

# Towards Configuration Support for Collaborative Simulator Development

– A Product Line Approach in Model Based Systems Engineering

*CoMetS 2011, Paris*

Henric Andersson, Magnus Carlsson, Johan Ölvander

Dept. of Management and Engineering  
Linköping University, Sweden

&

Saab Aeronautics, Linköping, Sweden

# Presentation of the presenter

- MsC in Control Engineering (1994)
- Modeling and simulation of power plants at ABB
- Saab Aeronautics
  - Flight Control
  - Project Management / Systems Engineering
  - Product Line Engineering
  - Research / PhD (2005 -> 2012)

# Content

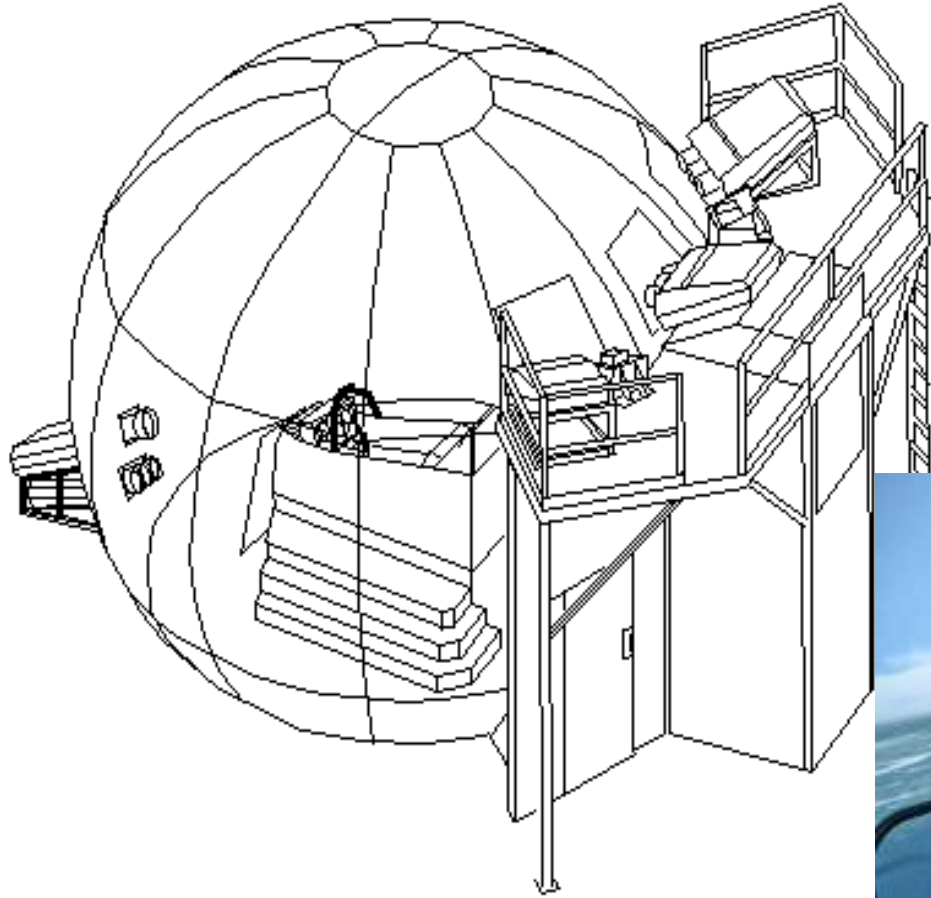
- Context - Simulator Usage
- Model Based Systems Engineering
- Product Line Engineering
- Example application
- Configurator implementation
- Conclusions & Further work

# Simulator example 1(3)



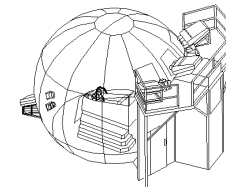
Engineer in the loop; Development, Training

# Simulator example 2(3)



Pilot in the loop; Training, verification

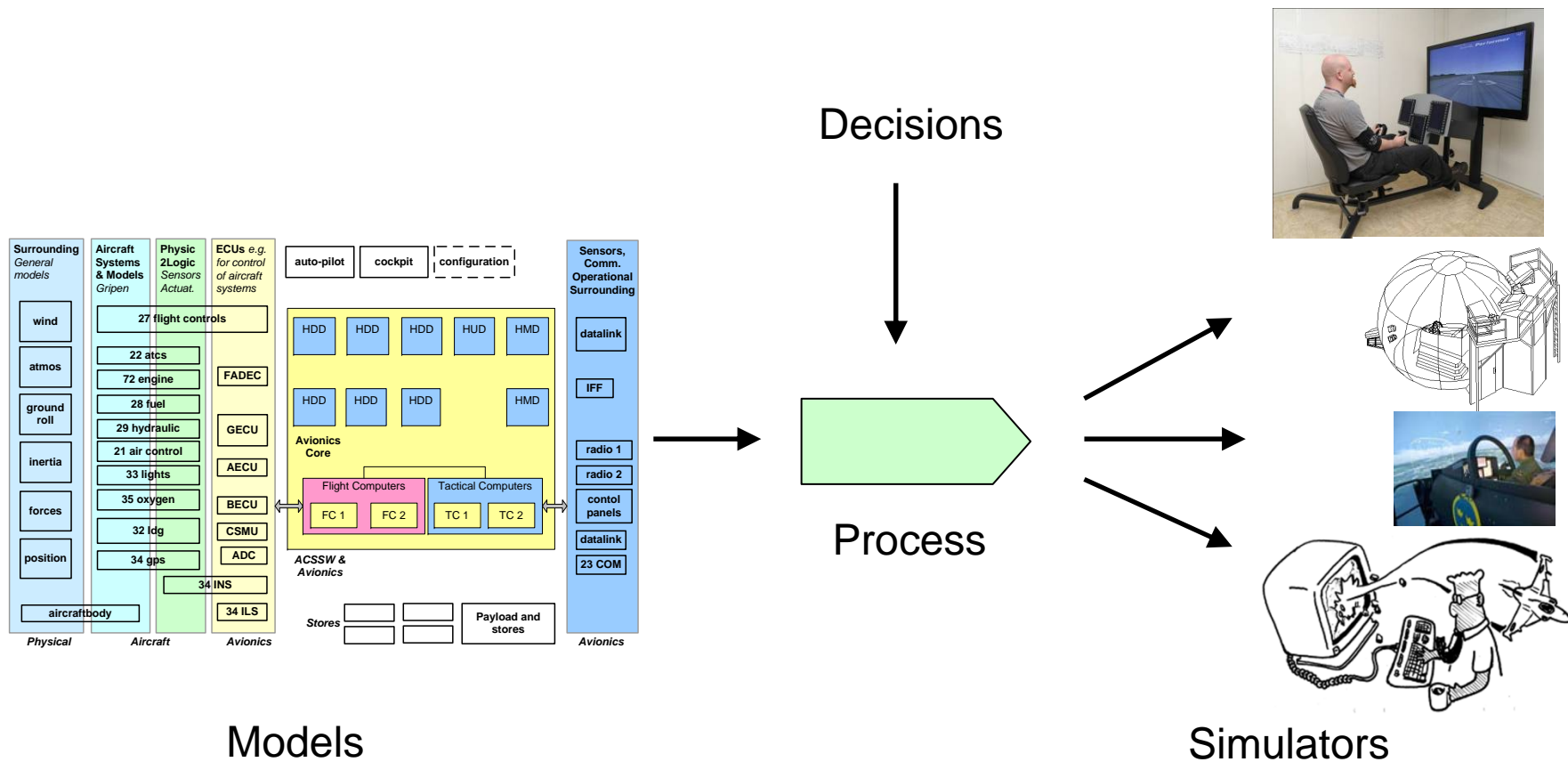
# Simulator example 3(3)



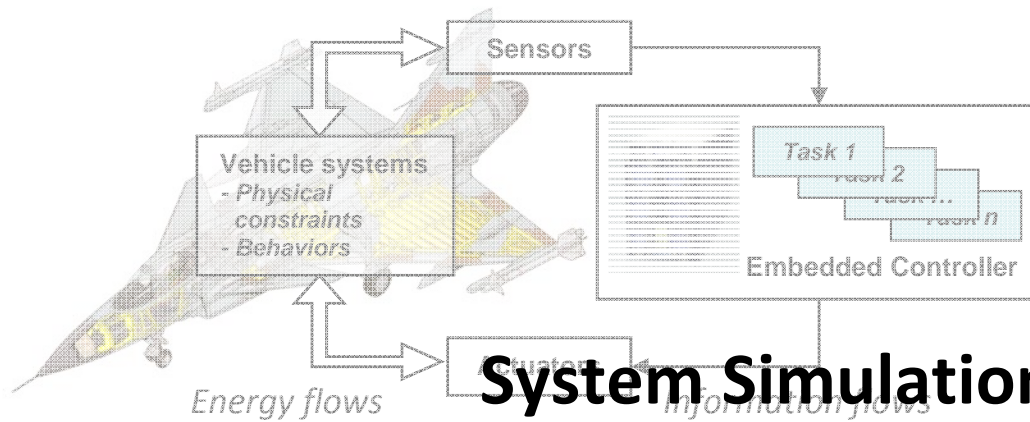
Desktop; batch simulation

# Simulator creation from models

The Product Line Approach – a concept for reuse

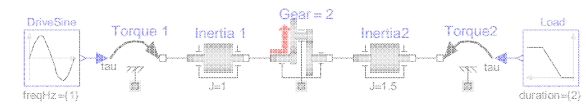
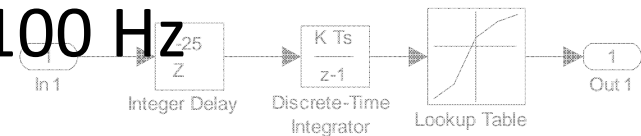
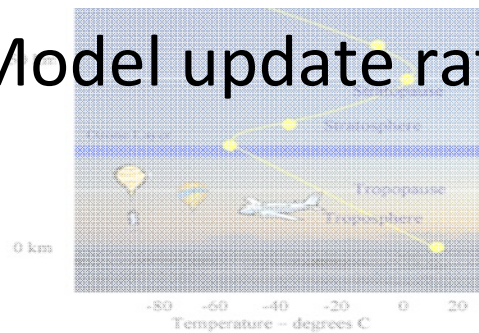
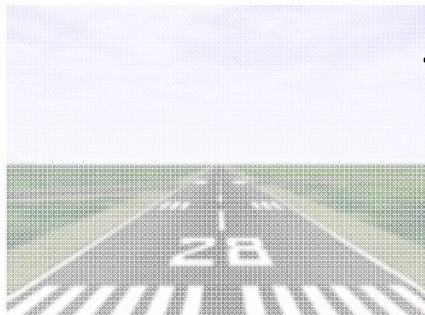
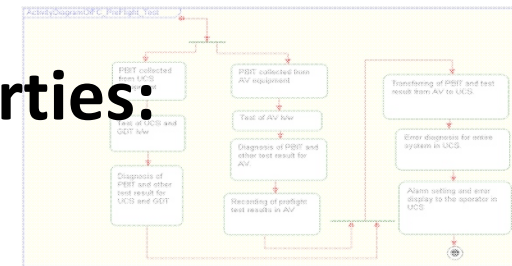
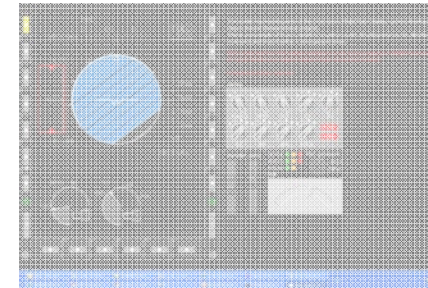


# MODEL BASED DEVELOPMENT



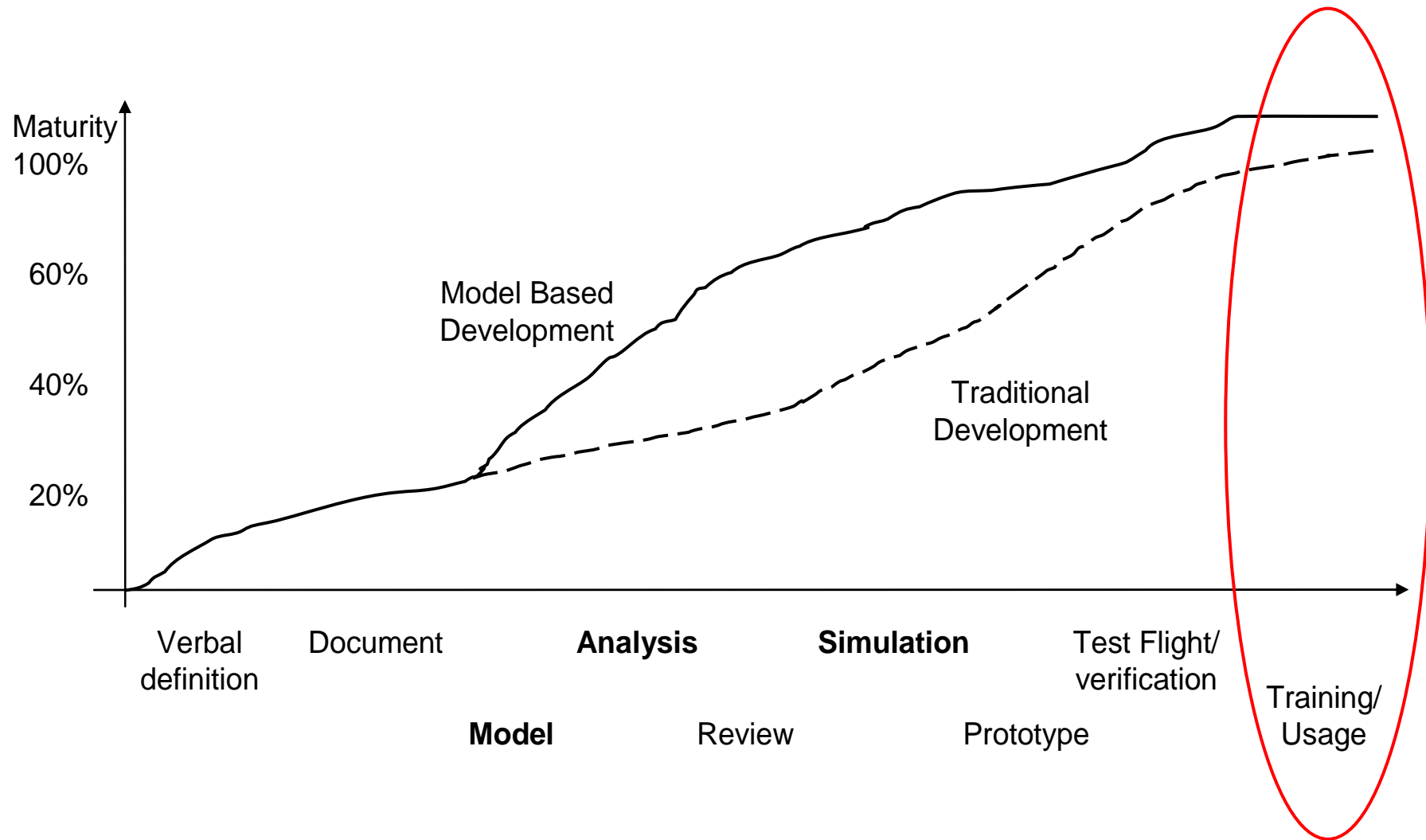
## System Simulation Properties:

- Global Scheduler
- Local solvers
- Model update rate 1-100 Hz

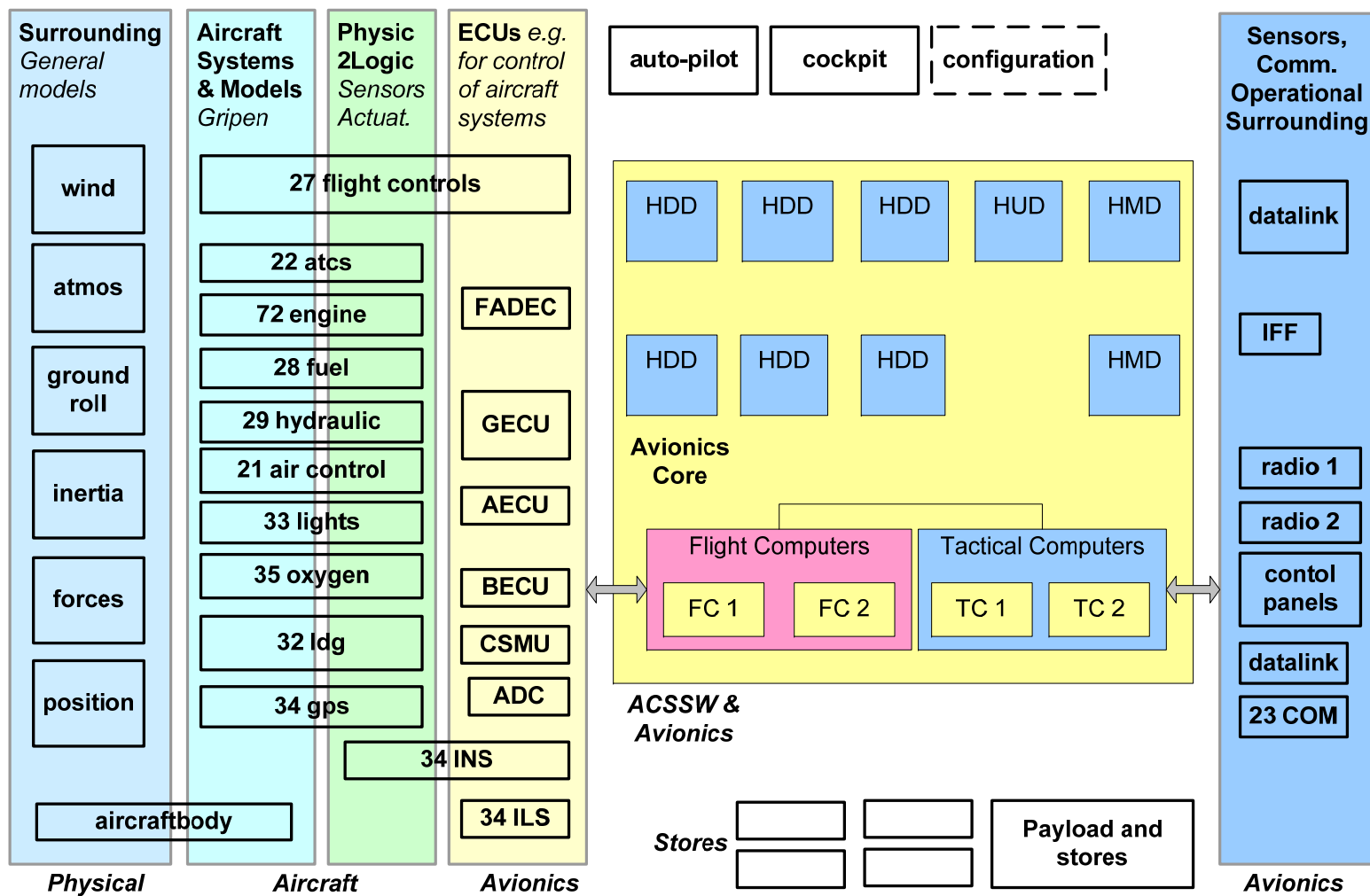




# THE VALUES OF MBSE



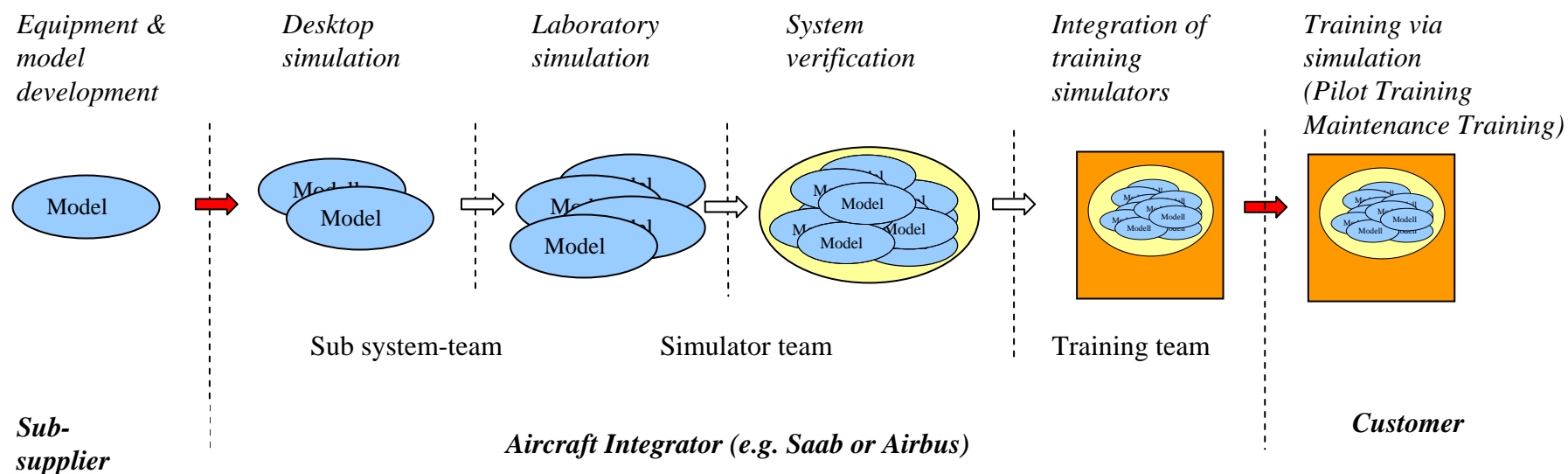
# Model overview and classification



# FLOW OF SIMULATION MODELS

## ➤ Origin of models :

- Deliverables from sub suppliers (e.g. Engine)
- In-house development of aircraft specific models (e.g. Mass & Inertia)
- In-house development of simulator specific models (e.g. Cockpit)

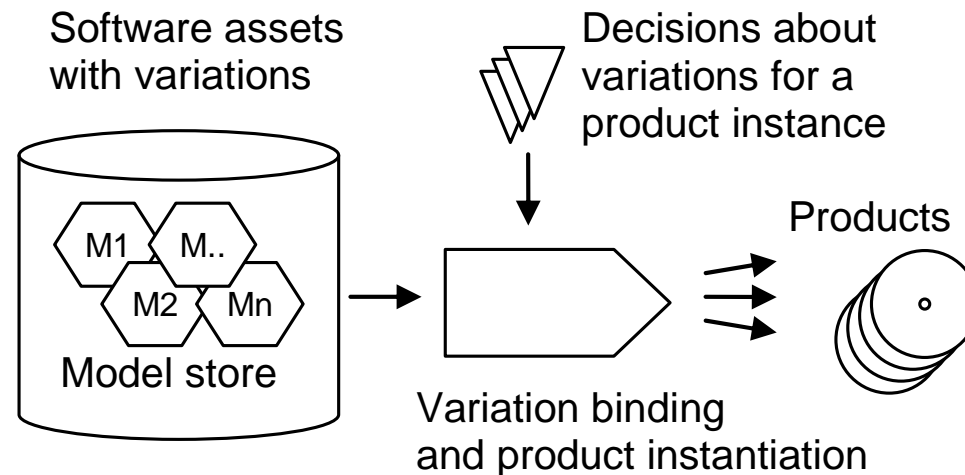


# Challenges in Aircraft Simulation

Some challenges in set-up and support of large-scale simulations:

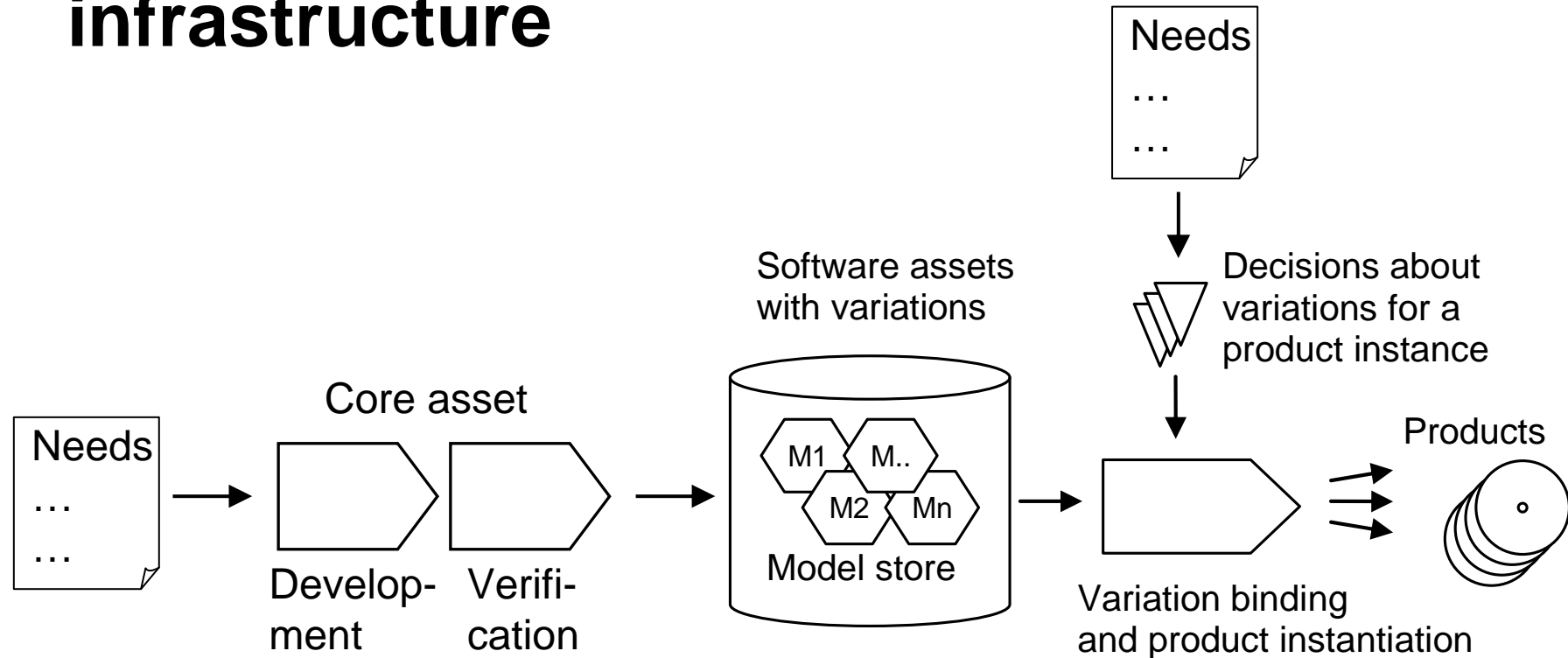
- Different operating systems & simulation platforms
- Many models ~100 including “legacy codes”
- Variants of the systems that the models represent
- Variants of “the same” model, e.g. different levels of fidelity
- Versions of models, e.g. due to error correction
- Parametric models with different sets of System Parameters
- Lack of standards & tools for collaboration

# Basic components of M&S Software Product Line (SPL)



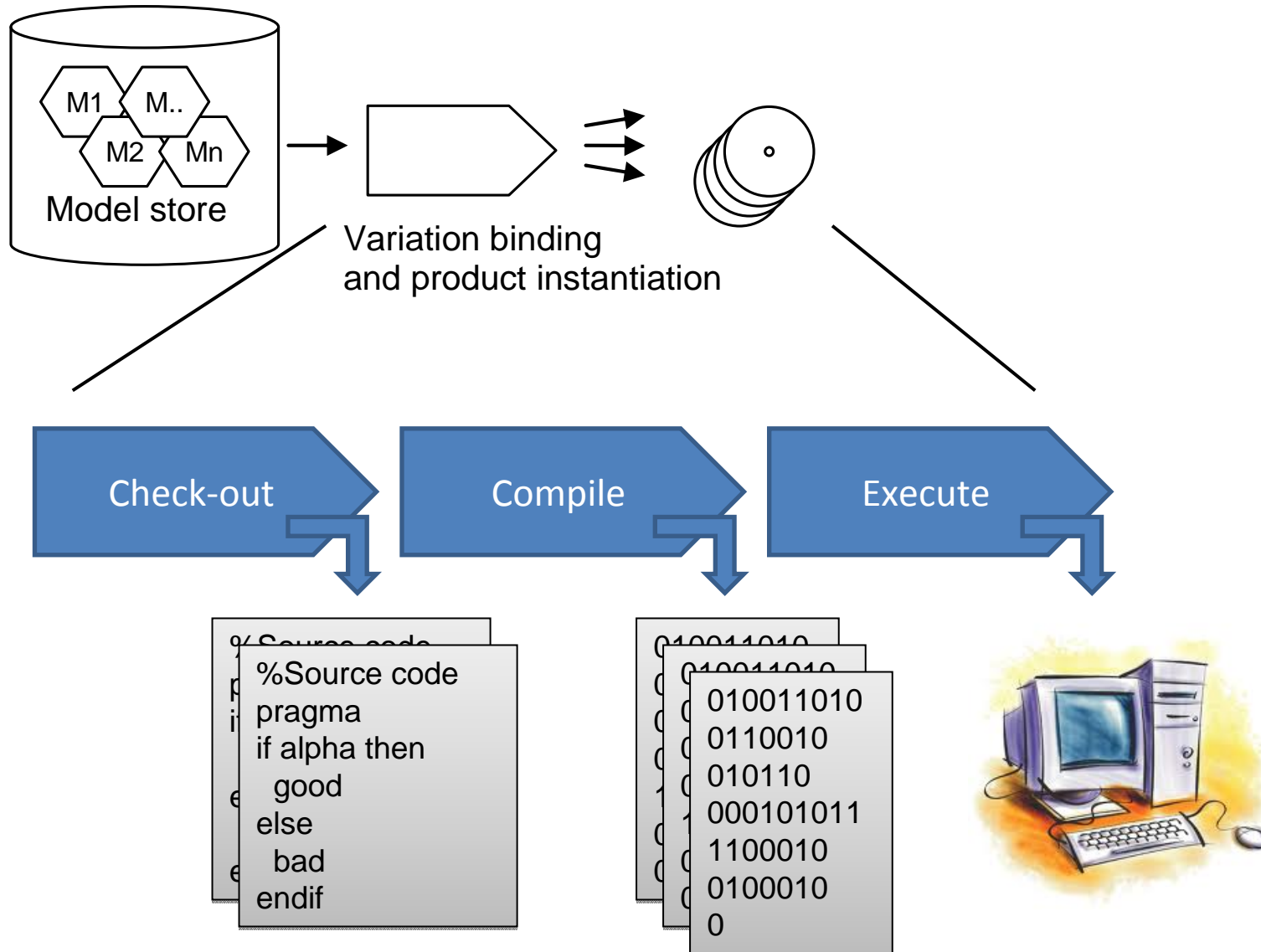
- Basic Mean: Assets with variations.
- Variation methods:
  - Model variants. A “variant master” describe their properties
  - Configurable models. Switches to instantiate desired behavior

# Collaboration aspects of the SPL infrastructure



- ▶ Collaboration at model supplier interaction
  - Architectural requirements and standards collaboration/agreement
  - Transparency of development status
- ▶ Collaboration at customer interaction
  - Early validation of product functions and properties

# Binding time alternatives

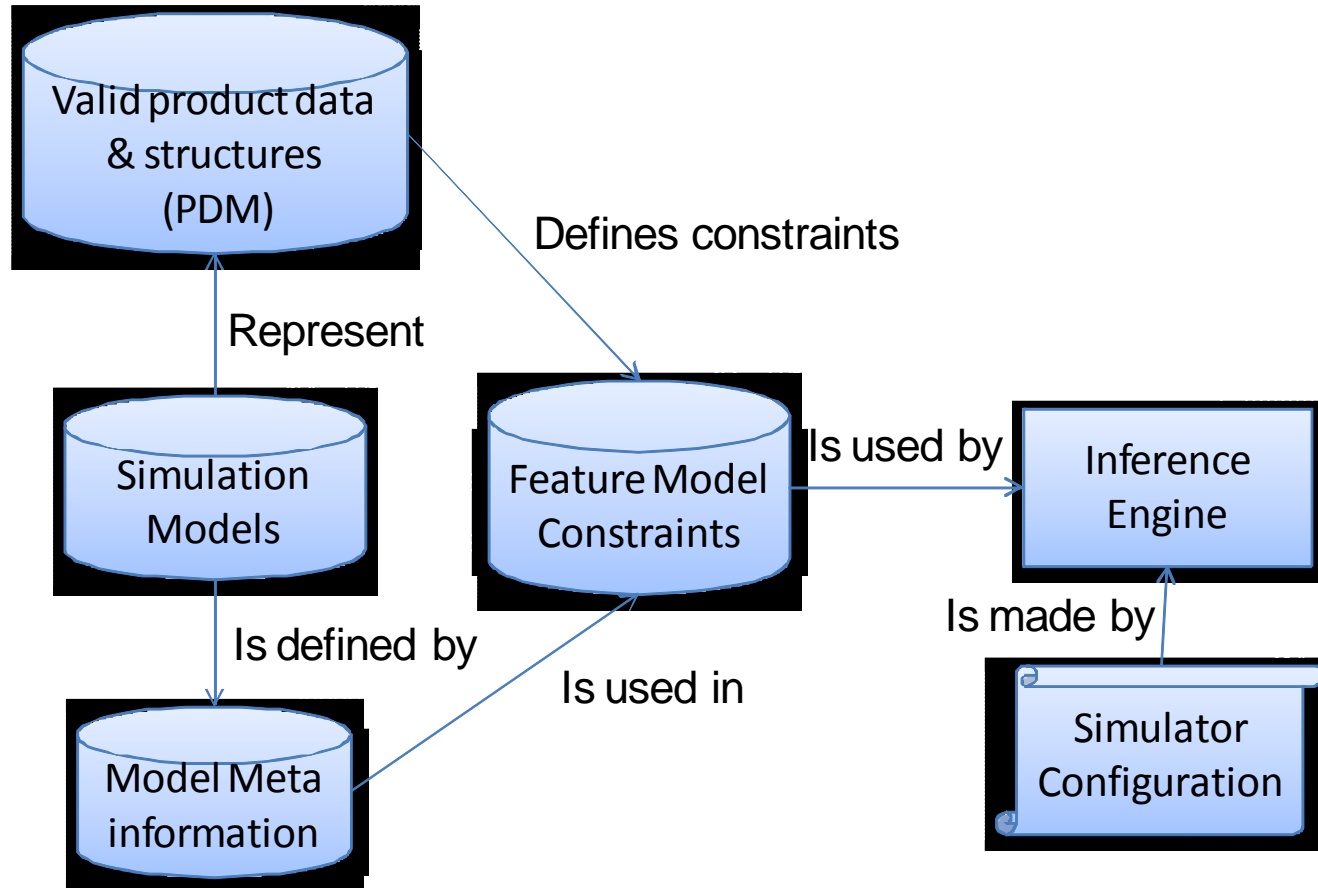


# Binding Time overview

Binding at Aspect	Check-out-time	Compile-time	Run-time
<b>Creates</b>	Different source code variants	Different object code variants	Different instances
<b>Used for</b>	Reliable configuration	Implementation oriented configuration	Fast reconfiguration
<b>Example</b>	When security / IRP aspects is important	Target / platform variation	Reconfiguration at end-user site

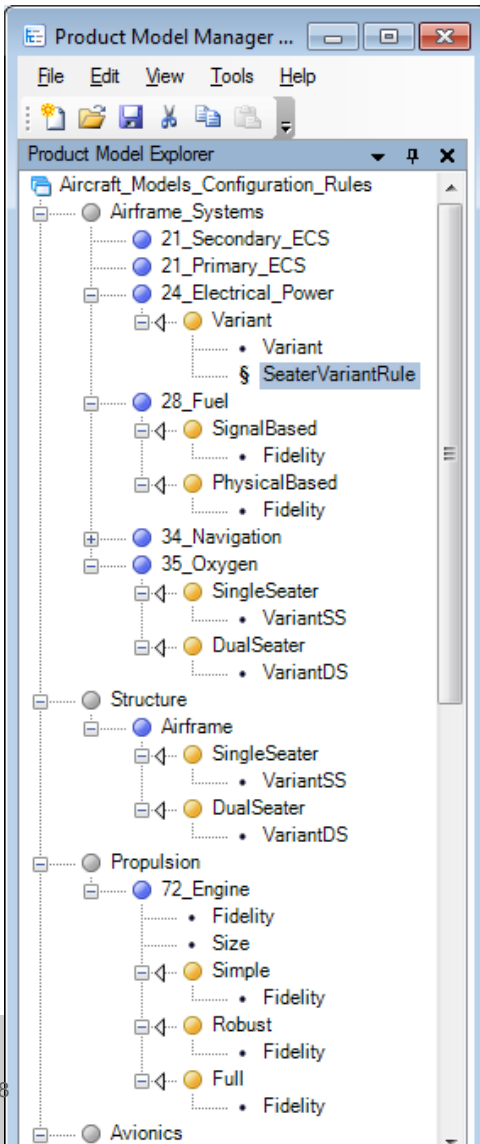


# Configuration & Customization System Architecture



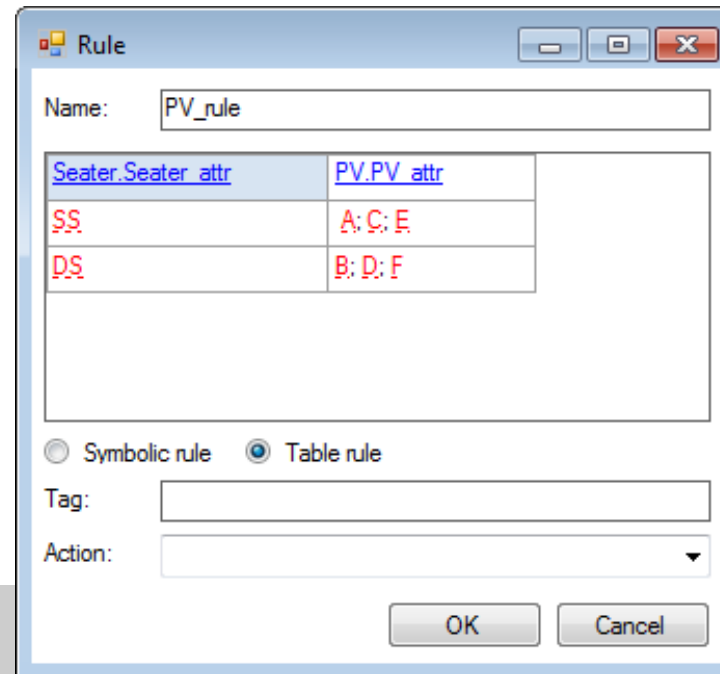
# Product Variant Master – an analyze

- the first step towards a configurator prototype



## Object oriented model of model variants

- Objects to represent model variants
- Attributes to repr. variation points
- Rules for valid / not valid combinations



# Prototype implementation

## – Tacton Configurator Studio (COTS tool)

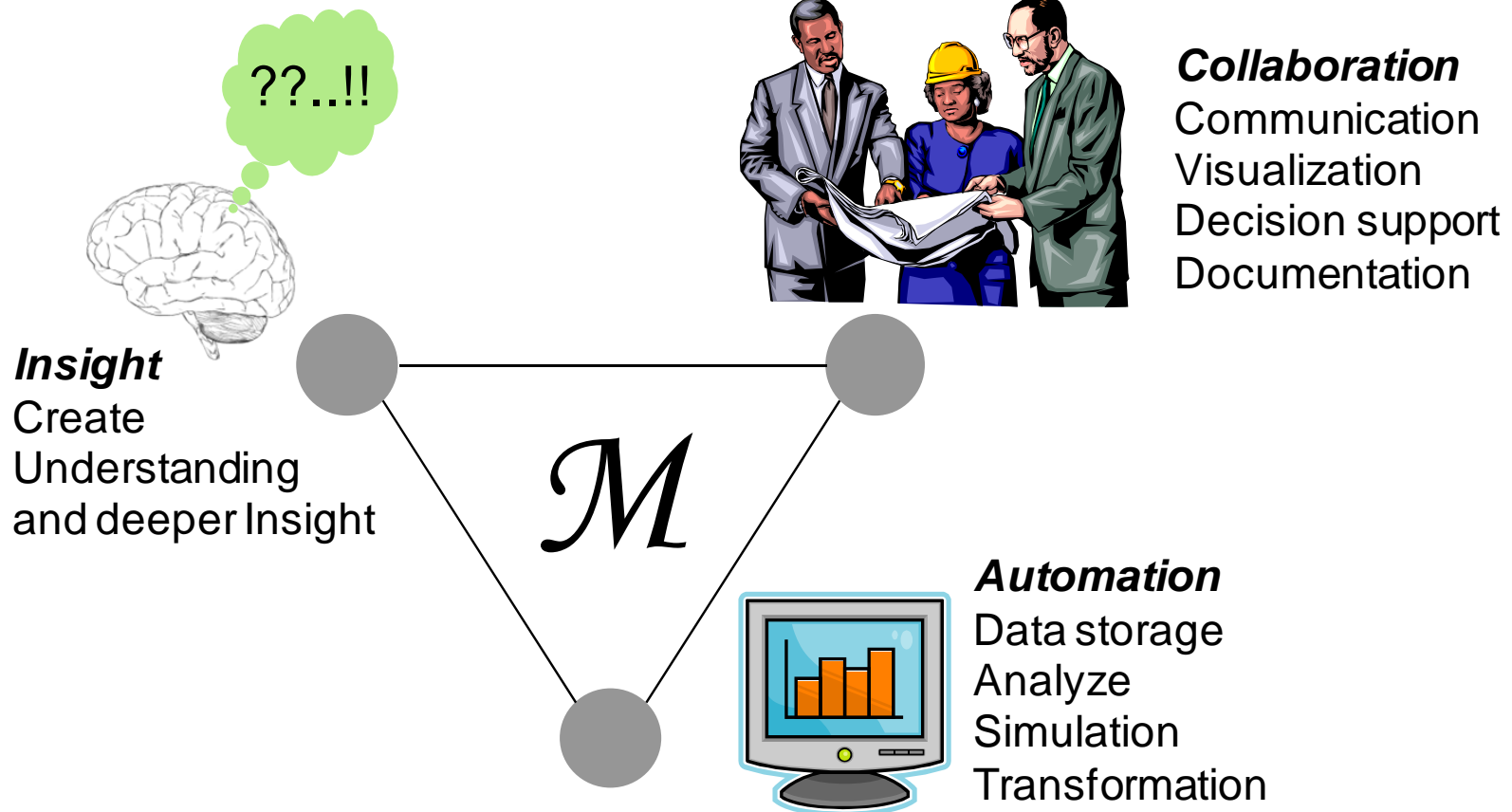
The screenshot displays the Tacton Configurator Studio interface. On the left, the 'Component Classes' tree lists various aircraft components such as ECS, Fuel, Hydraulic, GPS, and Engine models and connectors. The 'Configuration View' on the right shows a hierarchical 'Subpart Structure' for an 'AircraftSimulationRootPart', including 'Aircraft\_Variant', 'Models\_and\_Connectors', 'Aircraft\_Systems', 'Avionics', and 'Aircraft\_Software'. Below these views, the 'Properties' and 'Features' panels are visible. The 'Properties' panel shows fields for Name, Description, and Name. The 'Features' panel displays a table of features with their respective data types and parent classes.

Name	Type	Parent
isSecret	boolean	
ExecRate	float	
ExecTime	float	
AircraftType	01_aircraft_type_class	
Endianess	endianness_domain	
Model_2_conn_interface	model_2_conn_interfa...	
FidelityLevel	fidelity_level_domain	
CostPerHour	float	

The screenshot shows the 'Optimal Aircraft Simulation Application' configuration window. It features a title bar with 'intime' and a close button. The main content area is titled 'Aircraft Configuration' and contains several sections with dropdown menus and checkboxes:

- Aircraft:** Aircraft Variant: User Specified
- 21 ESC:** Model: ECS\_03 (General); Connector: ECS\_Connector\_01 (CD)
- 28 Fuel:** Model: Fuel\_HW (rig); Connector: Fuel\_Connector\_01 (CD)
- 29 Hydraulic:** Connector: Hyd\_Connector\_01 (CD)
- 34 GPS:** (No configuration options shown)

# THE UTILIZATION POINTS OF MODELS



# Conclusions and further work

## Conclusions

- Constraints input from PDM for integration of configuration data between PDM and the simulation environment
- Model Interface Compatibility is crucial. The emerging FMI (Functional Mock-up Interface) standard is promising for improved collaboration

## Further Work

- Develop a robust meta-model for model variability, configuration/customization
- Connection to emerging standards; PLMXML, FMI & SysML
  - Use XML, XSD & XSLT for data storage, exchange, presentation and mapping
- Validate prototype configurator implementation in the application project

# Thank You!