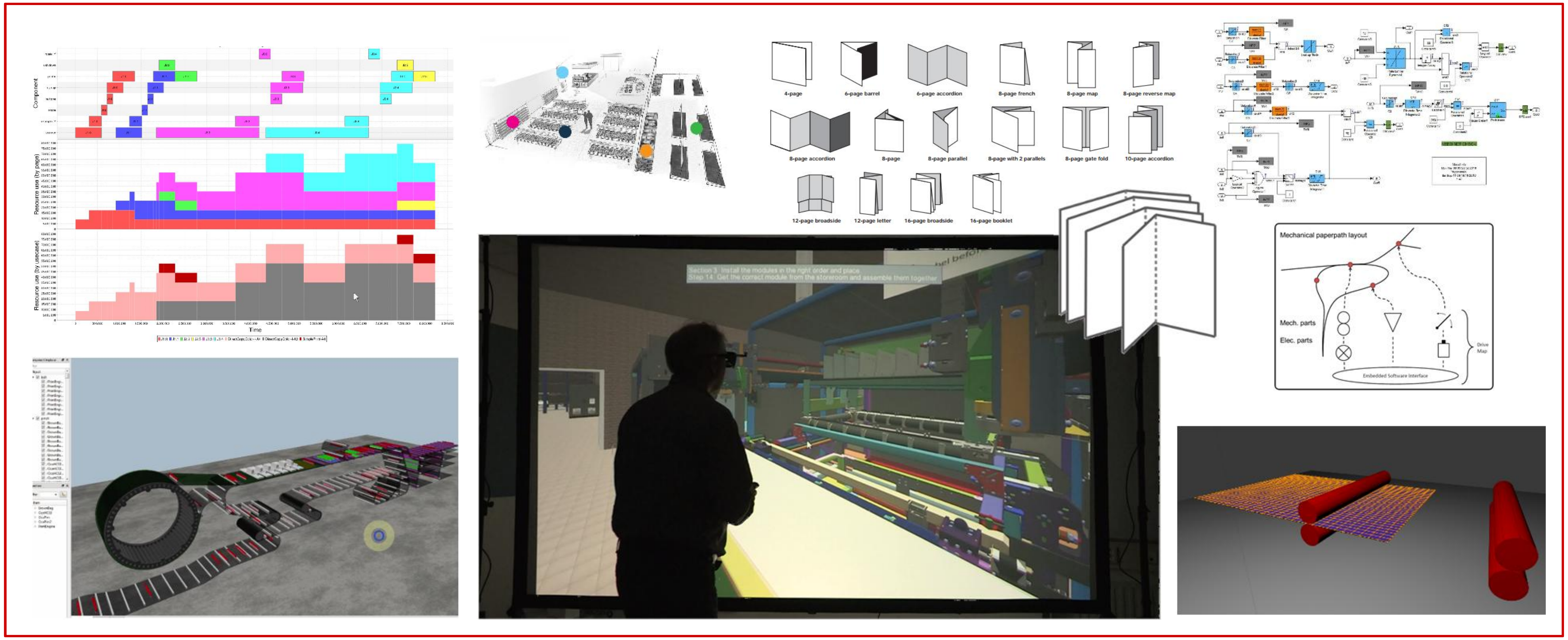


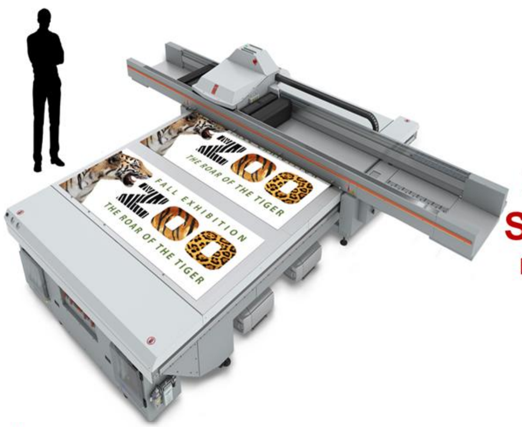
Success factors for Model Based Development at Océ

Toon van Dijk (June 5th, 2018)
Department Manager Embedded Software

A CANON COMPANY

Why MBD ?

- Time To Market
- Product Characterisics
    - Quality
    - Cost
    - Power Consumption
    - Performance
    - Usability
    - ....
- R&D Efficiency



Océ Arizona 6170 XTS printer

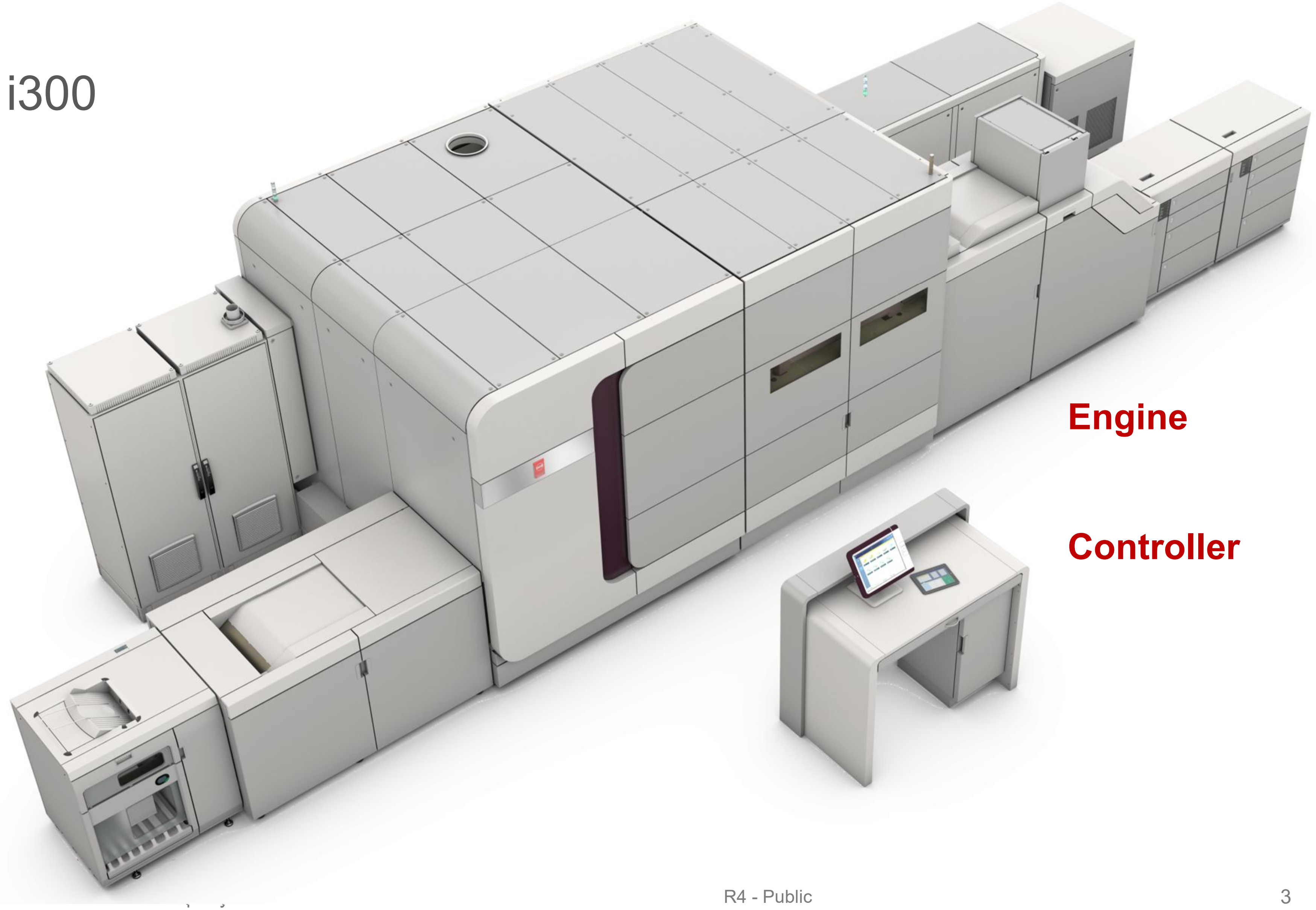
Océ VarioPrint i300 inkjet press

Océ ImageStream 3500 printer

Managing Complexity !

Breaking down a print system

i300

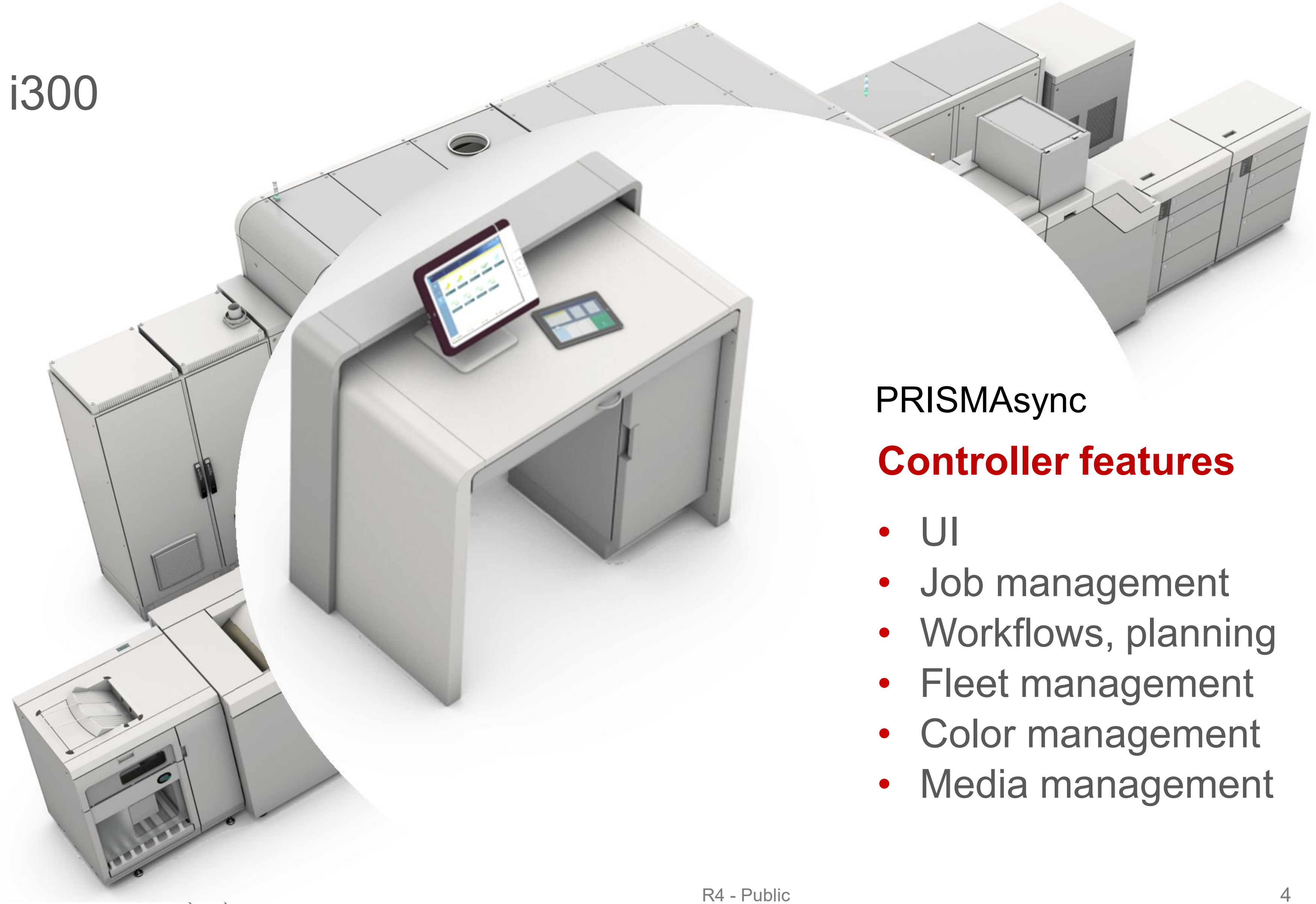


Engine

Controller

Breaking down a print system

i300



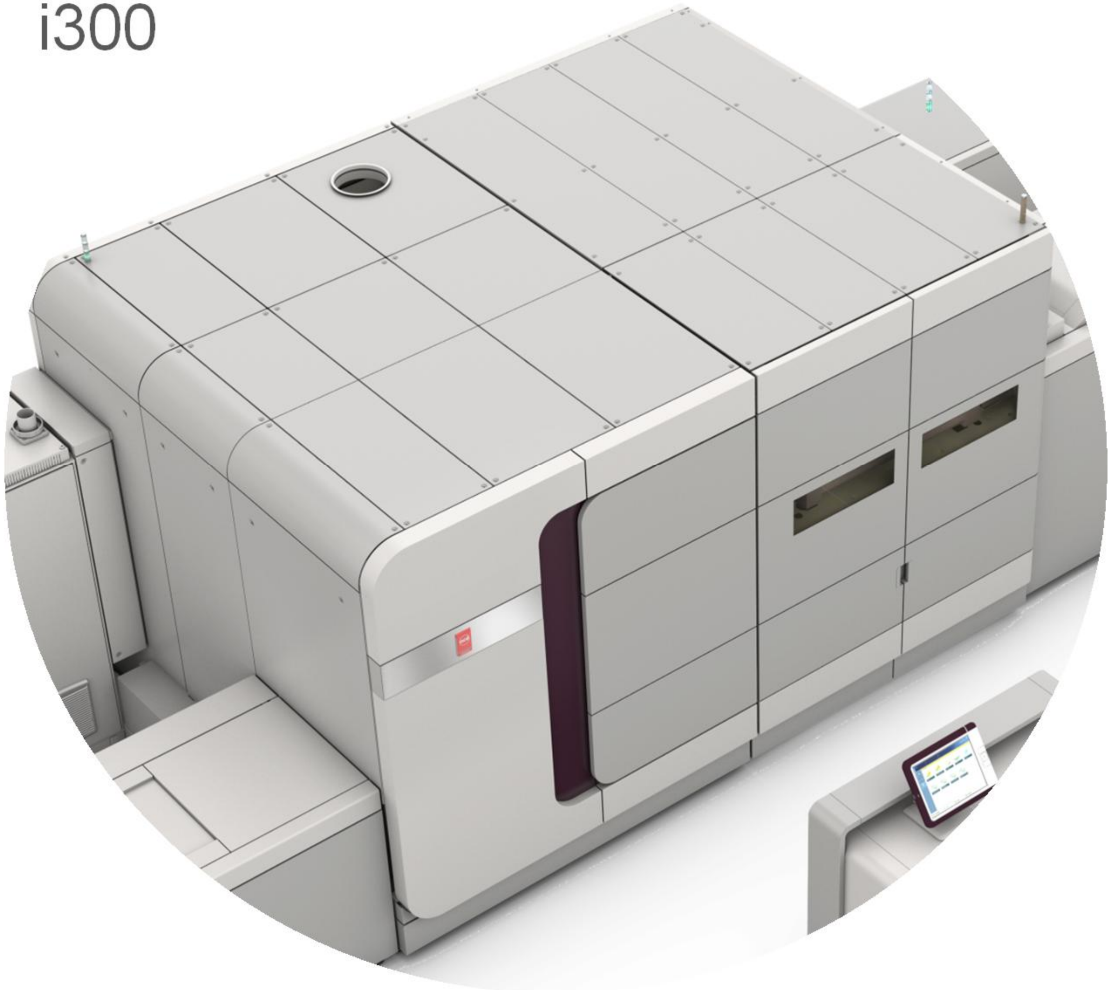
PRISMAsync

Controller features

- UI
- Job management
- Workflows, planning
- Fleet management
- Color management
- Media management

Breaking down a print system

i300

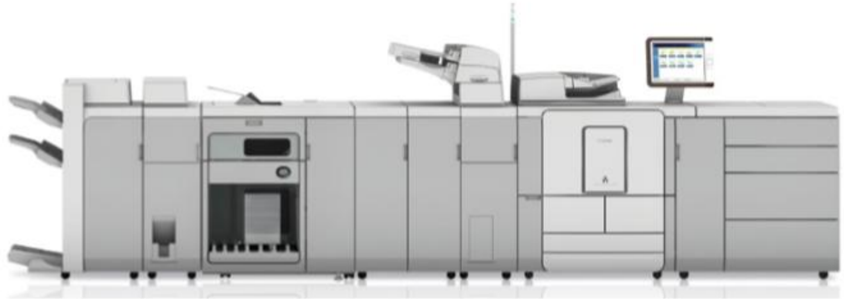


Engine features

- Brings engine alive with behavior (control, measurement)
- Systems level multi-disciplinary aspects:
  - Productivity
  - Data trends/analysis
  - Variability
  - Error handling

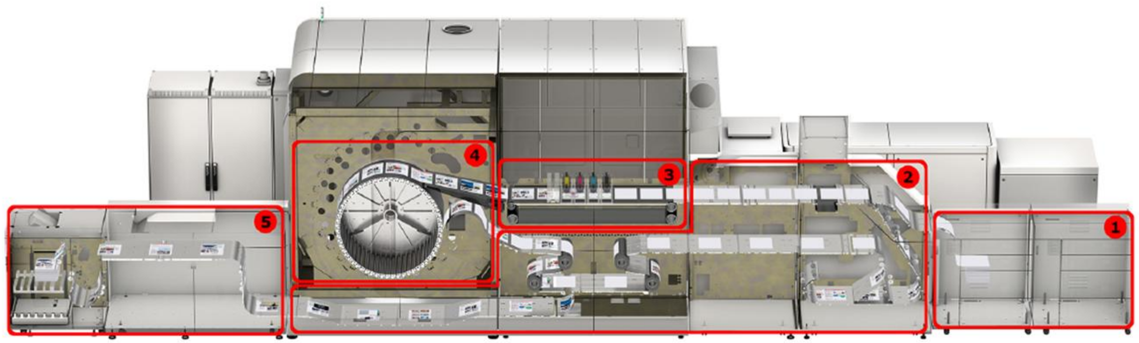
Variability

Same engine,
different configuration
(input and finishing/output)

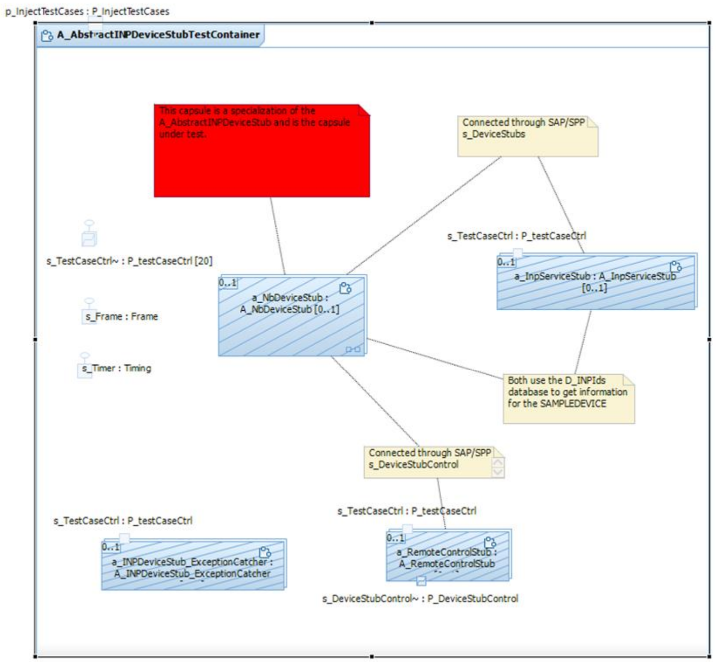


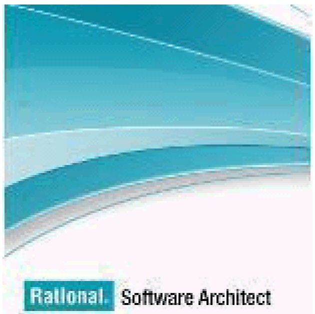
Comparable configuration,
different engines

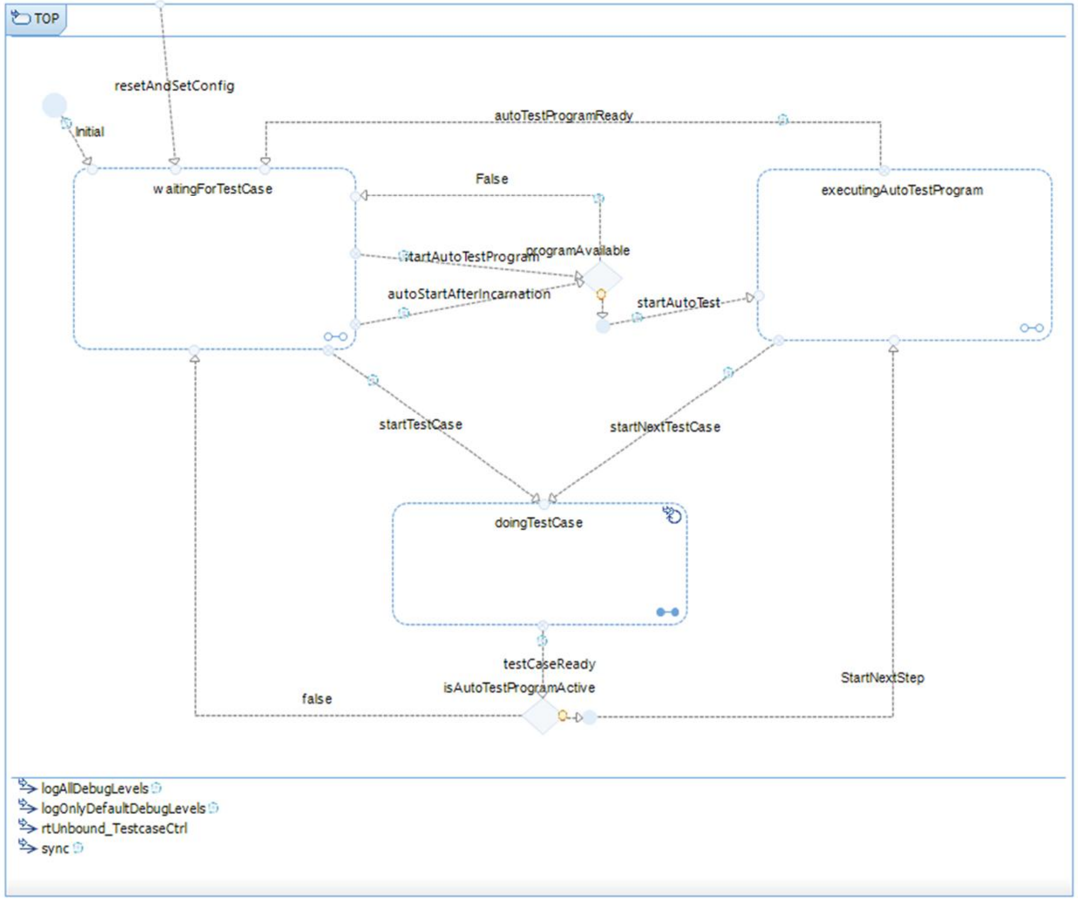
Variability
within the engine:
1. Paper input module
2. Paper transport module
3. Print module
4. Fixation module
5. Finishing equipment



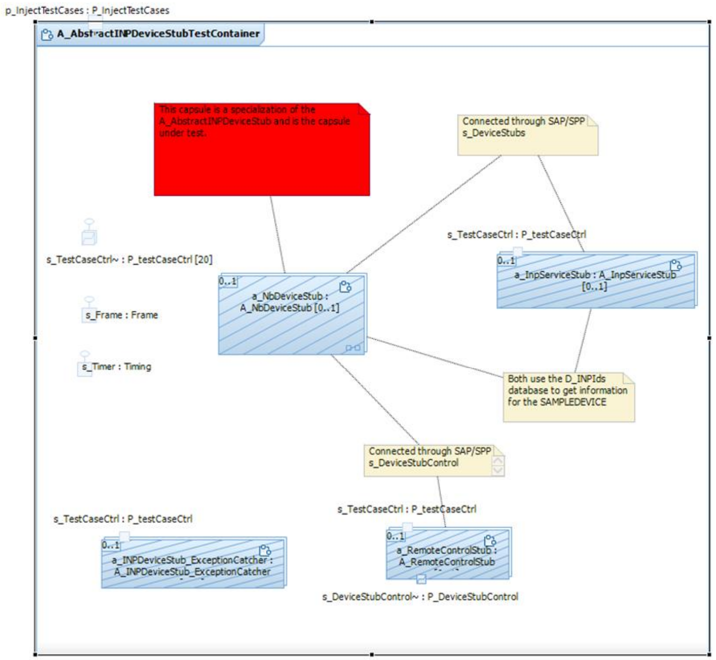
Modeling is no stranger to us



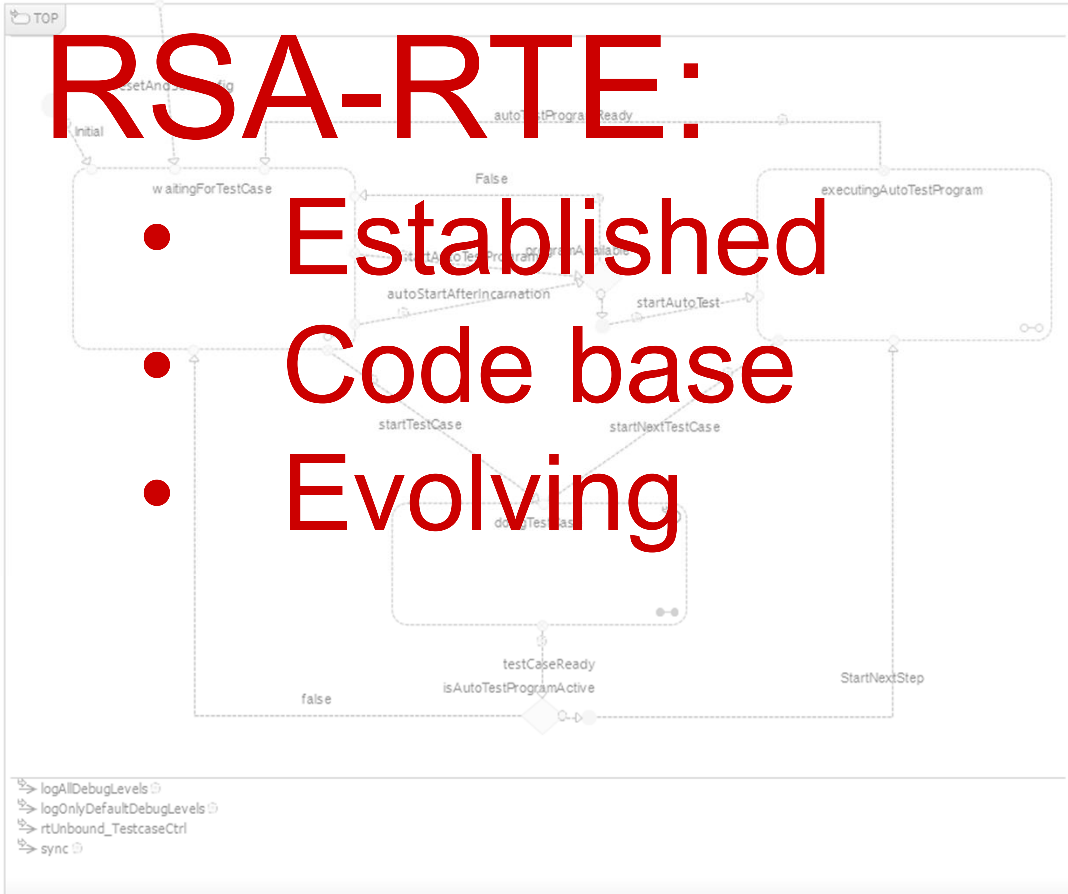




Modeling is no stranger to us



ROOM using RSA-RTE:



- Established
- Code base
- Evolving

Technologies

Controller (cut-sheet):

- Single configurable code base
  - C, C++, C#, Java, TypeScript, XML, ...
  - PowerShell, Python, Lua, ... (for build & test)
- Co-developed at five different R&D sites
- Supports ±8 product families
- Supports Océ and Canon engines

Engines:

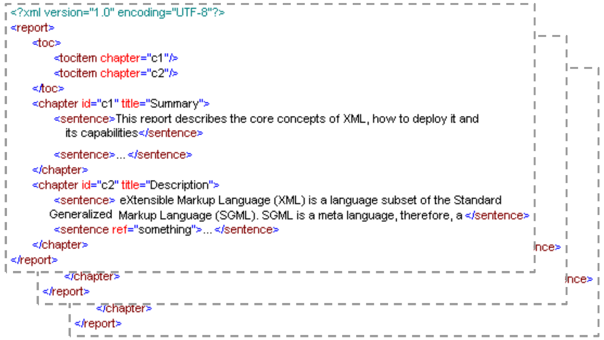
- Code bases per product family, with a shared reuse architecture and infrastructure
  - C, C++, VHDL, Matlab, RSA-RTE
  - Python and Lua for testing & data science
- Multiple sites, dedicated development of specific engine types

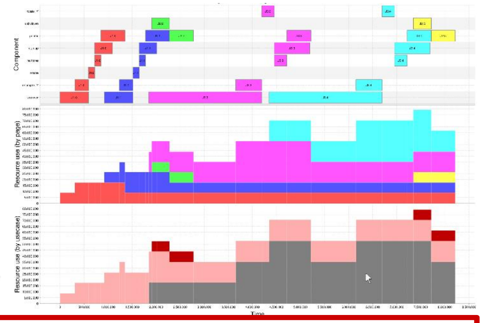
Both:

- Mature development/build/test environment:
  - Mature engineering tools and version control (VS, RSA-RTE, TFS)
  - Automatic deployment of development and test environment
  - Nightly build and test (unit tests, module tests, system test framework)
- MDE for complex control using state machines
  - Synthesis – custom MPS-based tooling (components, generates C++ & Boost) , RSA-RTE
  - Analysis – OIL (components & interfaces, proof-of-concept tooling)

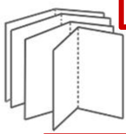
Some MBD examples

- Interface specifications
- System capabilities
- Features

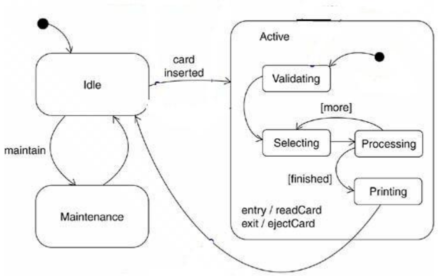

```
<?xml version="1.0" encoding="UTF-8"?>
<report>
  <toc>
    <tocitem chapter="c1"/>
    <tocitem chapter="c2"/>
  </toc>
  <chapter id="c1" title="Summary">
    <sentence>This report describes the core concepts of XML, how to deploy it and
     its capabilities</sentence>
    <sentence>... </sentence>
  </chapter>
  <chapter id="c2" title="Description">
    <sentence> eXtensible Markup Language (XML) is a language subset of the Standard
  Generalized  Markup Language (SGML). SGML is a meta language, therefore, a </sentence>
    <sentence ref="something">... </sentence>
  </chapter>
</report>
```

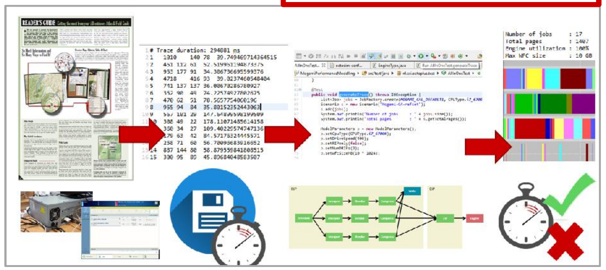


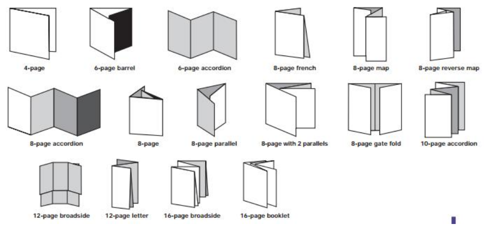
Integral productivity
(job, SW, HW, operator behavior, floor plan)



Performance



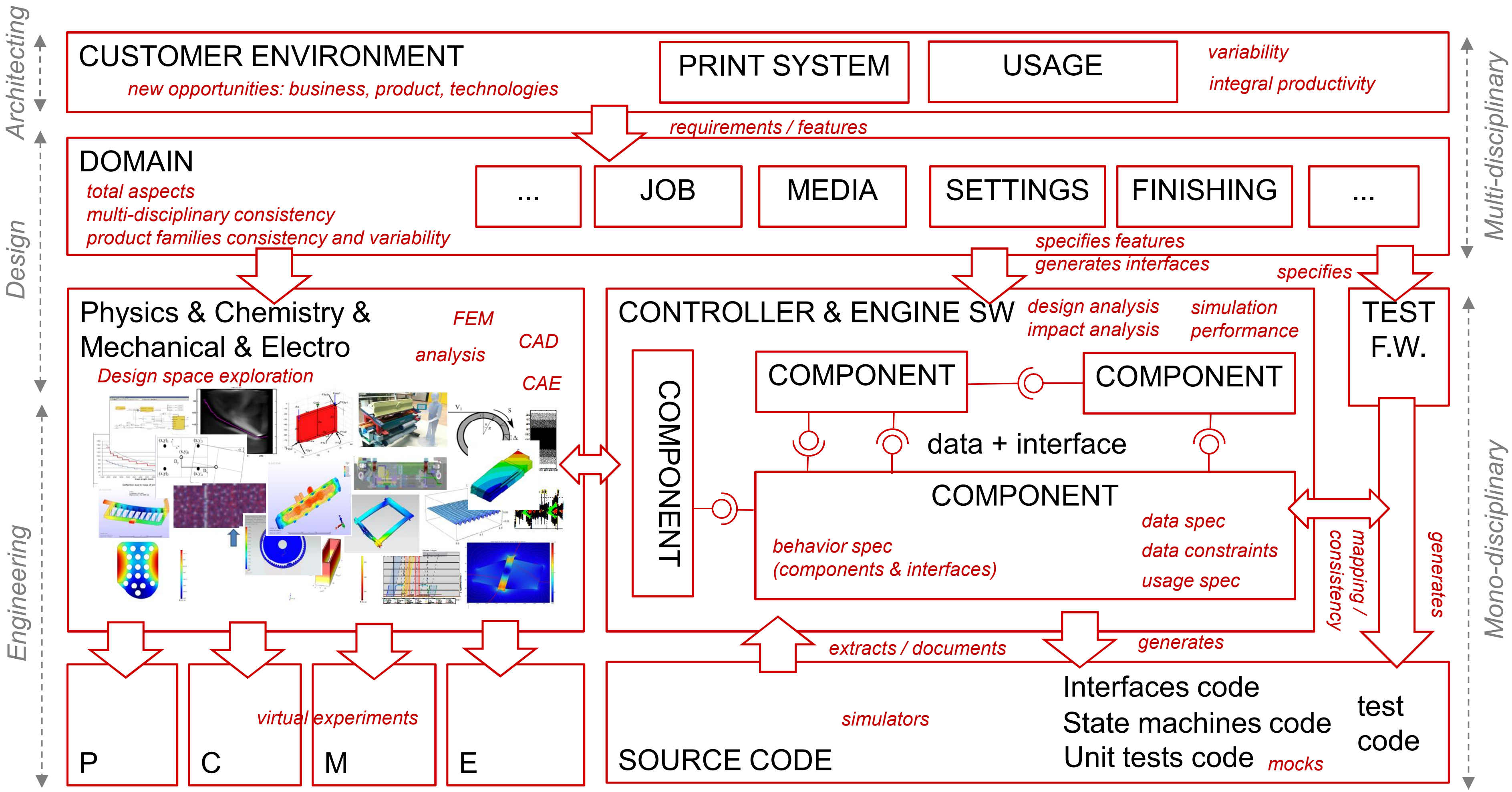




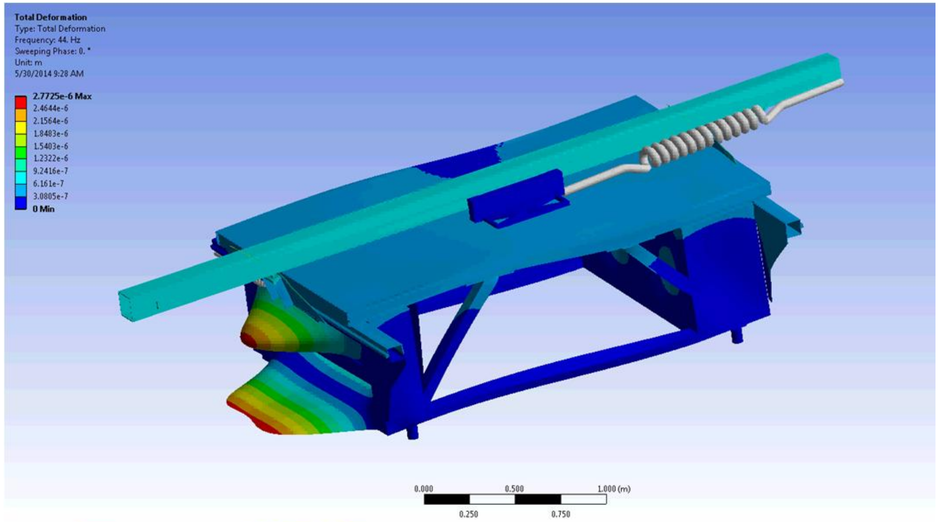
State Machines
(Component & Interface)

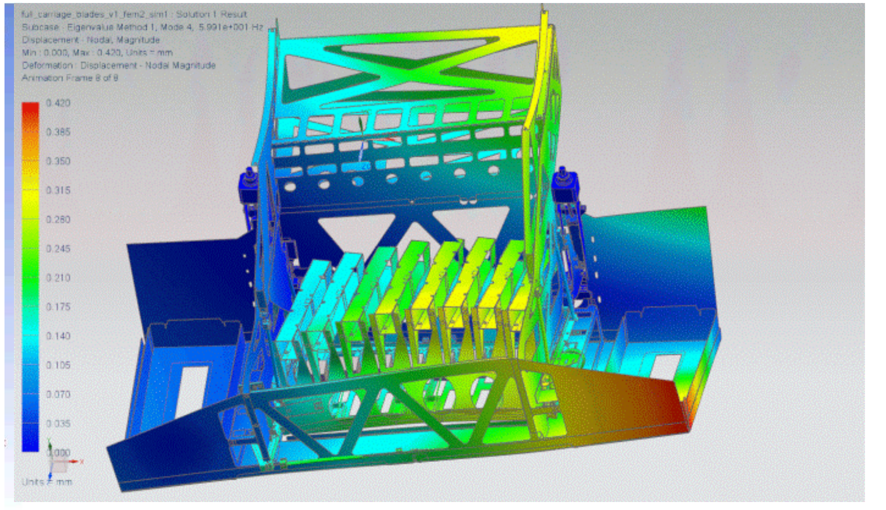
Impositions
(Finishing vs Applications)

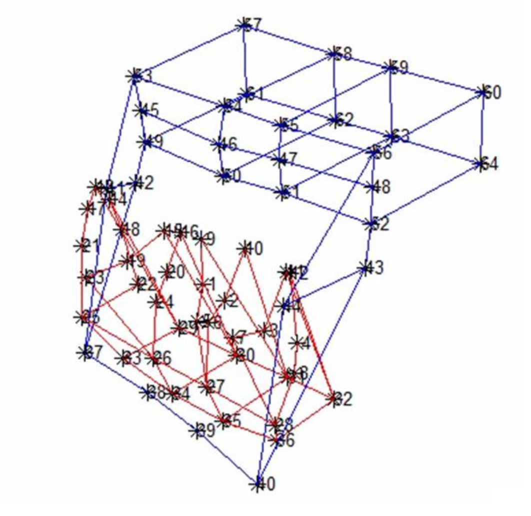
MBD landscape of opportunities

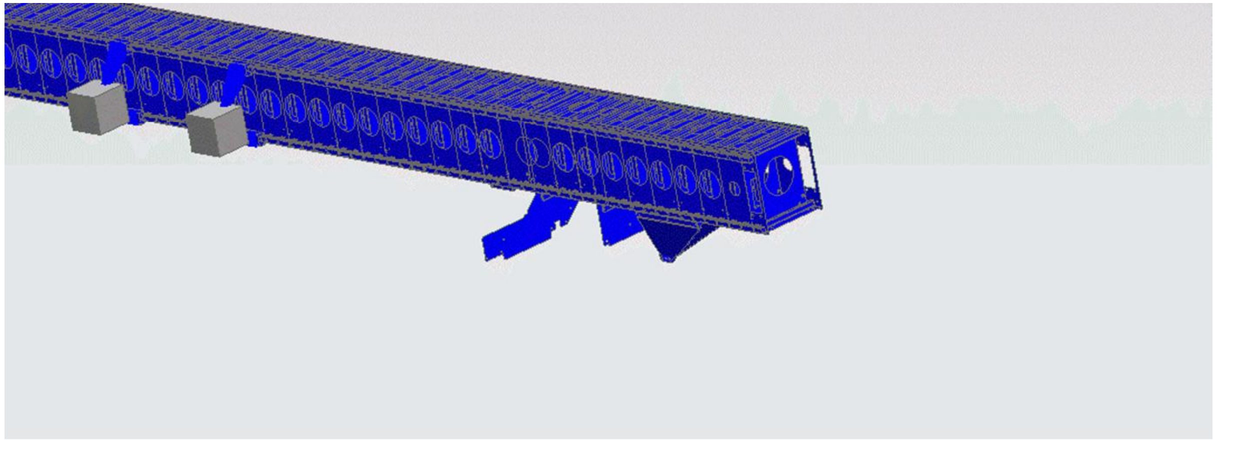


Example: In control over Dynamics

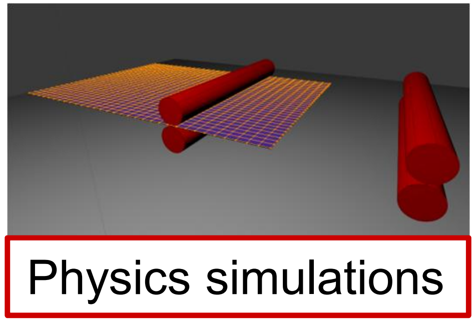


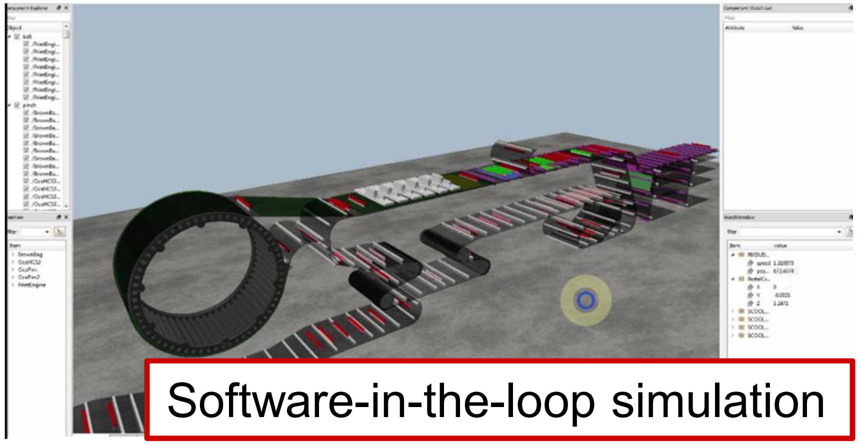


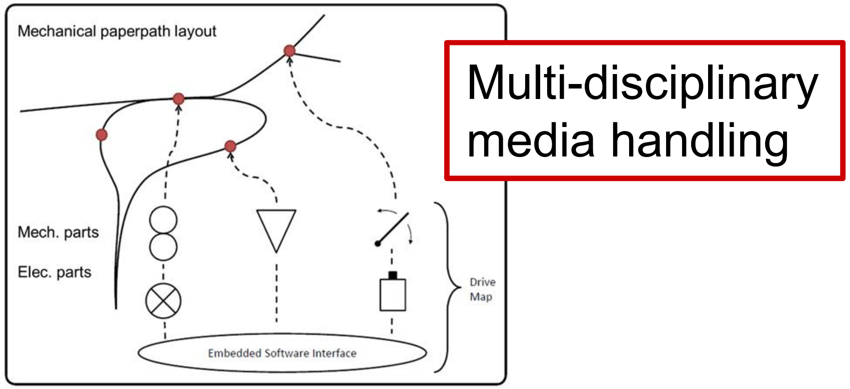


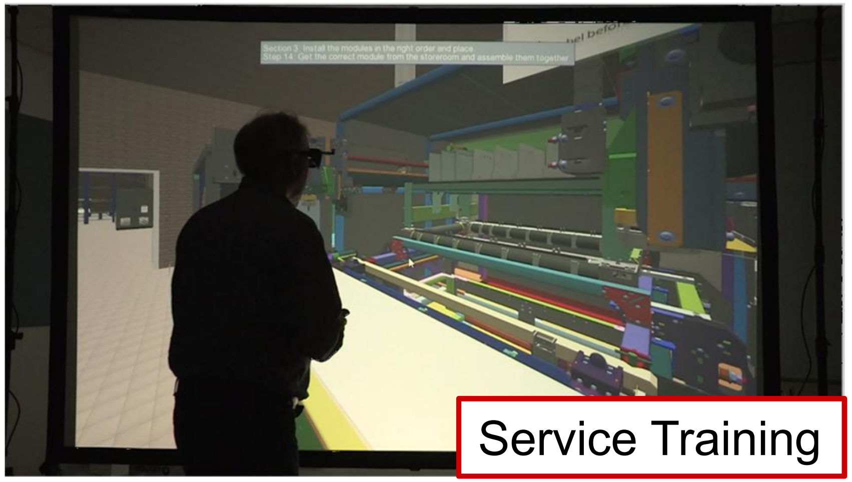


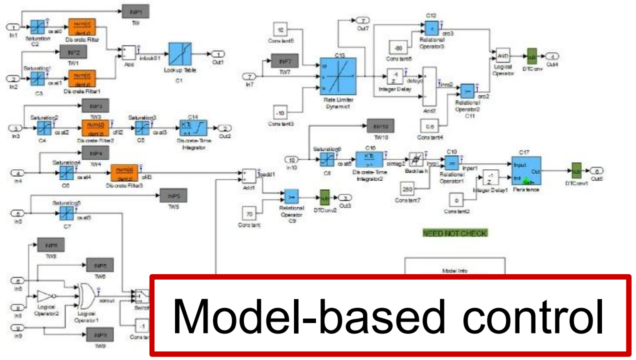
Other MBD examples….



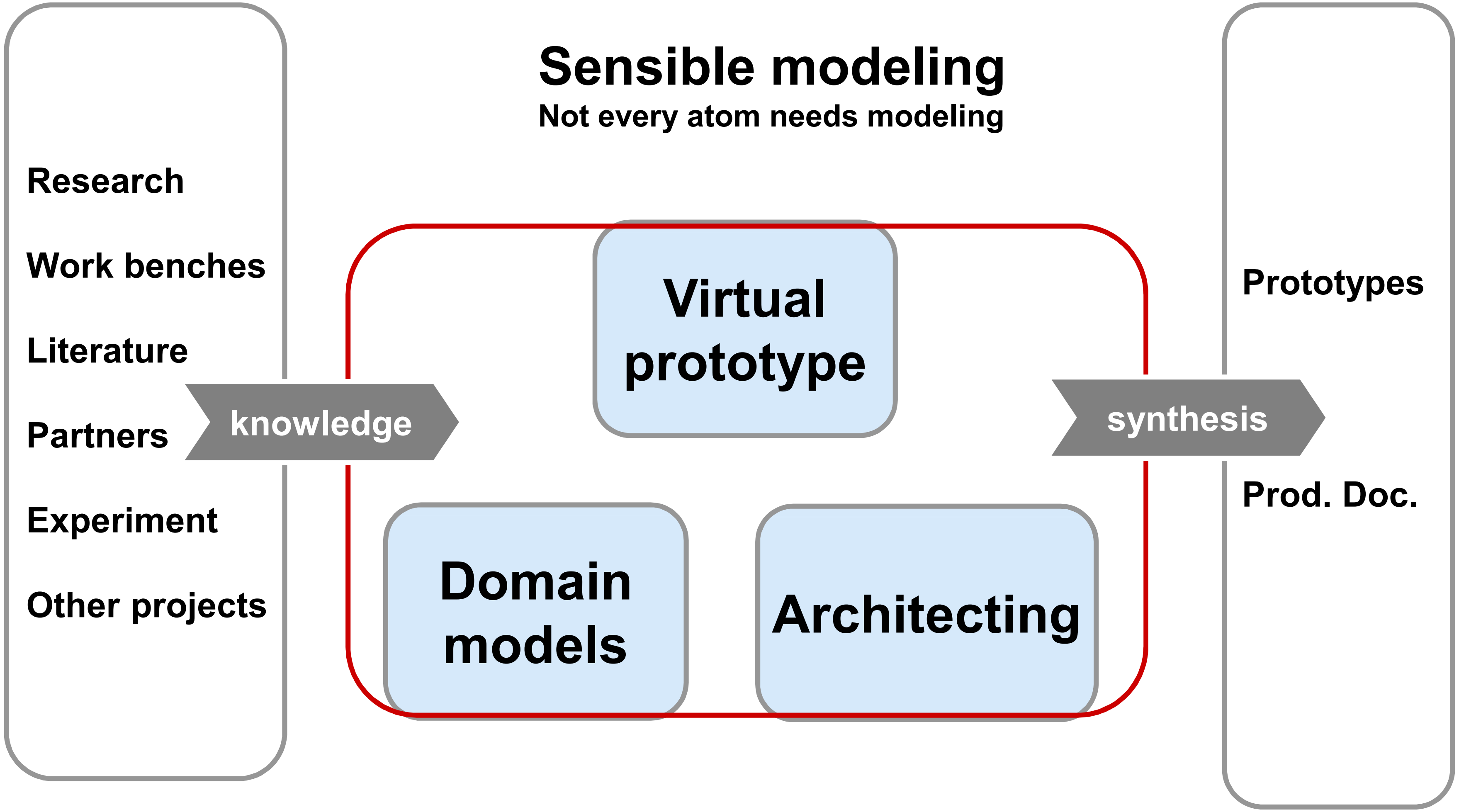






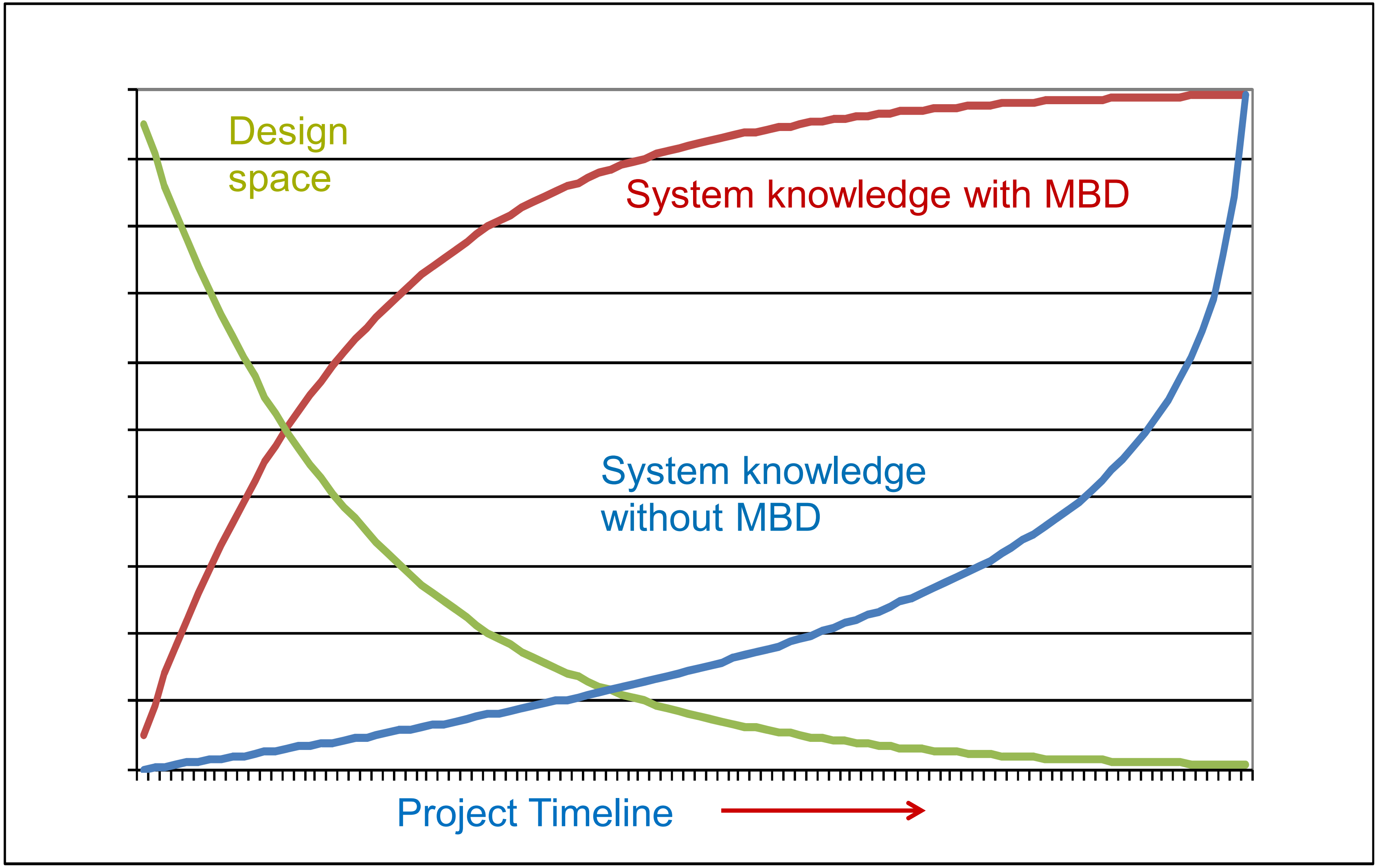


Product Development Vision

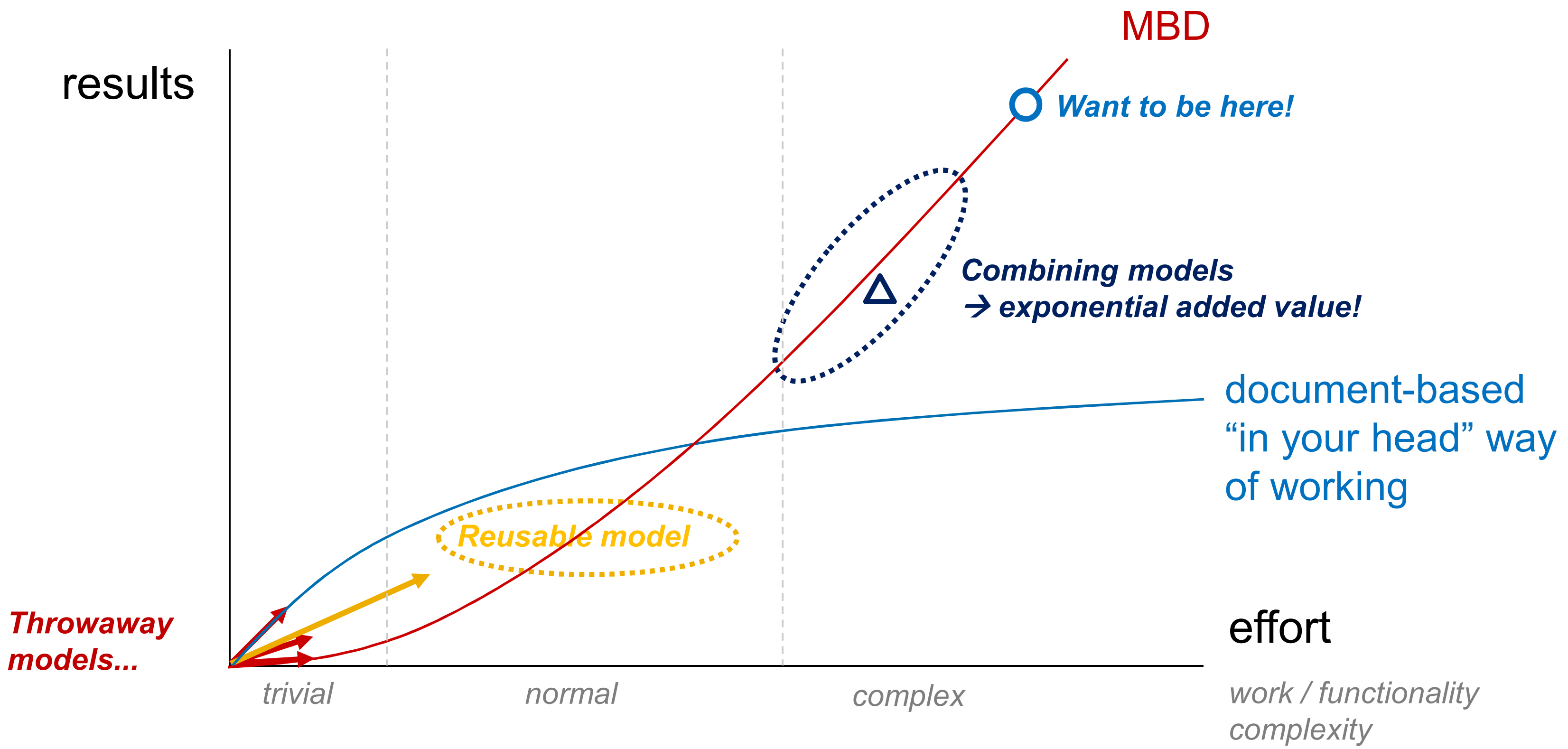


80% -- Risk reduction & architecture verification 20%

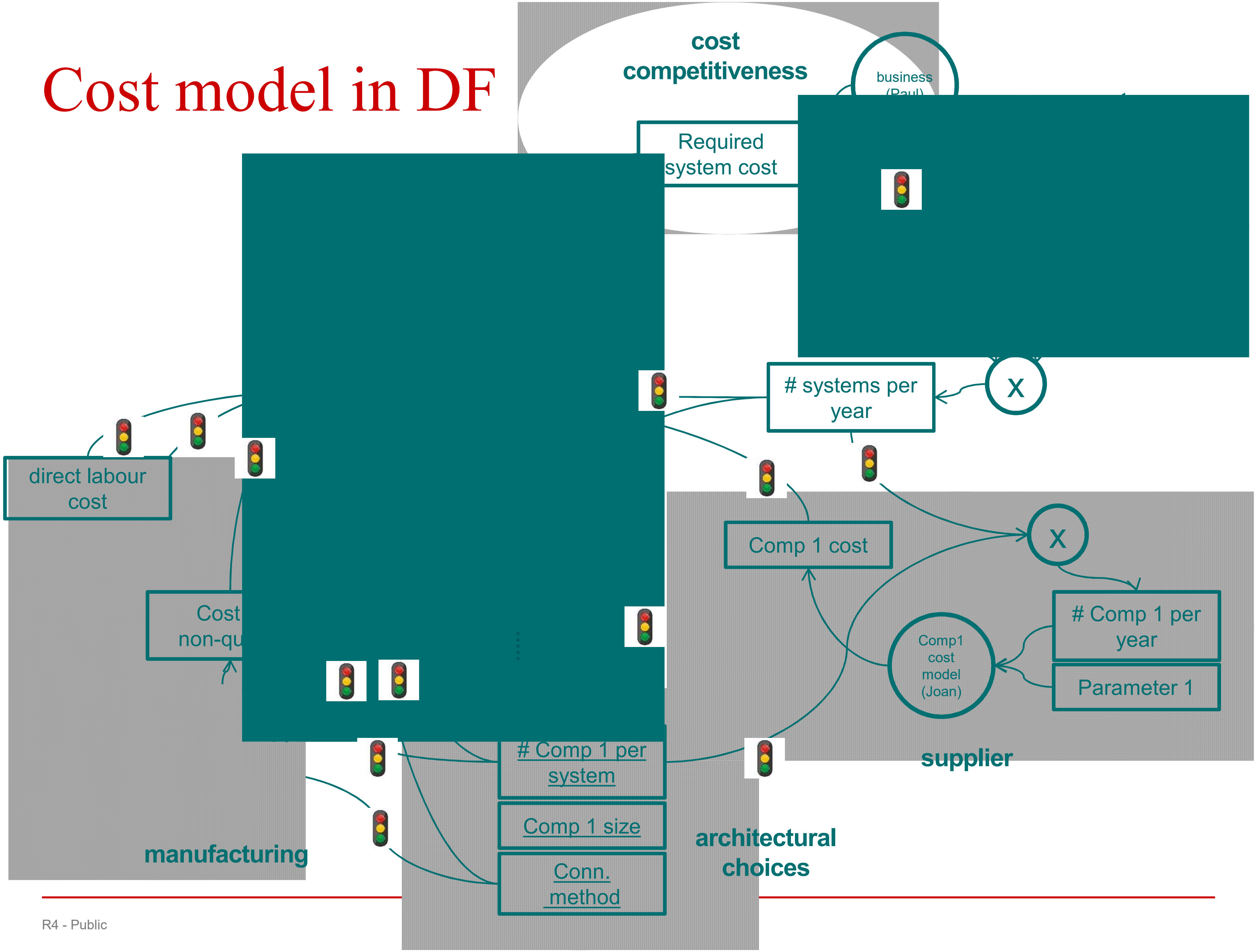
Model Based Design



Maintainable & reusable modeling



Cost model in DF



The Modeling Manifesto

We are uncovering better ways of developing products by doing it and helping others do it. Through this work **we have come to value**:

Virtual prototypes over physical models

Simple, easy-to-change models over complex detailed static ones

Demonstrable models over submarine ones

Modeling in teams over lone wolf modeling

Connected models over stand-alone models

Self-documented models over classic documentation

That is, while there is value in the items on the **right**, we value the items on the **left** more.

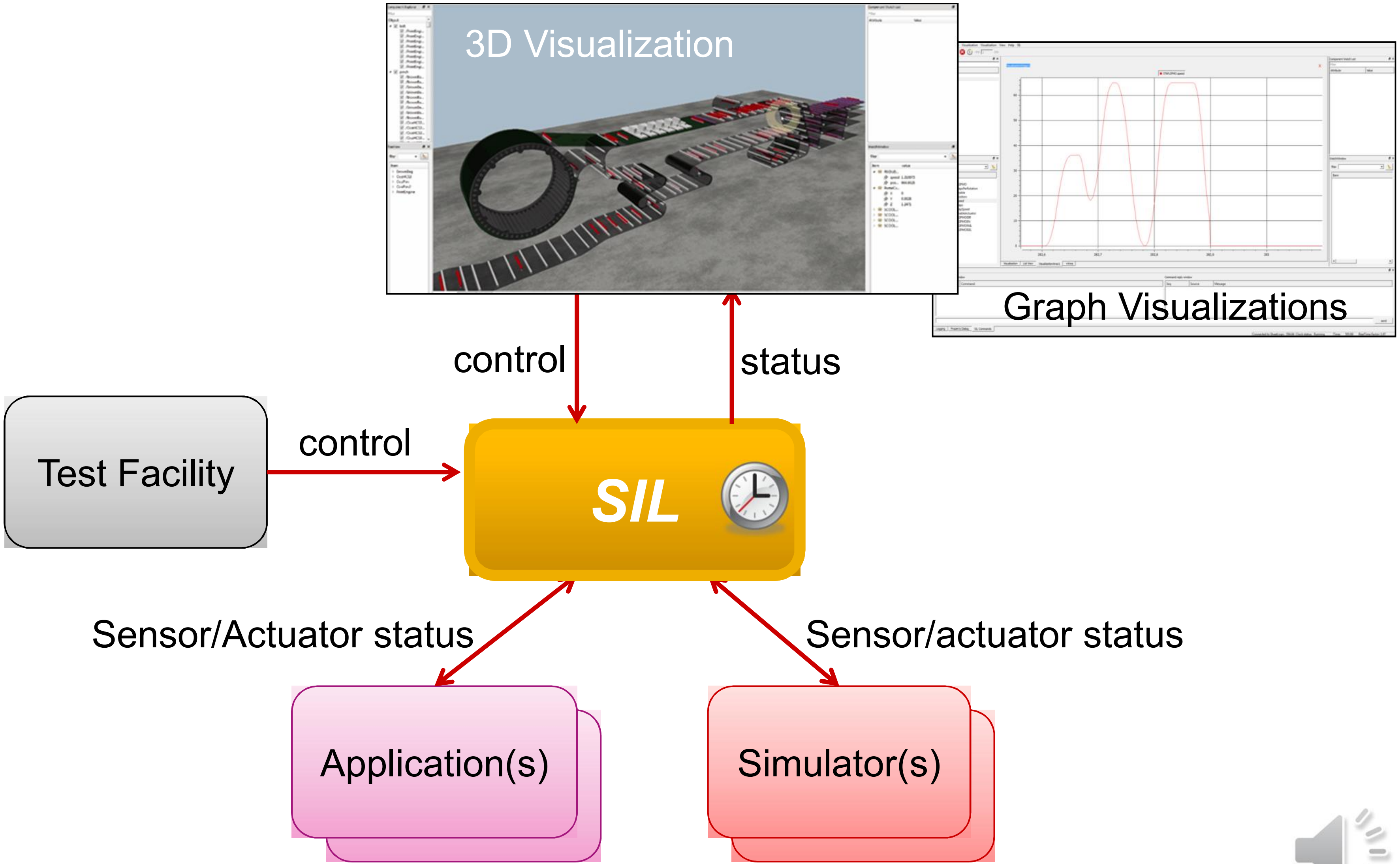
Why use virtual printer prototypes?

Océ VarioPrint 110

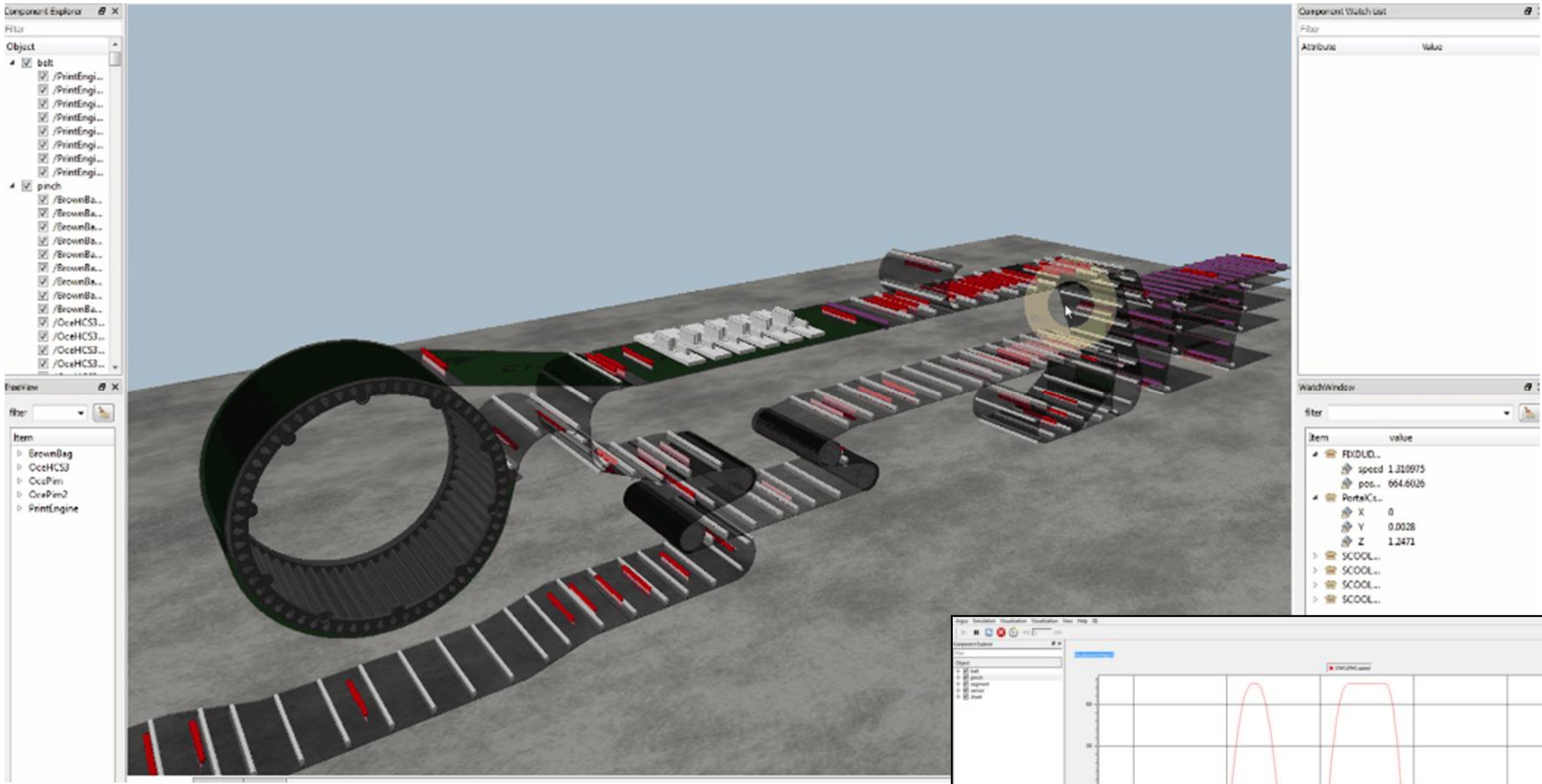
- Printer availability; costs
- Improved understanding
- Shorter development time
- 24/7 testing
- Environmental benefits

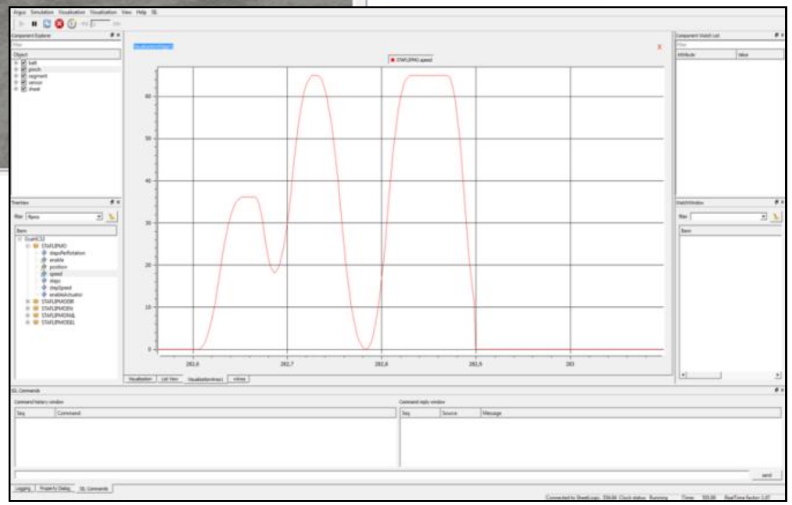
Océ VarioPrint i300

How it works: introduction to SIL



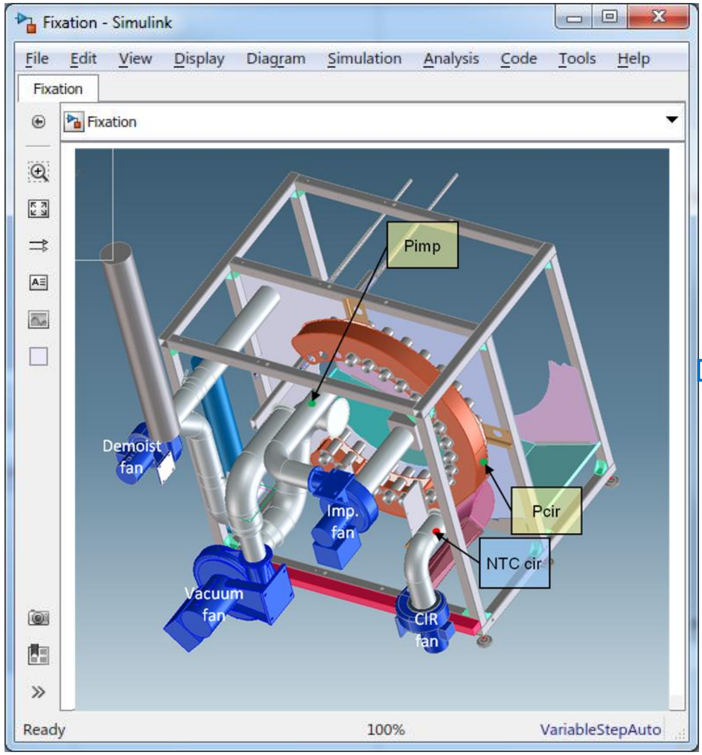
Case: VarioPrint i300 3D Visualization

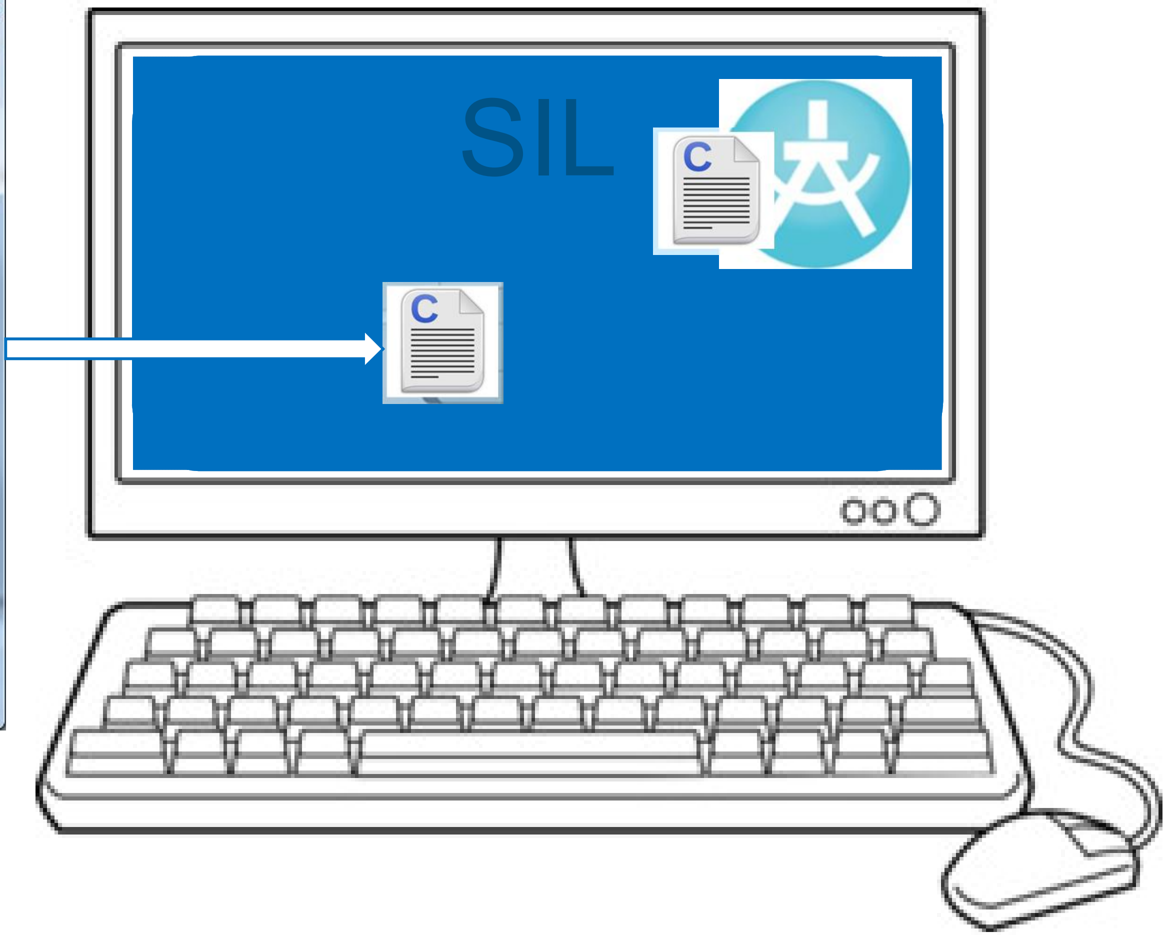




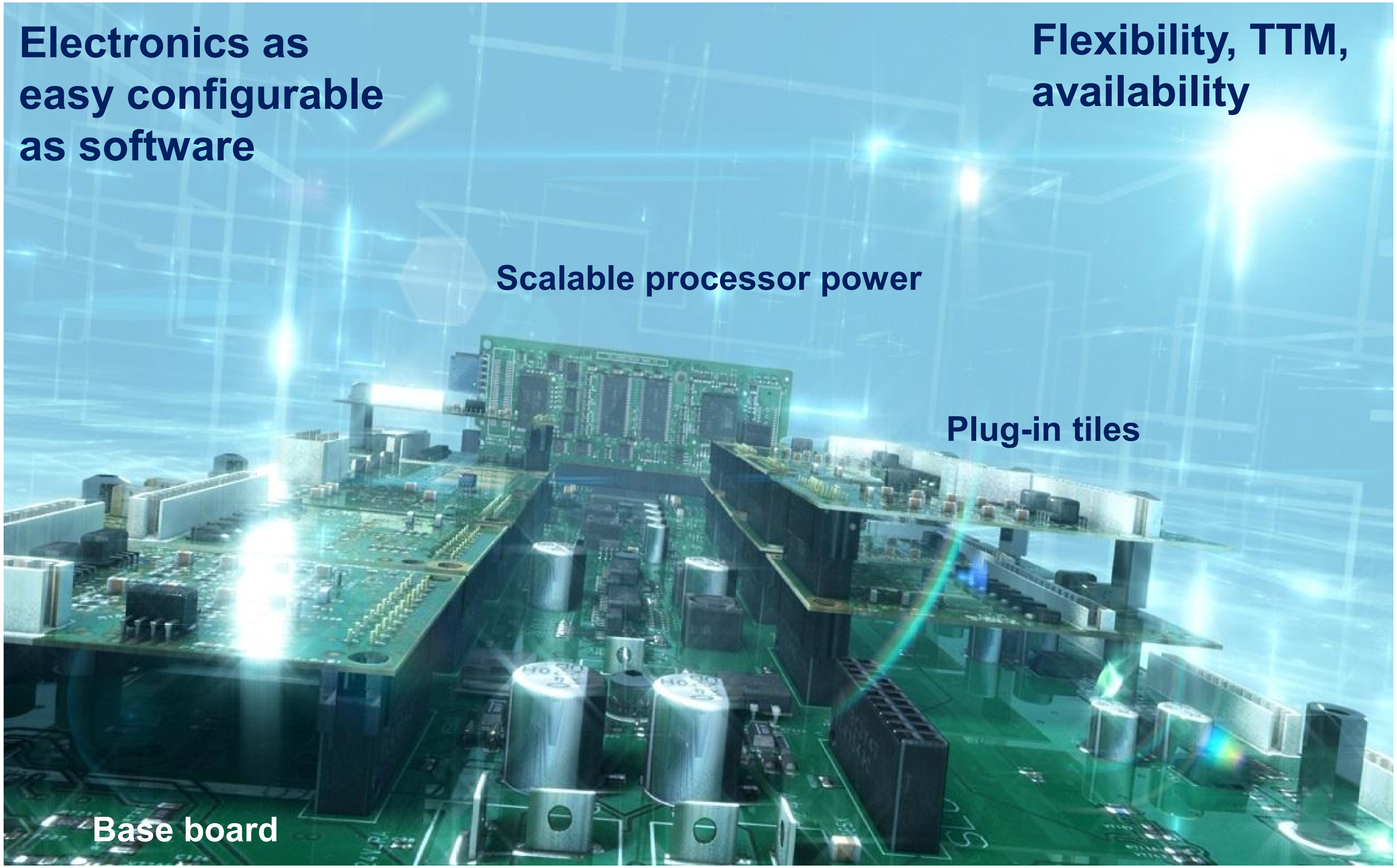
Detailed timing diagrams

Plant modelling & SIL



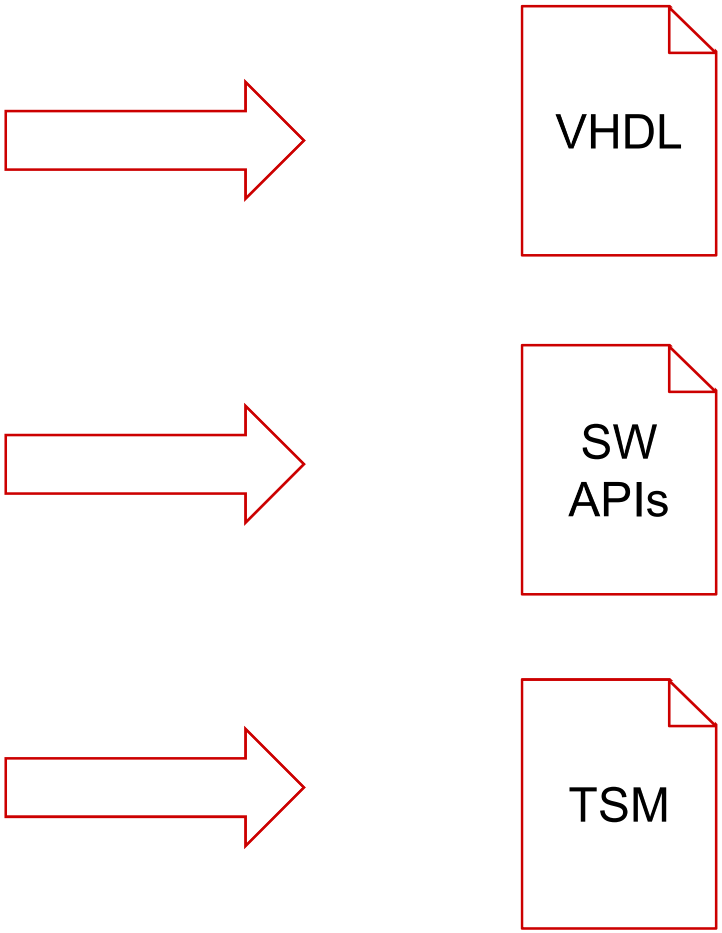


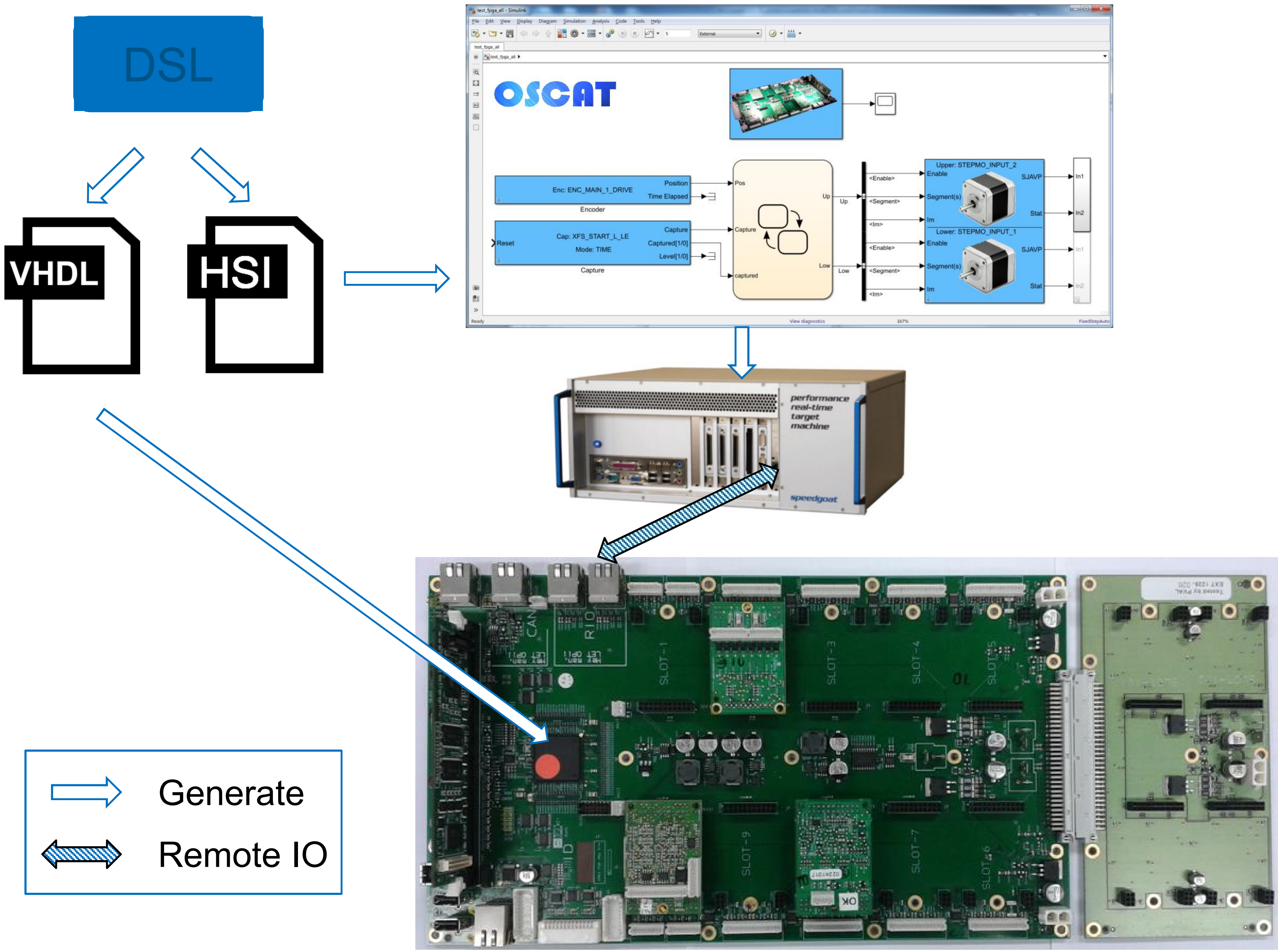
Modular Océ Reusable PlatForm



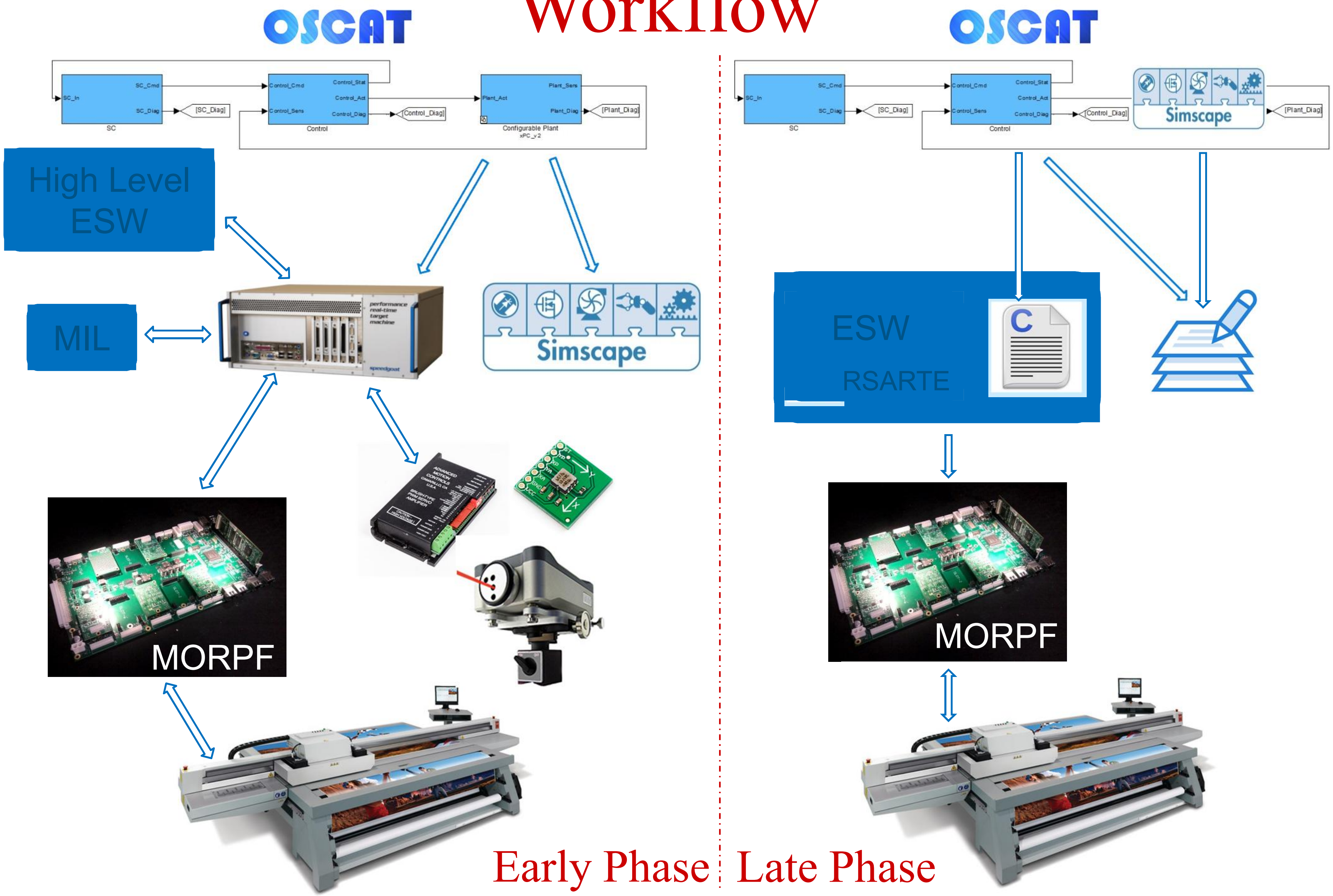
Electronics/Software Interface

```
Project/Board : DEMO_HSI
  BLDC3A5 Tile   PINCH_MOTOR
    slot : 1
    tile_name : bldc3a5
    tile_version : 2.0
    motor : DCMotor : PINCH_MOTOR
              pwm : PWM
                      name : PINCH_MOTOR_PWM
                      frequency : 20000
                      enable : PINCH_MOTOR_ENA_N
                      direction : PINCH_MOTOR_DIR
                      inverted : no

              enable : SimpleOutput
                         name : PINCH_MOTOR_ENA_N
                         activeState : ACTIVE_LOW

              direction : SimpleOutput
                            name : PINCH_MOTOR_DIR
                            activeState : ACTIVE_HIGH

              hall : HallDecoder
                       name : PINCH_MOTOR_HALL

              encoder : QuadratureDecoder
                          name : PINCH_MOTOR_ENC


    fault : SimpleInput
              name : PINCH_MOTOR_FAULT_N
              activeState : ACTIVE_LOW

    brksel : SimpleOutput
               name : PINCH_MOTOR_BRKSEL
               activeState : ACTIVE_HIGH

    spi : SPI
            name : PINCH_MOTOR_SPI
            chip selects : name : PINCH_MOTOR_EEPROM_CS
```



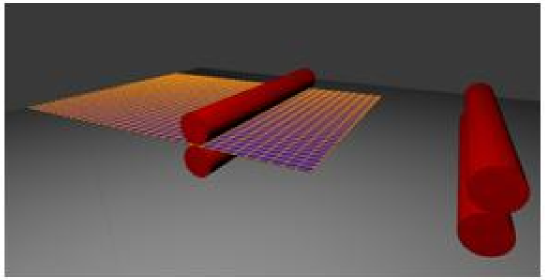

DSL
VHDL
HSI
OSCAT
Enc: ENC_MAIN_1_DRIVE
Position
Time Elapsed
Encoder
Cap: XFS_START_L_LE
Mode: TIME
Capture
Captured[1/0]
Level[1/0]
Reset
Upper: STEPMO_INPUT_2
Lower: STEPMO_INPUT_1
performance real-time target machine
speedgoat
Generate
Remote IO

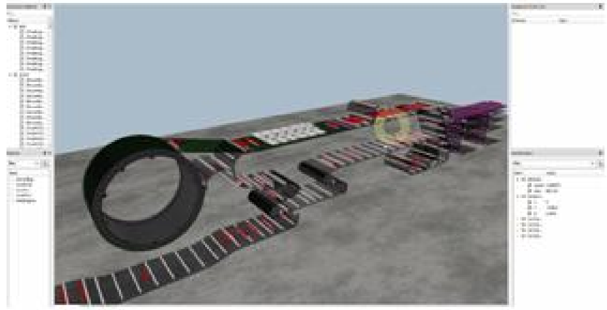
Workflow



Early Phase Late Phase

A3 Sheet Productivity





Machine Settings

Fanspeed

drumspeed

Media

A4

A3

DL908490
Mitsubishi

w, h, t,
d, MDλ,
CDλ, μ

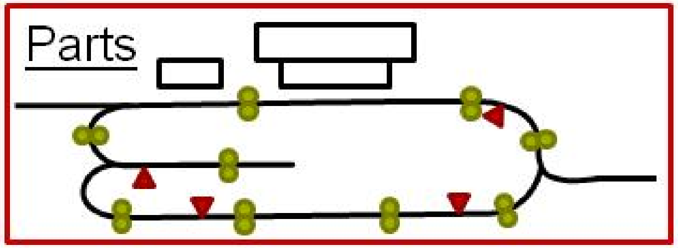
Timing Behavior

$v = v_{low}$

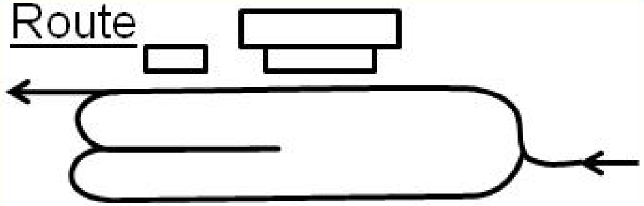
$v = v_{high}$

$a = a_{max}$

Parts

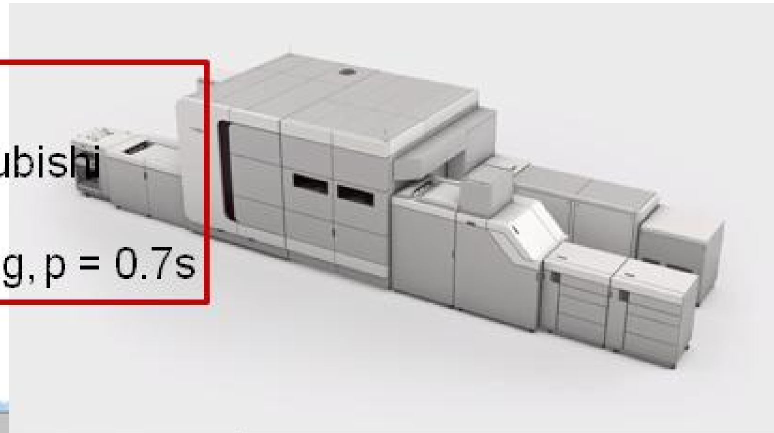


Route

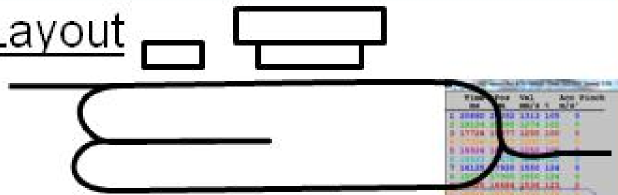


Job Specification

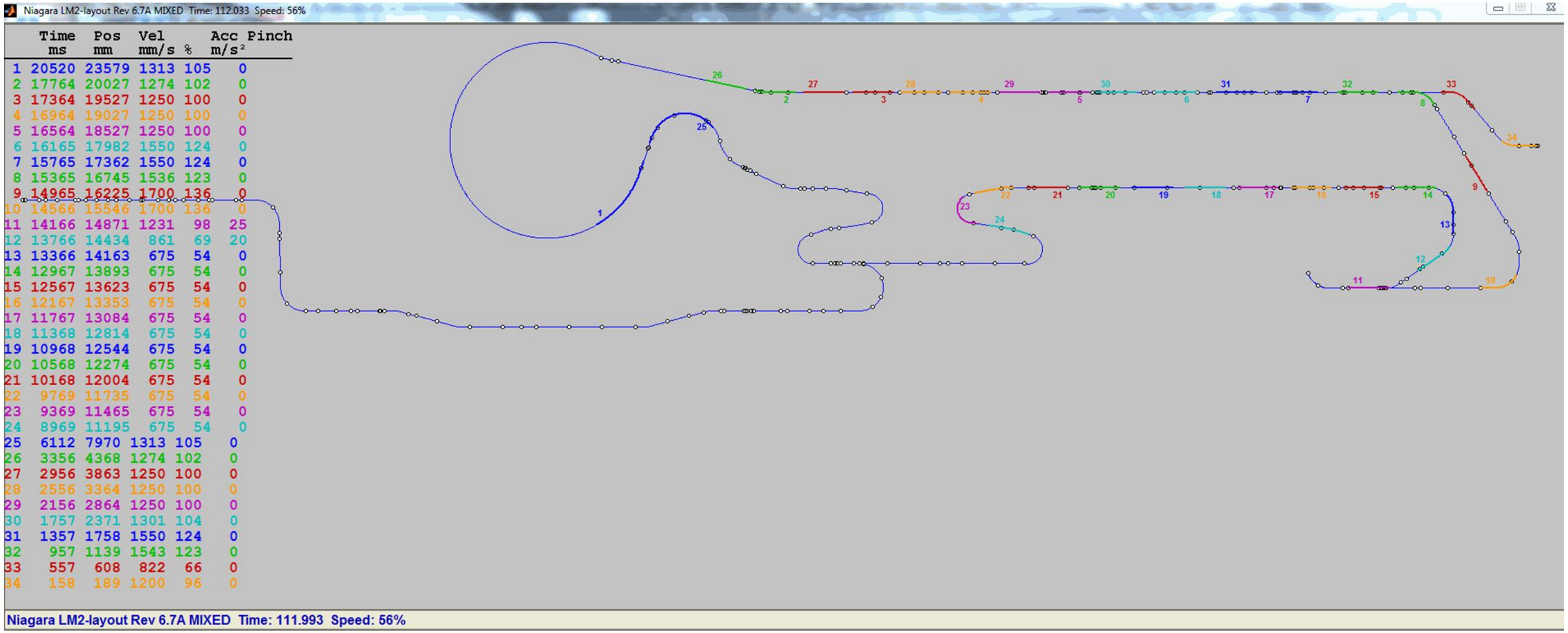
70 x DL908490 Mitsubishi
Simplex
Monotonic scheduling, p = 0.7s



Layout

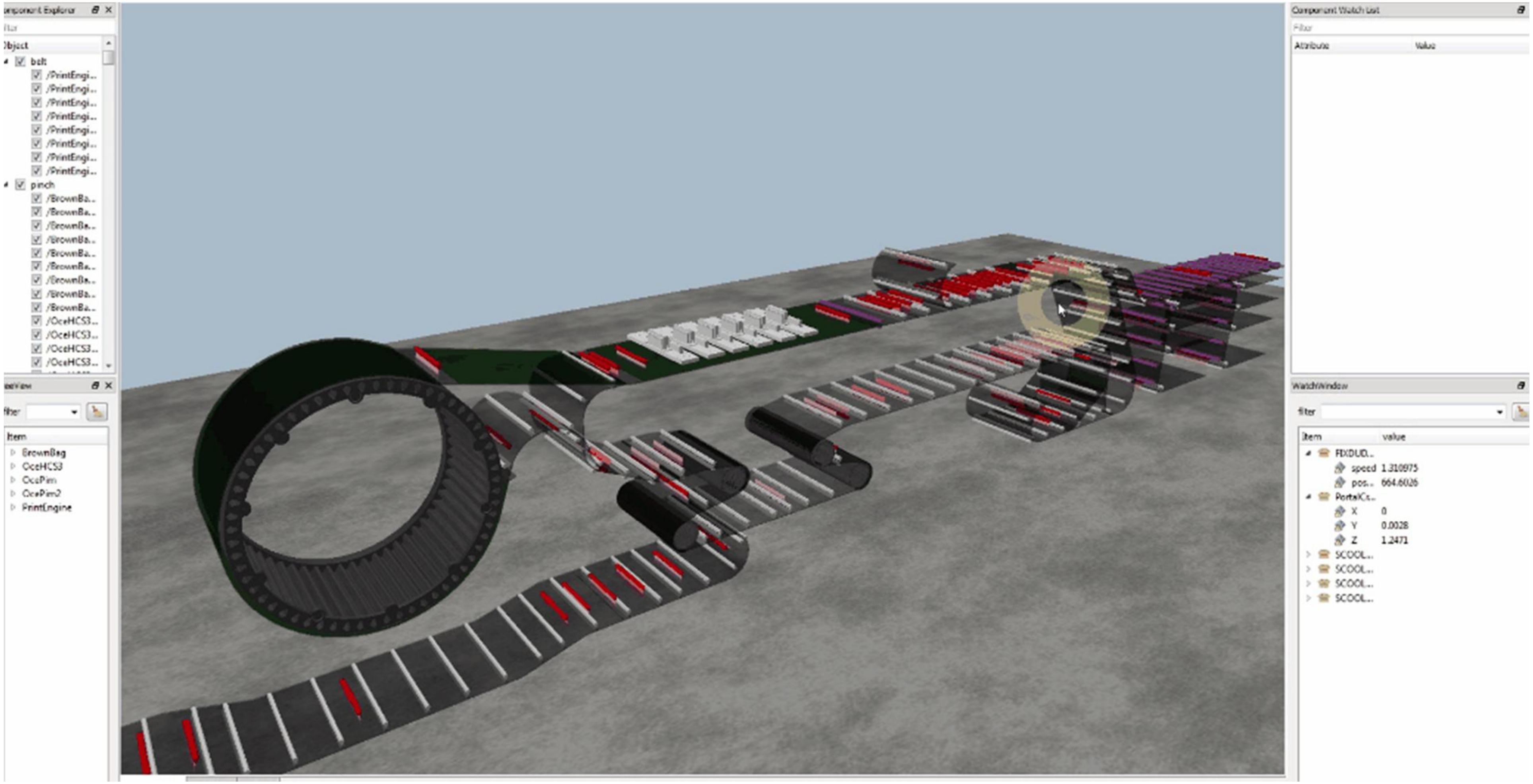


HappyFlow sheet timing simulation

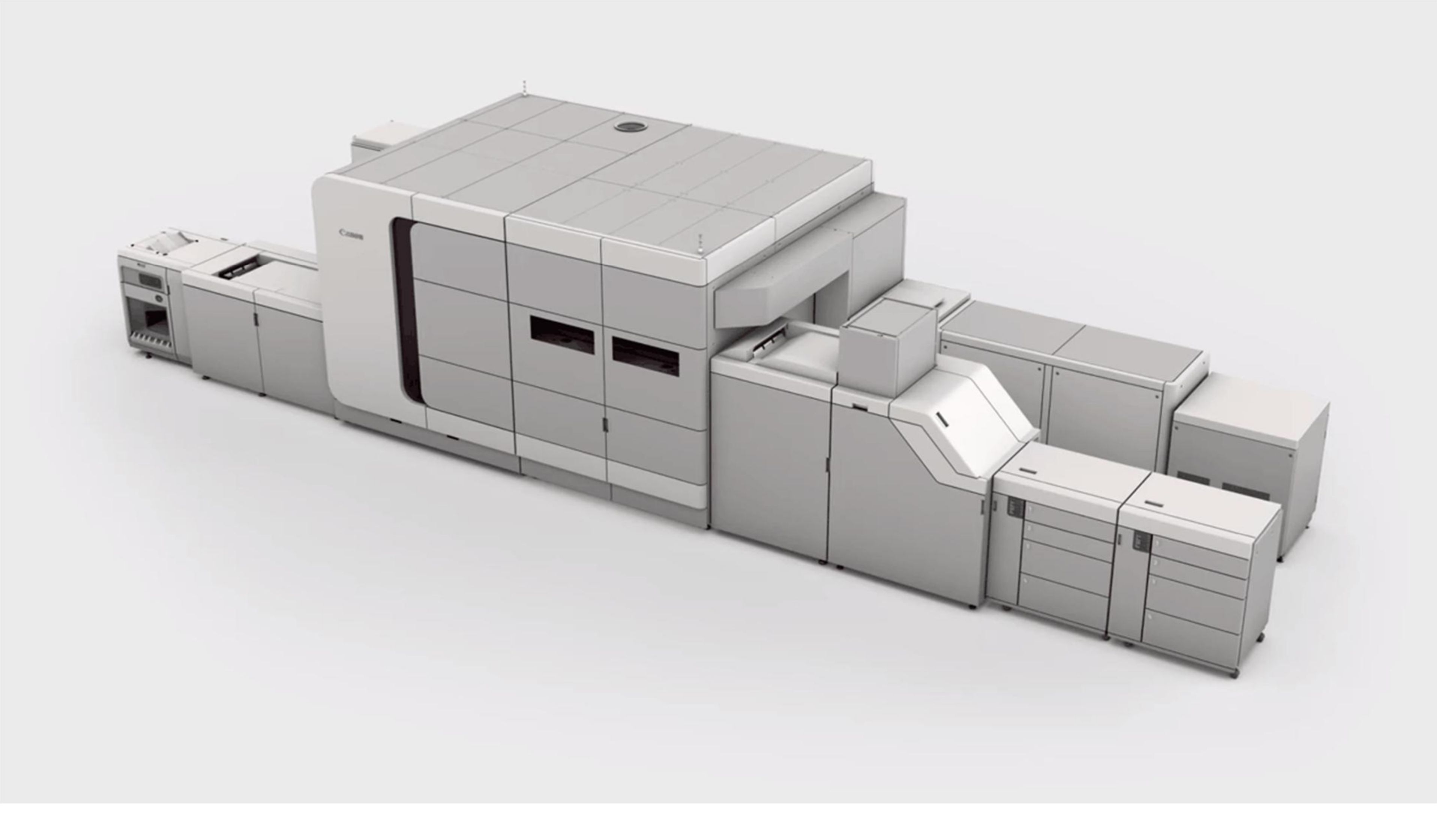


	Time ms	Pos mm	Vel mm/s	%	Acc Pinch m/s²
1	20520	23579	1313	105	0
2	17764	20027	1274	102	0
3	17364	19527	1250	100	0
4	16964	19027	1250	100	0
5	16564	18527	1250	100	0
6	16165	17982	1550	124	0
7	15765	17362	1550	124	0
8	15365	16745	1536	123	0
9	14965	16225	1700	136	0
10	14566	15546	1700	136	0
11	14166	14871	1231	98	25
12	13766	14434	861	69	20
13	13366	14163	675	54	0
14	12967	13893	675	54	0
15	12567	13623	675	54	0
16	12167	13353	675	54	0
17	11767	13084	675	54	0
18	11368	12814	675	54	0
19	10968	12544	675	54	0
20	10568	12274	675	54	0
21	10168	12004	675	54	0
22	9769	11735	675	54	0
23	9369	11465	675	54	0
24	8969	11195	675	54	0
25	6112	7970	1313	105	0
26	3356	4368	1274	102	0
27	2956	3863	1250	100	0
28	2556	3364	1250	100	0
29	2156	2864	1250	100	0
30	1757	2371	1301	104	0
31	1357	1758	1550	124	0
32	957	1139	1543	123	0
33	557	608	822	66	0
34	158	189	1200	96	0

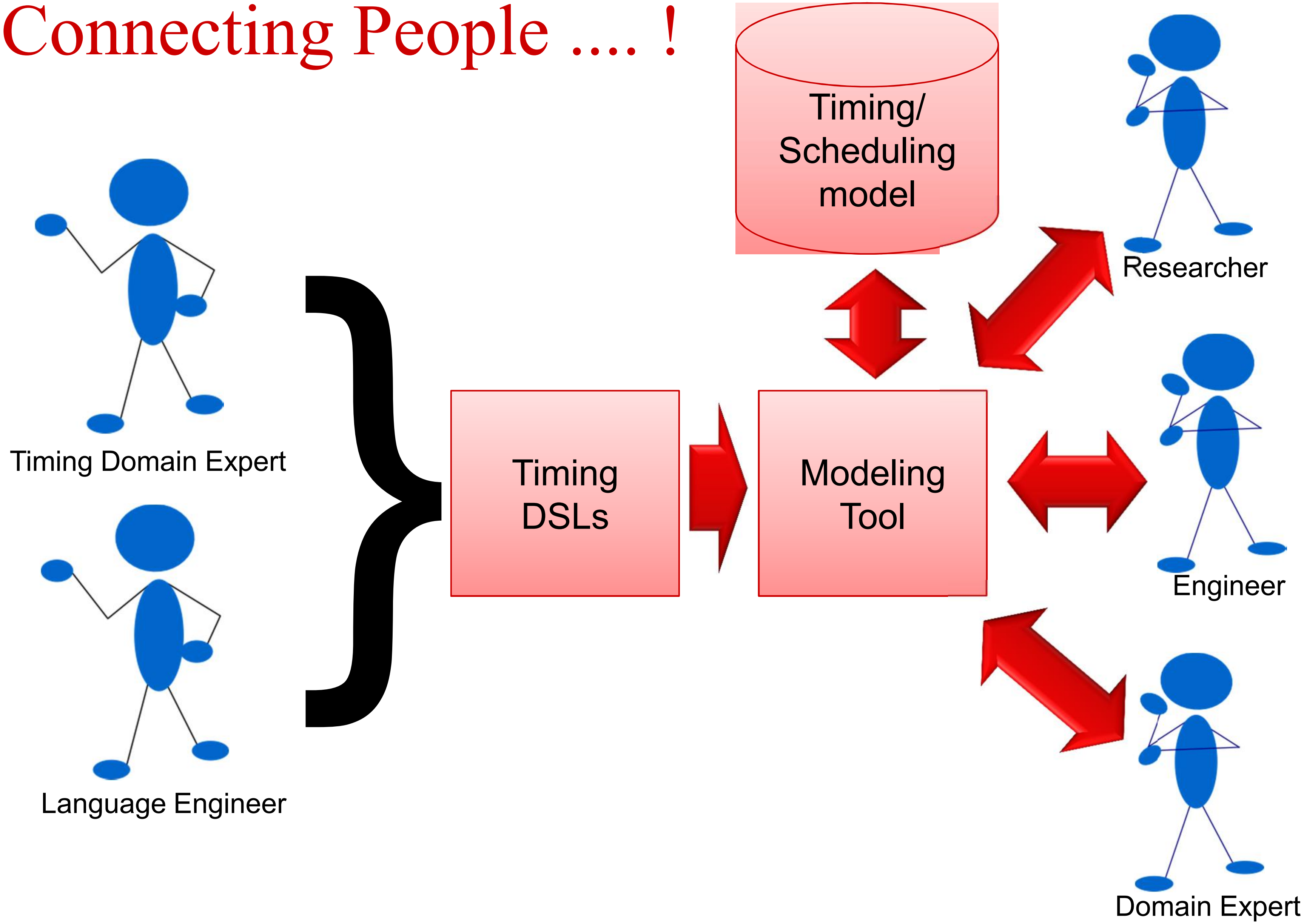
Timing combined with 3D-CAD

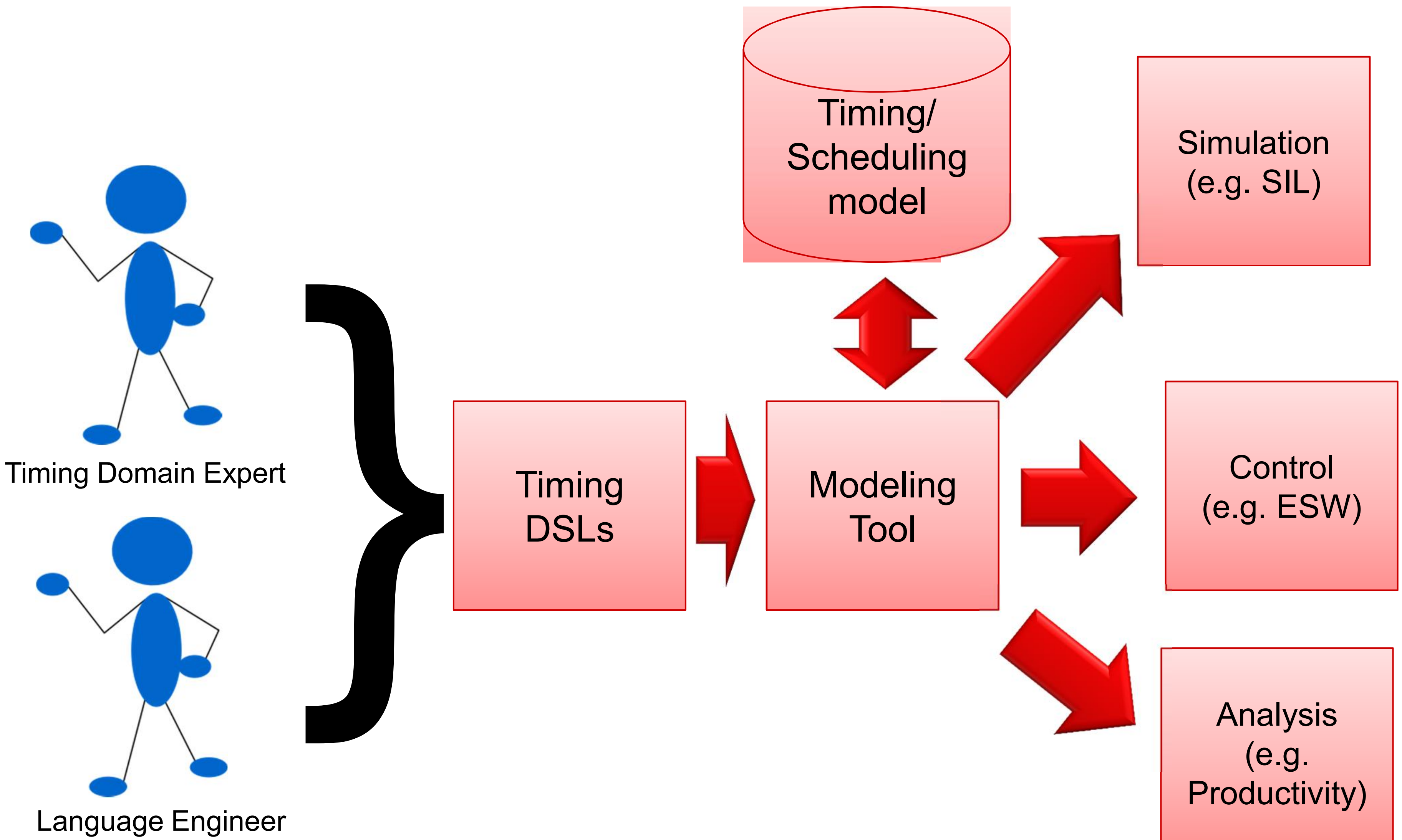


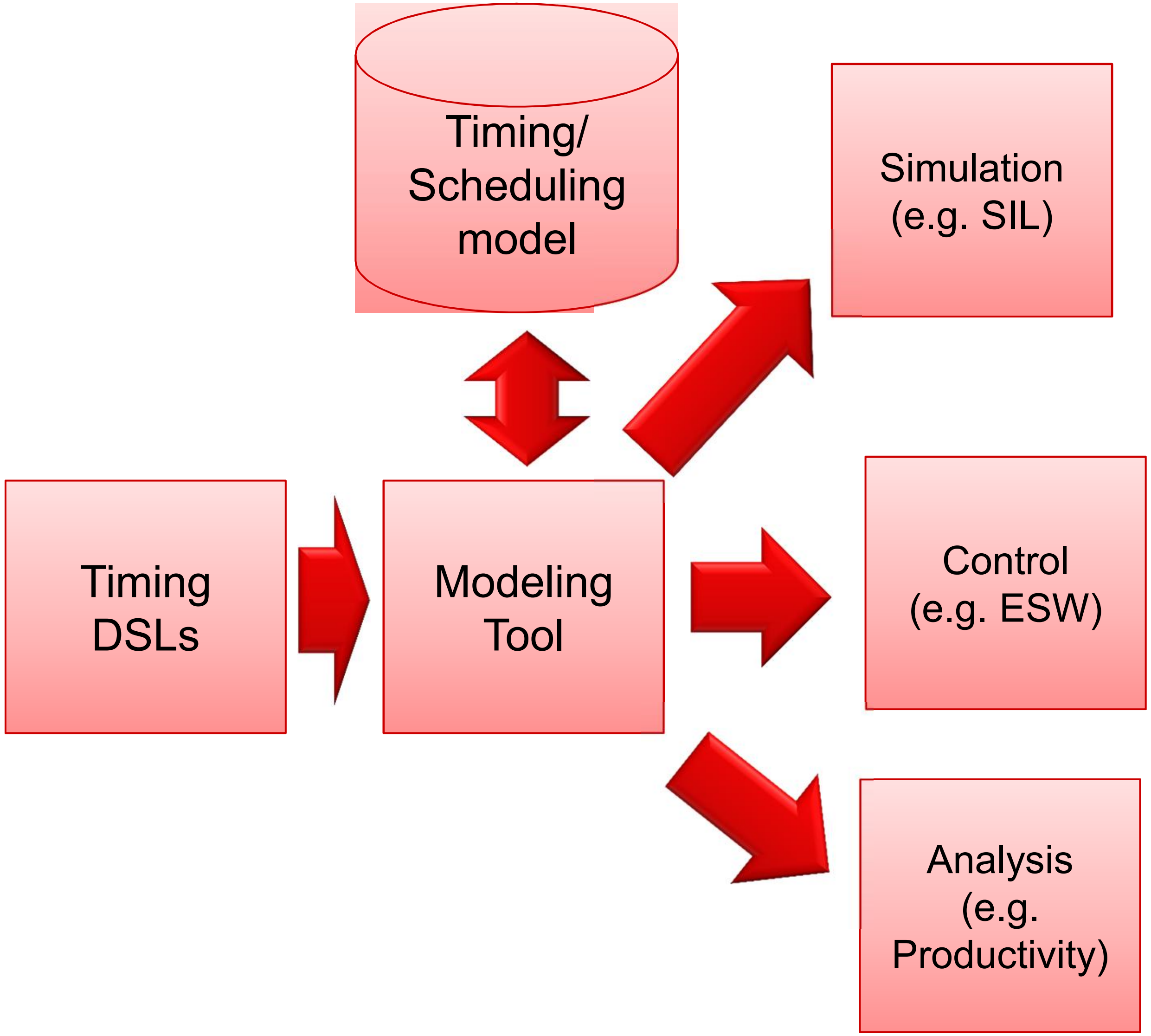
Rendered including print job

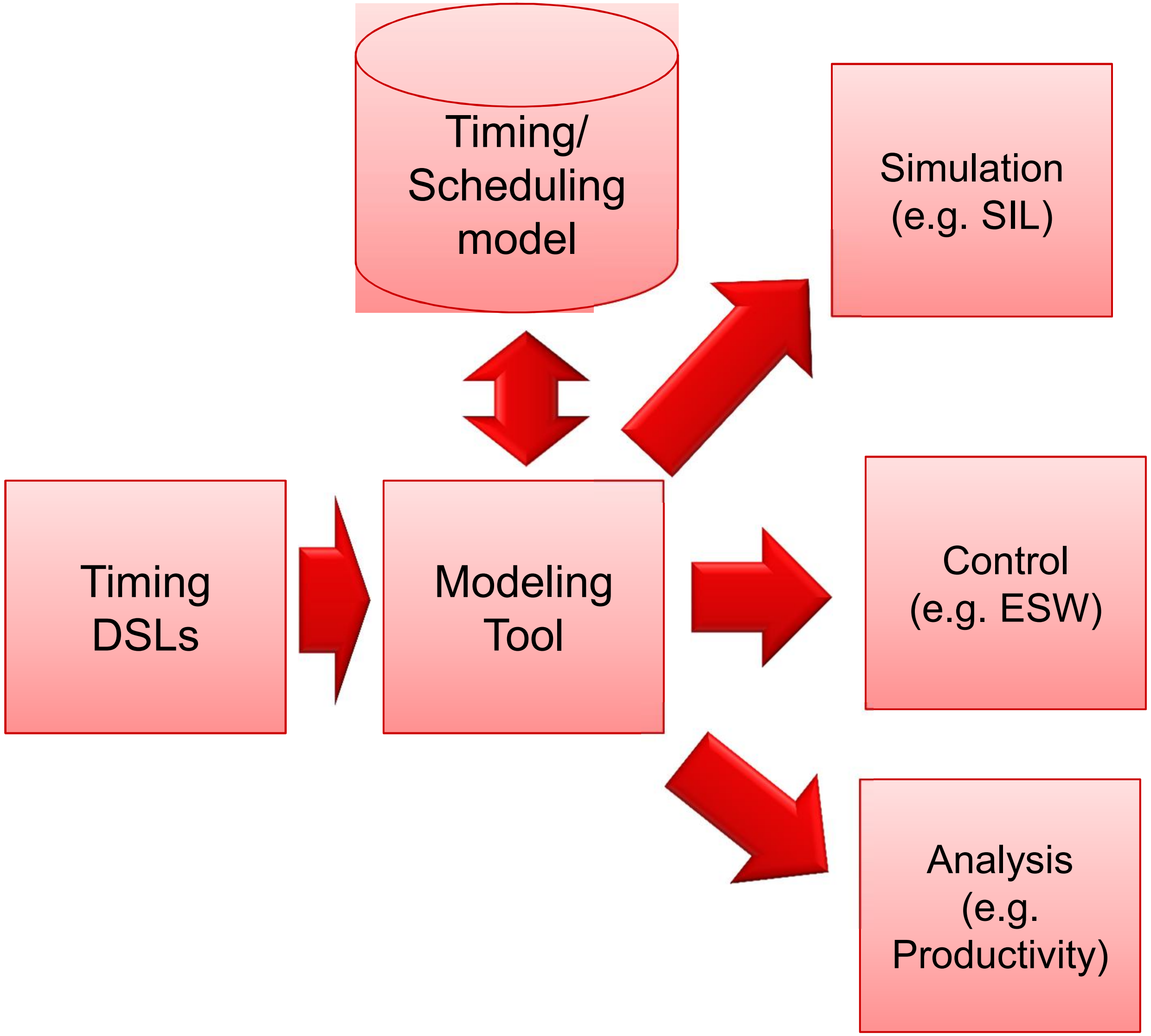


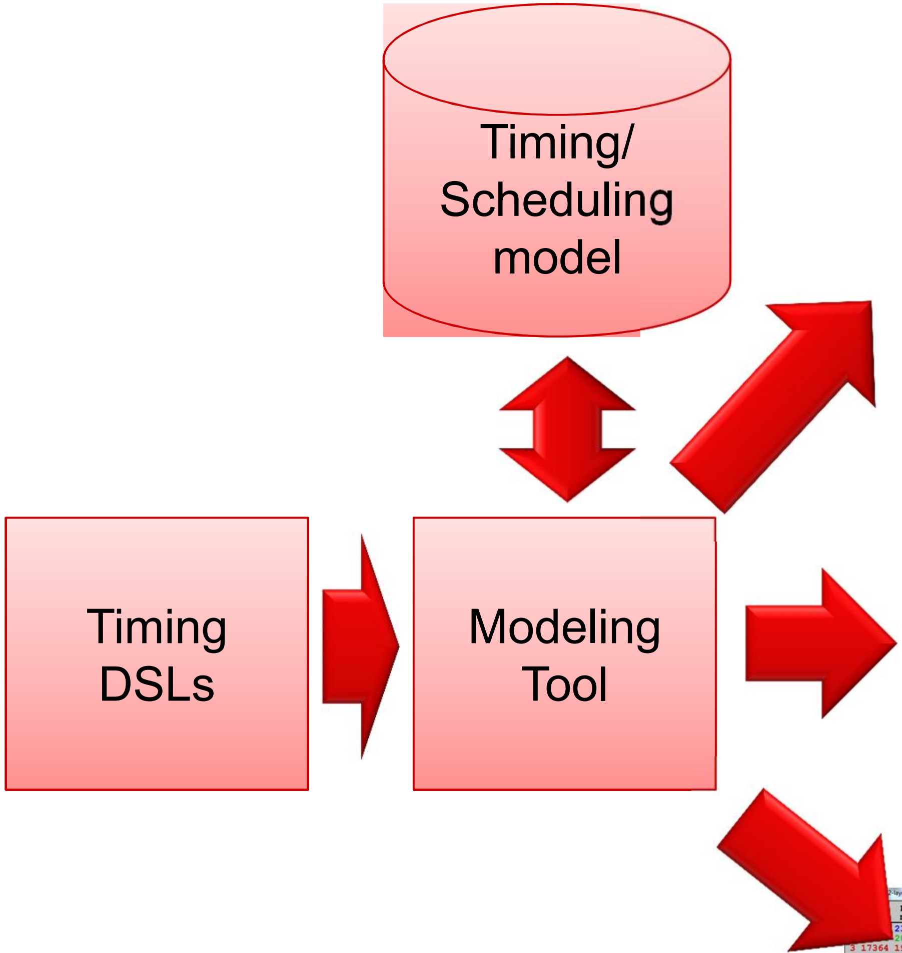
Connecting People .... !

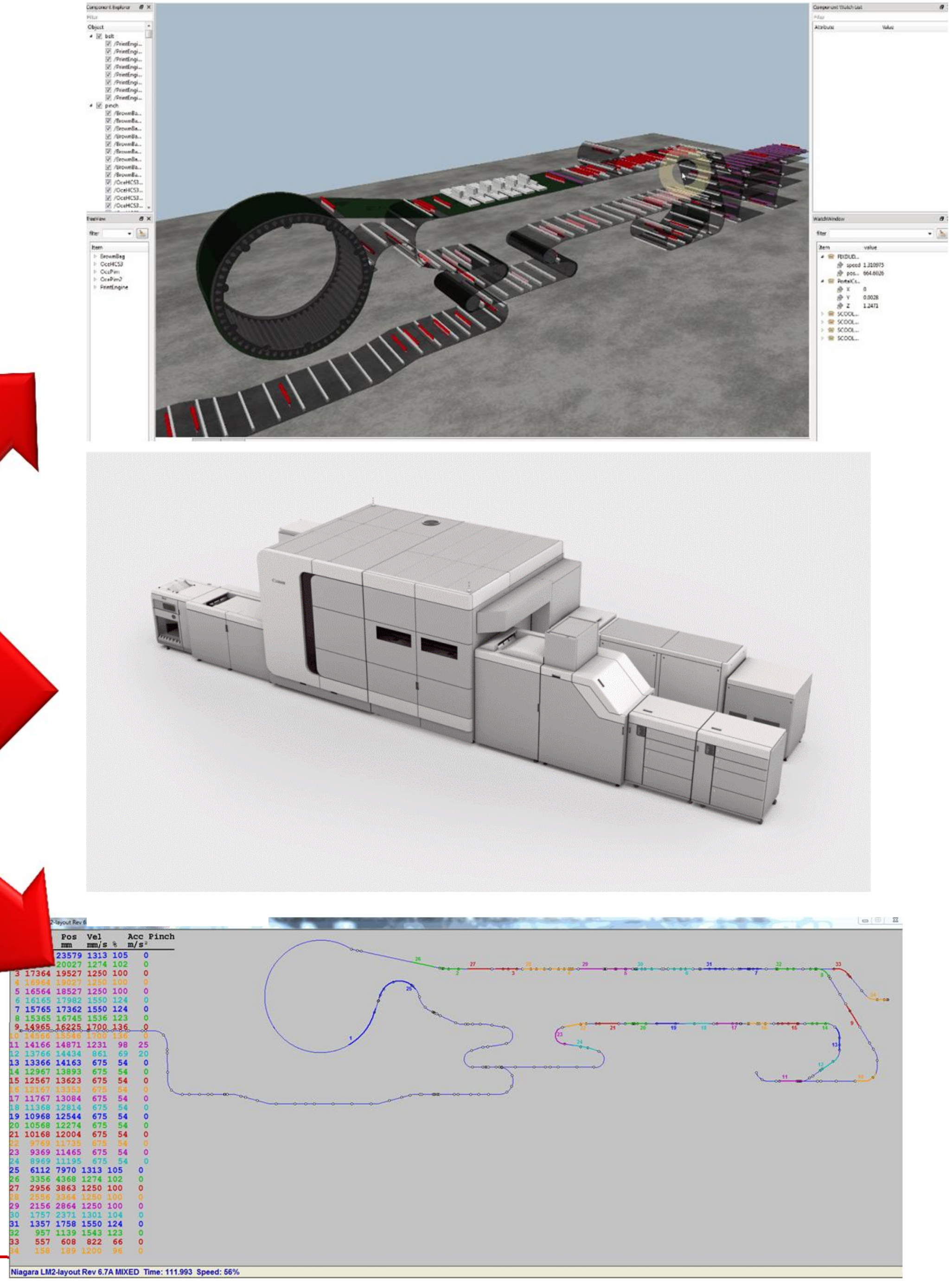



Timing/
Scheduling
model
Simulation
(e.g. SIL)
Timing Domain Expert
Timing
DSLs
Modeling
Tool
Control
(e.g. ESW)
Language Engineer
Analysis
(e.g.
Productivity)


Timing/
Scheduling
model
Simulation
(e.g. SIL)
Timing
DSLs
Modeling
Tool
Control
(e.g. ESW)
Analysis
(e.g.
Productivity)


Timing/
Scheduling
model
Simulation
(e.g. SIL)
Timing
DSLs
Modeling
Tool
Control
(e.g. ESW)
Analysis
(e.g.
Productivity)


Timing/
Scheduling
model
Timing
DSLs
Modeling
Tool


Niagara LM2-layout Rev 6.7A MIXED Time: 111.993 Speed: 56%

```
Job Specification NiagaraSchedule for DL908490Mitsubishi x 70:
  Orient sheet with LongEdgeFirst
  Follow route Simplex
  Adhere to timing behavior NiagaraTimingBehavior
  Use scheduler MONOTONIC with period 0.9 s
  Use profile recipe: <choose optional profile recipe>
```

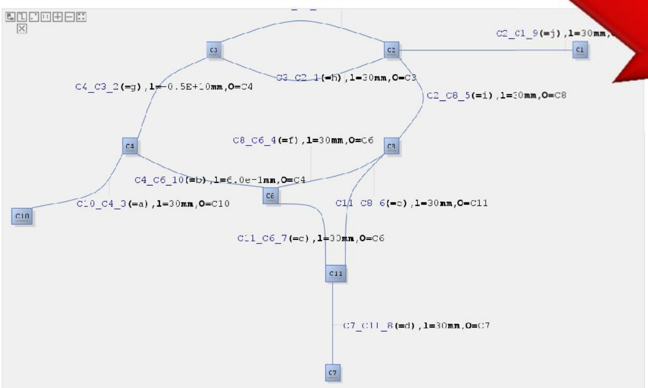
```
double decelerationDistance(double vNew, double vOld, double a) {
  return | (vNew^2 - vOld^2) / (2 * a) |;
} decelerationDistance (function)
```

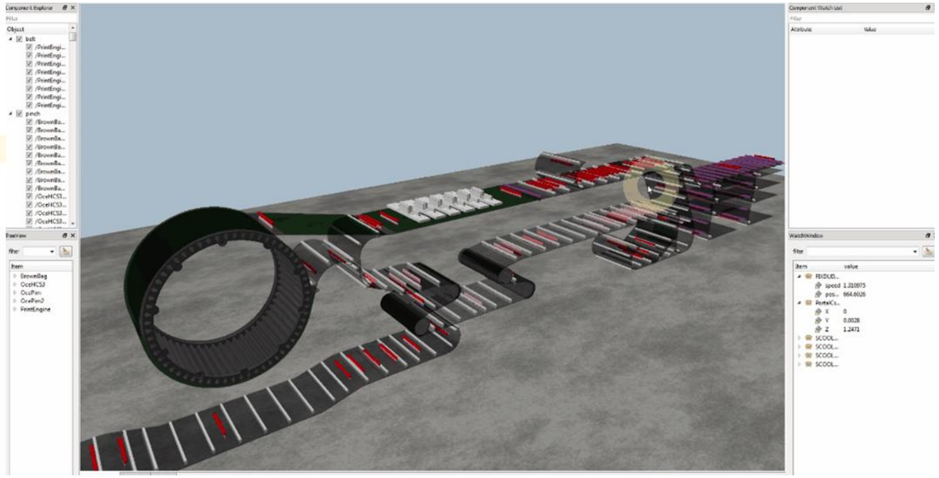
```
Action Turn ( speed vInitial, speed vMax ) using <choose optional parameter calculation model>
{
  Entered with speed vInitial
  Speed is vMax with LeadingEdge starting at at InterfacePoint1107_InterfacePoint1115_10_POMEBUSXLSE
  Stopped with TrailingEdge at InterfacePoint1225_InterfacePoint1365_14_FIXOURUSRPI + 10 mm and waite
  Flip Edges
  Accelerated to speed vMax
  Speed is vInitial with LeadingEdge starting at at InterfacePoint1107_InterfacePoint11
                                   FIXDEXUERRORFLAP_InterfacePoint156_4_FIXDEXUPI1 (=758
}
```

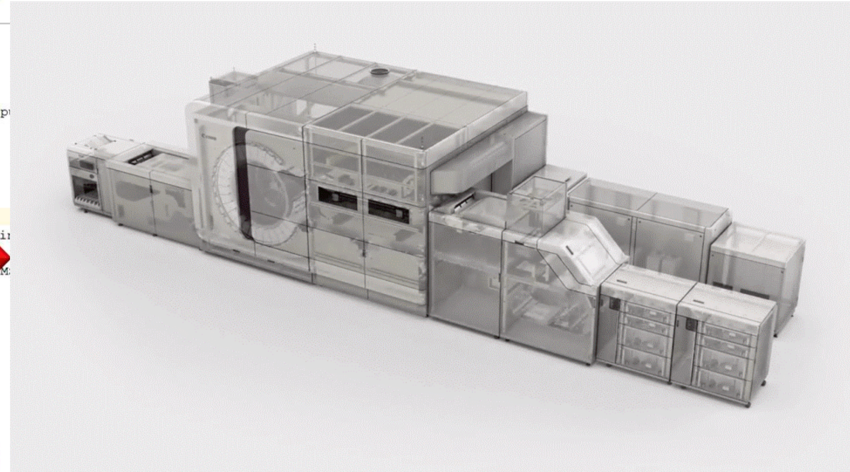
```
Routelist NiagaraRoutes for NiagaraPaperpath
Routes
  Route Simplex = ( [input - process - fixation - toOutputOrRet
  Route Duplex = (
      [input - process - fixation - toOutputOrReturn - return - pro      - fixation - toOutpu
Sectors
  Sector input : ‖InterfacePoint2473, TMJOIN, InterfacePoint2598, InterfacePoint579‖
  Sector process : ‖InterfacePoint579, InterfacePoint2630, InterfacePoint1‖
  Sector fixation : ‖InterfacePoint1, FIXDEXUERRORFLAP, InterfacePoint156‖
  Sector bypass : ‖InterfacePoint1, InterfacePoint156‖
  Sector toOutputOrReturn : ‖InterfacePoint156, SCOOLSCEUERRORFLAP, InterfacePoin    ‖
  Sector output : ‖InterfacePoint1588, InterfacePoint1775, InterfacePoi
      InterfacePoint1107, InterfacePoint1115‖
  Sector return : ‖InterfacePoint1588, InterfacePoint2068, TMSRUDEF, In
      InterfacePoint238, TMJOIN, InterfacePoint2598, InterfacePoint57
Entries
  entry entry connector : InterfacePoint2473
Destinations
  destination exit connector : InterfacePoint1224
First turn point:
Turn point tolerance (amount of extra segment after a turn point)
  0 mm
```

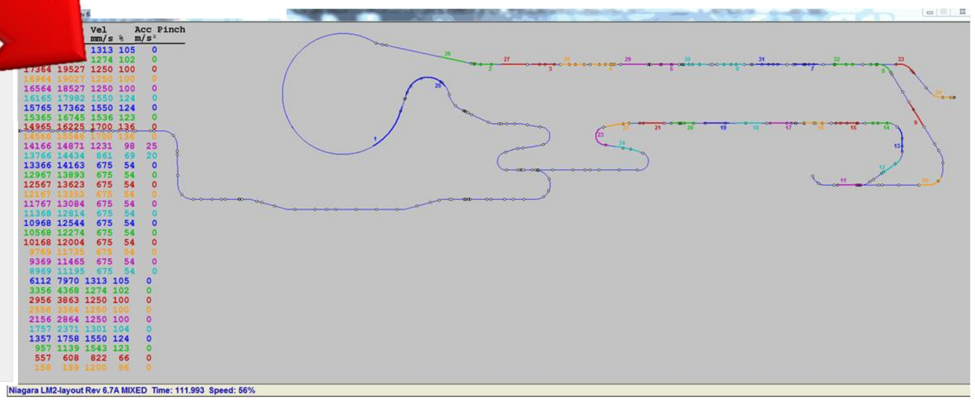
Timing DSLs

```
Layout ExamplePaperpath
```

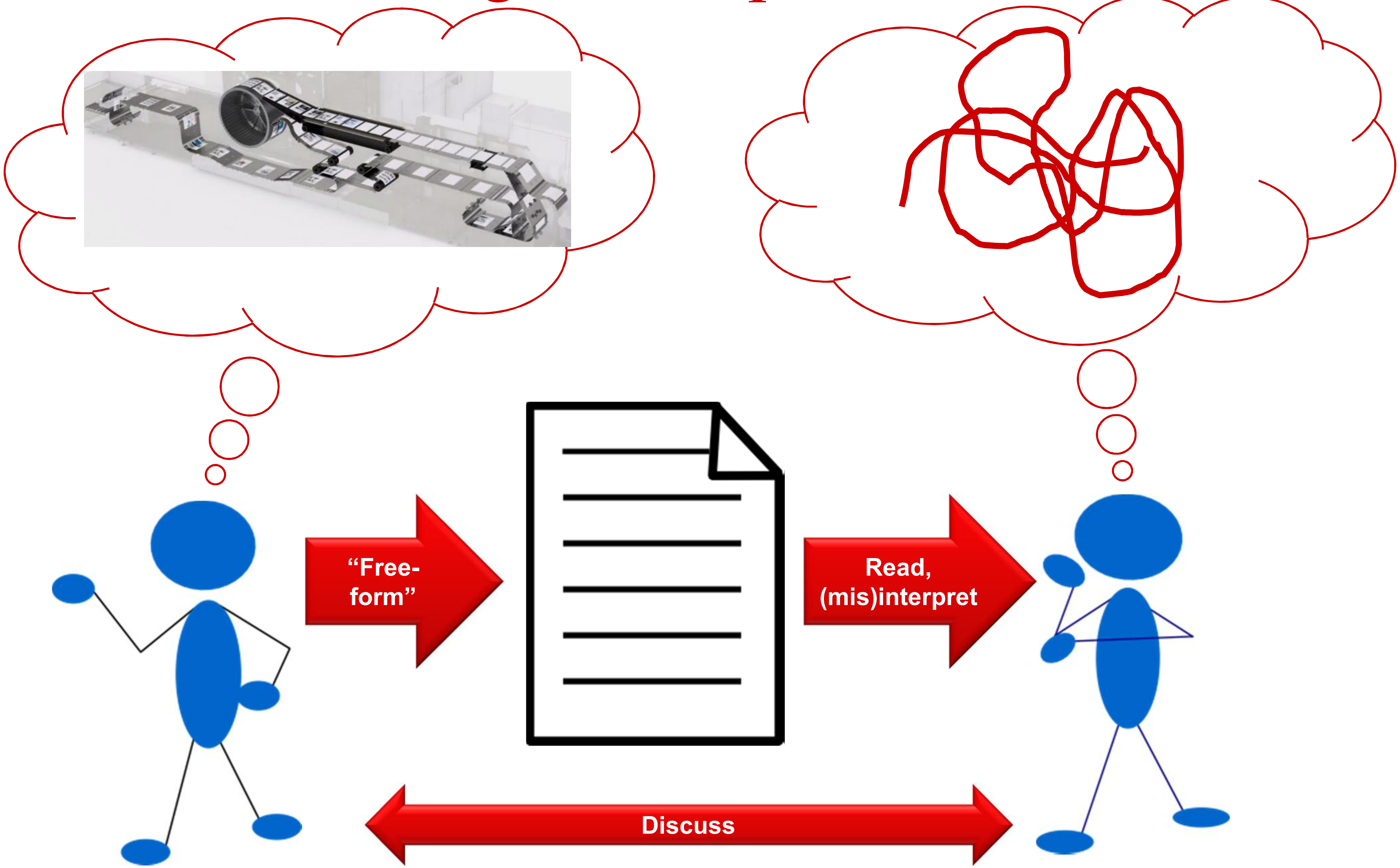


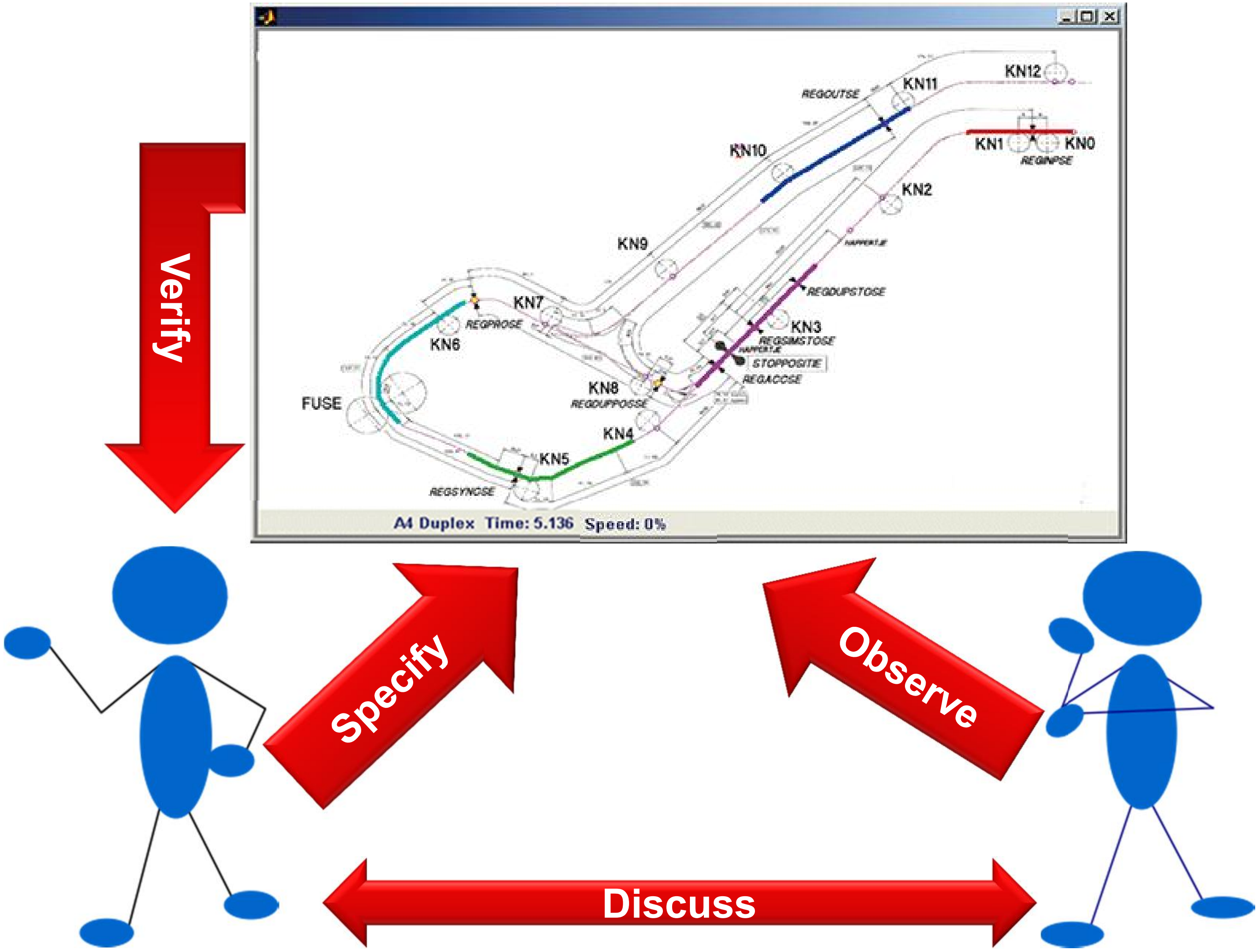


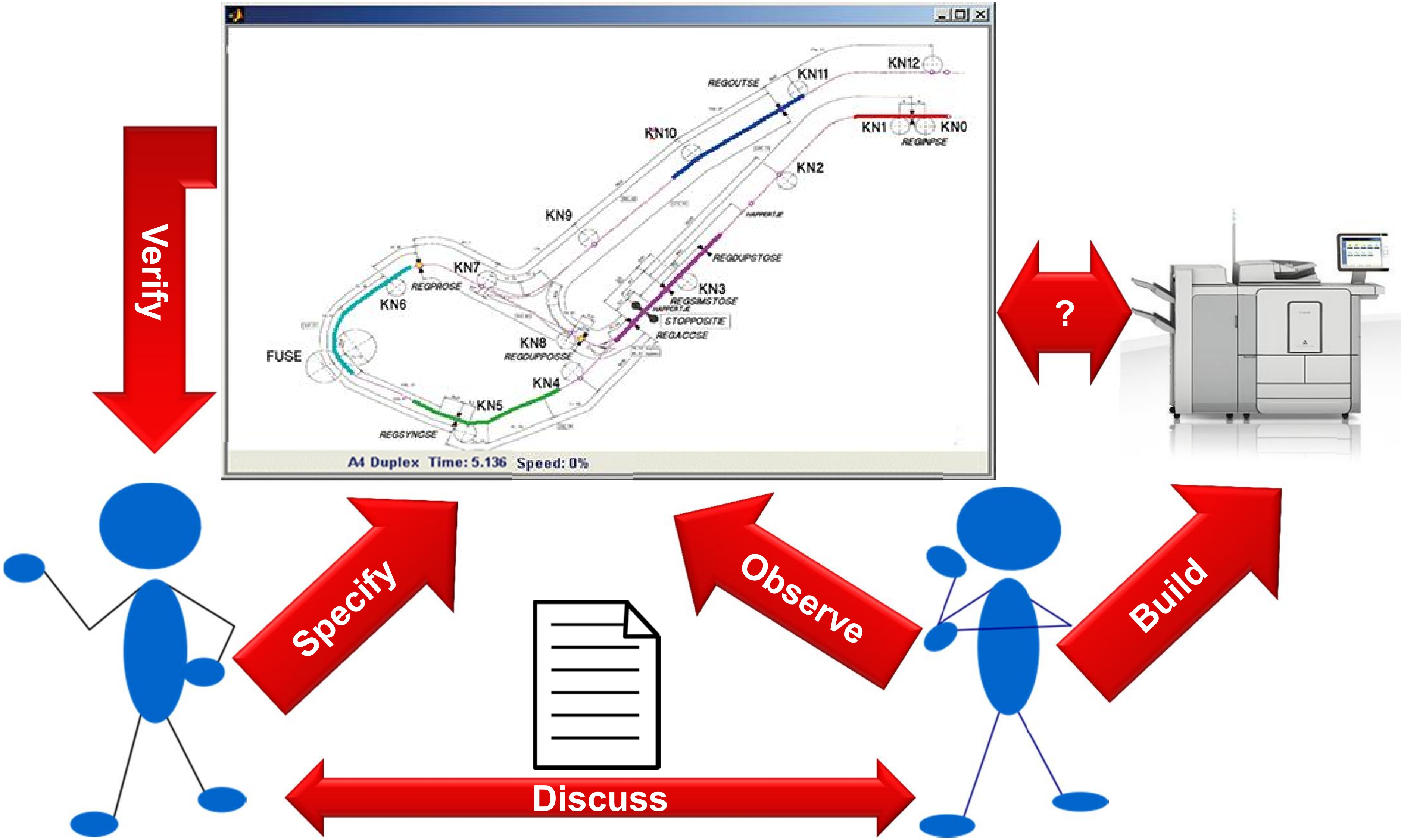


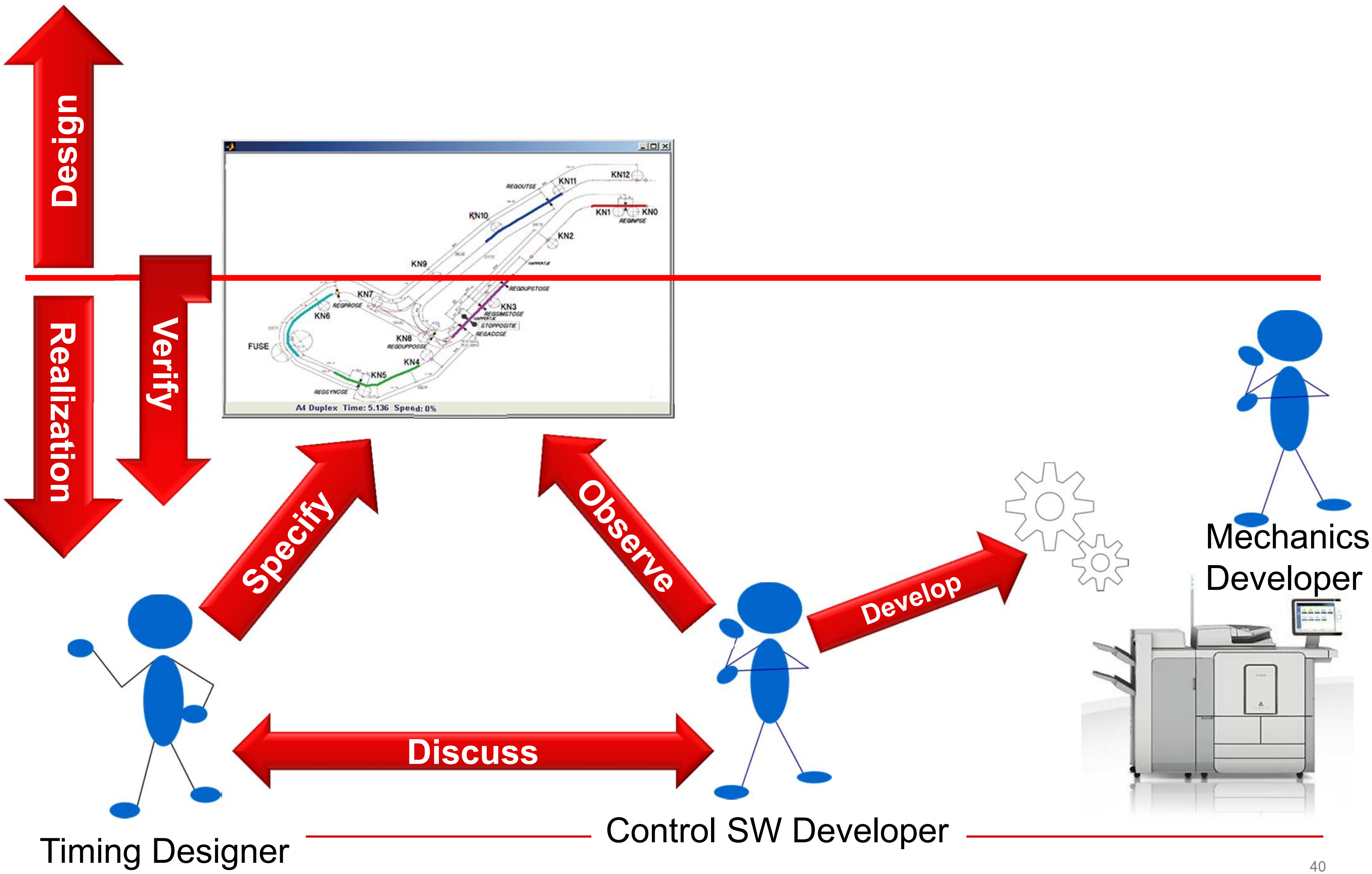


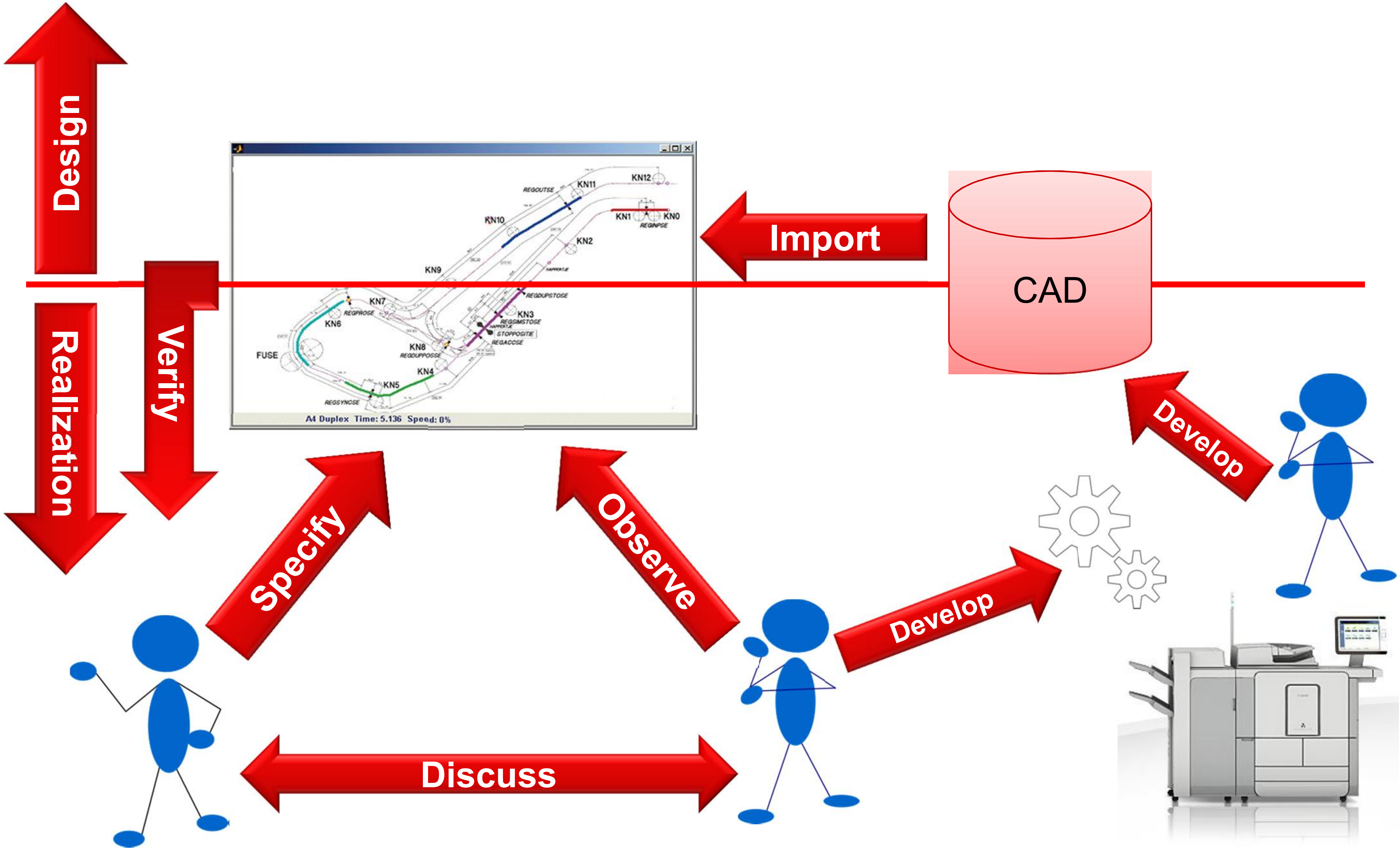
And Connecting Development Phases ….

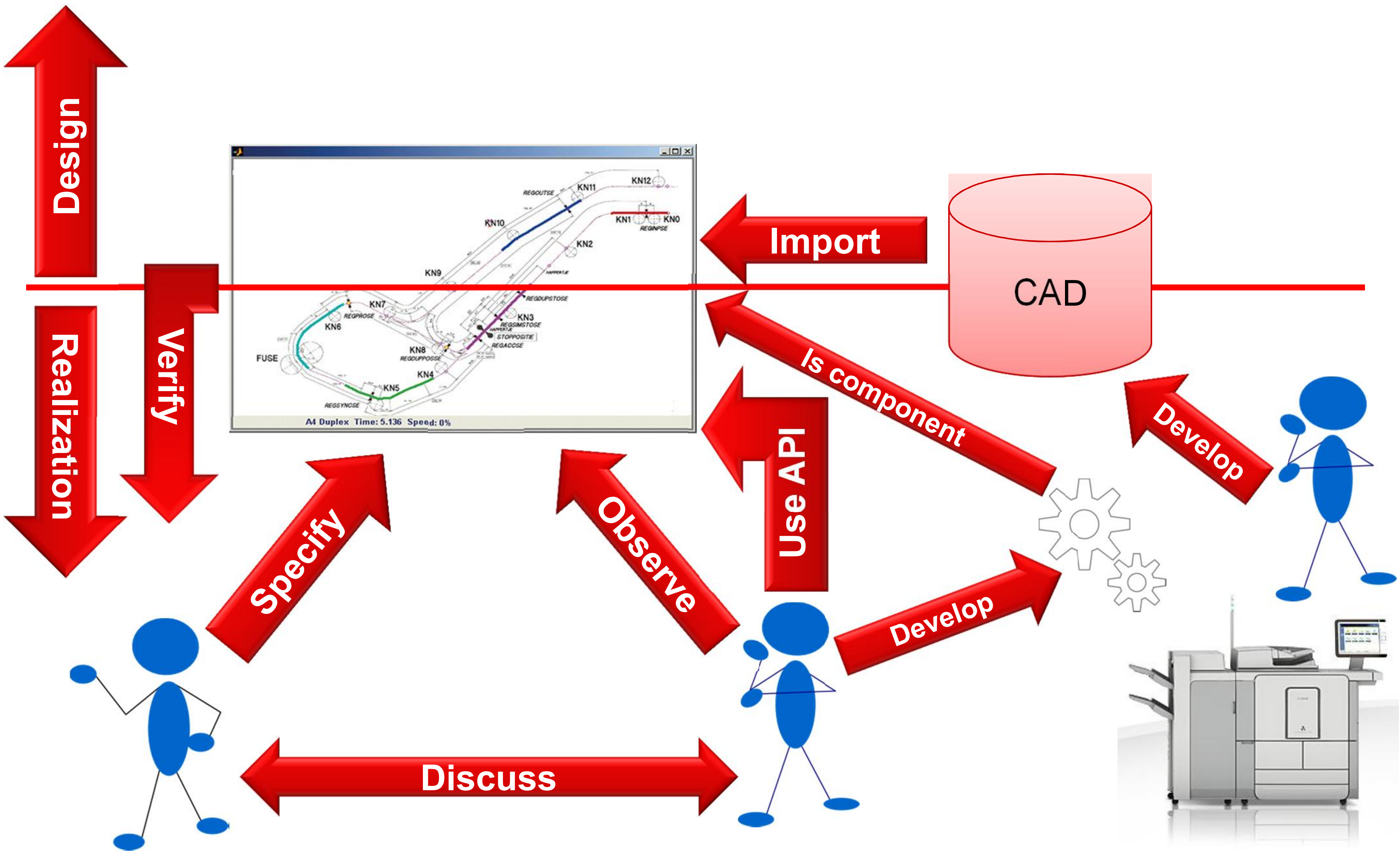



Verify
KN12
KN11
REGOUTSE
KN10
KN1
KN0
REGINPSE
KN2
KN9
REGDUPSTOSE
KN7
REGPROSE
KN3
REGSIMSTOSE
KN6
STOPPOSITIE
REGACCSE
KN8
REGDUPPOSSE
FUSE
KN4
KN5
REGSYNCSE
A4 Duplex Time: 5.136 Speed: 0%
Specify
Observe
Discuss


Verify
Specify
Observe
Build
Discuss
?
KN12
KN11
REGOUTSE
KN10
KN1
KN0
REGINPSE
KN2
KN9
REGDUPSTOSE
KN7
REGPROSE
KN6
KN3
REGSIMSTOSE
STOPPOSITIE
REGACCSE
KN8
REGDUPPOSSE
FUSE
KN4
KN5
REGSYNCSE
A4 Duplex Time: 5.136 Speed: 0%


Design
Realization
Verify
A4 Duplex Time: 5.136 Speed: 0%
Specify
Observe
Develop
Discuss
Mechanics
Developer
Timing Designer
Control SW Developer

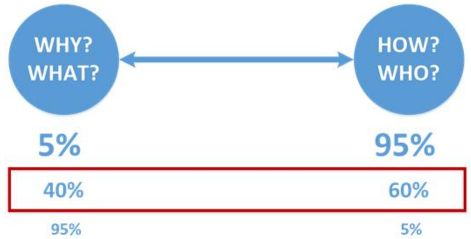

Design
Realization
Verify
Import
CAD
Specify
Observe
Develop
Develop
Discuss
A4 Duplex Time: 5.136 Speed: 0%
FUSE
KN0
KN1
KN2
KN3
KN4
KN5
KN6
KN7
KN8
KN9
KN10
KN11
KN12

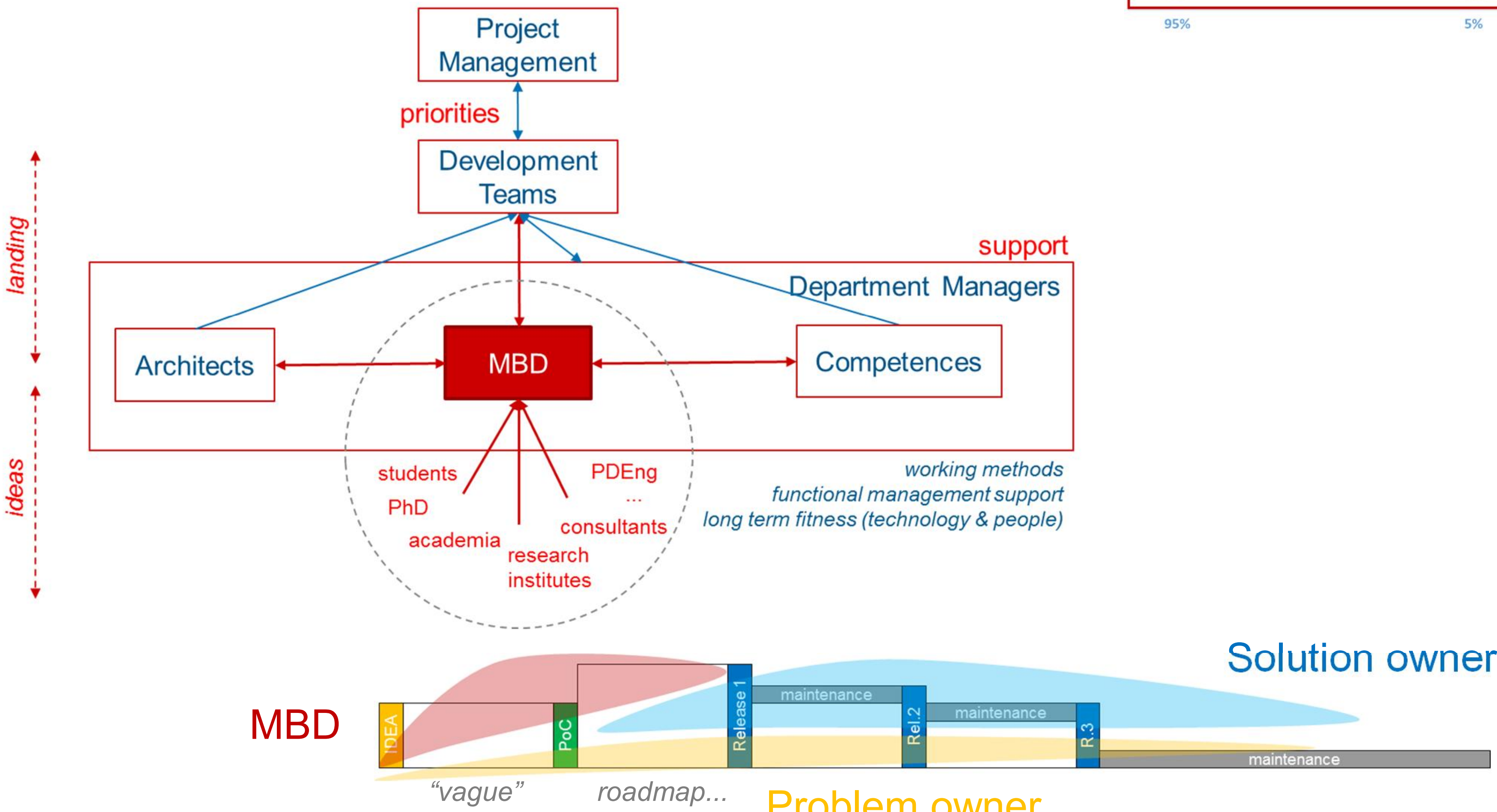

Design
Realization
Verify
A4 Duplex Time: 5.136 Speed: 0%
Import
CAD
Is component
Use API
Develop
Specify
Observe
Develop
Discuss

Make it "business as usual"

Introduce organization and process...

MBD supports the transition to:





Match MBD and product planning....

Model Based Development ?

Dare to dream !

Questions?

océ

A CANON COMPANY