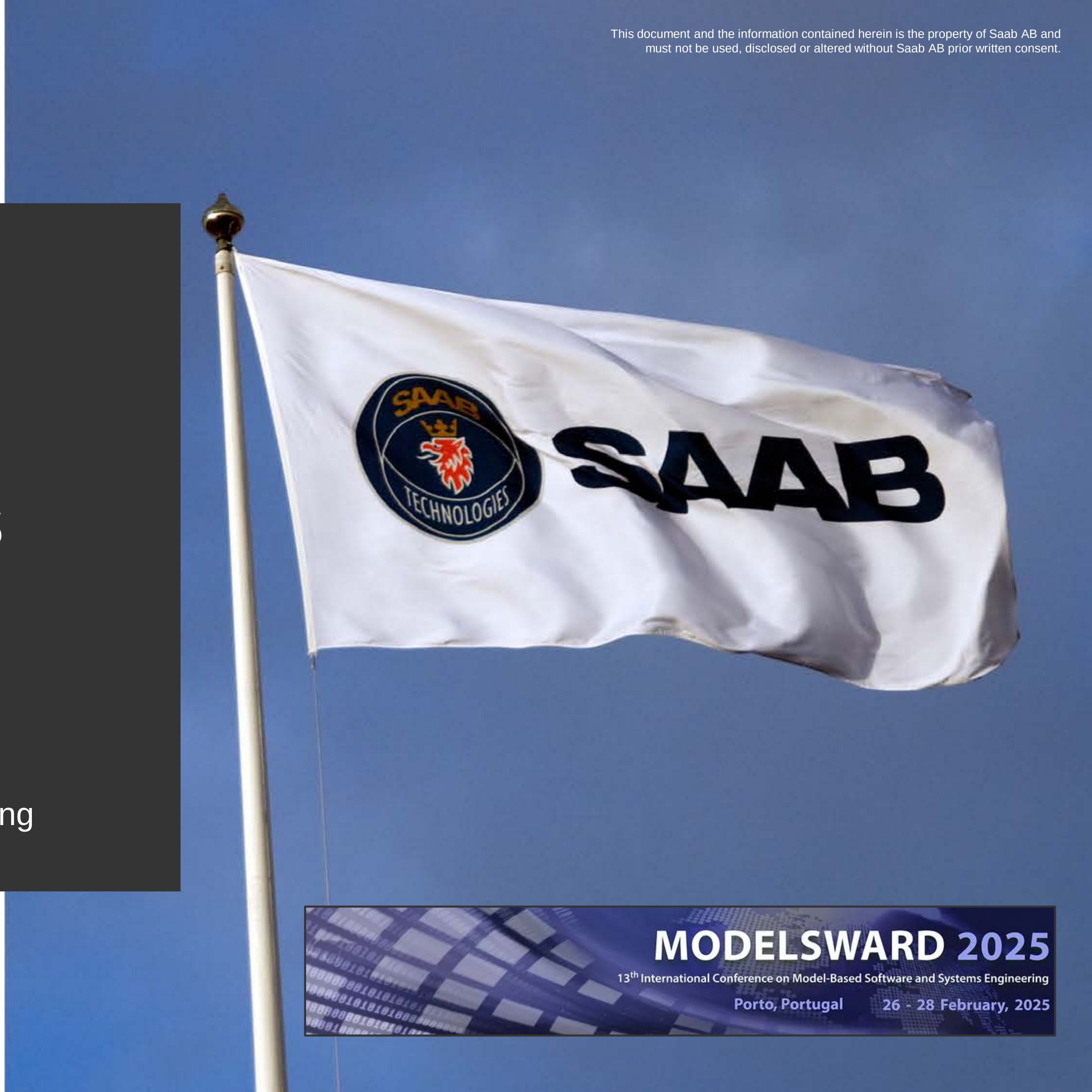
The Lessons the Models Taught us

Erik Herzog, Ph.D., CSEP

Saab Technical Fellow – Systems Engineering



SAAB Aeronautics – current development projects

Gripen E/F

GlobalEye

T-7

This is a story about Gripen E development

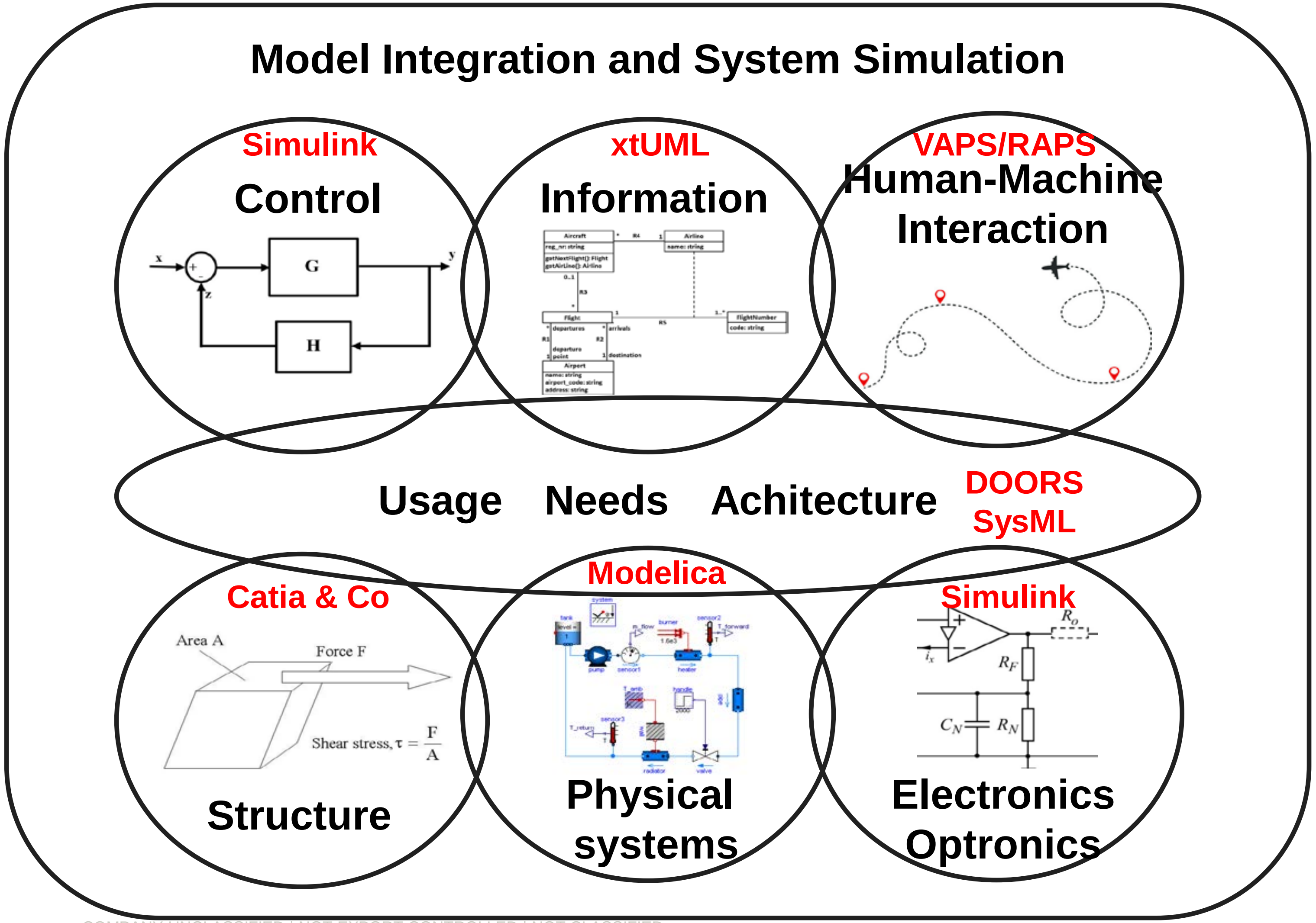
And what we have learnt in the process

Preliminaries – circa 2007

Future directions identified

- **MBSE is the future!**
  - All engineering disciplines should go model based
- **New process framework** – emphasis on architecture and design capabilities
- **New PLM system** for efficient configuration and information management

MBSE Domains – Gripen example

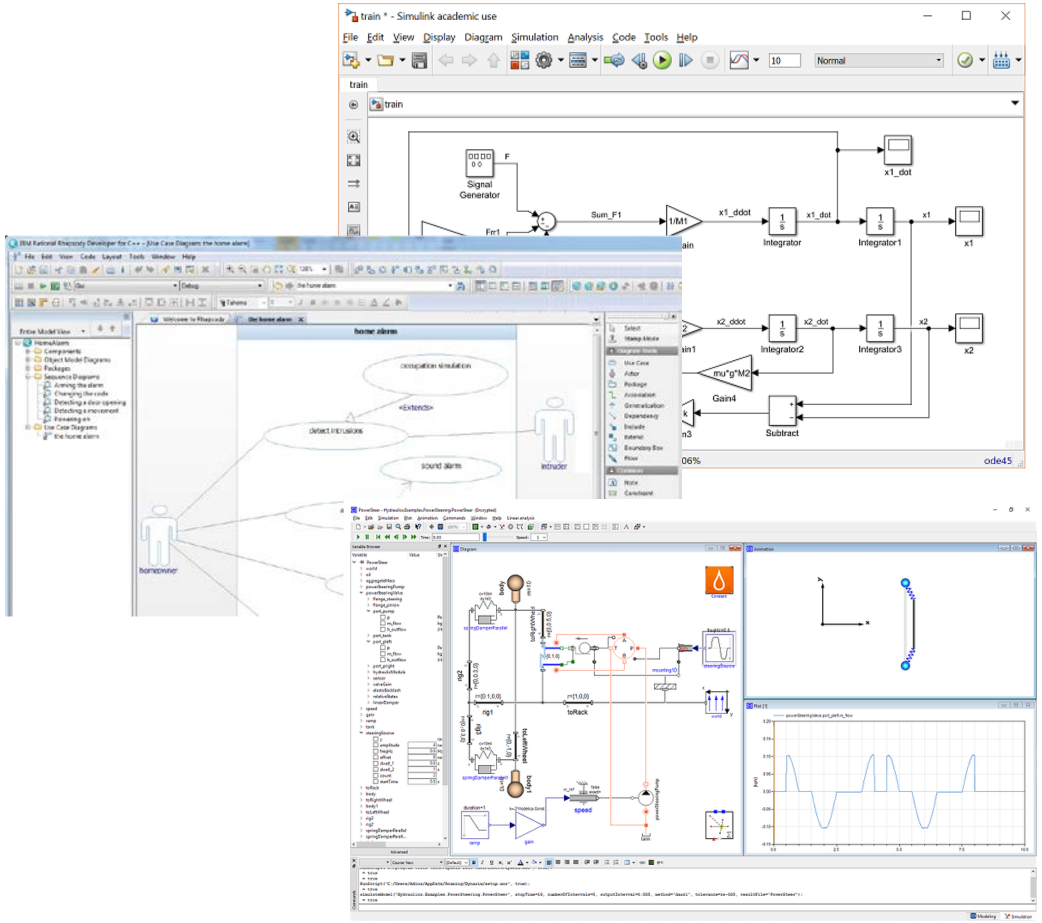


Models of information, objects, needs, services

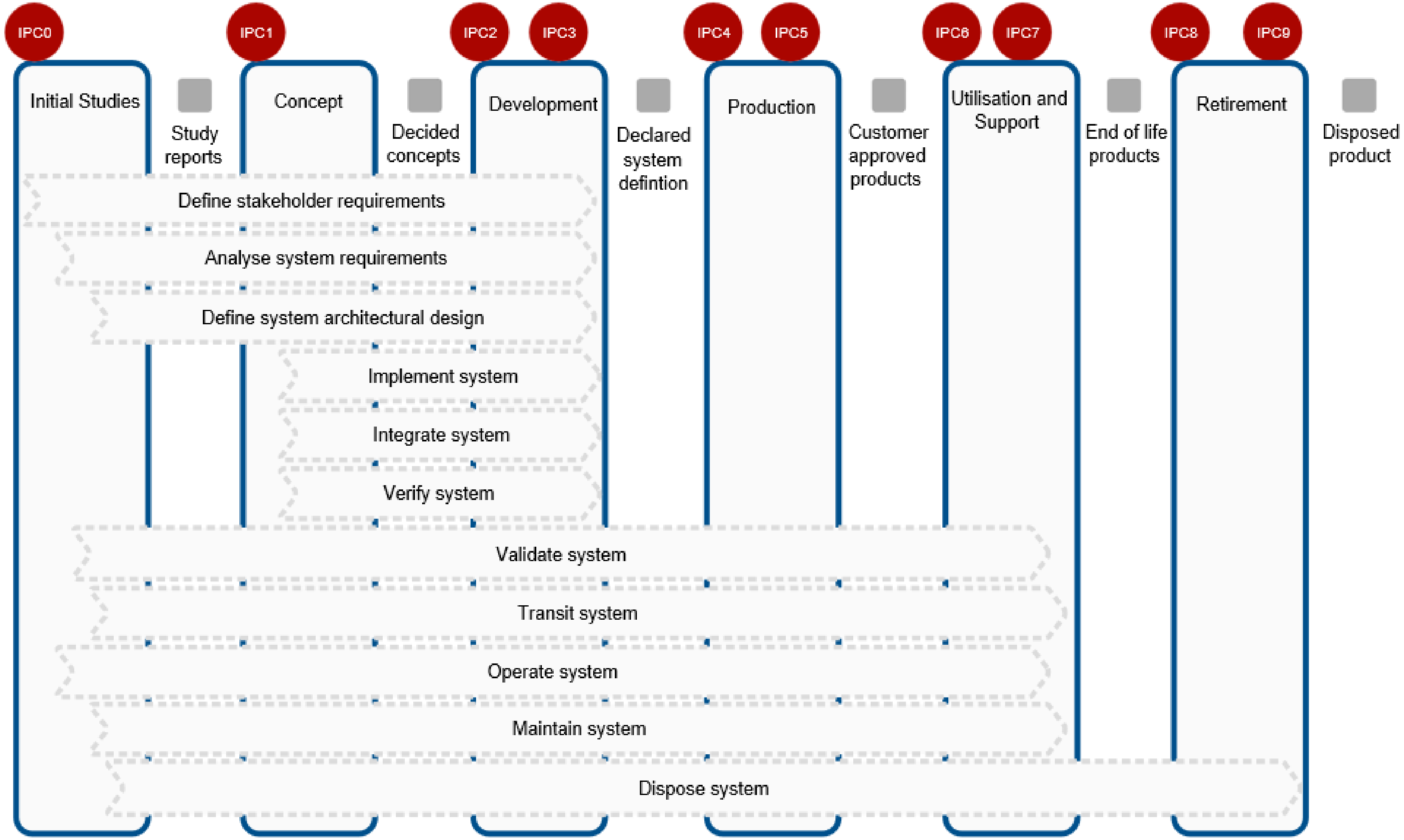
Models of physical objects

Expected benefits

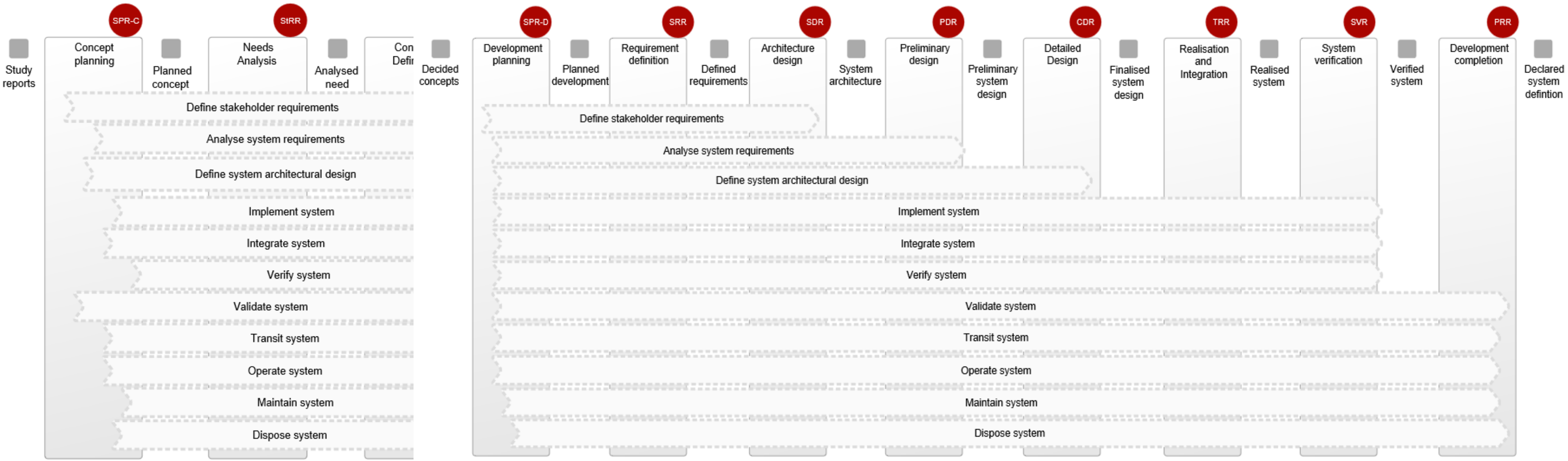
- **Improved communication** – ability to discuss design alternatives in an objective way
- Faster **knowledge capture**
- **Early validation** – ability to simulate design concepts to increase
  - Feasibility
  - Acceptance of solution
- **Improved accuracy** – ability to determine and tune performance early in development
  - Fewer flight tests
- **Improved quality** – right the (almost) first time
- **Improved efficiency** – quicker turn-around
- **Decreased risks** and higher confidence



New process framework – ISO 15288

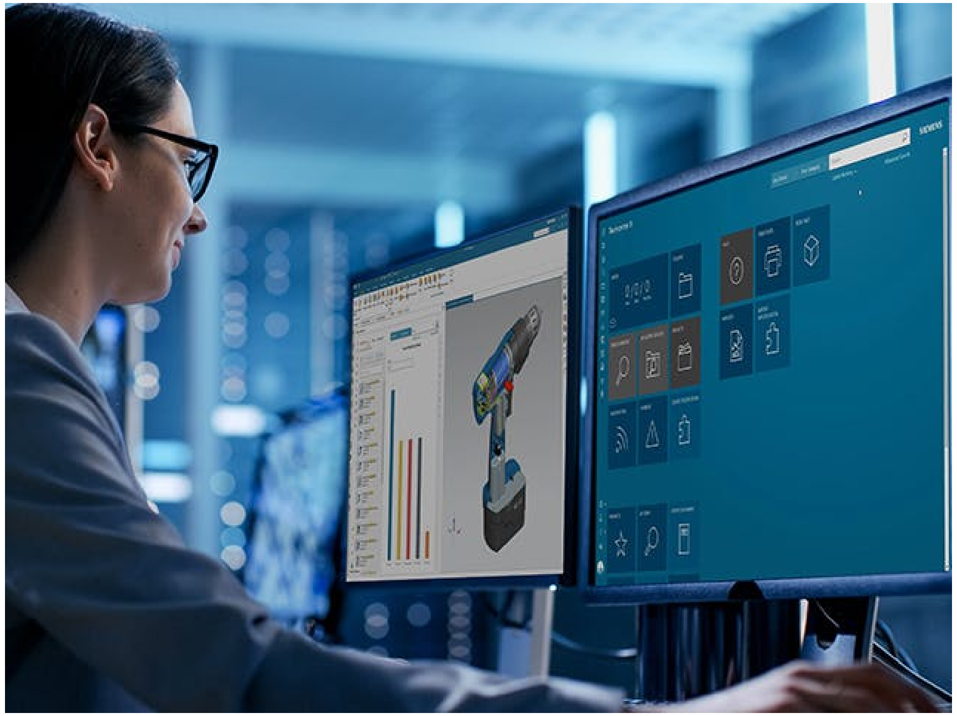


Process over lifecycle – including reviews



New PLM system - Teamcenter

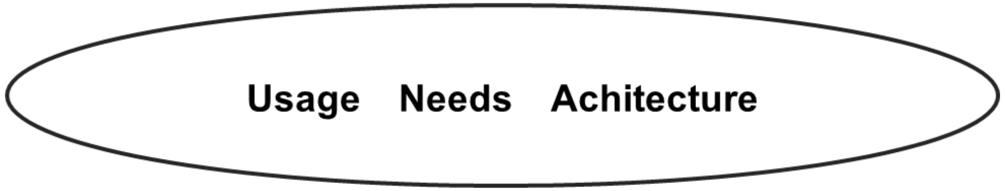
- Management of
  - Product structures
  - Variants
  - Change
- Approvals
- Declaration of conformance

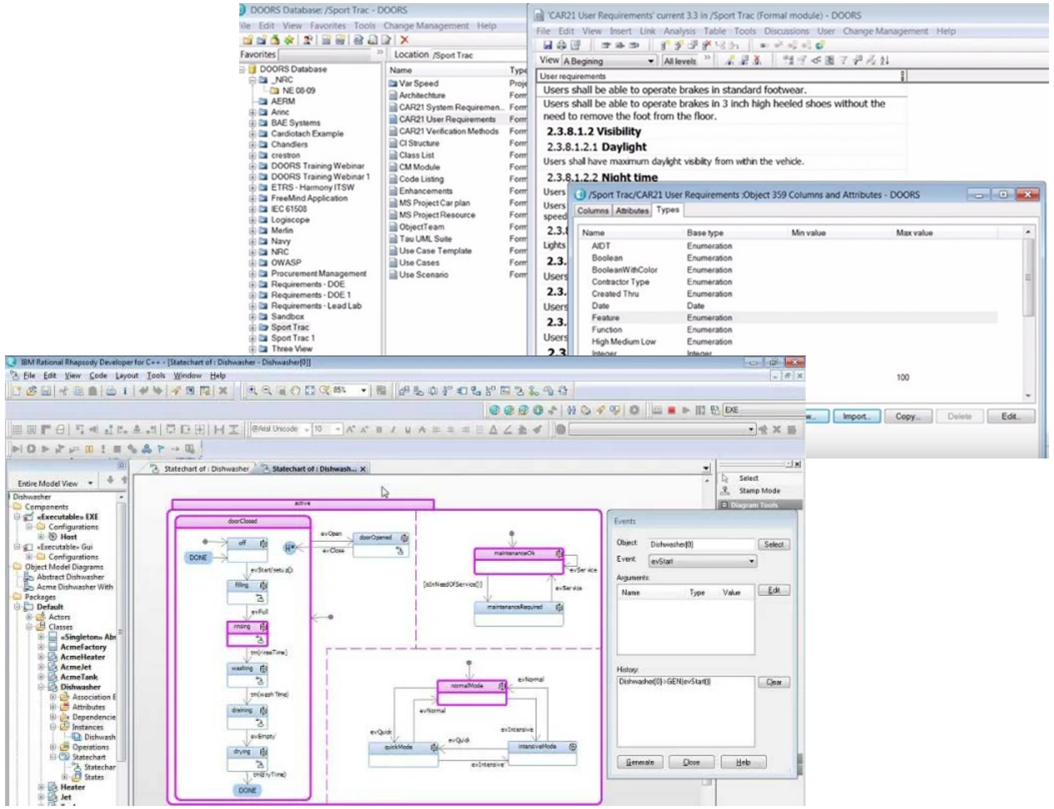


An example of thorough preparations

Needs and architecture

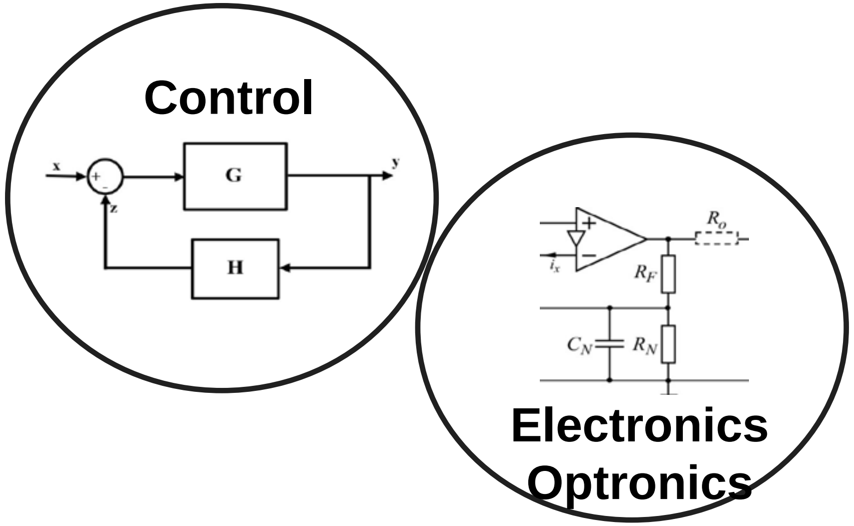
- IBM Doors for requirements management
  - The standard requirements management tool within the organisation
  - Expert support organisation
- Rhapsody with SysML
- Used in multiple projects prior to Gripen E
  - For modelling parts of legacy systems and new subsystems

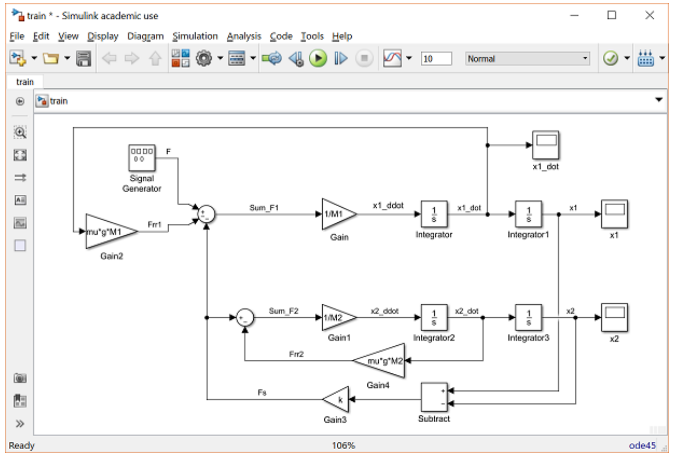




Control and Electronics & Optronics

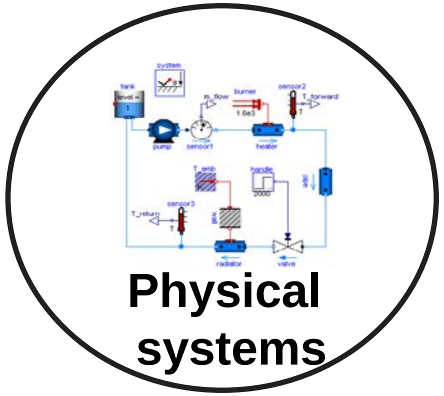
- Mathworks Simulink introduced as a new tool, previous experience with legacy tool
- Extensive concept studies and support from the supplier
- Code generation support – validated for RTCA-178C-level A
- Dedicated support organisation setup

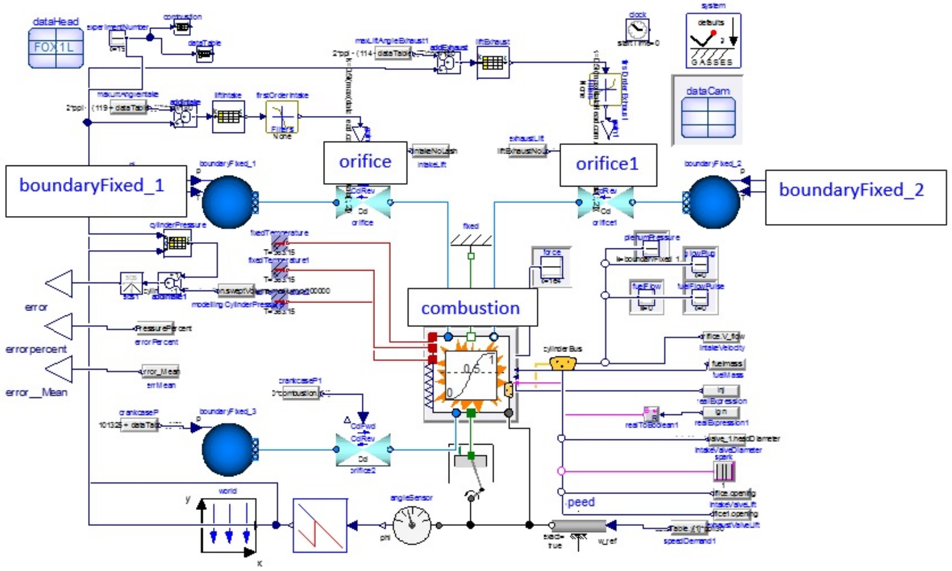




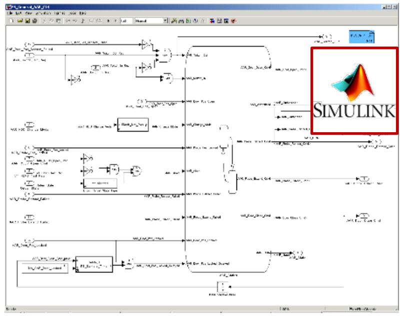
Physical systems – Modelica

- Dassault Dymola introduced as new tool
  - Previous experience with legacy tool
- Saab specific block libraries developed and validated by third party suppliers
- Modelica – Swedish origin – lots of competence available



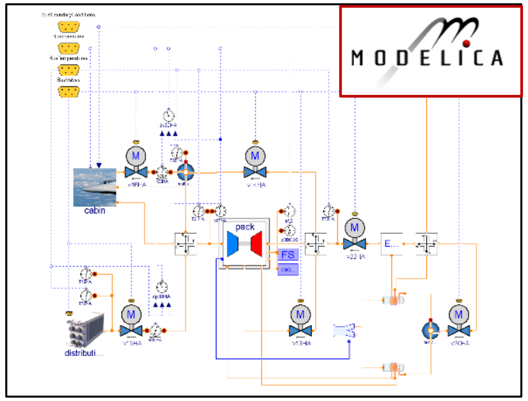


Iterative, model-based systems development (Gripen E)



Model, design and implementation of software

1



Model and simulation of physical system

Reuse

Test rigs & simulators

Flight tests

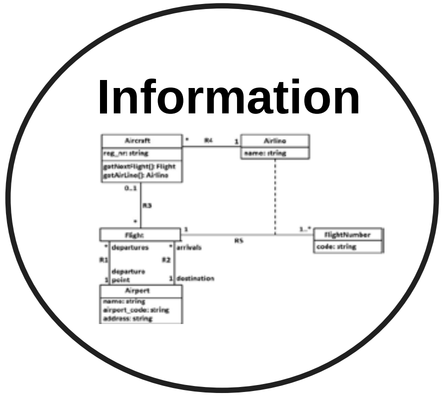
2

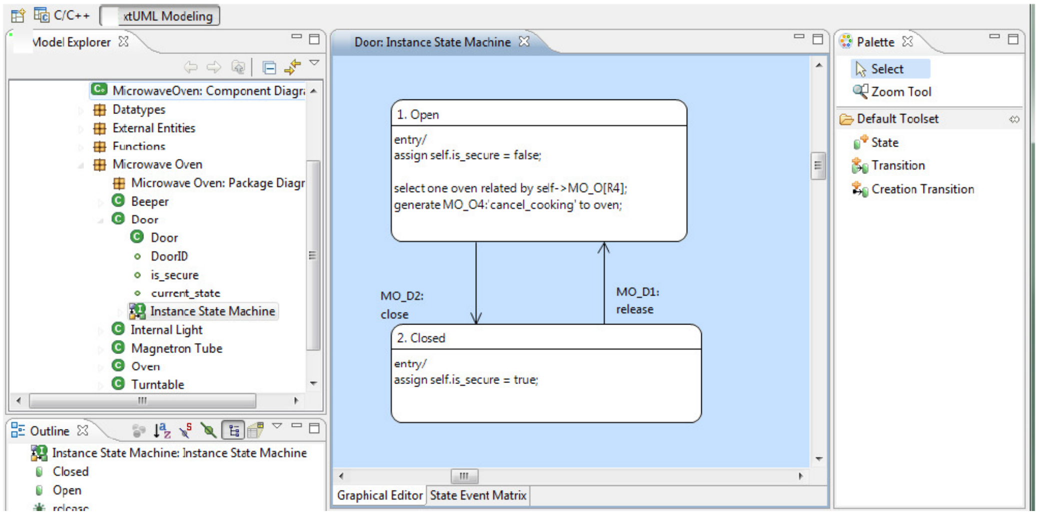
3

Calibration and validation of models
Minor updates of system design

Information – Model driven Architecture

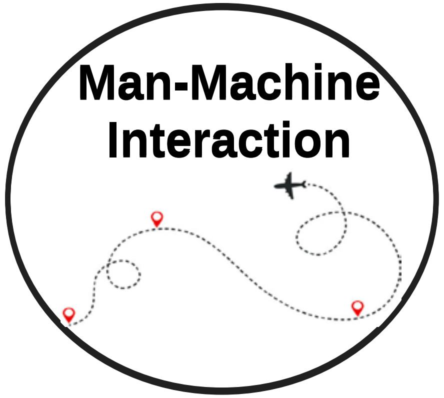
- Bridgepoint – xtUML selected for developing mission systems
  - Had been used successfully in sister organisation
- World authority in Domain Driven Development hired
- Extensive training programme
- Extensive investment in code generator development

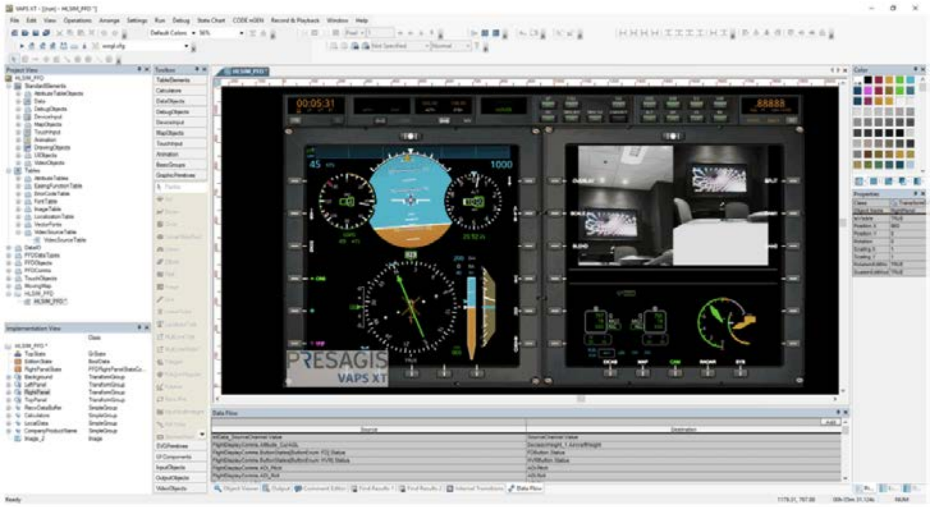




Human Machine Interaction

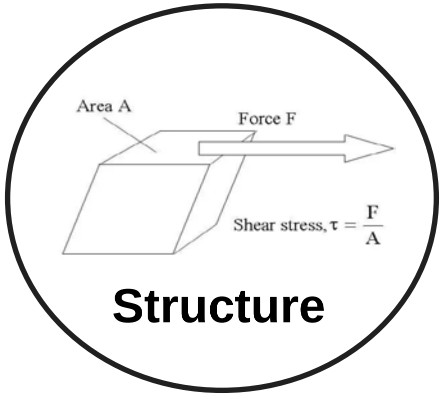
- Presagis VAPS XT for generating cock-pit display information
- Extensive experience within the organisation
- Qualified code generator
- ARINC-661 support

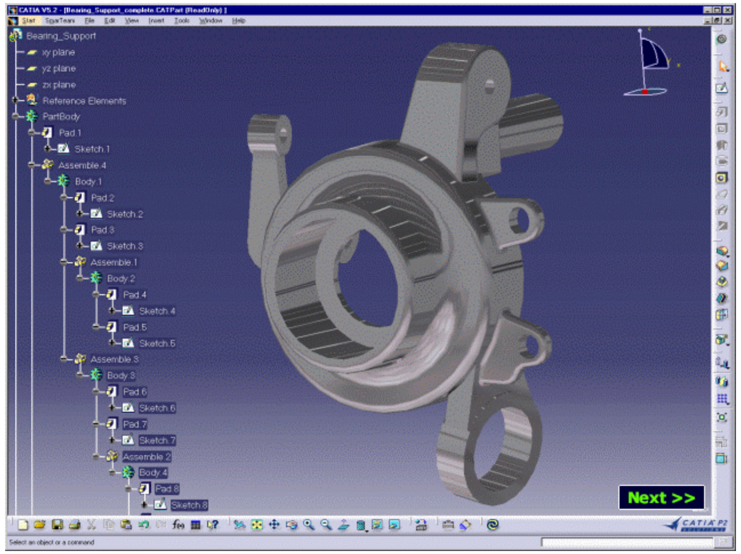




Structural design to production

- Dassault suite (Catia, Delmia etc) used for all activities from design to production
  - Validated at the Neuron demonstrator
- Integrated flow
- Digital workstations on the production line
  - No drawings at all!
- Design managed in VPM, integrated configuration management system
- Extensive support organisation



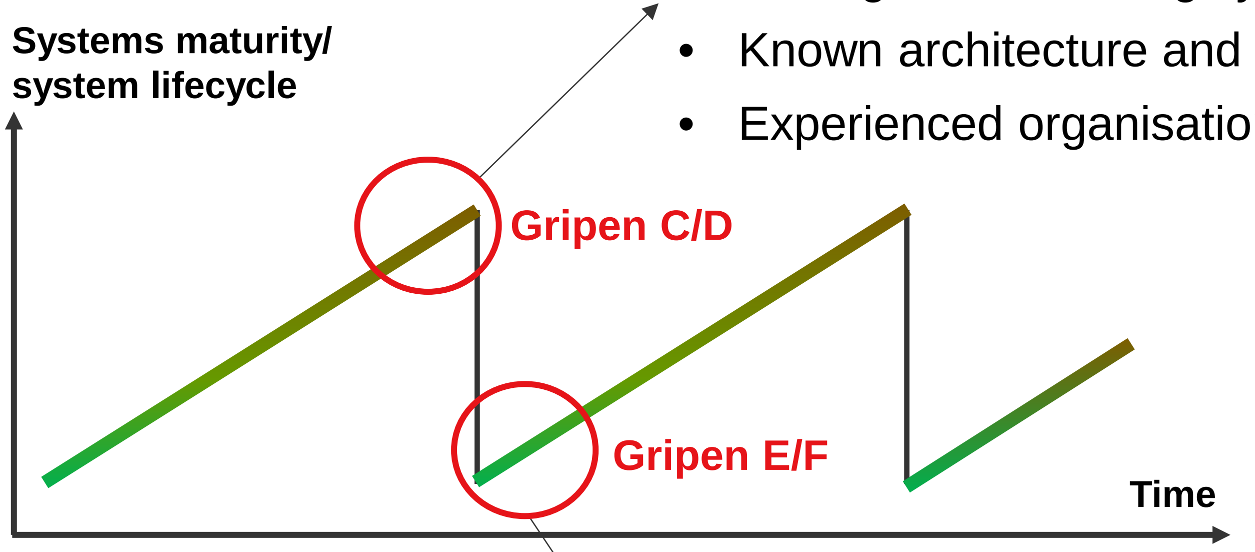


The lessons the models taught us

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

George Box (1919-2013)

Modelling different types of development



Brownfield

- Adding deltas to a highly mature system
- Known architecture and constraints
- Experienced organisation – in terms of Brownfield development

Greenfield

- No baseline system available
- New architecture – constraints are not known
- Inexperienced organisation – in terms of Greenfield development

"Profits from close attention, systematic reason, risk aversion, sharp focus, hard work, training and refined detail." (March 1999, p. 184)

EXPLOITATIVE LEARNING – THE BROWNFIELD ORGANIZATION

BROWNFIELD DEVELOPMENT promotes 'exploitative' learning, and the organization therefore expects:

- Learning to be goal-oriented and that expected outcomes and gains can be described.
- Management to reduce slack, facilitate coordination and communication, and to link activities to performance measures that can be monitored.
- Risky choices followed by failures, although they happen, are to be 'unnecessary'.

”Thrives on serendipity, risk-taking, novelty, free association, madness, loose discipline and relaxed control.” (March 1999, p. 184)

EXPLORATIVE LEARNING — THE GREENFIELD ORGANIZATION

GREENFIELD DEVELOPMENT promotes ‘explorative’ learning where the organization should expect:

- To learn in order to find new alternatives and new goals for development
- Experiments and projects involve high uncertainty and ambiguity, and outcomes and their merits may be difficult to define and difficult to manage
- Success is far from given, however, failures drive learning and therefore serve a purpose.

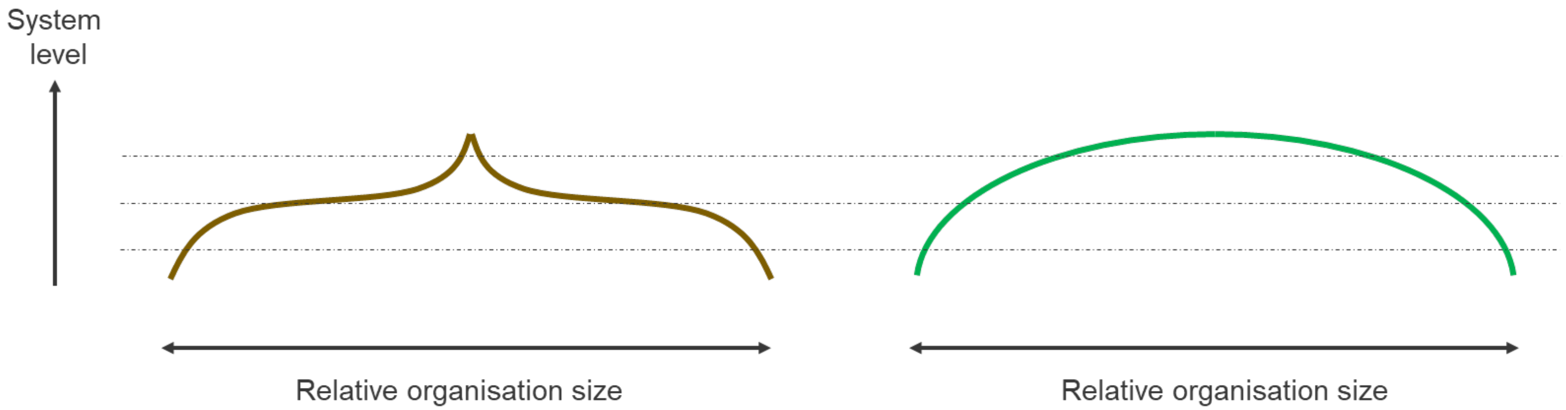
Management styles

Brownfield development

- Local risks
- Management can have a **weak** connection to the technique/realisation/ problem domain
- Management via allocation of whole problems – teams solve problems on their own

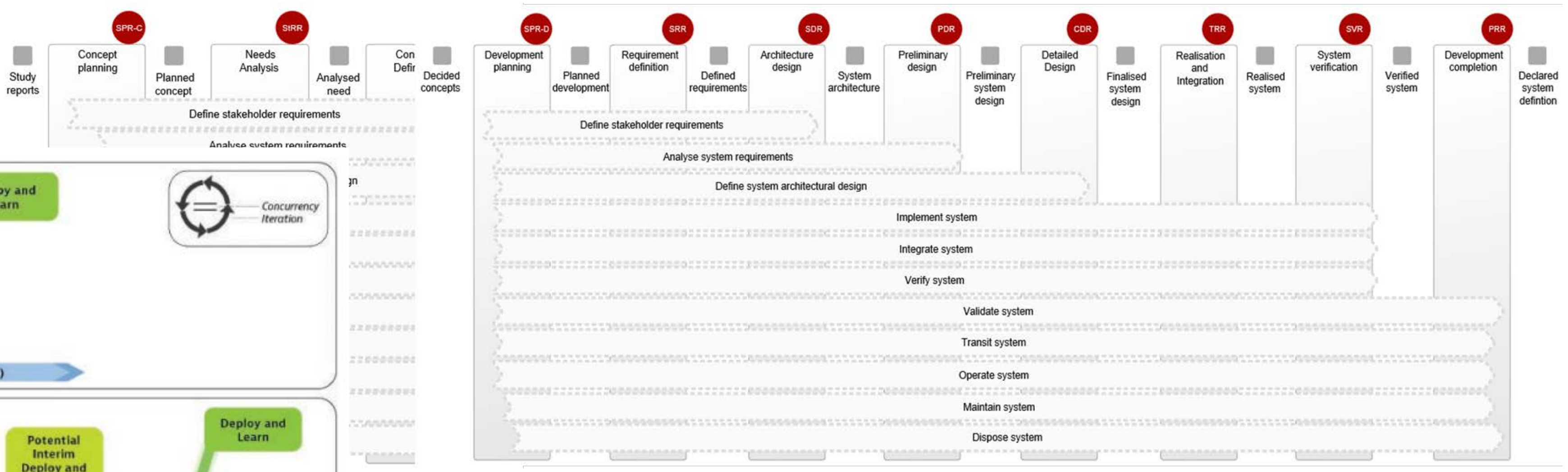
Greenfield development

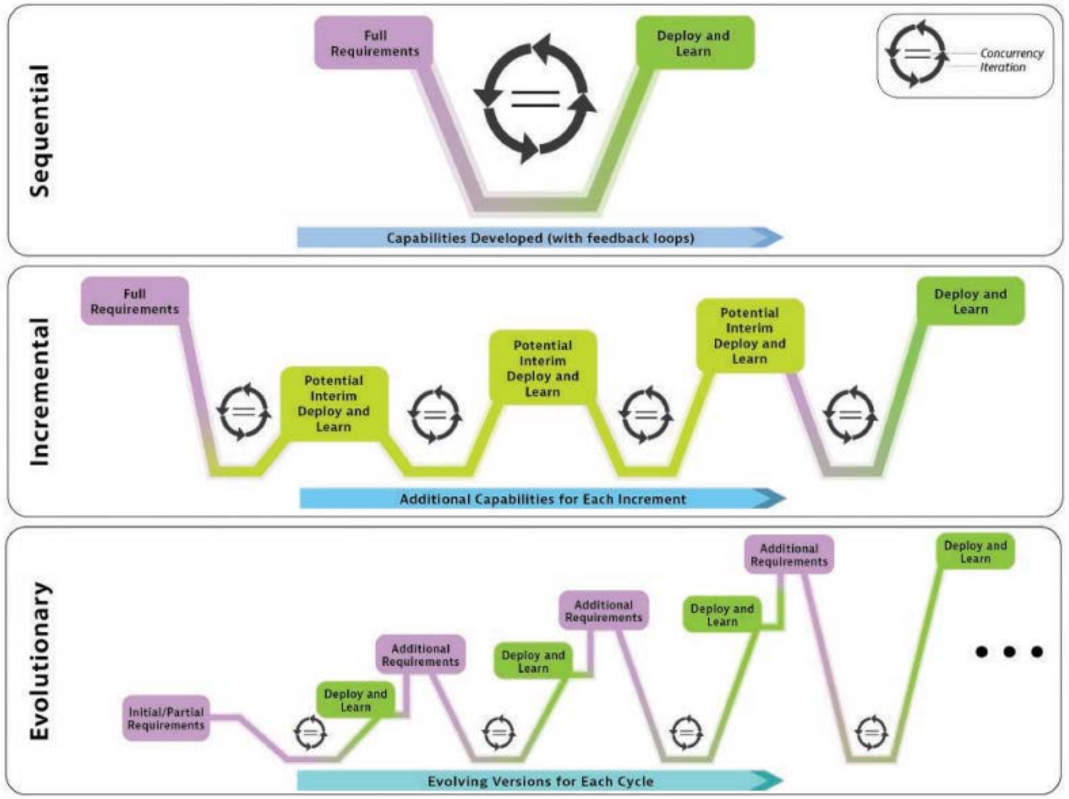
- Global risks
- Management needs a **strong** connection to the technique/realisation/ problem domain
- Management via structured systems engineering – allocating well-defined tasks to the teams



Can we assume that development is predictable?

The problem with the Vee models

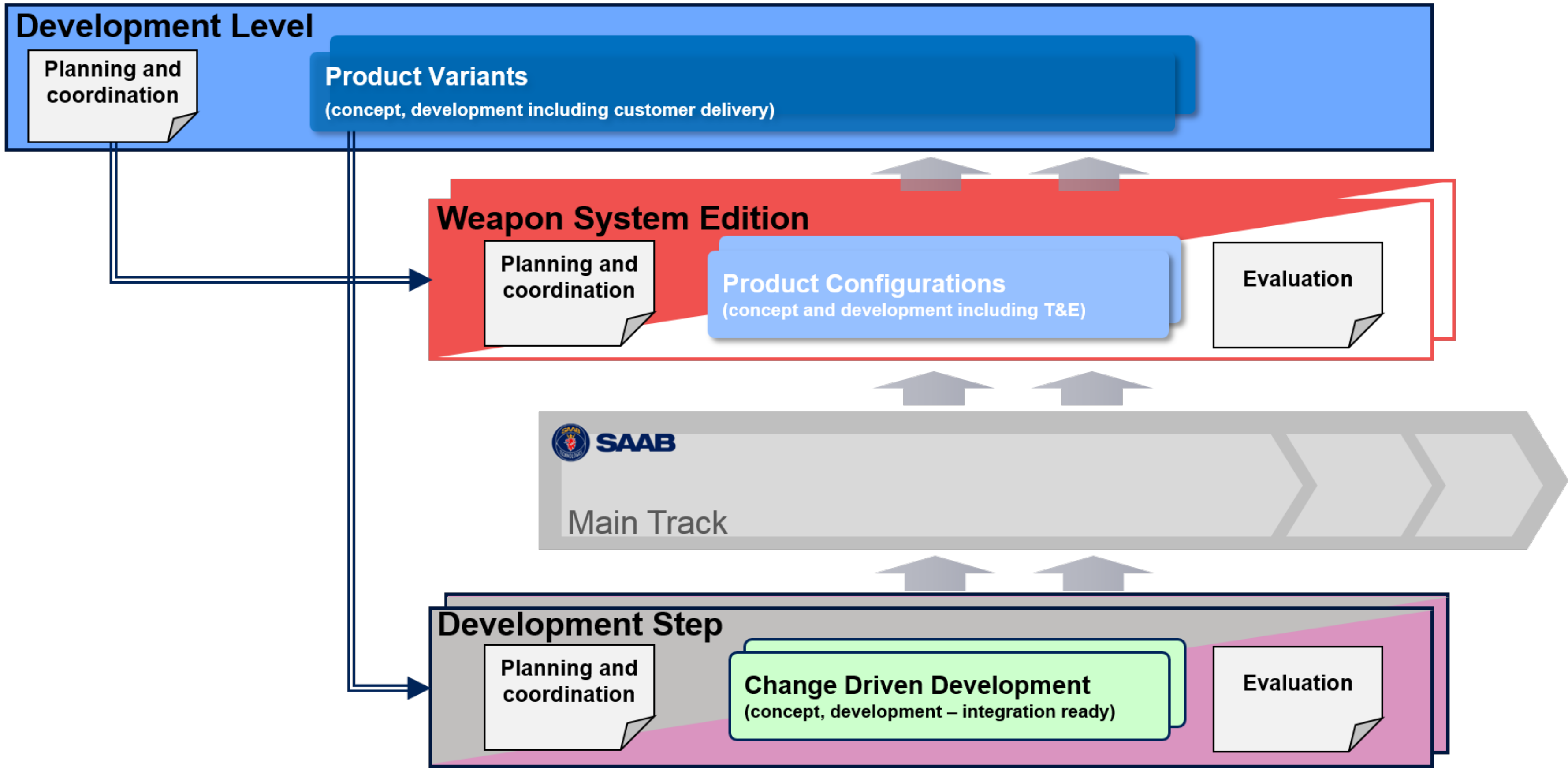




- Work started together is assumed to be integrated together
- Assumption: development is predictable

When future progress can't be predicted

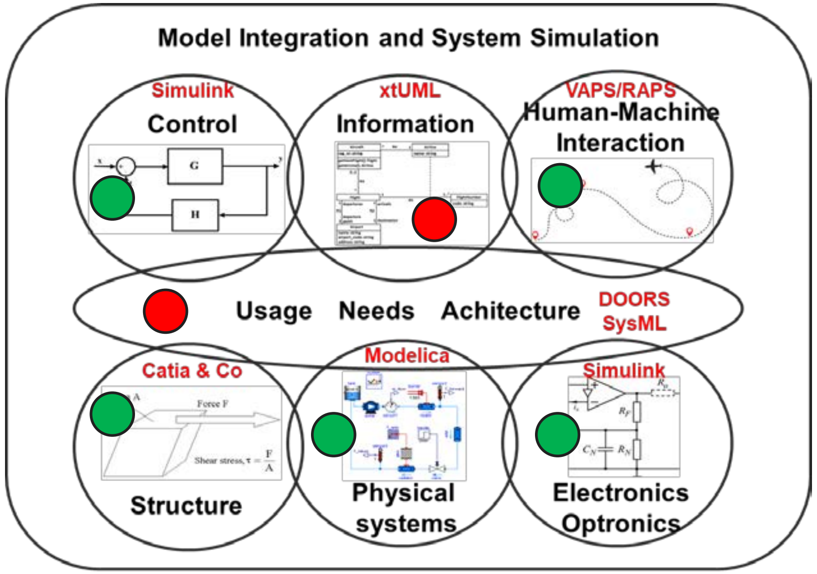
- From **plan-driven** to **integration-driven development**
  - **Anatomies** to manage integration opportunities
- Development is **asynchronous** to integration
- Make re-planning cost as low as possible

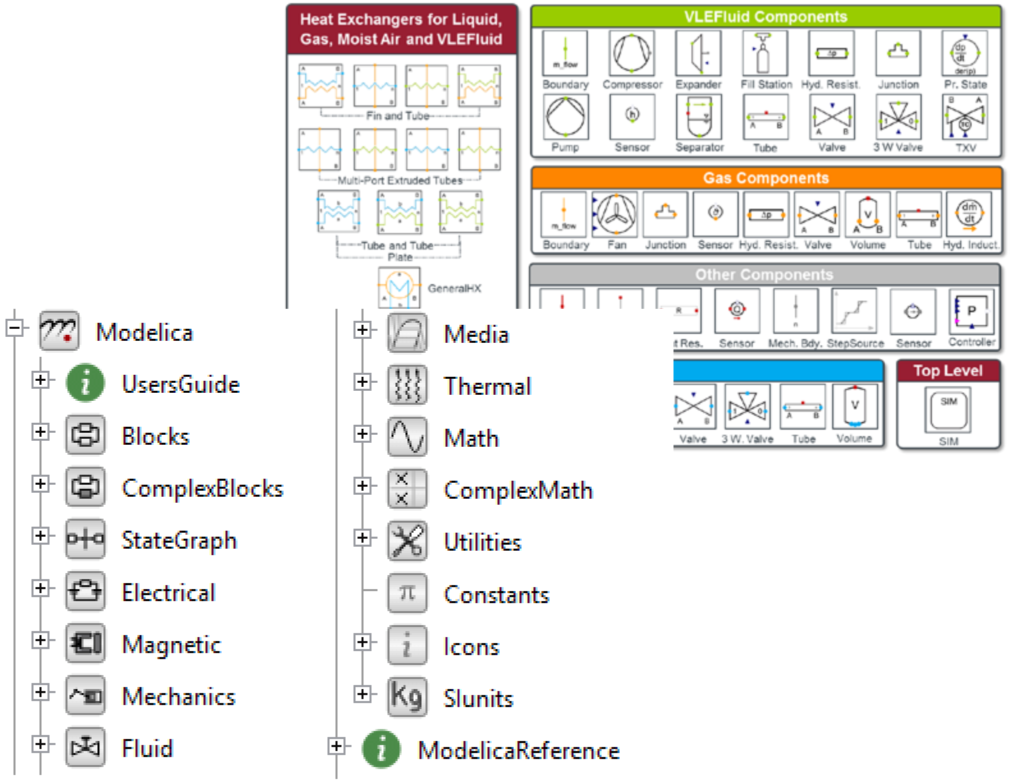


What the models taught us about language

The importance of block libraries

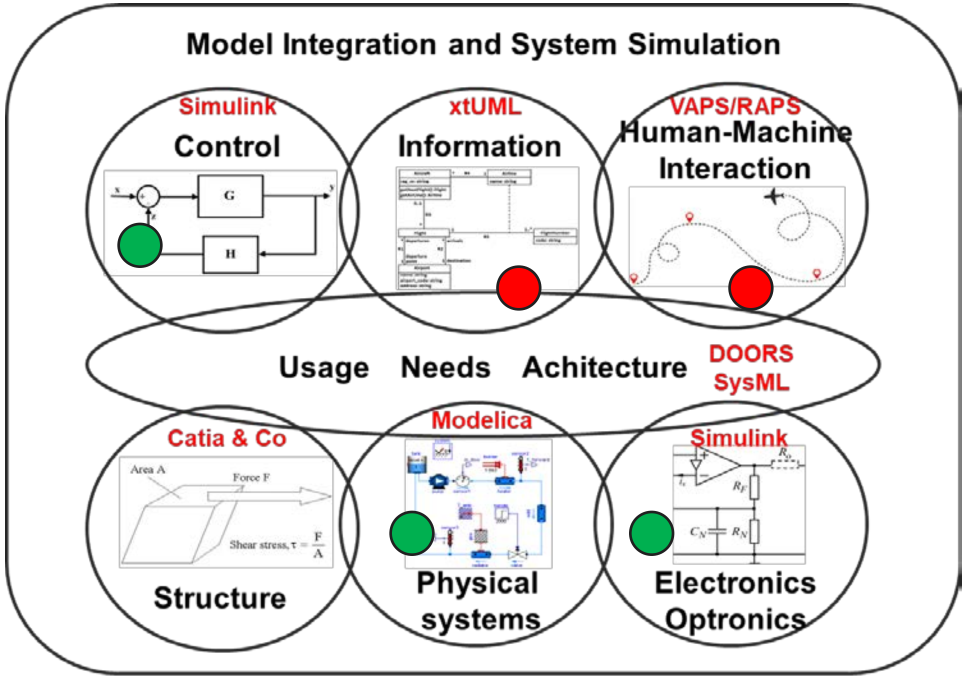
- Validated block libraries allow development teams to transition quicker to integration and verification
  - The cost of developing and verifying libraries is very high





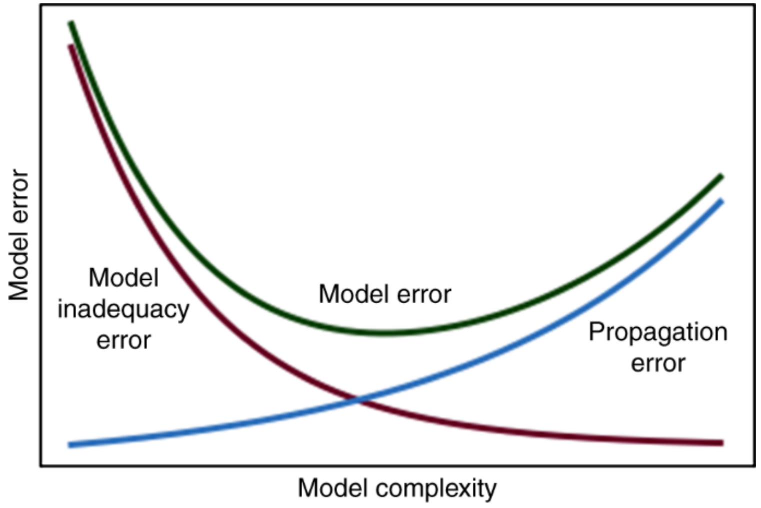
Code generation and integration

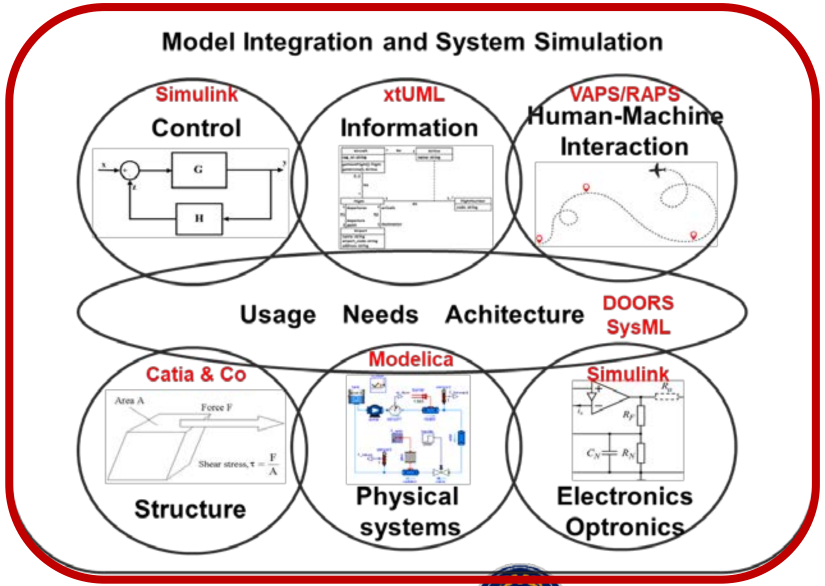
- What are the means for validating code generated by the tool?
  - Is there comprehensive simulation support?
  - Can the generated code be understood with a reasonable investment?



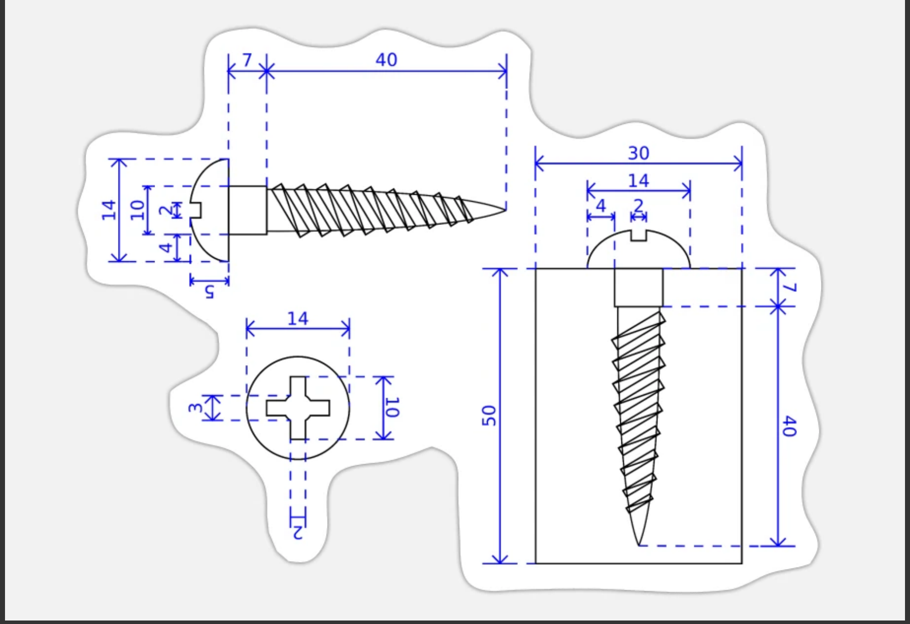
Creating good models – the rule book

- **Every model <u>must</u> have**
  - Well defined purpose
  - Known boundaries
  - Known limitations
  - Known fidelity
  - Known credibility
- When using models for simulation
  - Good understanding of the capabilities of the individual model
  - Operator must understand the
    - Detail and credibility of the simulation result
    - Relationship to actual product configuration

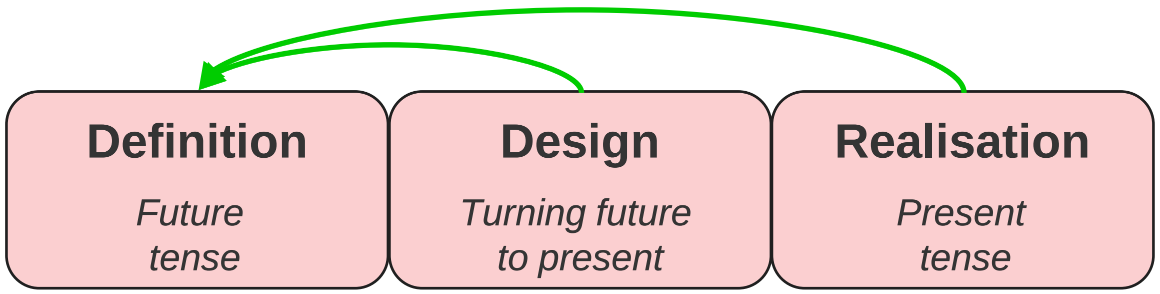




A framework for model based development



Proposed model framework



Definition model

Captures the **intended architecture**

Relatively undetailed

Used for communication and long-term memory, e.g. change management/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

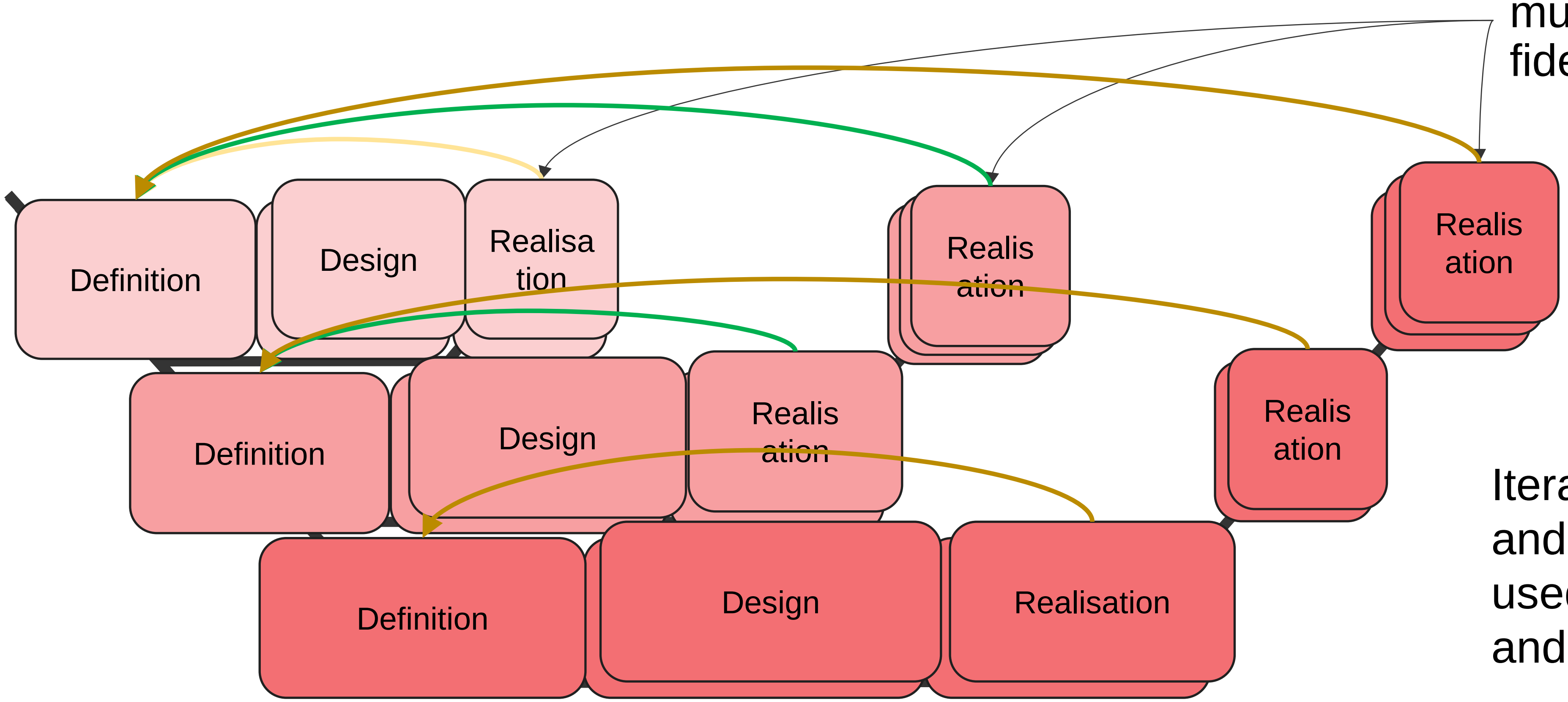
Realisation model (physical/virtual)

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

Feedback using virtual Realisations

Multiple virtual Realisations – multiple perspectives, fidelities and credibilities



Iterative feedback from virtual and physical Realisations are used to shape the Definition and Design models

Translation to the model world

Tenses and model types

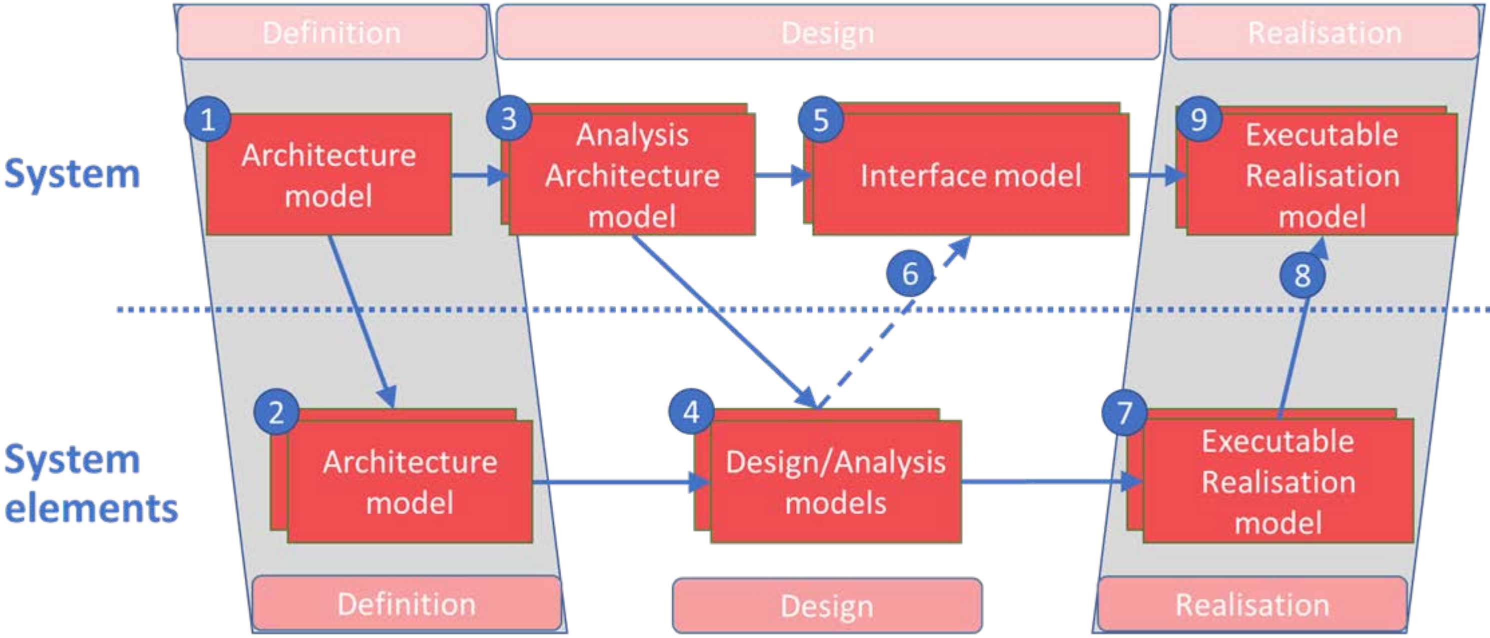
Architecture model

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

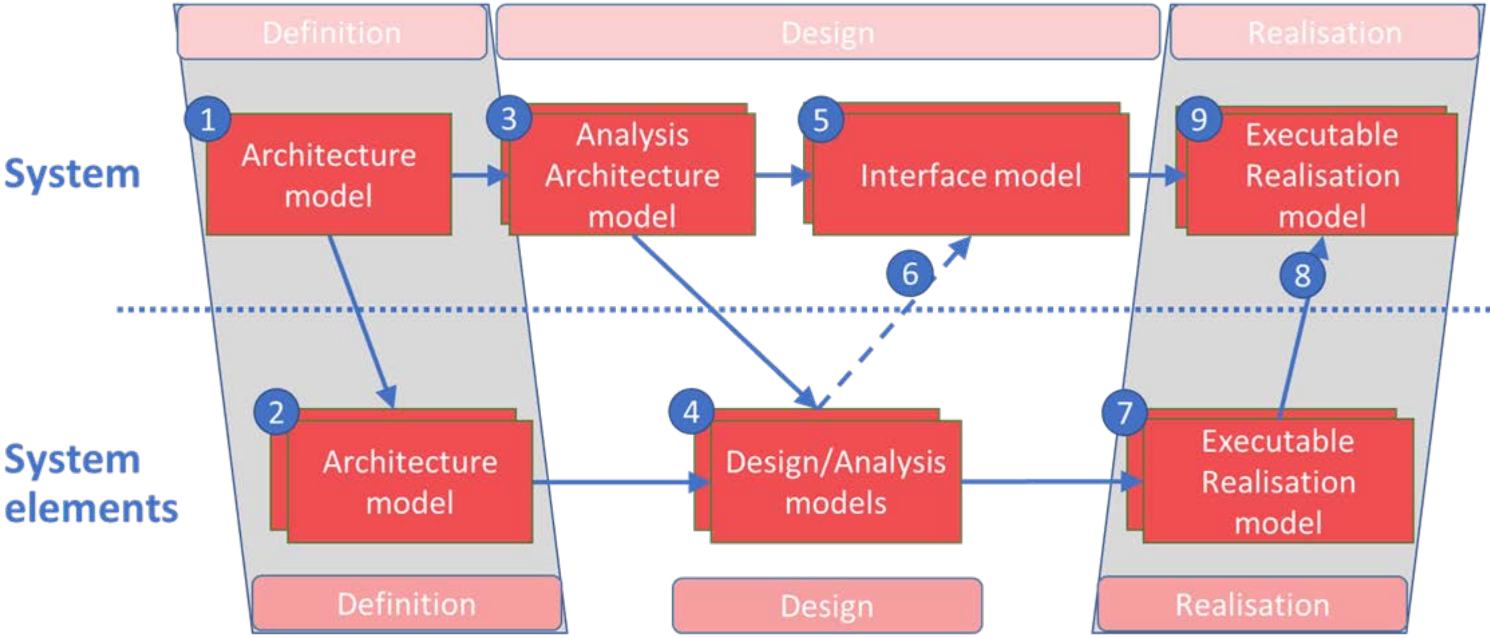
Design/Analysis model

Captures the emergent system design or system analyses

Interface model

Derived from the Architecture model and refined with design content

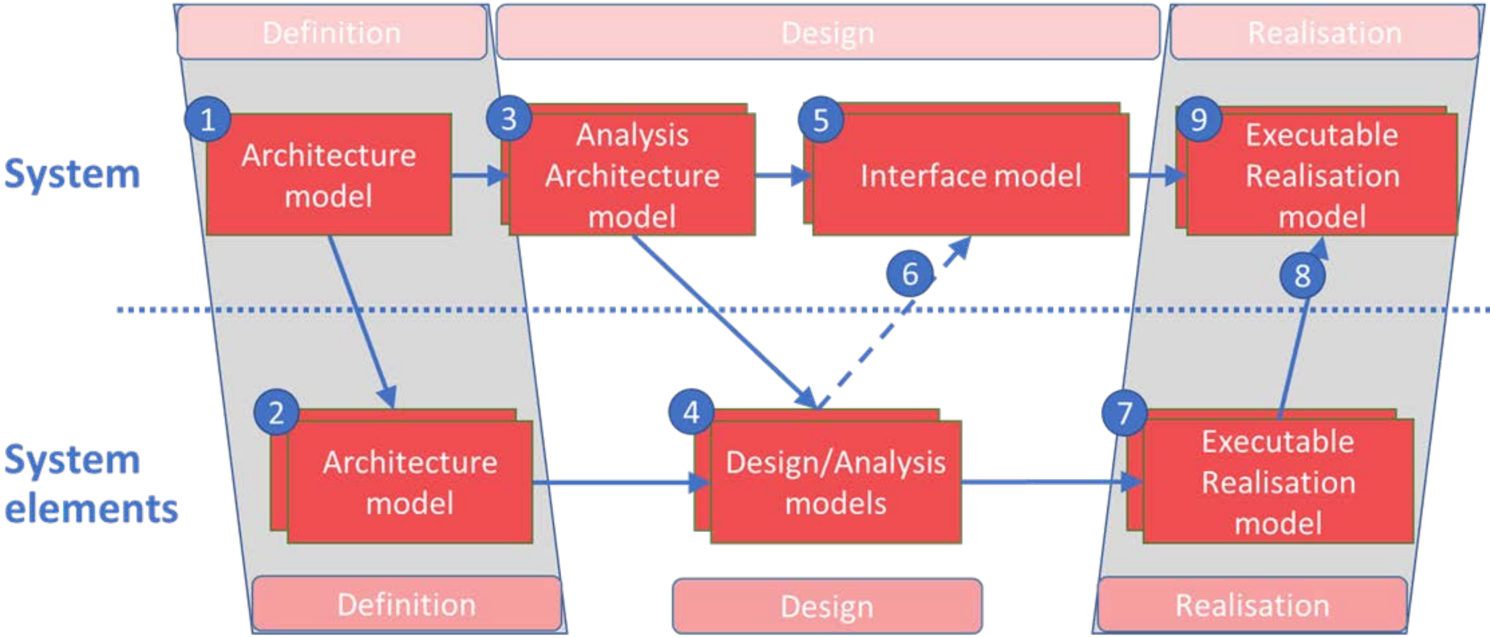
Purpose to provide the template for virtual and/or physical integration



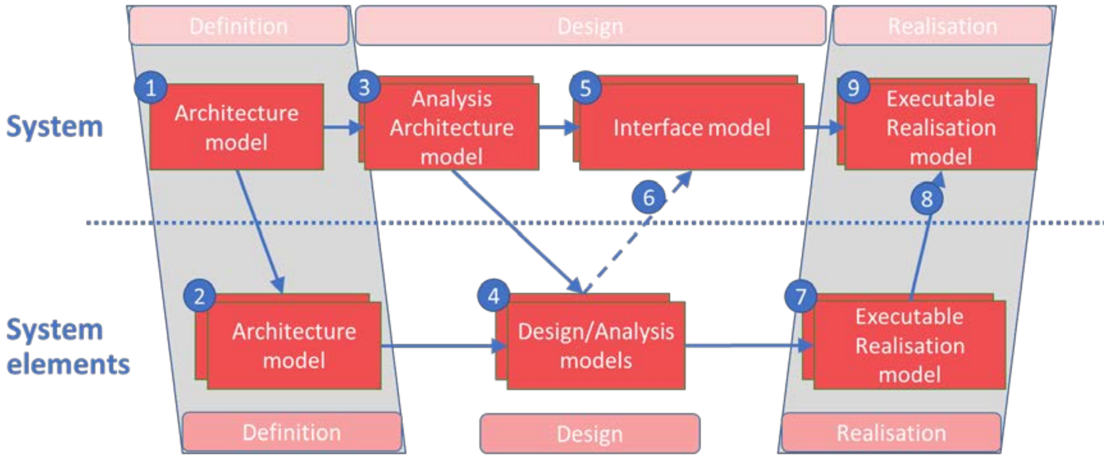
Tenses and model types

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 langauage 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**

Characteristics of a good model

Model characteristics	Architecture/Analysis architecture model	Design model	Executable realisation models	Interface models
Representation	Graphical **SysML, …**	Textual/Graphical **Modelica/Simulink/ Fortran, …**	- **FMI formatted**	Textual/Graphical **SSP/SysML v2**
Formality	Informal	Formal	Formal	Formal
Modelling approach	Descriptive	Descriptive/Analytic al	-	Descriptive
Relative fidelity	Low	High	High	High

Implications for the future

Summary

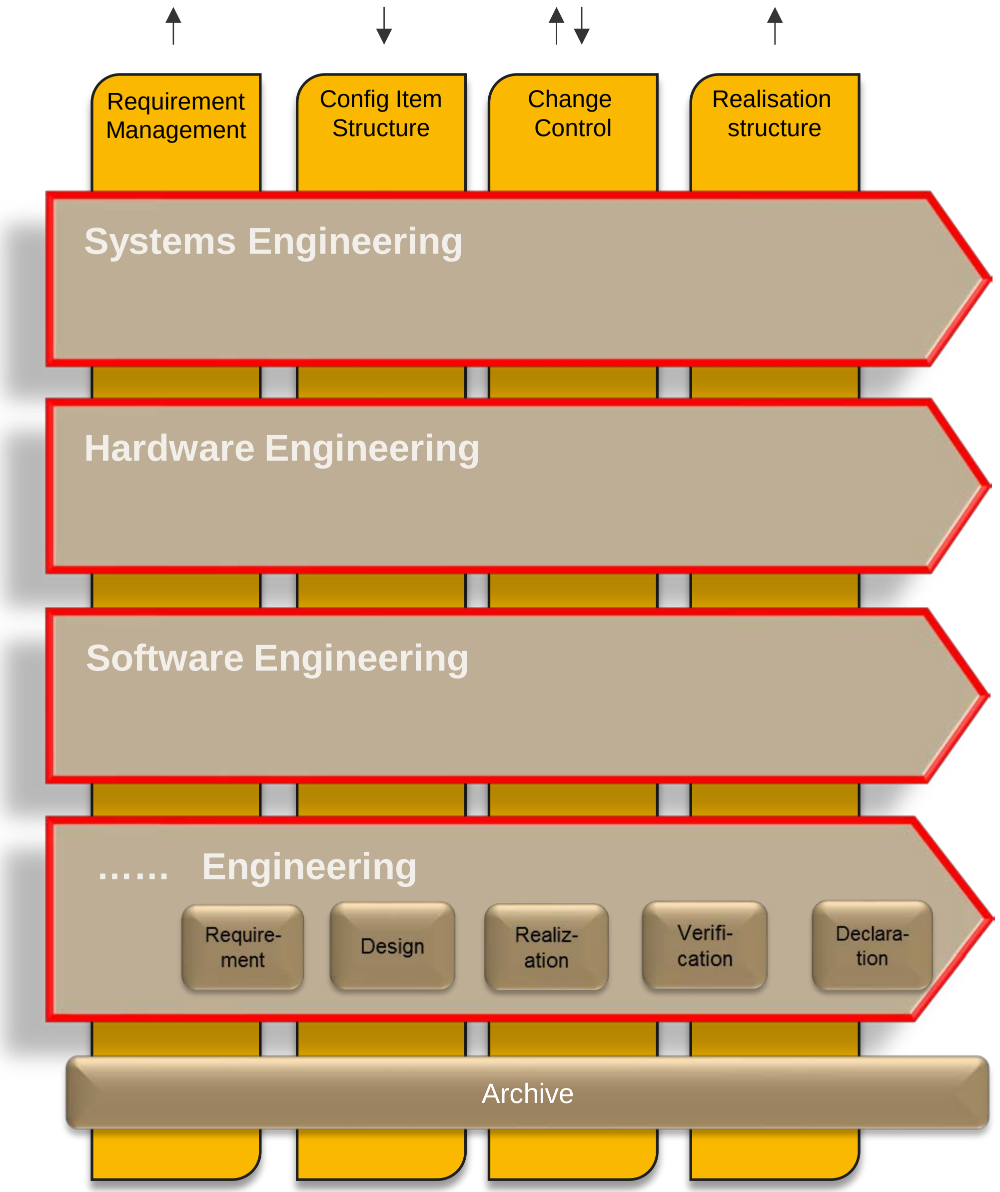
- Need to use multiple languages and methods in heterogeneous system development
- Critical systems – configuration control is essential
- Transition from stand-alone tools to integrated development environments
  - Configuration management an integrated capability
- Ensure that all stakeholders have access to relevant information
  - Desire to go from powerpoint as information carrier to information generated from the tool environments

Architecting the integrated development environment

Federated PLM

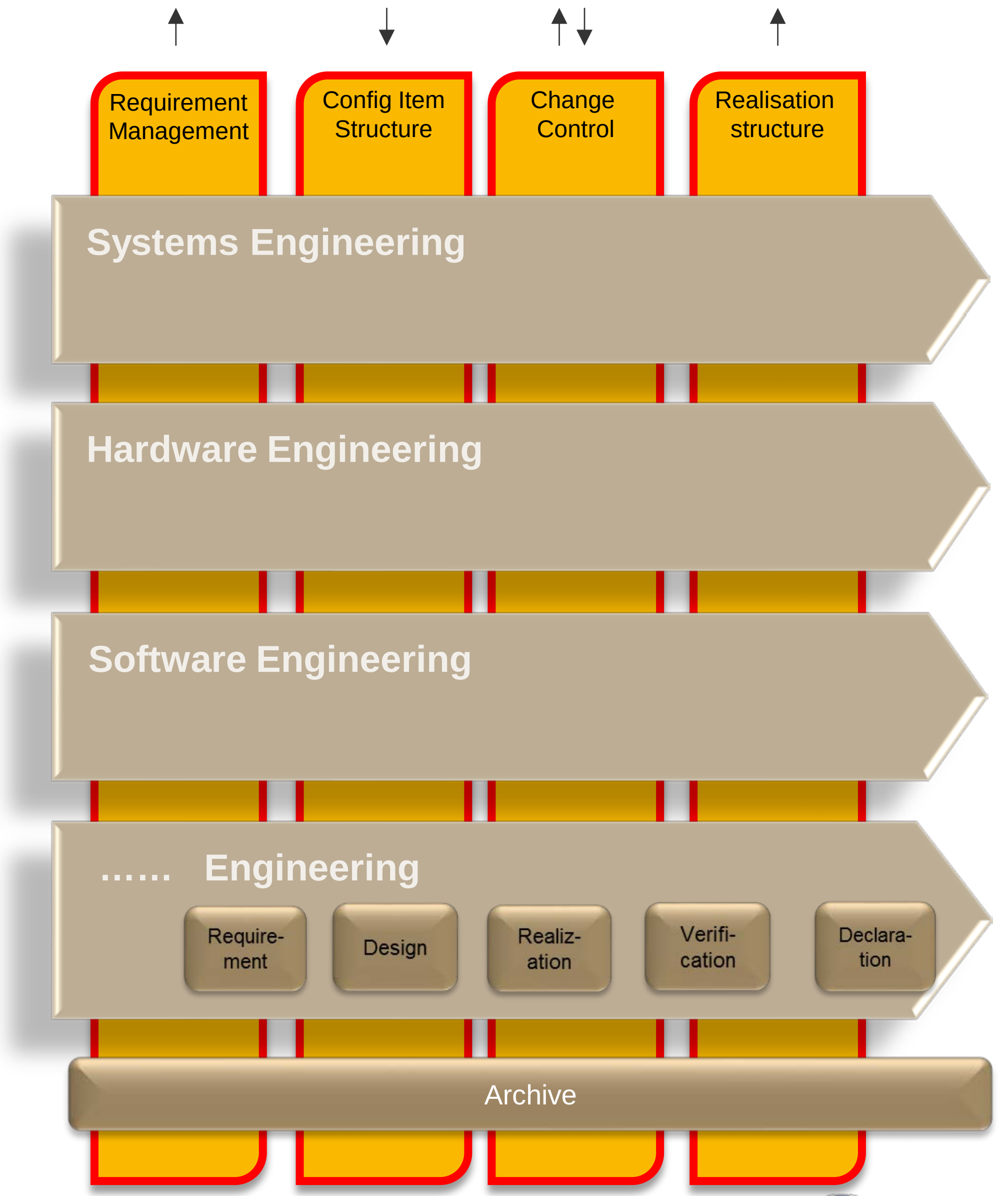
Modularity

- Optimise support for each engineering discipline
  - Maximise automation, as provided by the supplier
  - Minimise application family switching
- Bring together management and engineers in a single environment
  - E.g., Change management and Status reporting
- Ability to upgrade individual capabilities independent of others
- Redundant capabilities accepted
- Ability to replace environment without upsetting the complete PLM landscape



Traceability

- Need capability to ensure traceability and integrity of product data
- Traceability dimensions between engineering discipline environments
  - Requirements
  - Configuration item structure
  - Change management
  - Realization
- **Configuration Management** capability required for Requirements Traceability, Configuration item structure and Realization structure
  - Versions and baseline capability
- The OSLC standard offers the desired capabilities
  - Exploit for low cost and high quality integrations



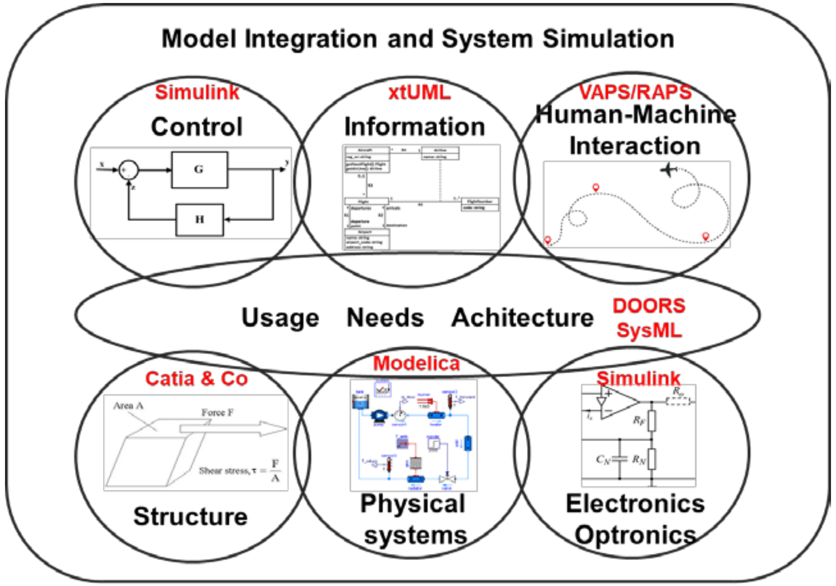
Is standards-based linking feasible

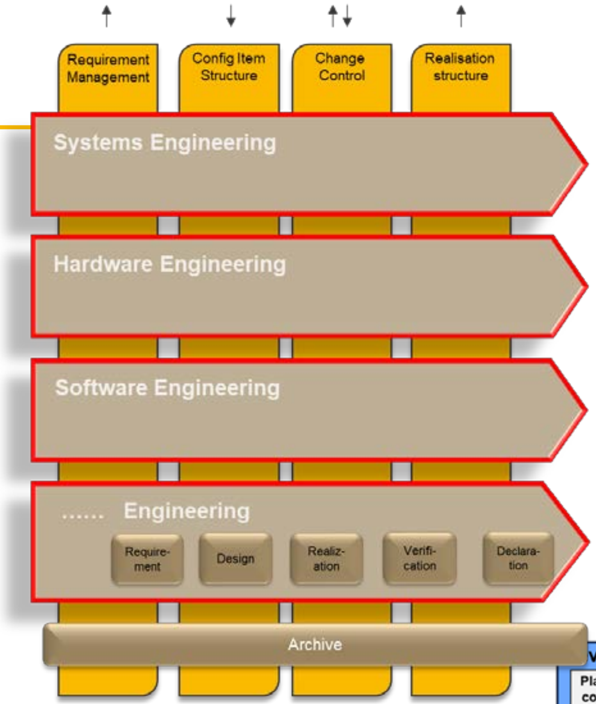
- **Federated PLM – feasibility dimensions**
  - Technical feasibility
    - Does OSLC offer industrial strength solutions for integrating stand-alone PLM systems?
  - Development efficiency
    - Does a federated PLM environment offer improved productivity potential in the short and long term compared to a monolithic, single supplier solution?
  - Operational feasibility
    - Can a federated PLM environment be maintained over time?
  - Realisation effectivity
    - Can OSLC interfaces be implemented and maintained at a reasonable cost?

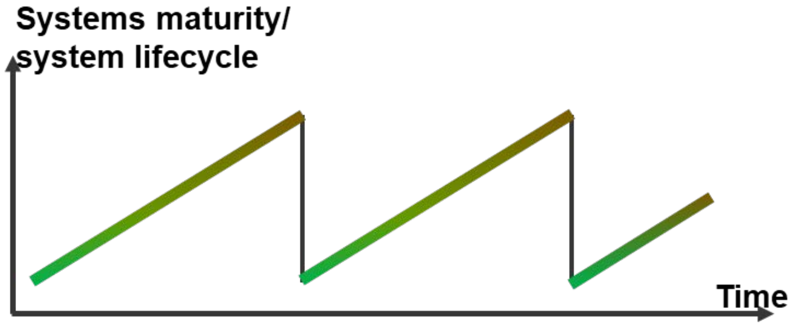
What about AI?

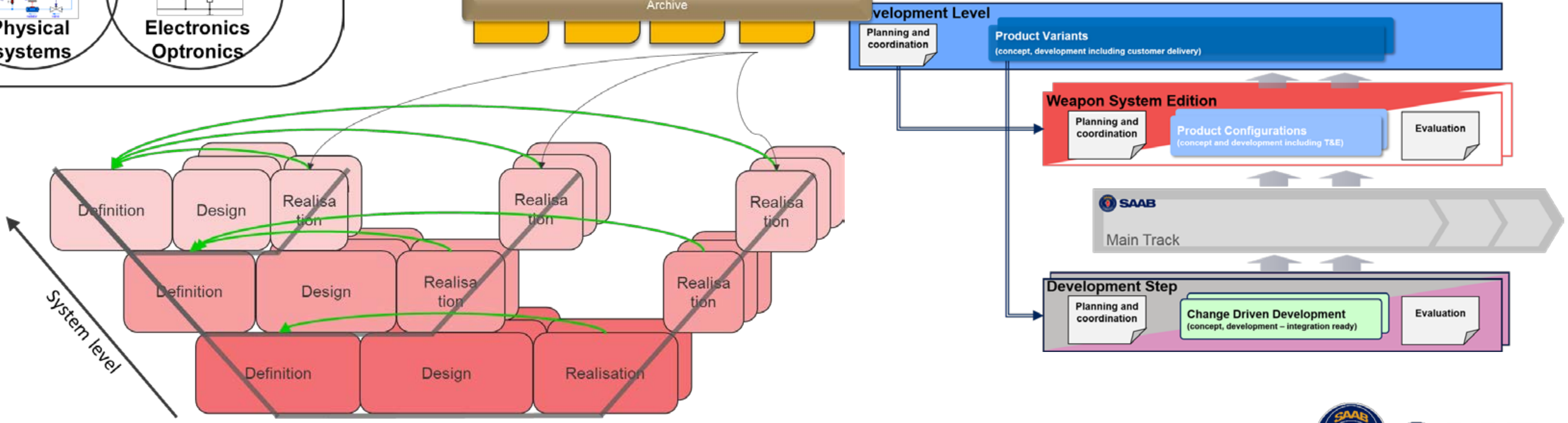
Conclusions

Conclusions









References

References

- Cederberg, L., Axehill, J. W., & Herzog, E. (2019). Experience from a Program for Accelerating the Creation of T-shaped Technical Leaders. *INCOSE International Symposium*, *29*, 707–722.
- Törngren, M., Asplund, F., Ericson, T., Granbom, C., Herzog, E., Lu, Z., … Others. (2020). Competence networks in the era of CPS--lessons learnt in the ICES cross-disciplinary and multi-domain center. *Cyber Physical Systems. Model-Based Design: 9th International Workshop, CyPhy 2019, and 15th International Workshop, WESE 2019, New York City, NY, USA, October 17-18, 2019, Revised Selected Papers 9*, 264–283. Springer International Publishing.
- Axehill, J. W., Herzog, E., Tingström, J., & Bengtsson, M. (2021). From Brownfield to Greenfield Development--Understanding and Managing the Transition. *INCOSE International Symposium*, *31*, 832–847.
- Herzog, E., Nordling Larsson, Å., & Tingström, J. (2022). Genesis--an Architectural Pattern for Federated PLM. *INCOSE International Symposium*, *32*, 782–791.
- Herzog, E., Larsson, Å. N., Sundin, O., & Goman, A. F. (2022). A 4-Box Development Model for Complex Systems Engineering. *INCOSE International Symposium*, *32*, 233–248.
- Axehill, J. W., & Herzog, E. (2022). Don’t Mix the Tenses: Managing the Present and the Future in an MBSE Context. *INCOSE International Symposium*, *32*, 824–838.

References

- Hällqvist, R., Naeser, J., Axehill, J. W., & Herzog, E. (2022). Heterogeneous Systems Modelling in Support of Incremental Development. *ICAS 2022*.
- Herzog, E., Axehill, J. W., Larsson, Å. N., & Aeronautics, S. (2022). Perspectives on models. *Proceedings of the INCOSE Workshop EMEA WSEC, Sevilla, Spain*.
- Herzog, E., Axehill, J. W., & Larsson, Å. N. (2023). Boxing Configuration Management--Configuration Change Management Meets the 4-Box Development Model. *INCOSE International Symposium*, *33*, 71–85.
- Axehill, J., Larsson, Å. N., & Herzog, E. (2023). Don't Look Outside the Box--Configuration Management Meeting a 4-box Development Model. *Center for Model-Based Cyber-Physical Product Development*, *31*(17), 22–22.
- Herzog, E., Tingström, J., Axehill, J. W., Larsson, Å. N., & Jouannet, C. (2024). From tears to tiers--architectural principles for federated PLM landscapes. *Proceedings of the Design Society*, *4*, 593–602.
- Hällqvist, R., Herzog, E., Axehill, J. W., & Palmer, J. R. (2024). Excuse me Sir/Madam, which Model? *INCOSE International Symposium*, *34*, 1560–1578.
- Herzog, E. (2024). Enabling Digital Engineering with Federated PLM--Experiences from the Heliple-2 Project. *34th INCOSE International Symposium*.