

# Domain Specific Languages: towards increased adoption of model-based systems engineering

Marcel Verhoef / European Space Agency (ESA/ESTEC)

# Domain Specific Languages: towards increased adoption of model-based systems engineering (?)

Marcel Verhoef / European Space Agency (ESA/ESTEC)

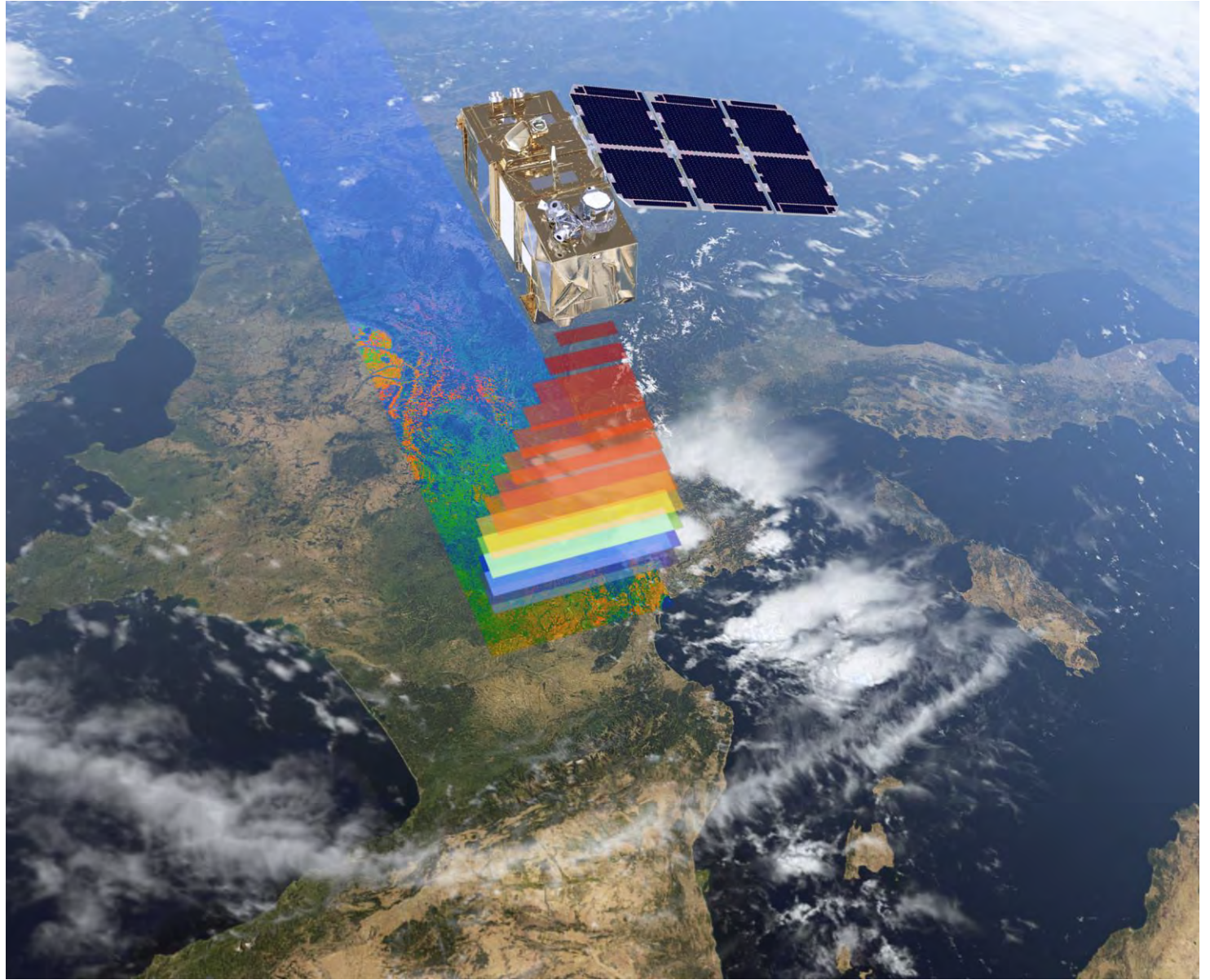
# Aerospace edition

*.... FIRST, LET'S GET IN THE MOOD! ....*



















# MBSE

*.... MODEL-BASED ENGINEERING OF SYSTEMS and SOFTWARE ....*

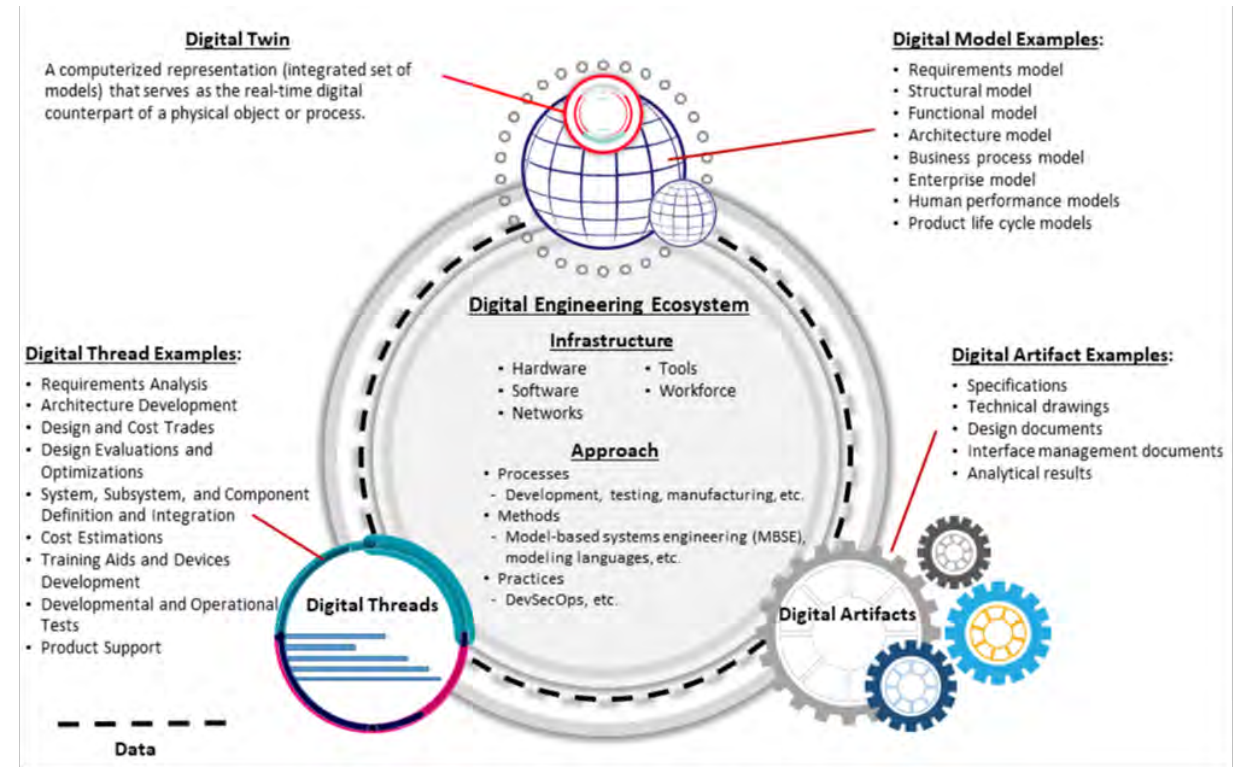




# MBSE – what do we mean, exactly

*“Model-based systems engineering (MBSE) is the formalized application of modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.”*

(INCOSE SE Vision 2020)



Ref: DoD Instruction 5000.97 – digital engineering (Dec 2023)

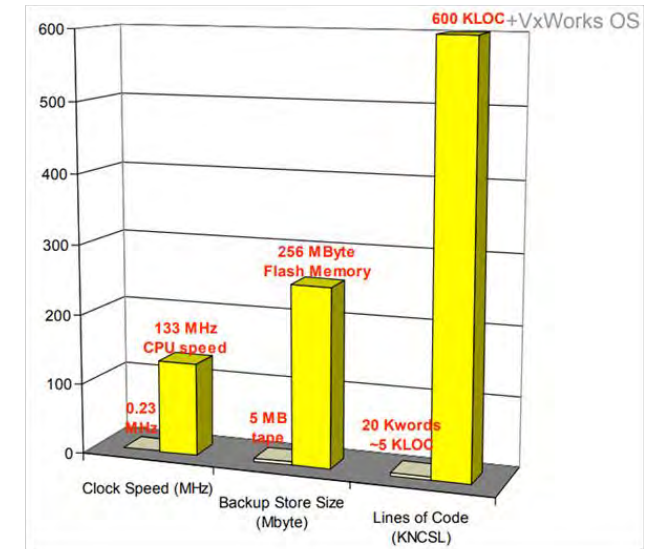
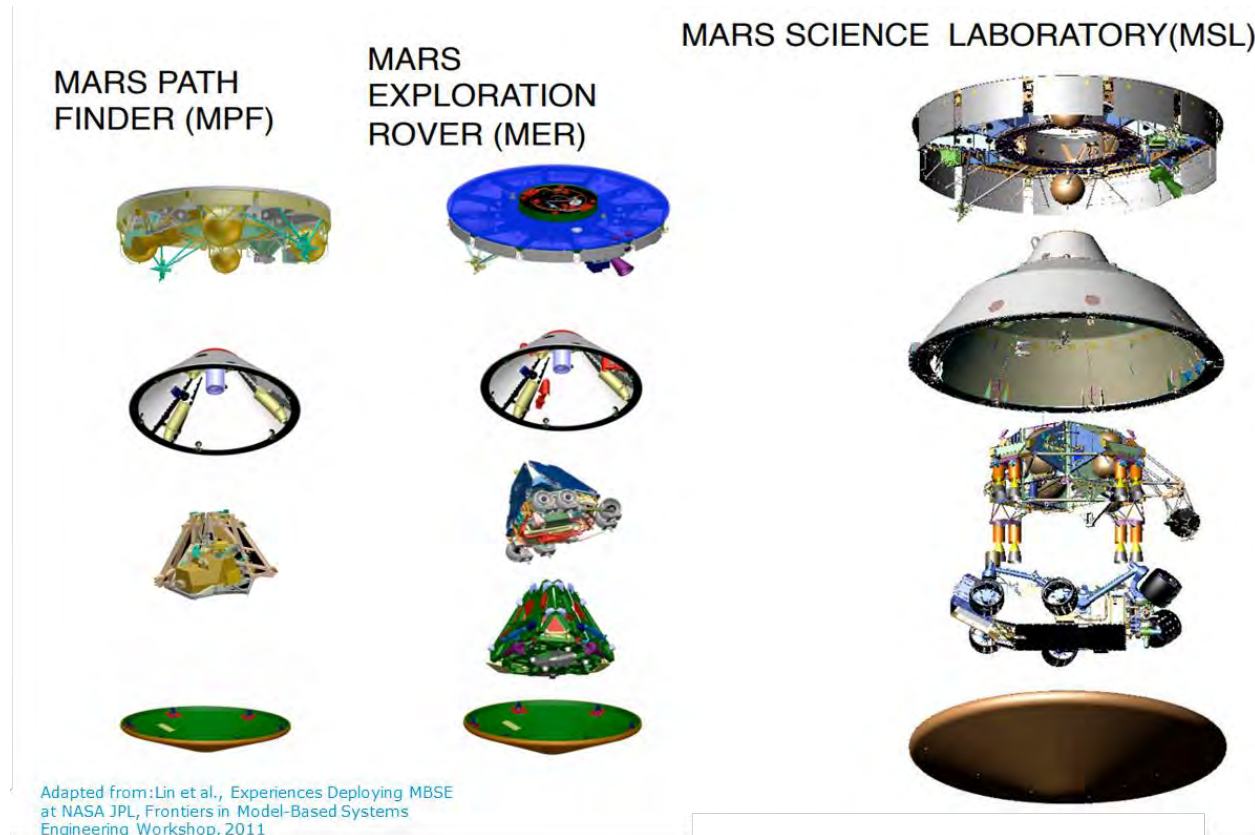
MBSE is the key enabling technology used to implement digital engineering

# MBSE – why are we interested?





# MBSE – why are we interested?



~500x faster CPU  
~50x more backup storage  
>>100x more code  
~0.3x software development time  
(540 person-months for MER /  
1609 person-months for Viking  
landers)

Complexity Increase ( $\approx$  doubling every 18 months)

# MBSE – why are we interested?

key to win the **time** – **quality** – **complexity** – **cost** battle is to *improve communication*:

- **Time**: we must *communicate more often* (iteration, access to consistent data)
- **Quality**: we must *continuously increase the confidence* of the information exchanged
- **Complexity**: we need to *succinctly communicate* (abstraction, depth, purpose)
- **Cost**: we need to *detect / prevent* potential *problems* as *early* as possible

model-based systems engineering (MBSE) addresses these concerns by:

- providing an *explicit notation to create models* (abstractions of the real world),
- providing means to *continuously verify the model* (to check internal consistency),
- providing means to *validate models early* (to check external consistency)

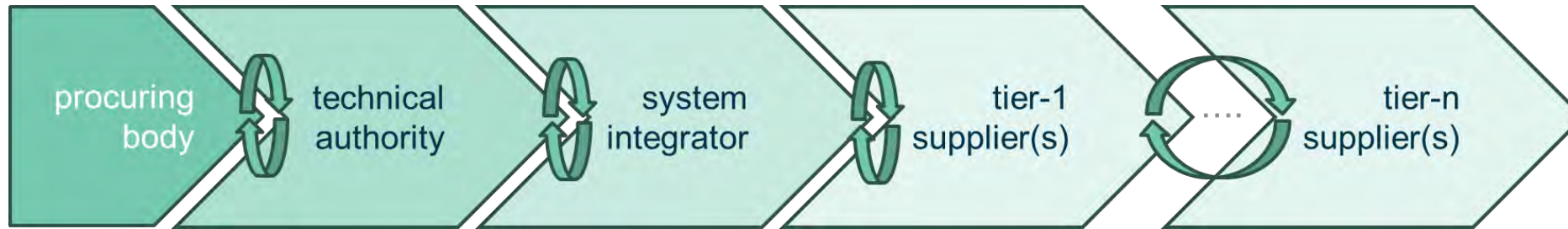
aim is to establish an *authorative source of truth*:

- across all disciplines (including pm and pa)
- across all life cycle phases
- across the supply chain

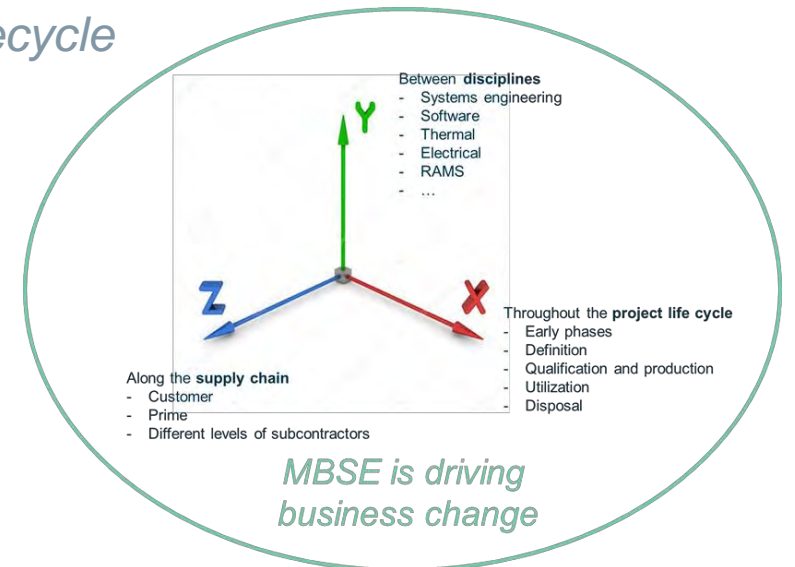
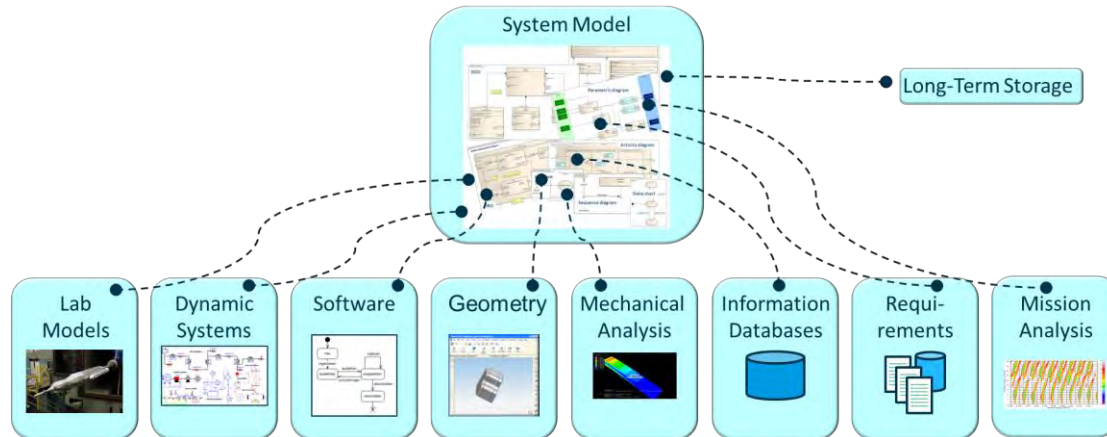


# MBSE – what we try to achieve

To address the needs of the entire space eco-system: to *avoid local optima* and *limit heterogeneity*



To support the *integration of the disciplines*, applied across the *entire lifecycle*

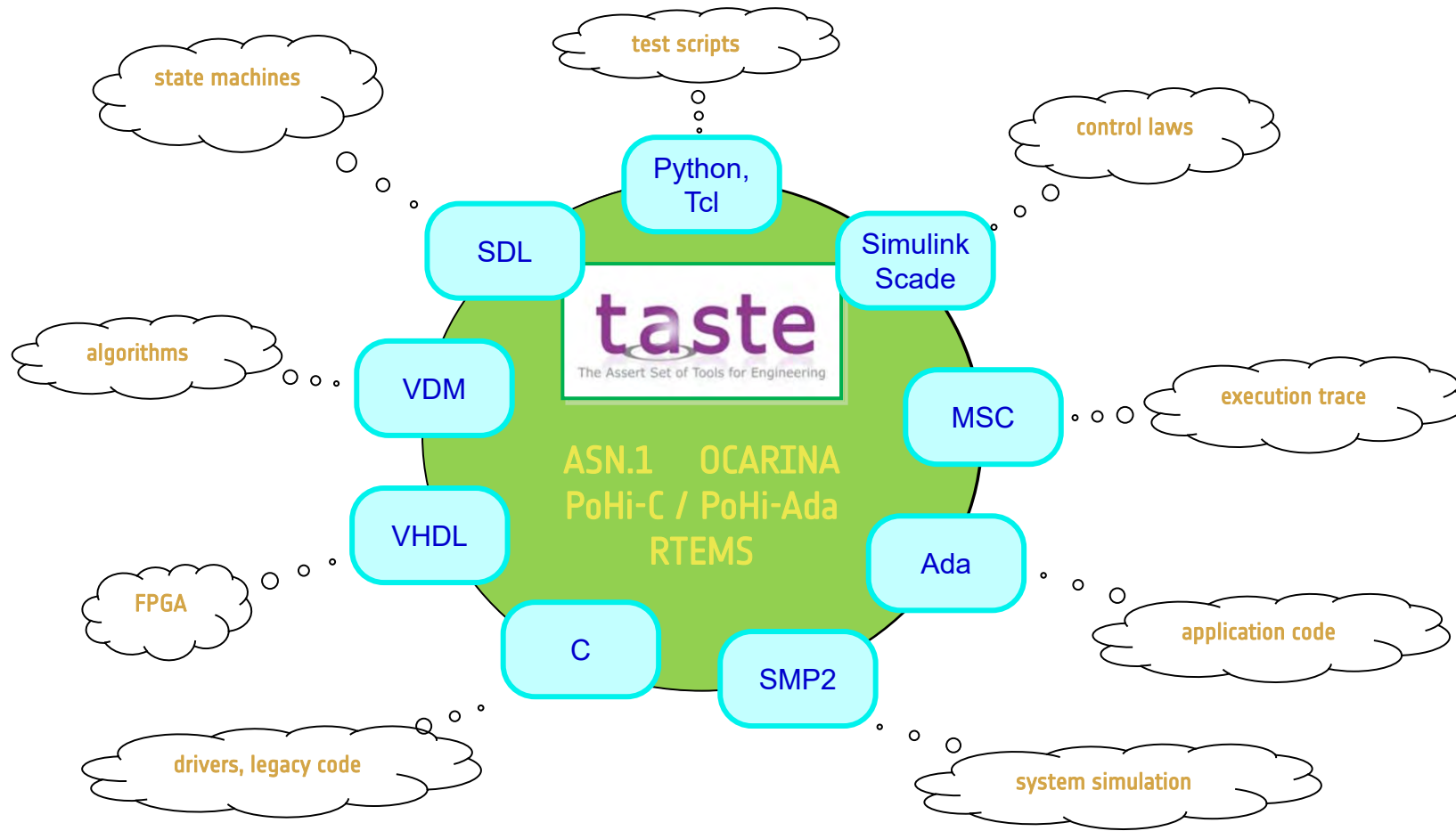


# Some examples





# Results achieved in software domain



**taste**  
The Assert Set of Tools for Engineering

**<https://taste.tools>**  
open-source software

# Results achieved in software domain

## ■ TASTE is:

- open-source tool suite for rigorous software engineering
- aimed at development of heterogeneous embedded systems
- focus on (but not limited to) space on-board software
- based on mature (formal) notations with long term support
- model-centric development with high levels of automation
- seamless interoperability offers DSL-like approach
- model synthesis towards wide range of target platforms
- robust tools maintained by active (but small) community





# Results achieved in software domain

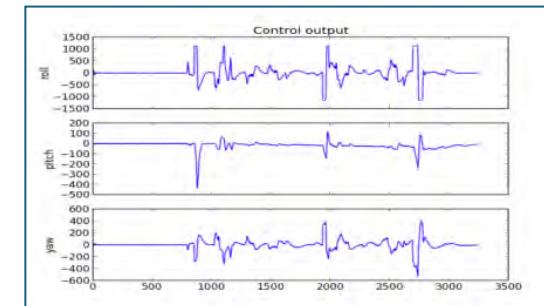
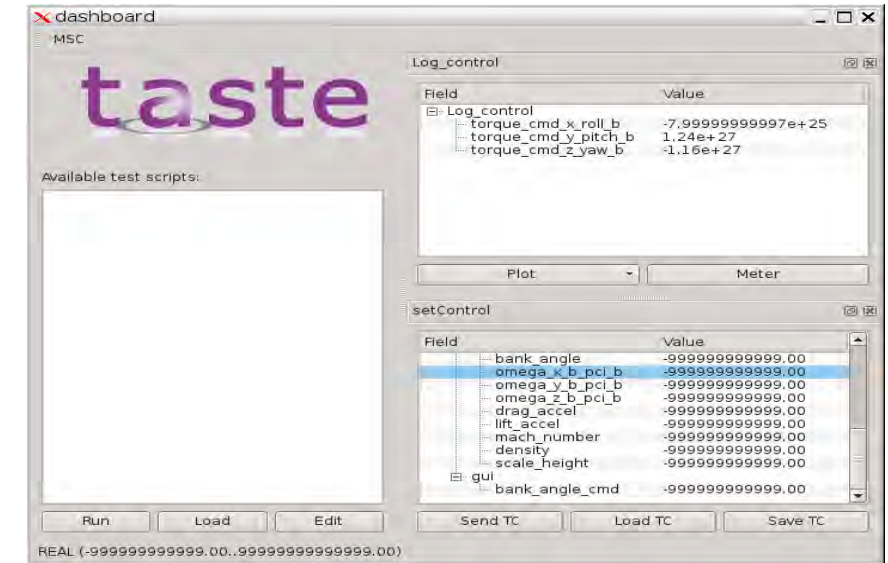
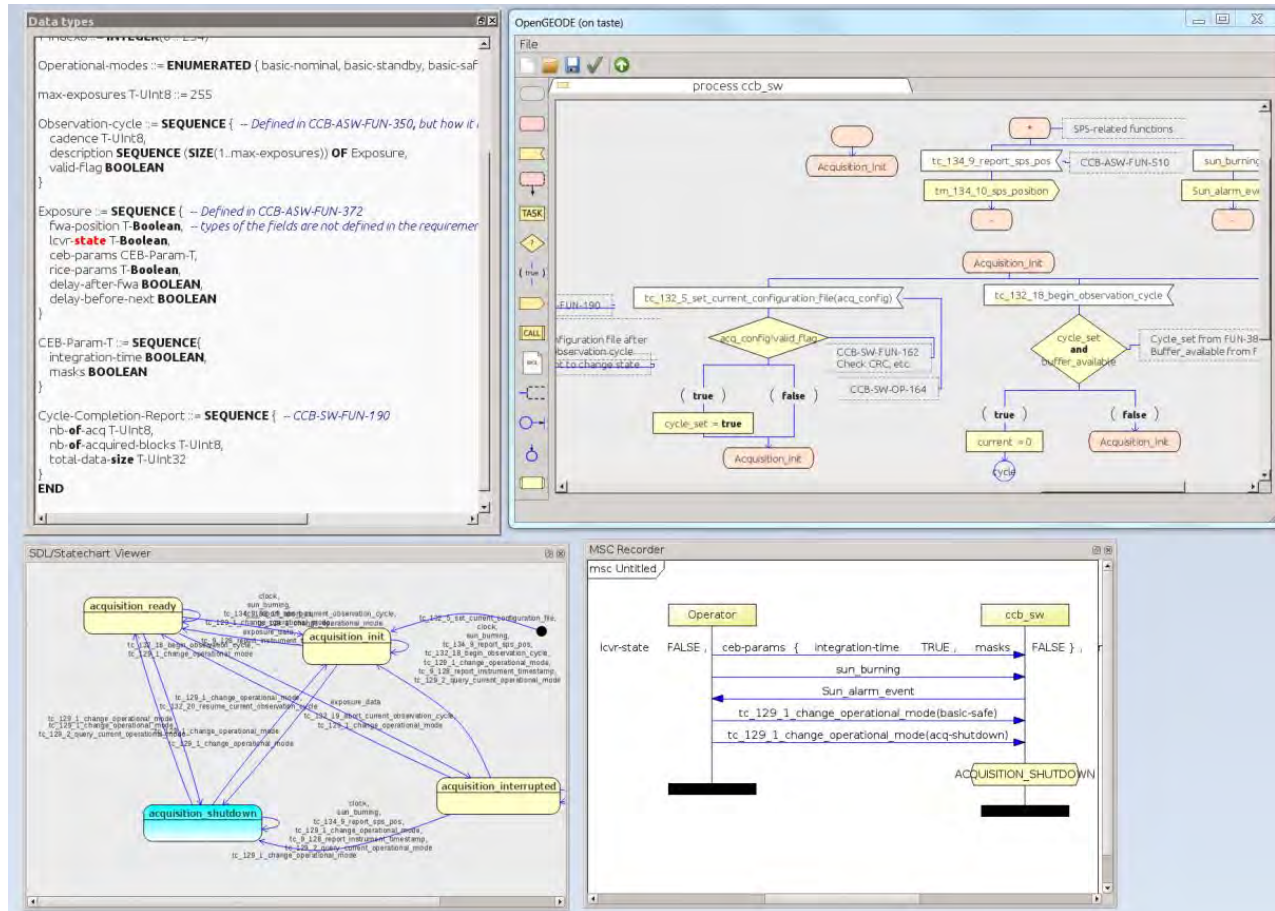
The TASTE development process consist of the following steps:

1. describe the system logical architecture (AADL) and interfaces (ASN.1)
2. describe the system behavior (SDL, SIMULINK, SCADE, VHDL, ADA, C, SMP2)
3. describe the deployment of functionality on the avionics (AADL)
4. generate code, build the system and download on simulator or target
5. monitor and interact with the system at run-time (test execution)

TASTE allows complementary analysis (re-)using the AADL model

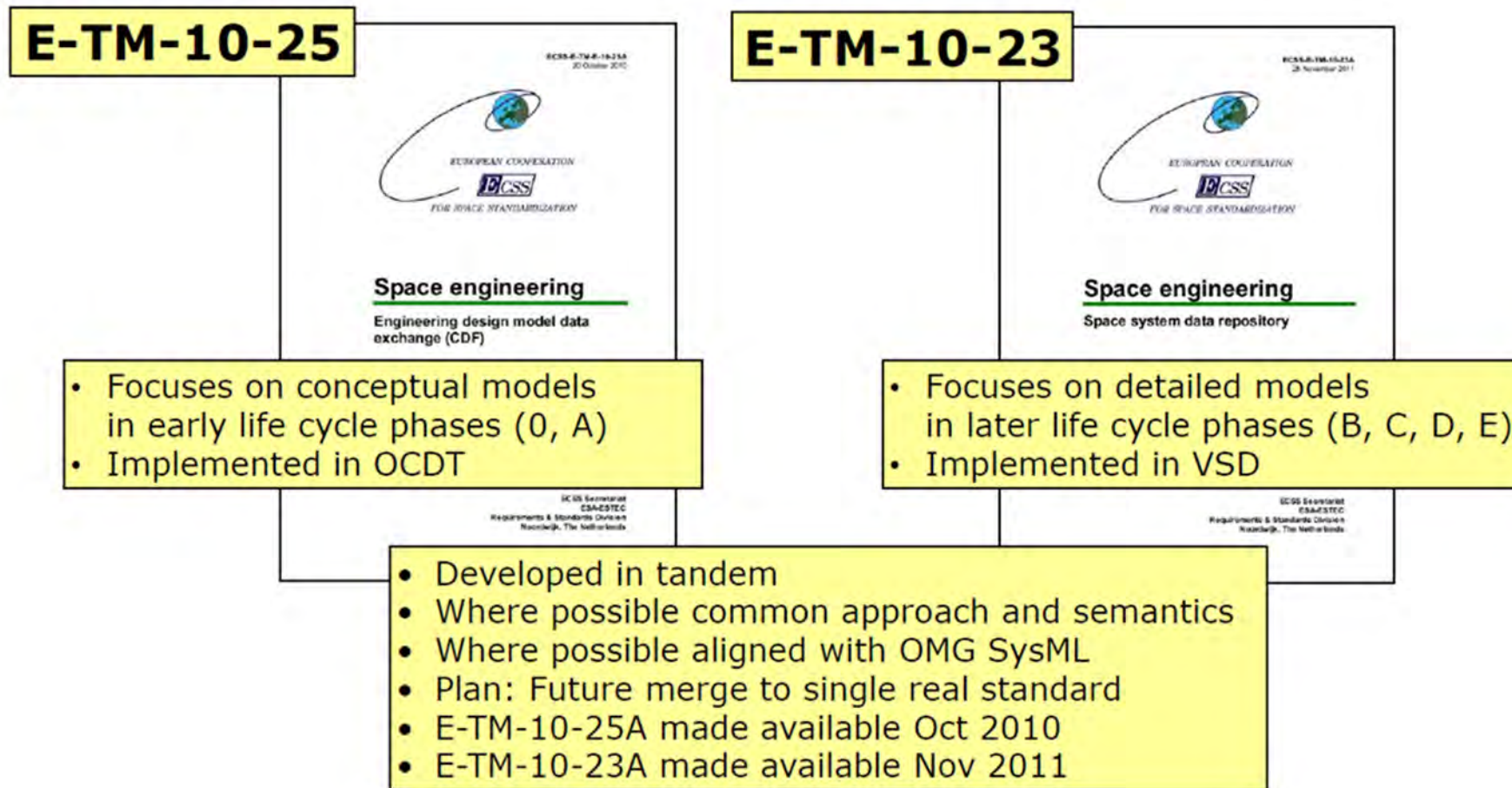
- Schedulability analysis using MAST and CHEDDAR tools
- Explicit fault behavior description using *System-Level Integrated Modelling* language
- Verification using COMPASS toolset (nuSMV, Markov-chain analysis tools)

# Results achieved in software domain

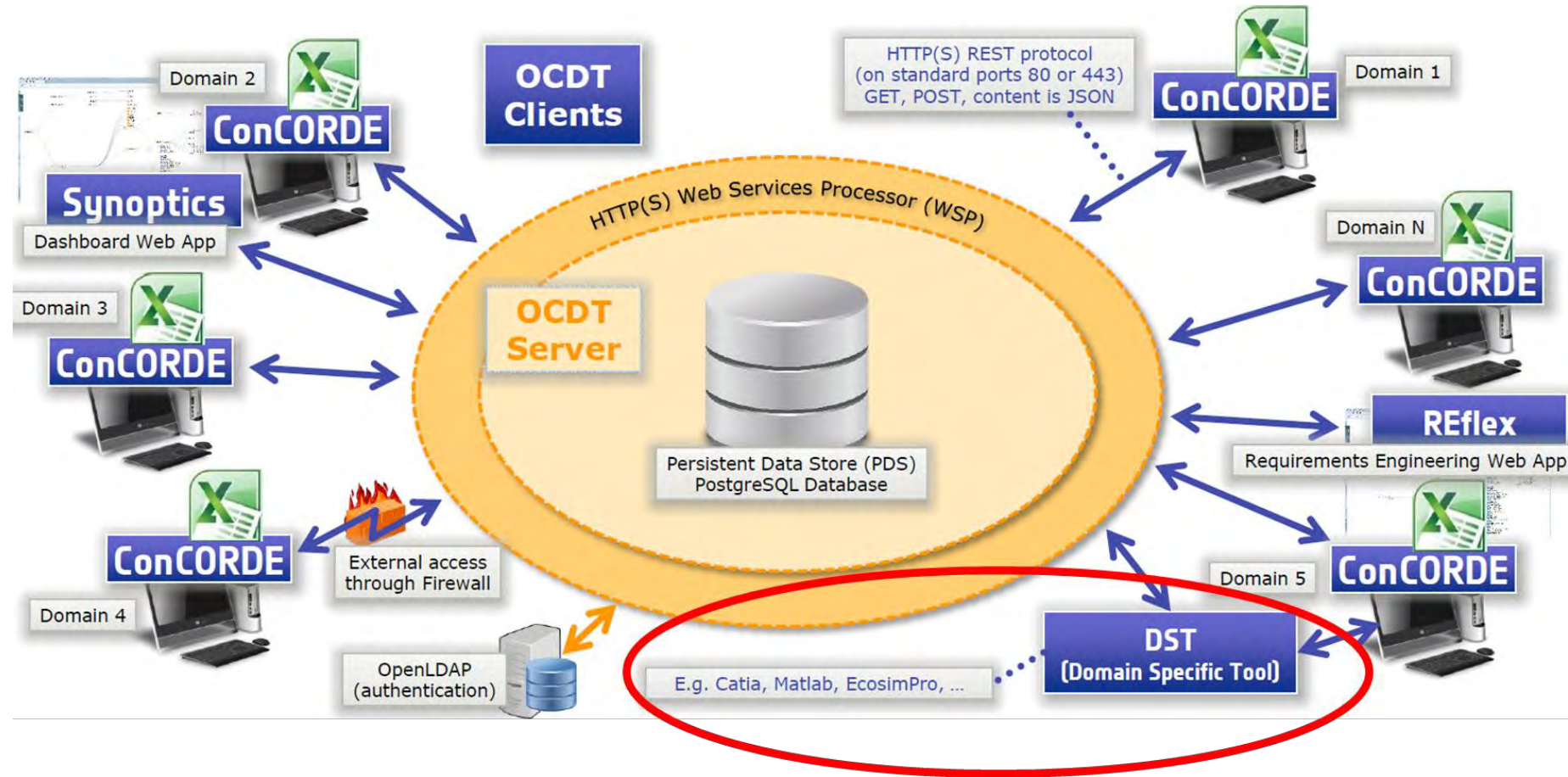




# Results achieved in systems domain

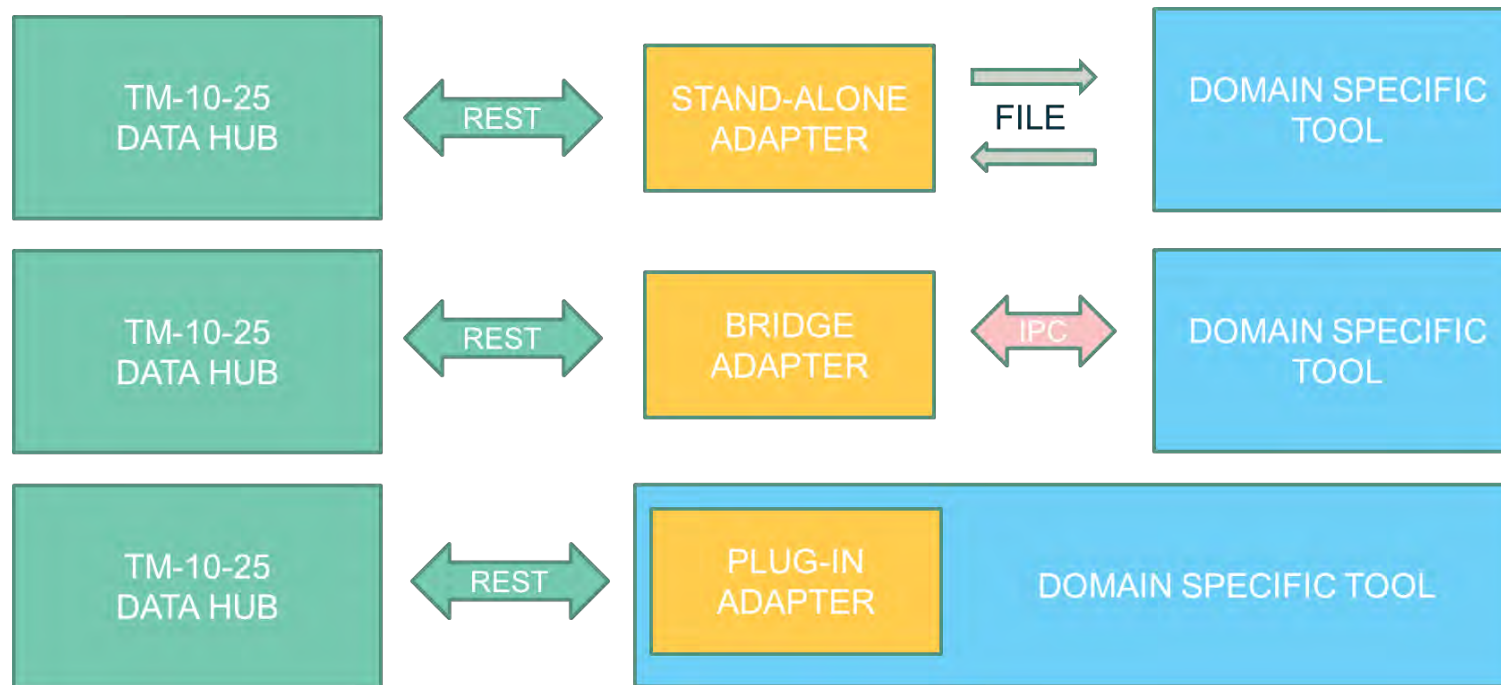


# Results achieved in systems domain



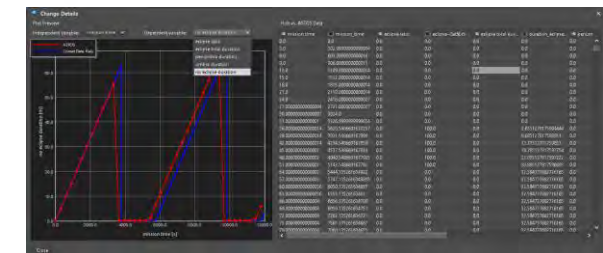
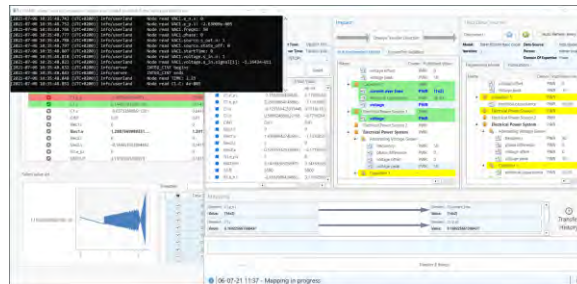
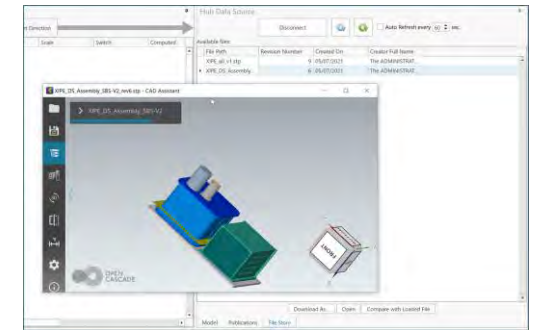
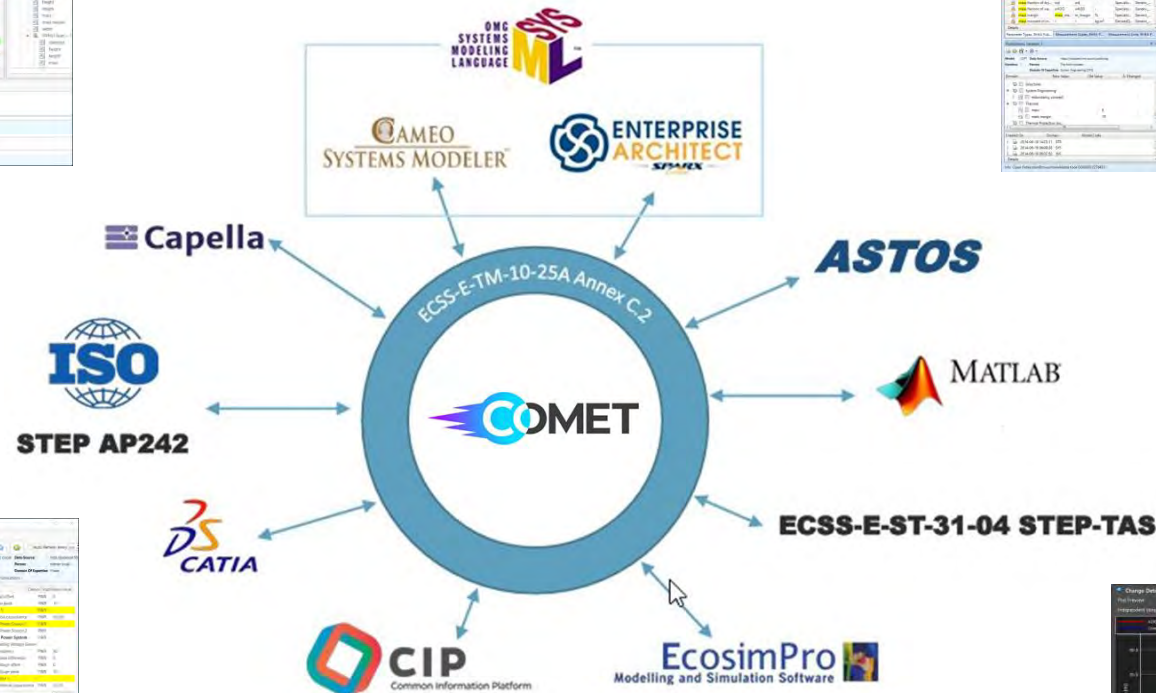
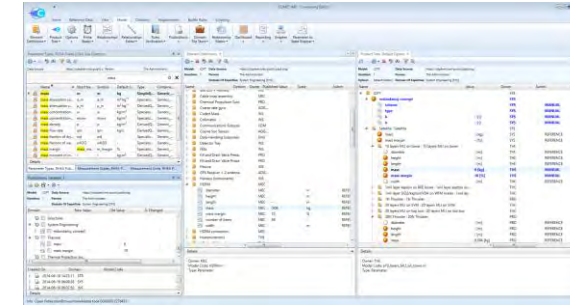
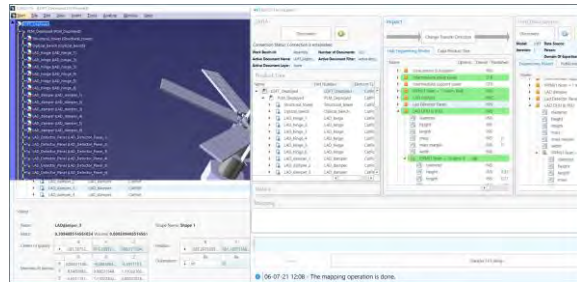


# Results achieved in systems domain



# Results achieved in systems domain

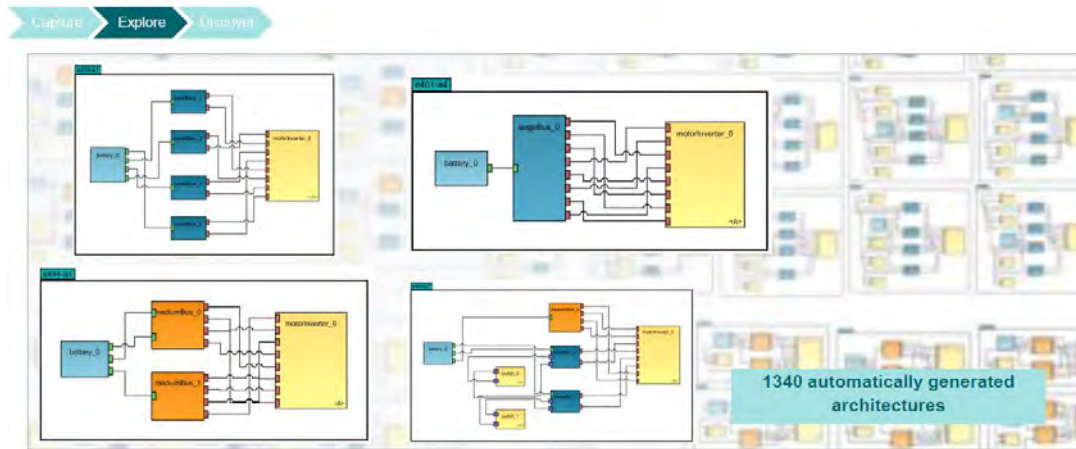
digital engineering hub pathfinder



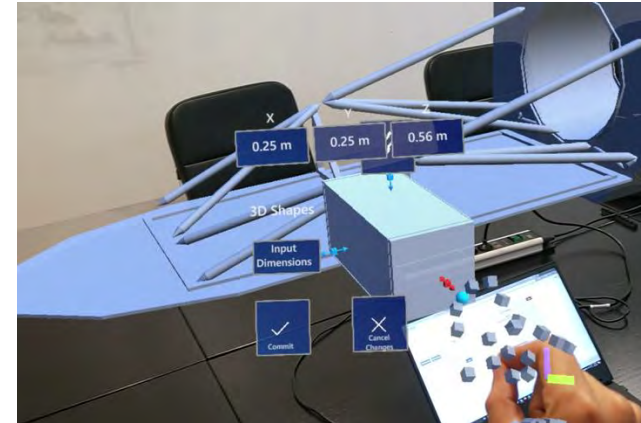
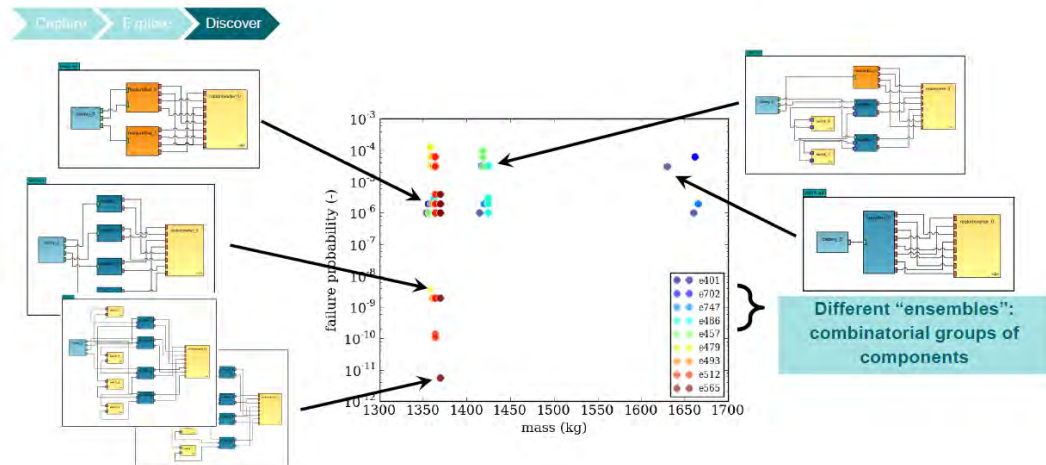
Paper: <https://doi.org/10.1109/MODELS-C53483.2021.00073>  
Presentation: <https://www.youtube.com/watch?v=nfop4kA5Bcc>



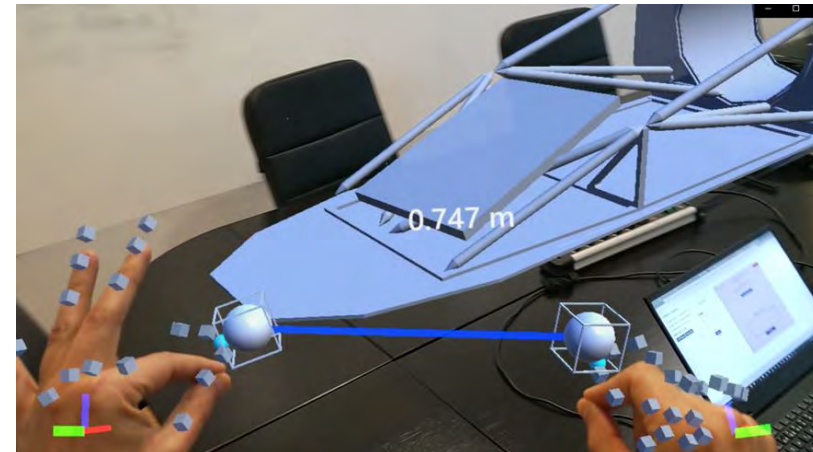
# Results achieved in systems domain



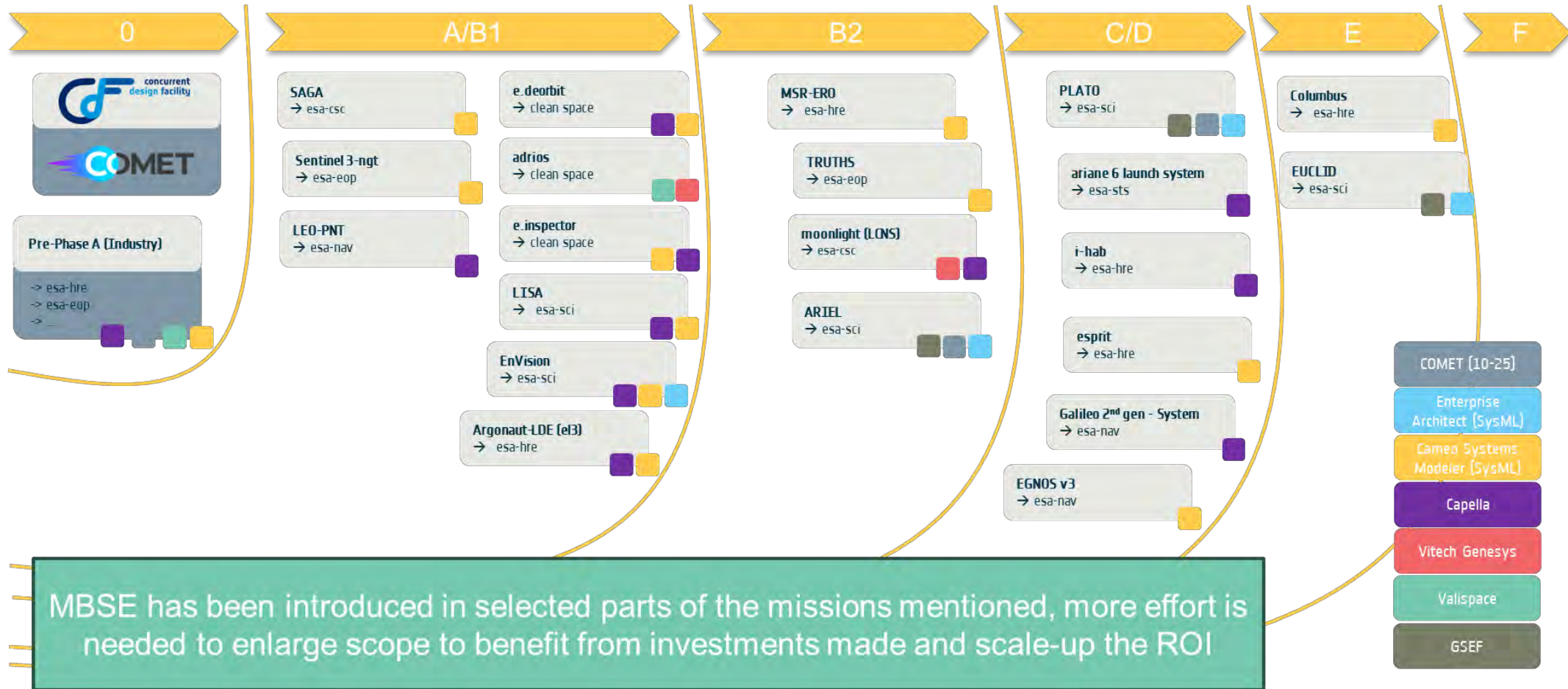
generative concurrent design



augmented reality for concurrent engineering (ARCE)



# What have we achieved to date?



# Our current challenges



## 1. Methodology and process

- MBSE affects engineering, project assurance, project management and procurement; impact on business is still underestimated (change takes time)
- Successful extended enterprise requires **uniformity of process and data**

## 2. Languages and tools

- Plethora of capable tools available, but **not fully exploited when used**
- **Level of investment and pace of change do not line up with need**

## 3. Infrastructure and data governance

- Demanding requirements on infrastructure to **deploy MBSE effectively**
- Digital collaboration natural tension information security and data protection

## 4. People

- **Large group of users alienated by poor user interface experience**
- Digital engineering requires development of new roles and skills



# MBSE 2024 key take-aways (1)



Change My View: Model Based

reddit.com/r/systems\_engineering/comments/1bpavpi/change\_my\_view\_model\_based\_systems\_engineering\_in/

Loonwerks ESA software # Decision Tree Adobe Acrobat

reddit

r/systems\_engineering • 2 mo. ago Rhedogian

### Change My View: Model Based Systems Engineering in 2024 is at best overhyped, or is at worst actively dying

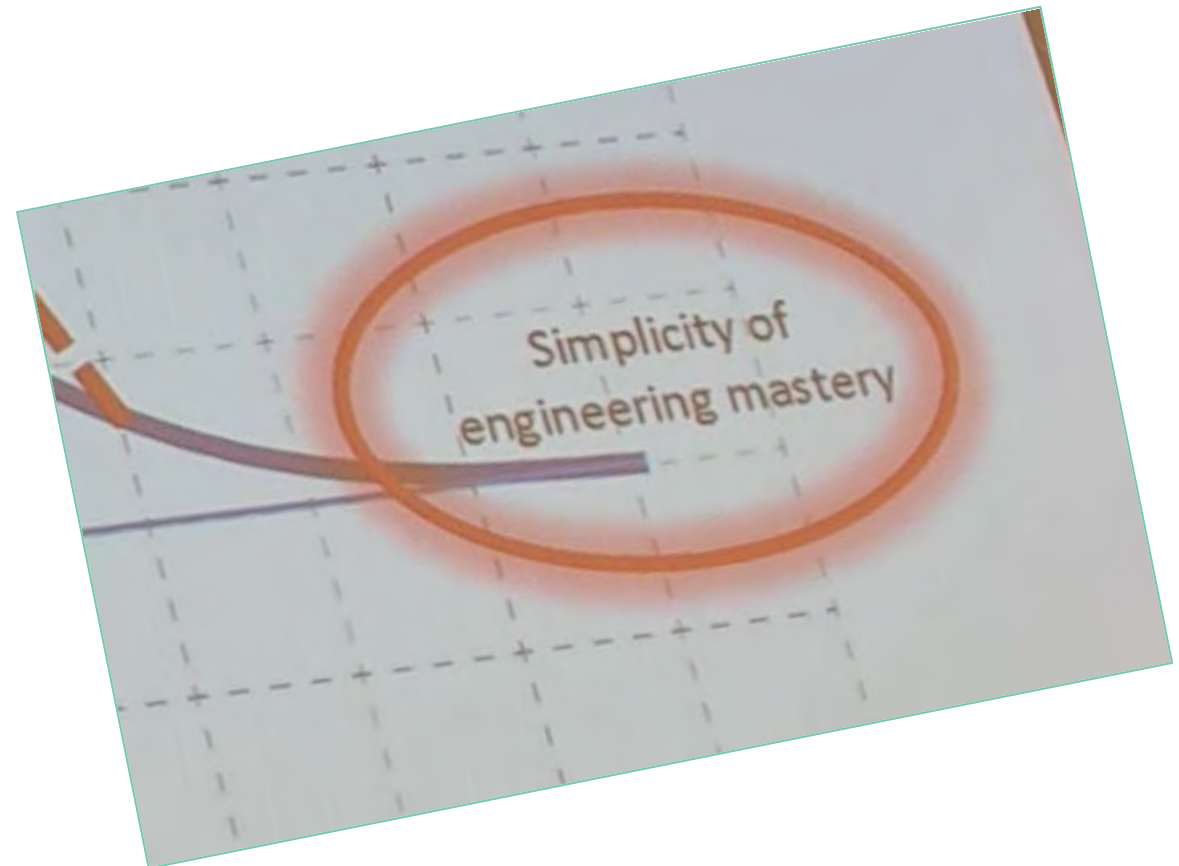
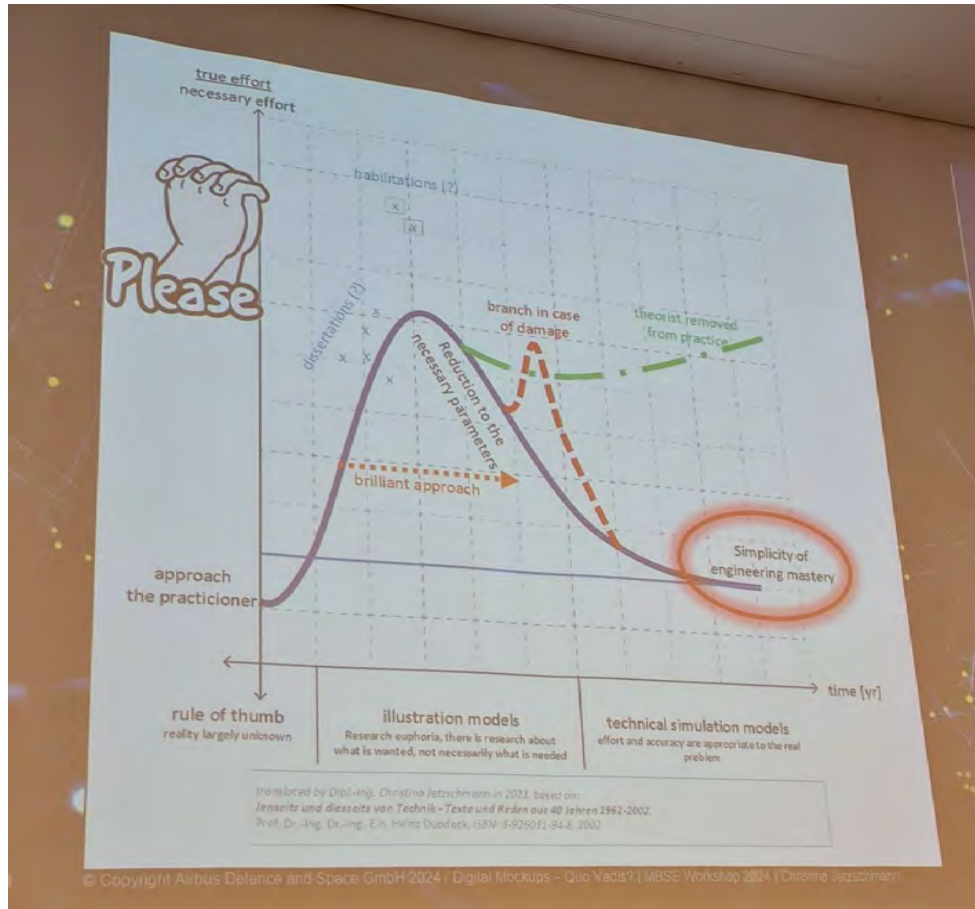
I know the title is a little controversial but I feel like this conversation needs to be had now within the community. For the past couple of years I've felt like more and more of a scam salesman trying to push this MBSE stuff onto people, and at this point it feels like it's time to let the reality of the situation have it's time in the light.

About me:

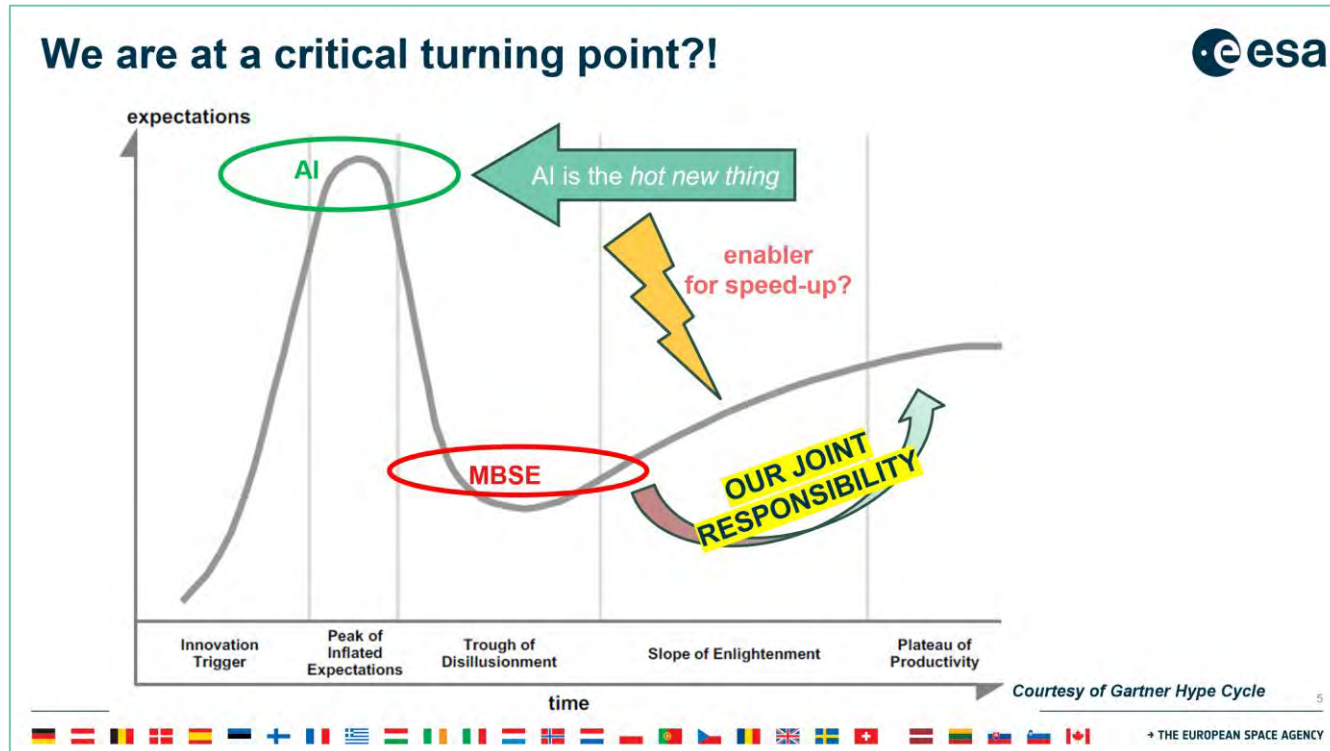
- Systems engineer for 5 years with a focus on MBSE
- Have done straight MBSE since undergrad and through my MS degree as well (BS/MS Aerospace Engineering)
- Currently holding the OCSMP-MBI certificate
- Have used Cameo almost exclusively, as well as quite a few different 3rd party integration suites (Syndeia, SBE Vision, Excel, etc.)
- Have attempted to push SysML in at least three different industries (commercial aerospace, automotive/tech, DoD aerospace)

My breaking point with letting go of MBSE has come pretty recently, and I've done my best to remain hopeful in the concept despite my doubts, but at this point I'm no longer confident in MBSE's ability to be a transformational force in system design as it's been sold.

# MBSE 2024 key take-aways (2)



# Why I am here at your conference



- What can I learn from the langdev community?
  - Simplified model creation
  - Improved model interaction
  - Model transformations
  - Semantic mappings



# What's in it for you

- ESA is targeting new research projects in 2025 – 2026 in this area
- Active and well-organised MBSE community in aerospace
- MBSE 2025 announcement to follow soon (Vilnius, November)



# MBSE in European space domain

- Slides and presentation recordings available on-line at:
  - MBSE 2020 workshop, see <http://mbse2020.esa.int>
  - MBSE 2021 workshop, see <http://mbse2021.esa.int>
  - MBSE 2022 workshop, see <http://mbse2022.esa.int>
  - MBSE 2023 workshop, see <http://mbse2023.esa.int>
  - MBSE 2024 workshop, <https://mbse2024.welcome-manager.de>



<https://mb4se.esa.int>



# LET'S ENGAGE!

*looking forward to an exciting conference!*

Marcel.Verhoef@esa.int