



Why are trivial things so hard?

LAMAS SING micro-talk #1

Jacek Malec

Dept. of Computer Science, Lund University, Sweden

September 23, 2015



Robotics and Semantic Systems

Website: rss.cs.lth.se

Our interests:

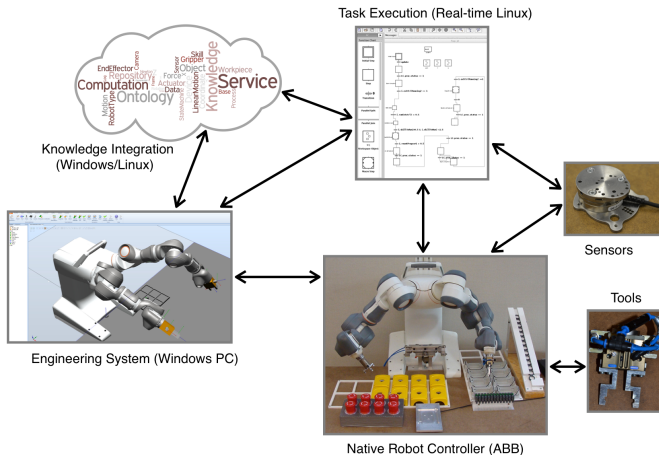
- natural language processing
- human-robot interaction
- knowledge representation and reasoning
- robotics and automation software

Our projects:

- EU: mostly robotic ones
- Swedish: mostly NLP



Context of my question





Some nice movies

MOVIES

illustrating learning by being told

Work of primarily Maj Stenmark

- 1 loopmovie_cut.mov
- 2 constraint.mov
- 3 cut2.mov



The question part: scheduling

Goal: to schedule robot arm(s) moves, taking into account: time of the moves, space (collisions), reachability, tools available, ...



The question part: scheduling

Goal: to schedule robot arm(s) moves, taking into account: time of the moves, space (collisions), reachability, tools available, ...

Why don't we use constraint satisfaction methods? There are so many effective CP tools around (check JaCoP at <http://jacop.cs.lth.se>)



The question part: scheduling

Goal: to schedule robot arm(s) moves, taking into account: time of the moves, space (collisions), reachability, tools available, ...

Why don't we use constraint satisfaction methods? There are so many effective CP tools around (check JaCoP at <http://jacop.cs.lth.se>)

model → solver



The question part: scheduling

Goal: to schedule robot arm(s) moves, taking into account: time of the moves, space (collisions), reachability, tools available, ...

Why don't we use constraint satisfaction methods? There are so many effective CP tools around (check JaCoP at <http://jacop.cs.lth.se>)

model → solver

models:

- MINIZINC (nice language, close to FOL)
- FLATZINC (constraints made clear)



The question part: scheduling

Goal: to schedule robot arm(s) moves, taking into account: time of the moves, space (collisions), reachability, tools available, ...

Why don't we use constraint satisfaction methods? There are so many effective CP tools around (check JaCoP at <http://jacop.cs.lth.se>)

model → solver

models:

- MINIZINC (nice language, close to FOL)
- FLATZINC (constraints made clear)

Surprise! Two arms + three tools + 5 parts + 4 assembly operations **explode** all solvers we tried



Reference

Tommy Kvant's Master's thesis (2015):
Task scheduling for dual-arm industrial robots through Constraint
Programming
<https://lup.lub.lu.se/student-papers/search/publication/5384696>



Relation to MAS Interaction?

At least two agents:

- 1 human instructor
- 2 robot

We would like to be able to show (prove?) that the human's intentions will be realized, possibly in an optimal way.

Sometimes interaction requires clarification (do you really mean that?)

Sometimes it is the robot who corrects the human (Usually you put a washer under a nut while screwing two items together!)



Relation to MAS Interaction?

At least two agents:

- 1 human instructor
- 2 robot

We would like to be able to show (prove?) that the human's intentions will be realized, possibly in an optimal way.

Sometimes interaction requires clarification (do you really mean that?)

Sometimes it is the robot who corrects the human (Usually you put a washer under a nut while screwing two items together!)

Thank you!