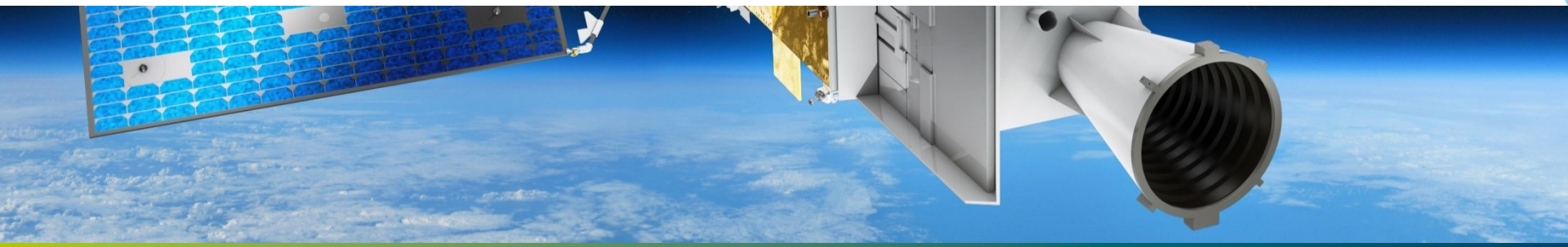
OHB System AG
Andreas Wortmann
28. September 2020, ESTEC, Virtual Setting

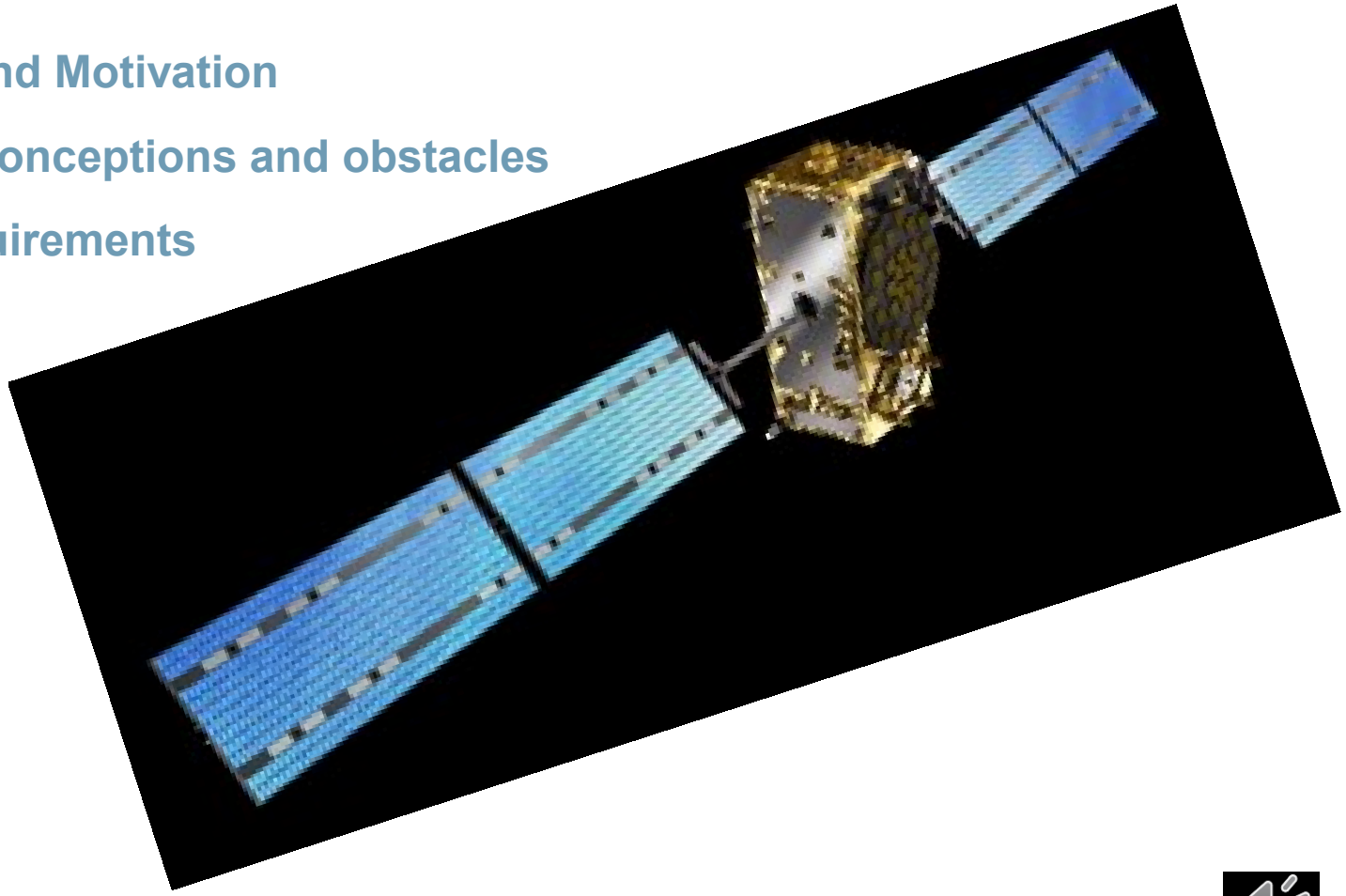


SPACE SYSTEMS

EXPERIENCES AND EXPECTATIONS WITH MODEL BASED SYSTEMS AND SOFTWARE ENGINEERING

Agenda

- Introduction and Motivation
- Common misconceptions and obstacles
- Essential Requirements
- Outlook



Introduction and Motivation

- Projects for a long time have engaged ideas of [illegible]ed engineering
  - in various flavors
  - in order to optimize quality and efficiency of development.

- ADCSS 2016 some examples have been presented
  - Software Development (UML, Rhapsody, EA)
  - Hardware/Software Codesign (focus in Interface Definitic
  - Timing Analysis (Mathematical Model)
  - Mission Analysis and Breakdown (Capella)
- Since than more projects engaged / related with MBSE
  - Requirements, Simulator Development
  - PDM

- Most activities have been carried out in isolated applications
  - not sharing data, structure and processes

→ Lessons Learned boil down to the presented Comments and Requirements

Common Misconceptions a

Language and Wording

The term 'model' is overloaded

- Simulator (modelling some behav
- Spacecraft model like the EM, PFM

MBSE commonly is mistakenly understood to be …

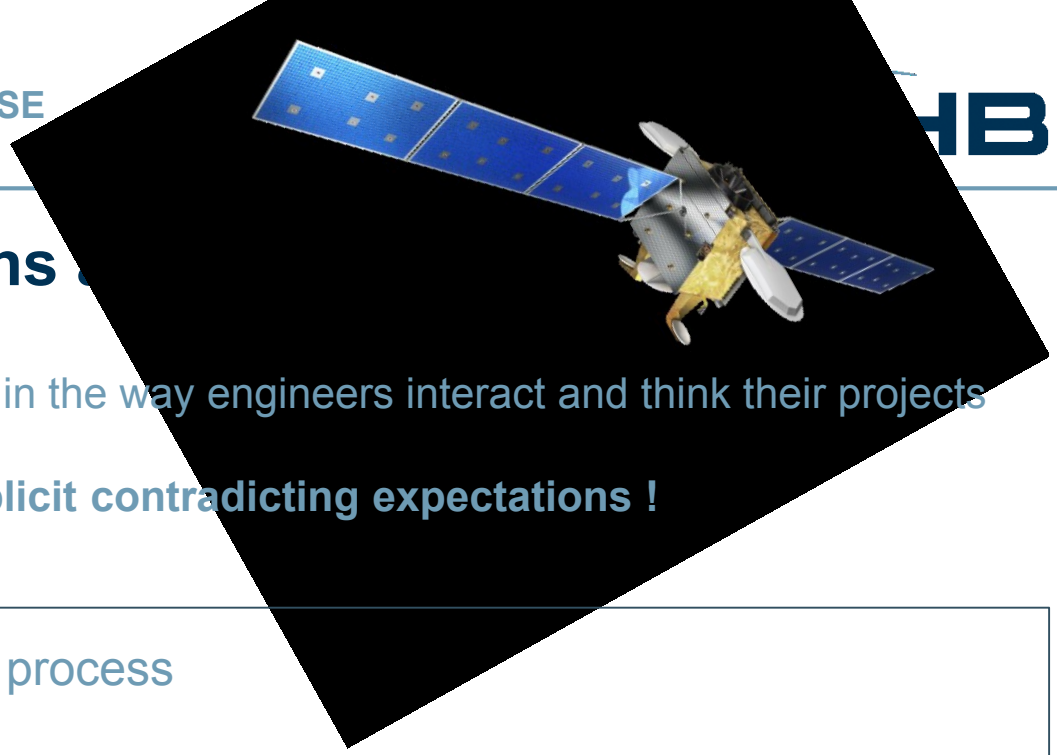
- Synonymously with “using UML or SysML”
- Something graphical

New terms with unclear or overlapping semantics introduced

- digital clone, digital twin
- digitalization, digital continuity
- …

Their use sometimes seems rather arbitrary

Common Misconceptions



MBSE calls for a significant change in the way engineers interact and think their projects

But there are implicit contradicting expectations !

Introducing MBSE is a long-term process

↯ Implicit expectation that the Return of Invest is quickly achievable in short term

Expectation that everything will be better, more efficient and cheaper

↯ Assumption that we don't have to change our way of thinking and

↯ Assumption that we don’t need to invest in related development processes

Common Misconceptions

Elements (source code, docs, configuration) with heritage can be fully reused

- Different tools and infrastructure call for different artifacts

Replacing artifacts with heritage by models inadvertently leads to loss of heritage

- Prior knowledge (good design pattern) are rigorously applied to all artifacts
- Heritage is in reuse of concepts and pattern rather than in source code
  → Possible by raising the level of abstraction when actually implementing

Essential Requirements

Some Basic Requirements against an envisioned MBSE Environment

- Scale to Size and Complexity of Future Systems
- Collaboration of many different Engineering Disciplines

Two (apparently) contradicting needs: an engineer …

- ... requires a stable baseline to base his work on
  (continuous changes will stifle progress)
- … requires the latest information in order to not design the wrong system

Both is required:

→ Transaction-based collaboration (e.g. git in s/w Eng. with branch/diff/merge)

→ Collectively edit a model in a “google docs style” (e.g. concurrent engineering)

Essential Requirements

User interface (UI) is important for acceptance and efficient operation

- 1000-button menus for doing simple jobs are not suitable
- Customized to the engineering task (only see the buttons and options needed)
- Tooling must remain live even with very large models being handled
  - Interactive failure and consistency checks and simple analyses
  - No fading out while editing
  - No “make”-button to trigger long lasting activities

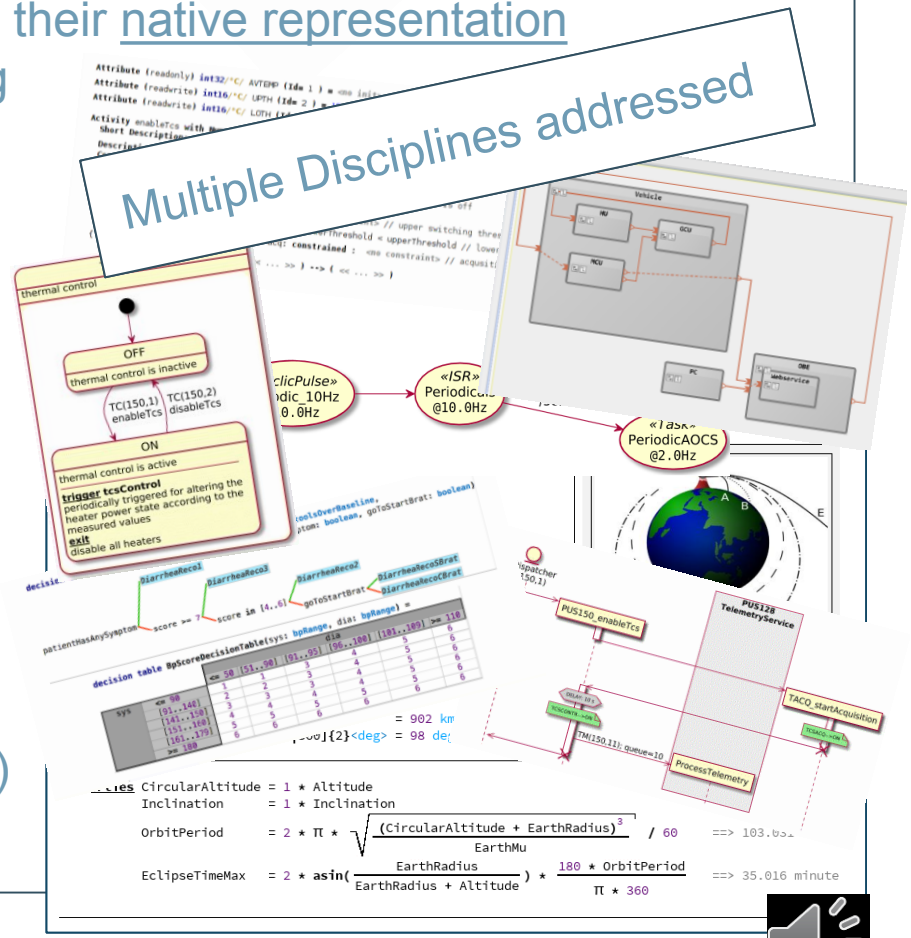
Essential Requirements

Edit domain specific aspects of a system in their native representation

- More than what's achievable when using UML/SysML or item trees

- Multiple paradigms including
  - prose-style (high level requirements)
  - declarative (type system)
  - behavioral (test, math expression)
  - structural (deployment) languages

- Multiple notations including
  - guided text (requirements)
  - tabular (lookup)
  - graphical (state machine, deployment)
  - symbolic (math, chemistry) notations

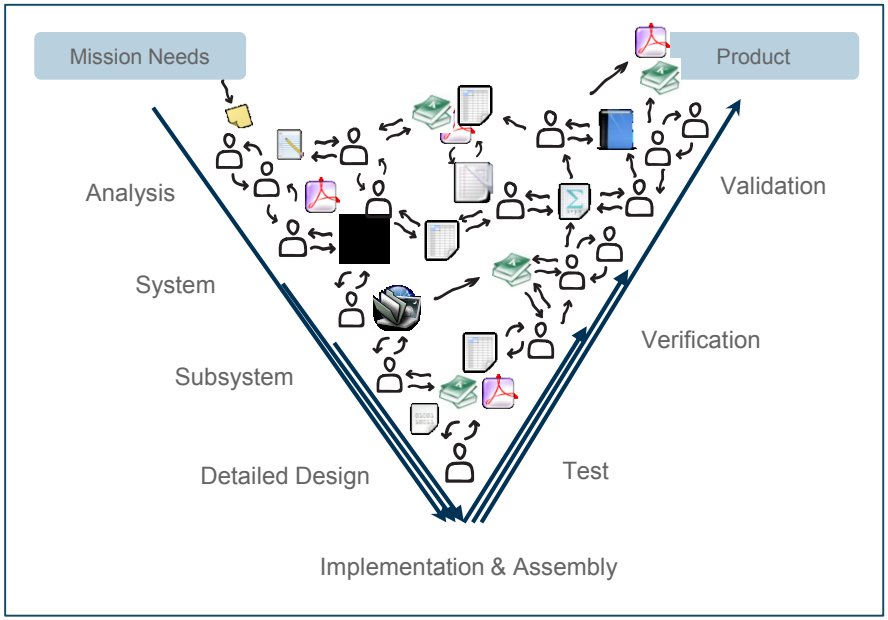


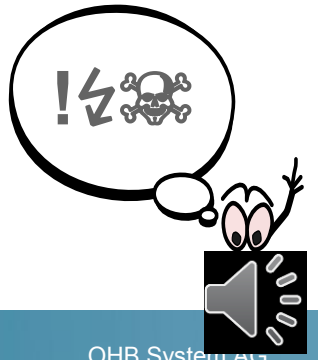
Outlook

How could such a future engineering framework look like ?

Now/Past:

- Discipline Specific Tools (CAD, Thermal, FMECA, Software, Radiation …) export into and import from General Purpose Formats (Excel, Word, CSV …)
- Data Exchange between Engineering Disciplines commonly is exchanged via such General Purpose Formats

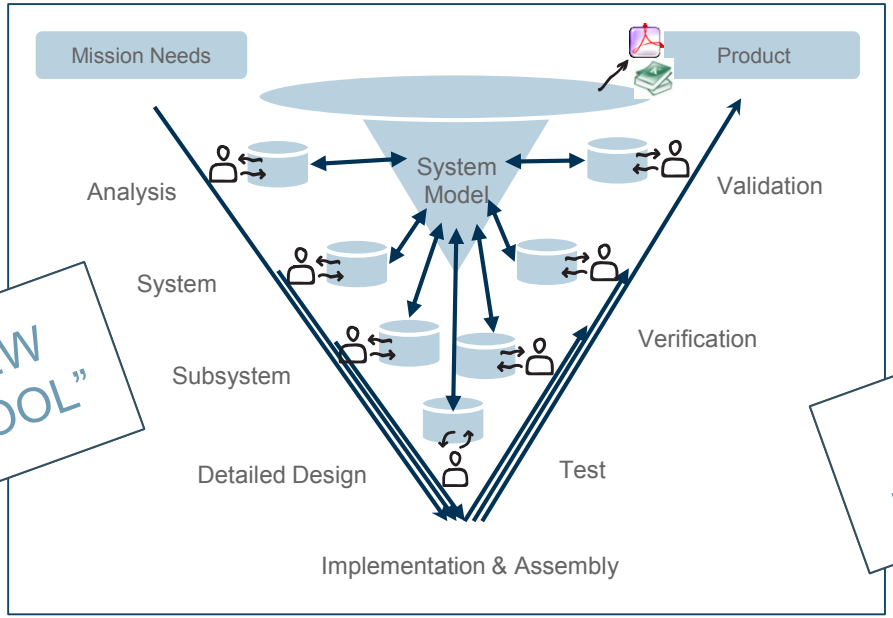




Outlook

1st Improvement:

- Substitute General Purpose Formats with a set of shared models
- Discipline Specific Tools operate on their own local models (MB Engineering) and may be transformed and exchanged among Tools
- In an MBSE environment data common to multiple disciplines is shared in a common model (need to synchronize with local models)



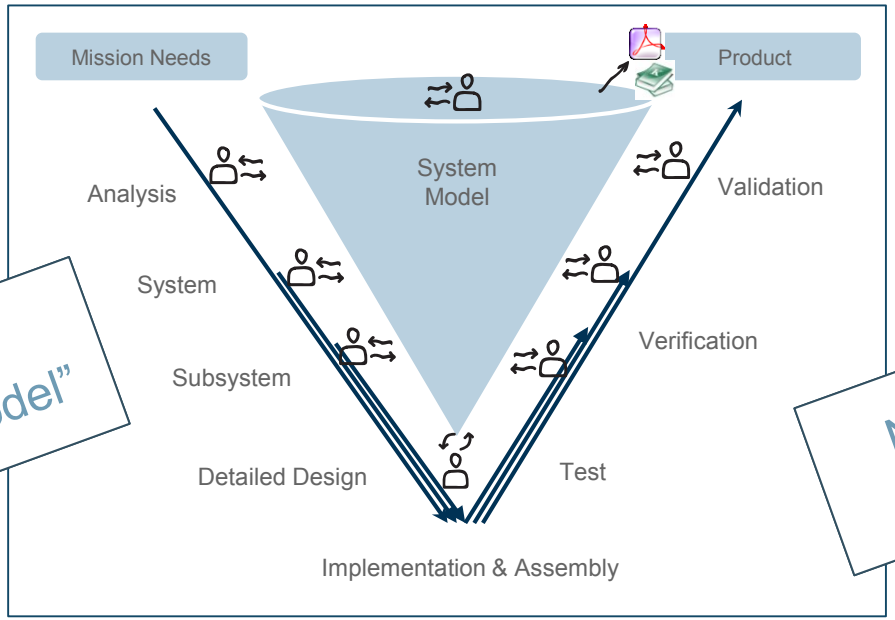
No "ONE BIG NEW ENGINEERING TOOL"

PDM implements many of these needs

Outlook

2nd Improvement:

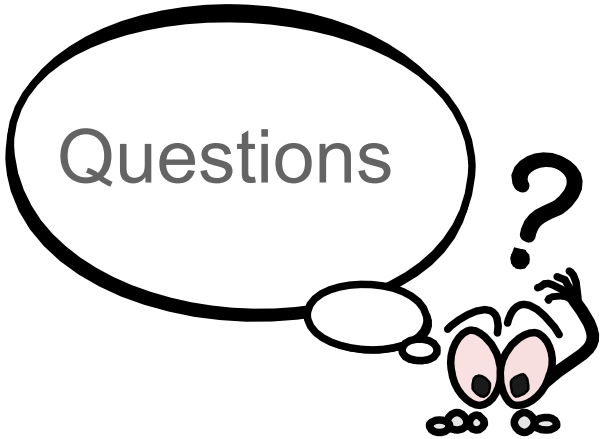
- Substitute the set of models with a single common model
- Discipline Specific Tools directly interact with the common model (requires API to support transaction-based or continuous exchange of data)
- Repo/Hub to provide collaboration features
- Workbench for direct model maintenance



Focus on a "Single Shared Model"

No common Tool, but common Framework

OHB System AG
Andreas Wortmann
28 September 2020, ESTEC, Virtual Setting



SPACE SYSTEMS

EXPERIENCES AND EXPECTATIONS WITH MODEL BASED SYSTEM AND SOFTWARE ENGINEERING