Assembly Sequence Planning and Program Generation for Flexible Automation

Ulrike Thomas

Institute for Robotics and Mechatronics, DLR

Content

- Motivation
- Intuitive programming
- Assembly sequence planning
- Methods for robot program generation
- Conclusion

Motivation

Model data base

Skill data base

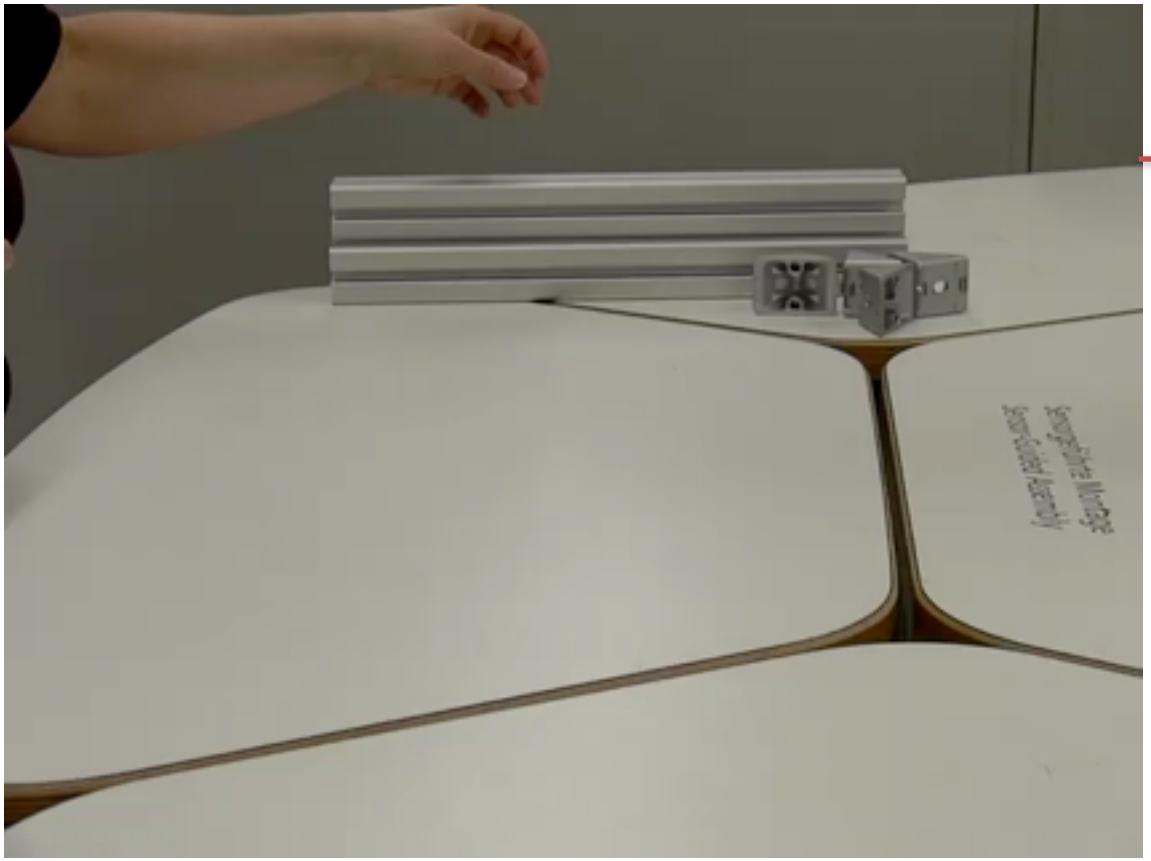
Knowledge data base

The magic something?

Robot Program

What to do?

Programming by Showing



Estimate Position of Objects

Consistency Checks

Specification of Assembly

Note: Sequence during demonstration does not matter

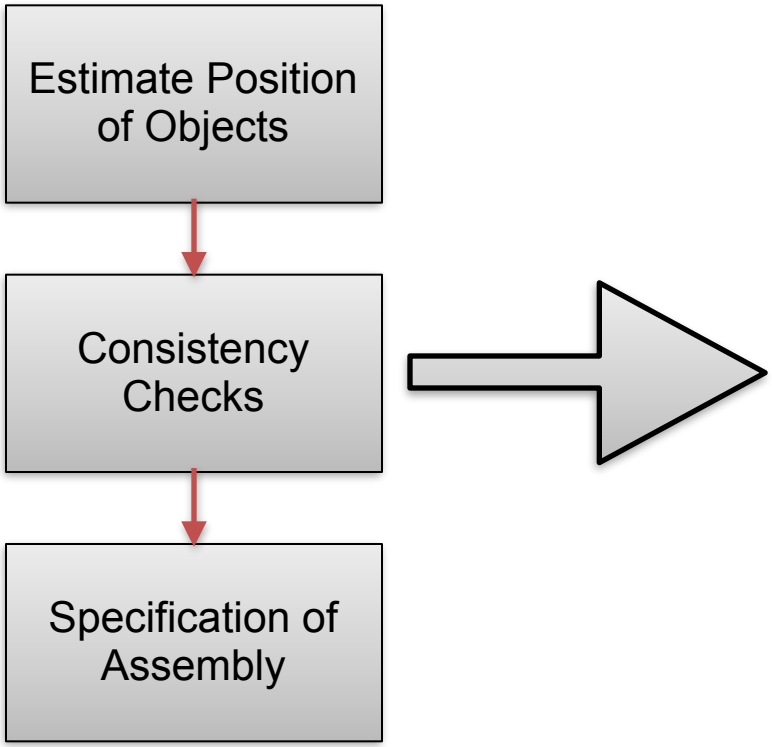
Programming by Showing

Estimate Position of Objects

Consistency Checks

Specification of Assembly

Programming by Showing



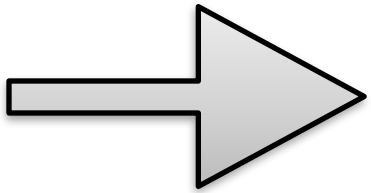
- Are the relative object poses consistent
  - Do objects overlap ?
  - Is the uncertainty space well defined?
- Are models correct, e.g. no violated meshes?
- Does the detection confirm to the knowledge data base
- Is the knowledge complete regarding planning?

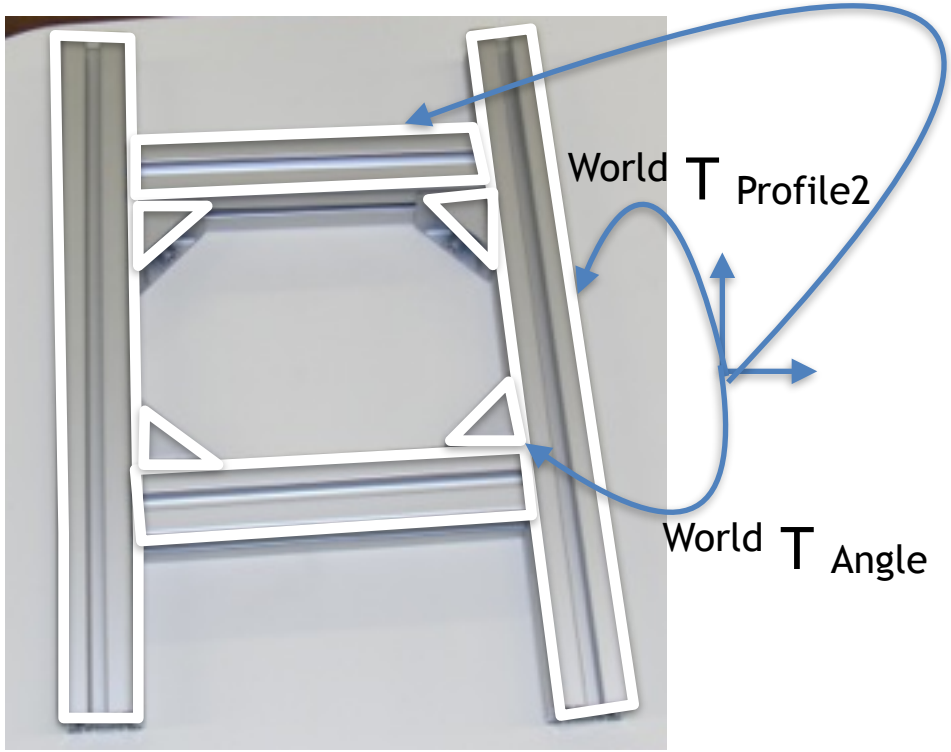
Programming by Showing

Estimate Position of Objects

Consistency Checks

Specification of Assembly





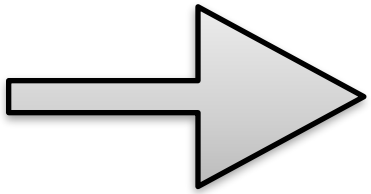
A coherent definition of the assembly group

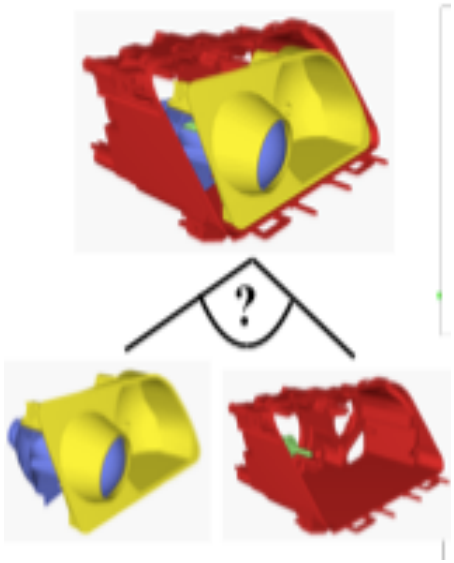
Steps of the Assembly Sequence Planning

Geometric Feasibility Tests

Recursive Graph-Cuts

Evaluation of Subassemblies and Sequences





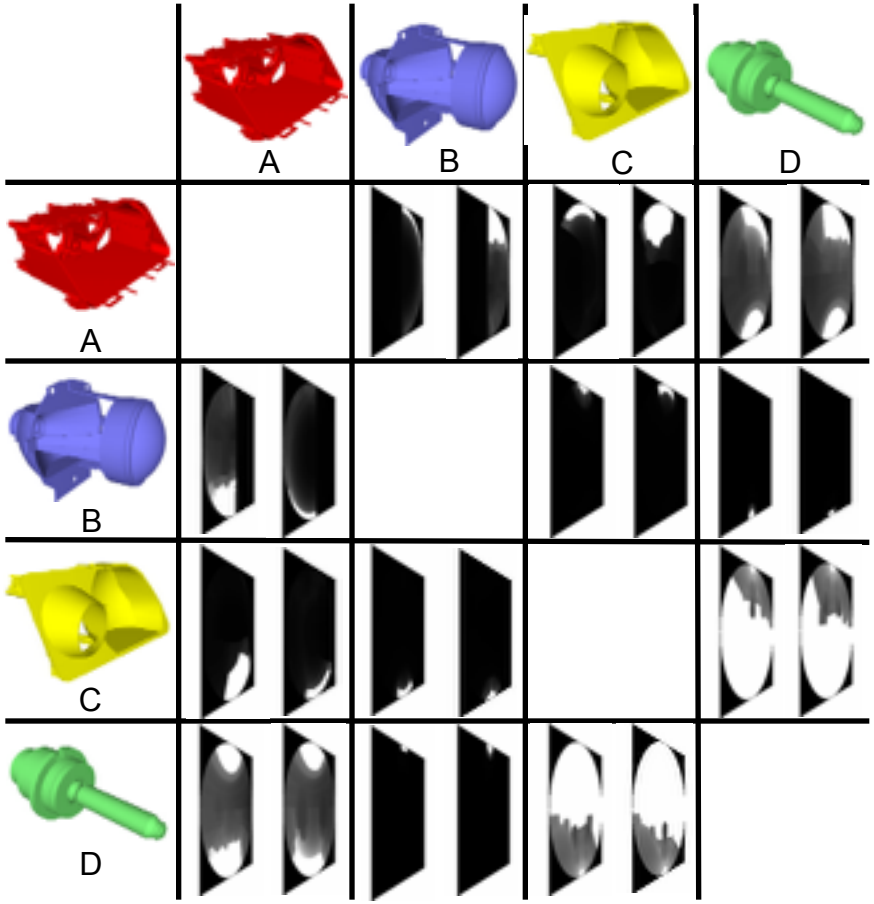
Disassembly into two disjunct groups:

$$\frac{1}{2} \cdot \sum_{i=1}^{n-1} \binom{n}{i} = 2^{n-1} - 1$$

Number of sequences in general:

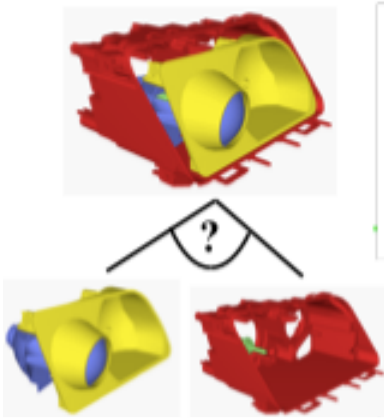
$$\sum_{i=2}^{n} \binom{n}{i} \cdot (2^{i-1} - 1) = \frac{3^n + 1}{2} - 2^n$$

Geometric Feasibility in 3D



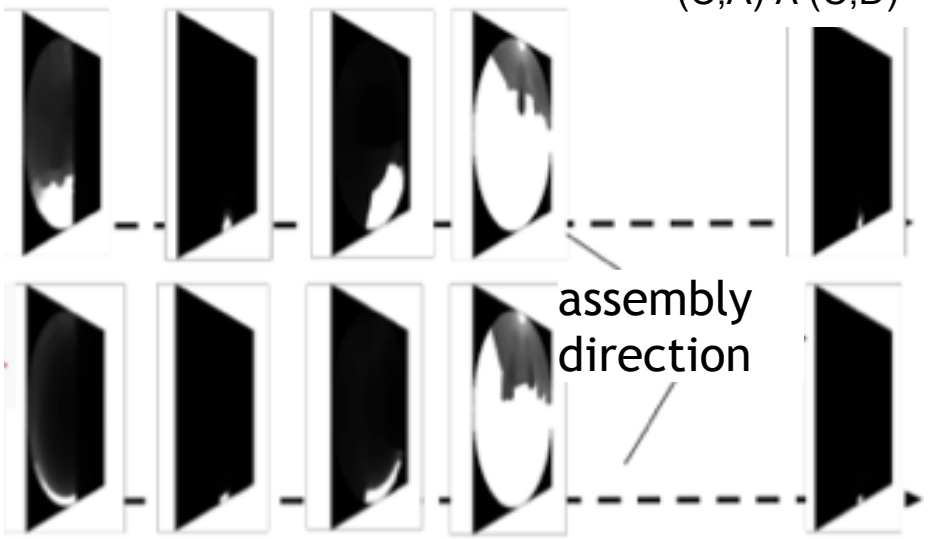
Look-up table of projected c-space obstacles

feasible?
(B,C)/(A,D)

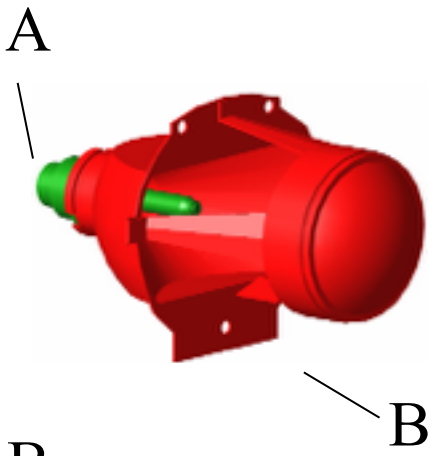


(B,A) (B,D) (C,A) (C,D)

(B,A) ∧ (B,D) ∧
(C,A) ∧ (C,D)



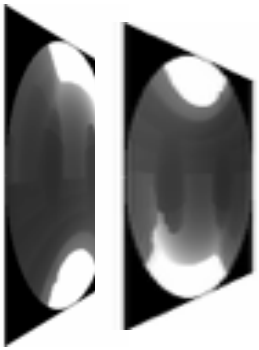
Accurate Estimation of Disassembly Directions



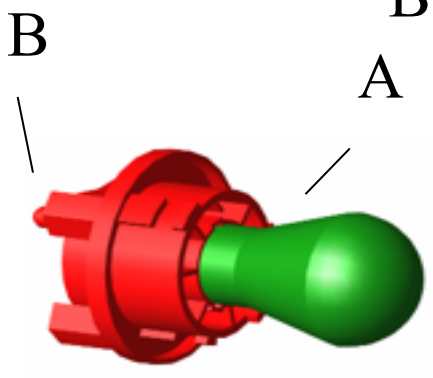
original data

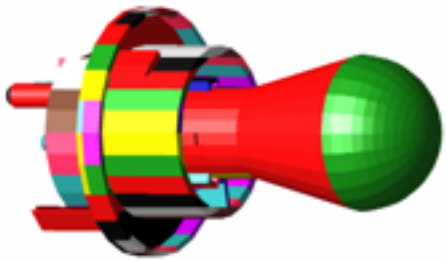
surface decomposition into convex patches

C-Space obstacles



map for disassembly directions





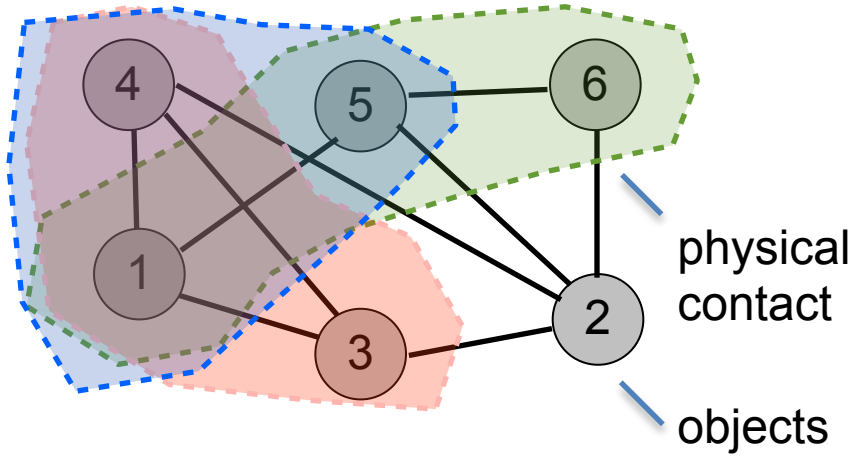
disassembly direction

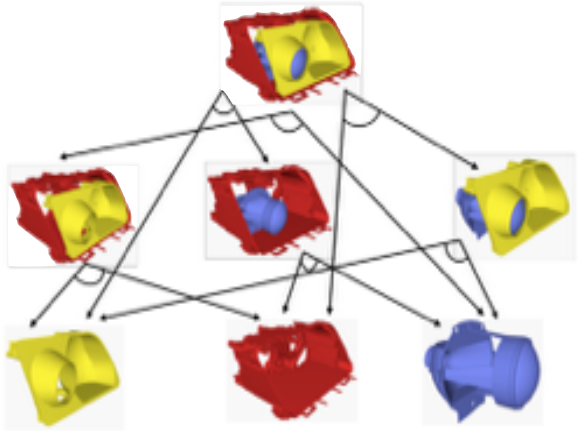
Steps of the Assembly Sequence Planning

Geometric Feasibility Tests

Recursive Graph-Cuts

Evaluation of Subassemblies and Sequences





Steps of the Assembly Sequence Planning

Geometric Feasibility Tests

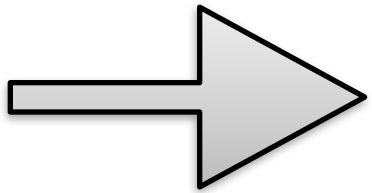
↓

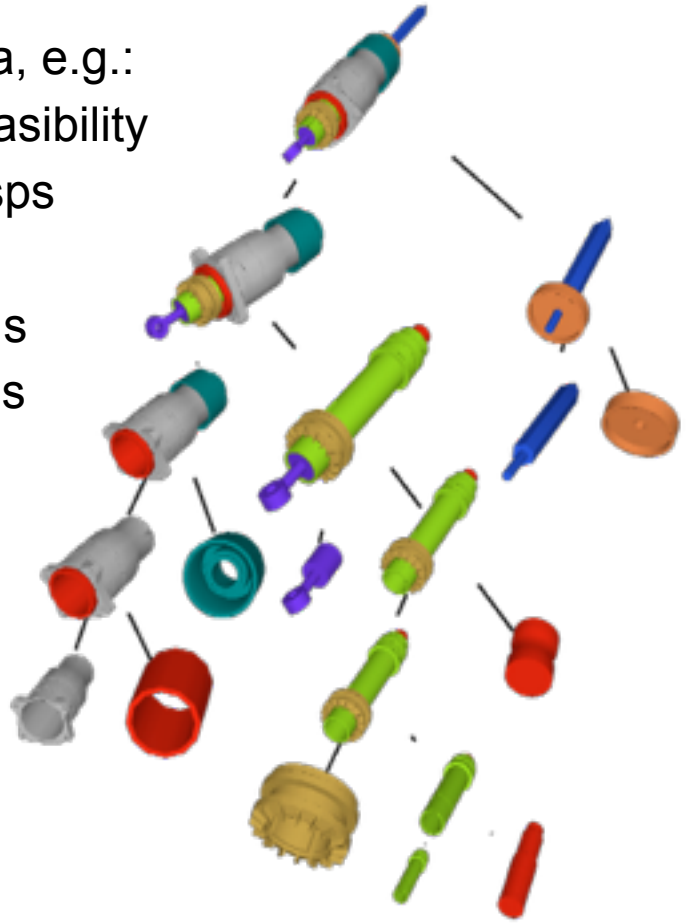
Recursive Graph-Cuts

↓

Evaluation of Subassemblies and Sequences

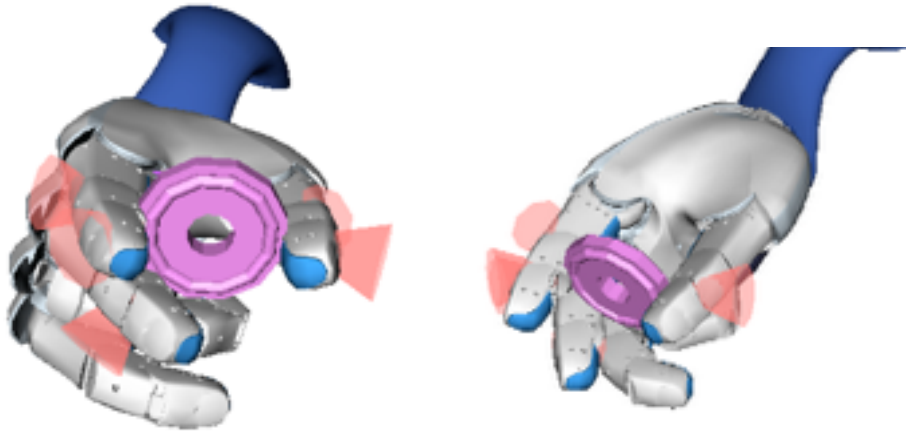
- Evaluation Criteria, e.g.:
  - Geometric feasibility
  - Feasible grasps
  - Stability
  - Available skills
  - Reorientations

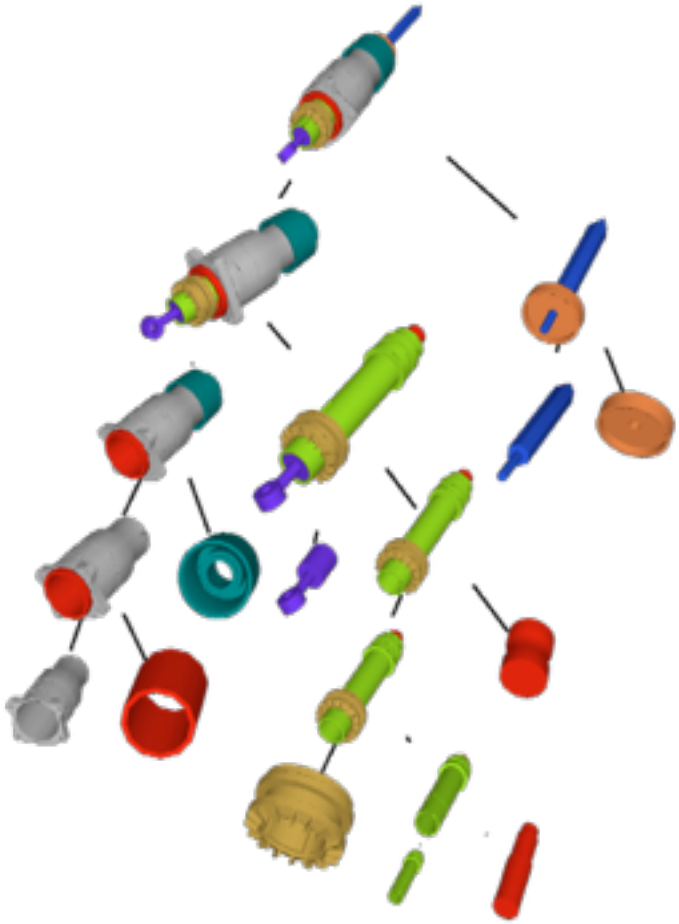




Assembly Sequence Planning; What is done and what is open?

- Current State:
  - Generating feasible sequences
  - Considering uncertainties in poses and shape
  - Considering capabilities of grippers

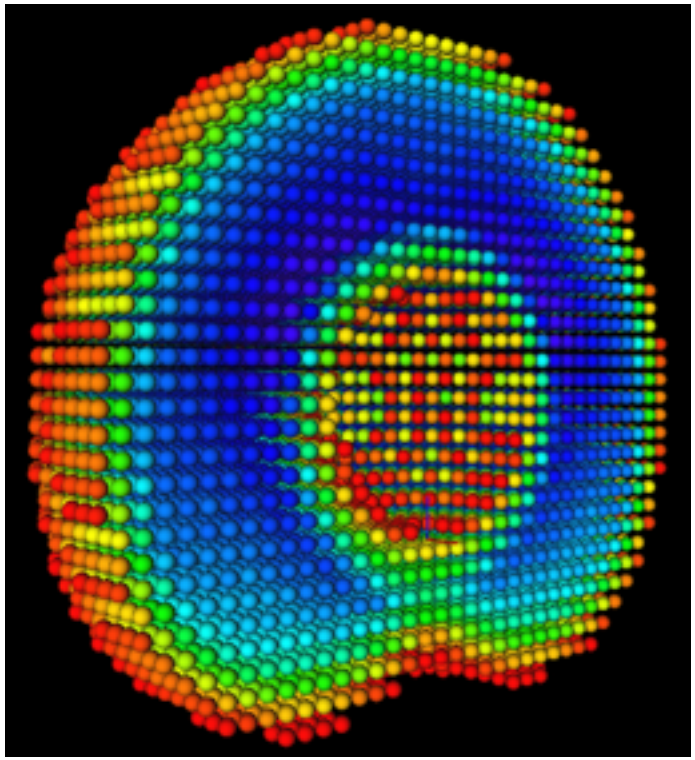




Assembly Sequence Planning; What is done and what is open?

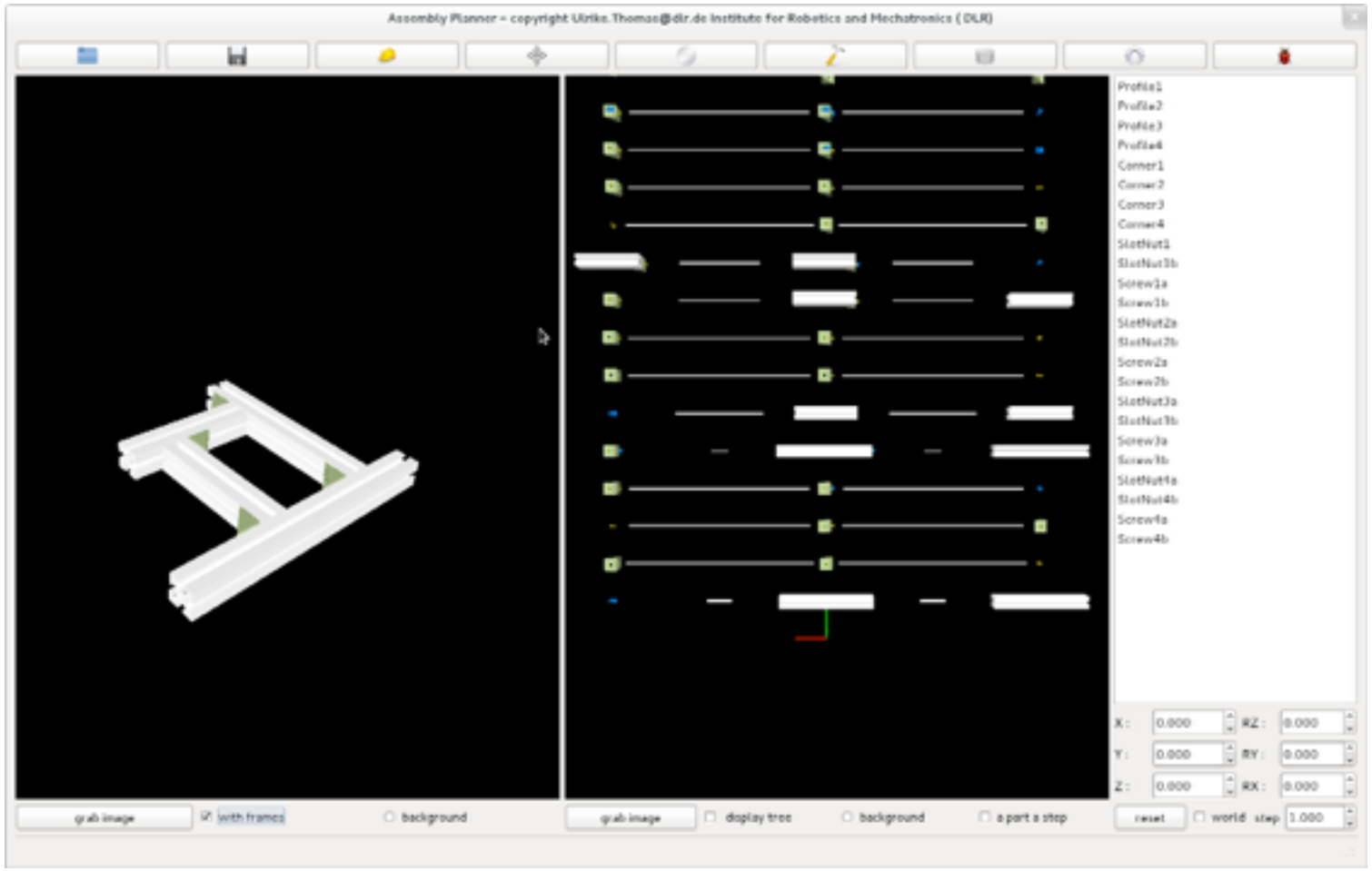
- Current State:
  - Generating feasible sequences
  - Considering uncertainties in poses and shape
  - Considering capabilities of grippers

- Some open issues:
  - Capabilities regarding robots
  - Application of fixtures
  - Optimization regarding time, energy

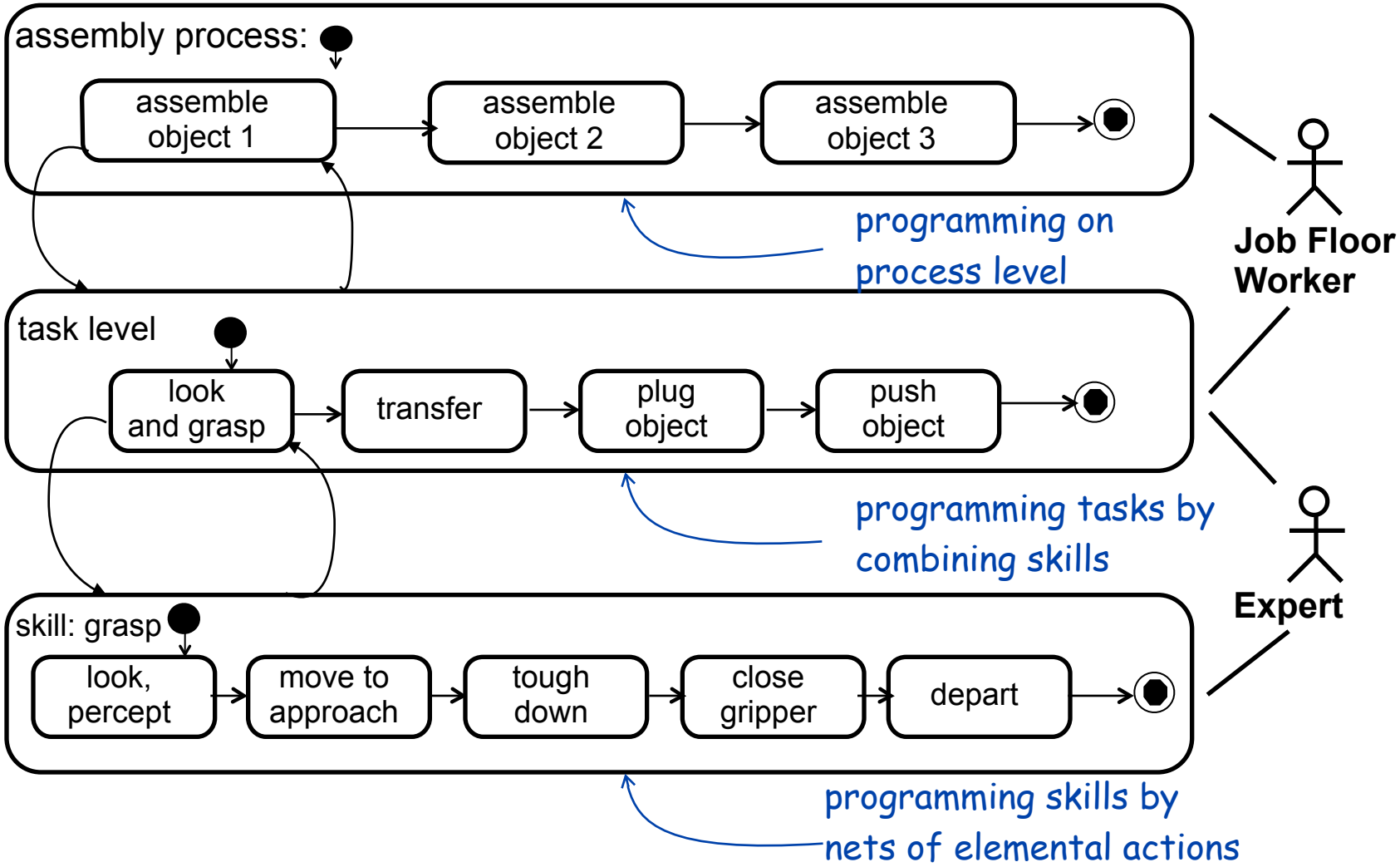


Capability map for KUKA LBR III

Assembly Sequence Planner
… close to be released to robot manufactures

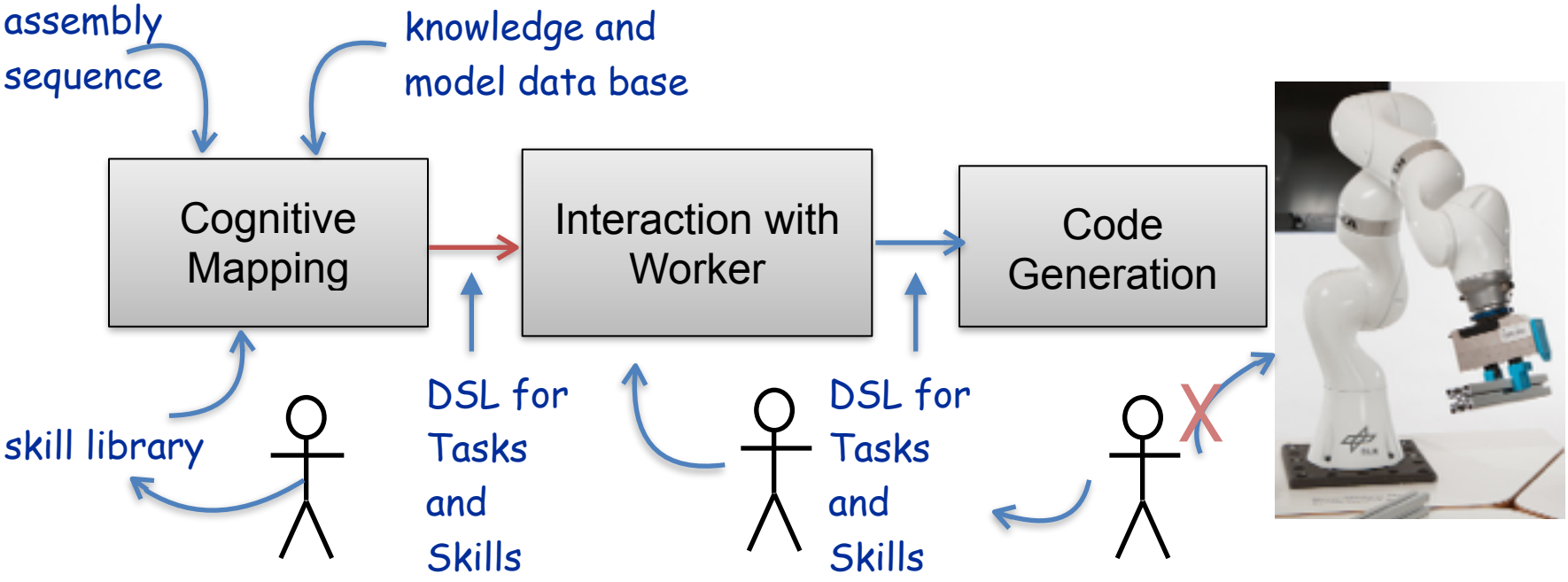


Robot Programming - Levels of Abstractions

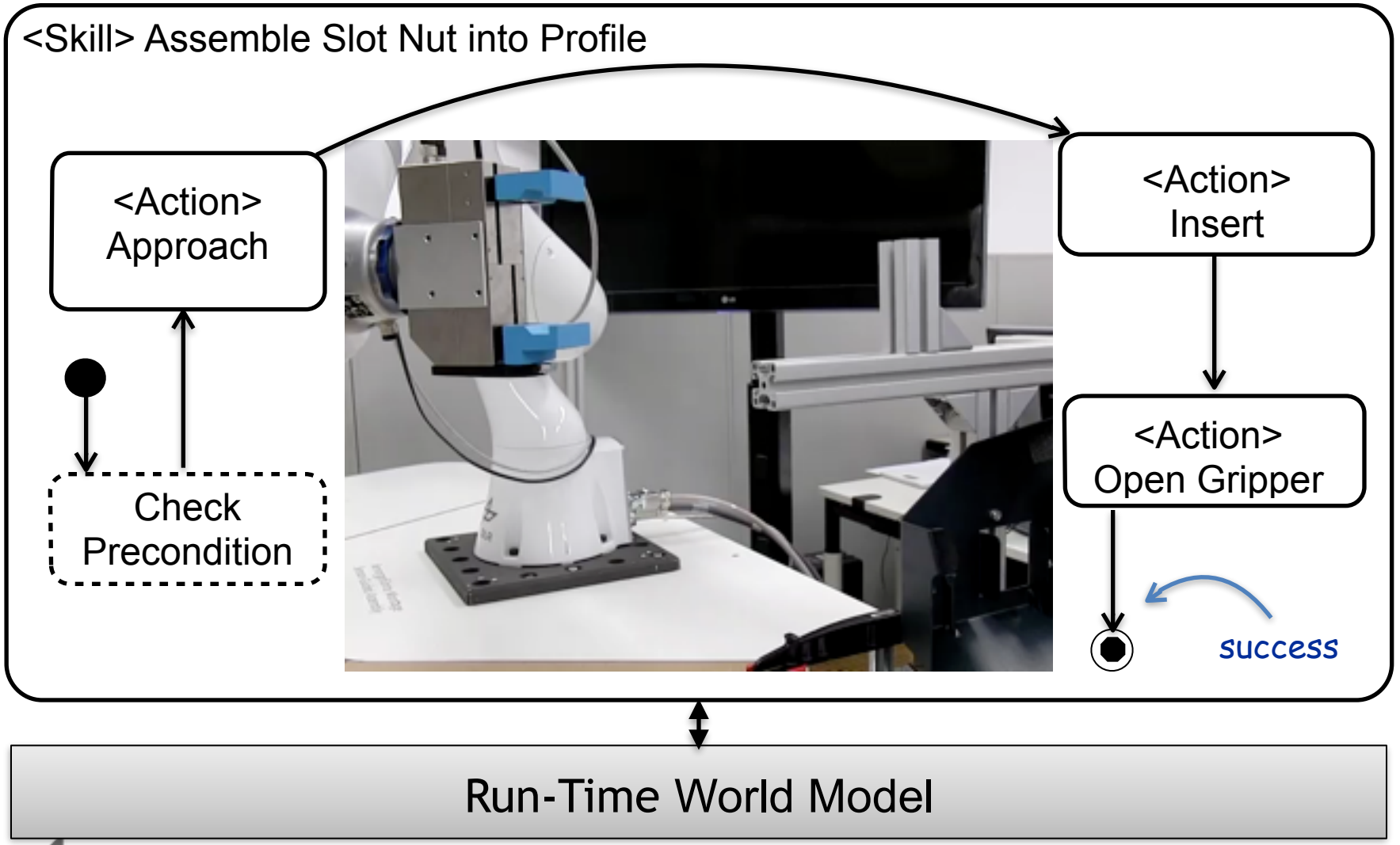


Mapping Assembly Sequences to Executable State-Charts of Skills

- Using rules from a knowledge data base
- Applying a skill library linked to the knowledge data base
- Generating State-Charts embedded with skills.
- A domain specific language used as interface between coding and planning

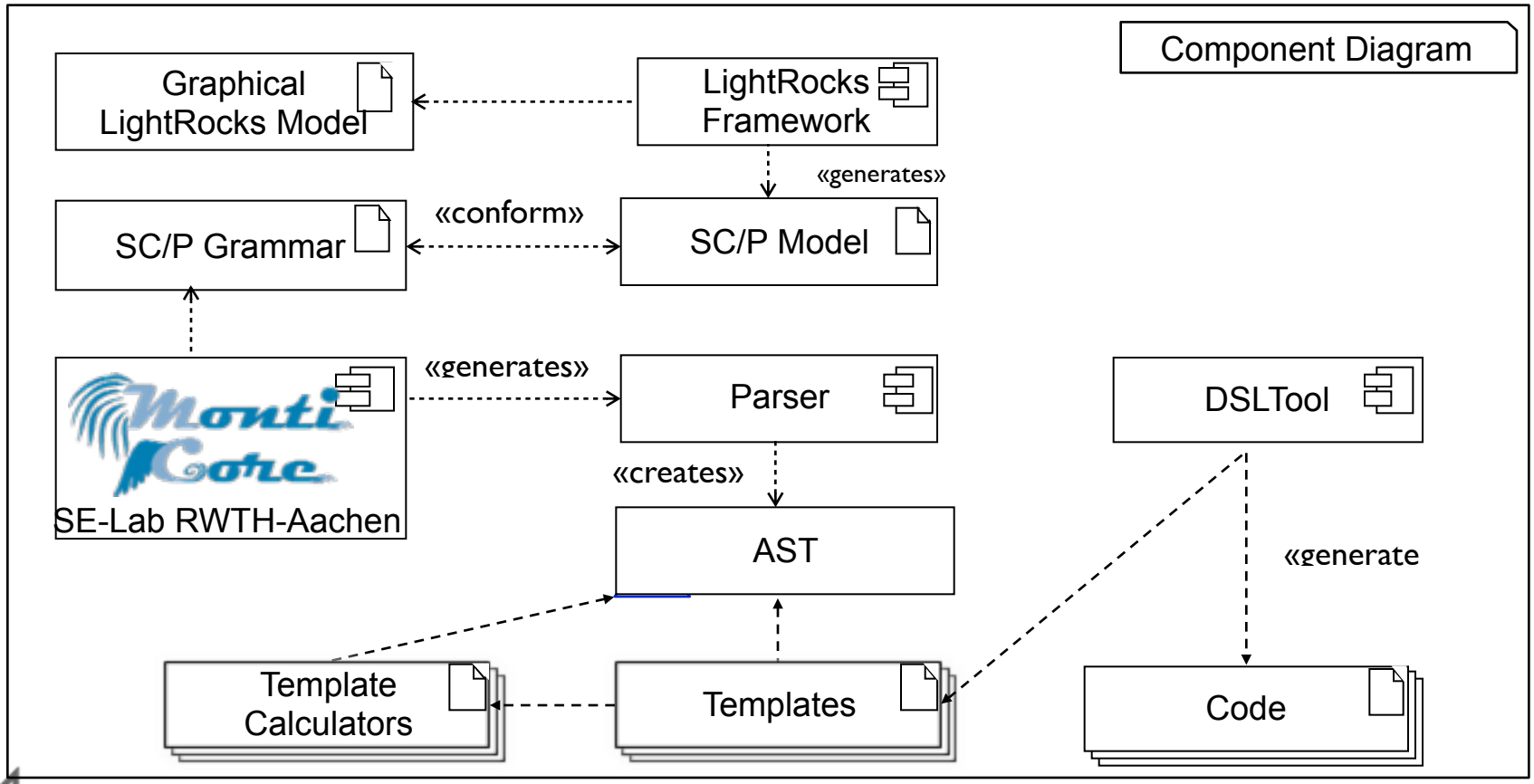


State-Chart based DSL for Skills



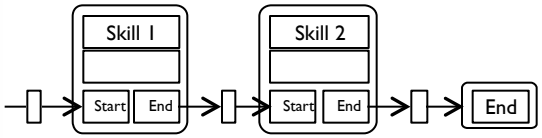
Architecture for Code Generation

- Applying SE-tools like MontiCore for code generation
  - DSL for skills called LightRocks
  - Code generation from templates



Architecture for Flexible Programming

Statechart based programming of Skills



Code Generation

Motion Primitive based API

Skill Library

Semantic Planning

Knowledge data base

Planning Modules

- Assembly Planner
- Grasp Planner
- Motion Planner

Environmental Model

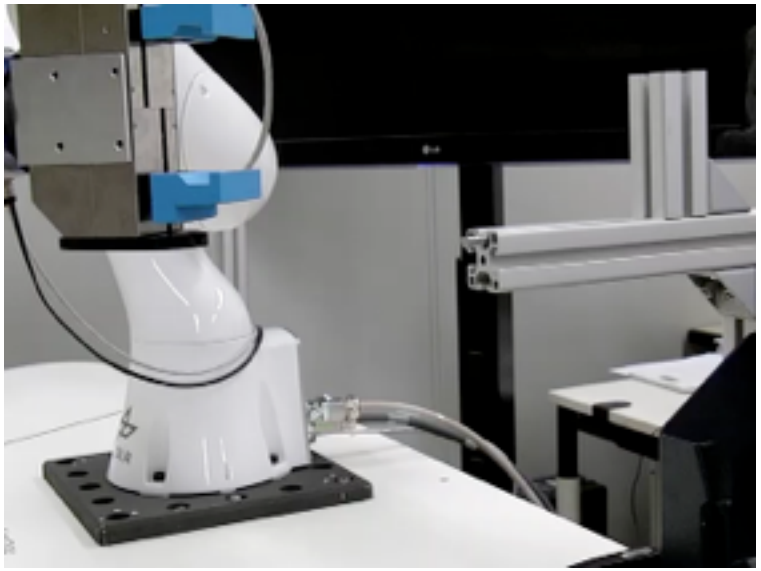
Perception Unit

Auto Calibration Module

SensorNet

Conclusion

- The framework will be shown with assemblies of ITM profiles
  - relatively structured environment
  - many variations possible
  - a certain set of skill necessary
- Generating assembly sequences
- A hierarchy of State-Charts for Robot Programming
- Using the DSL as intermediate representation between planning and execution
- Still, many open questions …
  - more autonomy
  - reactive planning
  - …



Thank you !

Visit us at Automatica and at our SMErobotics web page:

http://smerobotics.org

to keep you up to date!

The team:

at DLR:

Andreas Stemmer, Daniel Seidel, Korbinian Nottensteiner, Maximo R. Roa, Michael Kasaecker, Theodoros Stouraritis, Tim Bodenmüller, Ulrike Thomas

at KUKA;

Christian Scheuer, Marinus Danzer, Uwe Zimmermann