



34th Annual **INCOSE**
international symposium

hybrid event

Dublin, Ireland
July 2 - 6, 2024



Robert Hällqvist, Erik Herzog, Johanna Wallén Axehill, *Saab Aeronautics*
John R. Palmer, *The Boeing Company*

Excuse me Sir/Madam, which Model?

www.incose.org/symp2024 #INCOSEIS

The main messages

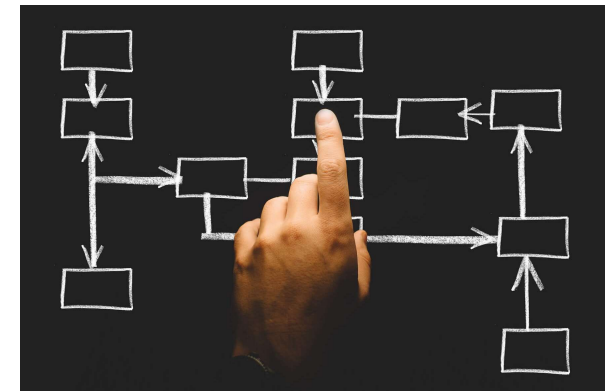
1

Why **one single model is not enough** to capture every aspect of a system throughout its development



2

The **need for a unified terminology**, covering all phases and usages of models



Boeing & Saab

Some of our products



Boeing 787



Gripen E



T-7A

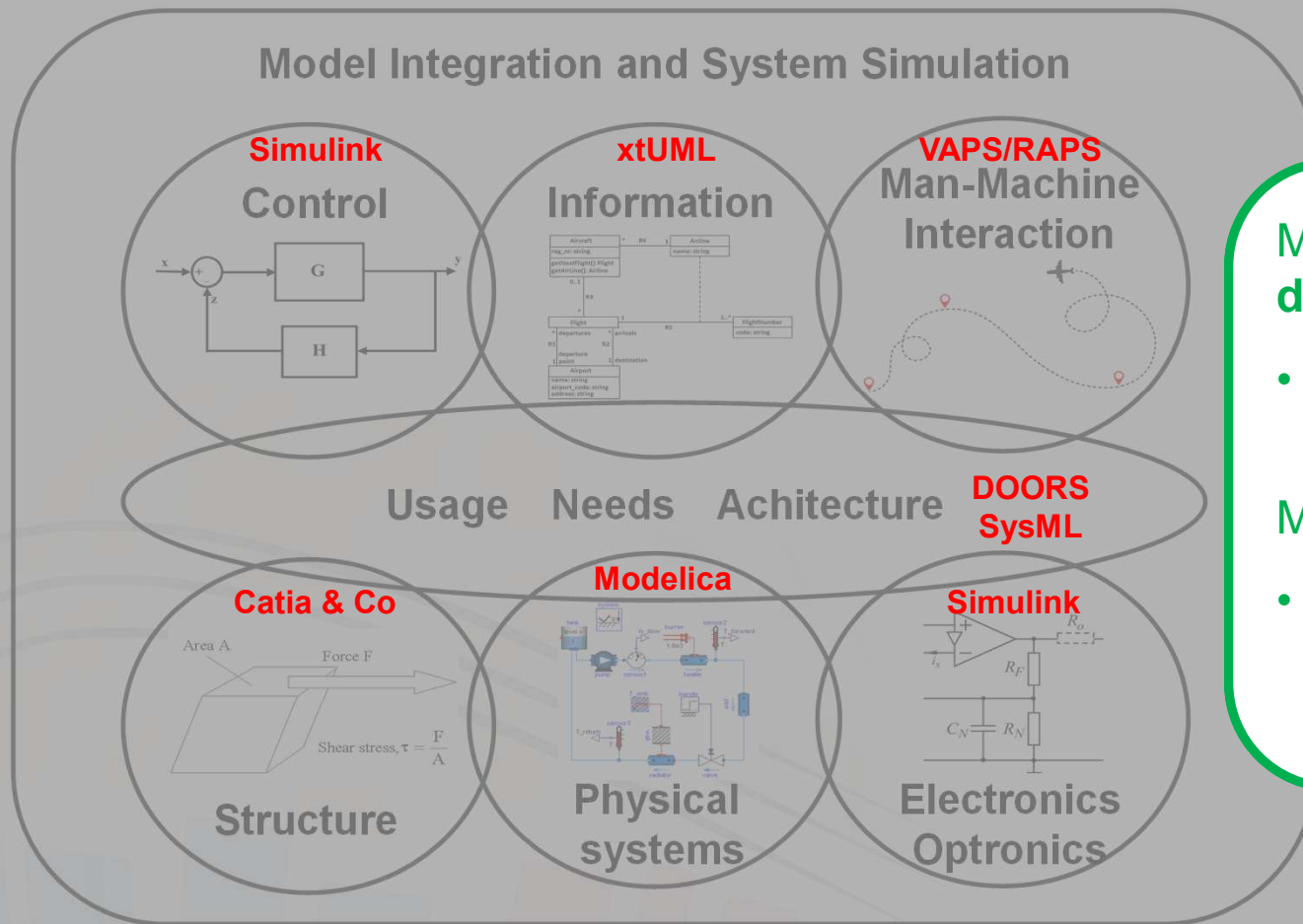
www.incose.org/symp2024 #INCLOSEIS



SAAB



Background MBSE Domains – Gripen example



Many different languages and domains

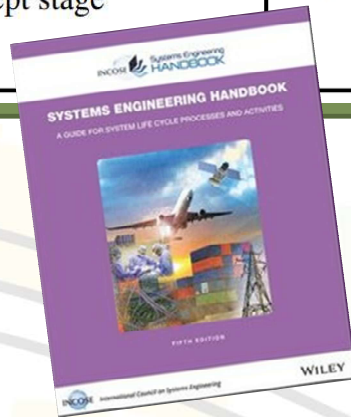
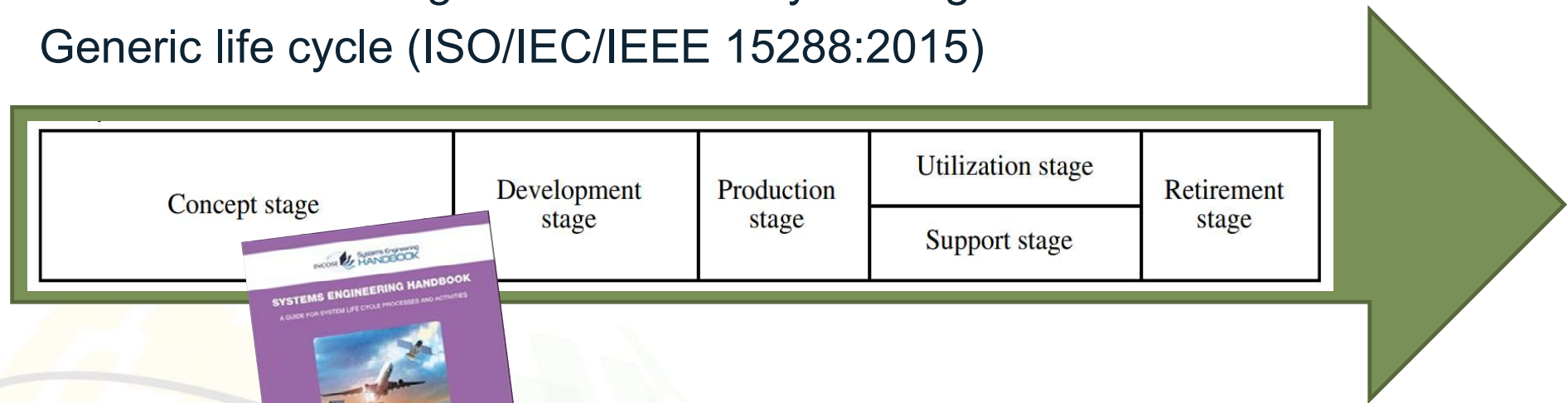
- Clearly defined purpose for each model

Multiple interrelated models

- Need for configuration mechanisms throughout all lifecycle phases

Modelling and simulation today at Saab

- Used to various degrees in a all lifecycle stages
- Generic life cycle (ISO/IEC/IEEE 15288:2015)



INCOSE. Systems Engineering Handbook:
A Guide for System Life Cycle Processes and Activities.
Fifth Edition, Wiley

www.incose.org/symp2024 #INCOSEIS

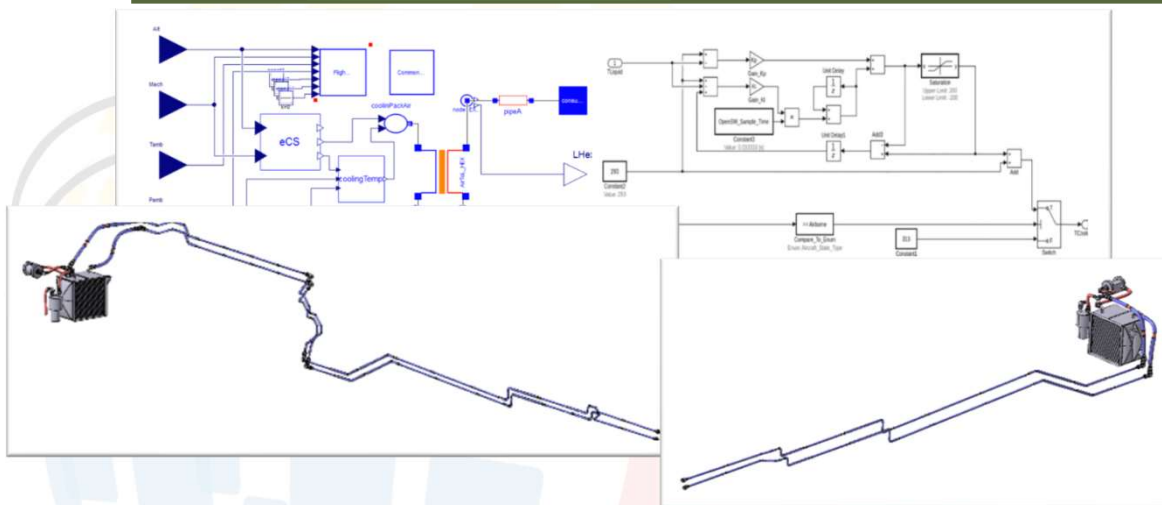
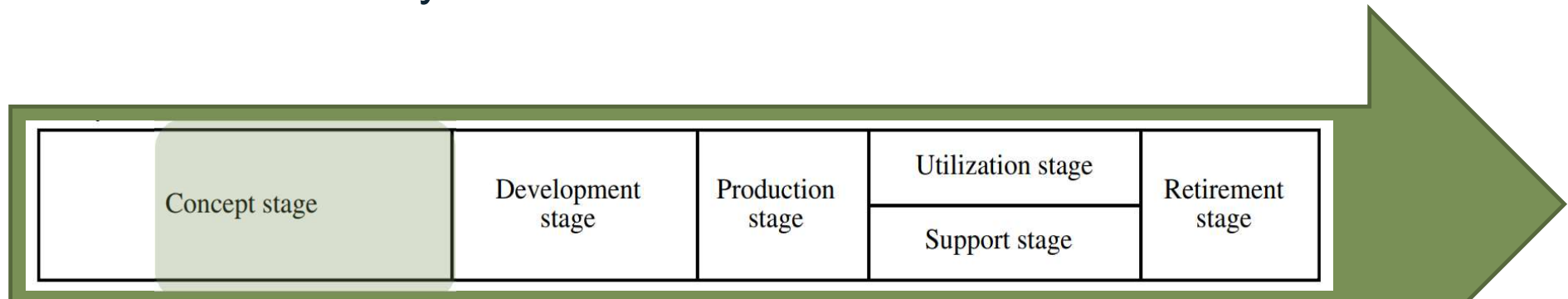


SAAB



Modelling and simulation today at Saab

- Evaluation of subsystem architectures



www.incose.org/symp2024 #INCLOSEIS

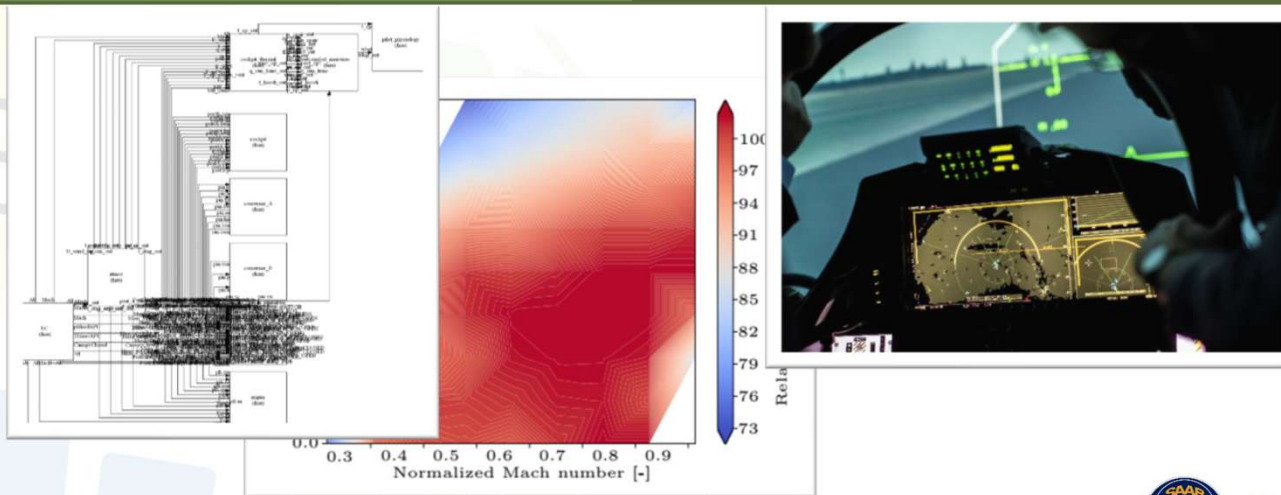
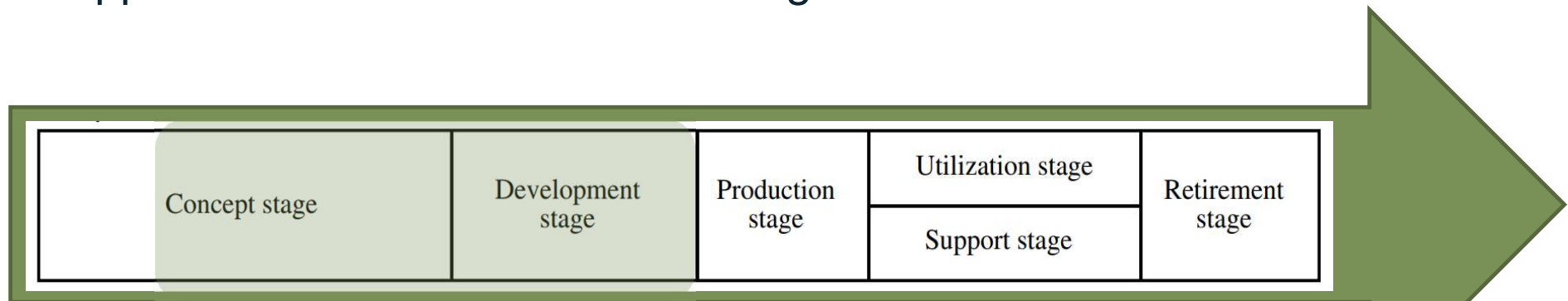


SAAB



Modelling and simulation today at Saab

- Support hardware and software design



www.incose.org/symp2024 #INCOSEIS

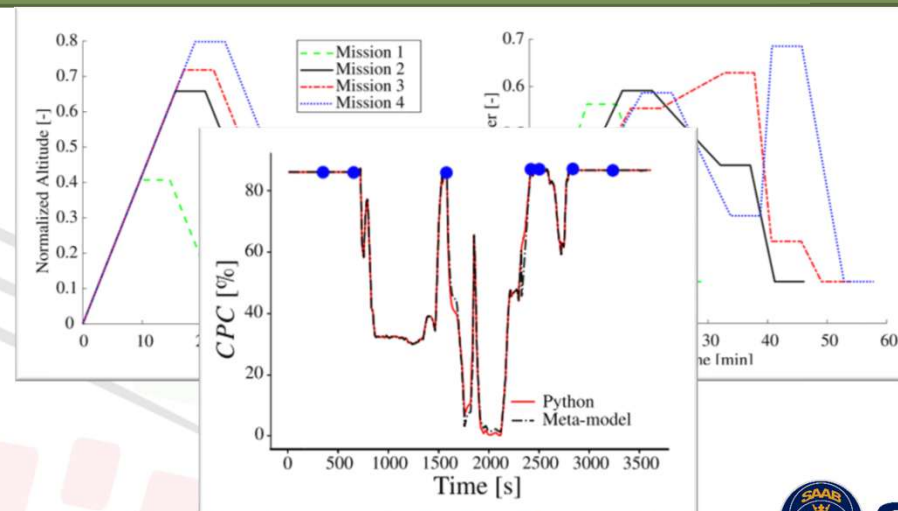
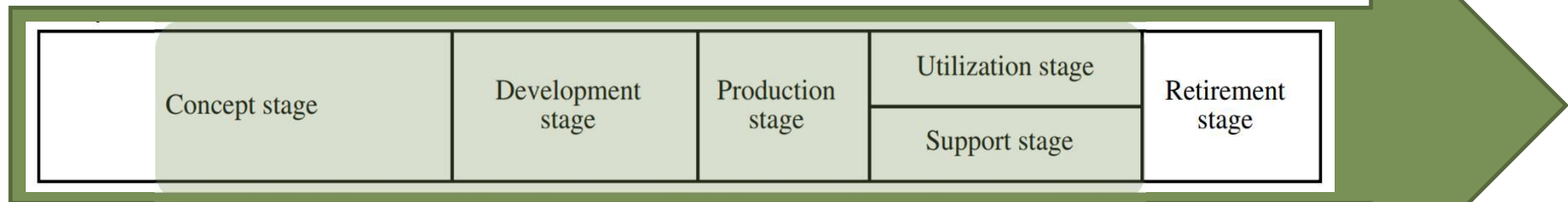


SAAB



Modelling and simulation today at Saab

- Software & hardware verification
- Decision and maintenance support



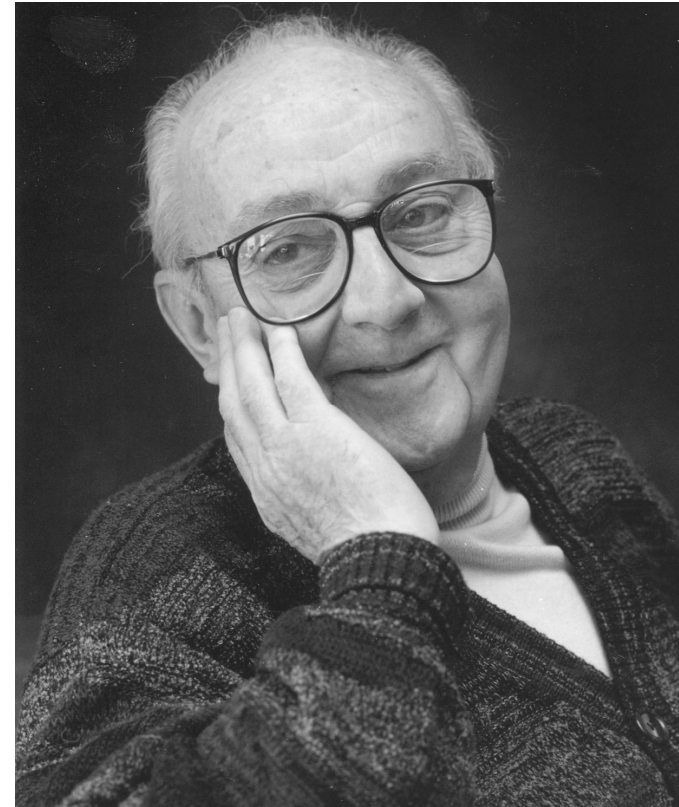
www.incose.org/symp2024 #INCOSSEIS



SAAB



**All models are approximations.
Essentially, all models are wrong but some are useful.**



George Box



SAAB

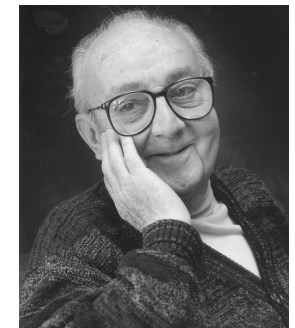


Some words on Credibility

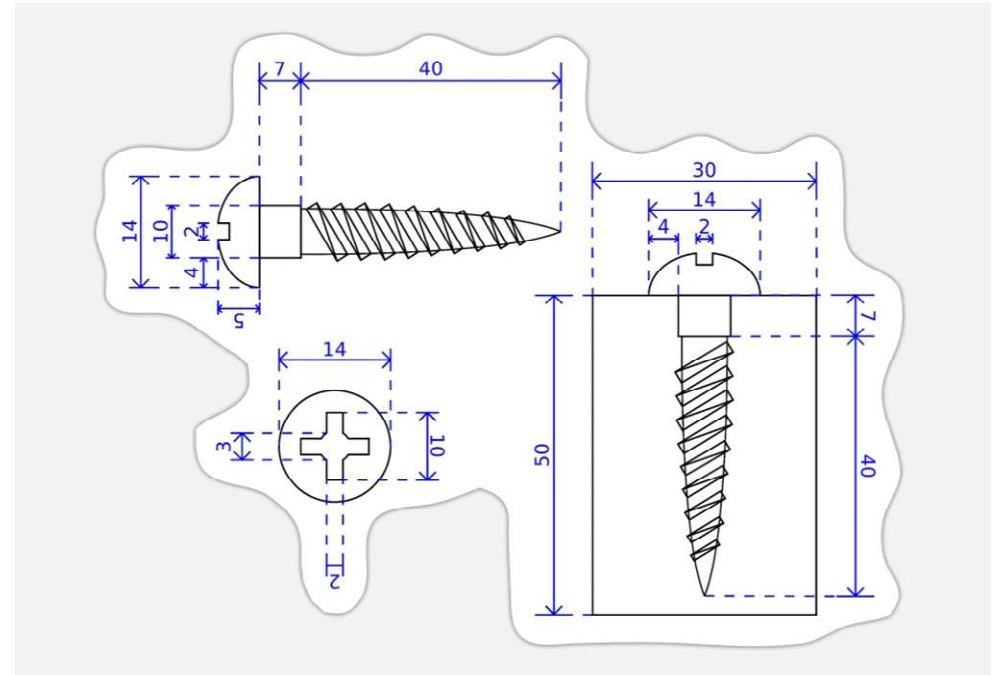


- All models are developed to fulfil some purpose – often not captured explicitly
- **Credibility:** The extent to which the model meets its purpose
- Models, whose credibility is unknown is of questionable value

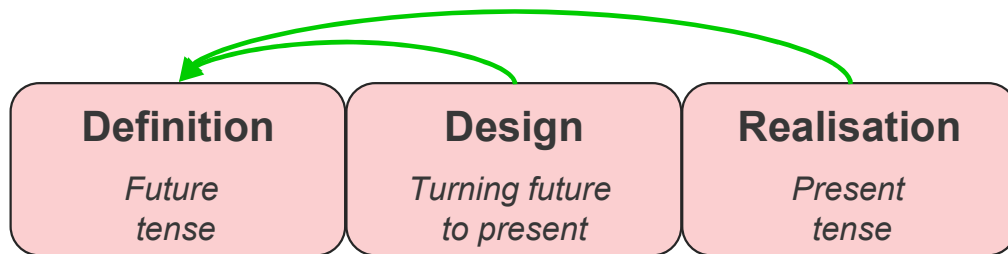
**Credible
models are
useful!**



Models capturing different views on a system



Proposed model framework



Definition model

- Captures the **intended architecture**
- Relatively undetailed
- Used for communication and long-term memory, e.g. change management and development planning
- For example, SysML as a common language

Design model

- Captures a system element from a **particular perspective**
 - Design or analysis focus
 - Interfaces and key properties
- **Multiple** Design models may be required to adequately represent the intent in a Definition model
- Multiple languages, e.g. Simulink, Modelica, CFD

Realisation model (physical/virtual)

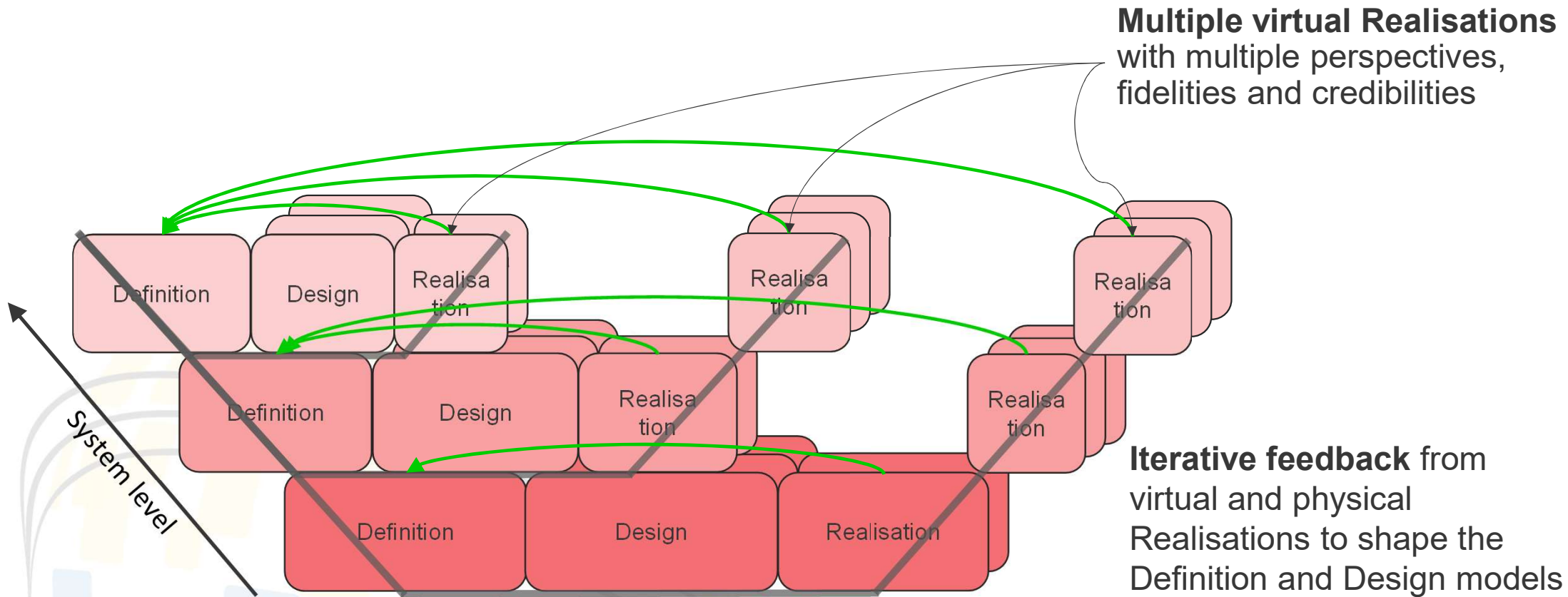
- **Multiple virtual** Realisations with different **fidelities** and **perspectives** may be created
- **Interface models** are required for both an executable realisation and a realisation of the physical system



SAAB

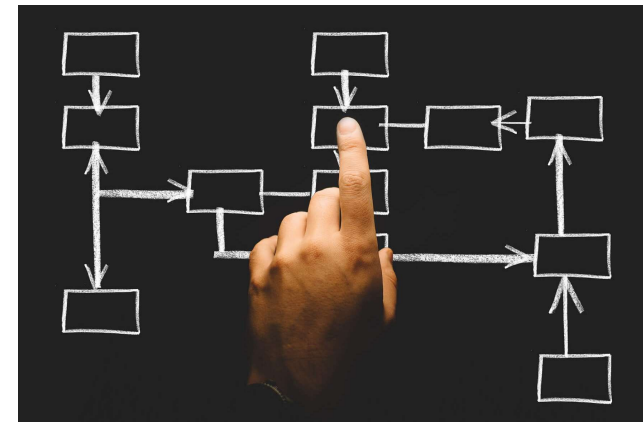
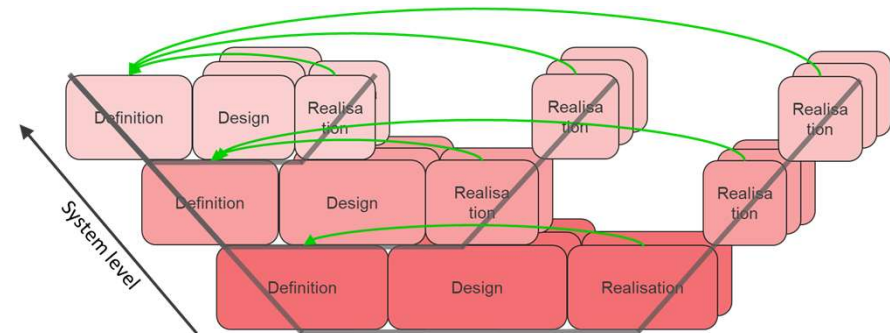


Feedback using virtual Realisations



Problem statement

- We have a conceptual model
- Need to define
 - Clear definitions for different model types used in development
 - Explicit expectations on each model type
- There is a need for a unified terminology!
 - Not just within Saab



SAAB





A Saab view on model types terminology

www.incose.org/symp2024 #INCOSEIS

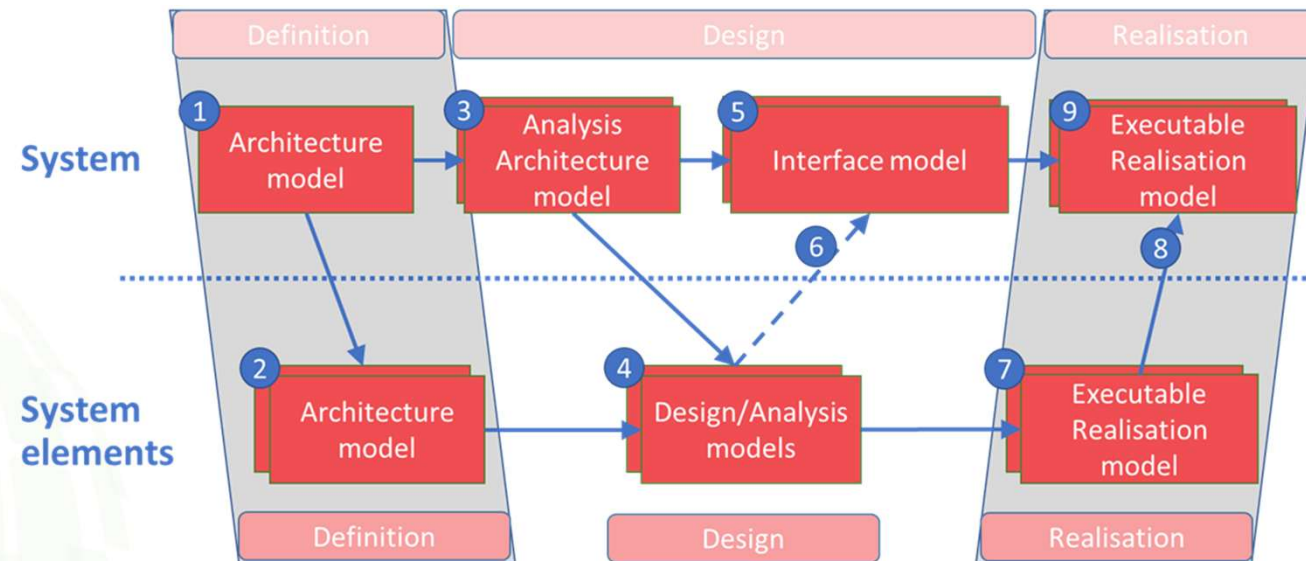
Tenses and model types

Architecture model

- Identifying system behaviour, system elements and interfaces

Analysis Architecture model

- Adapting the architecture for a particular analysis purpose
- May result in the addition or deletion of items compared to the architecture model



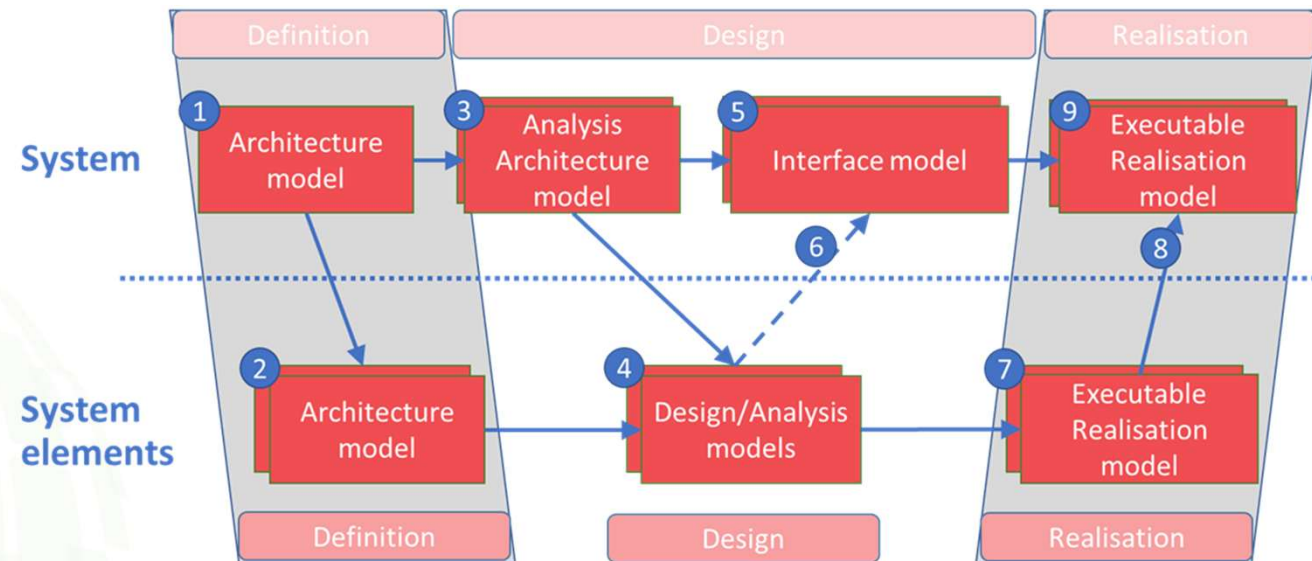
Tenses and model types

Design/Analysis model

- Captures the emergent system design or system analyses

Interface model

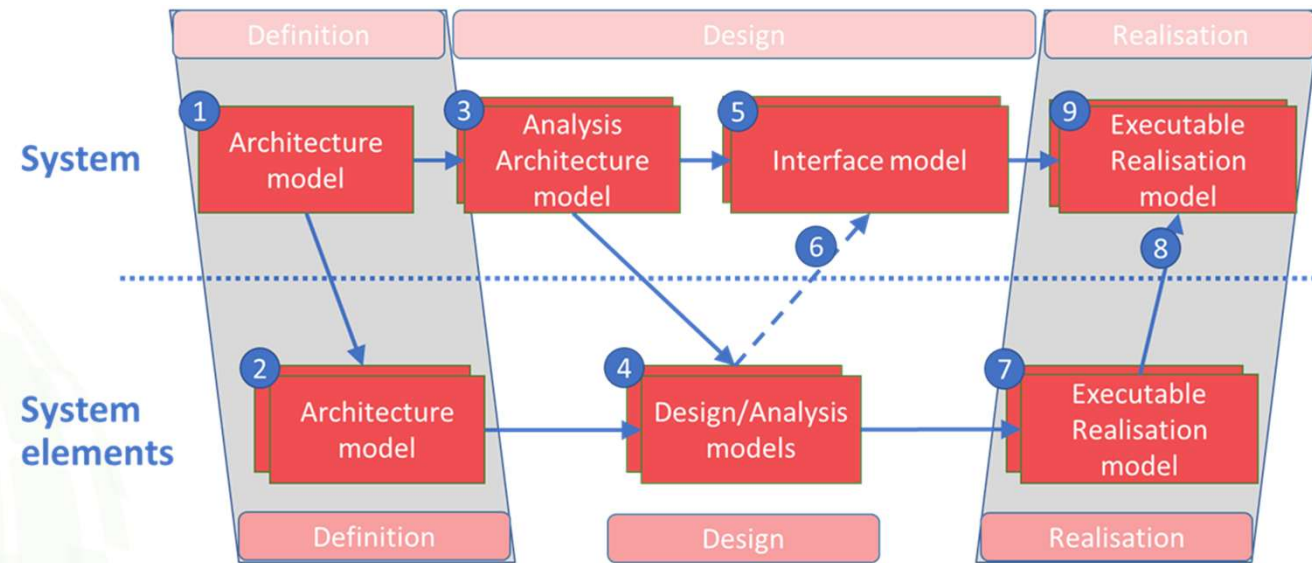
- Derived from the Architecture model and refined with design content
- Purpose to provide the template for virtual and/or physical integration



Tenses and model types

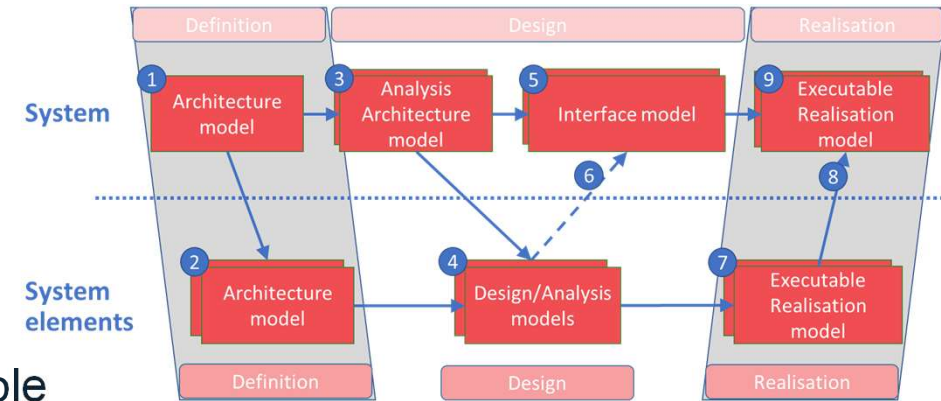
Executable Realisation model

- The virtual realisation of a system used for gaining insights and knowledge



Why separate models?

- **Architecture model:** Overall definition of the system – suitable for communication, not executable
- **Analysis Architecture model:** Meeting the needs for a particular analysis – based on the architecture but should not be included in it
- **Design/Analysis model:** Allowing the most appropriate modelling language for detailed design of a heterogeneous system
- **Interface model:** Detailed interface definition – in a language agnostic format for integration and creation of Executable Realisation models.

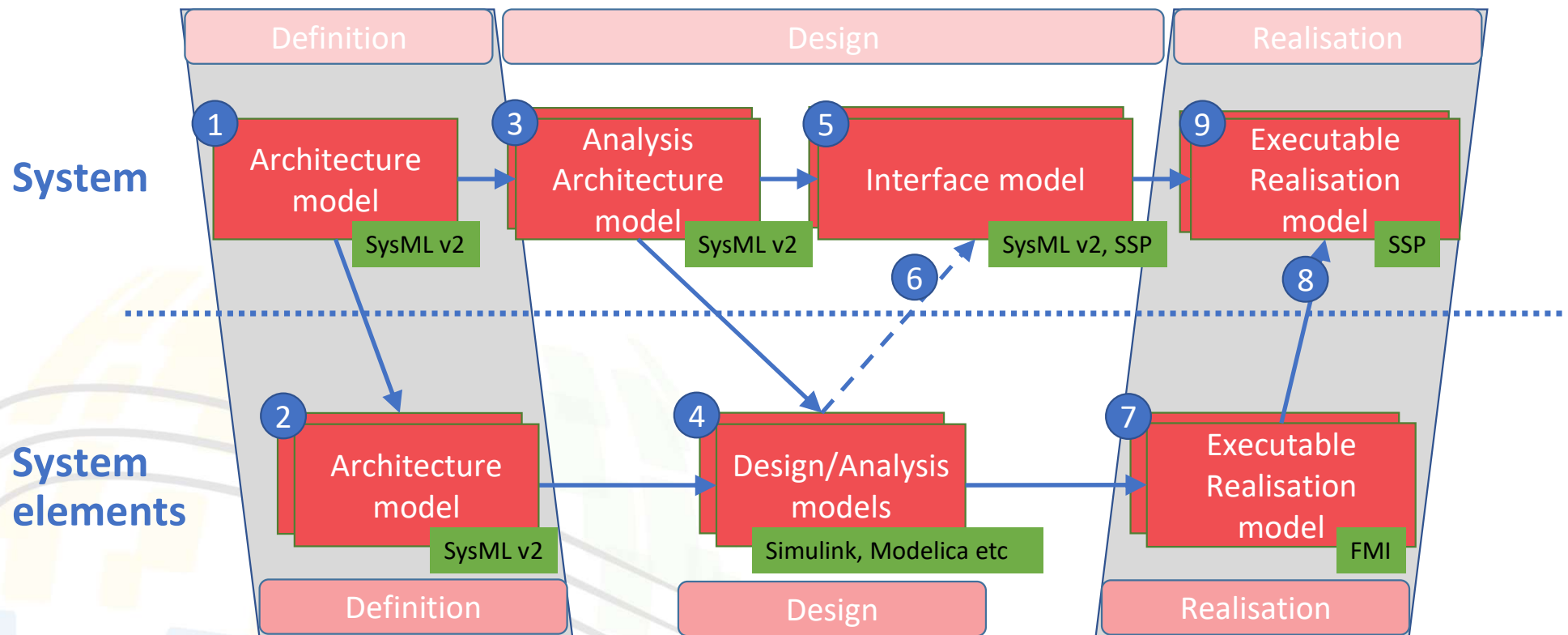


Analysis Architecture models can not be merged with the **Architecture model** as it would skew the Architecture model

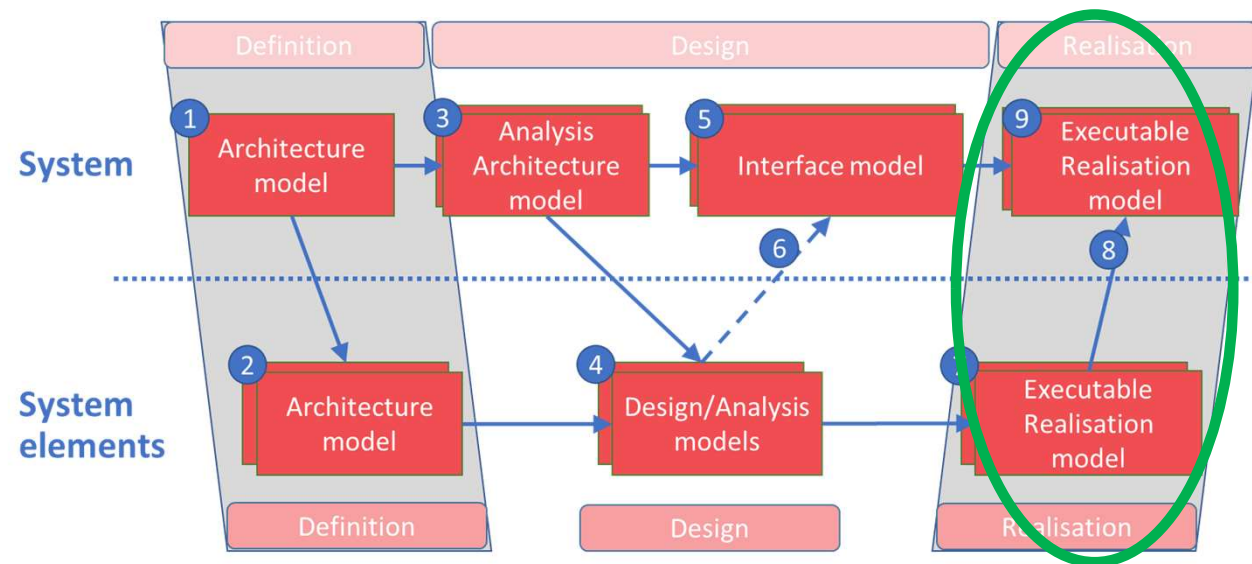
Design/Analysis models are there to take advantage of the power of domain specific languages

Interface models are distinct to allow interface refinement without having to change the **Architecture or Analysis Architecture models**

Tenses and model types



Nomenclature for executable models



Executable Realisation Model

A **digital artifact** that an experiment can be applied to in order to answer **questions about a corresponding physical logical artifact** (realised or not).

- *Example: A control system simulation model, used in a system simulator during development of a specific product configuration.*



Analysis Model

Executable Realisation Model

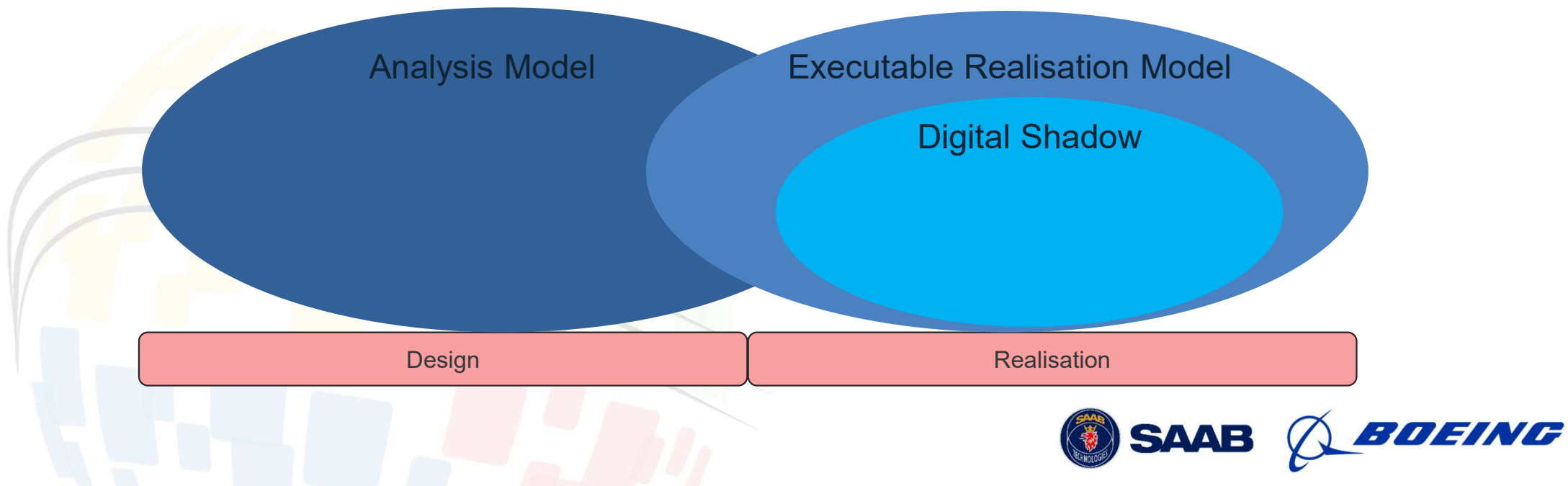
Design

Realisation

Digital Shadow

An **Executable Realisation Model** where relevant **information from the physical space is incorporated automatically**. *Passive* in the sense that it does not automatically influence the actual system it represents.

- *Example:* A model processing aircraft maintenance information, received automatically from the aircraft during or after flight.



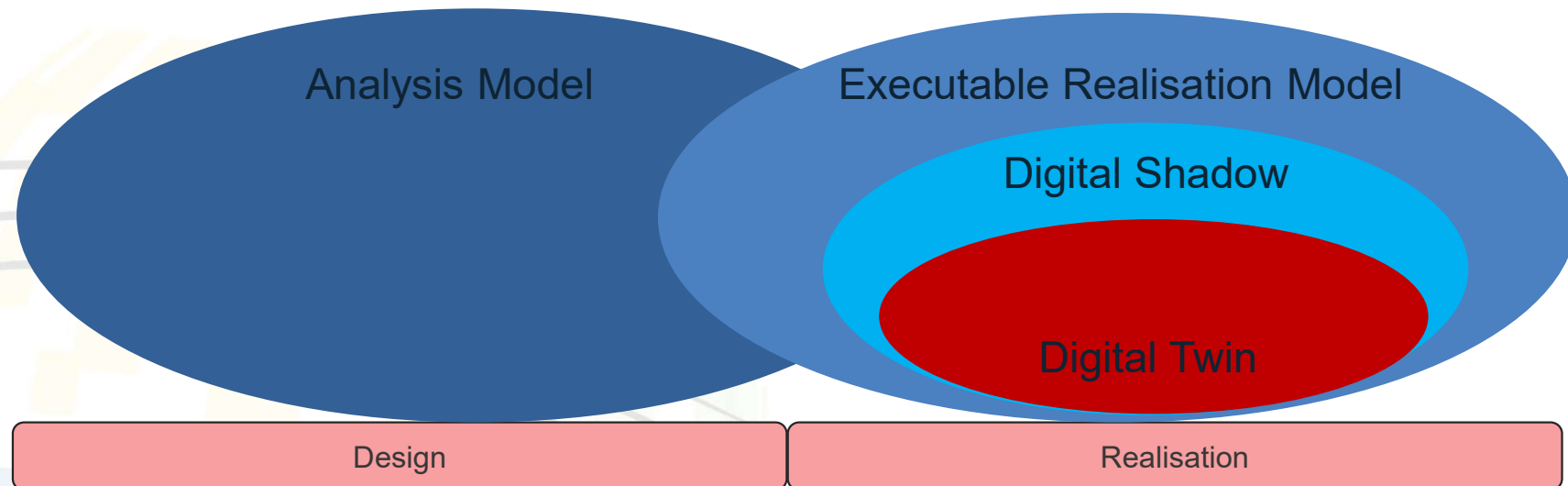
SAAB



Digital Twin

Executable Realisation Model where **any relevant bi-directional flow of information** between the virtual and physical spaces is **exchanged automatically**. The Digital Twin is **Active**.

- *Example:* A model used during equipment test, where software parameters in the equipment are changed automatically based on the information analysed.

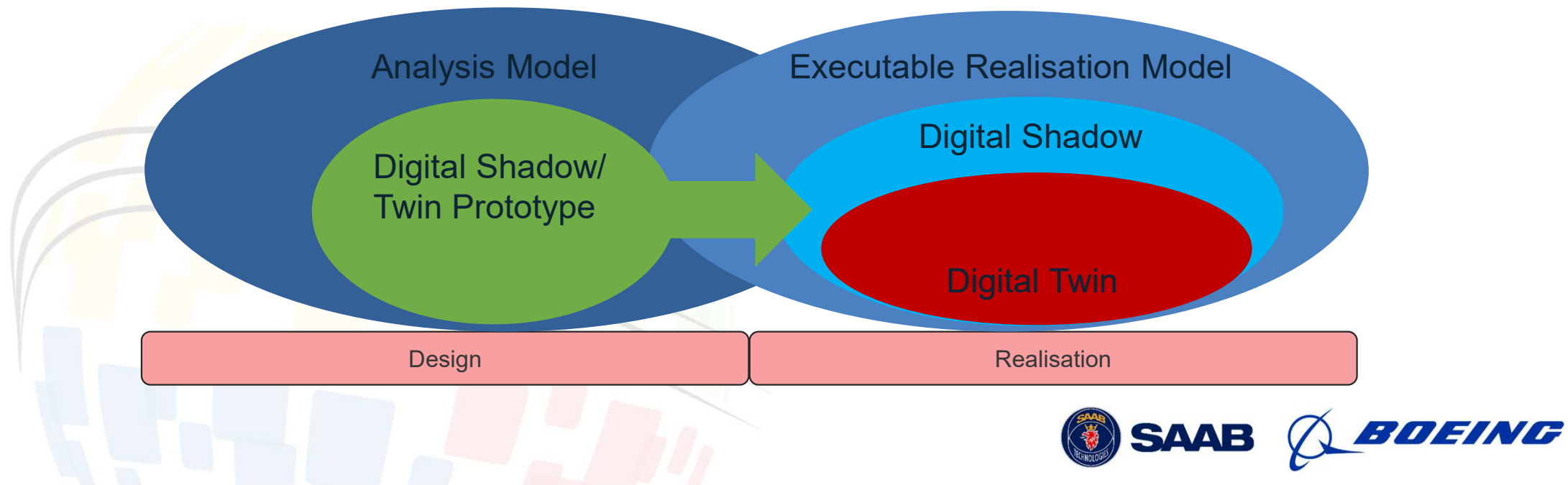


SAAB

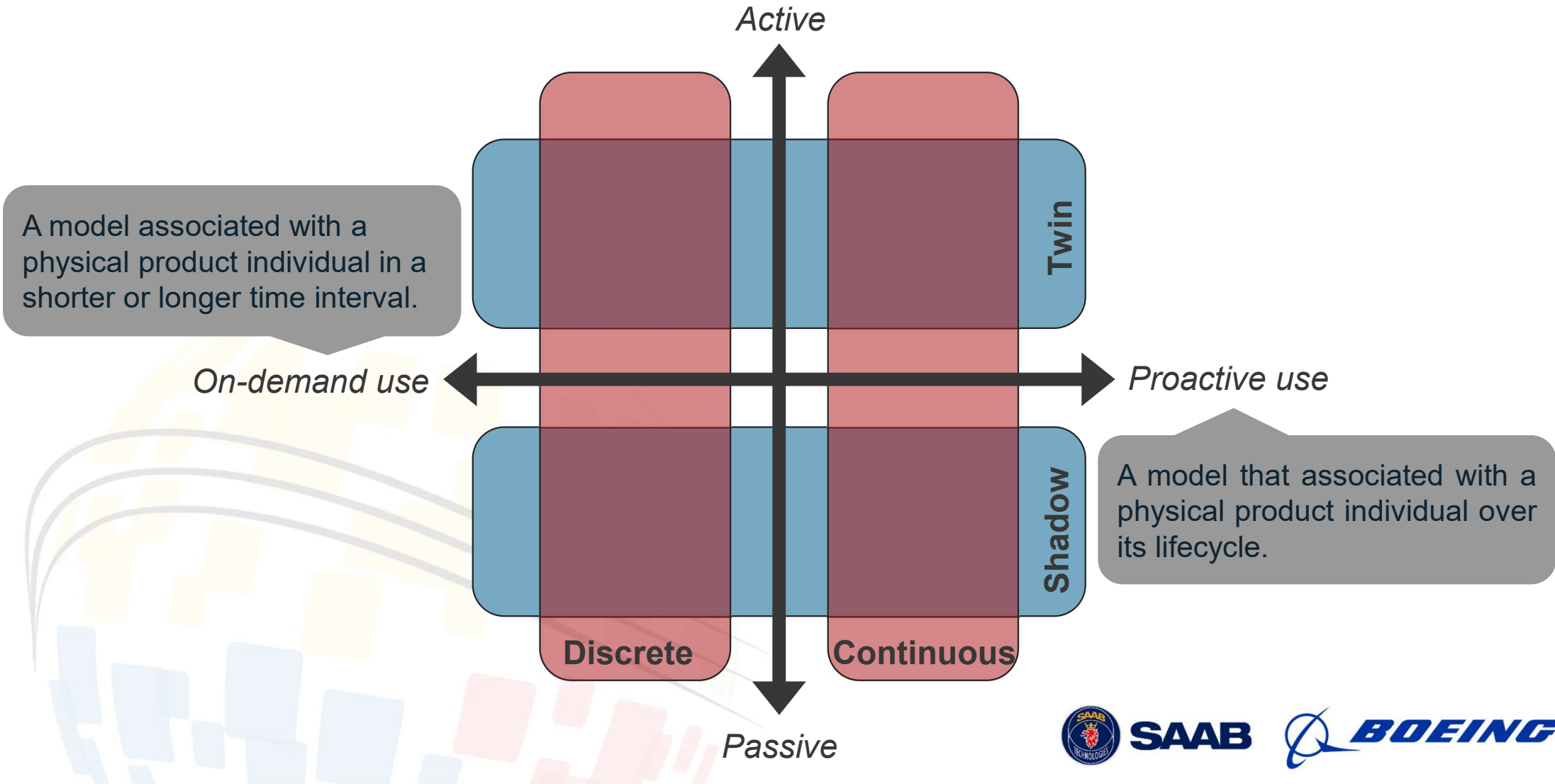


Digital Shadow/Twin Prototype

Analysis Model or Executable Realisation Model that is **identified as a candidate** to evolve to a Digital Shadow or Twin. Contains the structure and information necessary to create Digital Shadows or Twins if decided to do so.



Two more dimensions



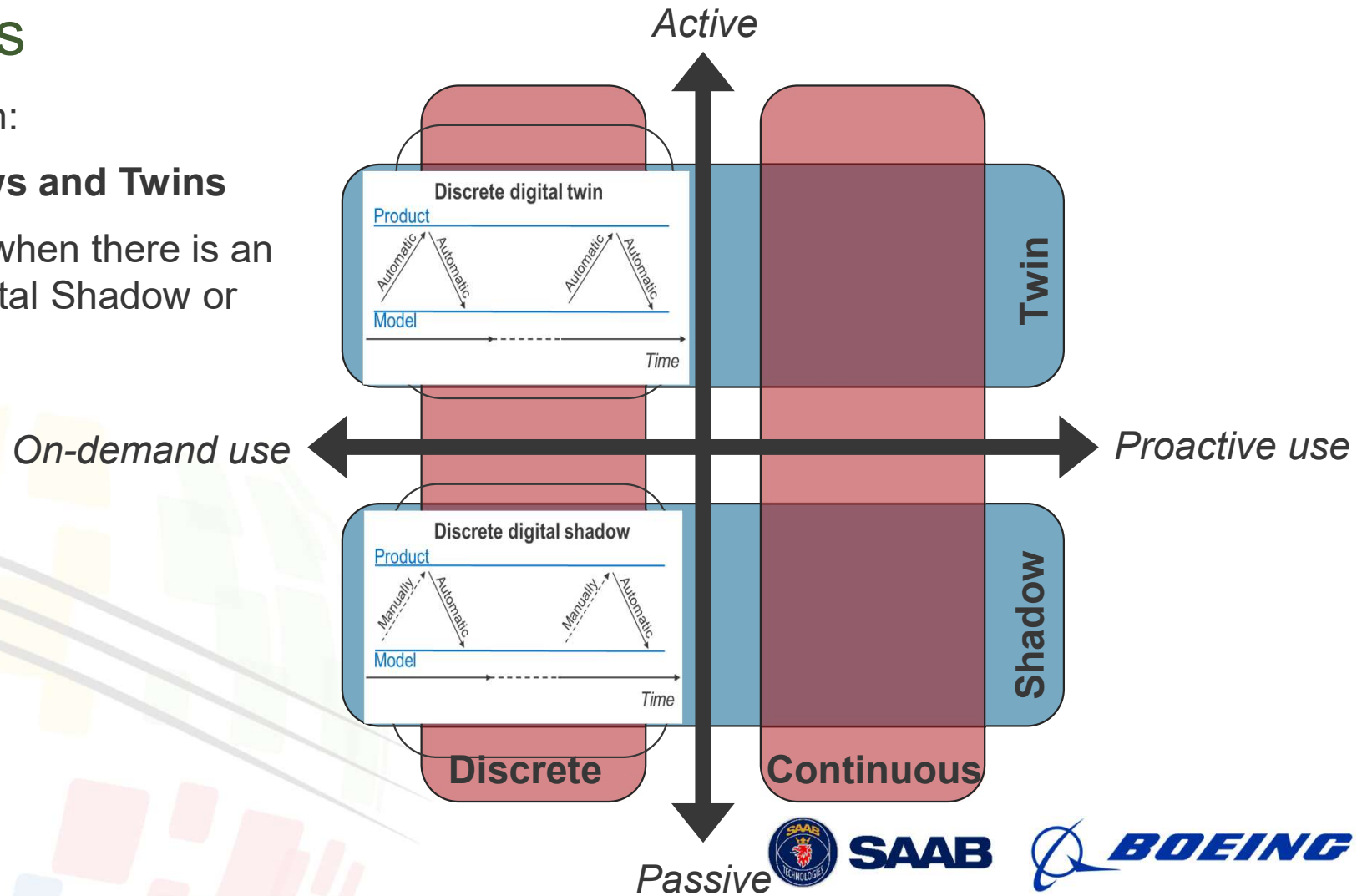
Digital Shadows and Twins

Use categories

On-demand use results in:

Discrete Digital Shadows and Twins

- Maintained and used when there is an explicit need for a Digital Shadow or Twin



Digital Shadows and Twins

Use categories

On-demand use results in:

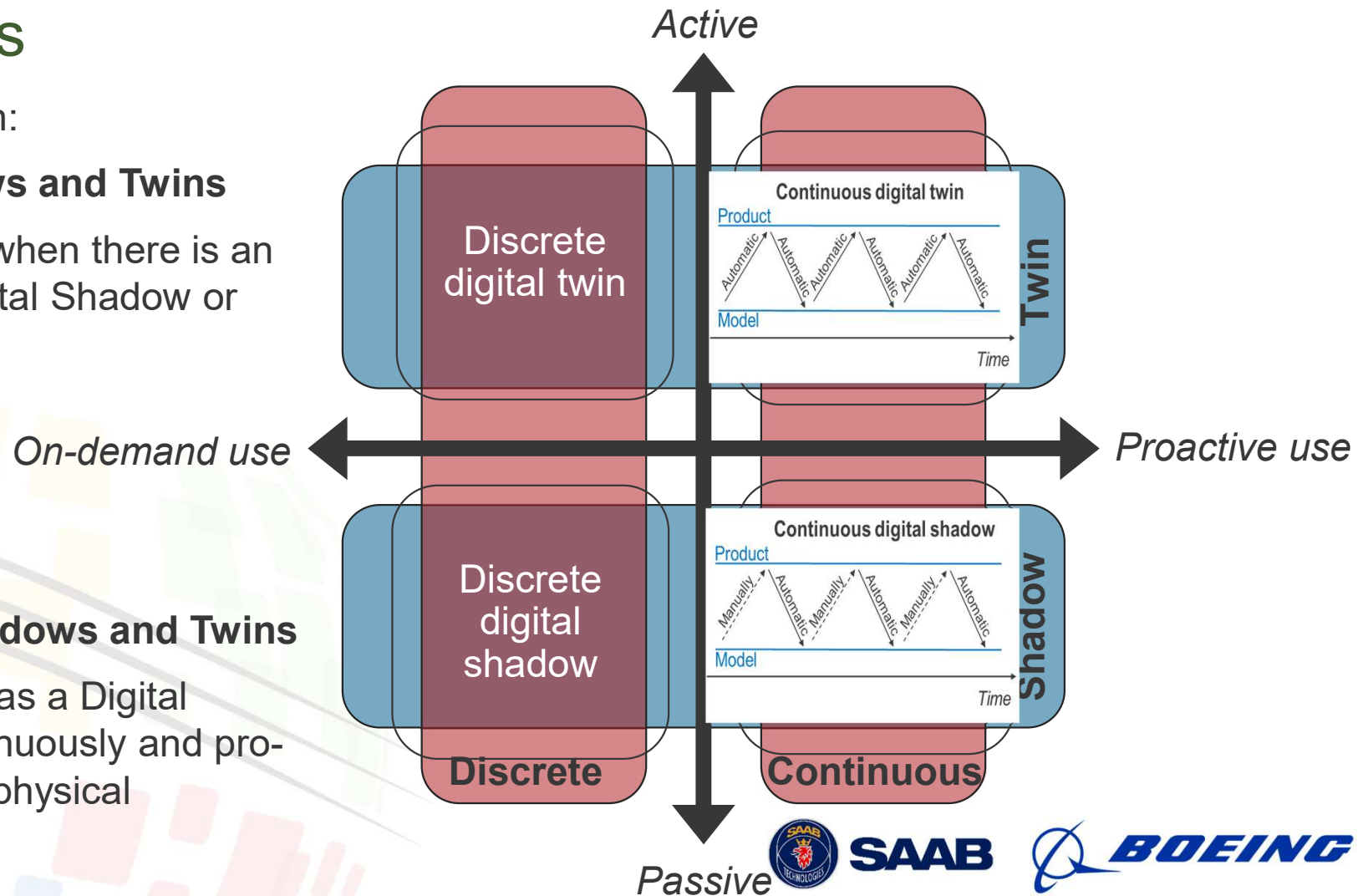
Discrete Digital Shadows and Twins

- Maintained and used when there is an explicit need for a Digital Shadow or Twin

Proactive use results in:

Continuous Digital Shadows and Twins

- Maintained and used as a Digital Shadow or Twin continuously and proactively alongside its physical counterpart.



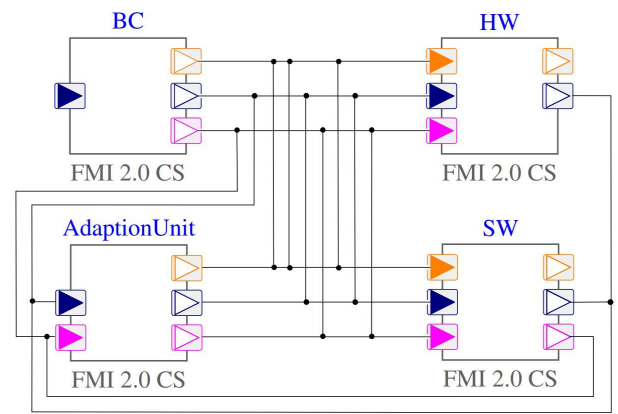
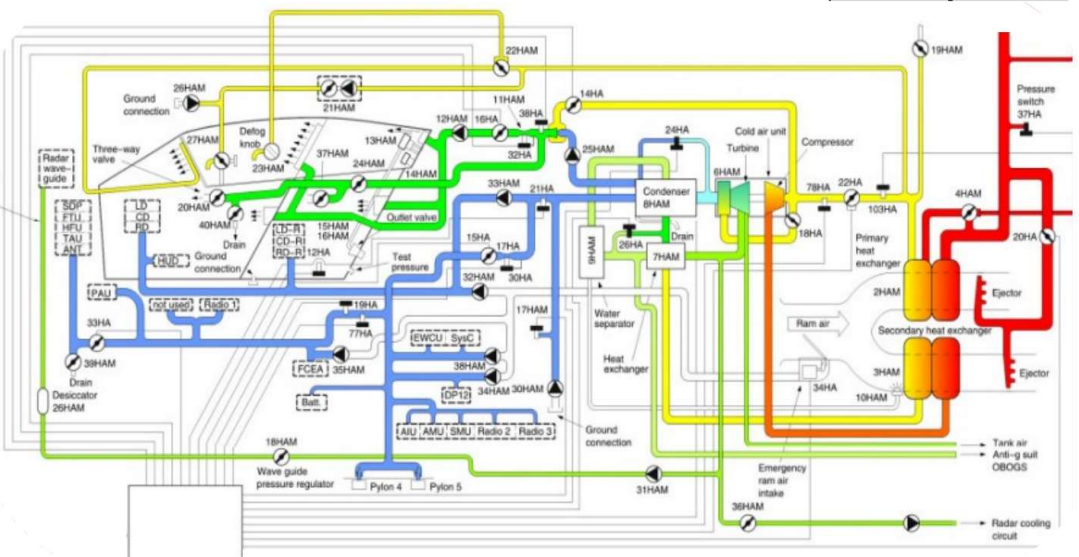


Example

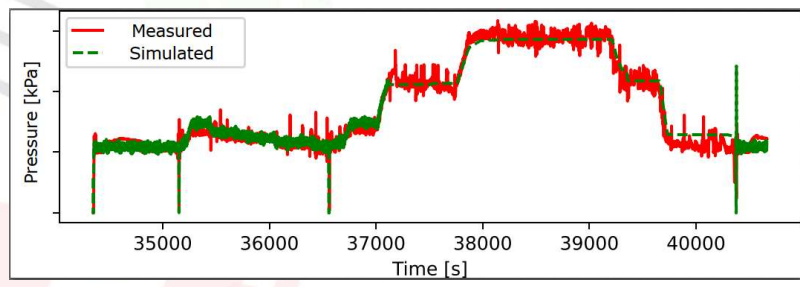
www.incose.org/symp2024 #INCOSEIS

Environmental Control System (example)

1 Architecture model



- 3 Analysis Architecture model
- 4 Analysis models
- 9 Executable Realisation model



Digital Shadow

Digital Twin



Related projects

Cyber physical systems simulation

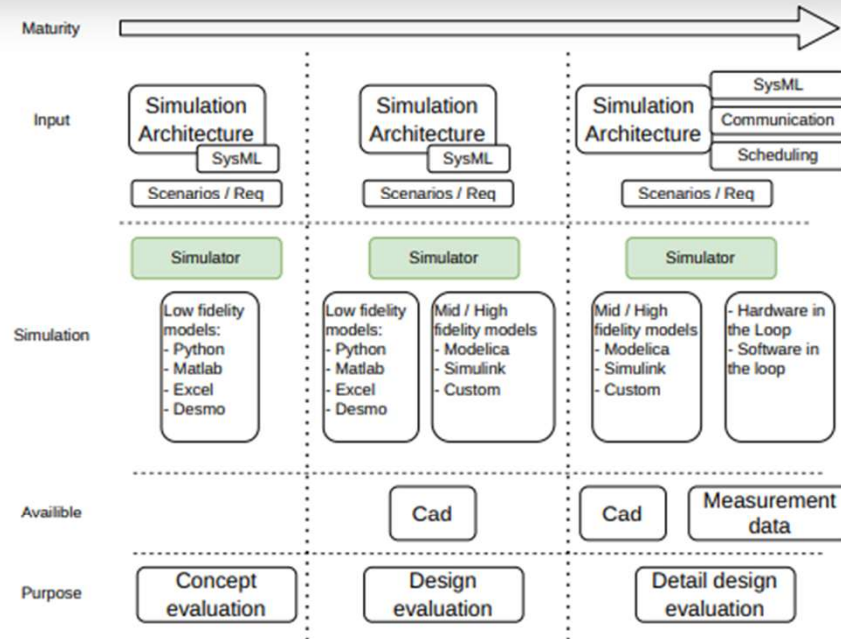
Applied MBSE

Erik Rosenlund

Software architect, PhD Student - Aircraft system simulation

Enable incremental development

As models and systems evolve

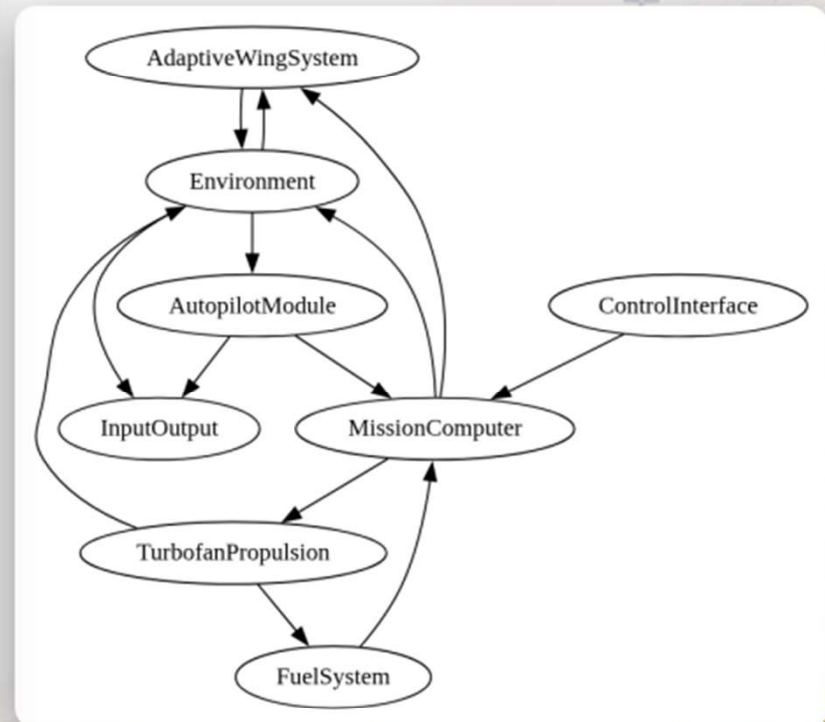


Evaluating CI/CD

Public application example

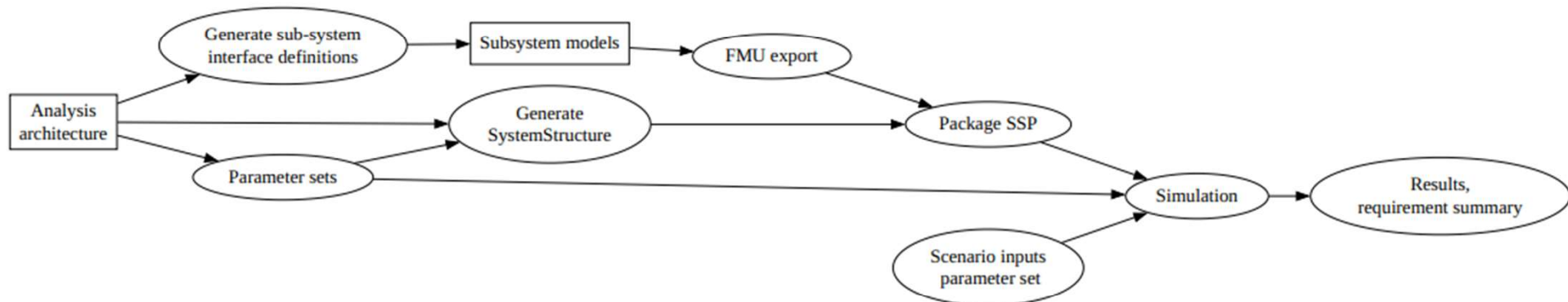
- Sysml v2 simulation architecture
 - Parts
 - Interfaces
 - Requirements
- Models

https://github.com/jkCXf9X4/ssp_airplane



Continuous Integration/Development

Design - Model - Simulate

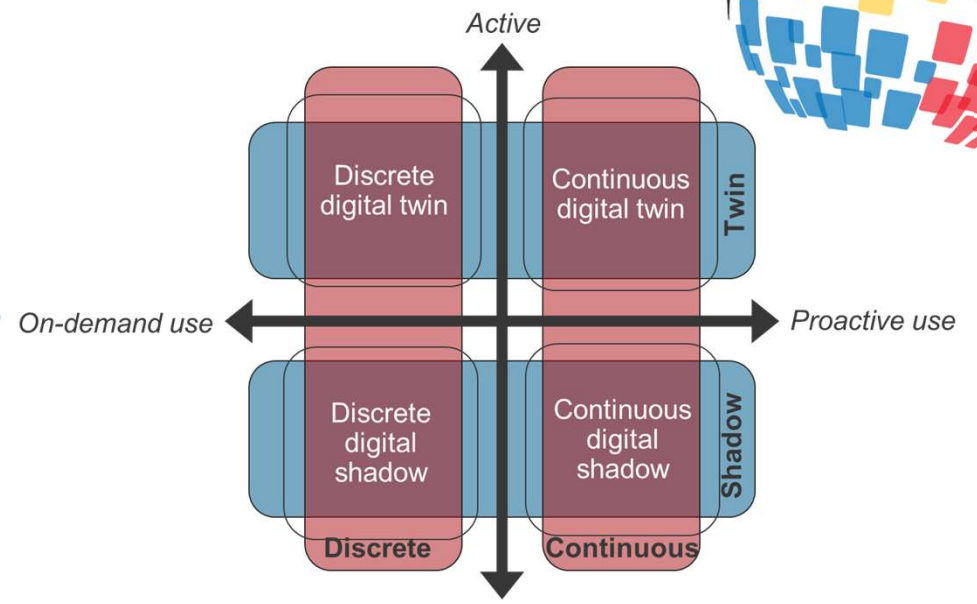
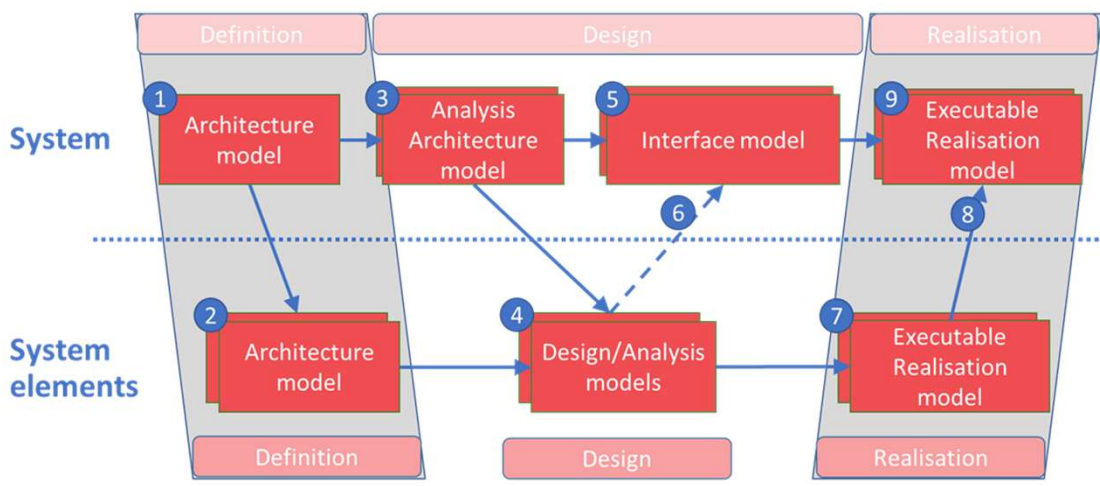


Supporting sysml frameworks

- pycps_sysmlv2
 - Python SysML v2 parser for CPS development
 - https://github.com/pyssporg/pysysml2_cps
- pyssp_sysml2
 - Python module for parsing sysml v2 and generating SSP artifacts
 - https://github.com/pyssporg/pyssp_sysml2

Supporting simulation frameworks

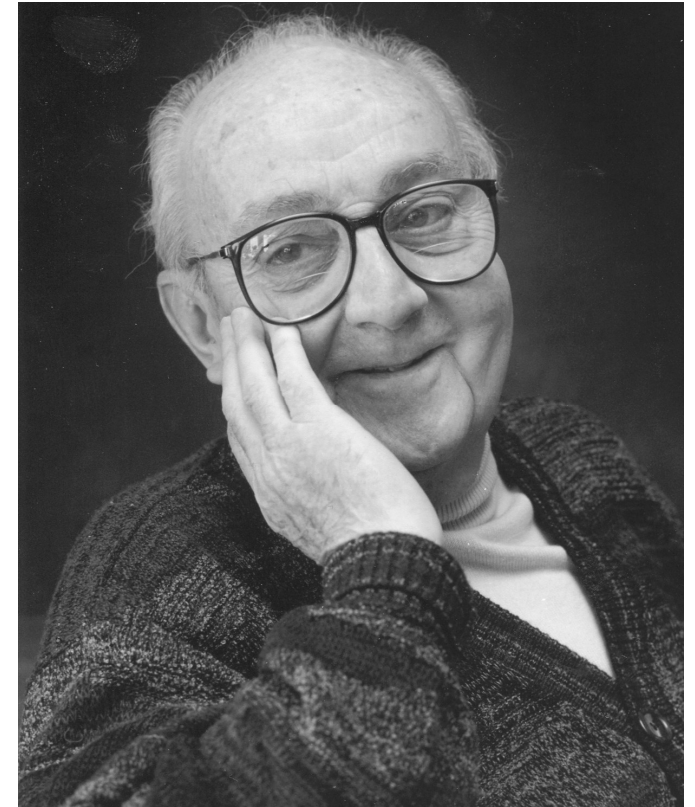
- SSP4SIM
 - Co-simulation engine
 - <https://github.com/jkCXf9X4/ssp4sim>
- SSP4CPP
 - C++ bindings for SSP and FMI
 - <https://github.com/jkCXf9X4/ssp4cpp>



Discussion & Conclusions



**Excuse me Sir,
WHICH model?**



George Box



SAAB



Summary

- Model architecture principles
- Proposal for unified Terminology
 - Covering Definition, Design and Realisation
 - Extension of Digital Twin and Shadow terminology
 - An extension of the previously published model framework
- Agreement on terminology simplifies discussions related to model properties, such as
 - Credibility
 - Model fidelity



NASA 2009. Standard for Models and Simulations: Credibility Assessment Scale, NASA-STD-7009, NASA Technical Standard System, published online: <https://standards.nasa.gov/standard/NASA/NASA-STD-7009>, accessed 2023-11-27



SAAB



Discussion topics

- What does the overall model landscape look like?
 - Does the proposed framework align with your needs and experiences?
- What is your preferred approach for capturing model purpose and credibility?
- Are there adequate mechanisms for maintaining configuration control for the complete model set?
 - Traceability between model types, and between model and system? (e.g. analysis architecture model and system architecture)



SAAB





34th Annual **INCOSE**
international symposium

hybrid event

Dublin, Ireland
July 2 - 6, 2024

www.incose.org/symp2024
#INCOSEIS