PLP IN PRACTICE

Wannes Meert

PLP 2014, July 17

With contributions by Jean-Michel Papy, Kurt Driessens, Joris Renkens, Davide Nitti, Guy Van den Broeck, Tinne De Laet, Siegfried Nijssen, Kathleen Marchal, Vitor Santos Costa, David Page, Jesse Davis, Hendrik Blockeel, Luc De Raedt

KU LEUVEN

Interacting with PLP languages

Open, continuous worlds

Diagnostics

Robotics

Biology

Healthcare

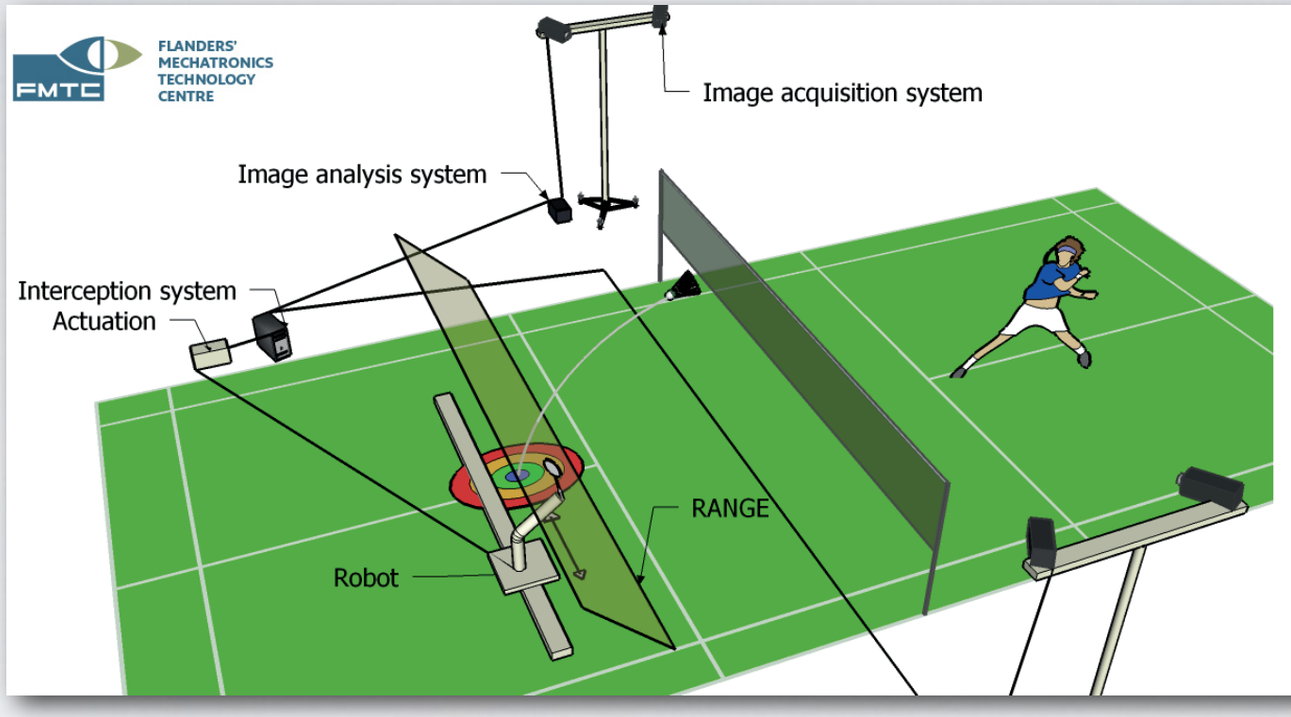
Prefer fast to correct

Building background knowledge

Diagnostics

PLP FOR DIAGNOSTICS

Jada, the badminton playing robot



If the shuttle is not returned, what went wrong?

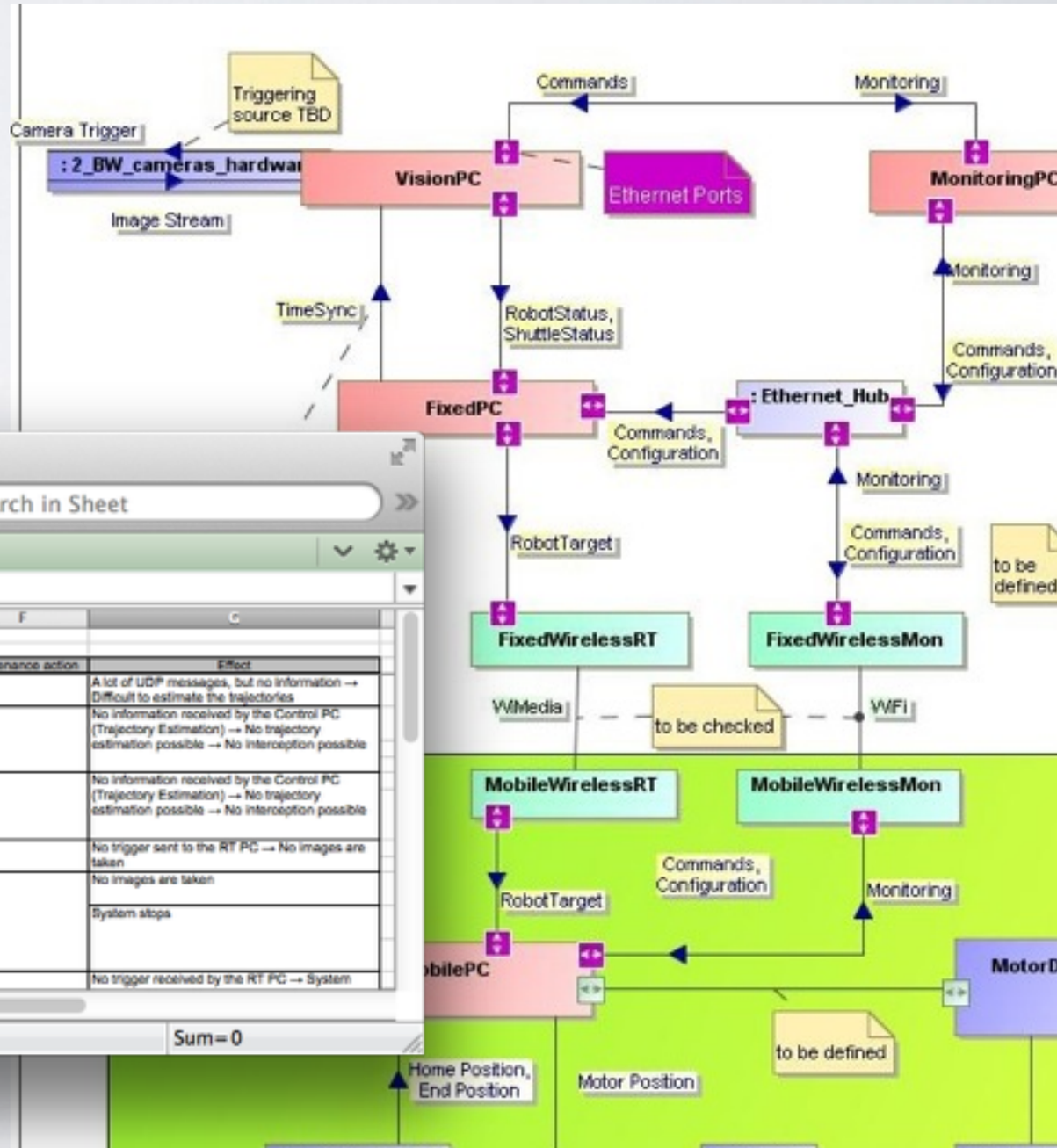
FMTC
INNOVATING TOGETHER FOR SUPERIOR MACHINES

FAILURE MODE METHODOLOGY

List the

- Components
- Functions
- Failure Modes
- Causes
- Detections
- Effects

Tools to communicate, not to reason automatically



fmea-worksheet-robot

Search in Sheet

Home | Layout | Tables | Charts | SmartArt | Formulas | Data | Review

G28

TO BE DIAGNOSED

Function	Component	Failure mode	Cause	Detection	Maintenance action	Effect
Start-up/Play	Calibration	Calibration failure	Calibration data wrong	Calibration check (to be implemented)		A lot of UDP messages, but no information → Difficult to estimate the trajectories
Start-up/Play	Network	Network communication failure	Ethernet cable disconnected	No diagnostic messages received by the Diagnostic PC		No information received by the Control PC (Trajectory Estimation) → No trajectory estimation possible → No interception possible
			Ethernet cable broken	No UDP messages received by the Diagnostic PC		
			Network switch broken			
Start-up/Play	RT PC	RT software not running	RT PC software has not been started	No messages from the RT PC received by the Diagnostic PC		No information received by the Control PC (Trajectory Estimation) → No trajectory estimation possible → No interception possible
			RT PC software has crashed			
Start-up/Play	Control PC	Control software not running	Control PC software has not been started	No messages from the Control PC received by the Diagnostic PC		No trigger sent to the RT PC → No images are taken
			Control PC software has crashed			
Start-up/Play	Camera	Camera not functioning	Something or someone has unplugged the camera	Error message		No images are taken
			Something or someone has broken the camera cable			System stops
			Something or someone has broken the camera			
Play	EtherCAT	EtherCAT communication failure	EtherCAT cable disconnected	UDP message rate sent by RT PC = 0		No trigger received by the RT PC → System

Extended FMEA (simplified)_2

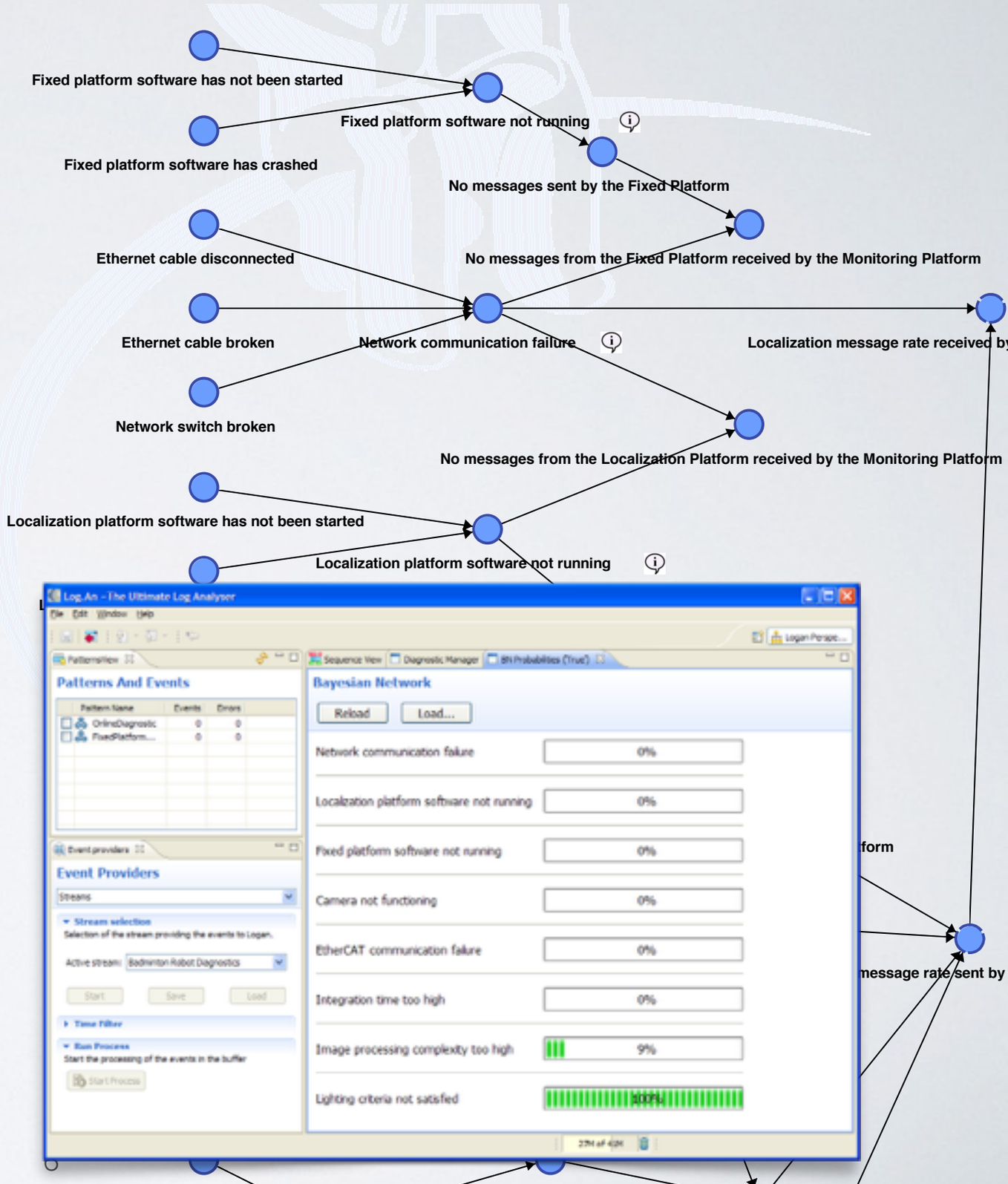
Normal View | Ready | Sum=0

BAYESIAN NETWORK METHOD

Moved to a computer interpretable model to infer causes of failures based on observations

But:

- Loss of hierarchy of components
- No idea how to fill in CPTS
- Too many causes generate too large CPTs



WHY PLP?

Hierarchy of components:

```
…
partof(visShuttleTrackSys, robot).
partof(Cam, visShuttleTrackSys) :- camera(Cam).
…
partof(armBatt, armActuationSys).
```

Propagation of failures:

```
Prob? failure(robot,noreturn) :-
          failure(visShuttleTrackSys, noShuttlePosEstimate).
      …
Prob? failure(armActuationSys,armNotAct) :-
          failure(motorBatt, insufPower).
      …
      0.8::failure(motorBatt, insufPower).
```

MULTI-VALUED VARIABLES

```
domain messrate = [0,50,75,+]. # {"<=50","<=75",">75"}.
```

Noisy-Max (causal) combinations

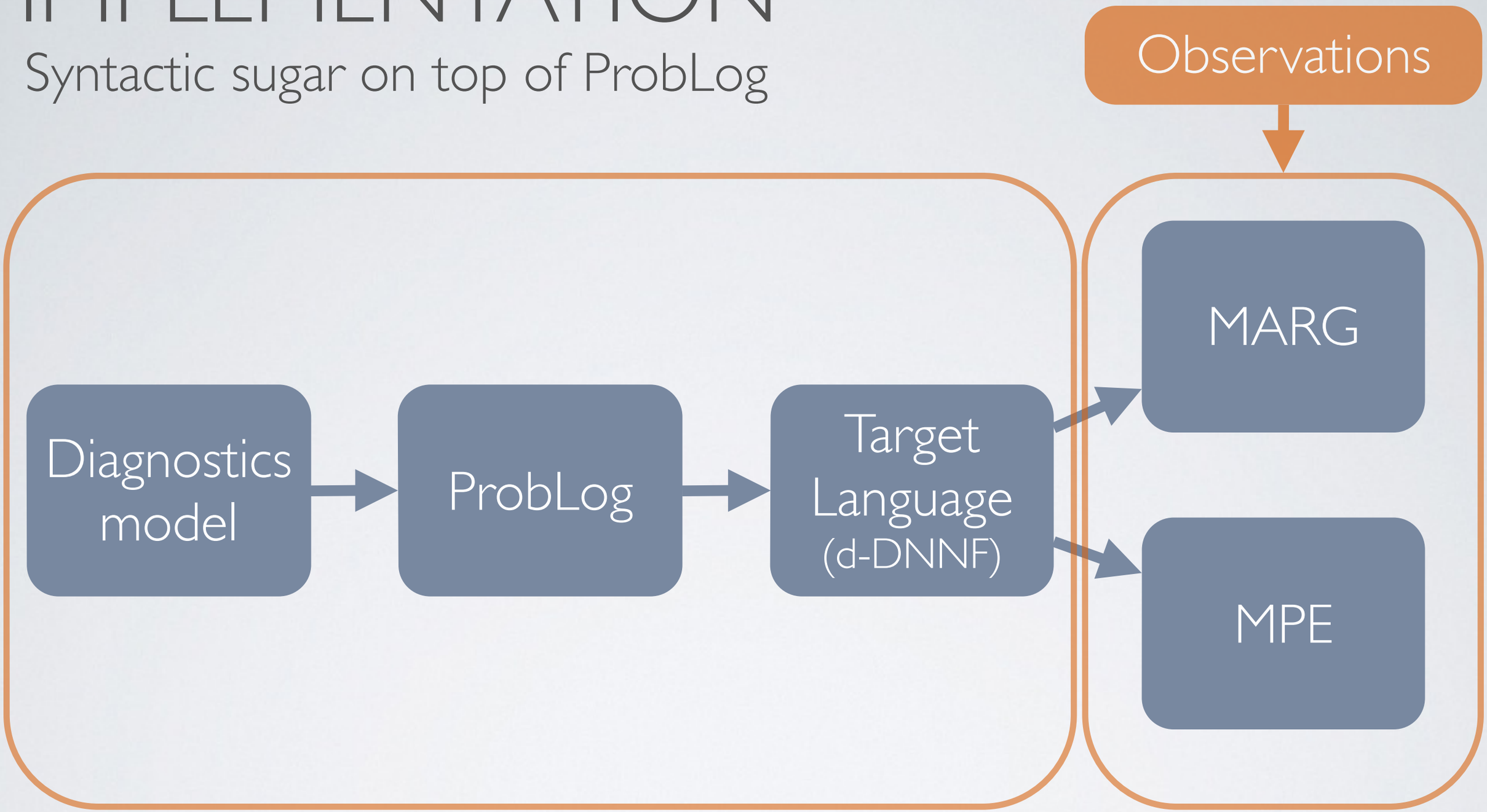
```
#                [0    ,50   ,75   ,+]
rec_rate:[0.98,0.01,0.01] :-  netfail.
rec_rate:[0.98,0.01,0.01] :- !netfail, sen_rate=messrate(0).
rec_rate:[0.10,0.90,0.01] :- !netfail, sen_rate=messrate(1).
rec_rate:[0.01,0.19,0.80] :- !netfail, sen_rate=messrate(2).
```

Need for multi-valued variables

Decision tree like

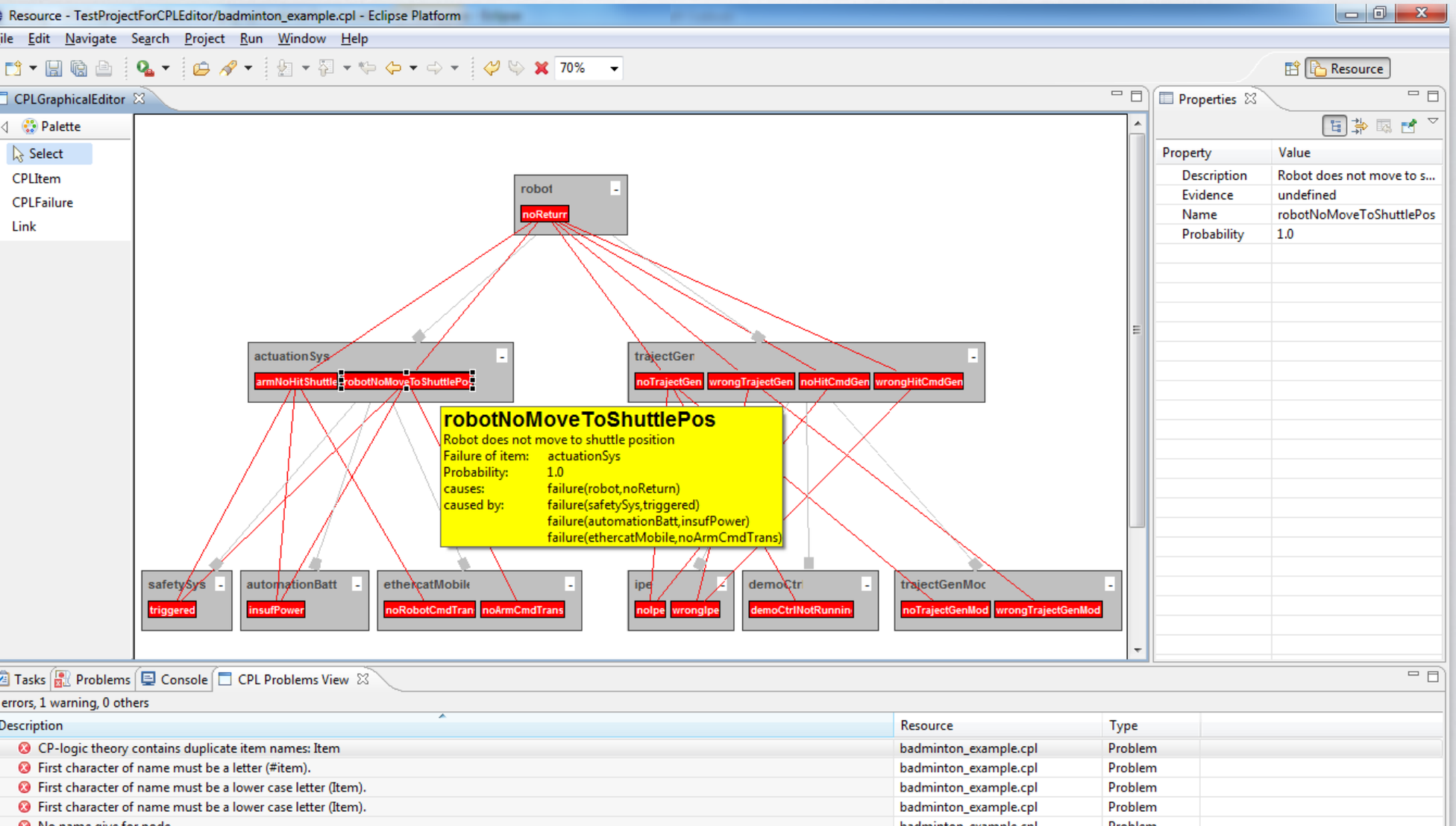
IMPLEMENTATION

Syntactic sugar on top of ProbLog



INSPECTABILITY

We added Bayesian net export to show correctness
→ SMILE XDSL and Norsys DNE cover most tools
We added exhaustive sanity checking
→ E.g. detection of always true/false random variables
We added visualisations based on the application domain



[Schramm et al. 2012]

PLP NEEDS

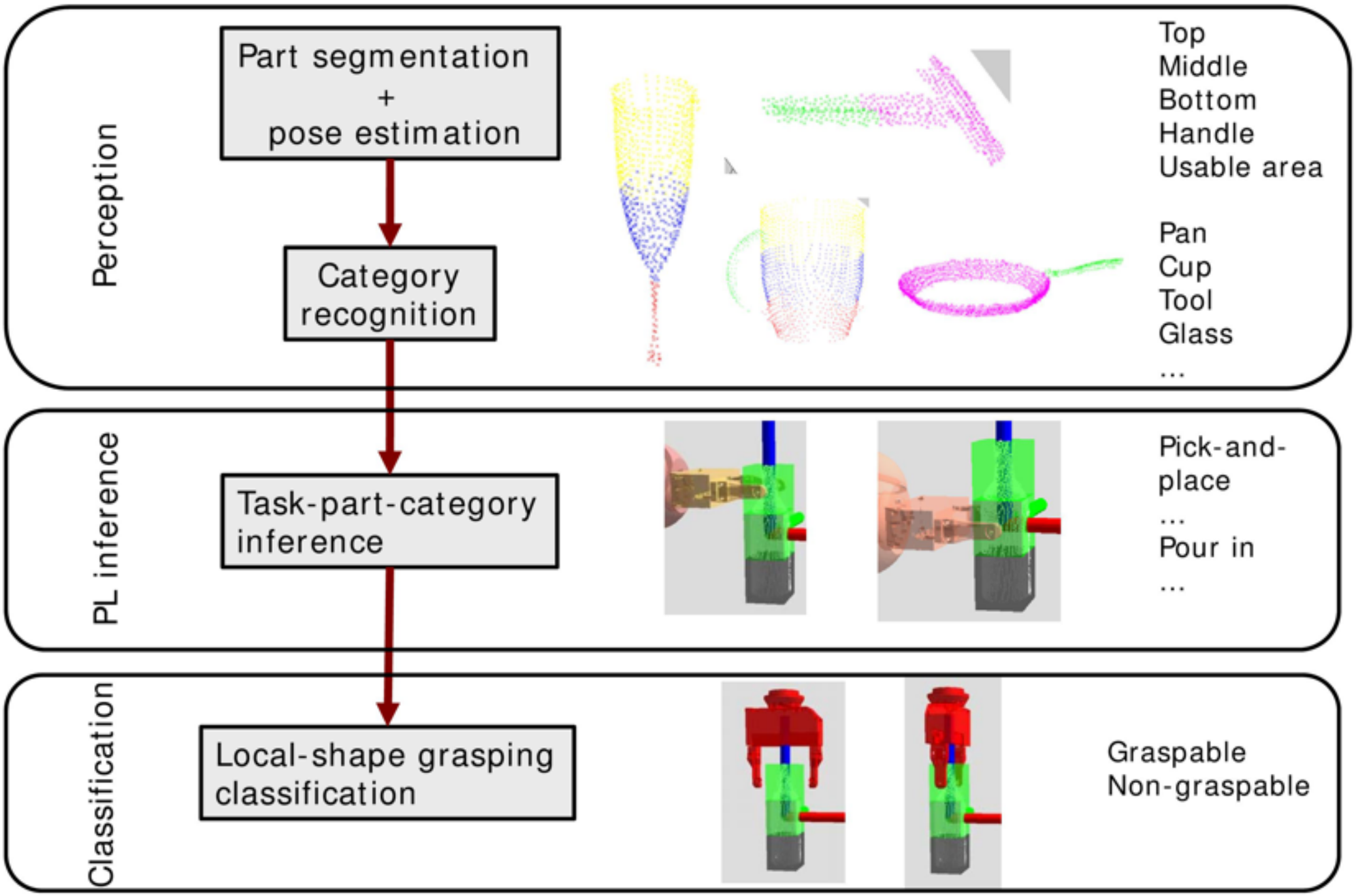
Boolean is too restricted, we think in categories. People want multi-valued variables

Modelling and evaluation are different tasks. Build separate tools

Debugging and visualisation are crucial

Robotics

PROBABILISTIC LOGIC PIPELINE

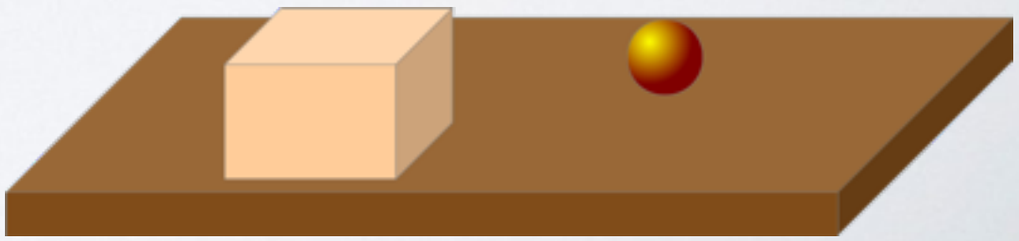


WHY PLP?

- Robotics is based on propositional models (BN) but relational representation is useful
- Bridging this gap is hard because most probabilistic logic languages have
  - Discrete representations
  - No efficient technique for temporal models
  - No open worlds

Relational:

- Objects (physical or abstract entities)
- Properties (color, utility, shape, size, …)
- Relations (spatial relations, interactions, …)
- Background knowledge (box is a container, balls have low friction)
- Evolution over time



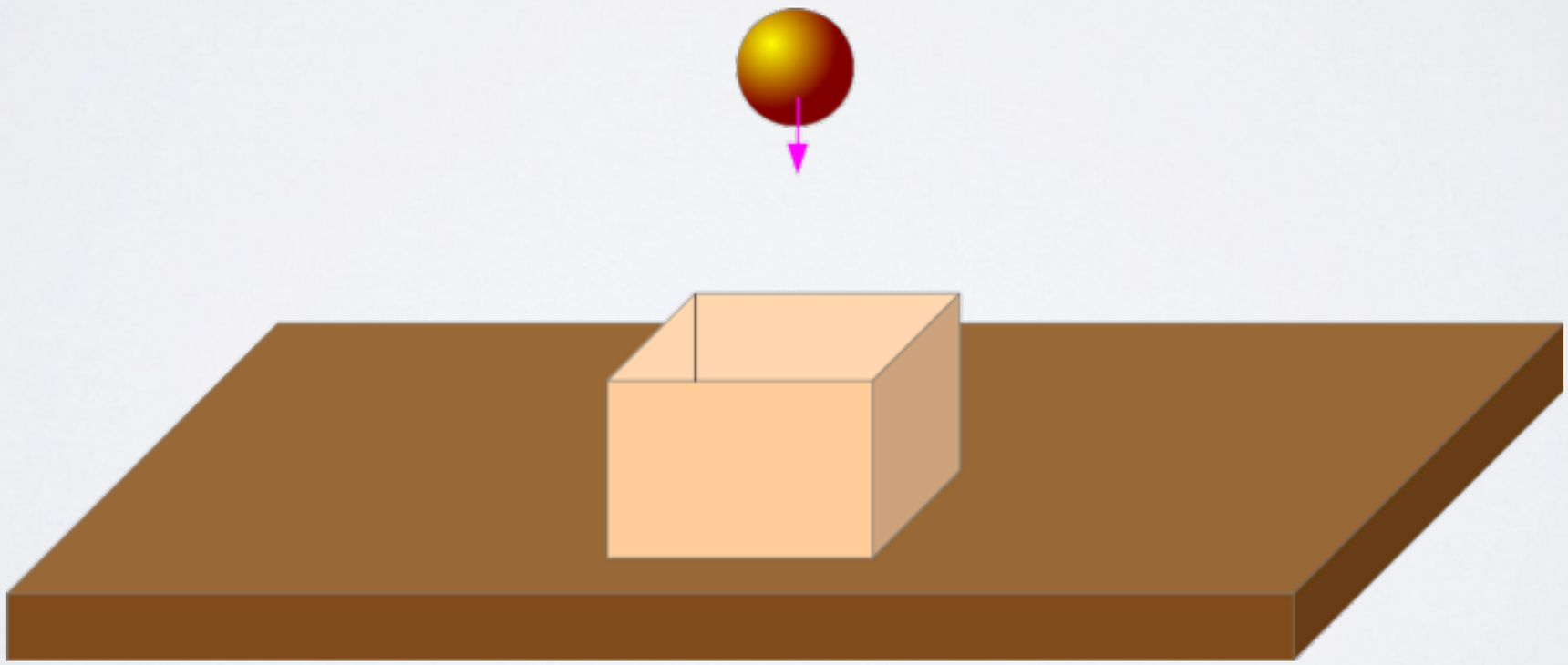
DYNAMIC DISTRIBUTIONAL CLAUSES

Noisy observations

$on_t(A,B) \leftarrow A \neq B, posZ_t(A) > posZ_t(B), posZ_t(A) < posZ_t(B) + const, \ldots$

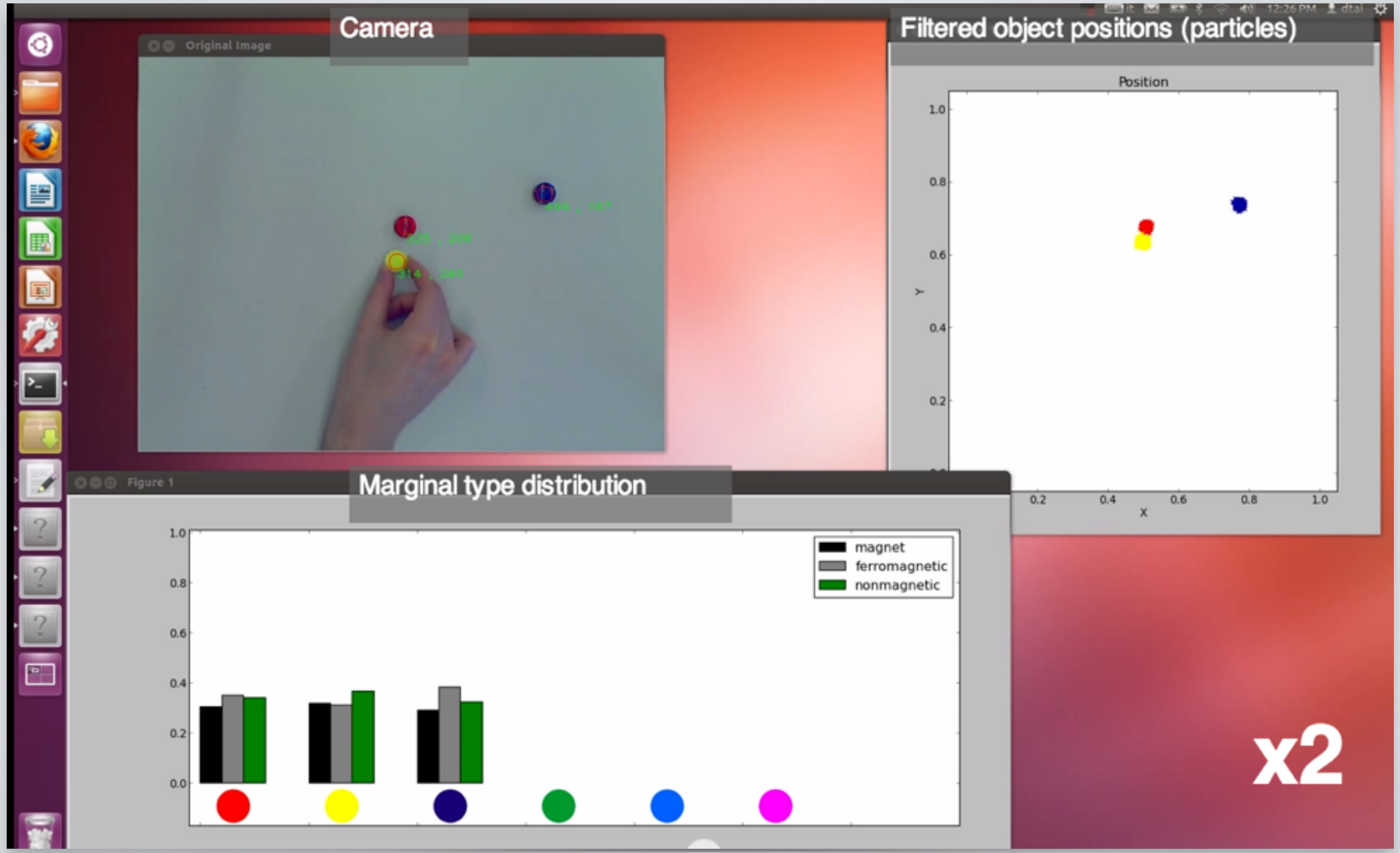
$accZ(A)_{t+1} \sim gaussian(0, 0.01) \leftarrow object(A), on_t(A,B).$

$accZ(A)_{t+1} \sim gaussian(-g, 0.01) \leftarrow object(A), !on_t(A,B).$



[Nitti et al., IROS 2013 & ICRA 2014]

ENCODE RULES OF MAGNETISM

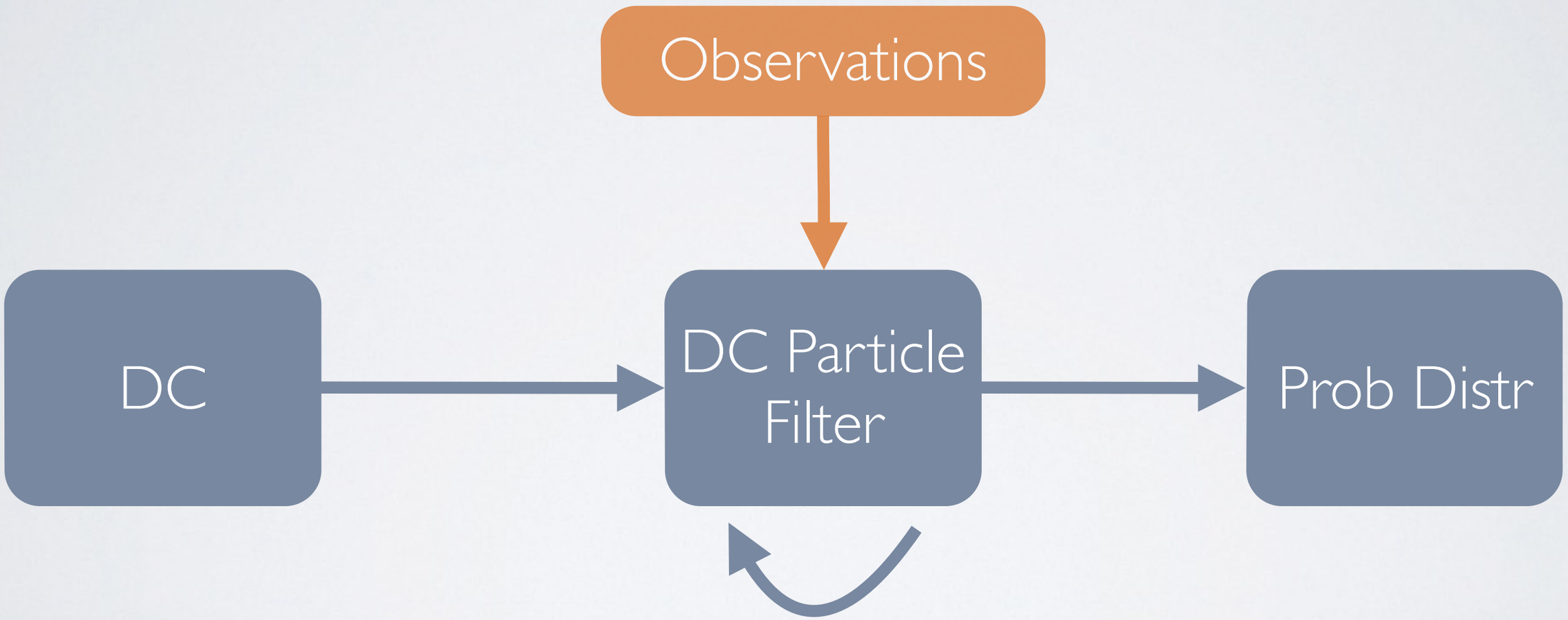


https://dtai.cs.kuleuven.be/ml/systems/DC/dcpf.html

IMPLEMENTATION

Relational Particle Filter

No exact approach feasible because of continuous functions



[Nitti et al., IROS 2013 & ICRA 2014]

PLP NEEDS

Continuous variables and mathematical expressions

This is wat makes Probabilistic Programming a success

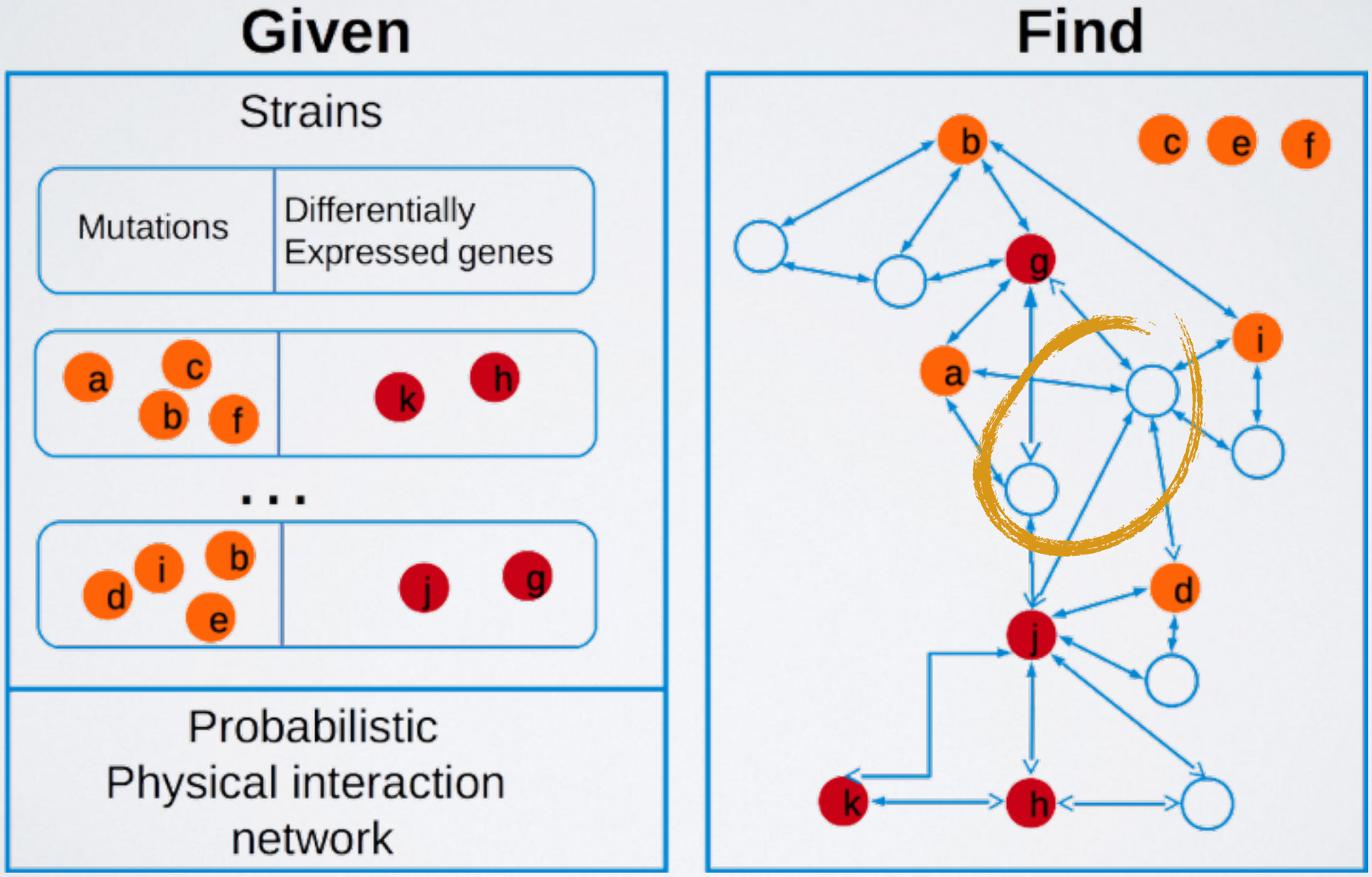
Open worlds, unknown number of objects a priori

Biology

PROBLEM STATEMENT

Large-scale — Noisy

Goal: Find gene-expression pathways from high-throughput data

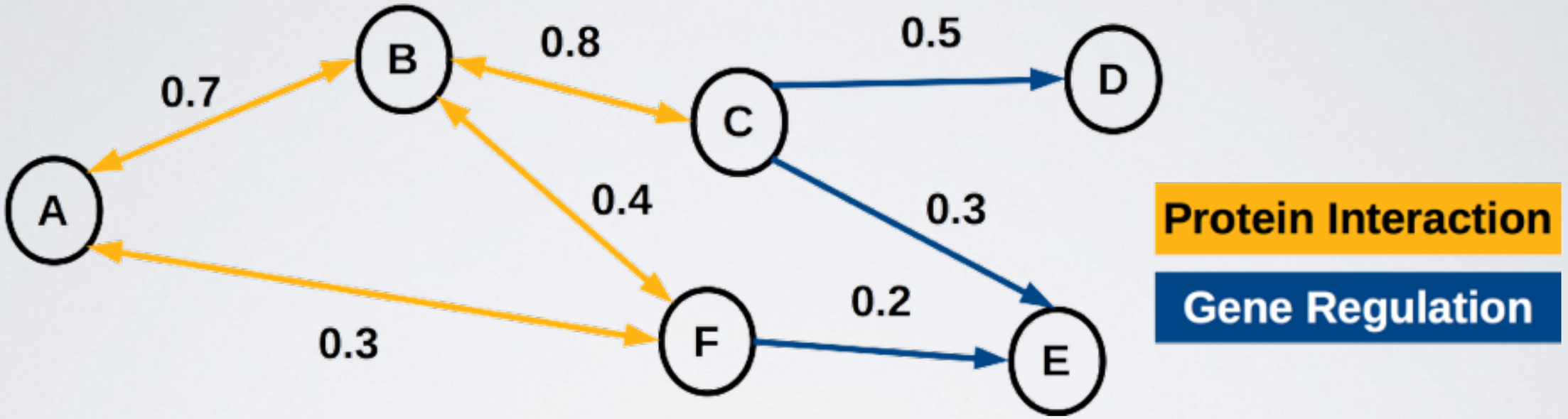


$$Cost = |edges| \cdot c_e - \sum_{(i,o)} P(path(i,o)) \cdot c_p$$

[De Maeyer et al., Molecular Biosystems 2013]

BIOLOGICAL INTERACTION NETWORKS

Model complex, probabilistic data



Define complex queries

WHY PLP?

Easy to change problem definition and input data without reimplementing the algorithm

→ New types of pathways, new types of interactions

Possible to plug in novel inference/optimization techniques without changing the problem definition

→ Switch between different approximate reasoning techniques

IMPLEMENTATION

E(#mutations connected to differentially expressed genes)
→ Natural for PLP inference

Intractable due to size of data: K-Best inference

Causal mutations can be connected to multiple differentially expressed genes with a small subnetwork

Finding optimal subnetwork
→ Decision Theoretic ProbLog

PLP NEEDS

Approximate inference techniques

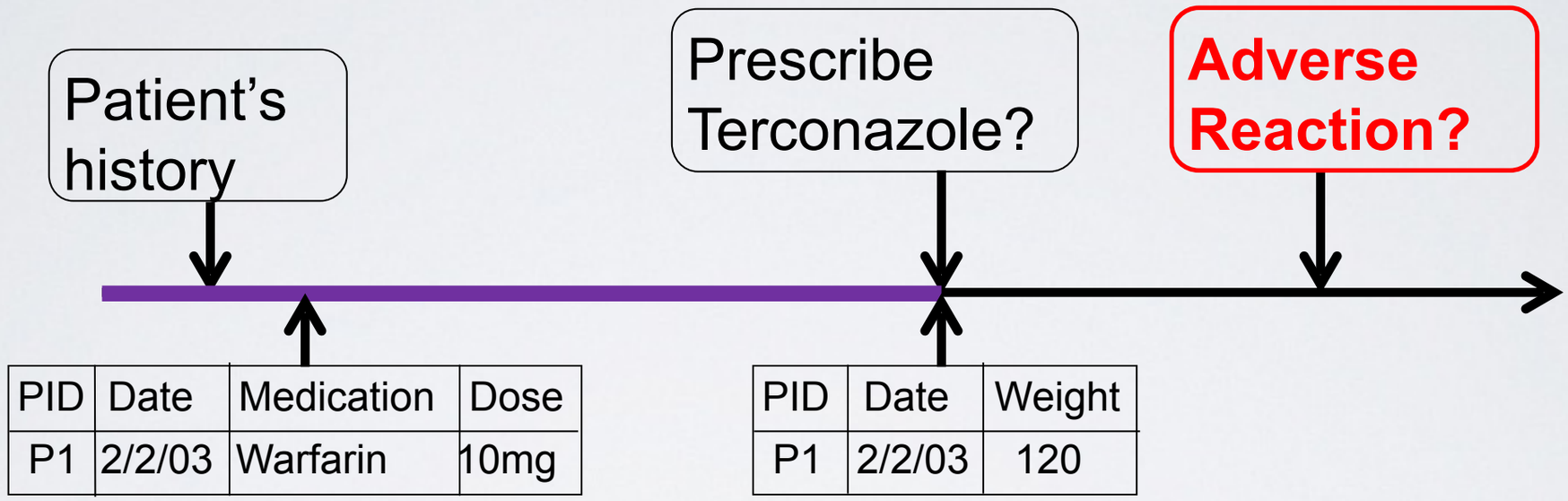
Inference is not enough, people need decision support

Healthcare

Integrate feature/predicate invention and program construction

[Davis et al., IJCAI 07, ICML 09]

ADVERSE DRUG REACTIONS



Will a patient have an adverse reaction to a drug?
What is the efficacy of a drug for a patient?
How will the patient's disease progress?

[Davis et al., ICML 2012]

ADVERSE DRUG REACTIONS

Drug

PID	Date	Medication	Dose
P1	5/1/02	Warfarin	10mg
P1	2/2/03	Terconazole	10mg

Diseases

PID	Date	Diag.
P1	2/1/01	Flu
P1	5/2/03	Bleeding

Observation

PID	Date	Weight
P1	2/2/03	120

0.85::Reaction(p) :- Drug(p, Zoloft) Lab(p, BloodSugar) …

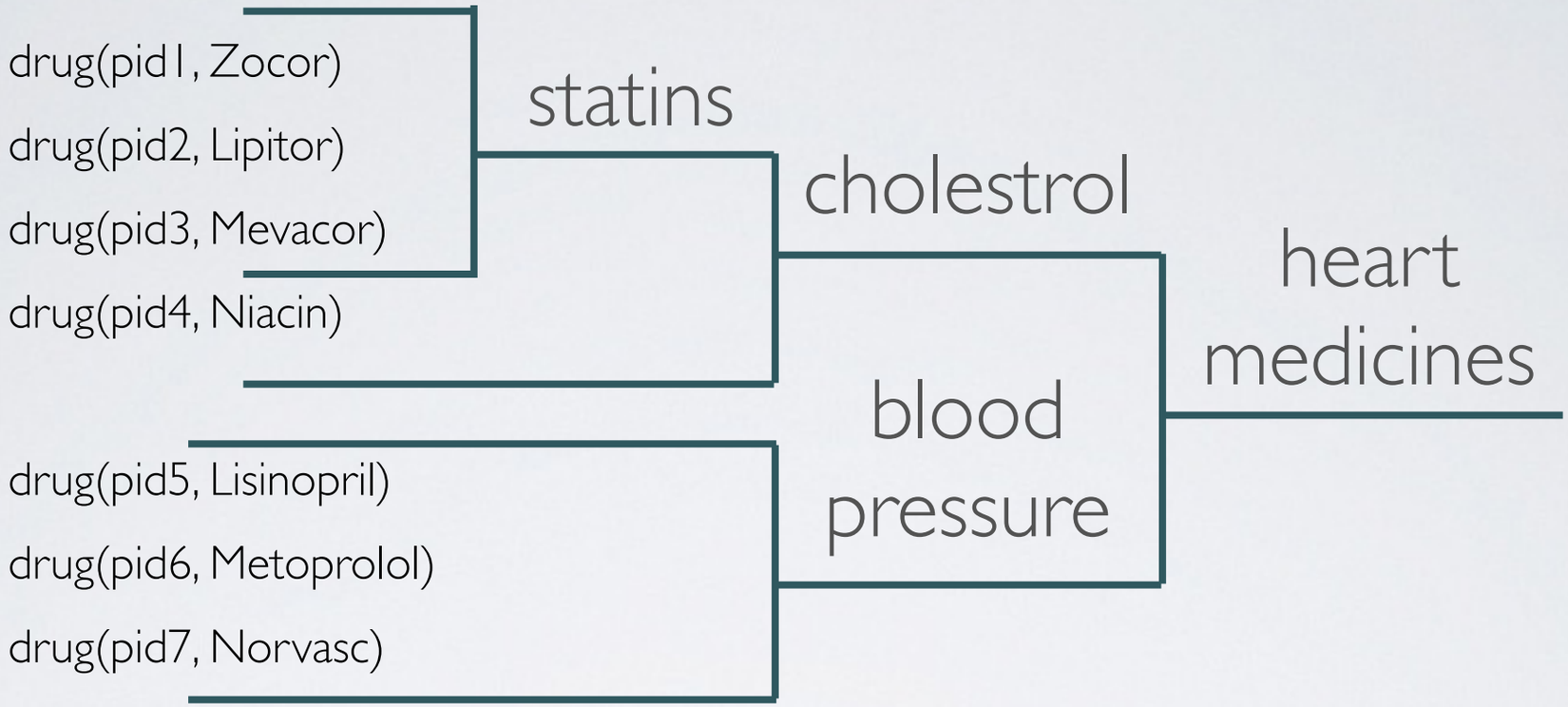
0:80::Reaction(p) :- Drug(p, Terconazole) ∧ Wt(p, w) ∧ w < 120.

Data and patterns refer to specific drugs or diseases

Regularities may involve drug or disease classes
E.g. enzyme inducers increase risk of internal bleeding

[Davis et al., ICML 2012]

GROUPINGS



Learn groupings	Use expert hierarchy
- Learn clusters about objects by altering learned rules - Use learned clusters in rules	- No hierarchy for drugs - ICD9/ICD10 exist for diseases, but there are some arbitrary choices - Unclear grouping: should we cluster drugs by diseases they treat or side-effects?

[Davis et al., ICML 2012]

WHY PLP?

Reacion(p) :- Drug(p, Zoloft) ∧ … ∧ …

Generalize rule to refer to learned (hierarchical) clusters of constants

Reaction(p) :- Hier_11(x) ∧ Drug(p, x) ∧ … ∧ …

Learn hierarchy definition by

Assign objects to hierarchy

Hier_11(Zoloft)
Hier_11(Paxil)
…

Reuse previously learned groups

Hier_11(x) :- Hier_8(x).

Implicit generation of hierarchical background knowledge and easy inclusion

[Davis et al., ICML 2012]

EMPIRICAL EVALUATION

[Davis et al., ICML 2012]

PLP NEEDS

To stay focussed on learning parameters and structure

Easy learning and merging of background knowledge

LESSONS LEARNED

Inspectable en visualizable programs

Multi-valued and continuous variables

Combine expert and experimental knowledge

Optimisation and decision support are valuable

Programs designed by non-PLP-experts quickly grow intractable

Real-world data and knowledge is intractable

Thank you.

KU LEUVEN