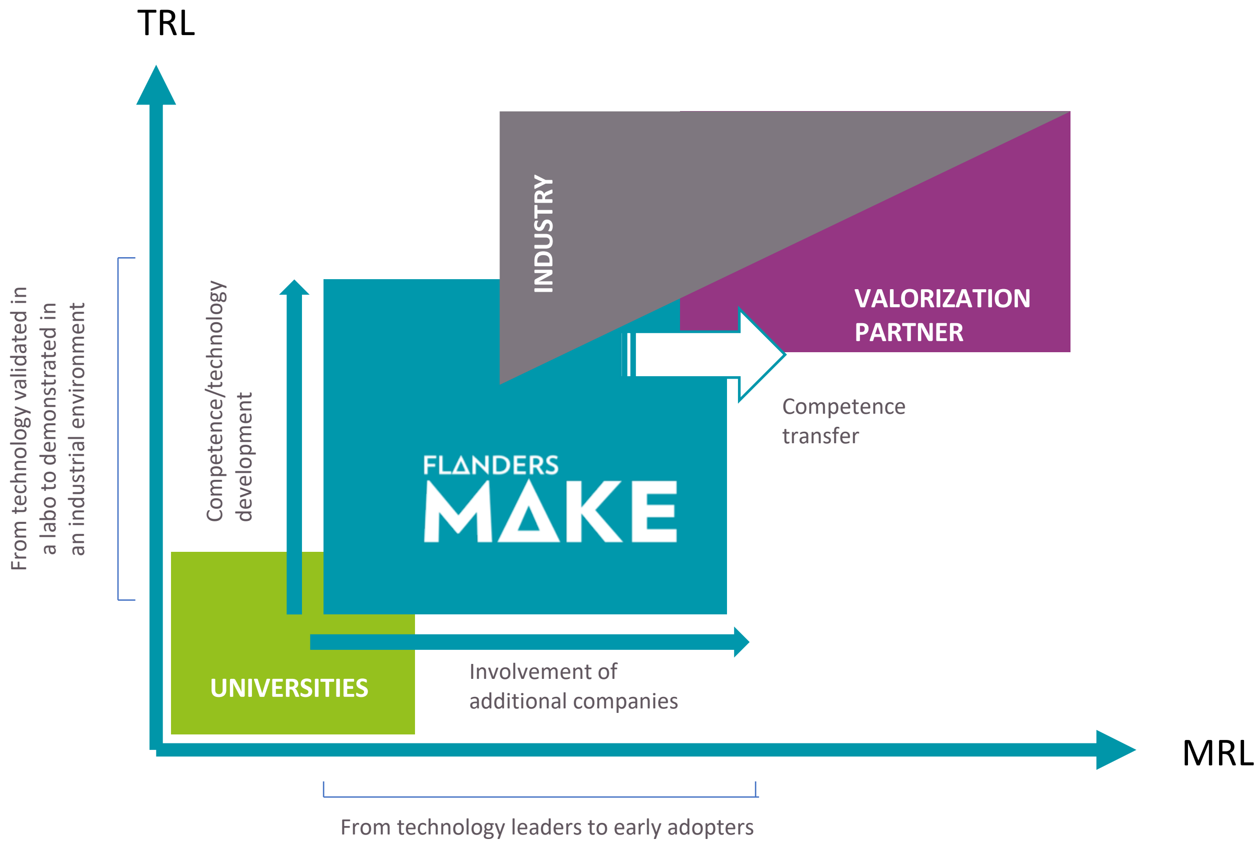
FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK

# AN INTEGRATED MODEL-BASED APPROACH FOR THE DESIGN OF ROBOTIC MANUFACTURING CELLS

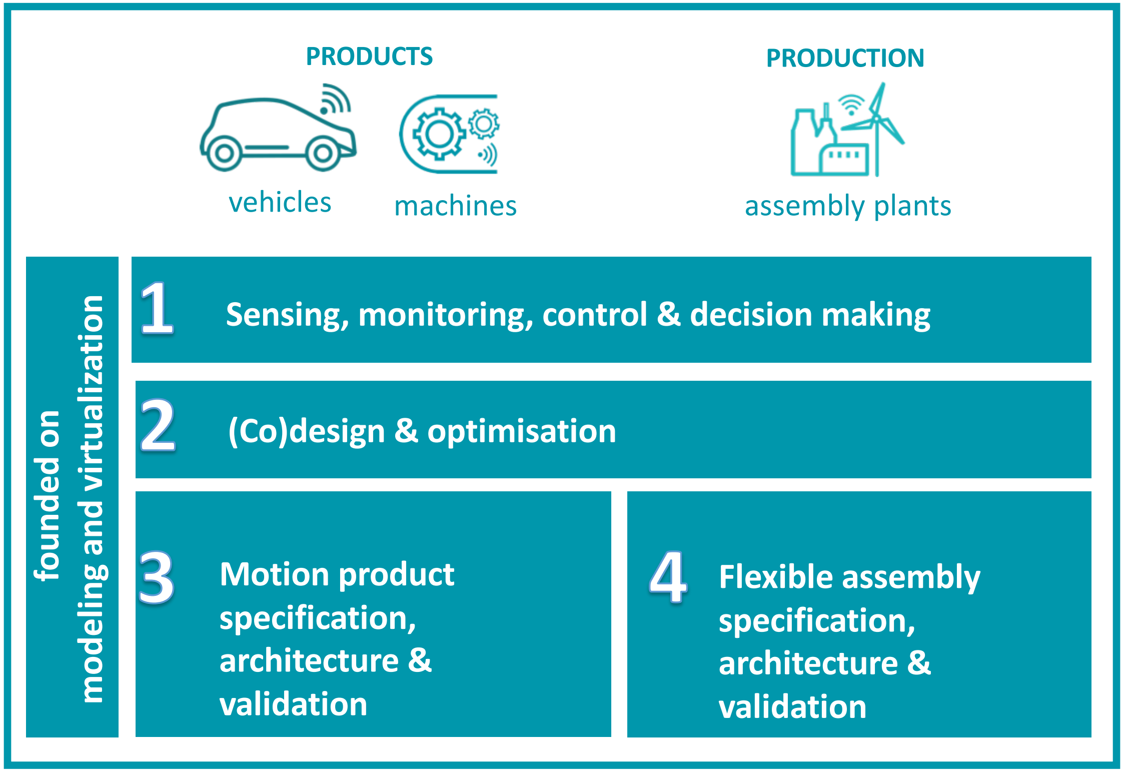
KLAAS GADEYNE

RESEARCH FELLOW, FLANDERS MAKE

# Flanders Make – Research center for the Flemish Manufacturing Industry

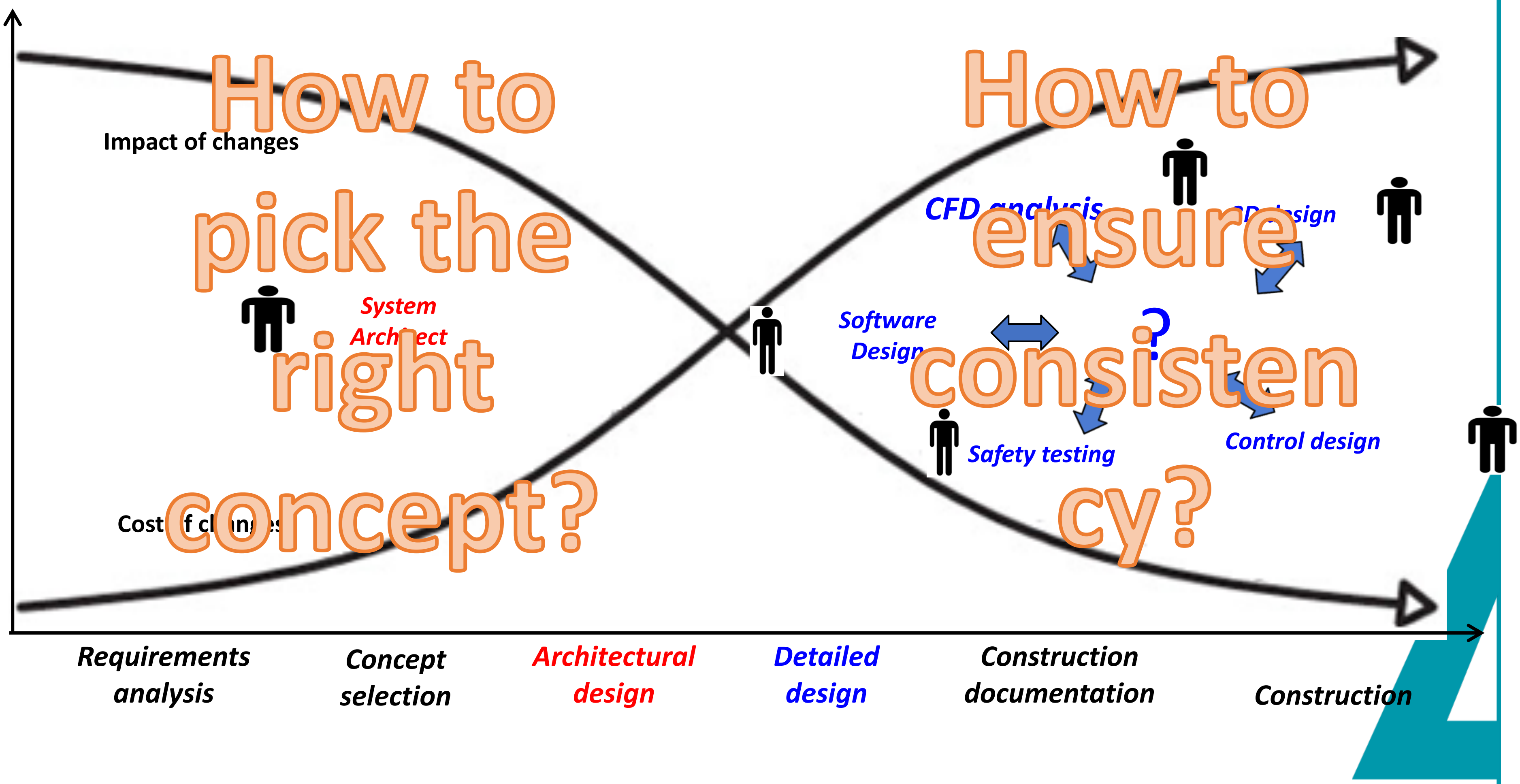


# Our organisation: focused around 4 key-competences in 2 fields of application

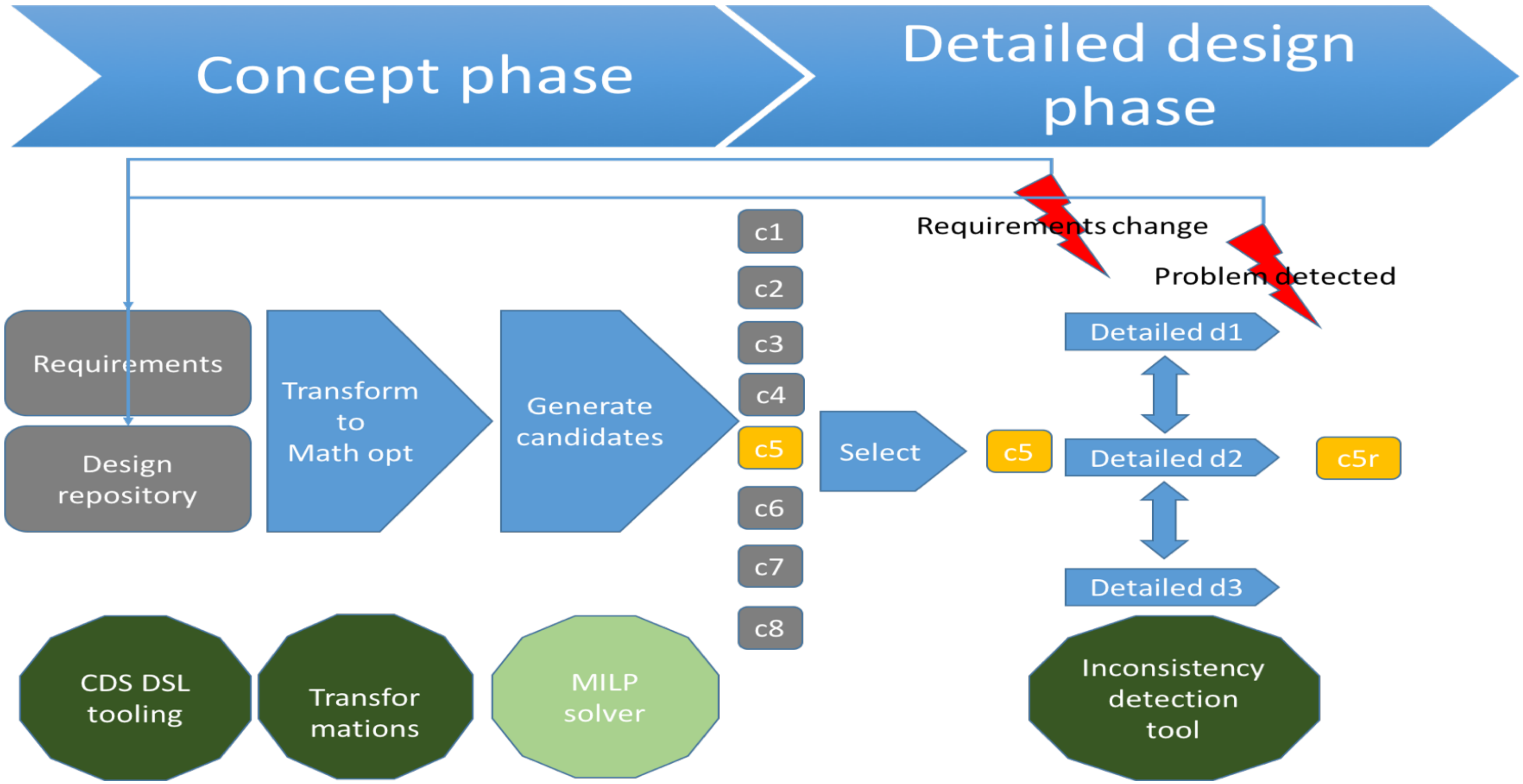


# Our partners: large companies and SME's

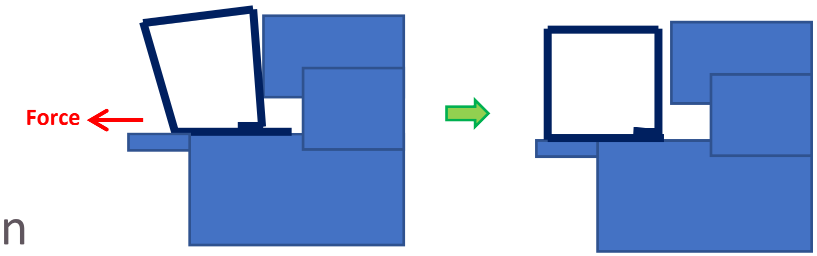
# Challenges during architectural and detailed design

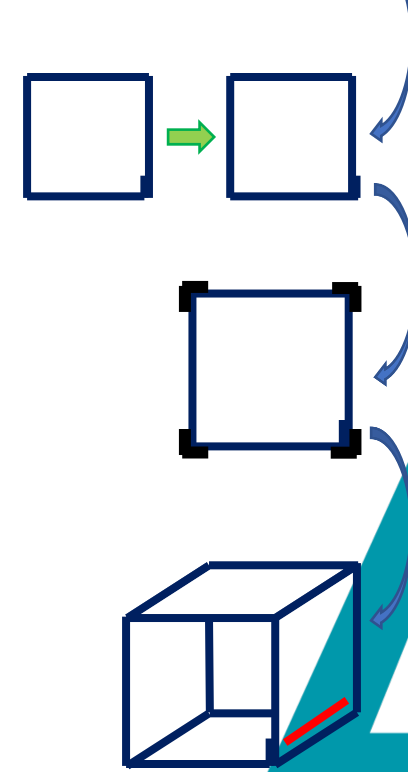


# A computer supported solution to improve the design process of complex systems



# Case study: The design of a robotic assembly/manufacturing cell





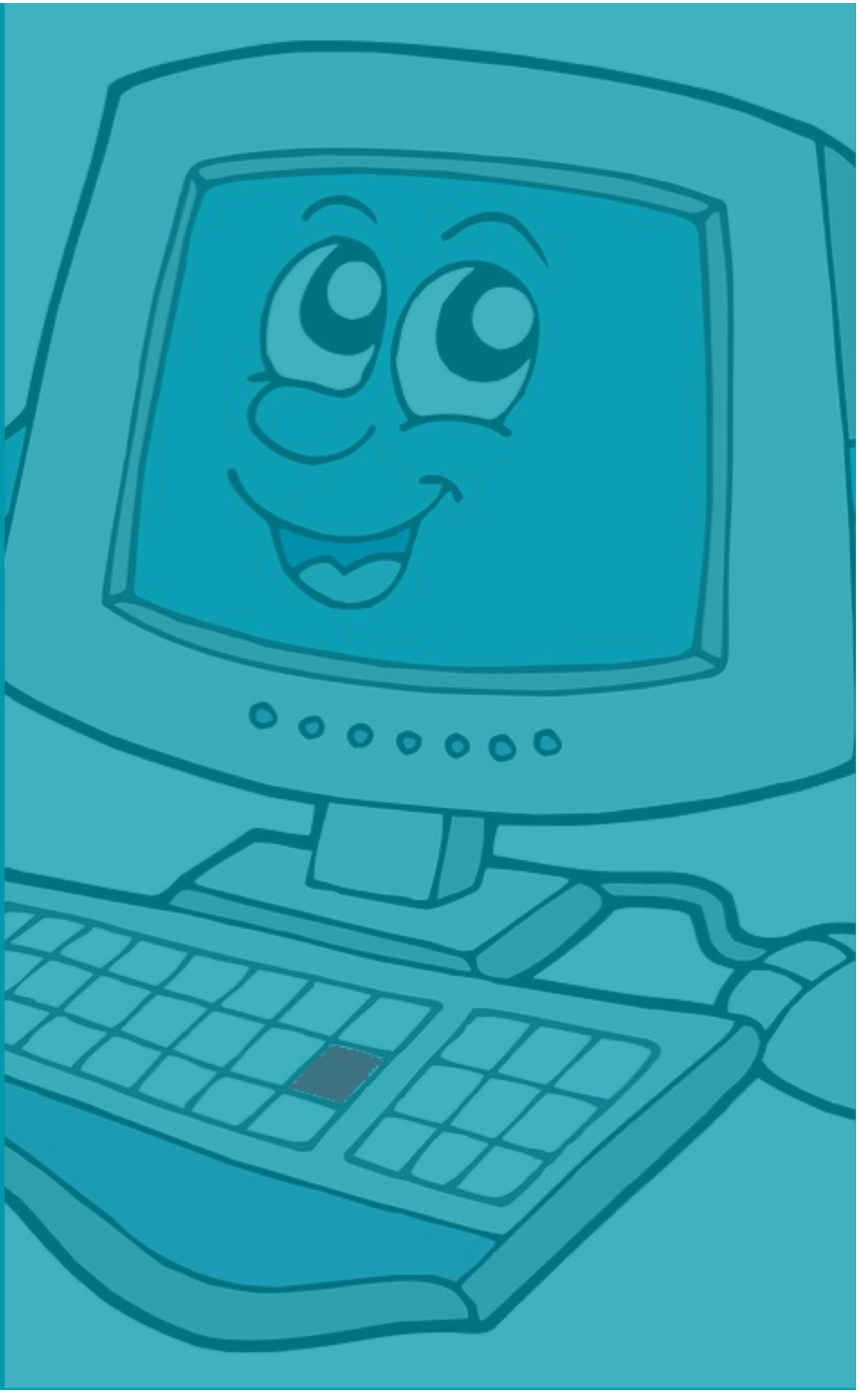
- Product to be manufactured:
  casing for fire protective valve in ventilation system

- Process steps
  - Extract
  - Correct folding
  - Join corners
  - Join seam
  - Transport between each step

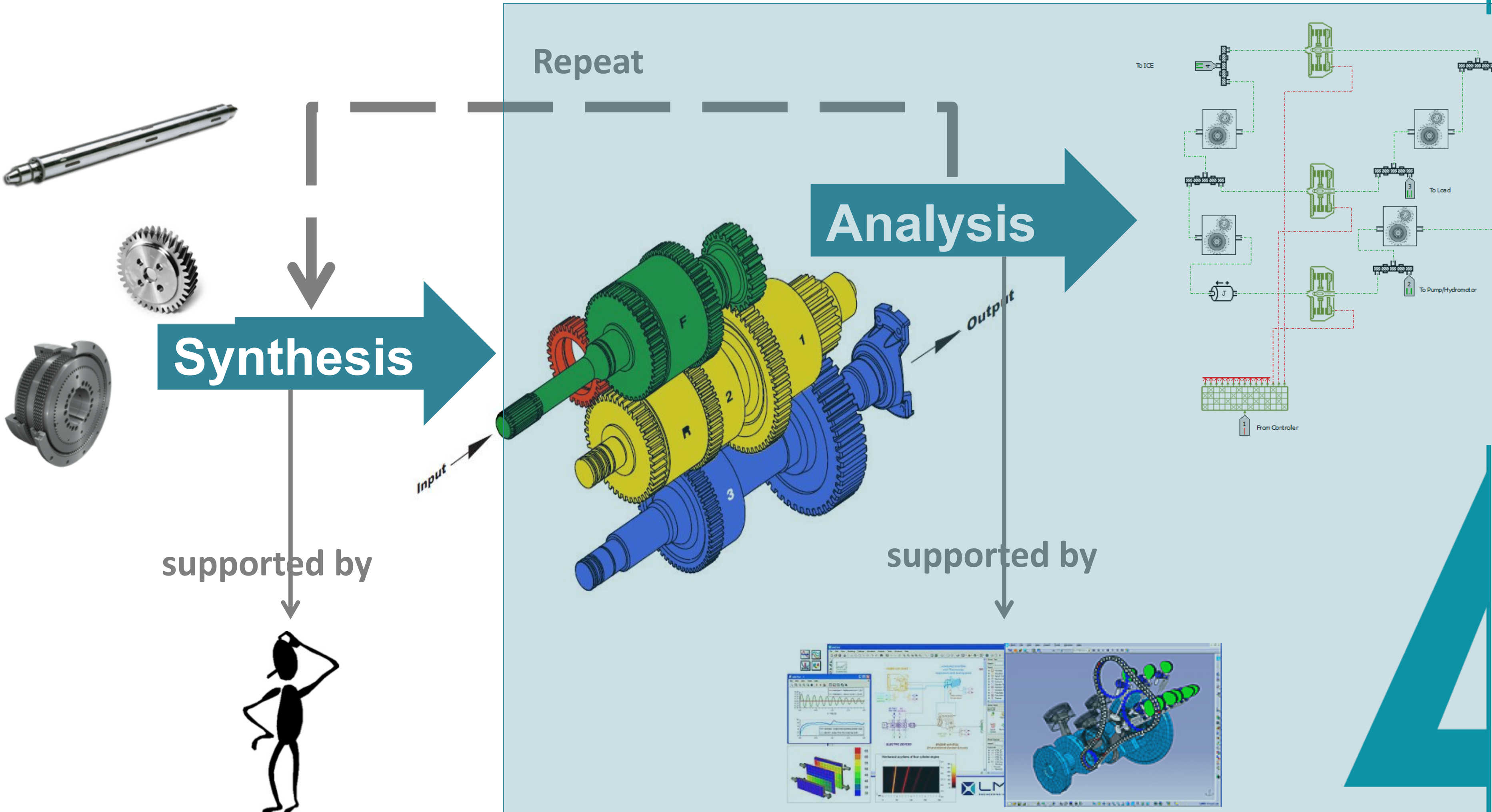
- Throughput > 40 cases/hr

- **Objective: Design a RAC that can implement this process as cheap as possible**

# COMPUTATIONAL DESIGN SYNTHESIS TO SUPPORT THE CONCEPT SELECTION PHASE OF RAC

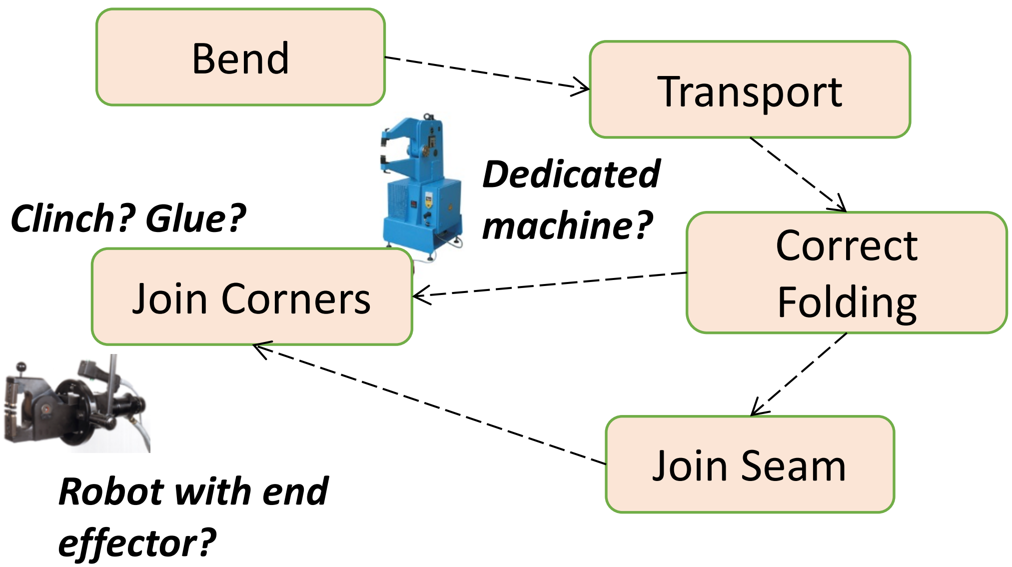


# Model-based computational design what?




"Classical" Model-Based Design

# Why is it hard to synthesize the right concept?

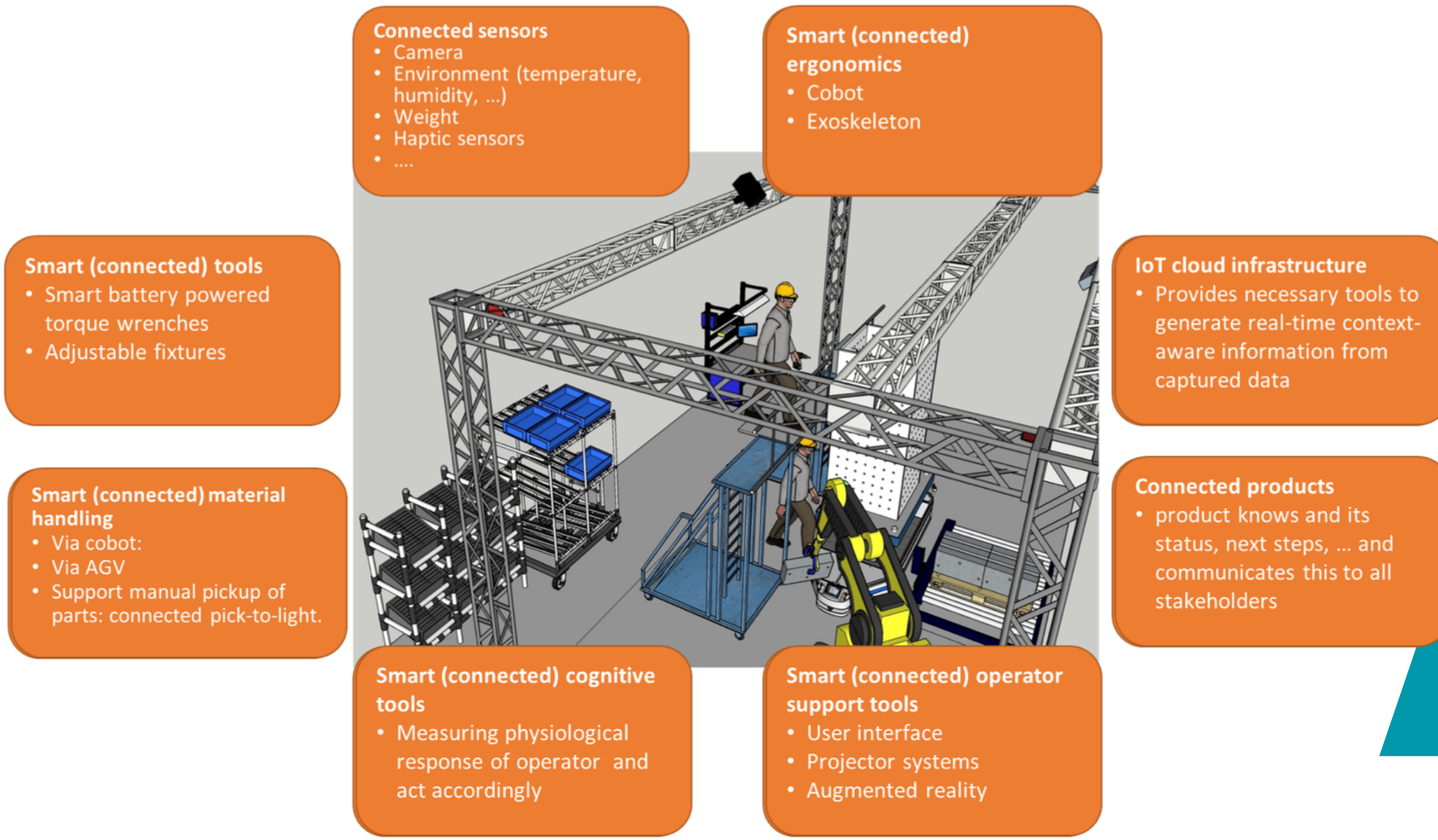



- Say there are 6 manufacturing steps, and each can be implemented by one of 3 alternative working principles
- Say an average of 3 resources implement each working principle

→ In worst case, over $3.4 \times 10^{30}$ different alternative cell designs!

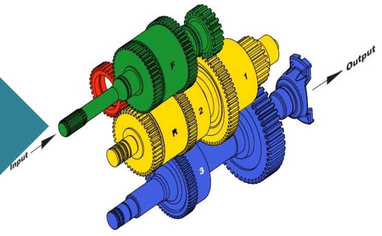
# In the future, it will only become harder

- Companies are confronted with **a demand for highly-customized products (lot size 1)** => Manufacturing cells need to be flexible as well
- In turn, this requires the use **of novel, smart assembly technological solutions**
  - Cobots, AGV’s, Smart operator support tools (e.g. projection systems), Intelligent racks



# The solution?

Synthesis

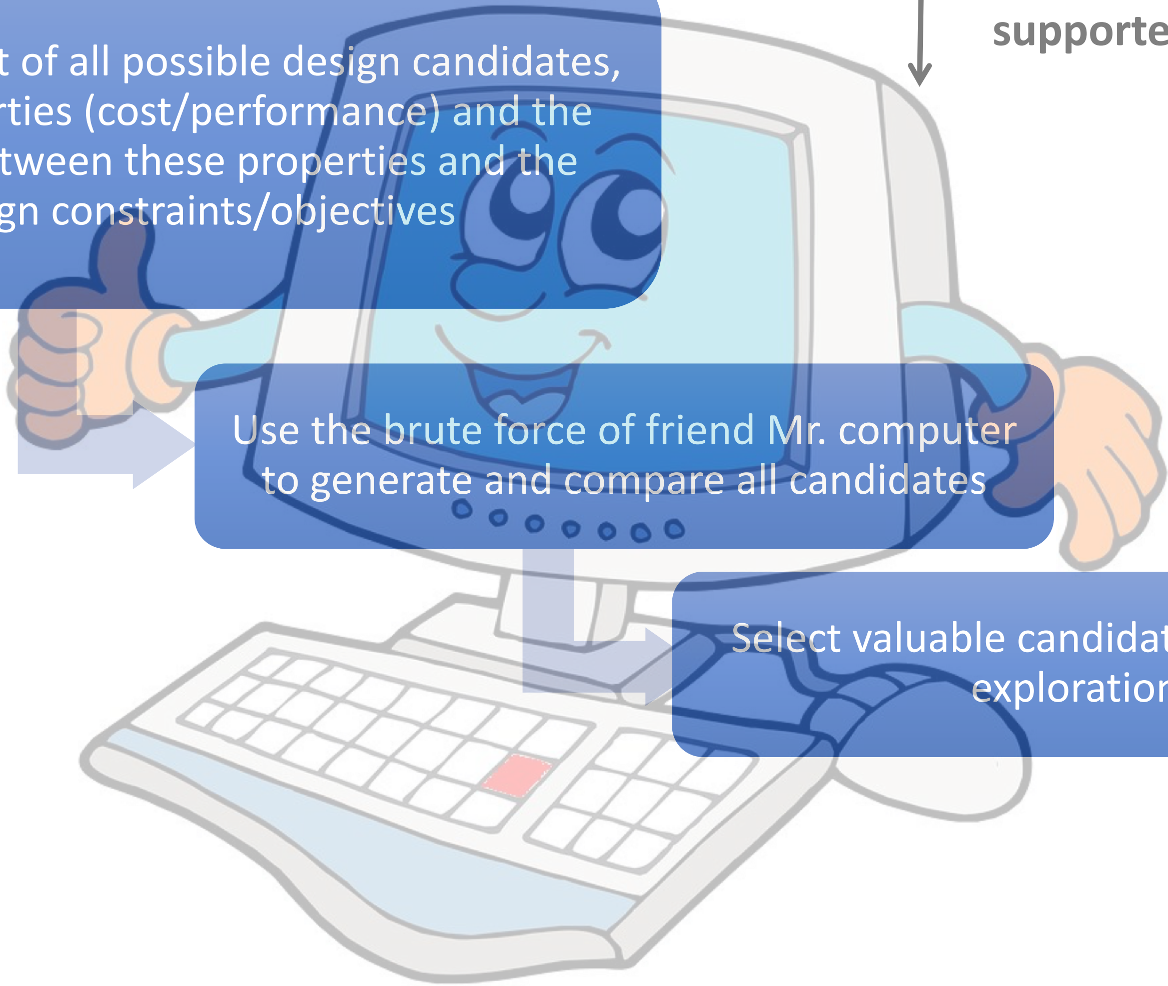


supported by

*Model* the set of all possible design candidates, their properties (cost/performance) and the relation between these properties and the design constraints/objectives

Use the brute force of friend Mr. computer to generate and compare all candidates

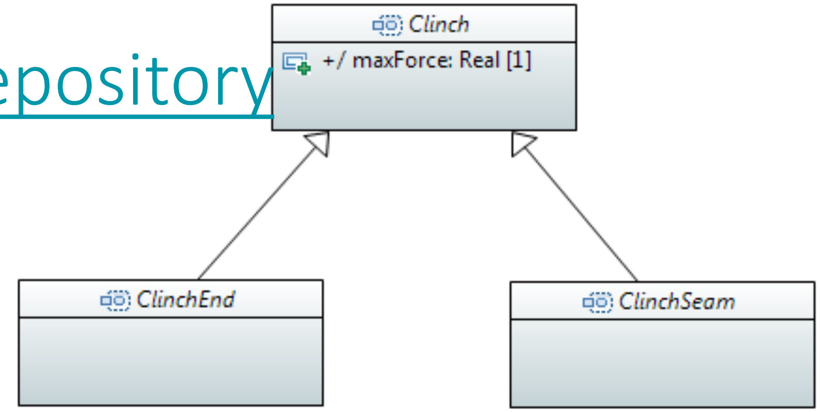
Select valuable candidates (for further exploration)



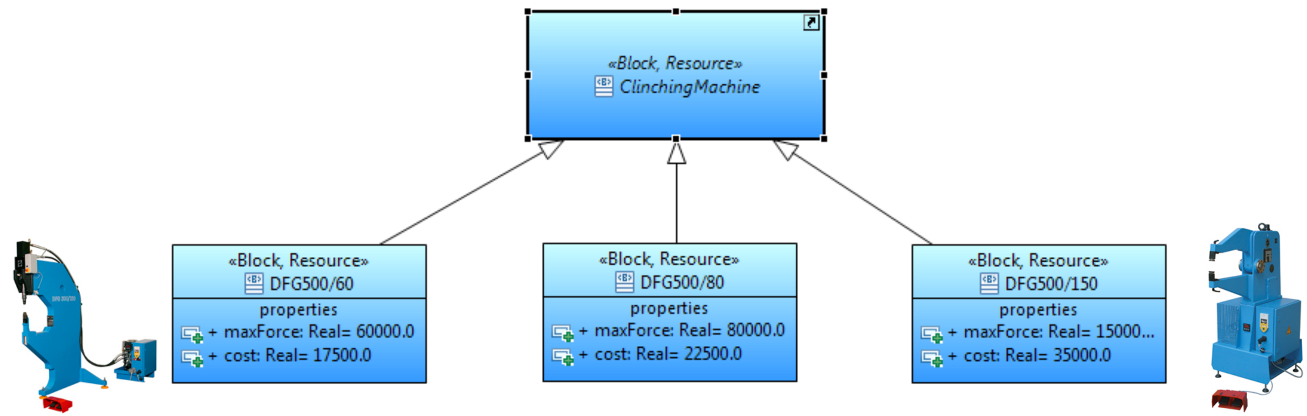
**=> Computational Design Synthesis**

# Step 1a: Modeling the design repository

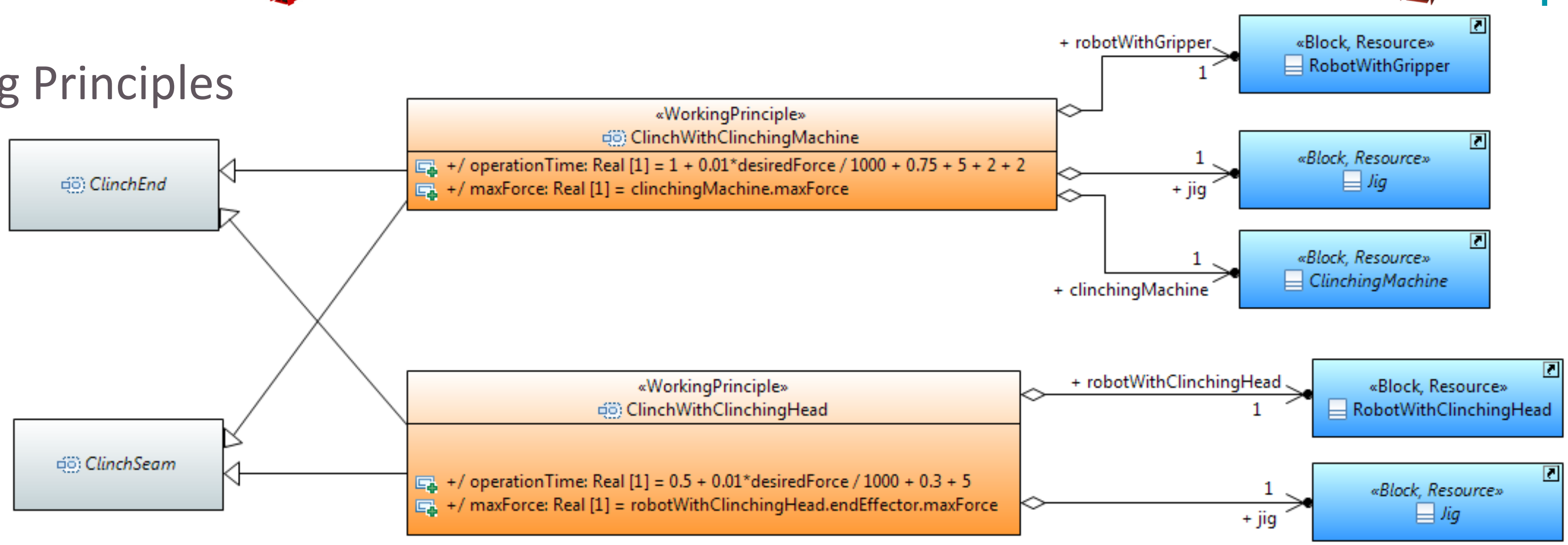
- Activities



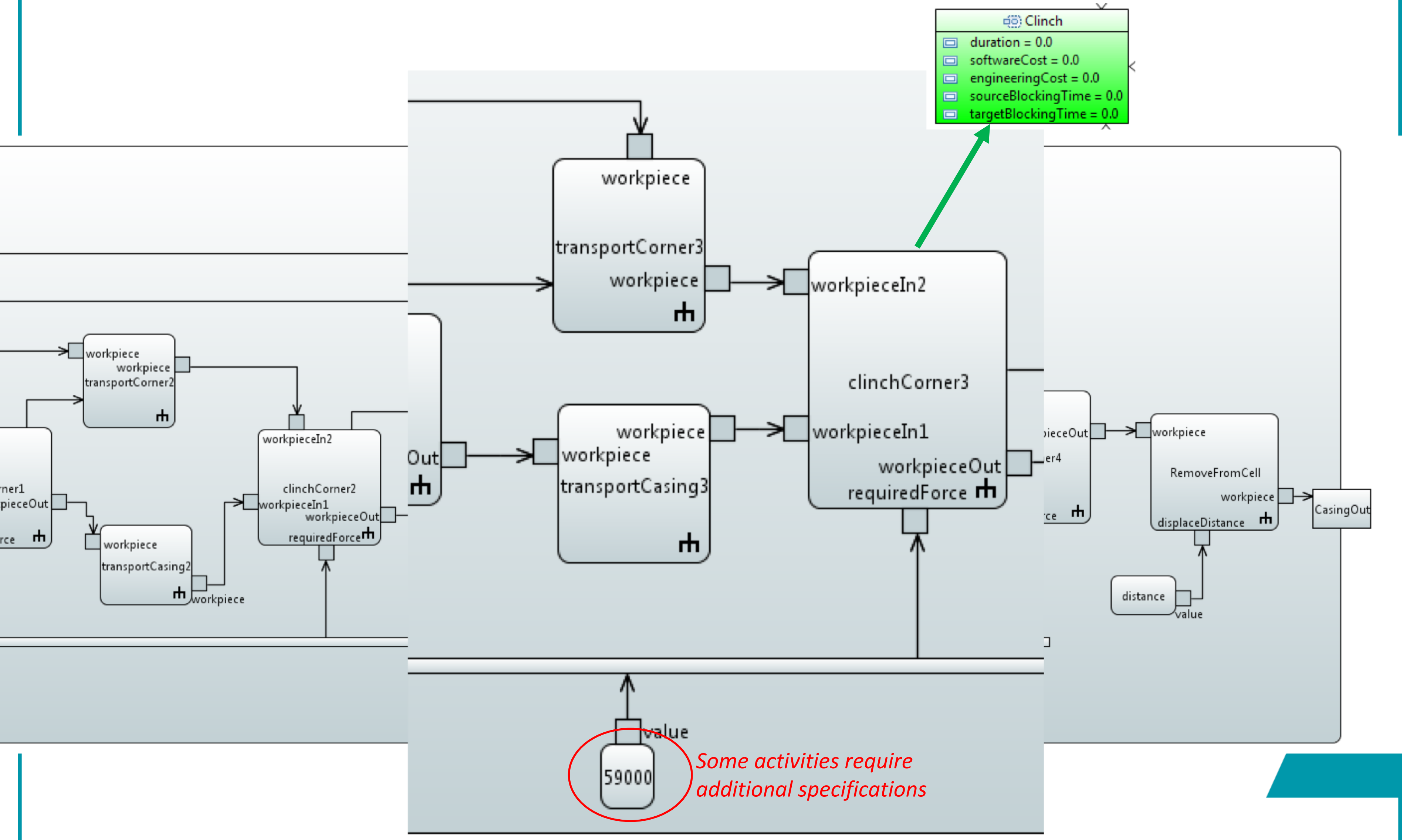

- Resources



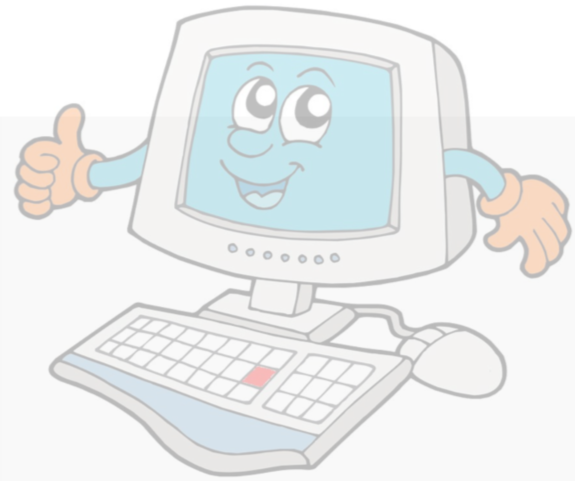

- Working Principles



# Step 1b: Modeling the Design Problem description



# Step 2: Let the computer do all the hard work



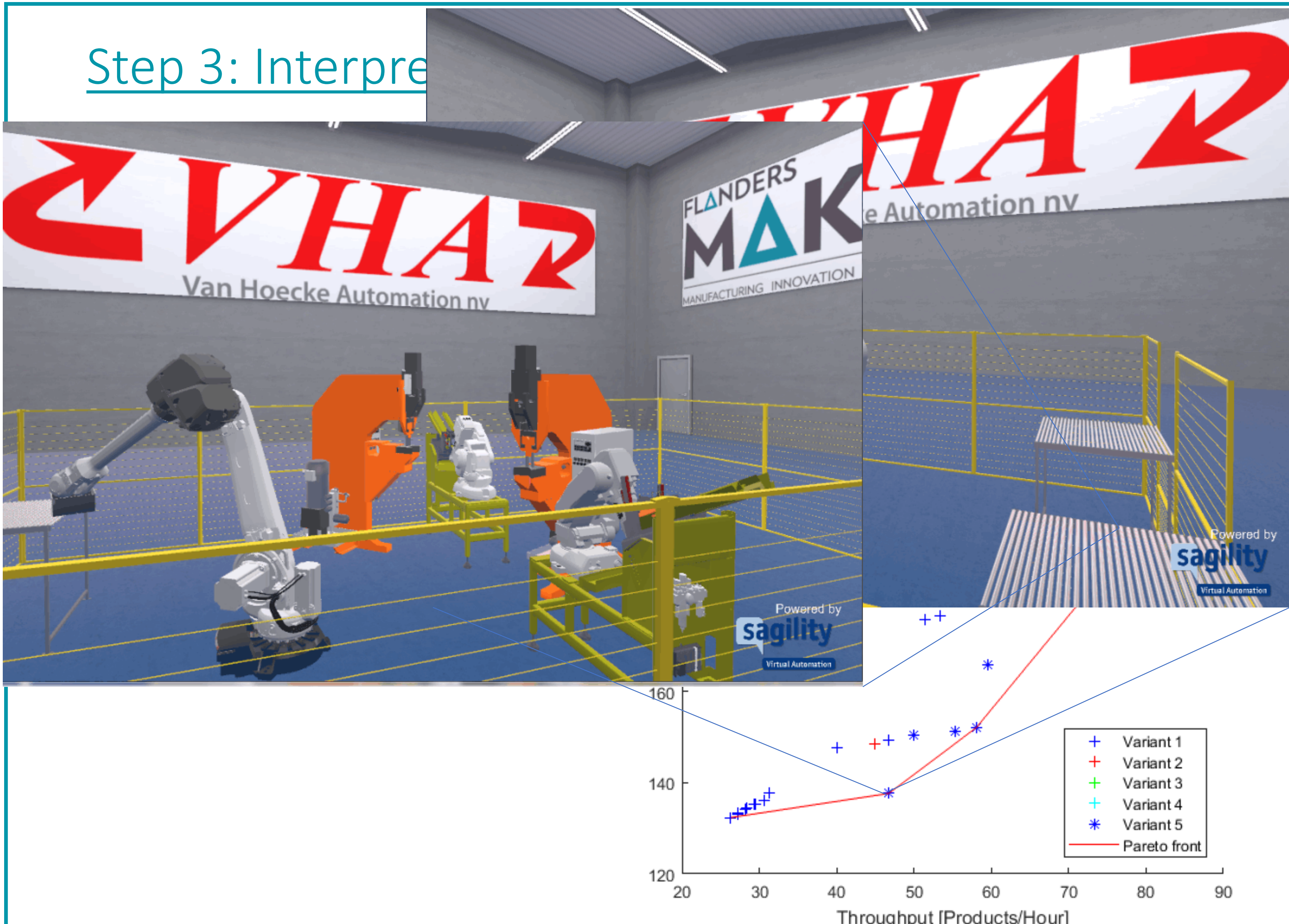
```
ATs(2).wp(1).name = 'DisplaceWithRobot';
ATs(2).wp(1).uuid = 'df08cf79-f188-3d25-a175-cd88682245f3';
ATs(2).wp(1).softwareCost = 1500.0;
ATs(2).wp(1).engineeringCost = 0;
ATs(2).wp(1).resTypes = {'RobotWithGripper'};
ATs(2).wp(1).isShared = [true];
ATs(2).wp(1).canBePerformed = {@(res,req) (req('workpiece.mass') <= reshape([res.robot_maxPayload],size(res)))};
ATs(2).wp(1).durationBase = @(req) (10.0);
ATs(2).wp(1).durationMod = {@(res,req) zeros(size(res))};
ATs(2).wp(1).sourceBlockingTimeBase = @(req) (5);
ATs(2).wp(1).sourceBlockingTimeMod = {@(res,req) zeros(size(res))};
ATs(2).wp(1).targetBlockingTimeBase = @(req) (2);
ATs(2).wp(1).targetBlockingTimeMod = {@(res,req) zeros(size(res))};

ATs(2).wp(2).name = 'DisplaceWithLinearActuator';
ATs(2).wp(2).uuid = '96c031e8-3f2d-3739-ae42-2ca83a914fda';
ATs(2).wp(2).softwareCost = 1000.0;
ATs(2).wp(2).engineeringCost = 12.0;
ATs(2).wp(2).resTypes = {'LinearActuator'};
ATs(2).wp(2).isShared = [false];
ATs(2).wp(2).canBePerformed = {@(res,req) true(size(res))};
ATs(2).wp(2).durationBase = @(req) (7.0);
ATs(2).wp(2).durationMod = {@(res,req) zeros(size(res))};
ATs(2).wp(2).sourceBlockingTimeBase = @(req) (2);
ATs(2).wp(2).sourceBlockingTimeMod = {@(res,req) zeros(size(res))};
ATs(2).wp(2).targetBlockingTimeBase = @(req) (2);
ATs(2).wp(2).targetBlockingTimeMod = {@(res,req) zeros(size(res))};

ATs(2).wp(3).name = 'DisplaceWithConveyorBelt';
ATs(2).wp(3).uuid = '4112e33d-9f2c-332e-83ad-2ad43c042c99';
ATs(2).wp(3).softwareCost = 2500.0;
ATs(2).wp(3).engineeringCost = 1500.0;
ATs(2).wp(3).resTypes = {'ConveyorBelt'};
ATs(2).wp(3).isShared = [false];
ATs(2).wp(3).canBePerformed = {@(res,req) true(size(res))};
ATs(2).wp(3).durationBase = @(req) (5.0);
ATs(2).wp(3).durationMod = {@(res,req) zeros(size(res))};
ATs(2).wp(3).sourceBlockingTimeBase = @(req) (1);
ATs(2).wp(3).sourceBlockingTimeMod = {@(res,req) zeros(size(res))};
ATs(2).wp(3).targetBlockingTimeBase = @(req) (1);
ATs(2).wp(3).targetBlockingTimeMod = {@(res,req) zeros(size(res))};
```

> 3000 lines generated MILP code

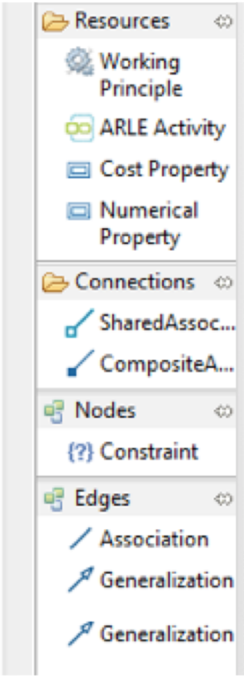
# Step 3: Interpre



# What do you need to bring this into practice?

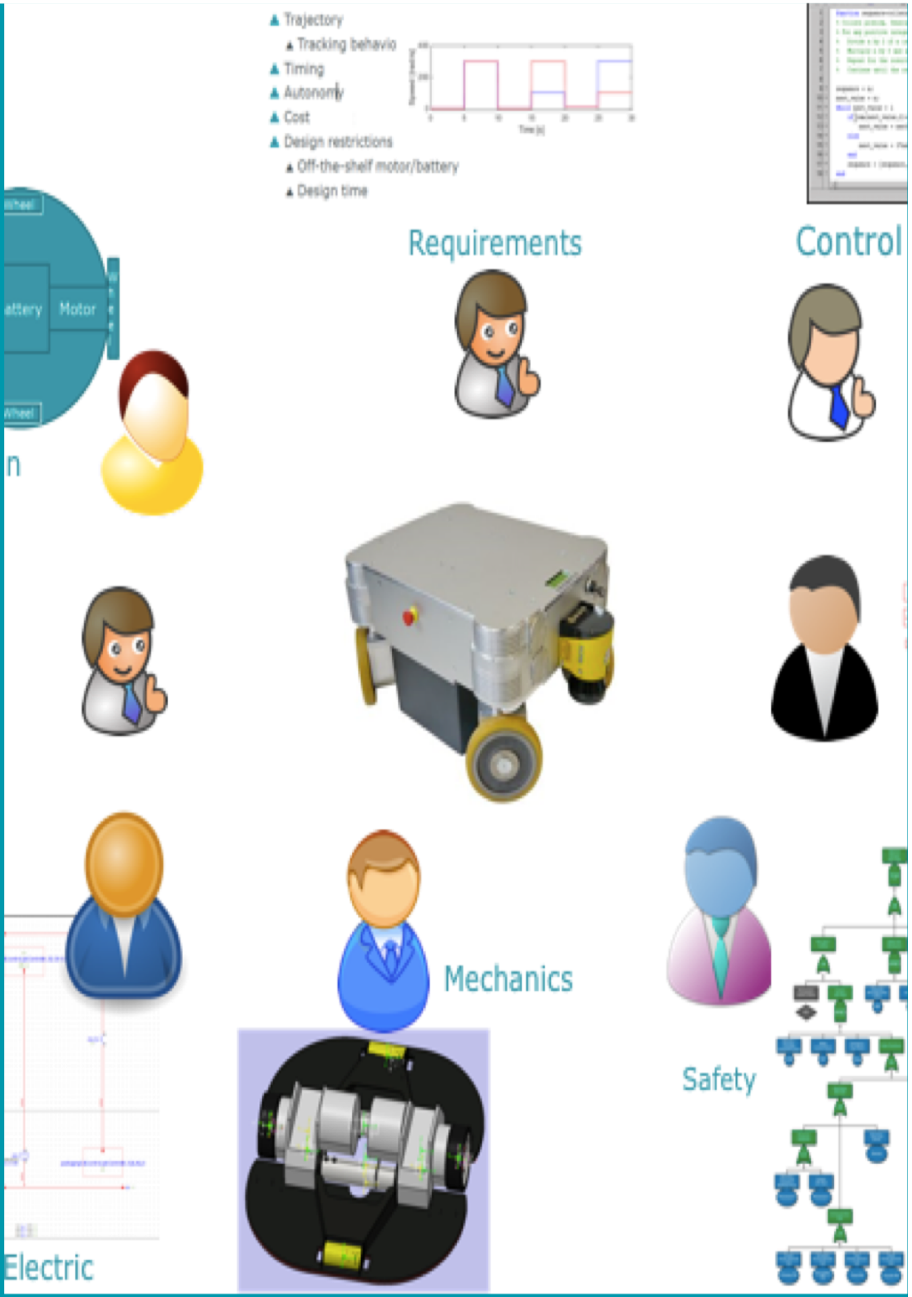
- A clear understanding of how your particular design problems can be mapped onto a numerical constrained optimisation program

- User friendly tooling that shields the designers from the mathematical complexity and allows to
  - Create and maintain a reusable design repository
  - Create and update design problem descriptions
  - Evaluate and interpret the mathematical solutions

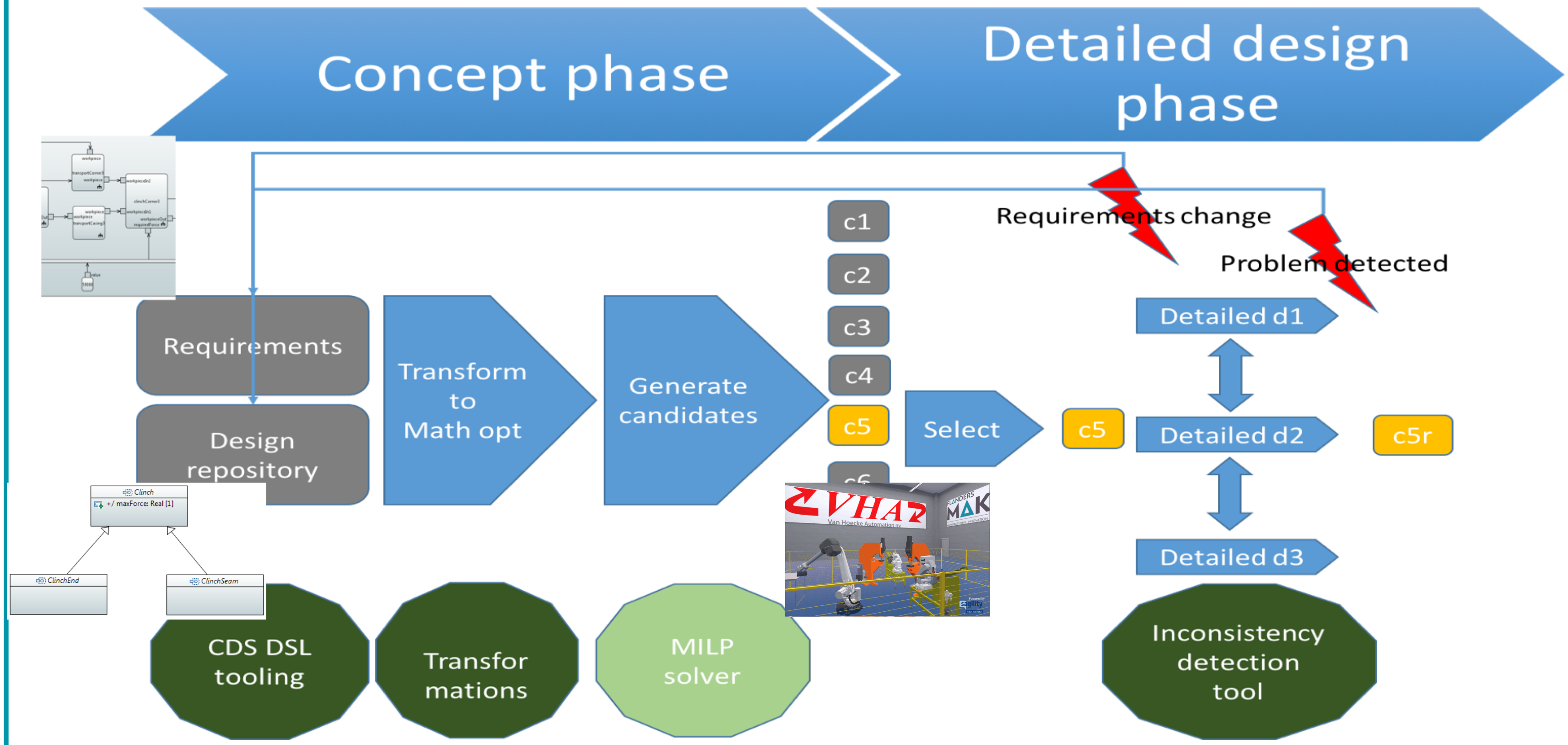



- The necessary transformations from the user friendly tooling to the mathematical rocket science and vice versa

# AUTOMATIC INCONSISTENCY CHECKING DURING THE DETAILED DESIGN PHASE



# A computer supported solution to improve the design process of complex systems



# Why is it difficult to ensure the consistency during detailed design?

- Using the generated conceptual solution as a starting point, engineers now have to refine it and analyse other aspects
  - Risk analysis/ safety countermeasures / safety performance analysis for the counter-measures (for instance safety functions)
  - Mechanical and electrical parts have to be further detailed
  - Control software
- To shorten the design cycle => Maximalize concurrent design
- However, as they work on the same design, they have **dependencies** onto each other
  - Software engineer needs additional sensor for accurately estimating position of the workpiece
    - Electrical engineer has to update electrical schemes
    - Mechanical engineer has to update CAD model
  - Software engineer has to communicate his design decisions to the appropriate engineers
- In practice: Communication often goes wrong and the resulting errors are often only detected very late => **Failing systems, additional design costs and delays**

# To make things worse

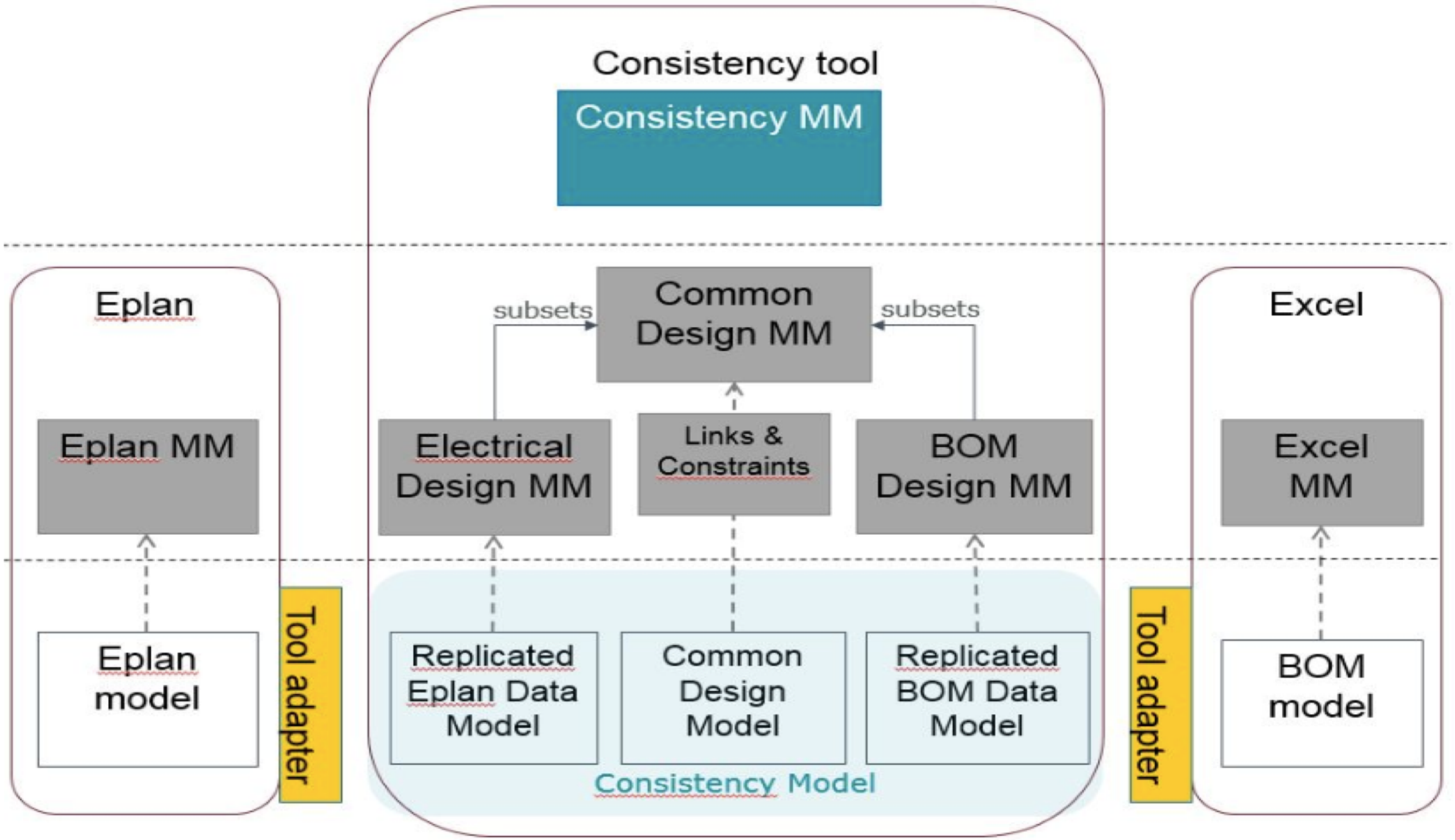
- Imagine that the customer comes back during the detailed design phase with the question if it is still feasible to have a throughput of 50 pcs/hour instead of 40 pcs/hour

Table 1. Original output of the mathematical optimization (left) and output after increasing the required throughput (right)

| Original (left) | After increasing throughput (right) |
| --- | --- |
| **Used Resources:** | **Used Resources:** |
| ≫ 1 × DFG500/60 | ≫ 2 × DFG500/60 |
| ≫ 1 × ABBIRB140-6/0.8 + MechanicalGripper | ≫ 2 × ConcreteLinearActuator |
| ≫ 1 × ABBIRB4600-40/2.55 + MechanicalGripper | ≫ 1 × ABBIRB140-6/0.8 + MechanicalGripper |
| | ≫ 1 × ABBIRB4600-40/2.55 + MechanicalGripper |
| **Performance:** | **Performance:** |
| ≫ Cost = 97068 [euros] | ≫ Cost = 132568 [euros] |
| ≫ Throughput = 43.9024 [parts/hour] | ≫ Throughput = 58.0645 [parts/hour] |
| ≫ BottleNeck = ABBIRB4600-40/2.55 + MechanicalGripper | ≫ BottleNeck = ABBIRB4600-40/2.55 + MechanicalGripper |

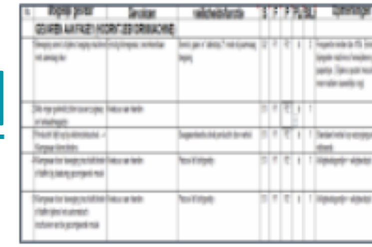
- Which part needs to be revisited?
  - Do we have still have enough IO's on our control systems for controlling the additional clinching machine and the two linear actuators?
  - Which parts of the safety analysis should we re-perform

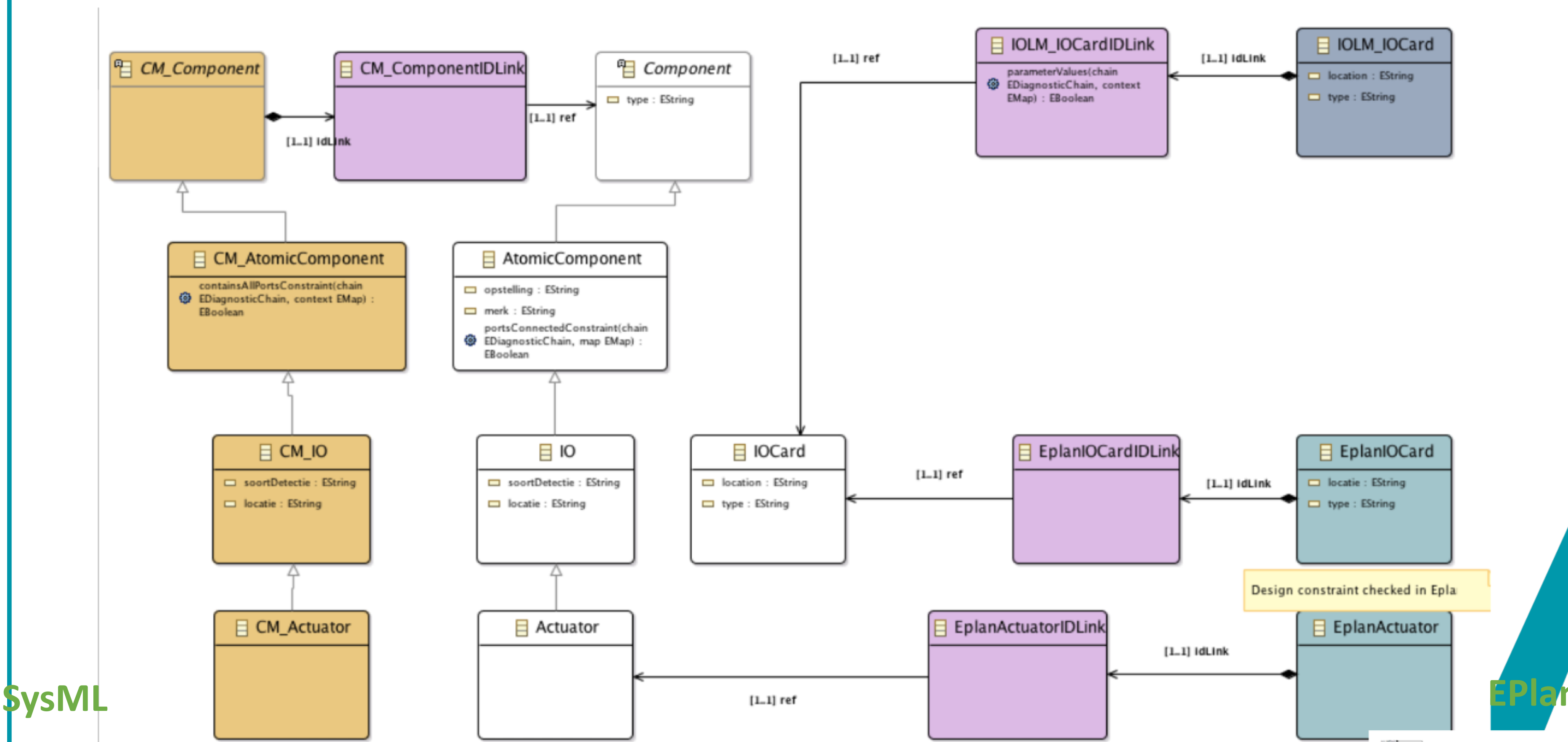
# Consistency tool architecture



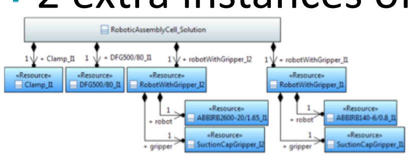
# Applied to the Robotic Manufacturing Cell

Excel



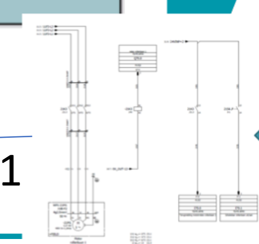



2 extra instances of CM_actuator



1

1



# Conclusions and future work

- The combination of a computational design synthesis tool and an inconsistency detection tool are two essential ingredients to support the design of today's complex systems such as robotic manufacturing cells.
  - The CDS tool allows for more exhaustive exploration of the design space in the early stages, but cannot avoid iterations, even when the detailed design is already on-going.
  - By basing both tools on related modeling languages and tools, the amount of new knowledge that has to be acquired by industrial users is reduced and the probability of adoption in an industrial context is increased.
- Even with this infrastructure in place, it is still better to avoid iterations than to execute them without inconsistencies… Future work will
  - Try to explicitly incorporate uncertainty in the CDS framework
  - Investigate the feasibility of automatic 'clustering' of design candidates
  - Try to work on reducing the setup cost of the inconsistency tooling
  - Work towards inconsistency 'resolution' (<-> detection)

# Questions/Remarks?

klaas gadeyne at flandersmake dot be