A method for quantitative evaluation of functional chains supported by a Capella add-on

TNO-ESI:

Alexandr Vasenev

Jacques Verriet

Koen Kanters

Jozef Hooman

Thermo Fisher Scientific:

Joost Dierkse

Olivier Rainaut

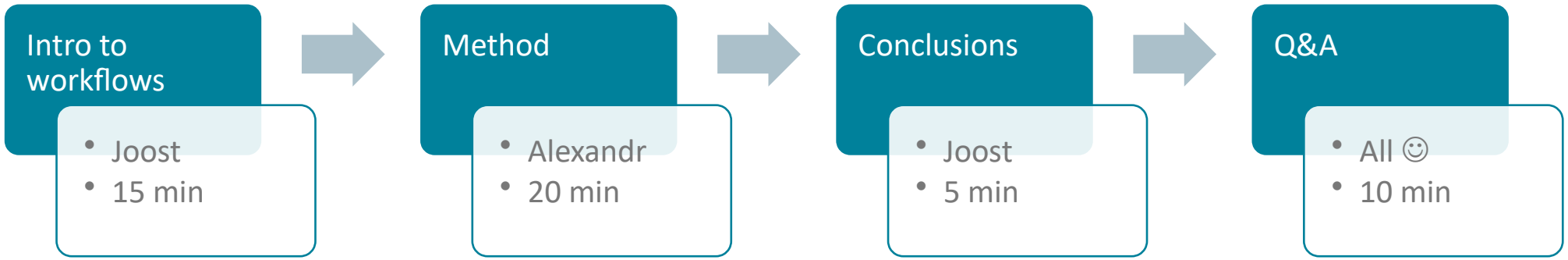
Jamie McCormack

ThermoFisher SCIENTIFIC

ESI

An initiative of industry, academia and TNO

Outline


Intro to workflows
Joost
15 min
Method
Alexandr
20 min
Conclusions
Joost
5 min
Q&A
All ☺
10 min

Introduction to workflows

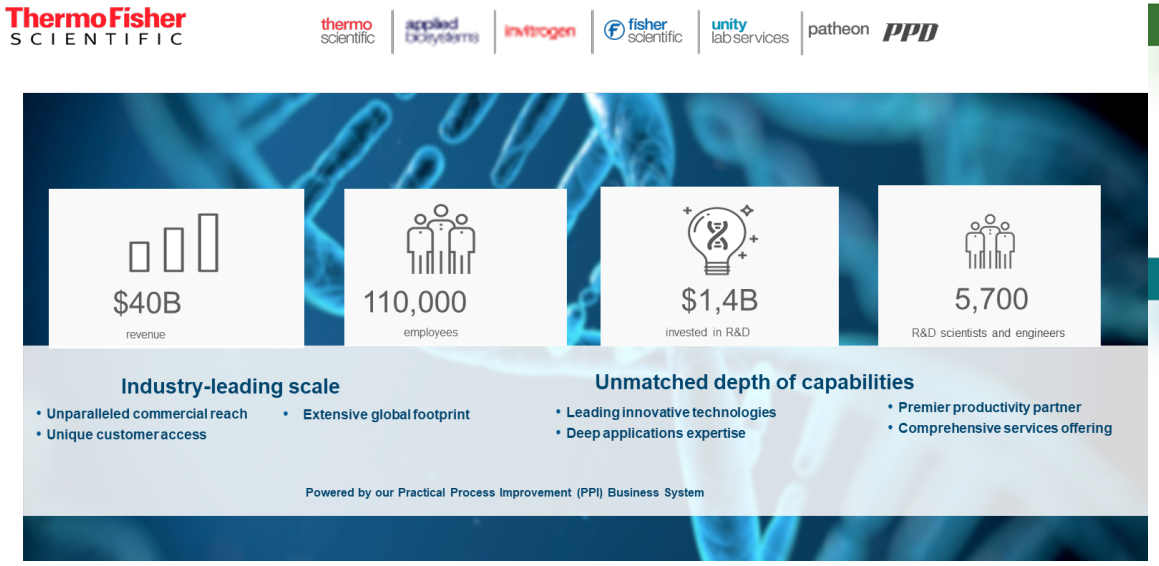
(Joost, 15 min)

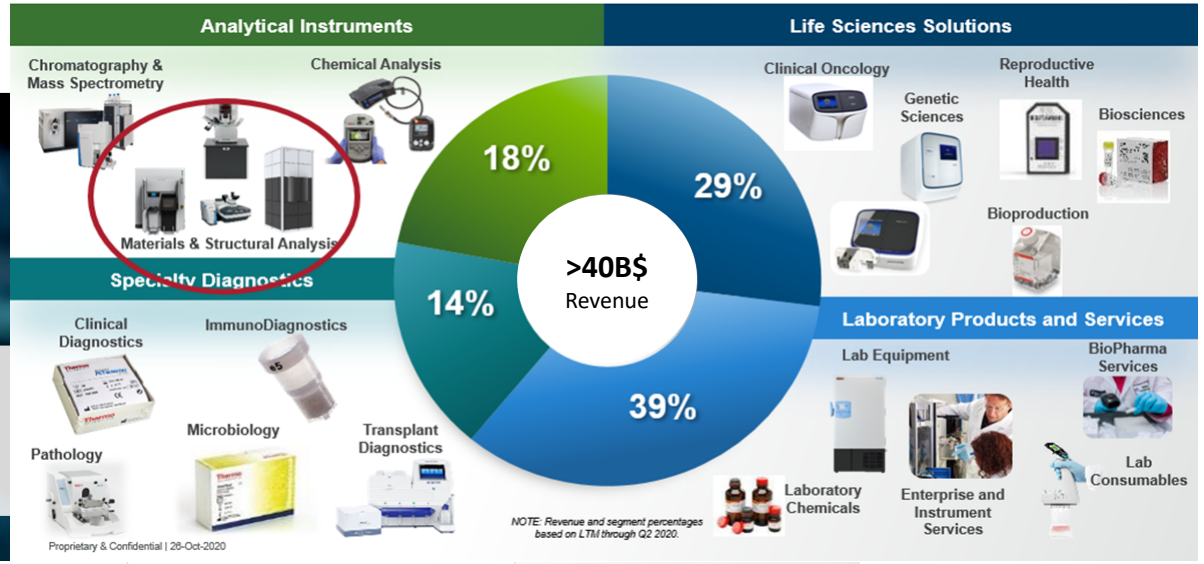
ThermoFisher SCIENTIFIC

ESI

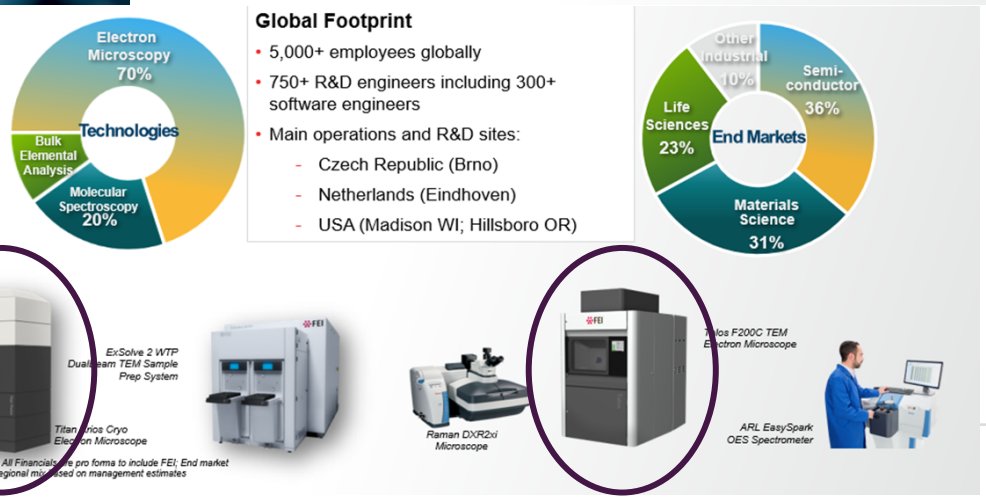
An initiative of industry, academia and TNO

Thermo Fisher Scientific





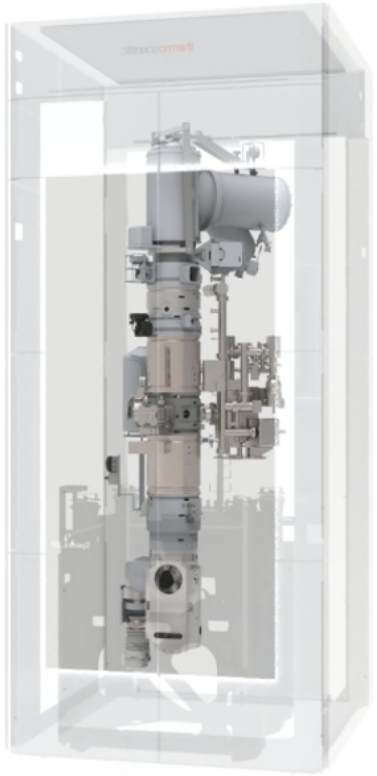
➢ Material and Structural Analysis



System Scope

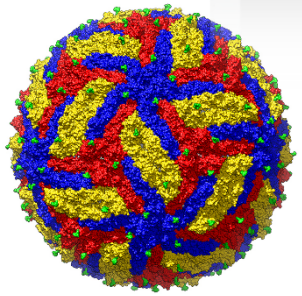
Transmission Electron Microscope

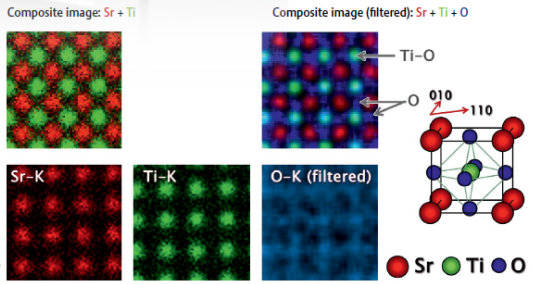
- **Dimensions: 1.6m x 1.6m x 3.0-4.3m**
- **Weight: > 1600kg**
- **Resolution: 50pm**
- **Cost: 1M$ – 15M$**

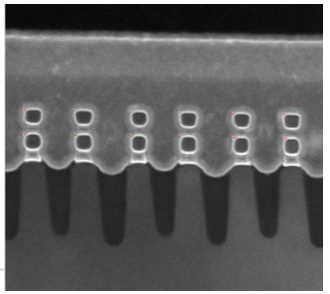


Applications

- **Life Sciences**
  - Virus and cell structure research
- **Material Sciences**
  - Chemical and material investigations
- **Semiconductor**
  - Process analysis / control

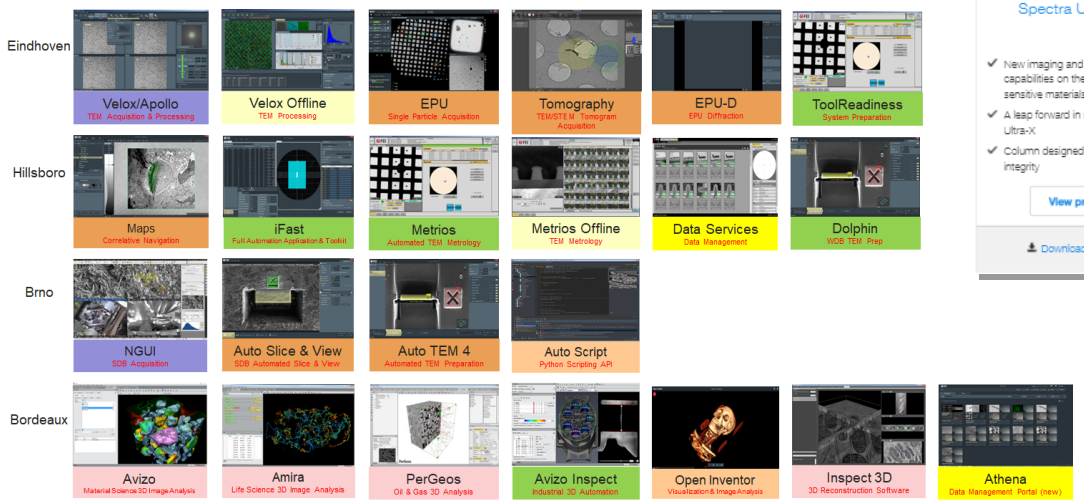


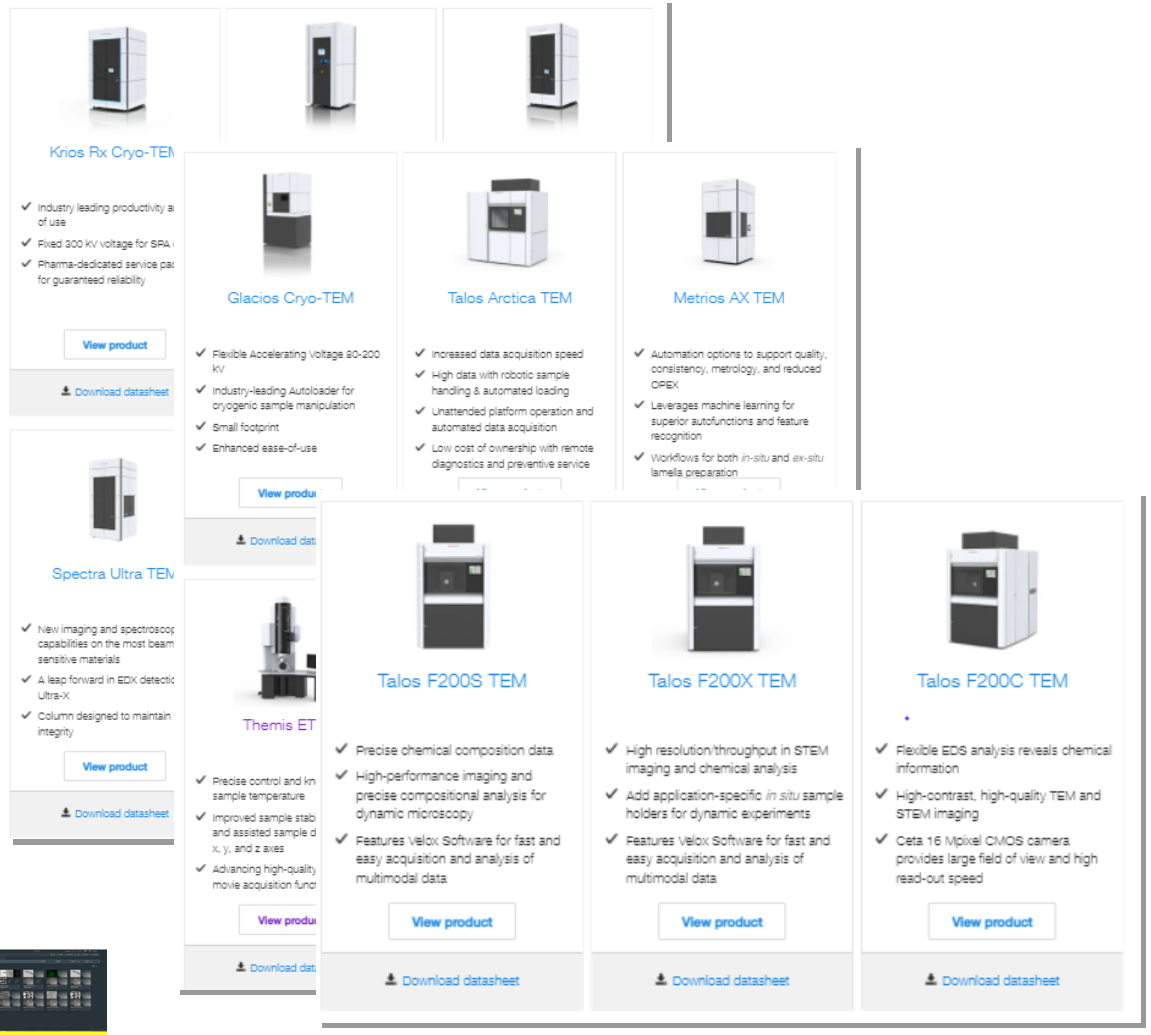




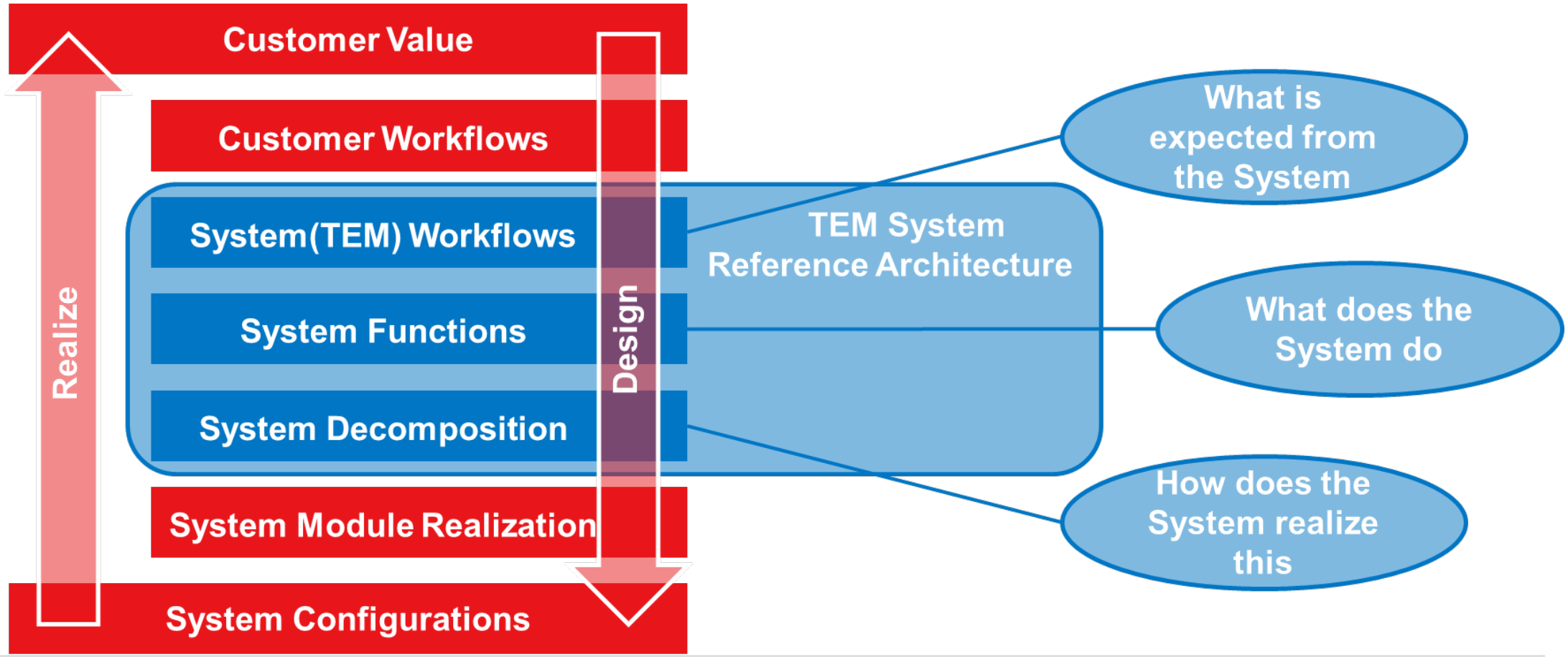
Background

- 15 active commercials TEM products
- More than 20 customer-facing applications
- More than 400 Modules
- 1 TEM Server Software Stack
- 1000+ active configurations
- Distributed Development



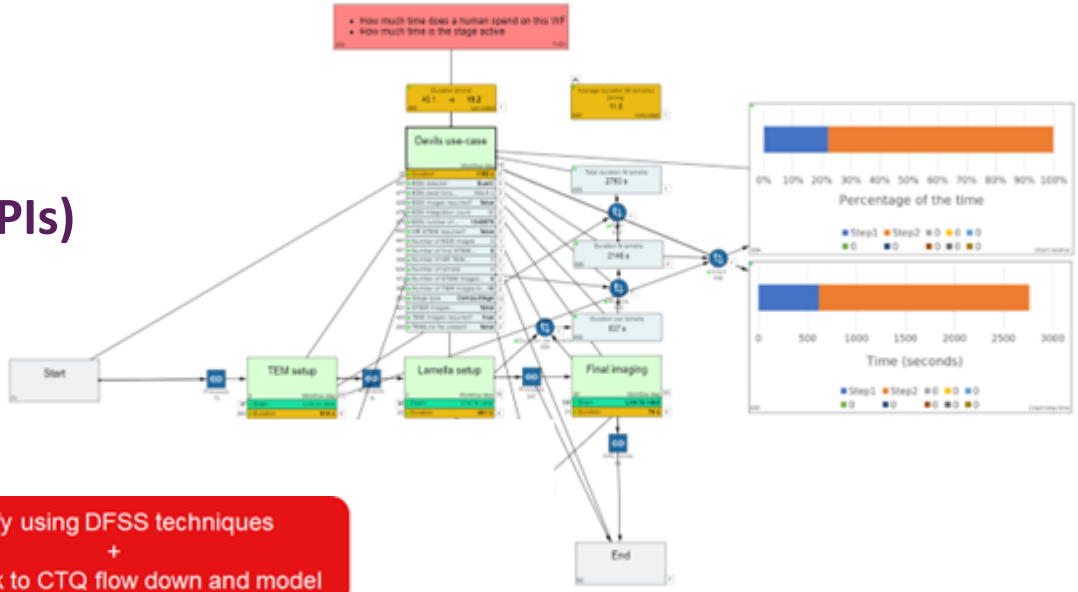


Reference Architecture Overview



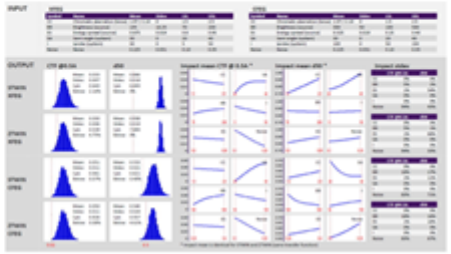
Workflow Analysis

- Identify Customer Value
- Identify Critical To Quality (CTQ) Parameters (KPIs)
- Allocate CTQ Parameters to Features

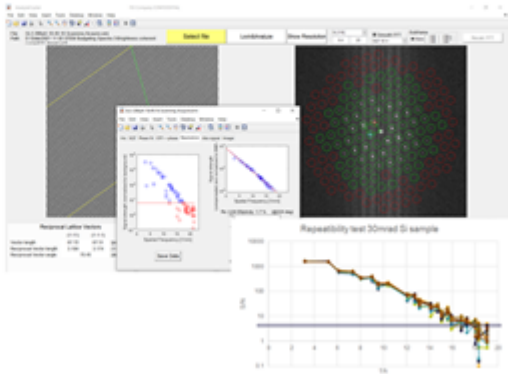


Start with CTQ Flow down From Customer needs via System CTQs to module CTQs

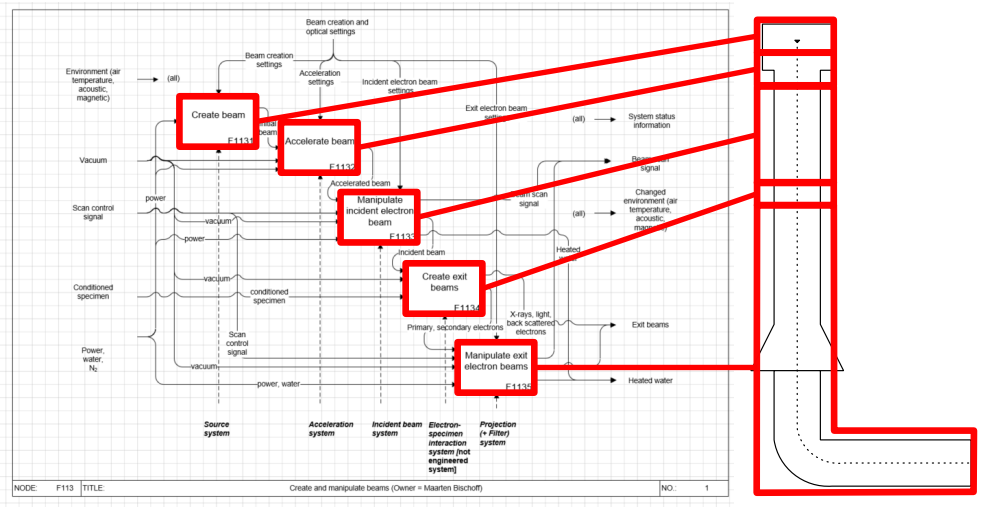
Build transfer function model by combining domain specific models to do impact analysis and tolerancing



Verify using DFSS techniques + feedback to CTQ flow down and model

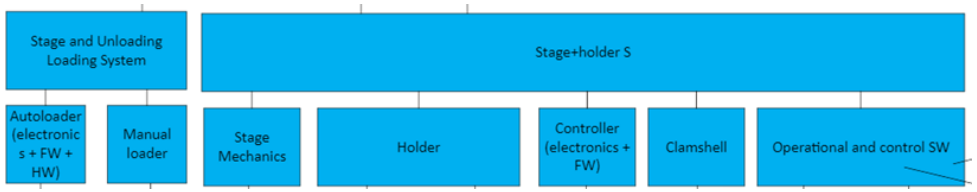


Functional Decomposition / System Decomposition



- **Decompose System into Functions**
  - Decompose CTQ parameters
  - Cover hardware and software

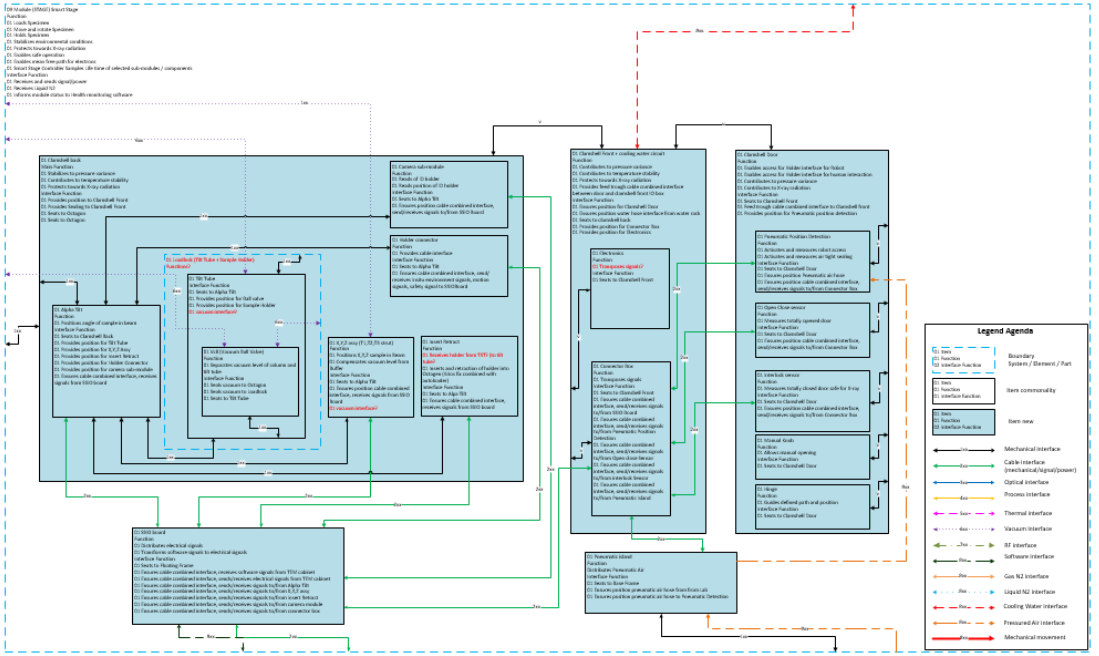
- **Decompose System into Modules**
  - Allocate Functions to Modules and Owners (Hardware and Software)
  - Logical grouping of functions
  - Define transfer functions for Modules (based on Function CTQ Parameters)



Logical Architecture / Compatibility

- **Identify interfaces between Modules**
  - Critical for compatibility
  - Used for risk management (FMEA)
  - Component naming aligned though Taxonomy

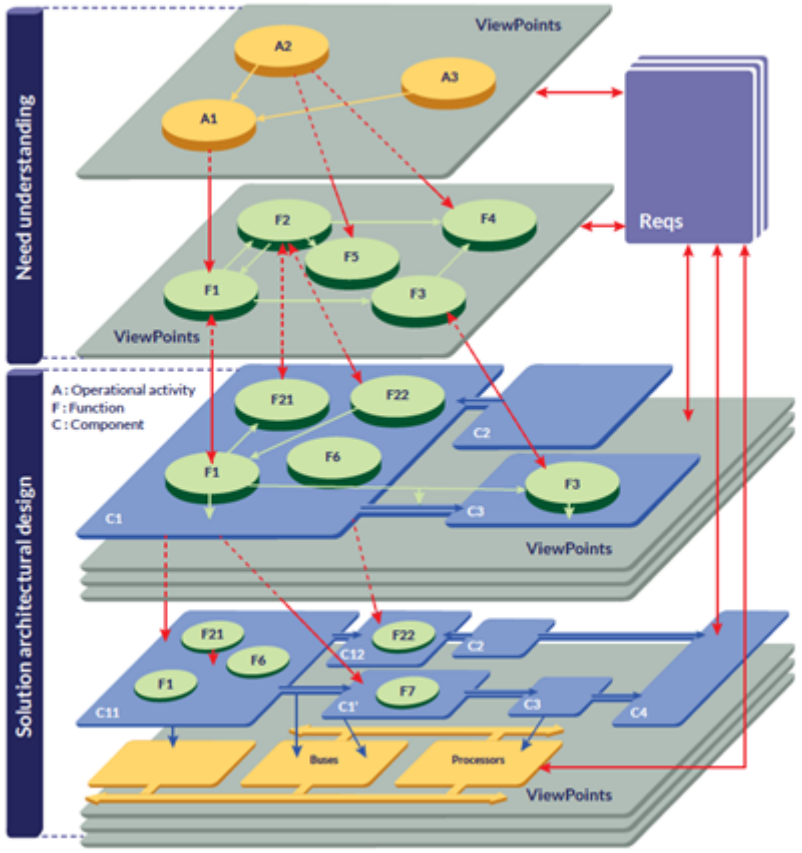
- **Define technical compatibility**
  - Map Module compatibility to Systems
  - Include Non-Standard Requests
  - Include Backward Compatibility



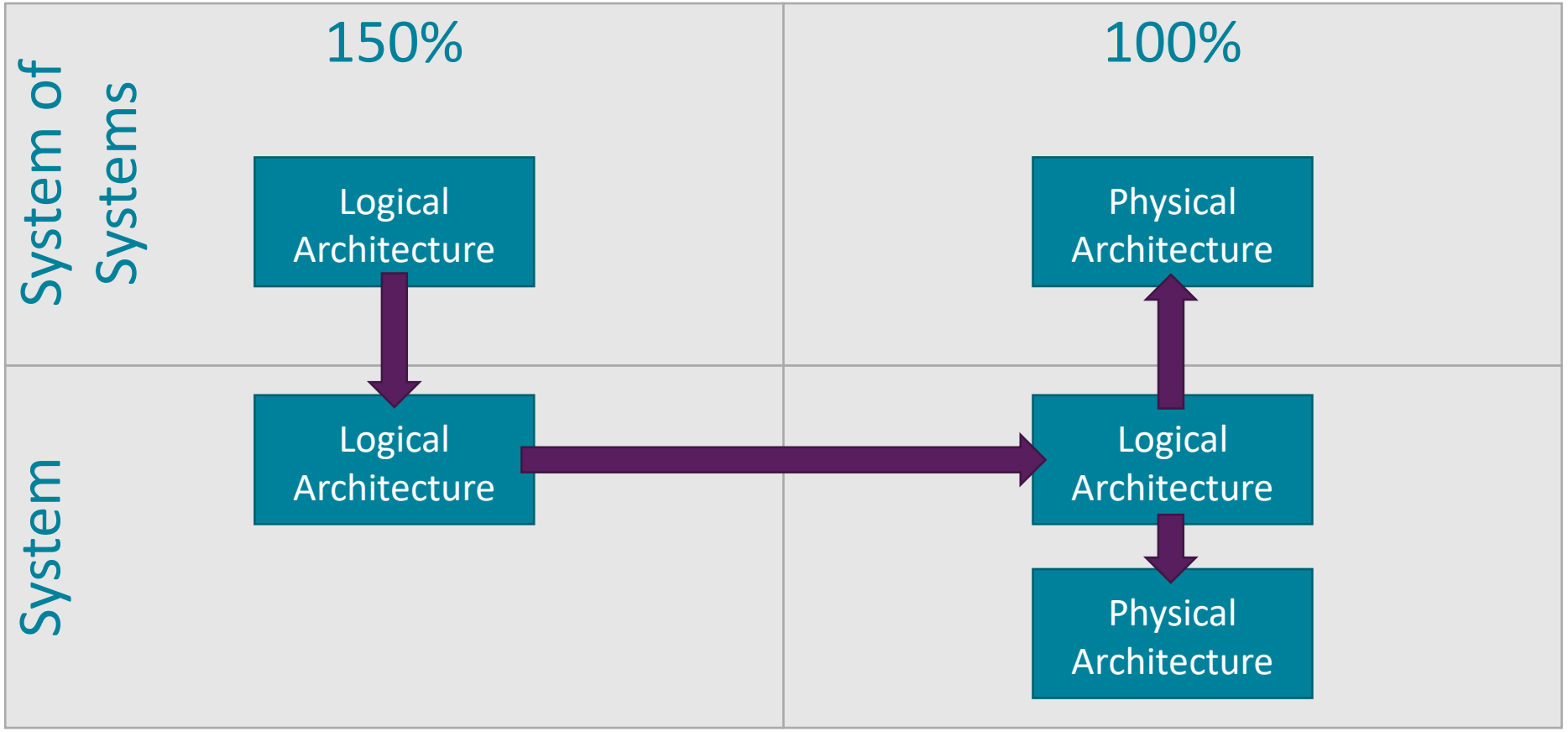
MBSE – Evolving Reference Architecture

- **Tool to support Reference Architecture**
  - Guarantee consistency
  - Ease maintainability
  - Increase automation opportunities

- **Process to ensure maintainability**
  - System-of-Systems approach
  - Separate models for System versions

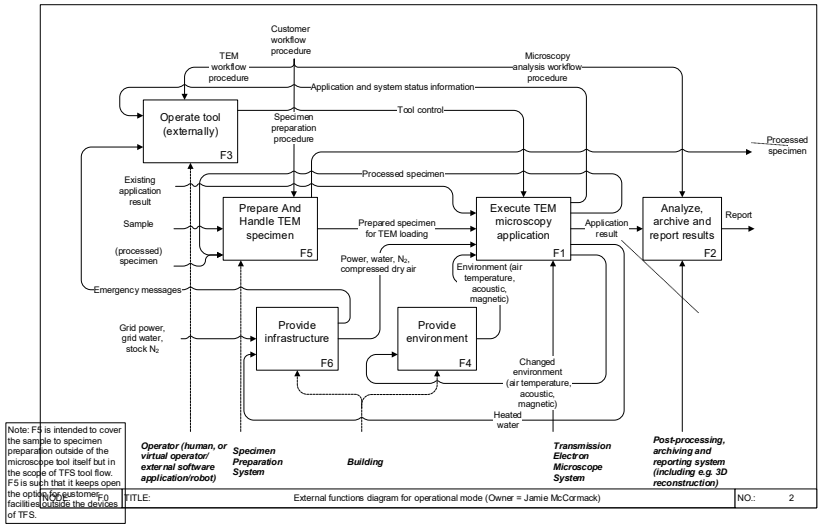


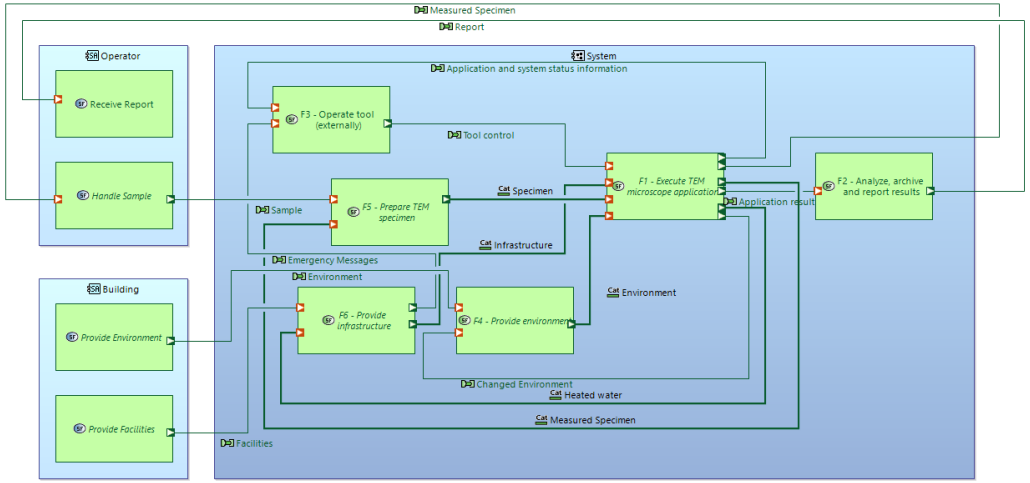
MBSE – Modeling Structure



MBSE – Transition

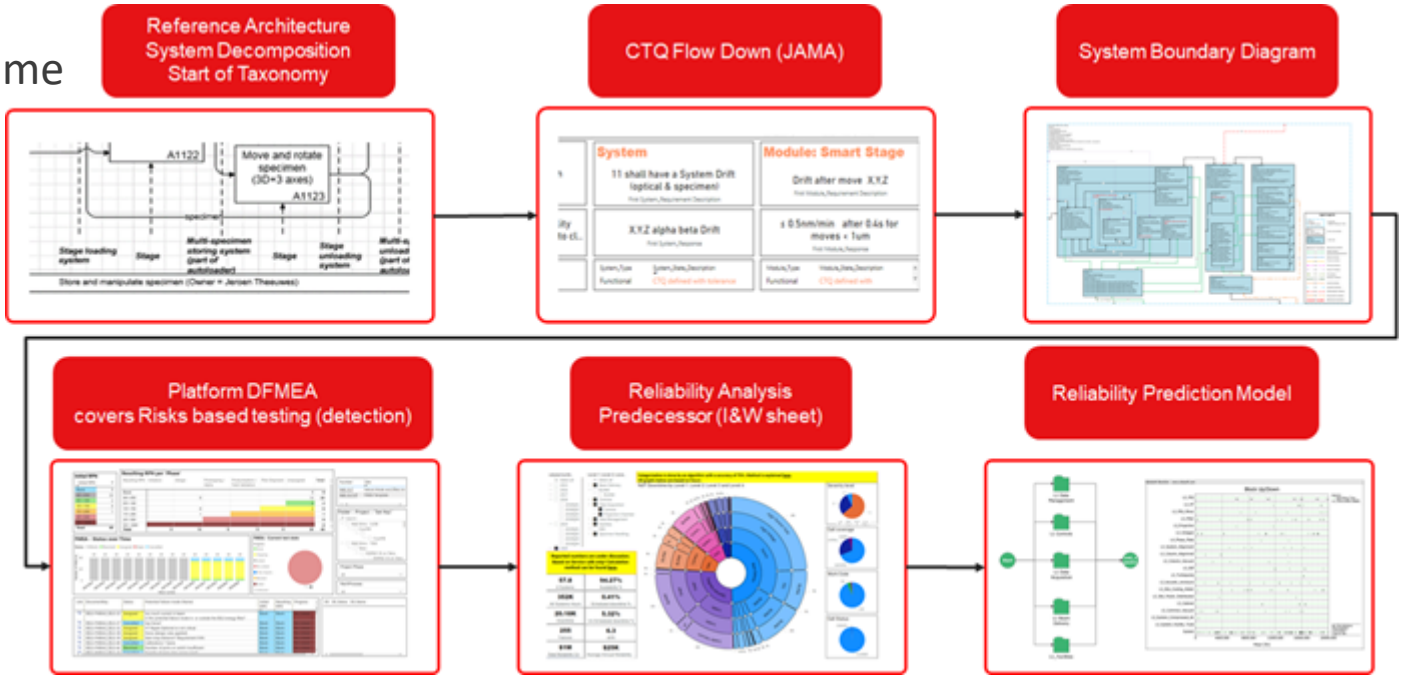
- **Transfer Reference Architecture to Capella**
  - Functional Decomposition
  - System Decomposition
  - Interface Specifications





MBSE – Maximizing Benefits

- **CTQ / KPI Calculation**
  - Attribute model with CTQ parameters
  - Extract performance based on configuration
  - Reliability prediction model
  - Define Customer Value at design time



Method for quantitative evaluations of functional chains

(Alexandr, 20 min)

ThermoFisher SCIENTIFIC

ESI

An initiative of industry, academia and TNO

ESI at a glance

Synopsis

- Foundation ESI started in 2002
- ESI acquired by TNO per January 2013
- ~60 staff members, many with extensive industrial experience
- 7 Part-time Professors
- Working at industry locations

Focus

Managing complexity of high-tech systems

through

- system architecting,
- system reasoning and
- model-driven engineering

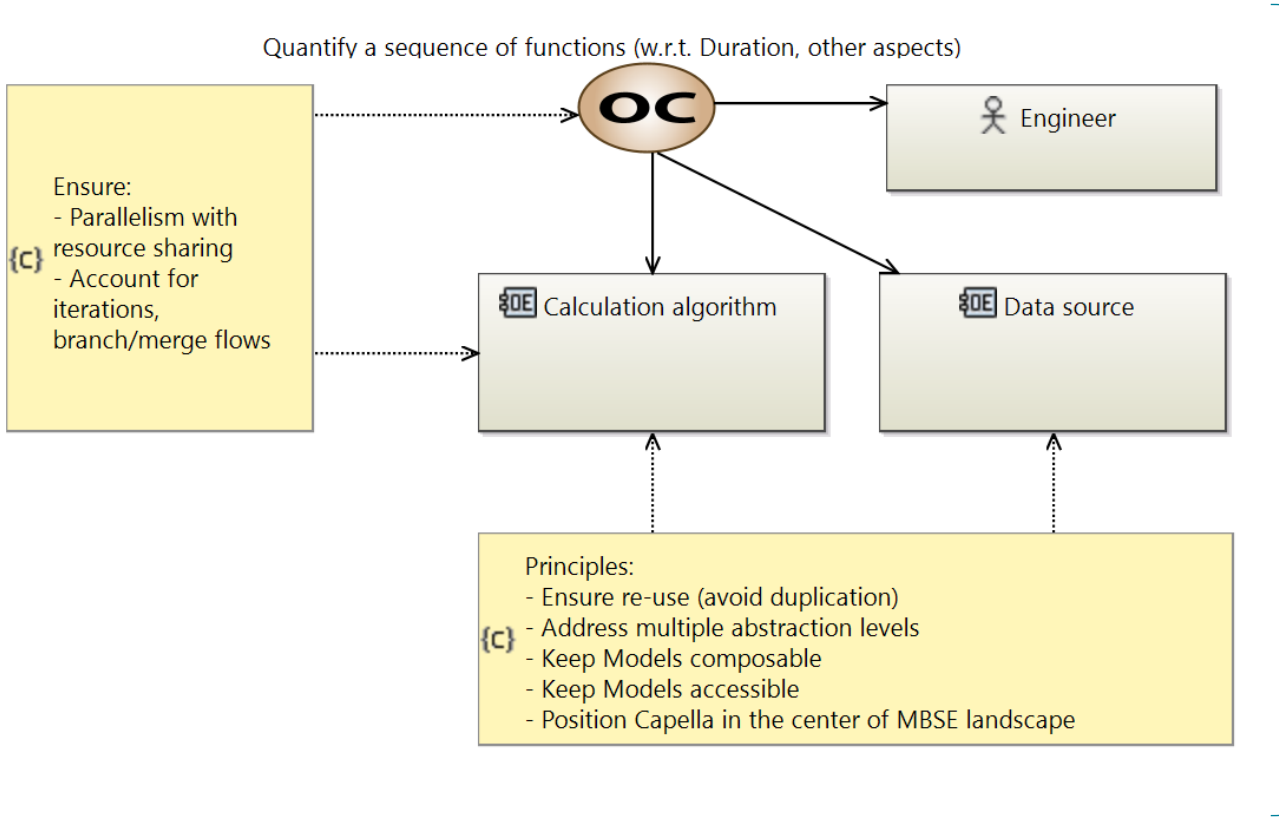
delivering

- methodologies validated in cutting-edge industrial practice

Partners

ASML, ThermoFisher Scientific, Philips, Vanderlande, itec, Thales, Canon, TNO, University of Amsterdam, TU Delft, Delft University of Technology, Radboud University Nijmegen, University of Twente., TU/e Eindhoven University of Technology, Capgemini engineering

What do we want to achieve?



Conclusion: we need simulation

Approach: Simulate functional chains in Capella

- **Why**: Natural fit to 'precedes' relation, Iteration/OR/AND nodes
- **To do**: Specify meaning of nodes, add properties

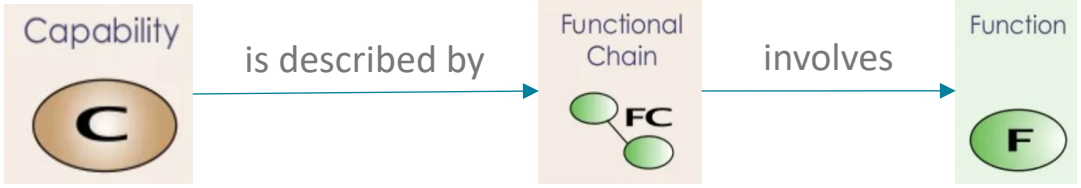
Intro to Functional Chains

Functional Chain: a specific path among all possible paths (using certain Functions and Functional Exchanges).

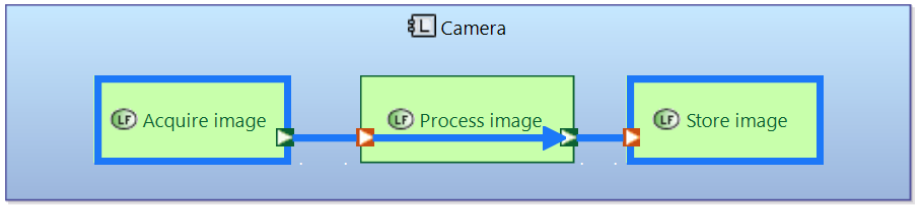
Systems Architecture Modeling with the Arcadia Method

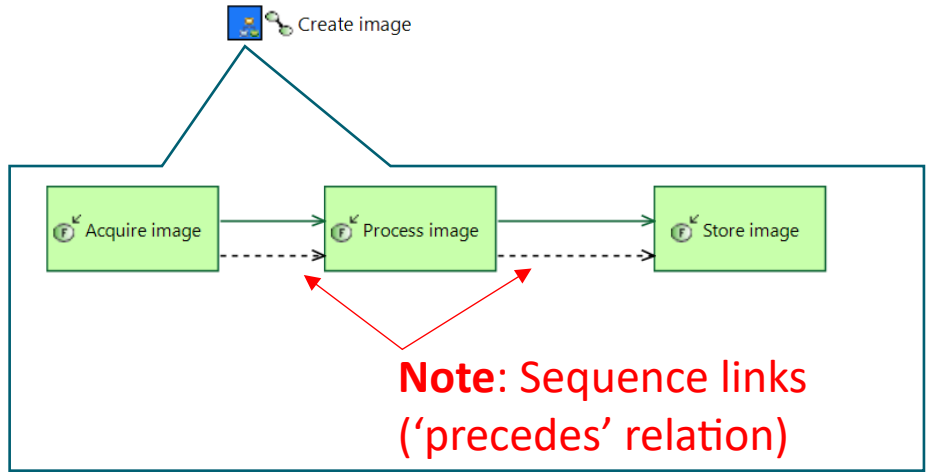
Pascal Roques

A Practical Guide to Capella



Example of a functional chain:



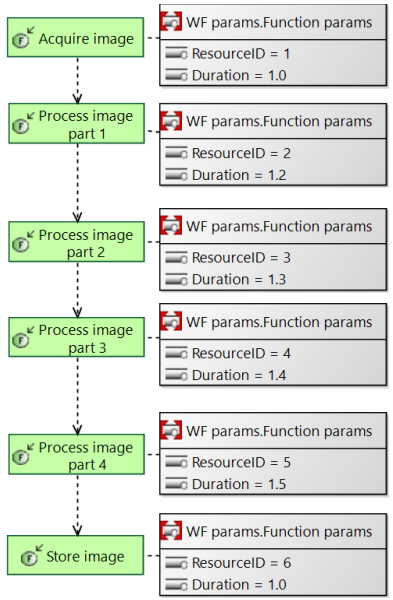


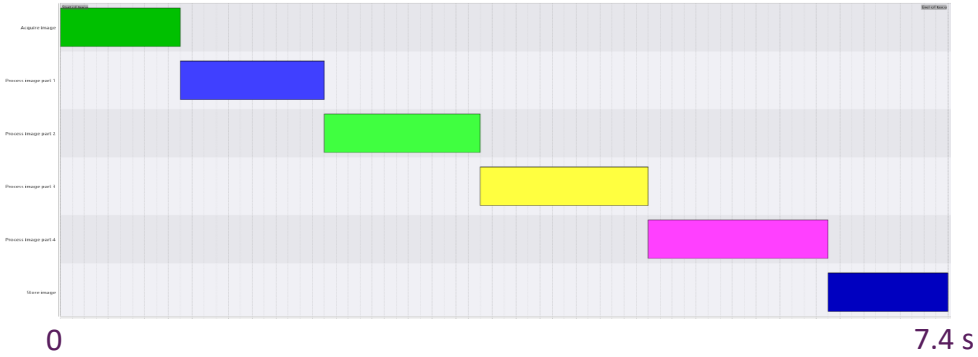
* A lead to functional modeling: Functional flow block diagram - Wikipedia

A method in a nutshell

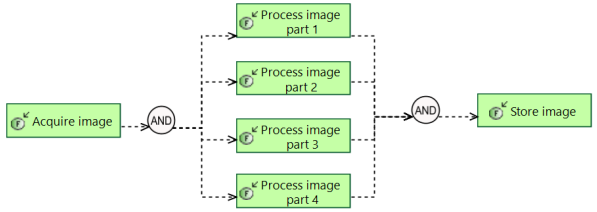
End goal: To get quantified results of functional chains, for instance:

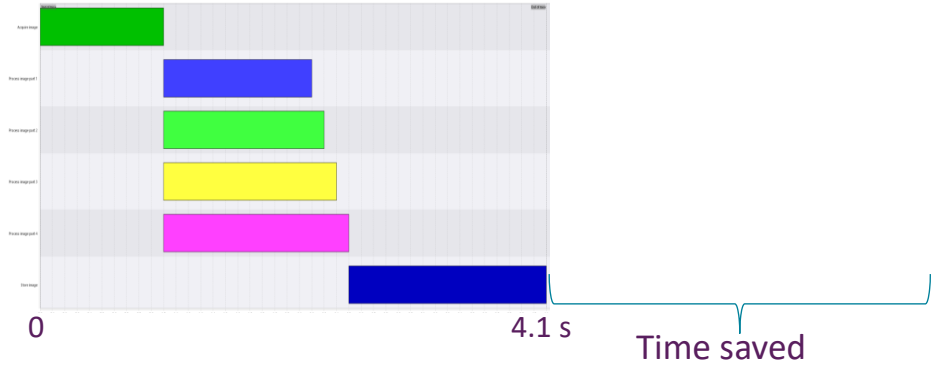
1





2





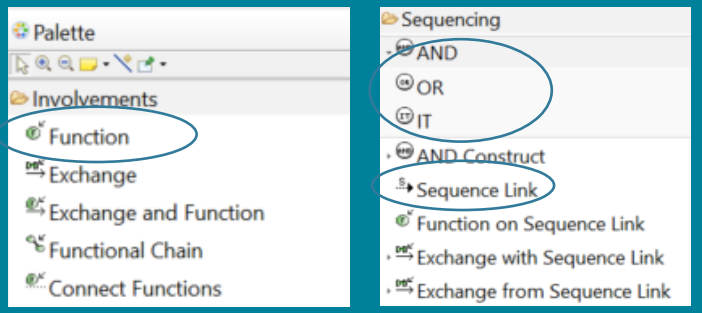
* Note additional property values: ResourceID and Duration

A method in a nutshell

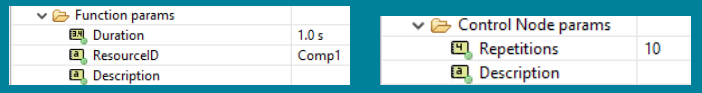
Steps:

1. Create a ‘simulatable’ functional chain (with specific property values)
2. Export and run the chain
3. Visualize result

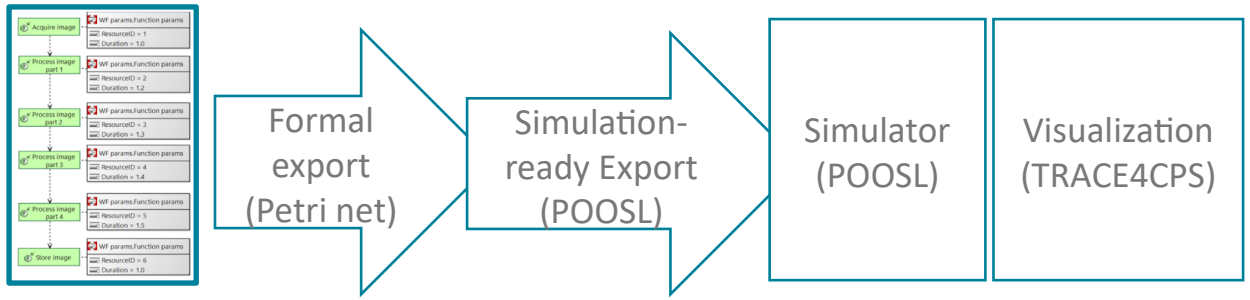
Used constructs:

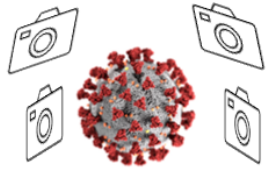


Extra parameters:



Overall architecture:



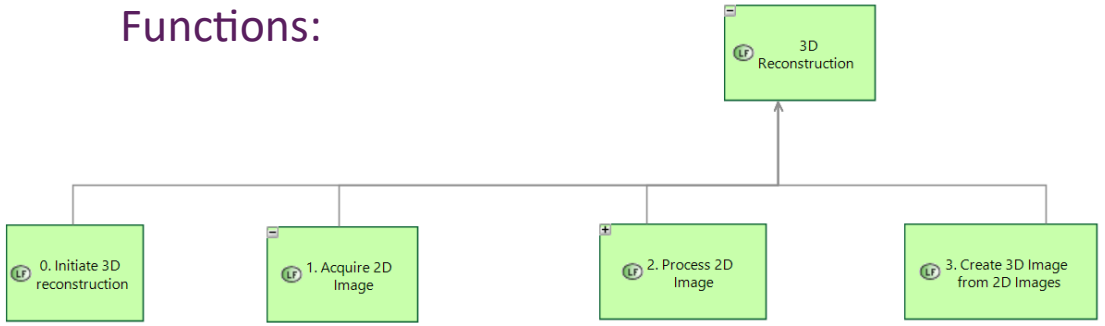


Demo:
construct a 3D model out of 2D images

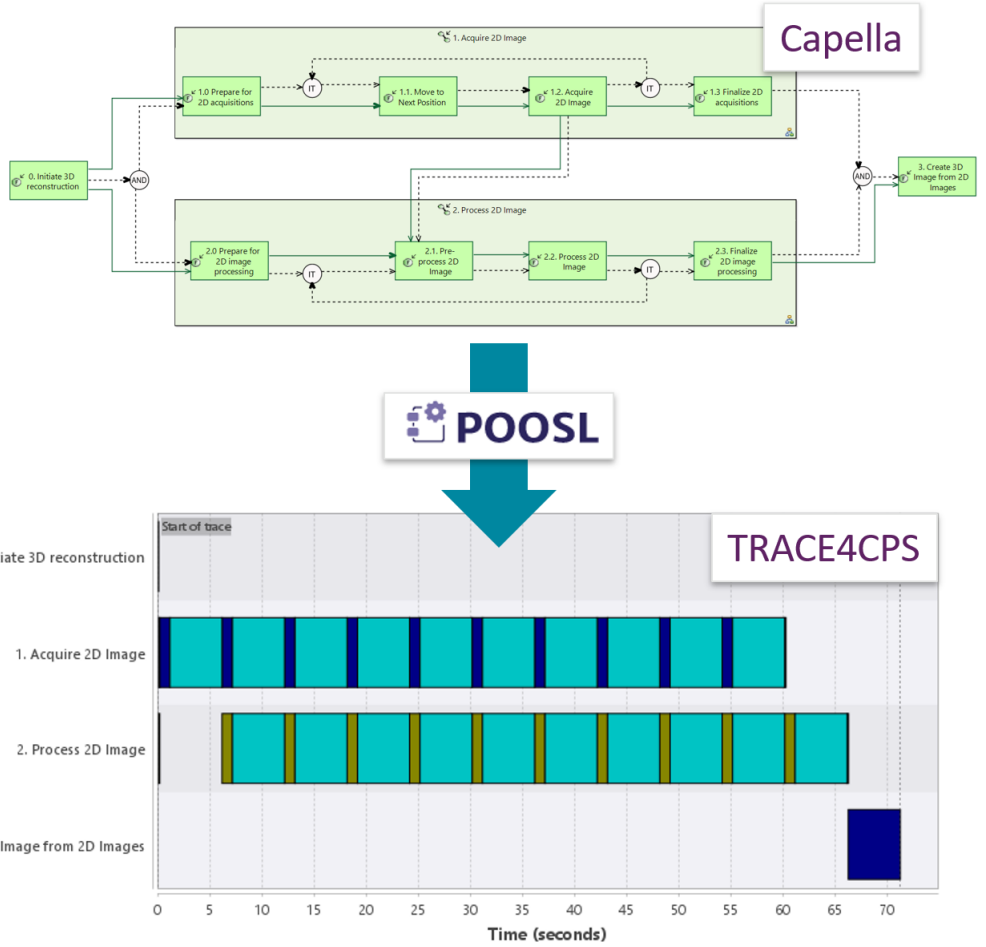
Highlights:

- A library of functional chains
- Generated graph

Functions:



Overview:



Behind the scenes

Formalized model

Workflow (element) – Graph structure (2/4)

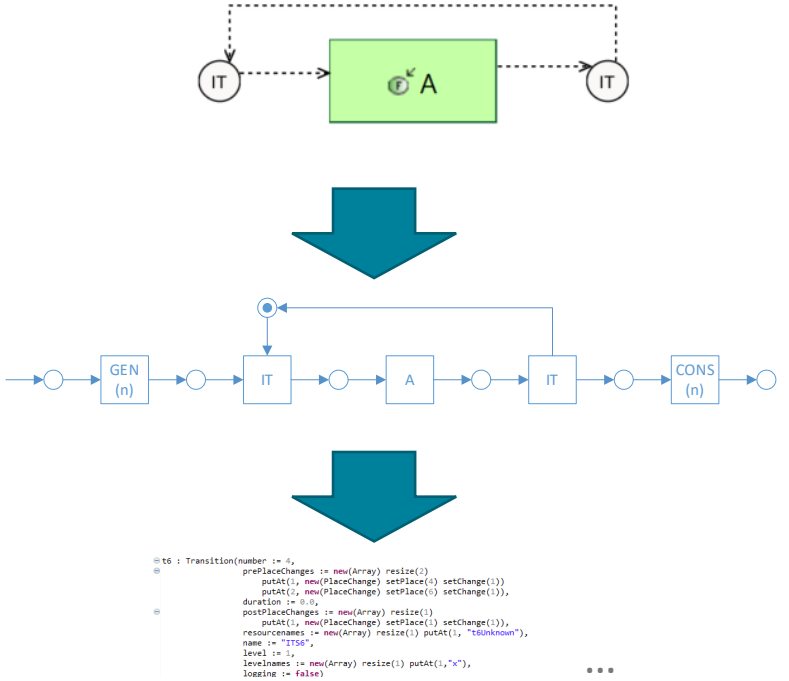
A workflow (element) $G = (N_F \cup N_C, A)$ satisfies following conditions

- N contains exactly one $START$ node, i.e. a node $n \in N$ with type $t_n = START$ (or $t_n = START\&END$)
- N contains exactly one END node, i.e. a node $n \in N$ with type $t_n = END$ (or $t_n = START\&END$)
- The $START$ node has no predecessor nodes, i.e. no incoming arcs
- The END node has no successor nodes , i.e. no outgoing arcs
- $INTERMEDIATE$ nodes have at least one incoming and at least one outgoing arc
- Every node $n \in N$ lies on a path from the $START$ node to the END node

Property values definition (PVMT add-on)

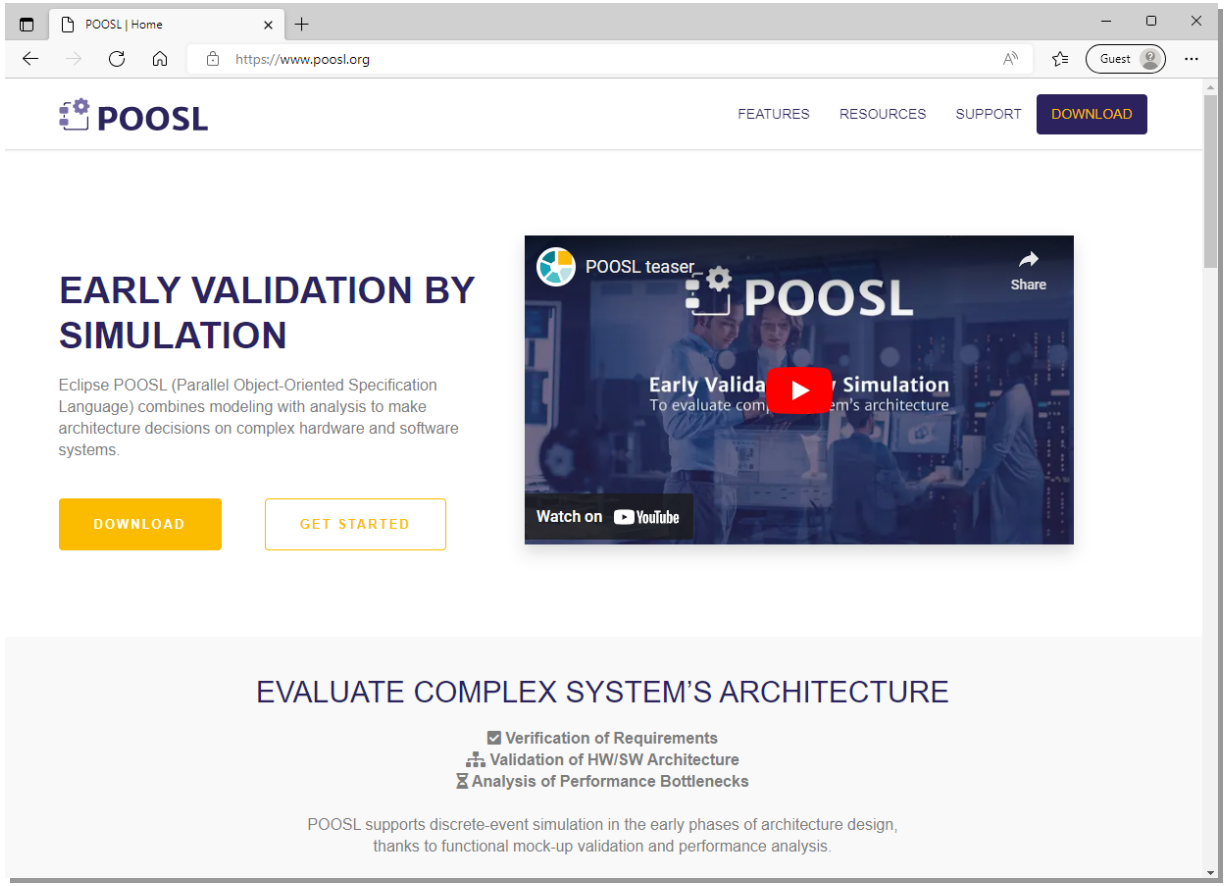
WF params		
Function params		
Scope	[SYSTEM, LOGICAL, PHY...	
Duration	floatProperty	0.0 s
Range [0.0 - 1.0E30]		
ResourceID	stringProperty	
Description	stringProperty	
Sequence Link params		
Scope	[SYSTEM, LOGICAL, PHY...	
Weight	floatProperty	0.0
Range [0.0 - 100.0]		
Control Node params		
Scope	[SYSTEM, LOGICAL, PHY...	
Repetitions	integerProperty	1
Range [1 - 1000000000]		
Description	stringProperty	

Transformations from a functional chain via Petri-net formalism to a simulation



POOSL

https://www.poosl.org/

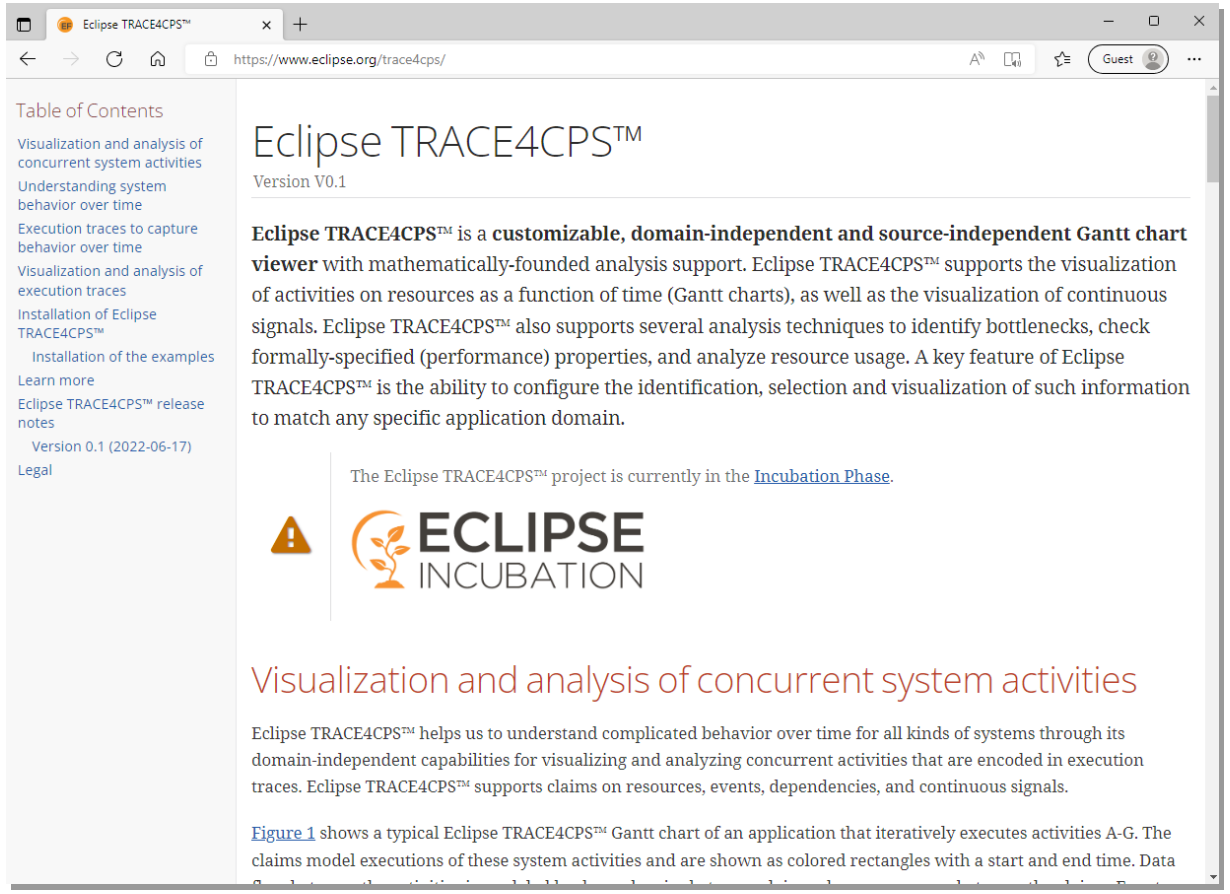


Functionality highlights:

- simulation of parallel processes,
- well-defined semantics

TRACE4CPS

https://www.eclipse.org/trace4cps/



Functionality highlights:

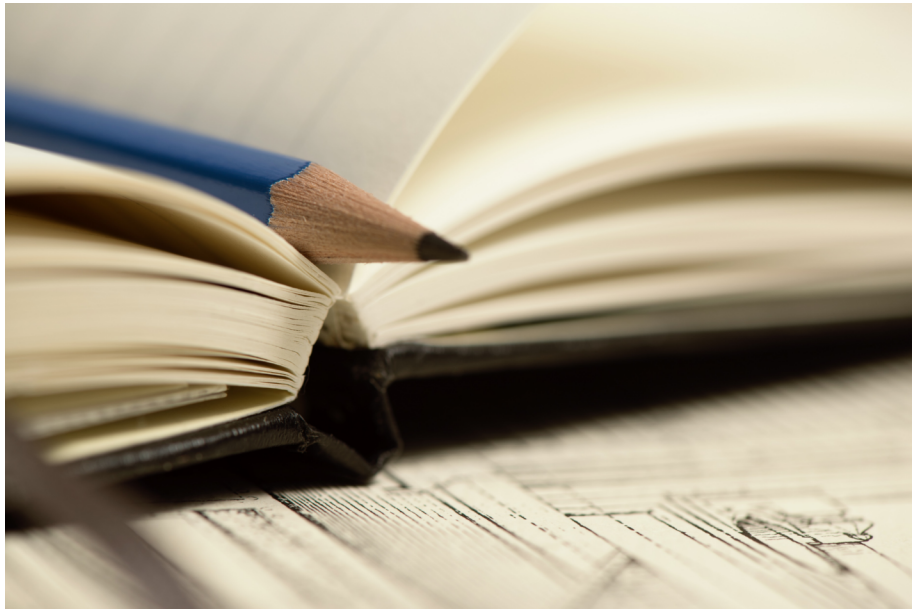
- critical path analysis,
- customization of visualization using user-defined attributes (grouping, coloring, filtering)

Experiences

We've validate the approach through interviews and an industrial case with:
- 12 Functional chains
- 4 levels of nesting
- 3 levels of functions + set of re-usabes functions
- ~35 functions (most used 2+ times)

It's **easy** to:

- explain the model to other stakeholders due to clear traceability
- quickly explore new options
- relate to Arcadia constructs (Functions, Functional chains, Components)



It's **good to remember** that:

- complexity can grow quickly, e.g.,
  - adding extra information
  - when re-using a sub-FC several times in the higher-level FC
  - potential links between Arcadia layers (e.g., to Configuration Items)
- modelers should agree on and carefully follow a modeling convention
- as with any toolchain, one shall pay attention to versioning and exceptions
- there is an entry bar to such a project: Arcadia and Capella knowledge, programming skills

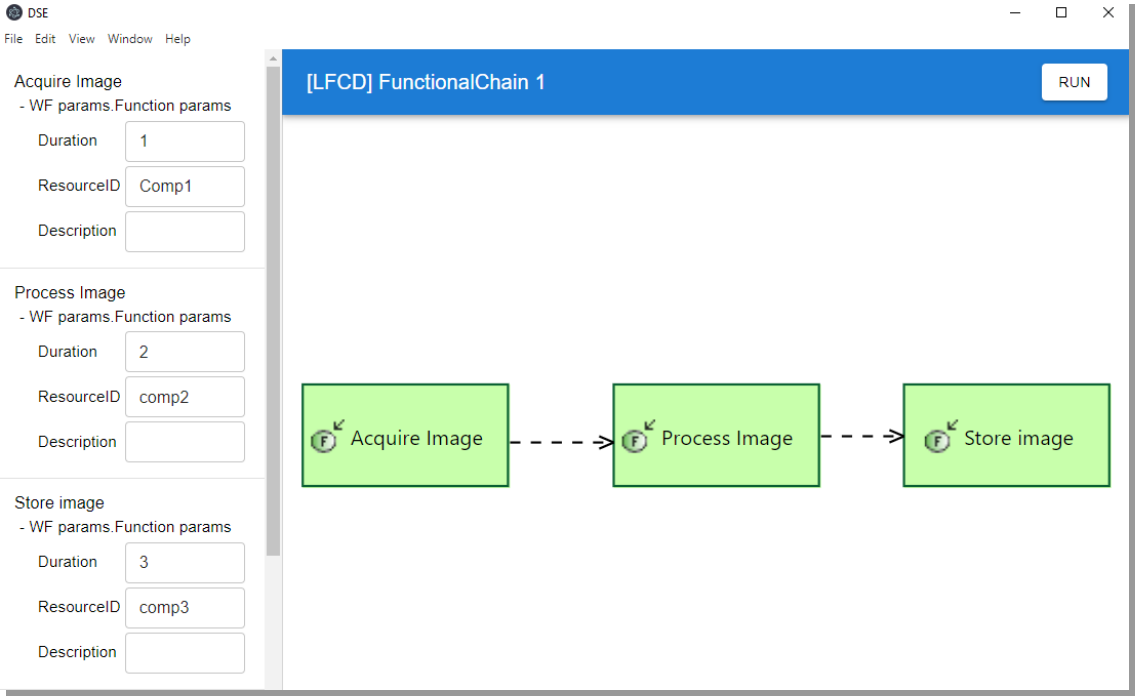
Work in progress

Design Space Exploration

- The user can vary timing parameters
  - min-max durations
  - iteration numbers
  - weights of conditional arcs
- Allocate functions to resources
- Defining duration as a function of involved components

Quantifying other properties (e.g., cost, reliability)

- Specifying parameters
- Exporting components for analysis using other techniques



Early example

Summary

We created a way to simulate Functional Chains with steps:

1. Create a ‘simulatable’ Functional chain
2. Export and run the chain in POOSL
3. Visualize result (TRACE4CPS)

To note:

- We’ll write a generic report
- We consider releasing the code, subject to discussions:
  - On licensing
  - Vision of project stakeholders

Some leads:

– *Interested about MBSE and high-tech industry?*
Check ESI report ‘MBSE in the high-tech equipment industry’
https://esi.nl/news/blogs/mbse-tno-report-2022

– *Interested in POOSL-TRACE4CPS native integration?*
Check TRANSACT project (https://transact-ecsel.eu/).

Conclusions

(Joost, 5 min)

ThermoFisher SCIENTIFIC

ESI

An initiative of industry, academia and TNO

Conclusions – Solution Space

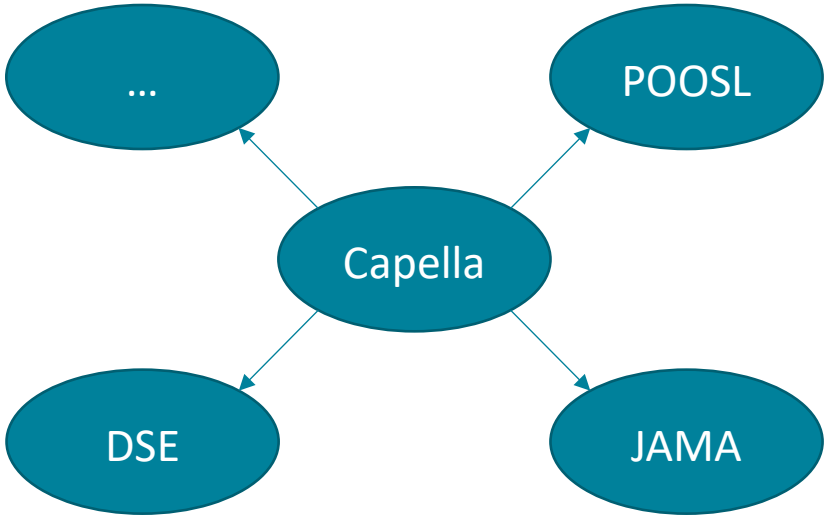
- **Proof of Concept delivered**
  - Simulation of Workflows
  - Capella Integration

- **Systems Engineering Goal**
  - Prevention of double recording of Information
  - Customer Value Maximization through design space exploration

Conclusions & Next Steps

- **Collaboration with ESI**
  - Short develop-review cycles
  - Regular checks on the deliverables and goals
  - Applying scientific methods in industry on specific cases

- **Next Steps**
  - Capella Model as authoritative source of truth
  - Tools that use Capella Model as input



Q&A (10 min)

ThermoFisher SCIENTIFIC

ESI

An initiative of industry, academia and TNO

Your questions and thoughts?

Your experience on:

1. Quantitative analysis of any Capella diagram, not just functional chains

2. Simulation/analysis of functional chains in general