PTC® PRODUCT & SERVICE ADVANTAGE®

Model-Based Product Line Engineering

Enabling Product Families with Variants

Matthew Hause & Jim Hummell
Technical Fellow & Principal Engineer, MBSE

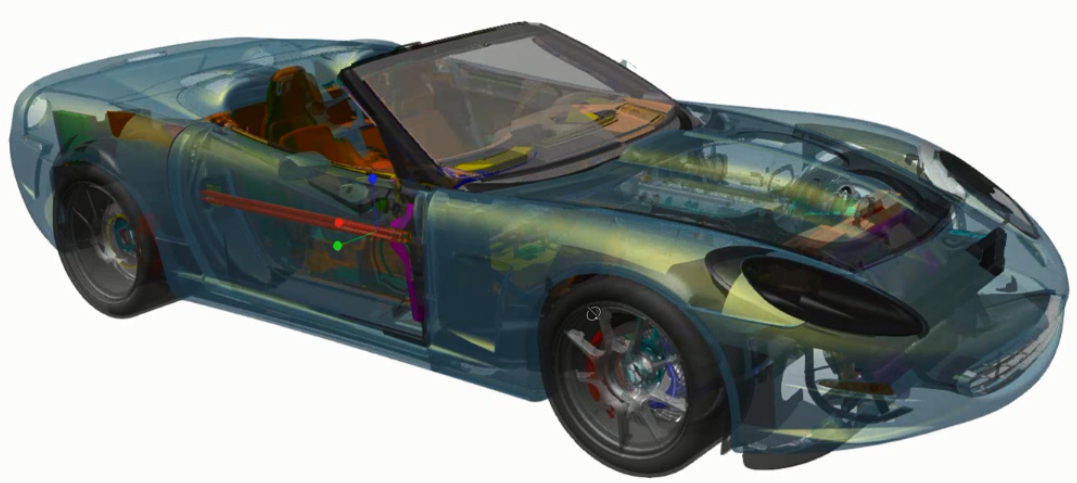
July 2015

Systems Engineering Challenges

PTC®

Smart connected systems & products

- Growing complexity & functionality of systems & software
  - Larger share of a products cost & capability is software
  - System & sub-system Integration
  - Customer, certification, regulation & standards compliance needs
- Larger, more distributed & distinct discipline teams
  - Communication language barriers & collaboration
  - Implementing common, architected Goals
- Increasing time pressures
  - Shorter development cycles
  - Delivering on schedule
- Quality assurance
  - Risk of building the wrong system
  - Increased costs of later stage errors
- Cost reduction demands
  - Total development cost
  - Risks & costs of delays or cancellation



The Solution … Model-Based Systems Engineering

PTC®

Design before you build

- Standard based graphical modelling
  - Common language
    - Improves understanding
    - Facilitates collaboration
    - Achieves stakeholder buy in
  - Problem abstraction, to see the ‘wood from the trees’
- Systems engineering process automation
  - Tools enable a more efficient systems engineering process
  - Tangible designs to review, finding problems earlier
  - Traceability from requirements through models to system
  - Enables Rapid Prototyping, Simulation & Trade Studies
- Reduces the total cost of systems engineering
  - Reduce learning curve & cost with an industry standard language
  - Capture system design IP to reduce risks & retain value
  - Optimized allocation to mechanical, electrical & software engineering
  - Design & build the right systems, right

System Product Line Engineering (PLE) Challenges

Product line explosion

- Increasing number of product families
- Increasing number of products in families
- Understanding product similarity
- Maximizing reuse
- Understanding product variations
- Deciding between options
- Development cycle time
- Commercial product needs
  - Customize existing capabilities to suit client requirements
  - Redeploy common systems & software to the Market
  - Time from requirements to cash

The Solution … Model-Based Product Line Engineering

PTC®

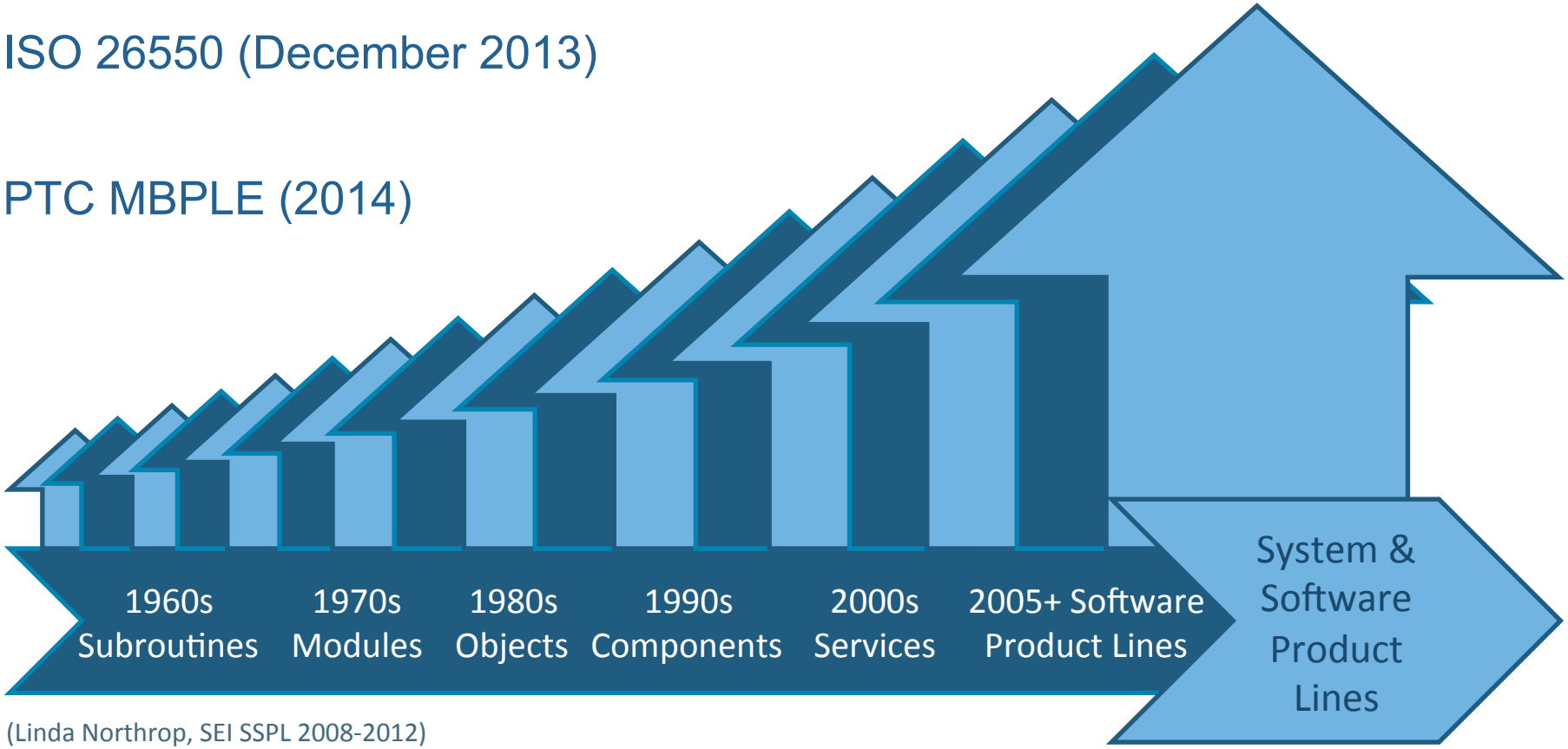
Designing a single system platform rather than as creating a multitude of products

- MBSE + Modular Design + Variation
  - Common language improves
    - Communication
    - Collaboration
    - Stakeholder buy in
  - Architected modular design & reuse
  - System product lines designed up front
- Maximum commonality & minimal variation
  - Less duplicated effort with optimized reuse
  - Parallel working through ‘design by contract’
  - More commonality between designs and implementations
  - Managed product line complexity

The History of Reuse

Model-based Systems & Software Engineering (2006) +
System & Software Product Line Engineering (2001-2008)

- ISO 26550 (December 2013)
- PTC MBPLE (2014)

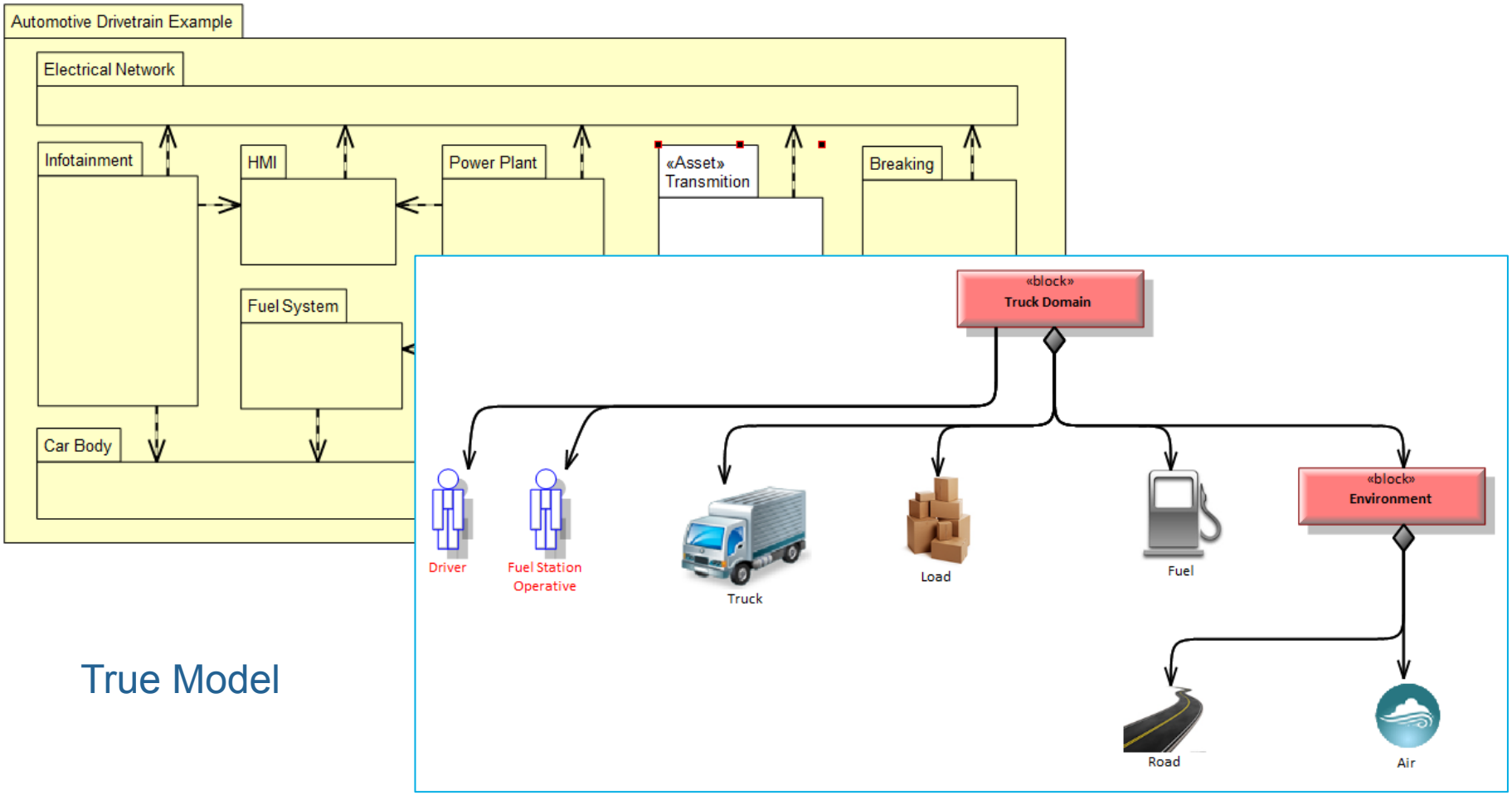


(Linda Northrop, SEI SSPL 2008-2012)

Model-Based Systems Engineering

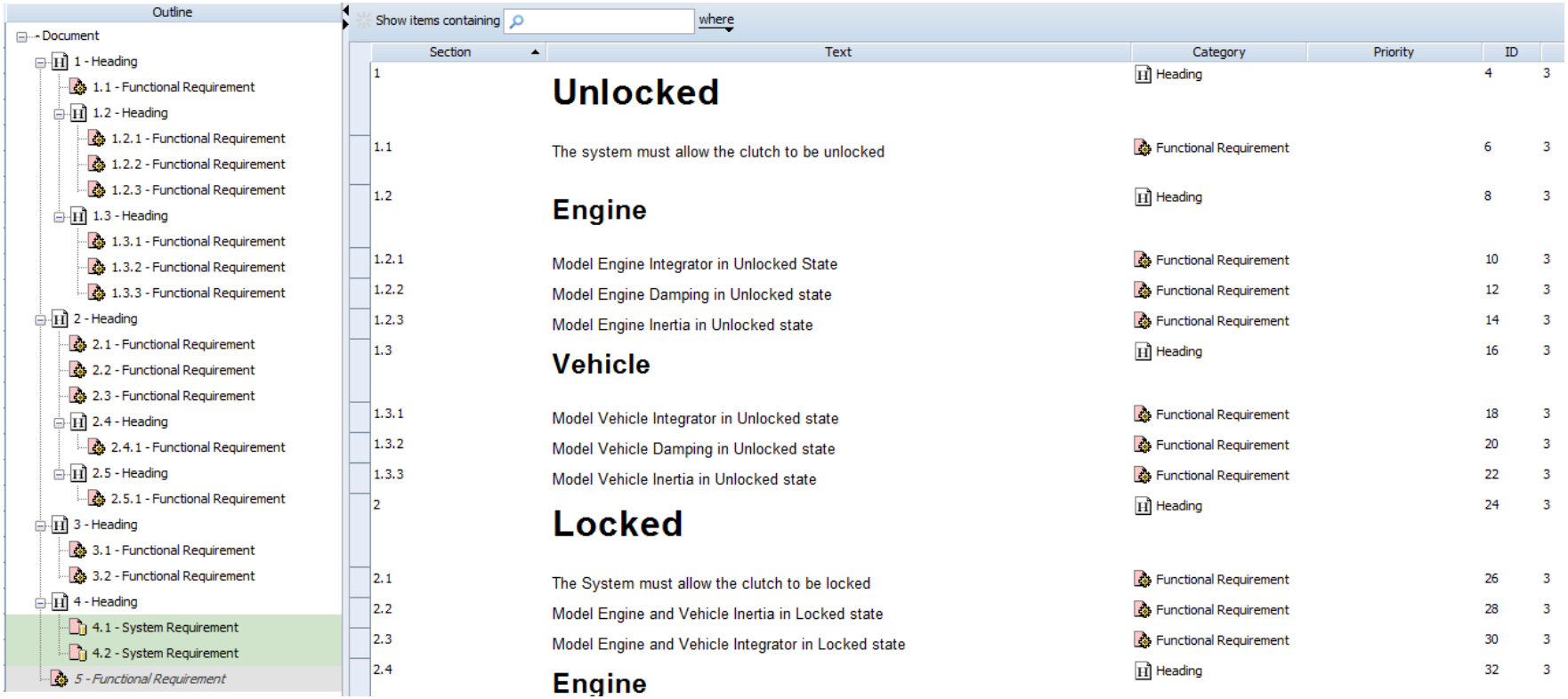
System Domain View - SysML

- Scope & Stakeholder Identification for Requirements Elicitation



True Model

Stakeholder & Environmental Requirements Elicitation & Analysis

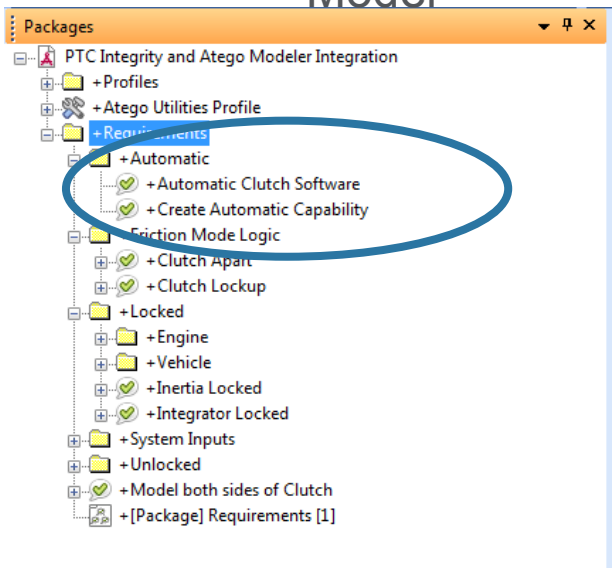


Synchronizing Requirements with PTC Integrity

Round tripping with PTC Integrity Modeler – Integration for Lifecycle Manager

- Updates can be made from either Modeler or PTC Integrity Lifecycle Manager
- Bi-directional synchronization to update both the model and requirement document in tandem

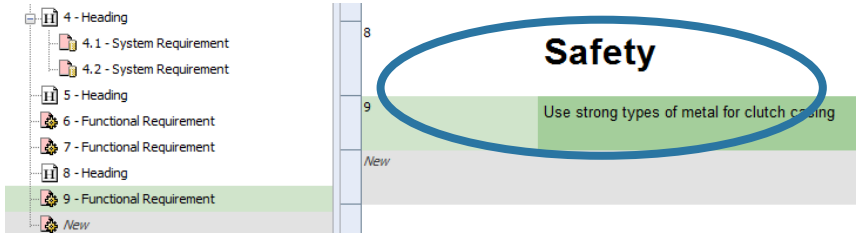
Model



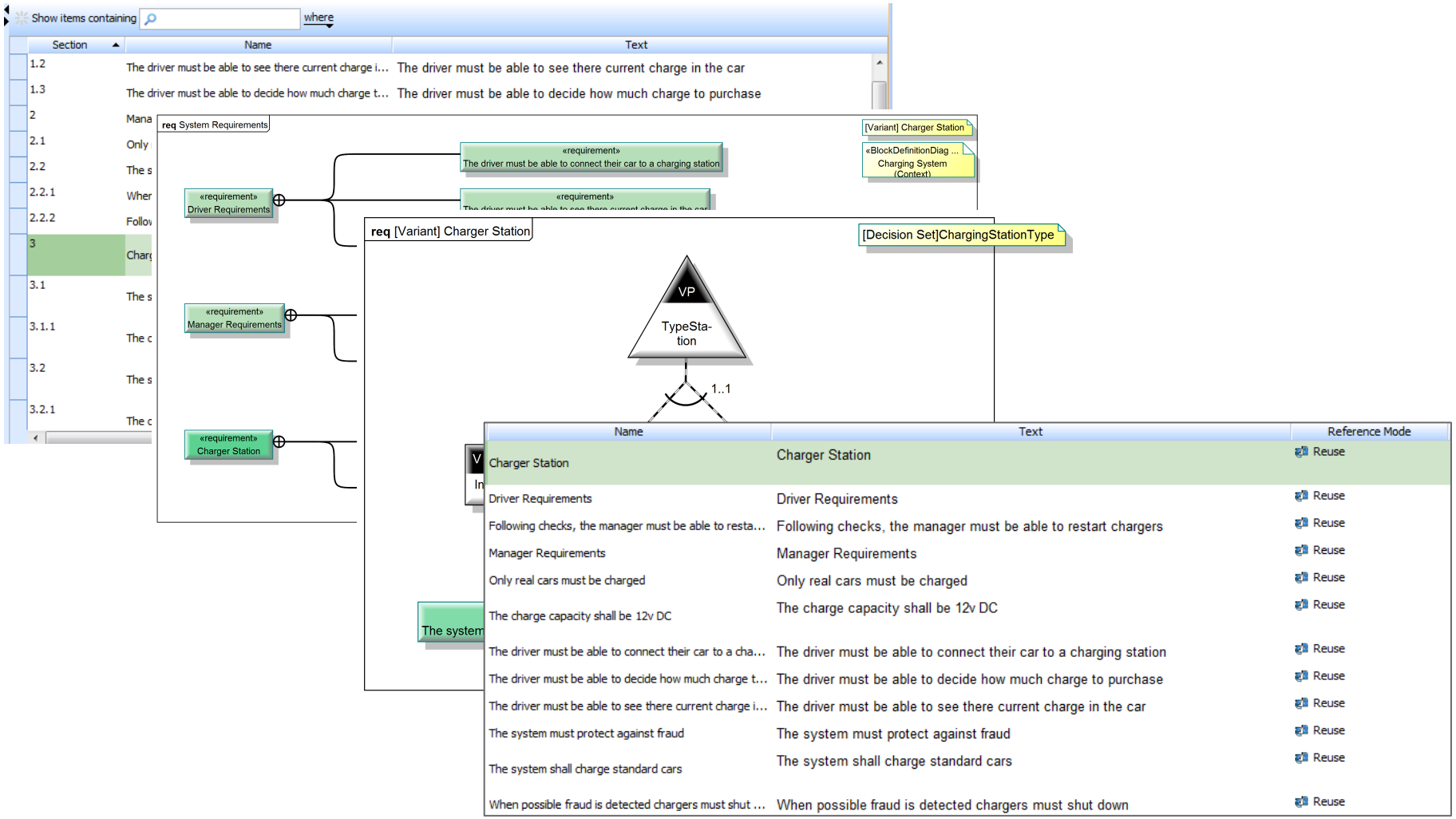
Requirements & Traces

Model Element

Requirements Management



Product Line Branching of Requirements

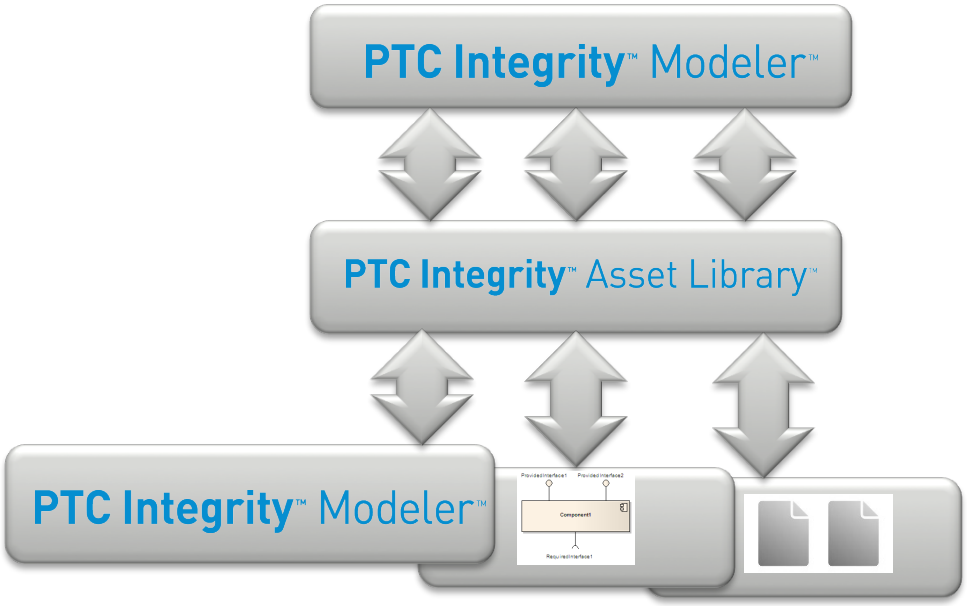


Asset Based Modular Design

Asset-Based Modular Design

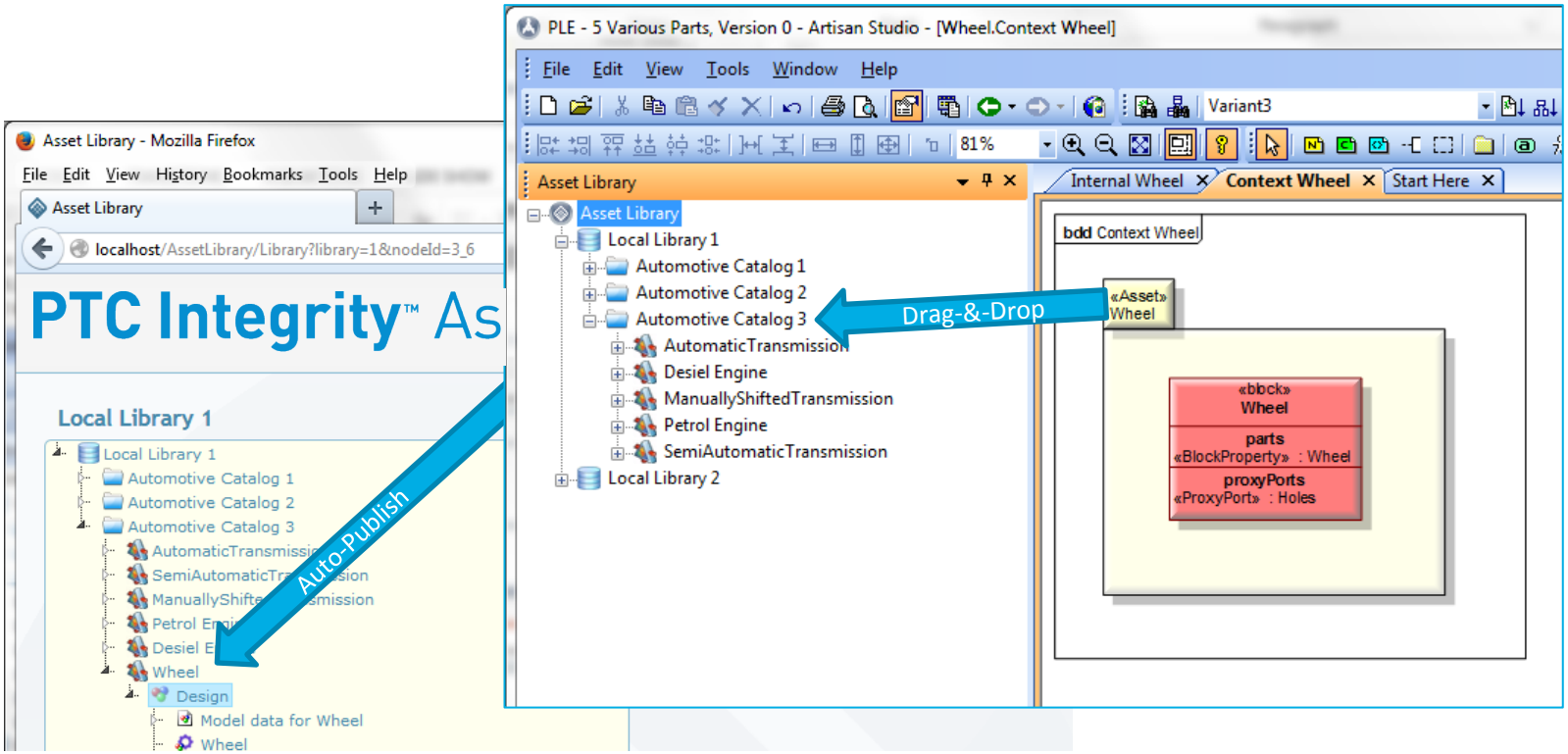
- Design the same way you Build
  - Construct Systems of Sub-Systems (SoS)
  - Use Services to build your Application (SOA)
  - Plug Components together (CBD)

- Modular Design
  - Top-Down, Architected
    - Specification (& Requirements) Driven
    - Parallel Working
    - Separation of Concerns
  - Bottom-Up, Asset Mining
    - Un-modeled Assets
    - Other Modeling Tools
    - Legacy Integration
    - Published Interfaces (e.g. IDL)



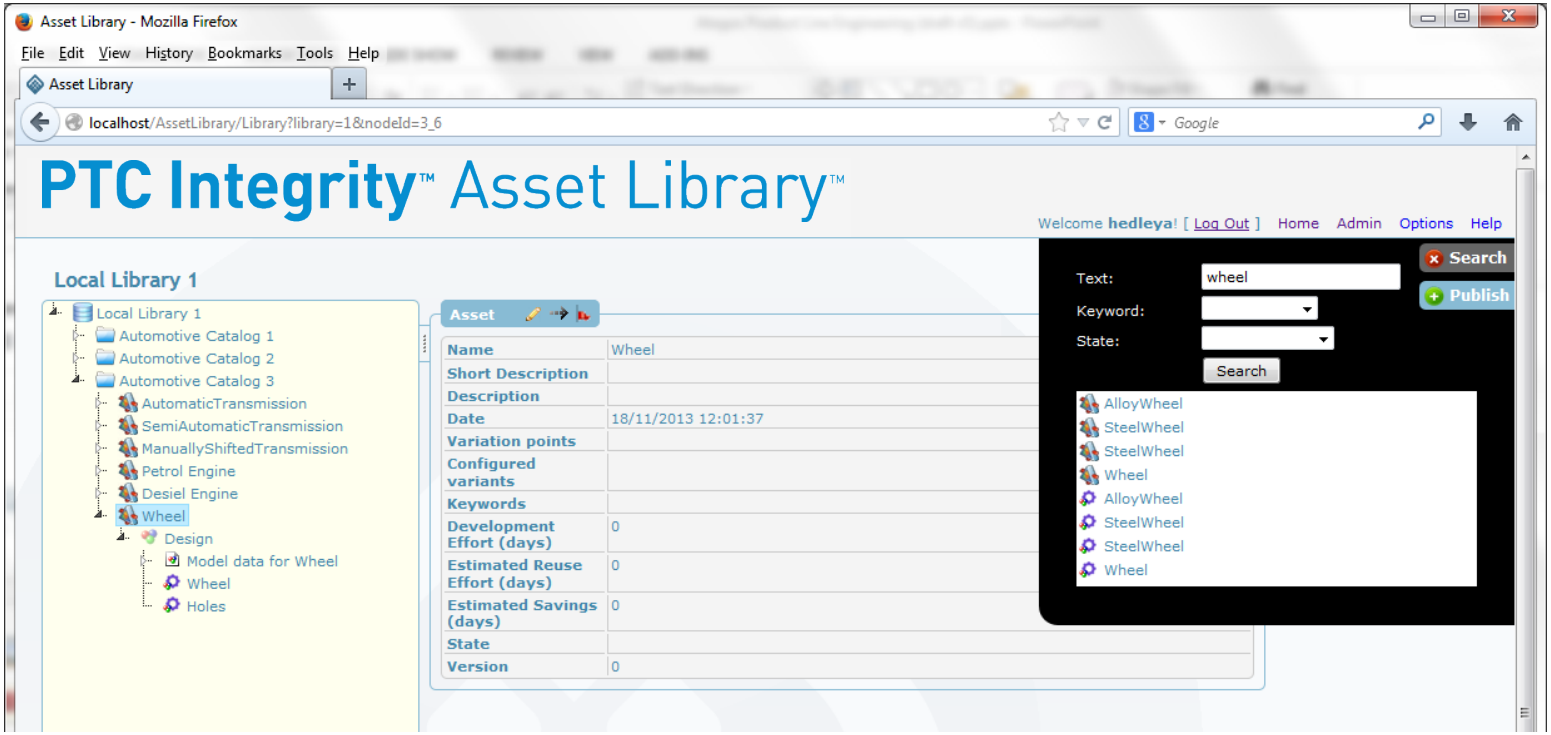
Asset-Based Modular Design

- Publish from Sub-system model into PTC Integrity Asset Library
  - Auto-creates Trace Links



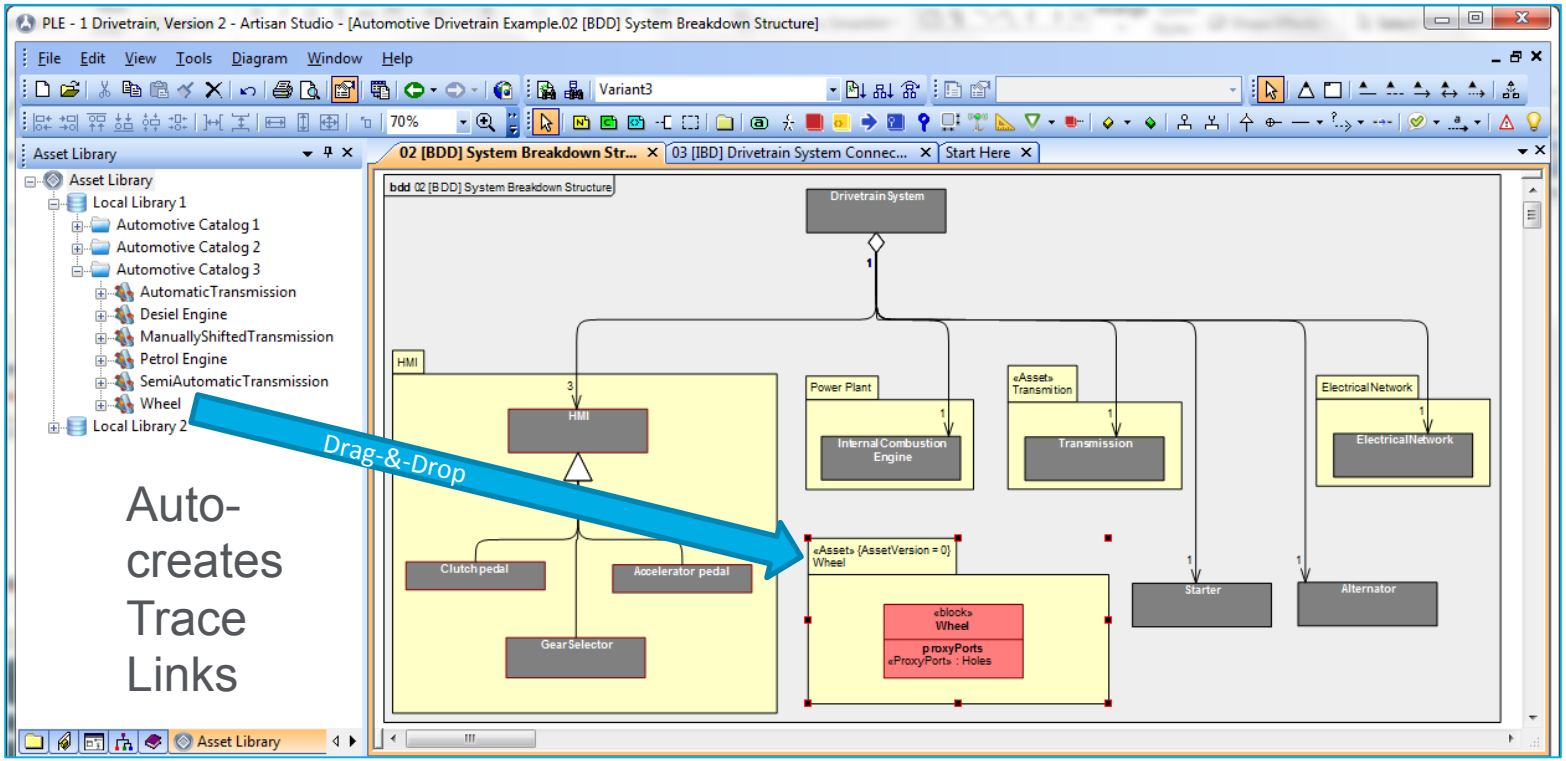
Asset-Based Modular Design

- Search PTC Integrity Asset Library for Sub-systems



Asset-Based Modular Design

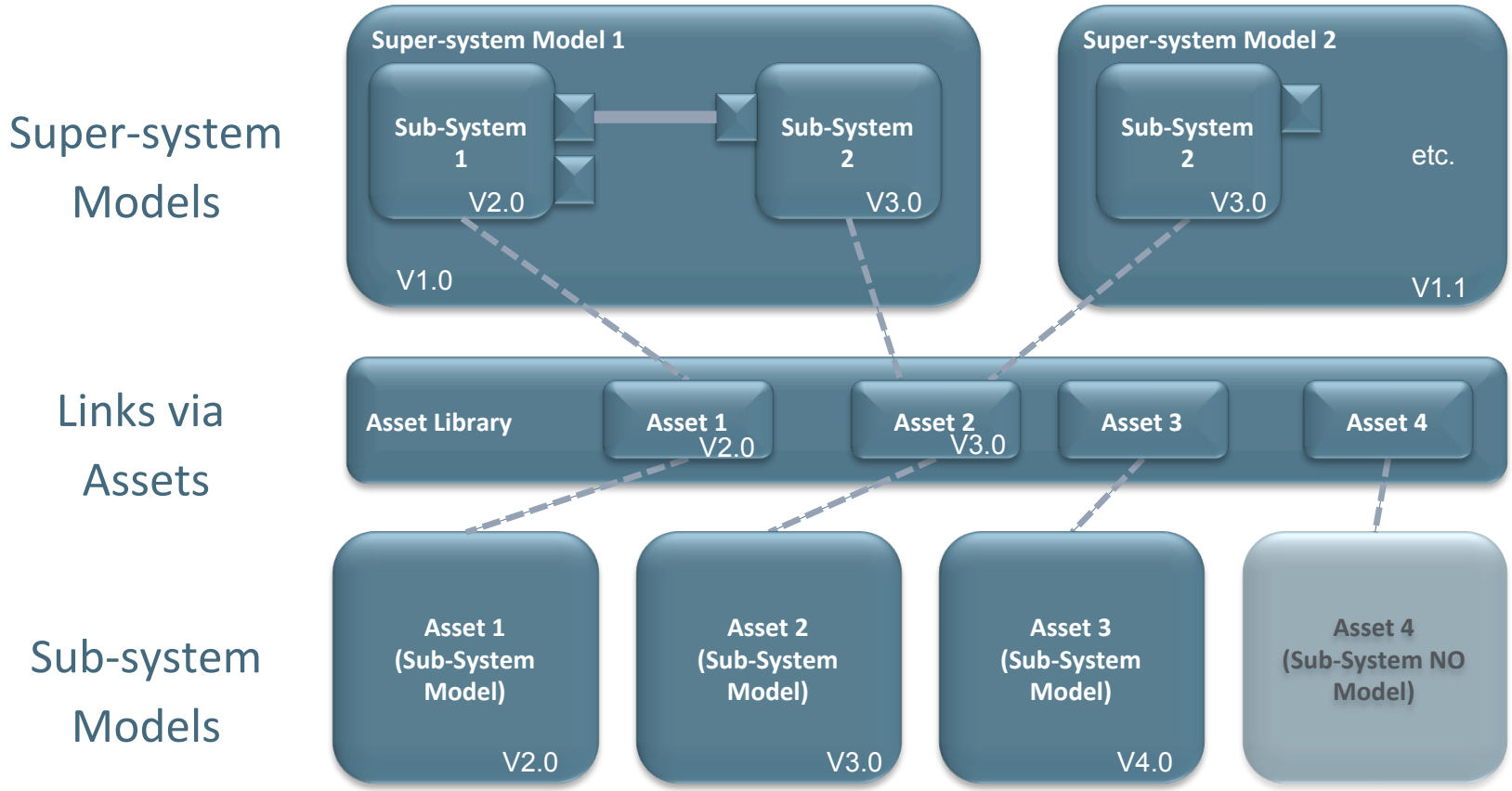
- Use Sub-system from PTC Integrity Asset Library in Super-system Model (BDD)



Asset-Based Modular Design

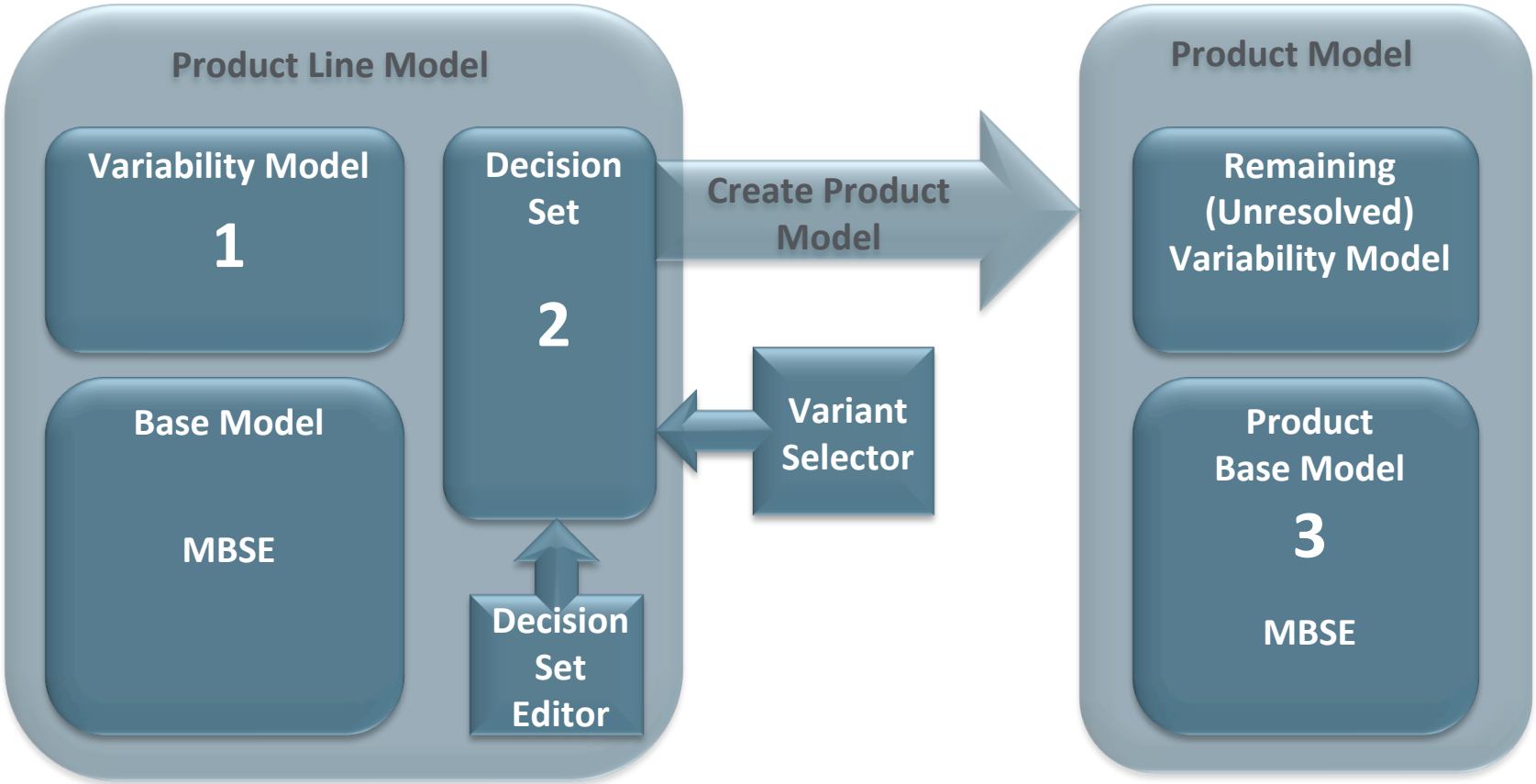
PTC®

- Super-system Model = Configuration of Versioned Sub-systems



Model-Based Product Line Engineering

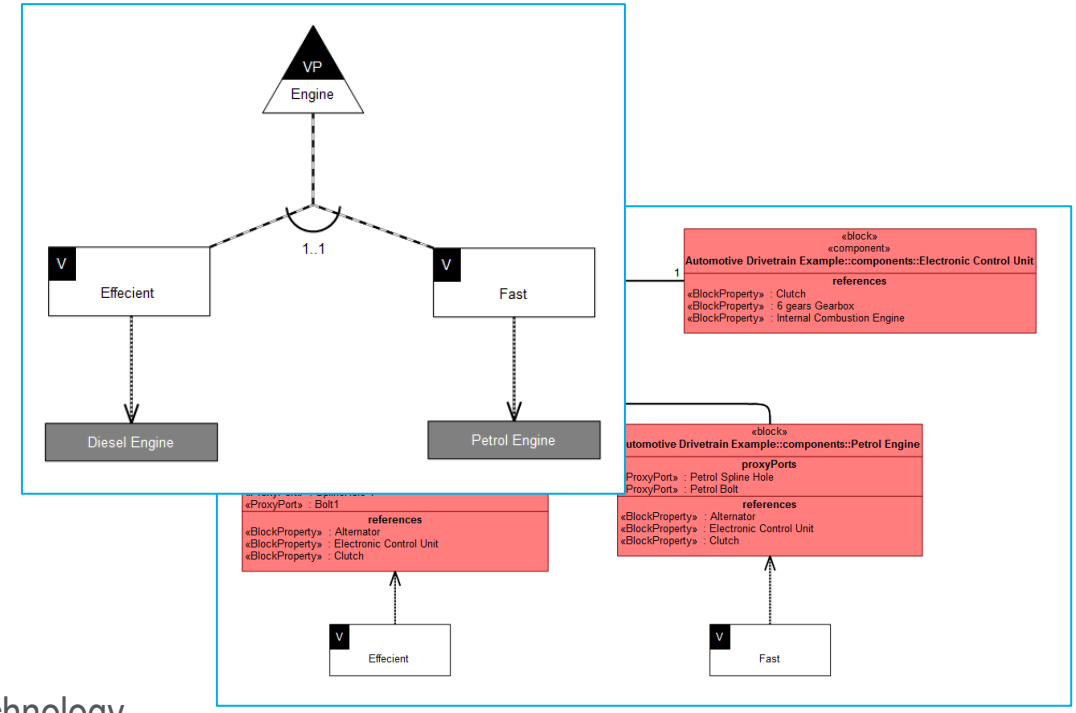
Modeling Product Lines



1 - Variant Modeling

- Variant Diagram
- Variation on all Diagrams
- Simple Notation

Variation Point
Variant
Variability Dependency
Mandatory/Optional
Requires Dependency
Excludes Dependency
Artifact Dependency
Alternate Choice

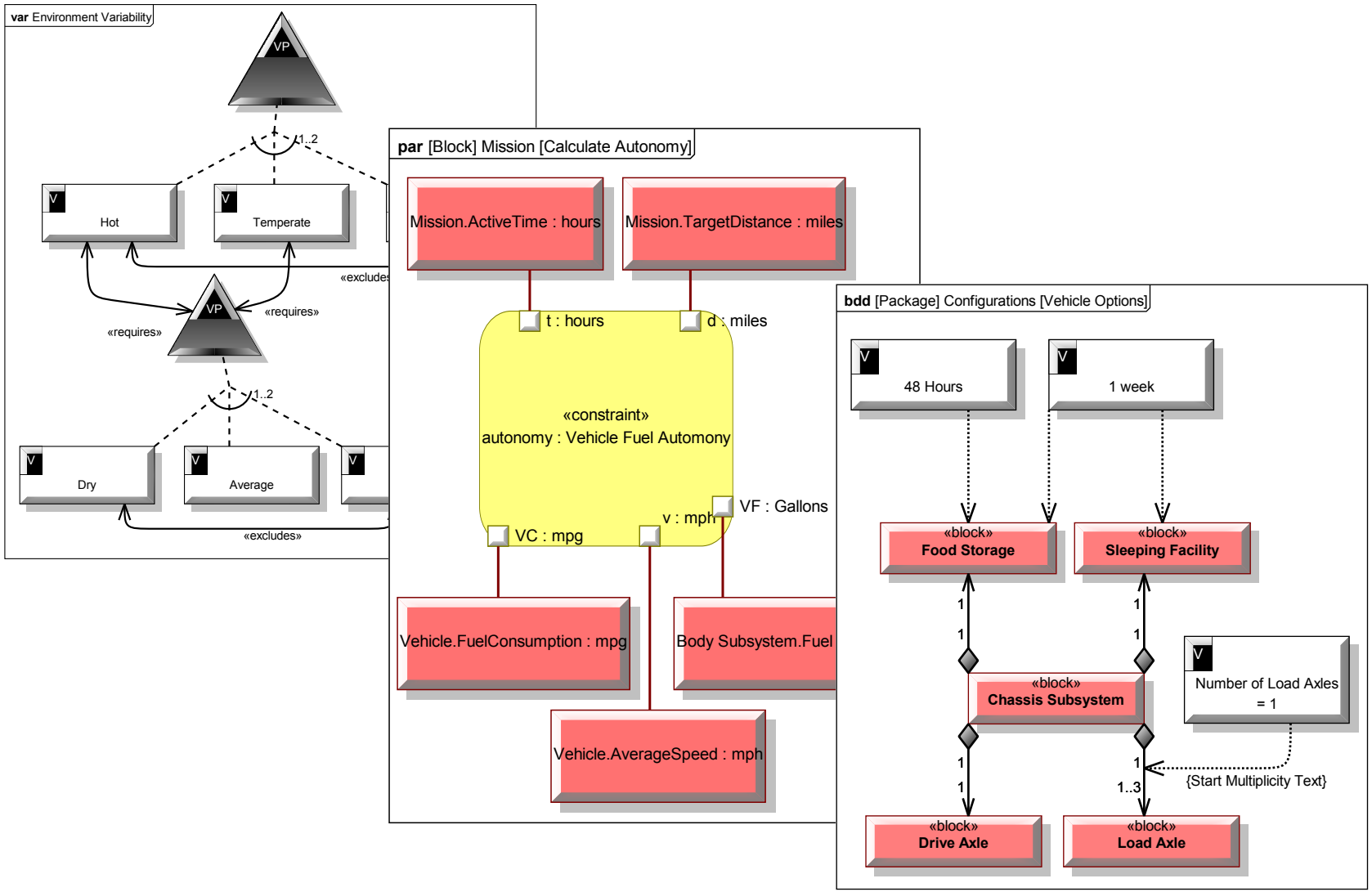


- OVM

PALUNO, The Ruhr Institute of Software Technology
Software Product Line Engineering (Pohl et al - Springer 2005)

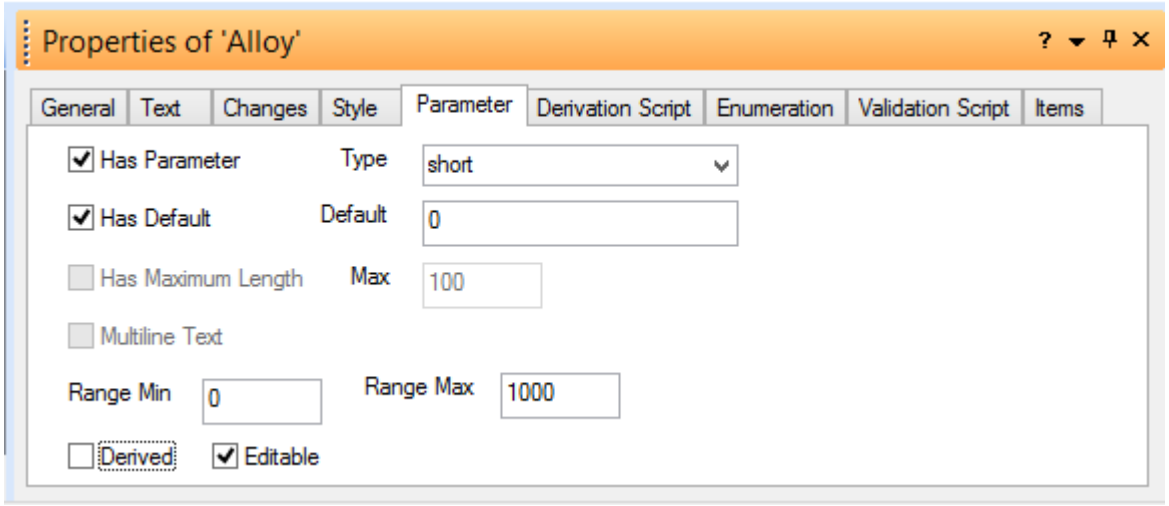
Complexity can be added

Parameters can be used that tied into values in the model itself to allow or not allow you to select a Variant

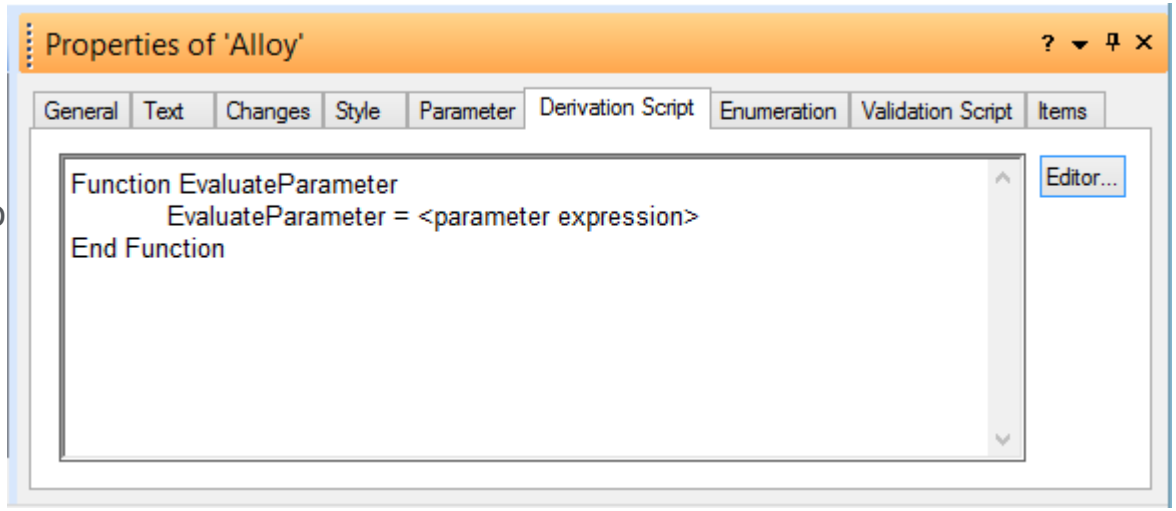


Additional Rules

- There are additional rules you can add.
- Variant can have a parameter that can have some values.
- Based on the value you will be able to select it during the Decision or not.

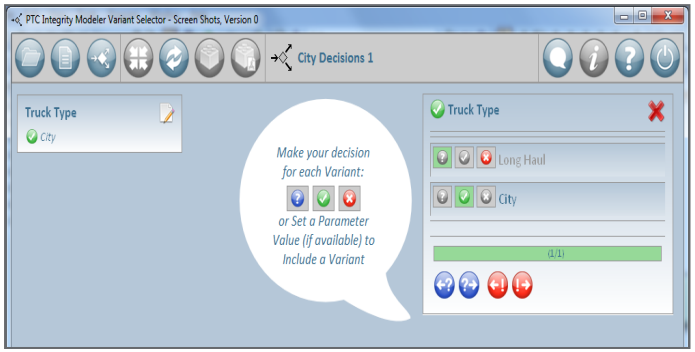


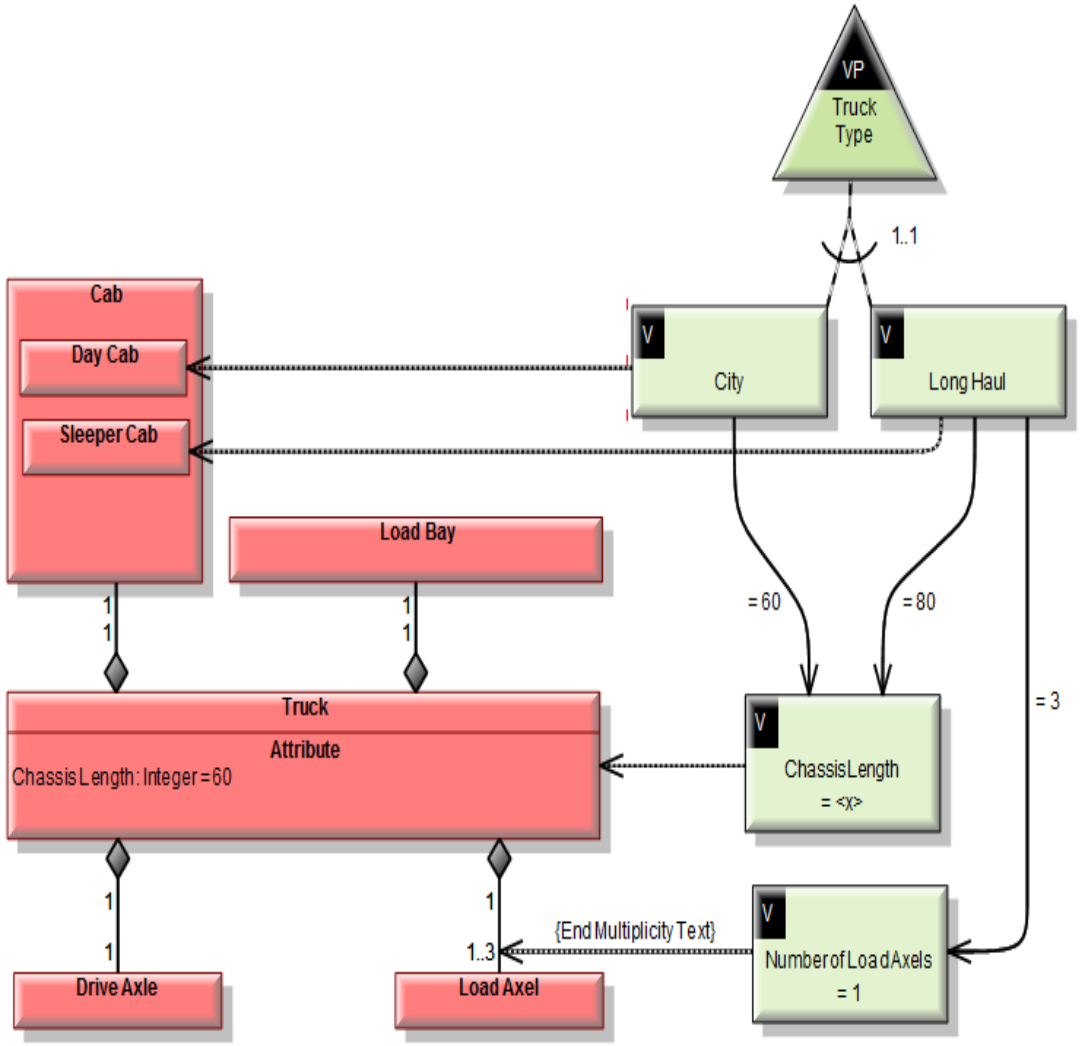
- Other additions are the use of Derivation Scripts that allow the users to have full control over when a variant is allowed or not allowed to be selected.
- Based on other modeling values and or rules the user defines.



Variable Parameters

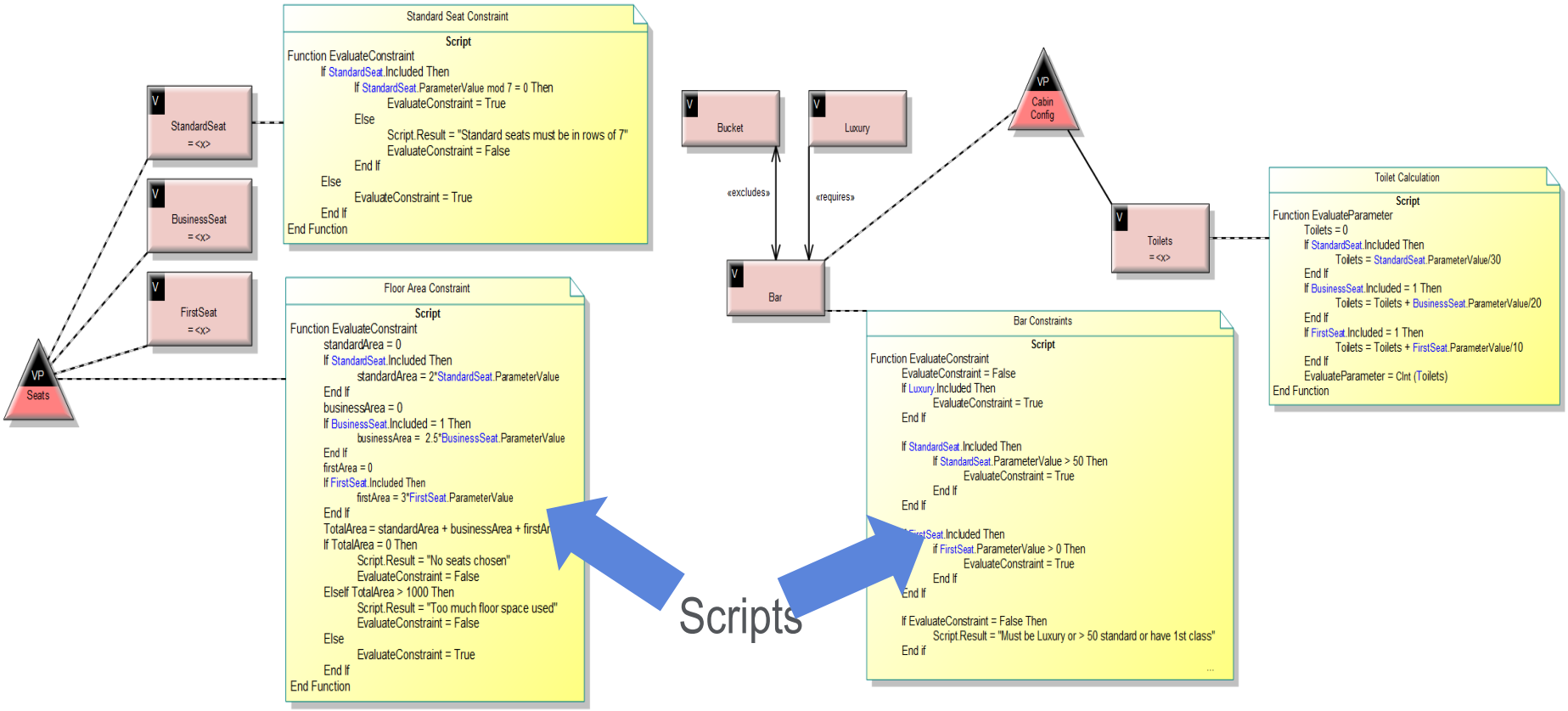
- Variable Parameter Passing





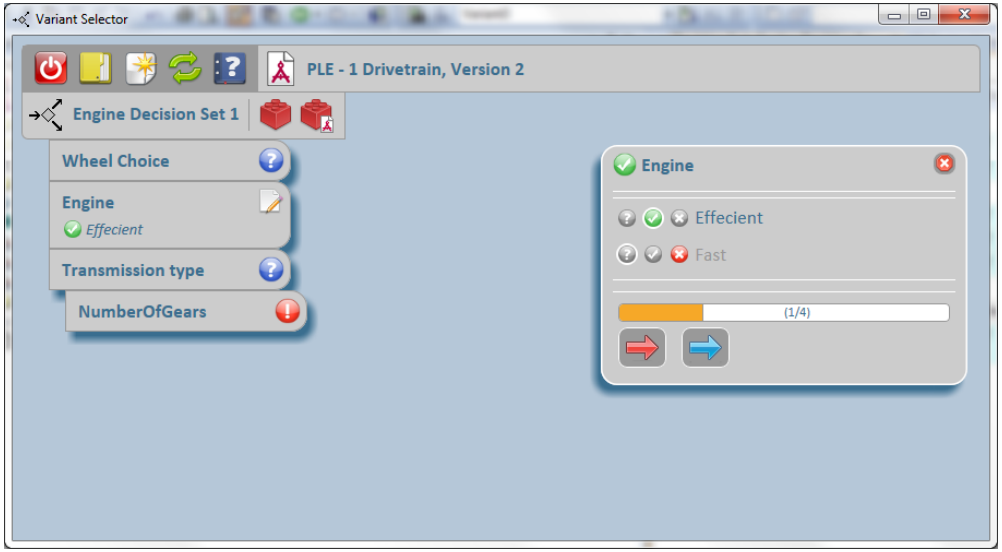
Variable Parameters

- Derivation & Validation Scripts



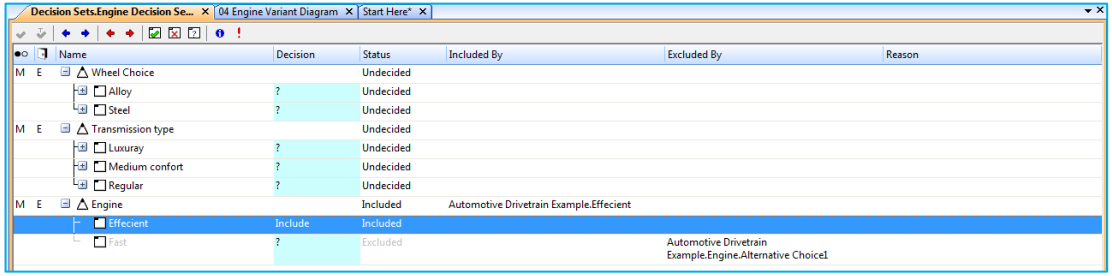
2 - Variant Selection

- Variant Selector
  - Browser User Interface
  - External Variation Points Only
  - Jump to Next Decision/Problem
  - Progress Bar



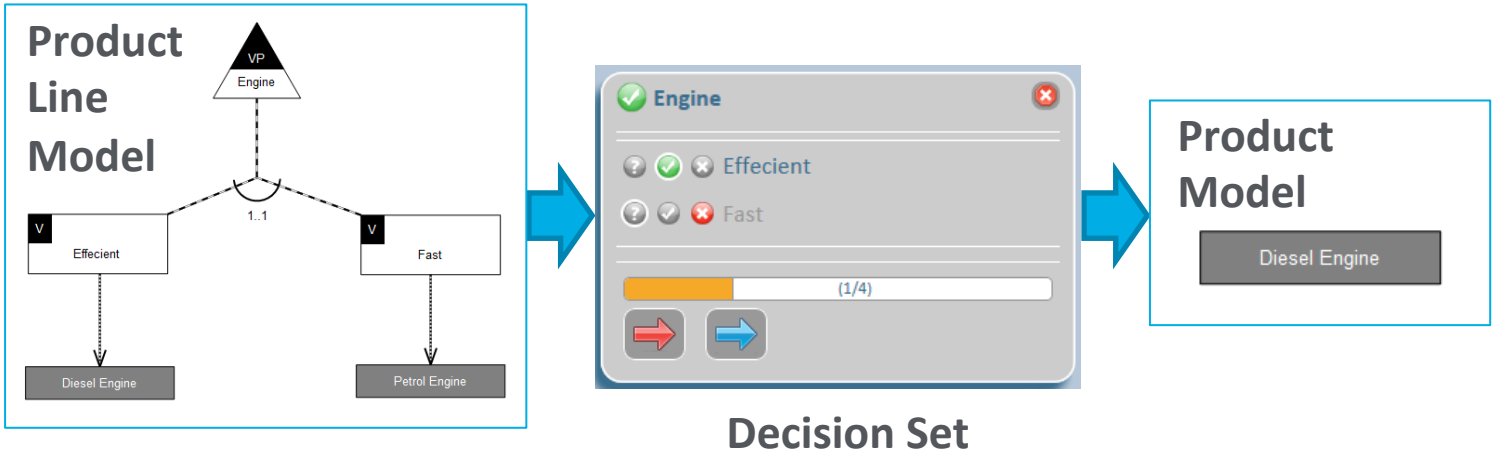
- Decision Set Editor
  - Variant Debug
  - External & Internal Variation Points
  - Jump to Next Decision/Problem

- Both Edit the Same Decision Sets



3 - Product Model Creation

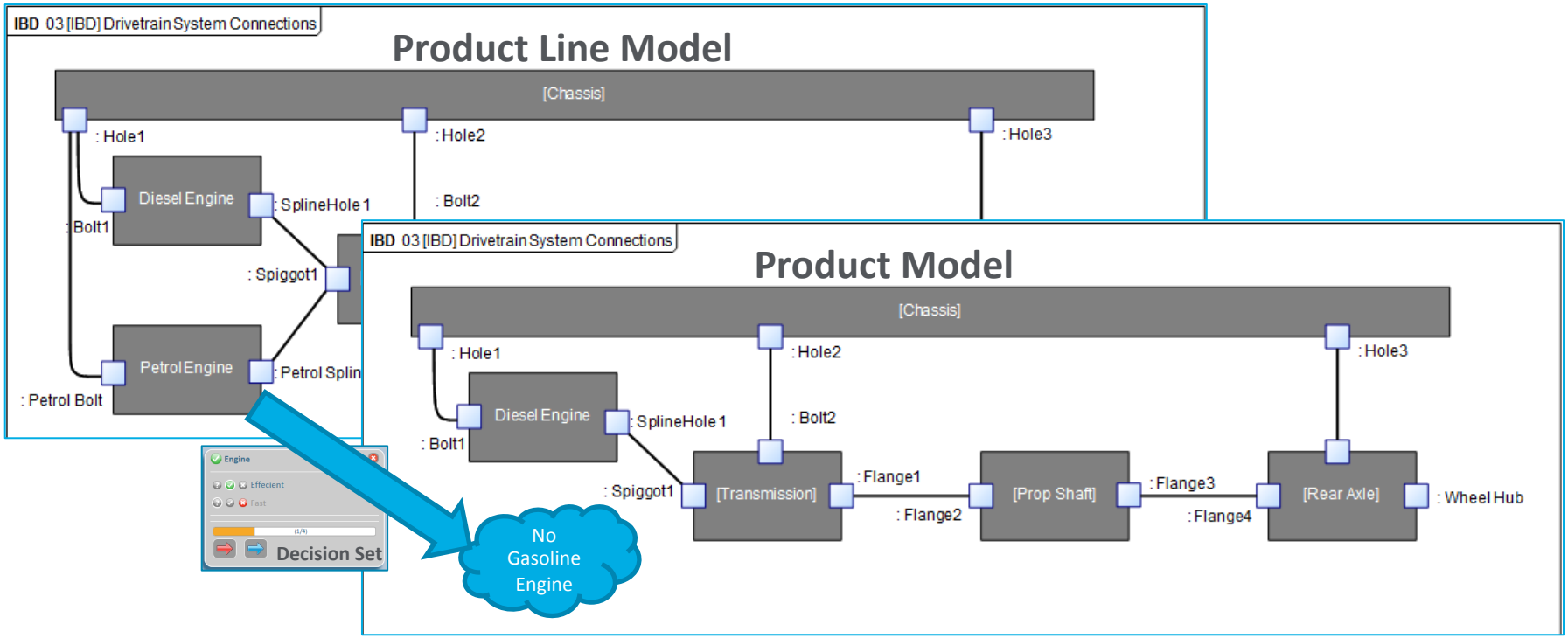
- Auto-Create Product Models
  - Applies Variability Decisions
  - Unnecessary Variation Points, Variants & Base Model Artefacts Removed



- Creates New Product Model Branch, Original Product Line Model Retained
- Product Model suitable for Trade Studies, Simulation, Documentation & Generation

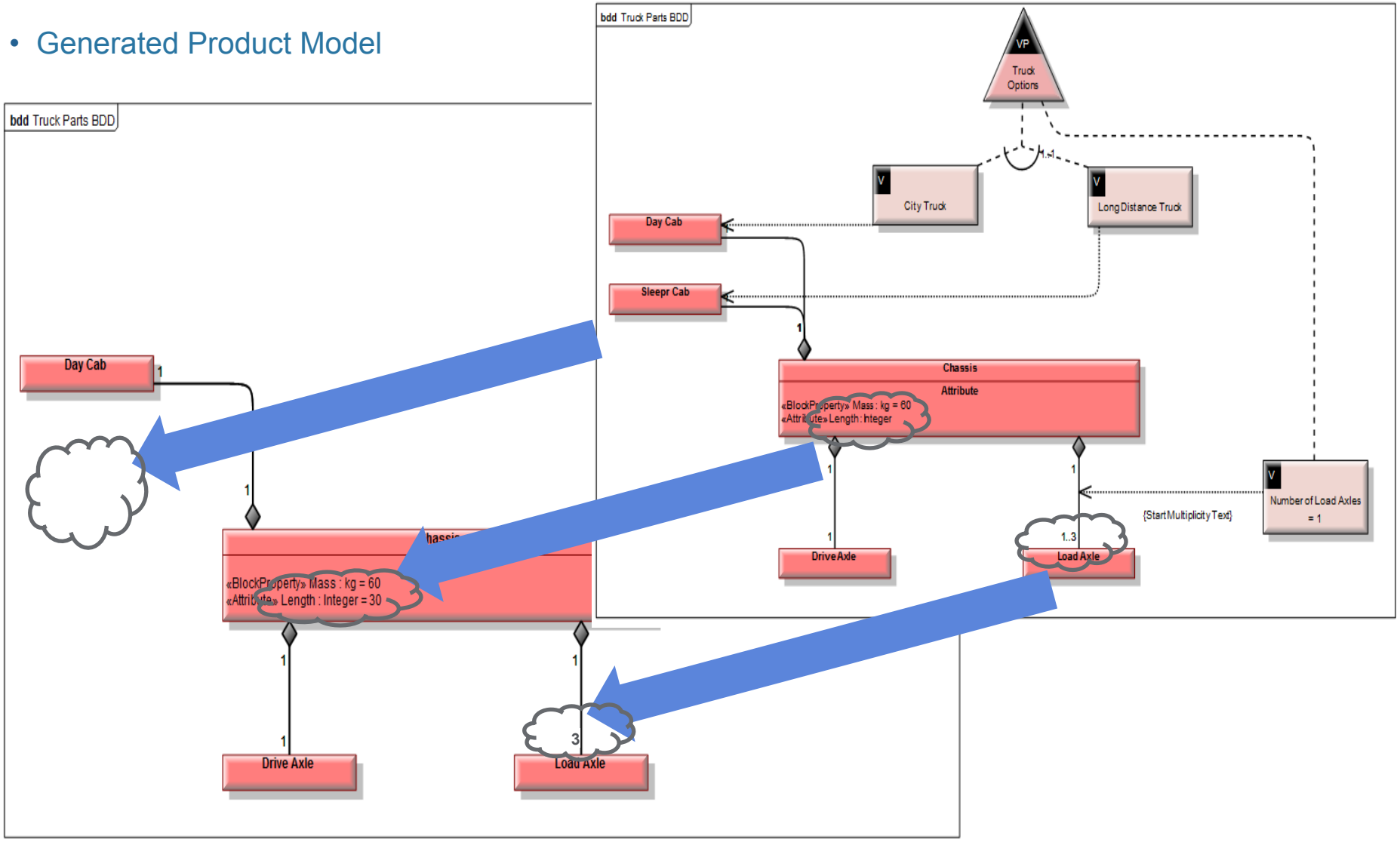
3 - Product Model Creation

- Auto-Create Product Models (IBD)



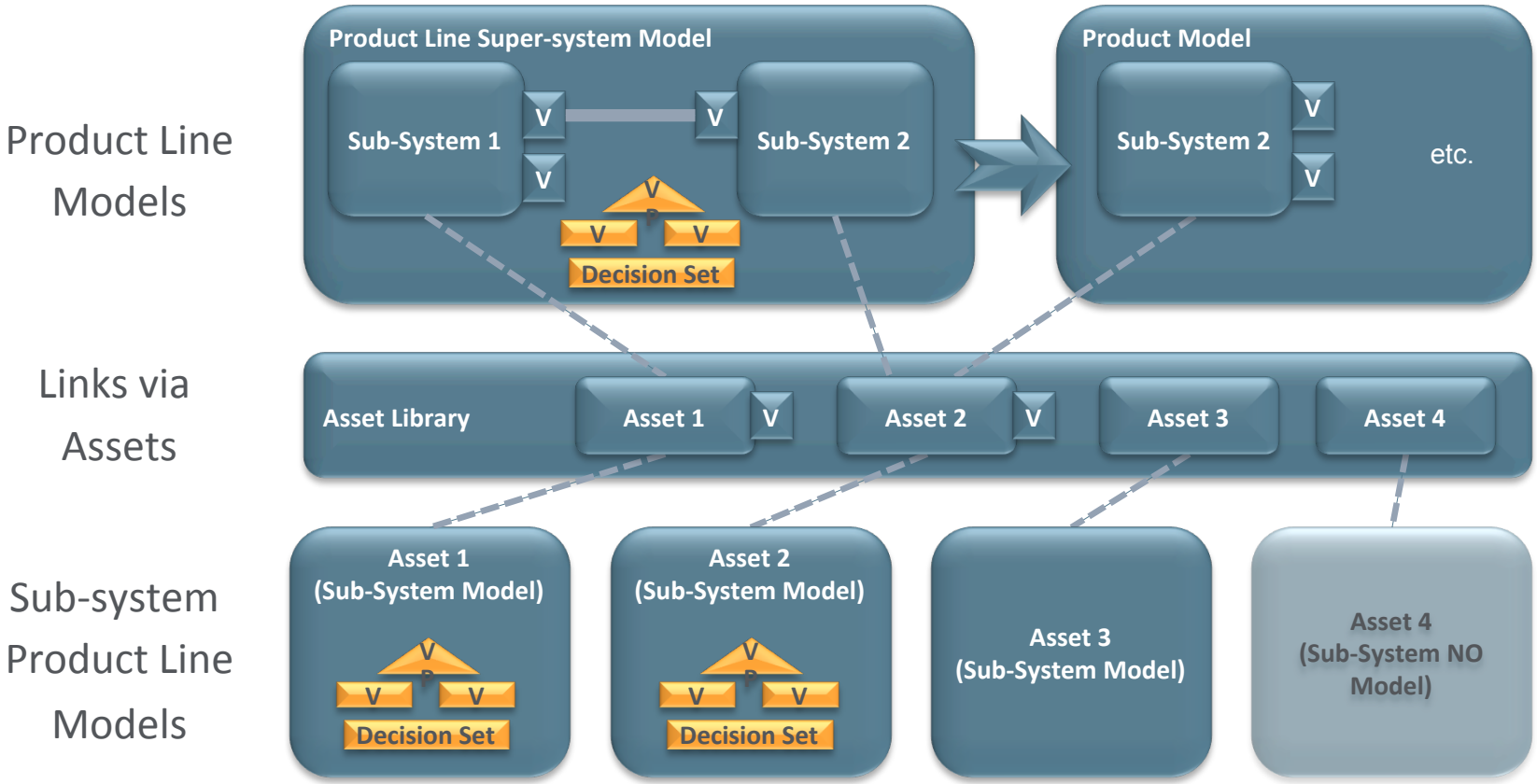
Variable Parameters

- Generated Product Model



Model-Based Product Line Engineering

- Integrated MBSE, Modular Design & Variability Modeling = Model-Based Product Line Engineering



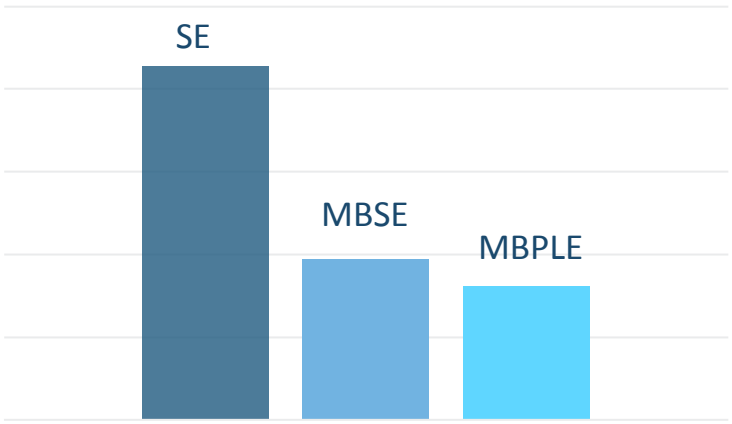
Development Cost Reduction & Delivery Time Improvements

- SE (Non-Modelled Systems Engineering)
  - 59% of Projects Delivered on Time
- MBSE (Model Based Systems Engineering)
  - 62% of Projects Delivered on Time

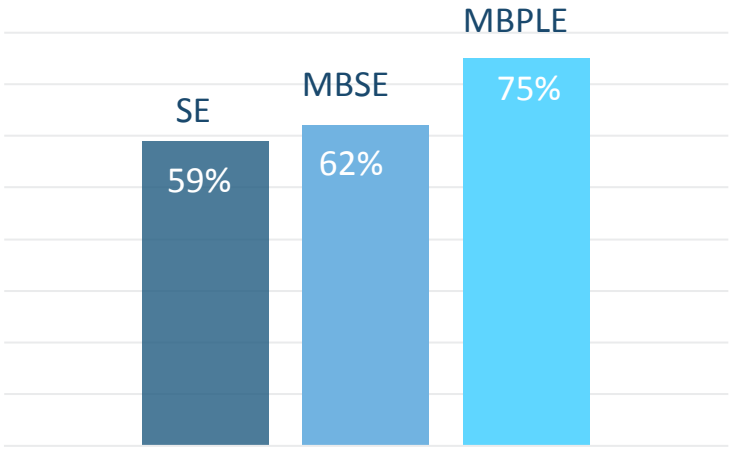
  Compared to SE
  - 55% Reduction in Total Development Cost per Project
- MB-PLE (Model Based Product Line Engineering)
  - 75% of Projects Delivered on Time

  Compared to SE
  - 62% Reduction in Total Development Cost per Project

(EMF 2013 Independent Survey Results from 667 Systems engineering respondents)



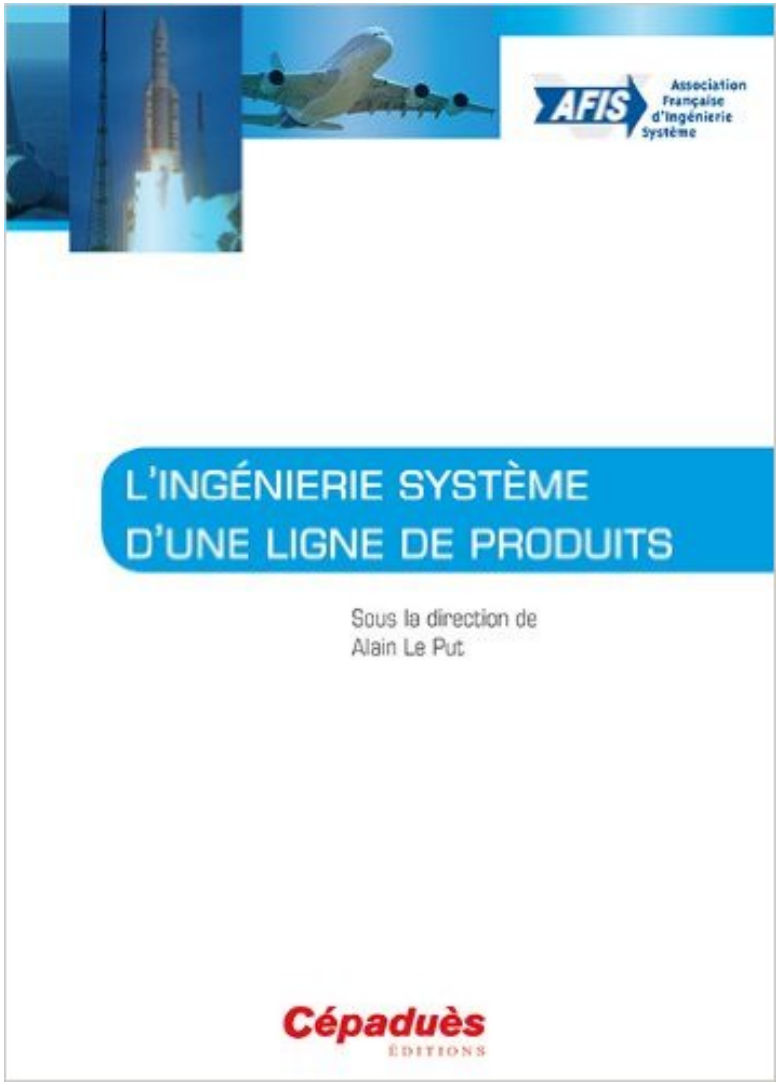
Development Cost per Project

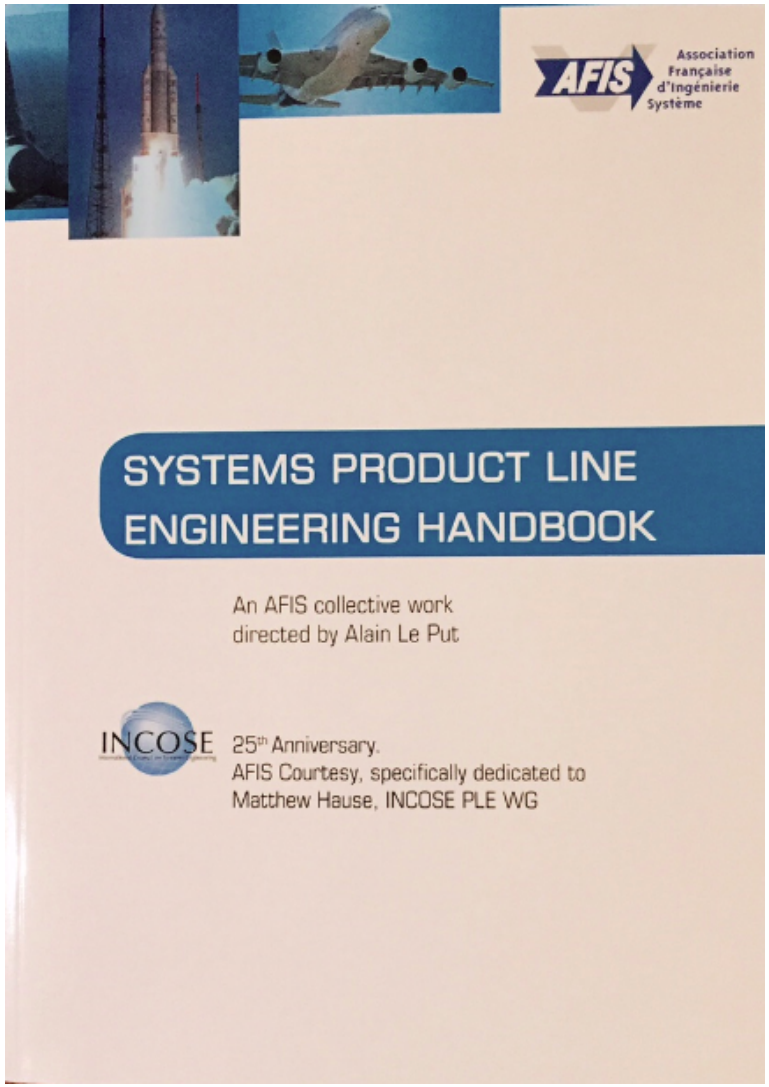


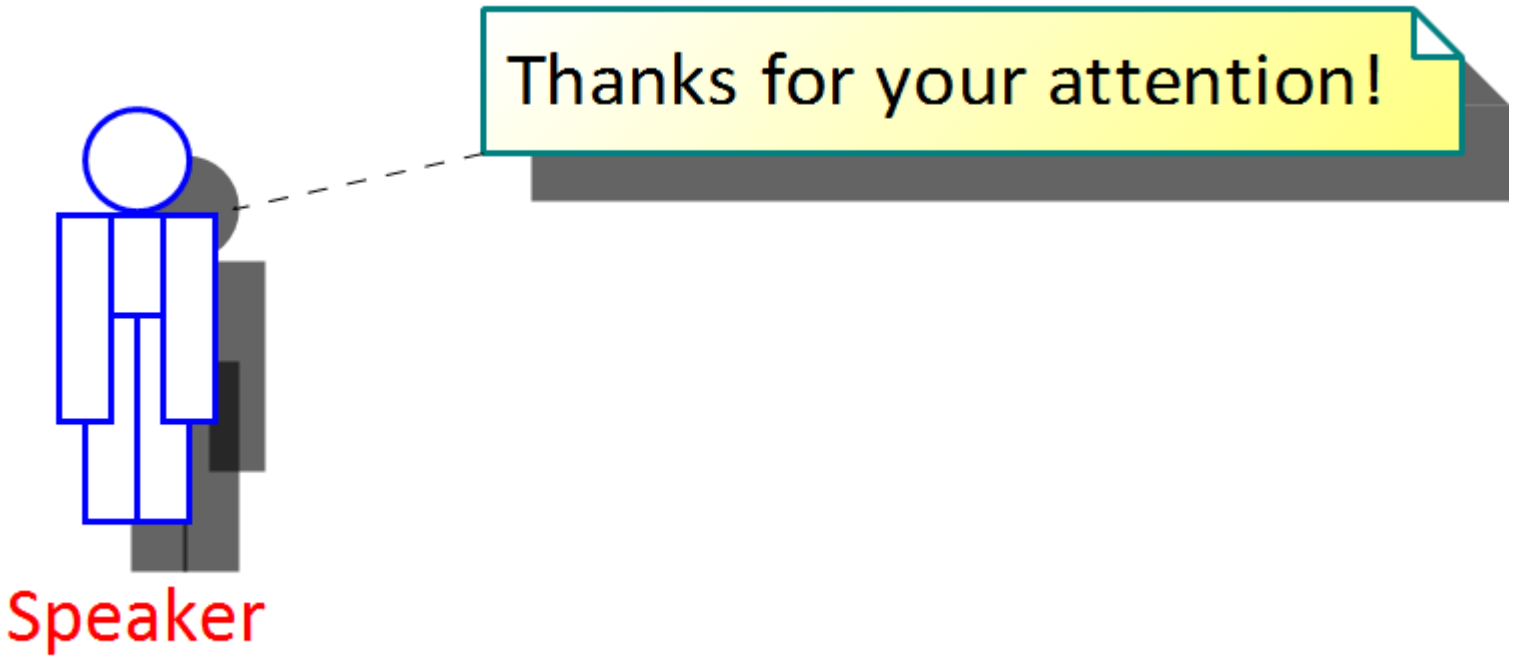
On Time Delivery

AFIS Book on Product Line Engineering (French/English)

These books are the foundation of the INCOSE Systems Product Line Engineering Handbook






Thanks for your attention!
Speaker

PTC®

PRODUCT & SERVICE ADVANTAGE®