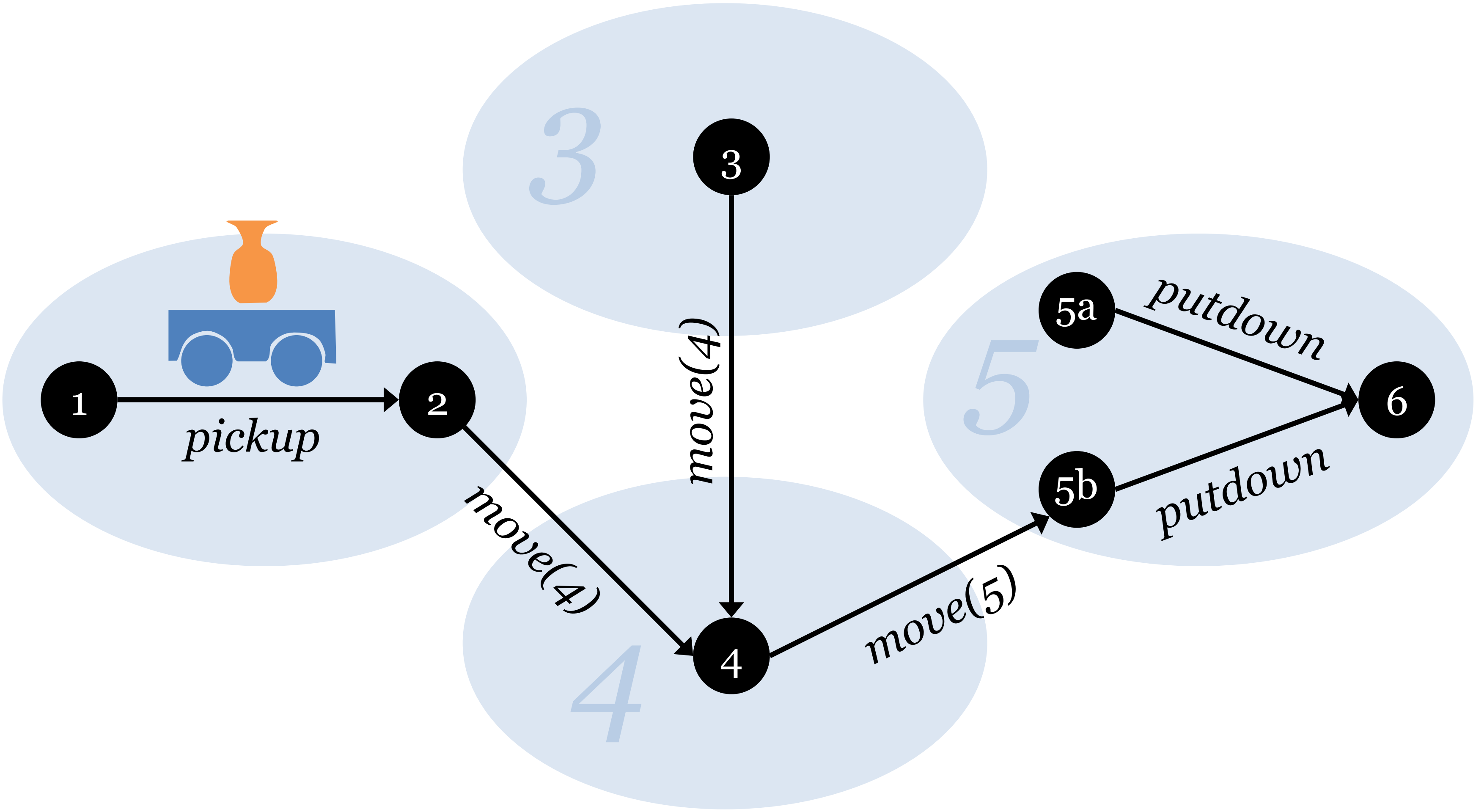
Learned rules result in new states and transitions in the domain model – with probabilities

daniel.sykes@imperial.ac.uk

# Updated factory floor model



# New factory floor plan

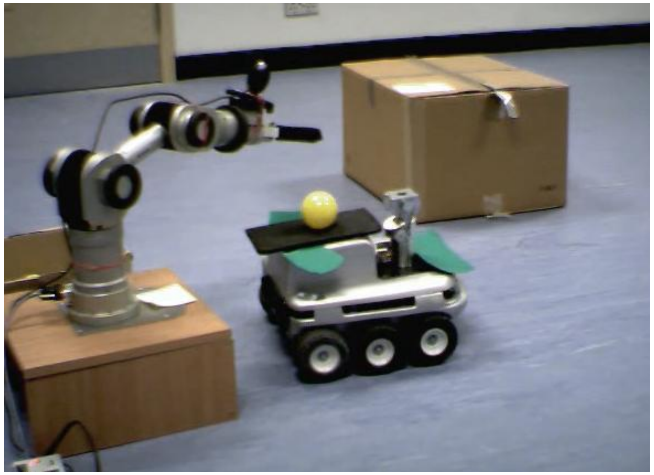


# Experience

Robot navigation: global failure rate reduced from 30% to 10%

Human-readable explanations

```
succeeds(move(L1, L2), T) :-
                holdsAt(at(L1), T),
                connected(L1, L2),
                L2 != loc3.
```



daniel.sykes@imperial.ac.uk

# Challenges

- High complexity of ILP
  - Limit rule length with mode declarations
- Improve tool support (GD, integration)
- Scope for adaptation is limited by set of actions and sensed conditions
  - Cannot learn rules based on conditions not present in traces
- Opportunity for starting from a minimal model – exploration of environment

daniel.sykes@imperial.ac.uk

# Summary

- Behavioural model revision using ILP
  - Traces gathered from plan execution
  - Missing states/transitions
  - Estimated probabilities, find maximum likelihood hypothesis
  - Mitigate inaccuracy and incompleteness (*uncertainty*) in model
- Revised model remains human-readable

daniel.sykes@imperial.ac.uk

Thanks, questions?

daniel.sykes@imperial.ac.uk