

# SUPPORT TOOL FOR SYSTEM ARCHITECTING AND MODEL-BASED ENGINEERING

AN INDUSTRIAL CASE STUDY



 TU Delft Andres A. Alvarez Cabrera

 TU Delft Maarten Foeken

UNIVERSITY OF TWENTE. Krijn Woestenenk

 oce Guy Stoot


 oce Chitrlekha Pillai



# INTRODUCTION

## SUPPORT TOOL FOR SYSTEM ARCHITECTING AND MODEL-BASED ENGINEERING

---

- Motivation
- Language and Framework
- Case Study: 
- Architecture model
  - Architecture Capture
  - Model generation (structural)
  - Model generation (simulation)
- Preliminary Results
- Conclusion
- Acknowledgements

# MOTIVATION

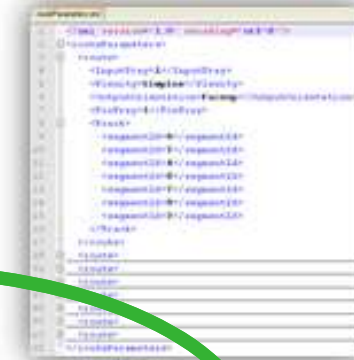
## EXAMPLE DESIGN PROCESS

Subsystem = Paper flow path

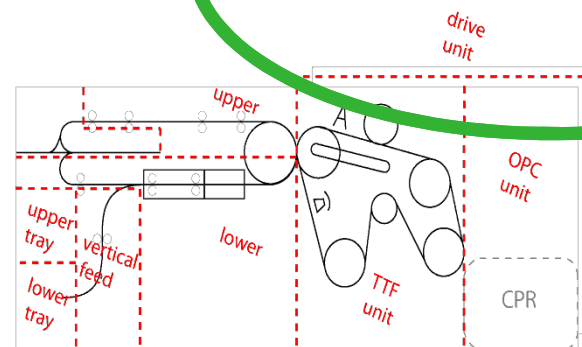
- Design of a single (sub)system
- “Divide and conquer” among several teams
  - Different models/language (PLM, Matlab, UML/SysML, STEP, etc.)
- Different views/partitions (management, assembly, control, etc.)



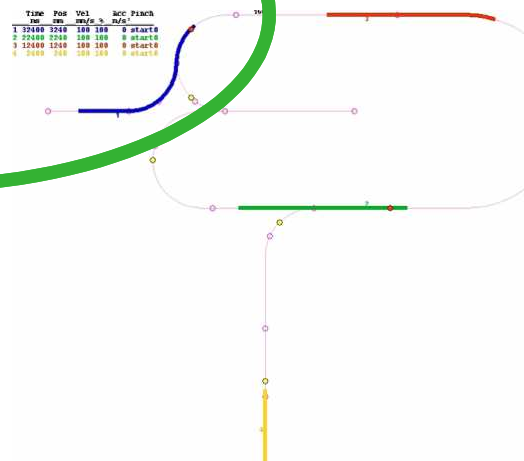
3D model



Custom process model



Conceptual Layout



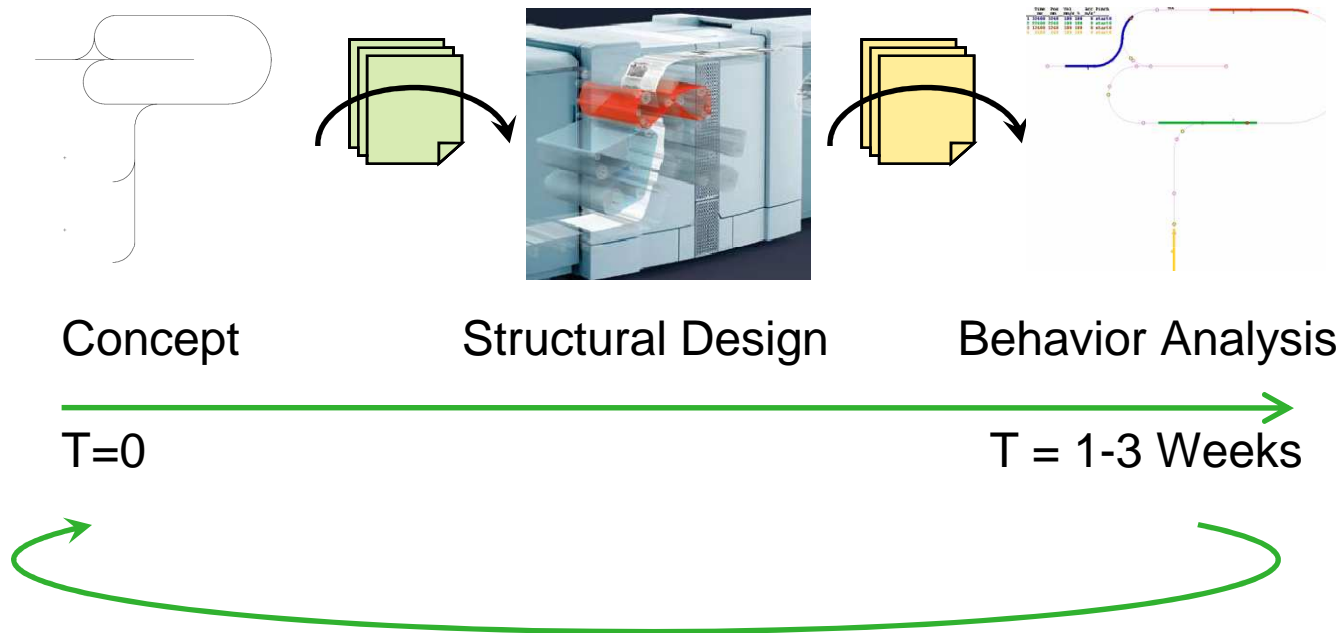
Custom simulation model

Iterate

# MOTIVATION

## EXAMPLE DESIGN PROCESS

- Design is Iterative, Domains are Separated, Time is consumed



- Hand-Off of information is often the same.
- Meetings and documents
- Architectures do not change often K1
- Design Process does not change often
- Tooling does not change often

While Design != Good

While Design Decisions Change (10 concepts)

## Dia 4

---

**K1**      architectuur verandert wel in de ogen van de architect:

altijd cold/warm process/ paper path etcetera

Woestenenk; 26-10-2010

# MOTIVATION

## WHAT IS MISSING?

---

- Missing? A **common language** to describe the system for **support of the design process**
- Model the **shared information** between the designers and tools
  - What information is shared?
- Advantages: **Integration, goal overview, consistency, consensus**
- Additionally: support **automation** of **information flow/transformation**
  - Meaning: (partially) automate design process

K3

## Dia 5

---

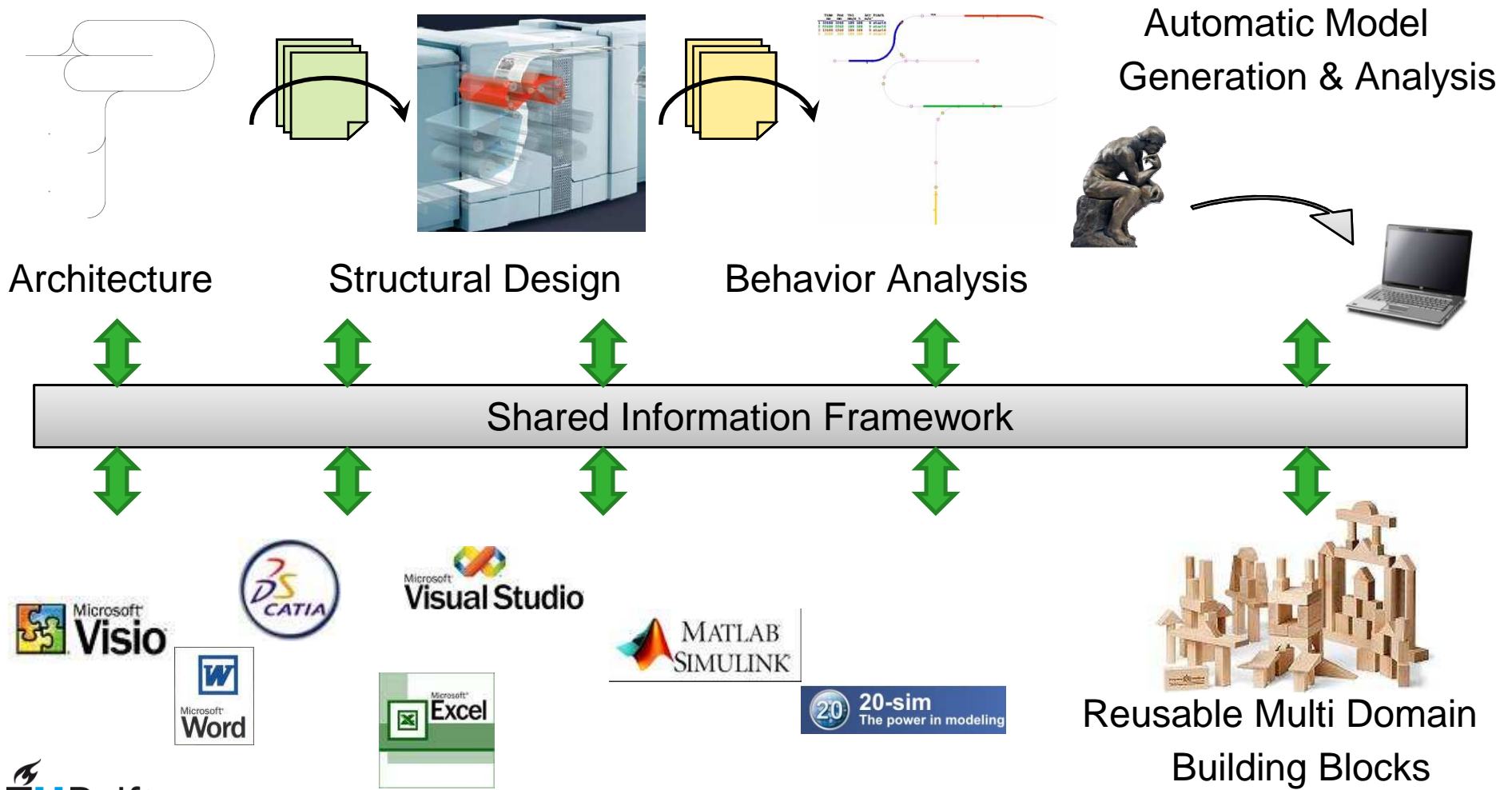
**K3**

generate solutions for multiple total aspects concurrently

Woestenenk; 26-10-2010

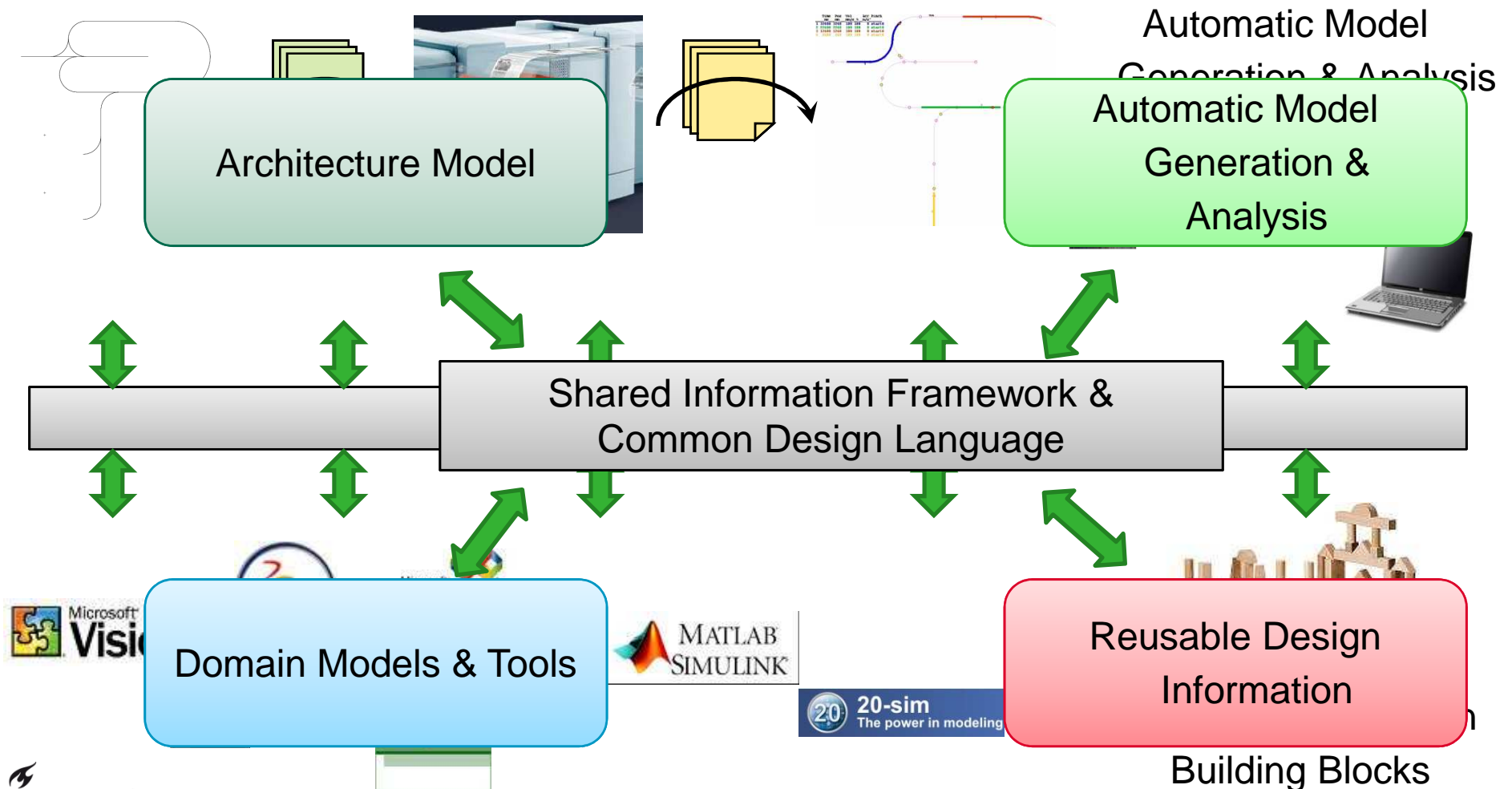
# BASIC FRAMEWORK

## REUSABLE DESIGN PROCESSES AND AUTOMATION



# BASIC FRAMEWORK

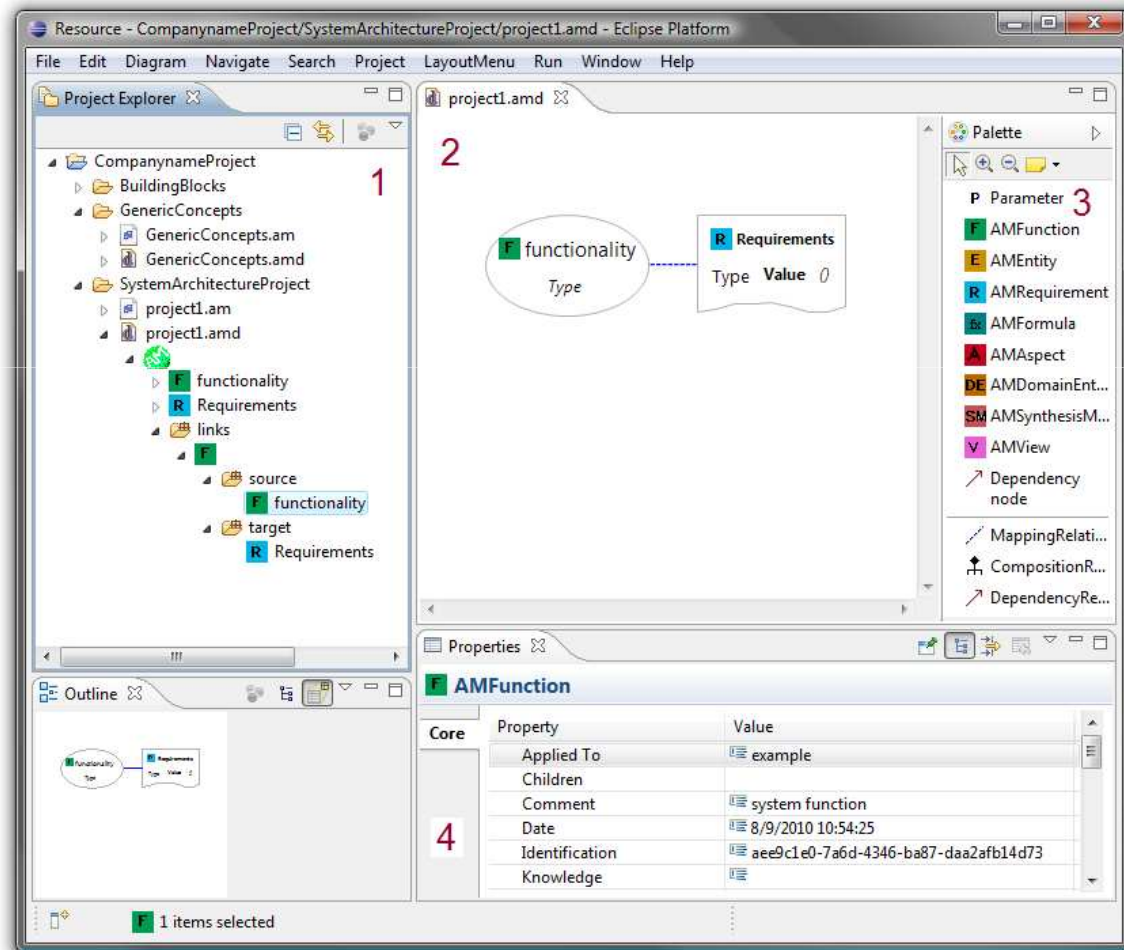
## REUSABLE DESIGN PROCESSES AND AUTOMATION



# BASIC FRAMEWORK

## SUPPORT TOOL FOR SYSTEM ARCHITECTING AND MODEL-BASED ENGINEERING

- Eclipse Graphic Modeling Framework: AMTool
- Extendable for Knowledge Based Engineering & Model Based Engineering
- Growing Popularity in Industry
- Various case studies



# BASIC FRAMEWORK

COMMUNICATE ACROSS DISCIPLINES IN A SINGLE LANGUAGE

- Capture Information
- Machine Readable

## Architectural Concerns

to

## Domain Information

using

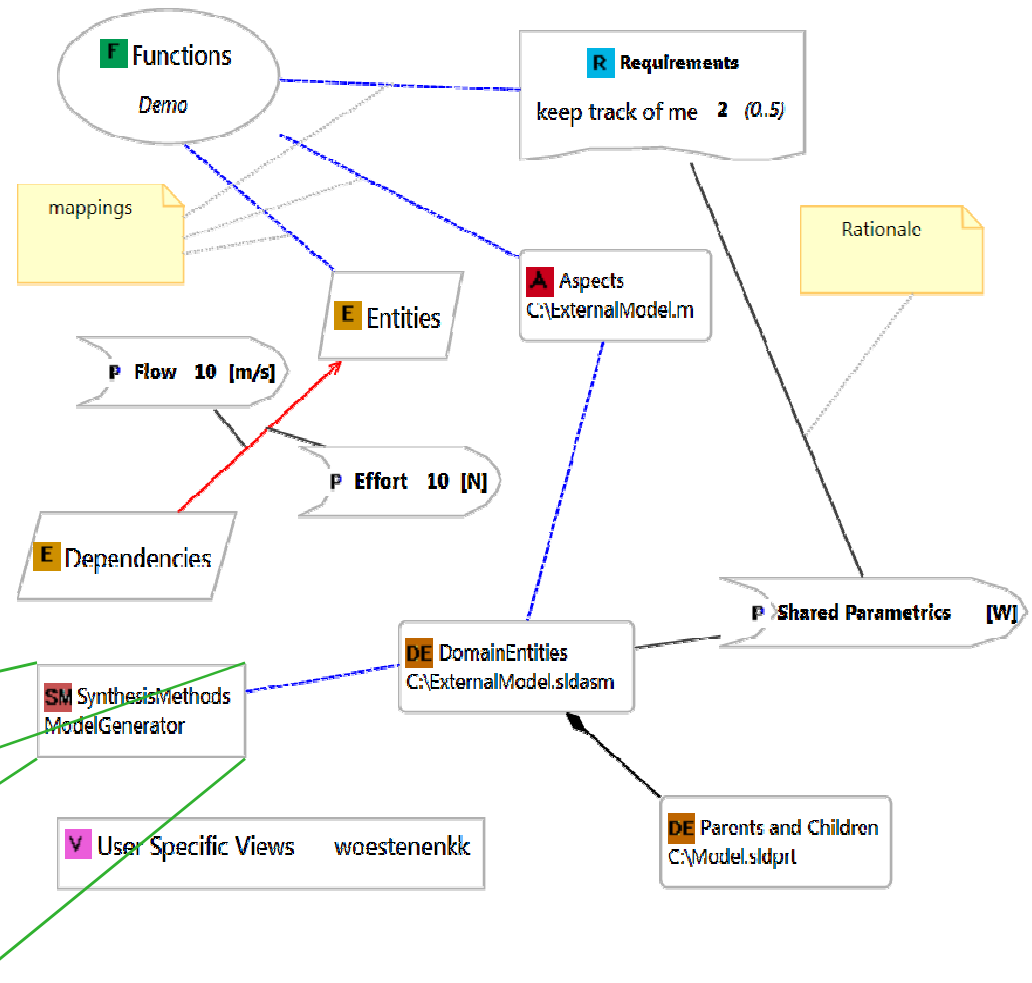
## Sets and Graphs:



and

## Basic attributes:

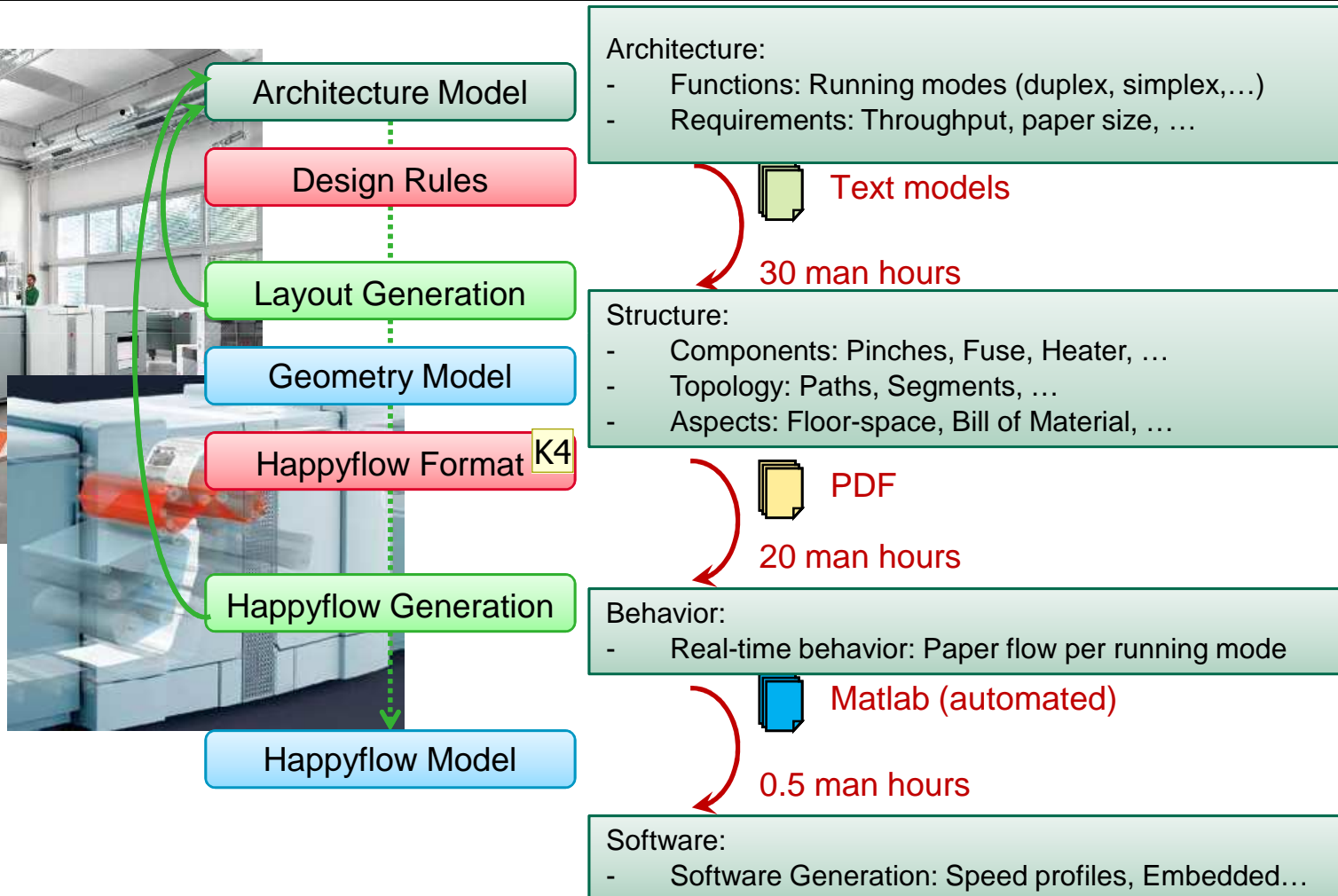
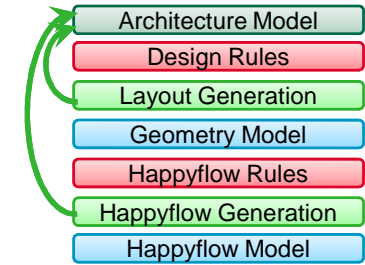
Property	Value
Applied To	Electronics
Comment	
ComposableObjects	AM Entity #BasicPart#, AM Domain Entity Electronics#BasicPart#
Date	10/6/2010 15:24:18
Dependencies In	
Dependencies Out	
Identification	245219fc-3255-45d9-9dd0-e0855a9b6ae4
Knowledge	OCESB.Electronics.SignalConnector
Map To AM Views	AM View Vanderlande Case Rationale
Name	SignalConnector
Range	
Status	
Type	SignalConnector
Unit	
User	woestenenkk
Value	
Version	1.0





# CASE STUDY AT OCÉ

## DESIGN ITERATION OF A PAPER PATH



## Dia 10

---

**K4**

rules to format!

Woestenenk; 26-10-2010

**K5**

andere vorm rules/process/model

Woestenenk; 26-10-2010



# OCÉ CASE STUDY

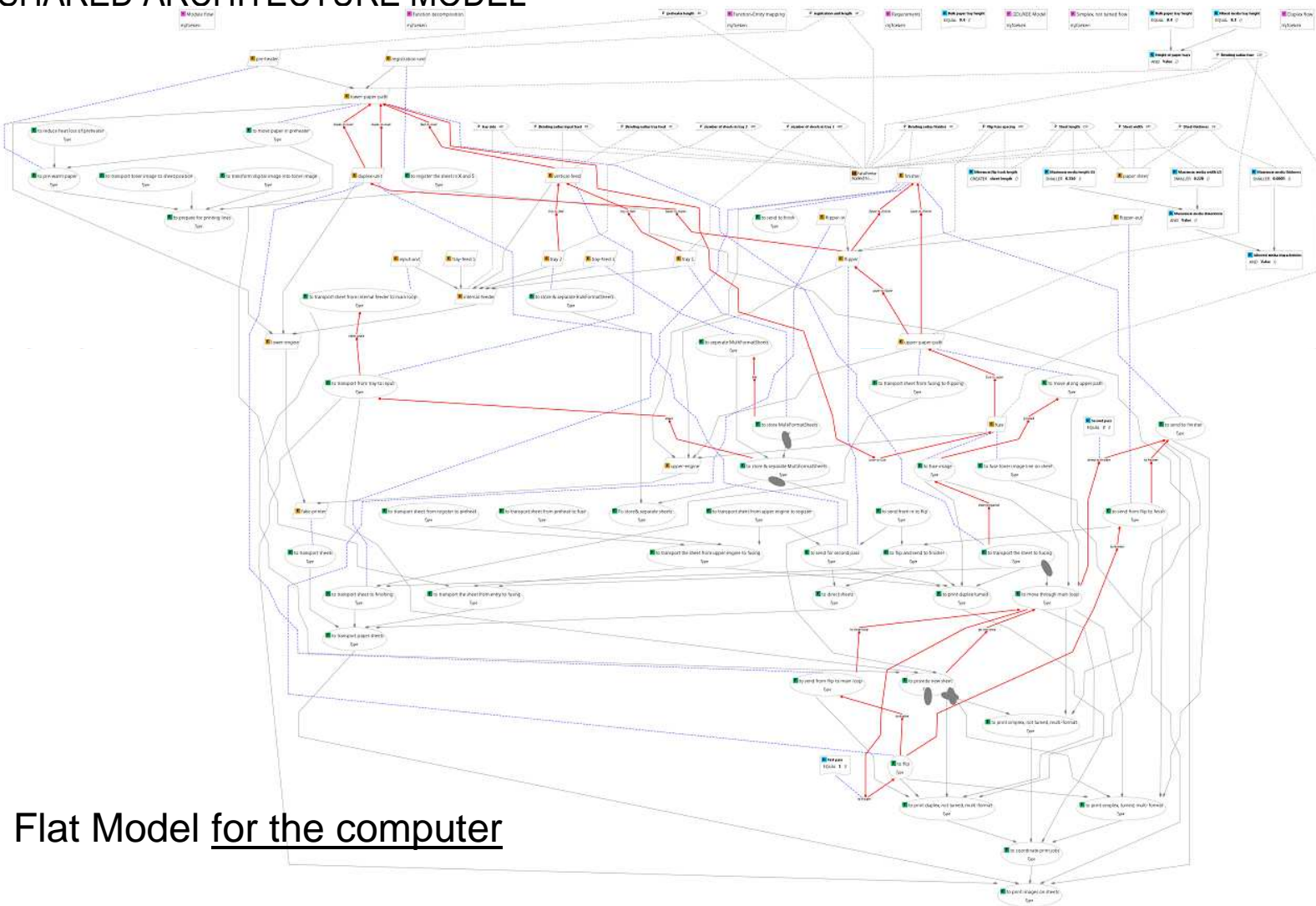
## SHARED ARCHITECTURE MODEL

- Architecture Model
- Design Rules
- Layout Generation
- Geometry Model
- Happy flow Rules
- Happy flow Generation
- Happy flow Model

Architecture:  
- Function  
- Requirement

CaseO

- Du
- Fur
- Fur
- GD
- Mo
- Rec
- Sim
- Sim



Requirements  
Decomposition

Flat Model for the computer

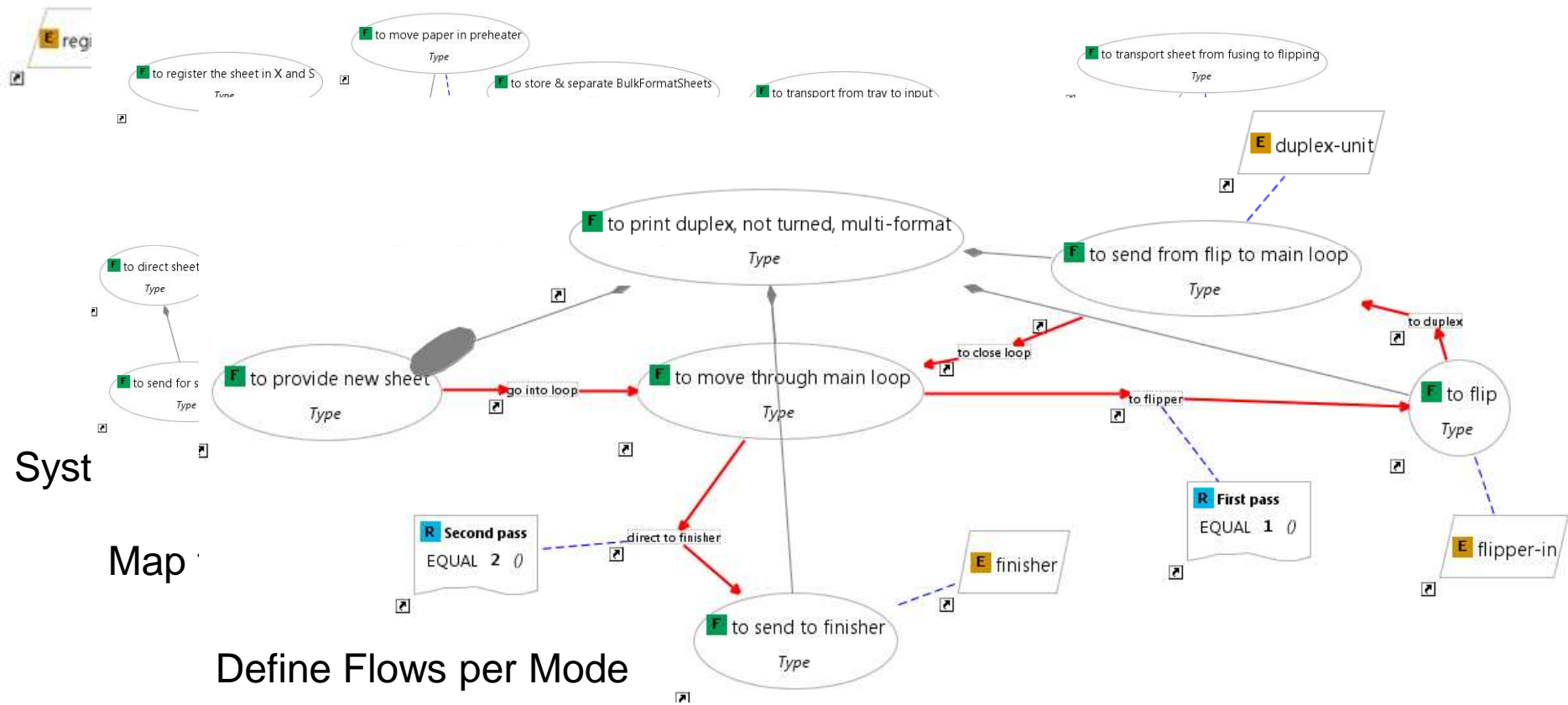




# OCÉ CASE STUDY

## SHARED ARCHITECTURE MODEL

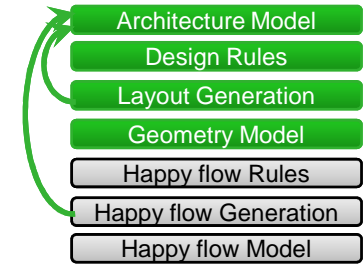
- Architecture Model
- Design Rules
- Layout Generation
- Geometry Model
- Happy flow Rules
- Happy flow Generation
- Happy flow Model





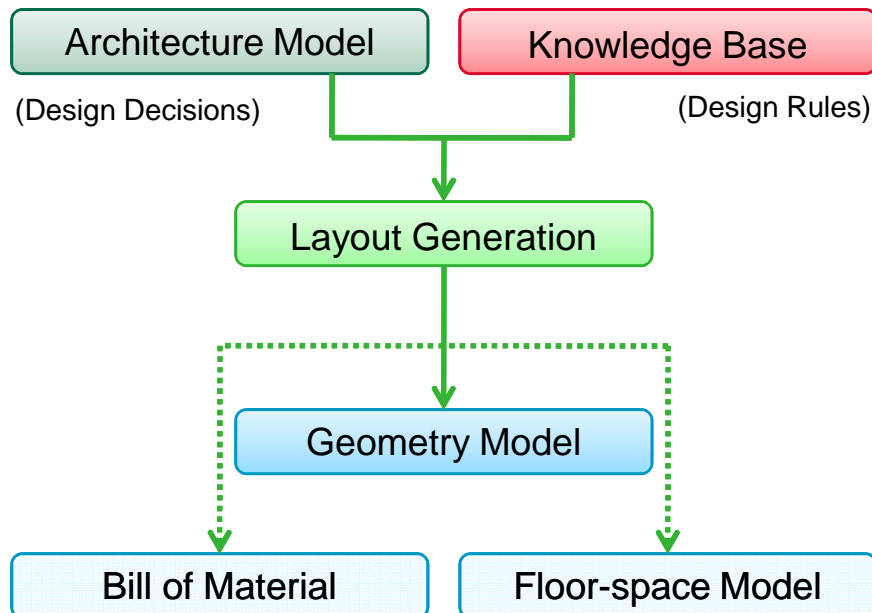
# OCÉ CASE STUDY

## GENERATE PARTS AND TOPOLOGY



Structure:

- Components: Pinches, Fuse, Heater, ...
- Topology: Paths, Segments, ...
- Aspects: Floorspace, Bill of Material, ...

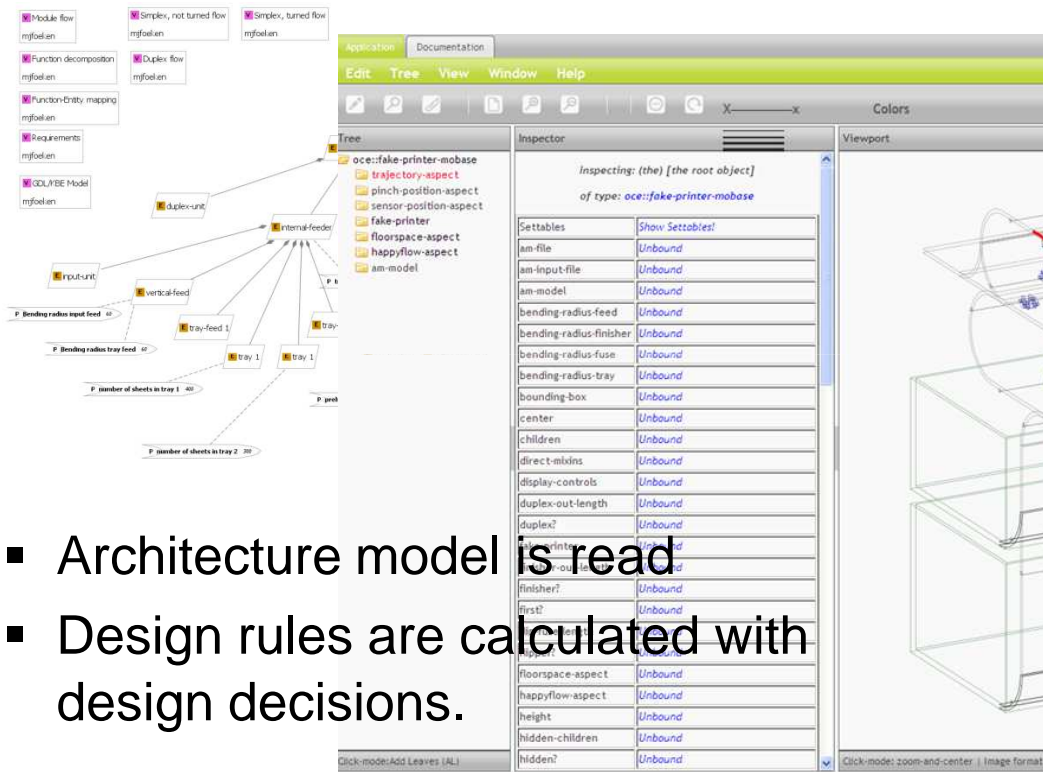
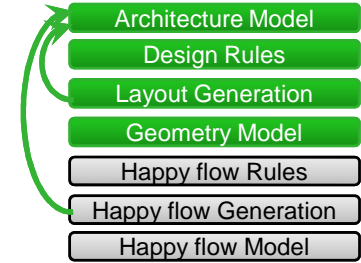


- Parts and Topology are dependent on functions and their flow
- Define a parametric model
- Capture design knowledge in Knowledge Based Engineering Framework
- Example:
- Shortest paper length (**L**) determines distance between pinches (**P**)

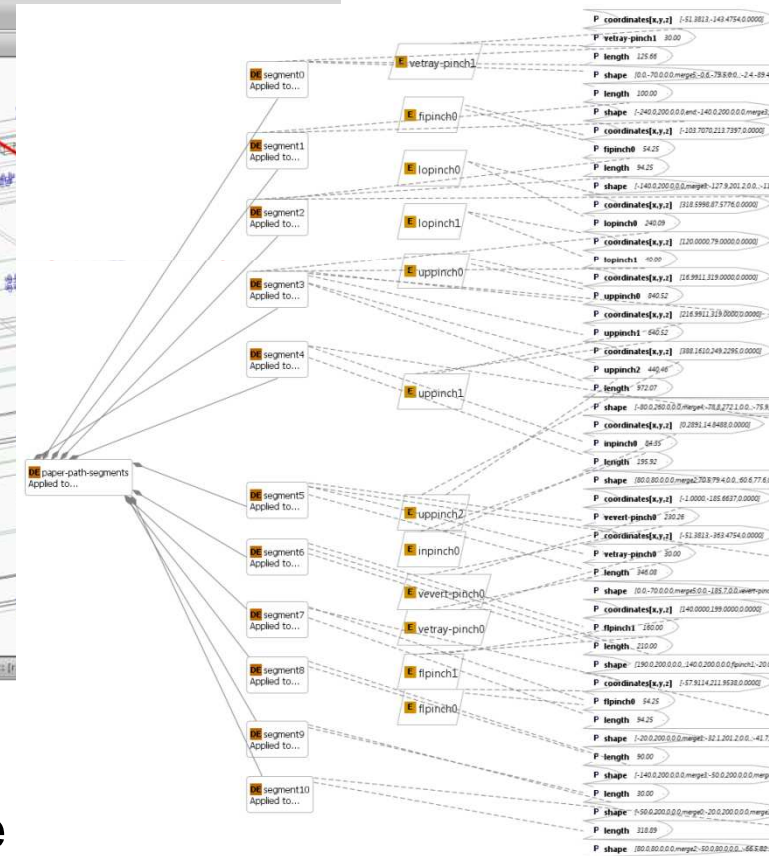


# OCÉ CASE STUDY

## LAYOUT GENERATION



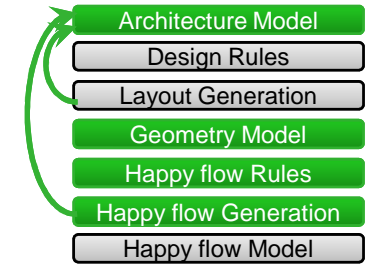
- Architecture model is read
- Design rules are calculated with design decisions.
- Exceptional prepares topology and a partition
- information back to Architecture Model
- Generate multiple variants in design space





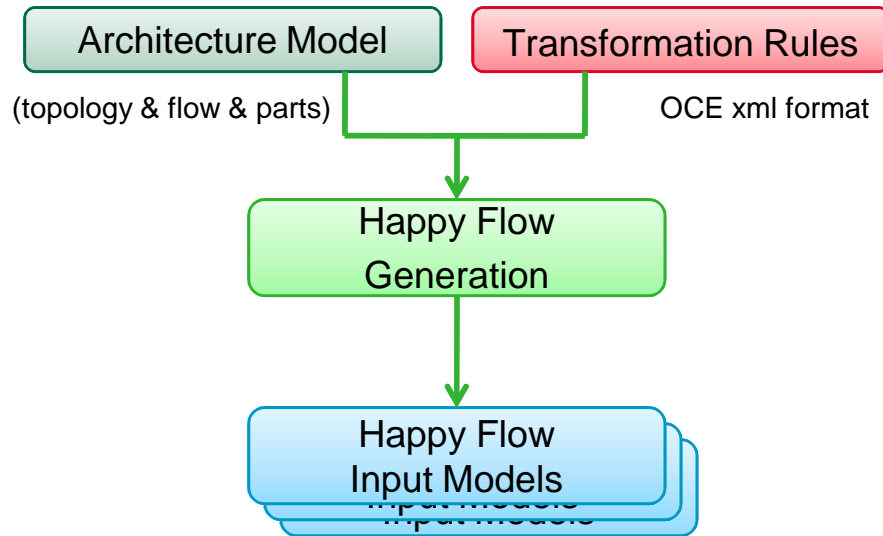
# OCÉ CASE STUDY

## HAPPY FLOW GENERATION



Behavior:

- Real-time behavior: Paper flow per running mode

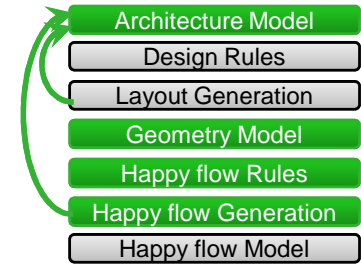


- Function flow describes run modes (simplex, duplex, A4, A3)
- Topology describes distances, parts
- Parts have behavior profiles (paper acceleration)
- Generate Simulation Model
- Generate Embedded Software

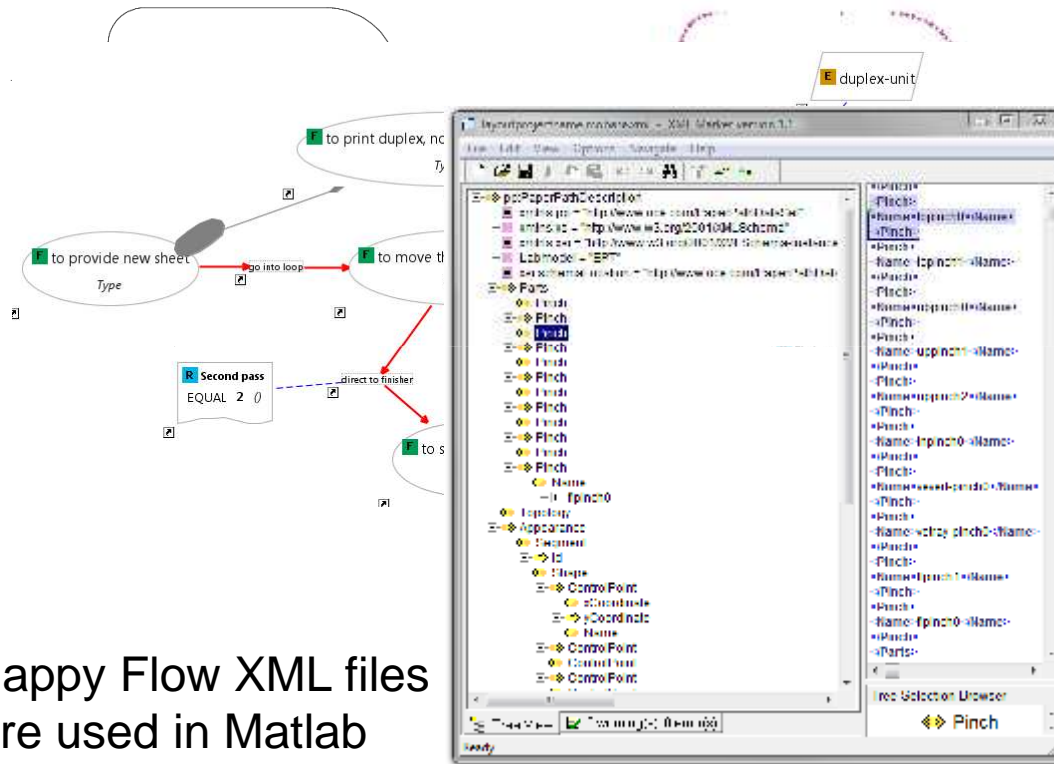


# OCÉ CASE STUDY

## HAPPY FLOW GENERATION



- Split up the segments in segments from function flow
- Write Component information to xml



```

<PointOfInterest>
  <Type>Pinch</Type>
  <Position>202.8</Position>
  <Name>DupPi1Pi</Name>
</PointOfInterest>
</Segment>
</Topology>
<Appearance>
  <SegmentAppearance>
    <Id>1</Id>
    <Shape>
      <ControlPoint>
        <xCoordinate>390.0</xCoordinate>
        <yCoordinate>690.7</yCoordinate>
        <Name>start1</Name>
      </ControlPoint>

```

```

% Operation modes & sequences
duplex = [5, 4, 3, 2, 9, 3, 10, 1]
simplex = [5, 4, 3, 10, 1]

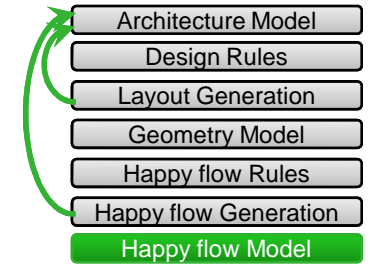
```

Happy Flow XML files  
Are used in Matlab  
model  
(Proprietary)

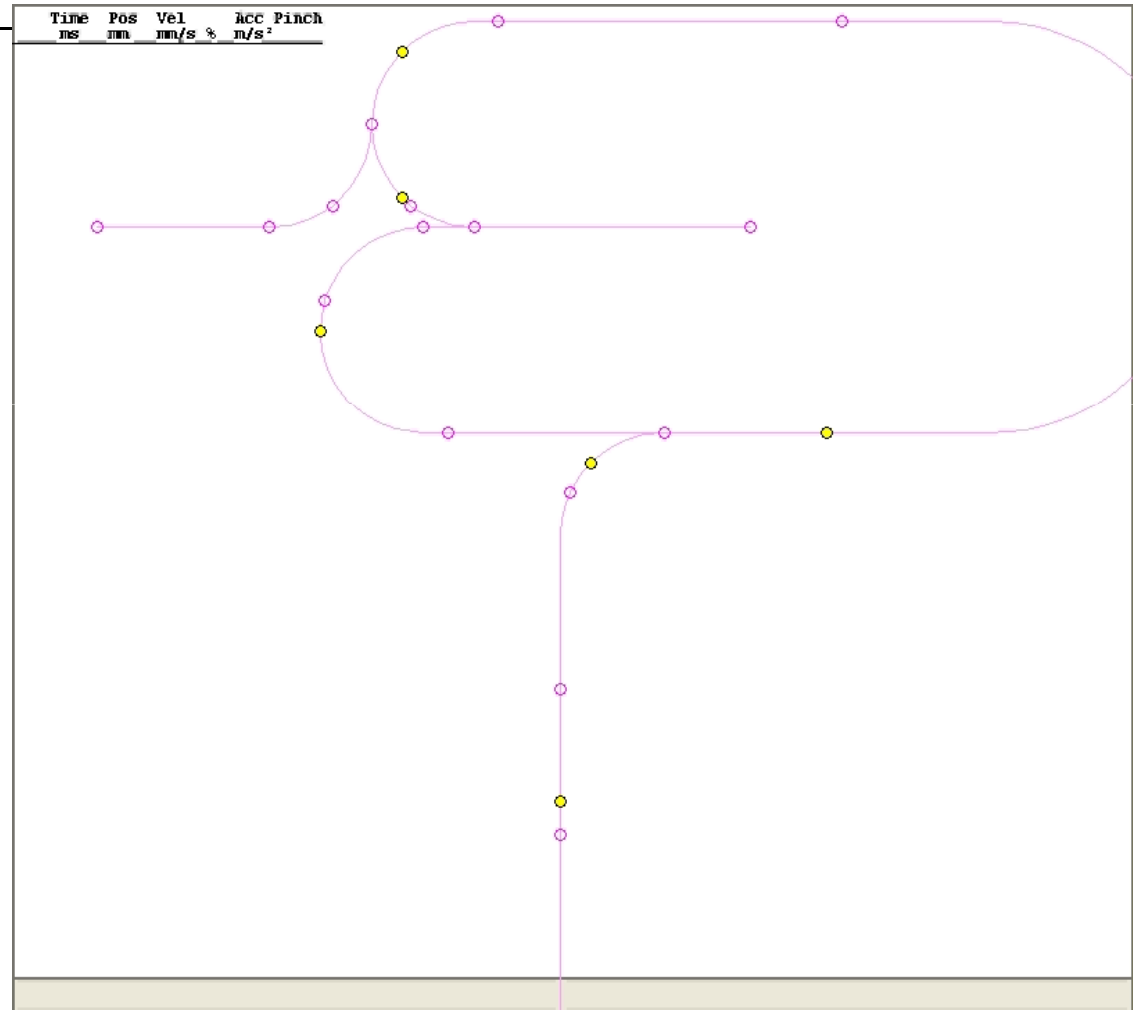


# OCÉ CASE STUDY

## GENERATED REAL-TIME BEHAVIOR



- Result: Happy Flow Model
- Use existing tooling for
  - Real-time simulation
  - Software Generation



# PRELIMINARY RESULTS

## BENEFITS OF AN ARCHITECTING TOOL

---

- Found design rules and design decisions gave insight
  - Share info through multiple views on flat model
  - Overview and consistency through shared data.
  - Automate hand-off
- 
- Matching Tool Chain to Design Process: 3 weeks → 1 hour

# CONCLUSION

## BENEFITS OF A CENTRAL ARCHITECTING TOOL

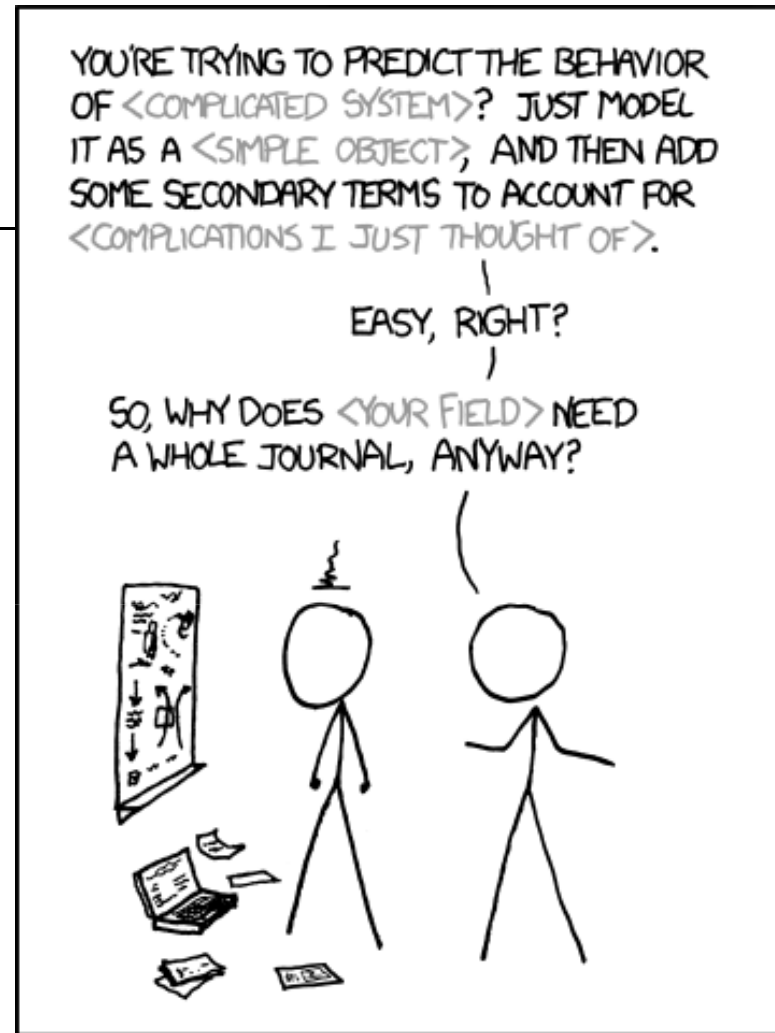
---

- **Enabling concurrency** by providing a mechanism that keeps the information, which is the input and output of the development processes, consistent. This eliminates conflicts caused by incorrect information. Finding, changing and sharing information will be easier due the unambiguous structuring of the Information
- **Enabling short feedback cycles** by (multidisciplinary) simulations by making the information digitally available.
- **Enabling balancing** of the effects of a change over more design aspects and disciplines by simulations, leading to a higher quality of the product.
- **Enabling shorter feedback cycles** by synthesizing of software and hardware based on the digitally available product information.
- **Enabling more reuse** by providing easier access to information of the components and platforms and managing the complex change management.

# ACKNOWLEDGMENTS



- Dutch Innovation Oriented Research Program „Integrated Product Creation and Realization (IOP-IPCR)“ of the Dutch Ministry of Economic Affairs.



LIBERAL-ARTS MAJORS MAY BE ANNOYING SOMETIMES, BUT THERE'S *NOTHING* MORE OBNOXIOUS THAN A PHYSICIST FIRST ENCOUNTERING A NEW SUBJECT.