pure systems

MBSE & PLE

Systematic Reuse in Engineering

18.09.2023

Introduction

Robert Hellebrand

Principal Field Application Engineer
robert.hellebrand@pure-systems.com
(+49) 0391 544569-38

pure-systems at a glance – HQ in Magdeburg

Established 2001, Privately Held, HQ in Germany

Automotive, Aerospace/Avionics, Defense/Security, Rail/Transportation, Industry Automation, Semiconductor

pure::variants is fully compliant with ISO 26580, a standard for Feature-based Product Line Engineering

Memberships AUTOSAR, INCOSE/GfSE, OASIS Open, prostep ivip,

Research projects SAFE, SPES-XT, VARIES, REVaMP²

pure variants

pure consult

Today´s Challenges

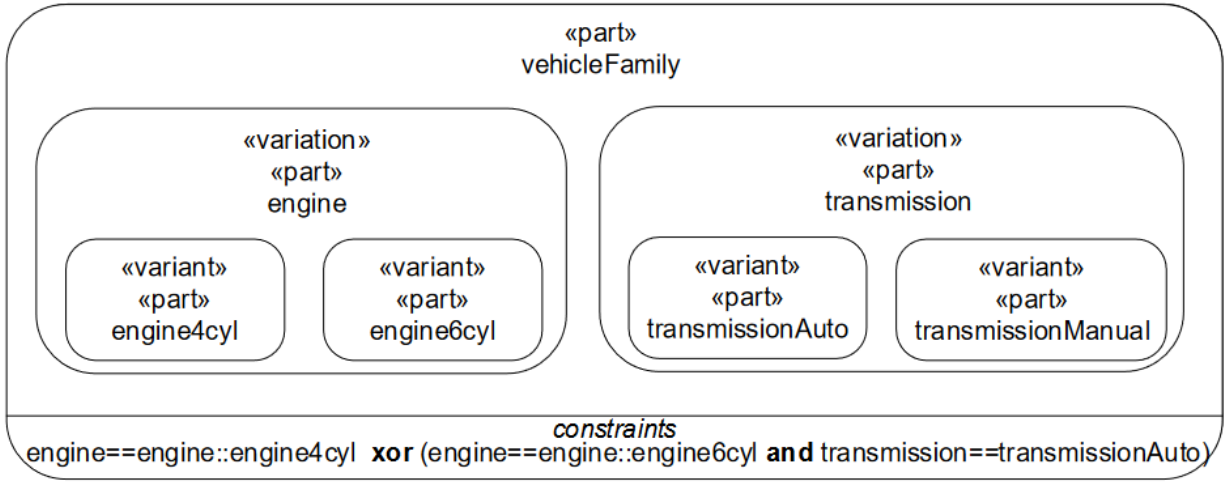
- Modern world requires
  product variants suited to specific needs.
- The complexity of products
  is constantly increasing.
- Customer expects improvements
  along the entire product lifecycle.
- New business models aim to
  upsell new features & services
  for products which are market-ready.

Variability in SysML V2

Variability v2

SST

- Vehicle family is the superset model (150%)
- *Variation* points represent elements that can vary
  - Can be applied to all definition and usage elements
- A variant represents a particular choice at a variation point
- A choice at one variation point can constrain choices at other variation points
- A system can be *configured* by making choices at each variation point



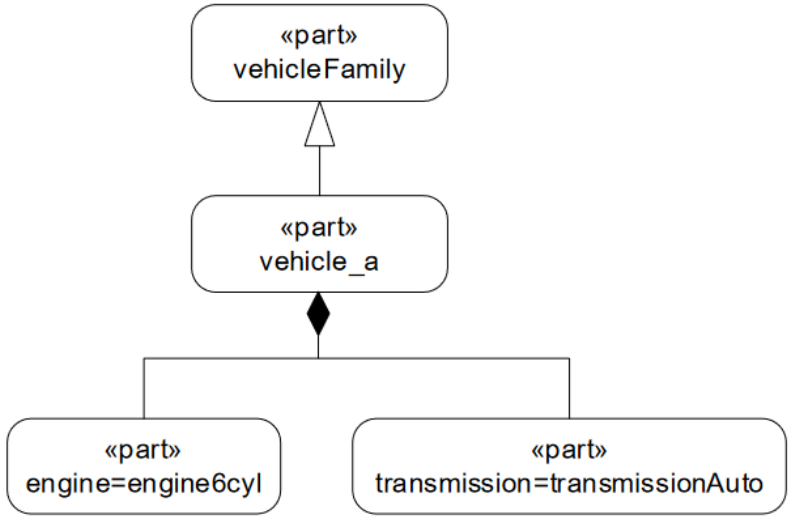
46 11 November 2022

Source: https://github.com/Systems-Modeling/SysML-v2-Release/blob/master/doc/Intro%20to%20the%20SysML%20v2%20Language-Graphical%20Notation.pdf

Selected Vehicle Configuration

SST

- Vehicle_a subsets vehicleFamily to represent a particular design configuration
- Selected parts must satisfy variability constraints
  - Model is inconsistent if constraints are not satisfied
  - Variability modeling applications can automate the selection of valid configurations



47

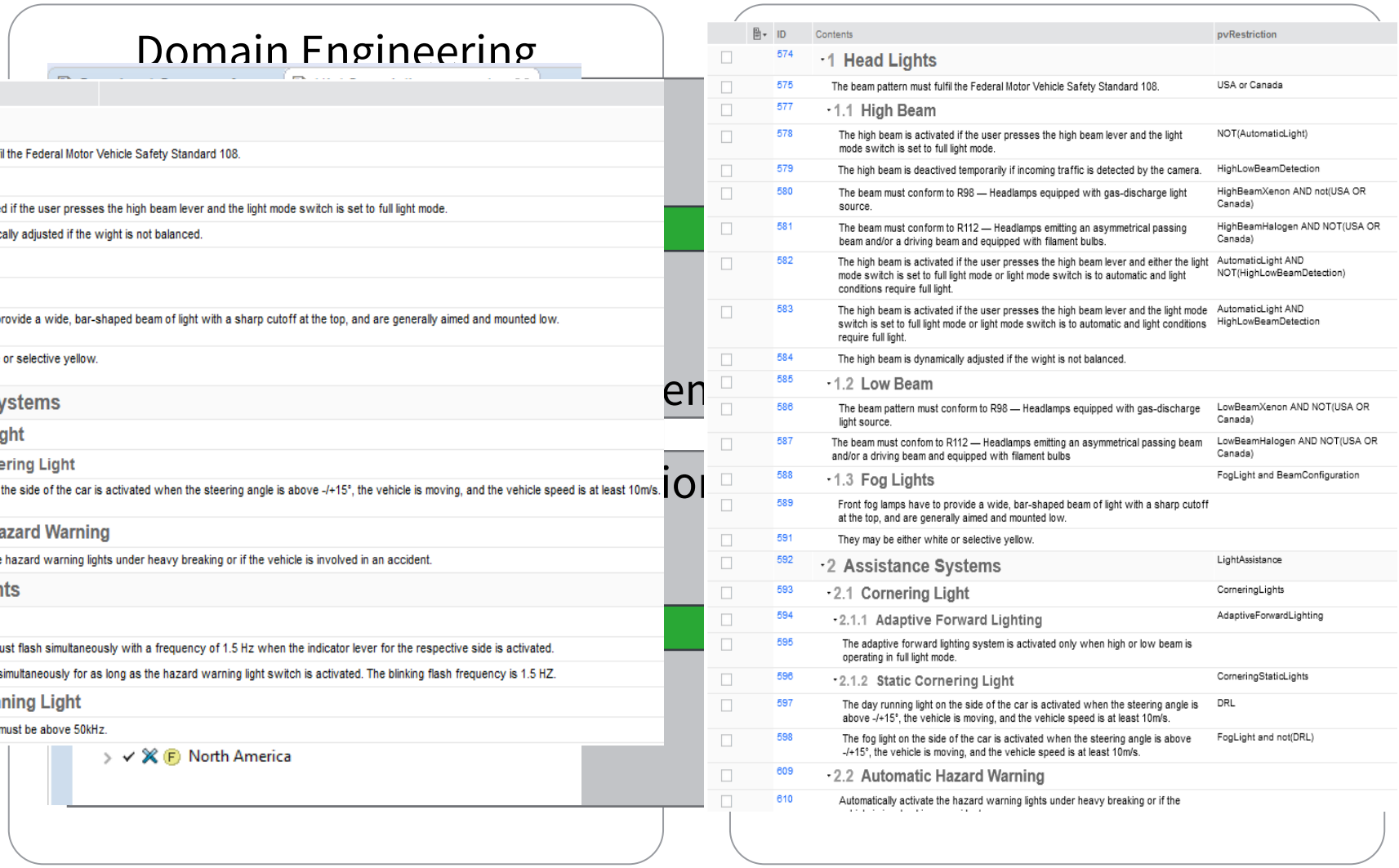
11 November 2022

Source: https://github.com/Systems-Modeling/SysML-v2-Release/blob/master/doc/Intro%20to%20the%20SysML%20v2%20Language-Graphical%20Notation.pdf

Product Line Engineering

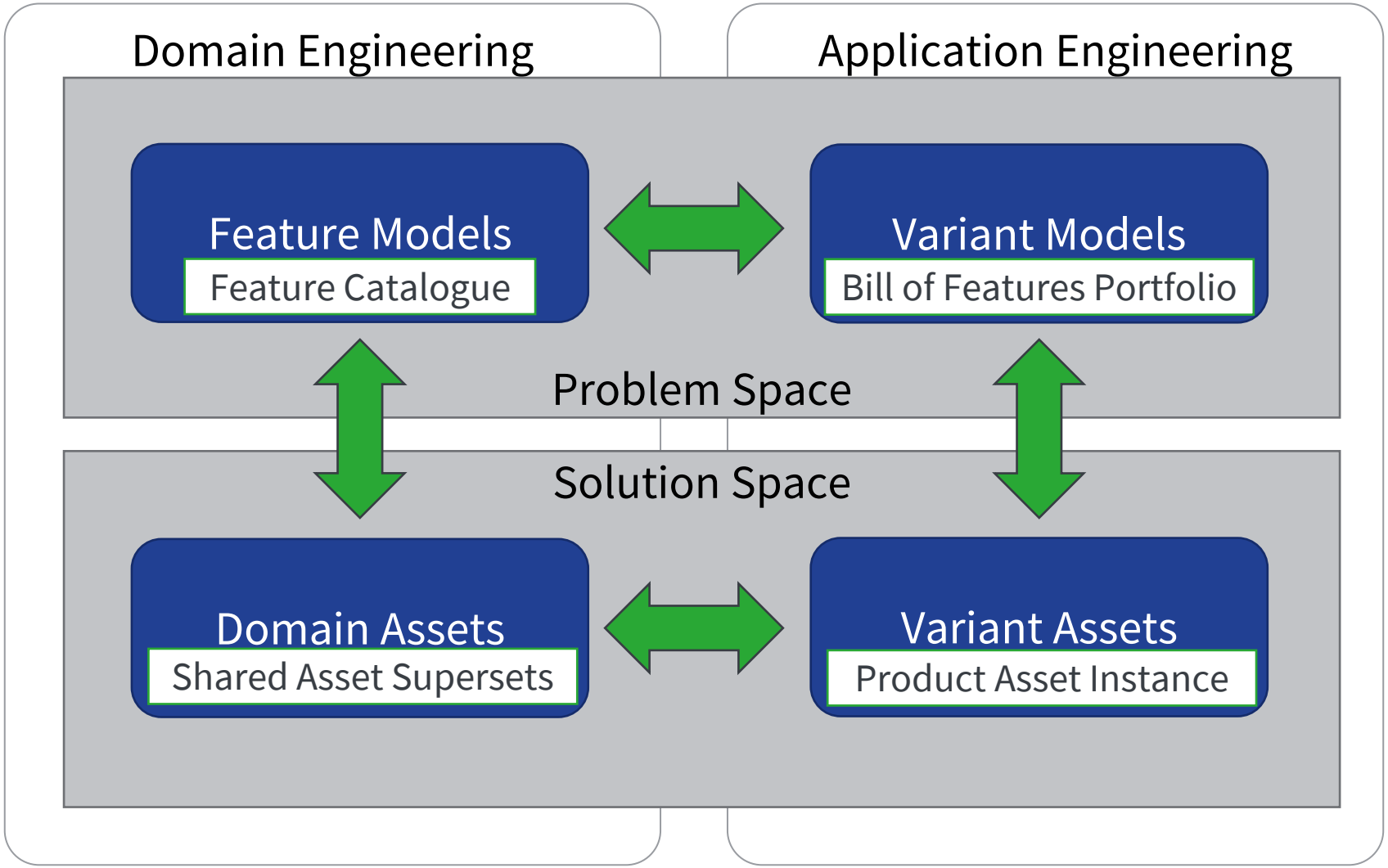
ISO 26580 | pure::variants

Product Line Engineering with pure::variants

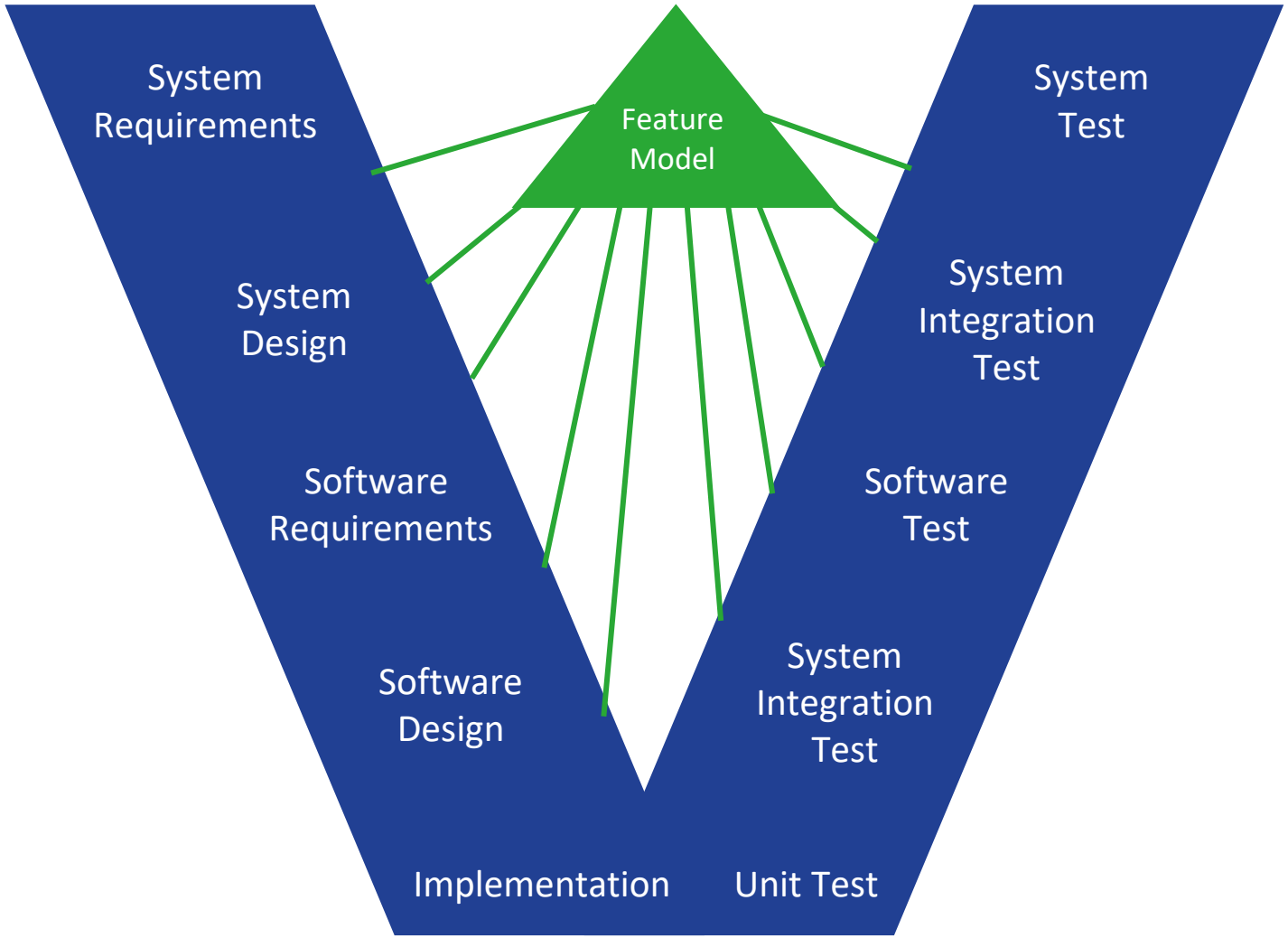


Product Line Engineering with pure::variants

ISO 26580

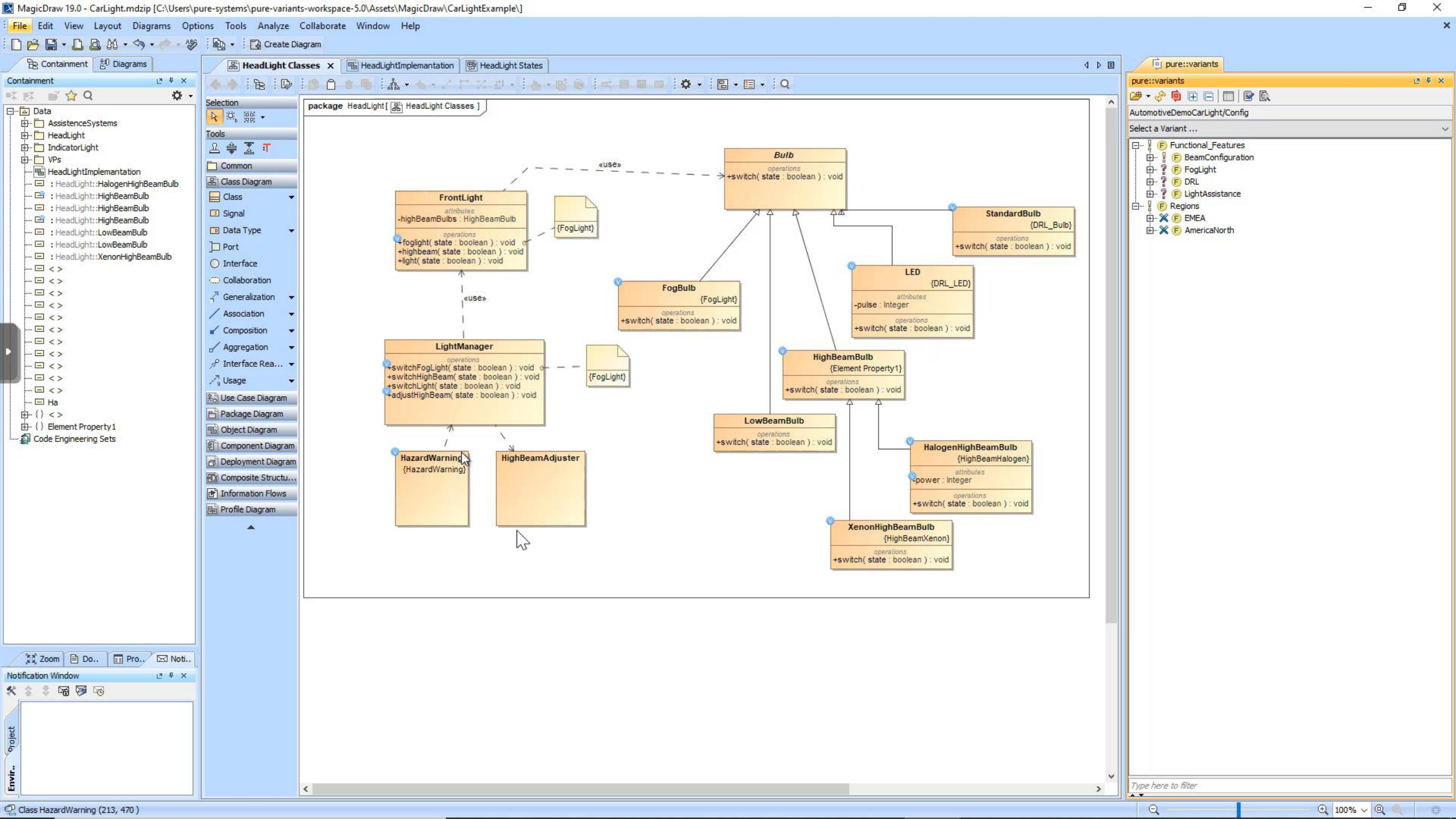


Consistent Reuse across the Lifecycle



MBSE & PLE

pure::variants Connector for Cameo Magic & Cameo No Magic


MagicDraw 19.0 - CarLight.mdzip [C:\Users\pure-systems\pure-variants-workspace-5.0\Assets\MagicDraw\CarLightExample\]
File Edit View Layout Diagrams Options Tools Analyze Collaborate Window Help
Create Diagram
Containment Diagrams
Containment
Data
AssistenceSystems
HeadLight
IndicatorLight
VPs
HeadLightImplementation
: HeadLight::HalogenHighBeamBulb
: HeadLight::HighBeamBulb
: HeadLight::HighBeamBulb
: HeadLight::HighBeamBulb
: HeadLight::LowBeamBulb
: HeadLight::LowBeamBulb
: HeadLight::XenonHighBeamBulb
Element Property1
Code Engineering Sets
HeadLight Classes
HeadLightImplementation
HeadLight States
Selection
Tools
Common
Class Diagram
Class
Signal
Data Type
Port
Interface
Collaboration
Generalization
Association
Composition
Aggregation
Interface Rea...
Usage
Use Case Diagram
Package Diagram
Object Diagram
Component Diagram
Deployment Diagram
Composite Structu...
Information Flows
Profile Diagram
package HeadLight [HeadLight Classes]
«use»
Bulb
operations
+switch(state : boolean) : void
FrontLight
attributes
-highBeamBulbs : HighBeamBulb
operations
+foglight(state : boolean) : void
+highbeam(state : boolean) : void
+light(state : boolean) : void
{FogLight}
«use»
StandardBulb
{DRL_Bulb}
operations
+switch(state : boolean) : void
FogBulb
{FogLight}
operations
+switch(state : boolean) : void
LED
{DRL_LED}
attributes
-pulse : Integer
operations
+switch(state : boolean) : void
LightManager
operations
+switchFogLight(state : boolean) : void
+switchHighBeam(state : boolean) : void
+switchLight(state : boolean) : void
+adjustHighBeam(state : boolean) : void
{FogLight}
HighBeamBulb
{Element Property1}
operations
+switch(state : boolean) : void
LowBeamBulb
operations
+switch(state : boolean) : void
HazardWarning
{HazardWarning}
HighBeamAdjuster
HalogenHighBeamBulb
{HighBeamHalogen}
attributes
-power : Integer
operations
+switch(state : boolean) : void
XenonHighBeamBulb
{HighBeamXenon}
operations
+switch(state : boolean) : void
pure::variants
AutomotiveDemoCarLight/Config
Select a Variant ...
Functional_Features
BeamConfiguration
FogLight
DRL
LightAssistance
Regions
EMEA
AmericaNorth
Type here to filter
Zoom Do.. Pro.. Noti..
Notification Window
Class HazardWarning (213, 470)
100%

Summary

- Product Line Engineering (PLE) is complementary to Systems Engineering
- PLE externalizes dependency knowledge to feature models
- PLE applies dependency knowledge consistently across engineering assets and tool borders in order to implement a holistic reuse of engineering assets
- PLE is standardized in ISO 26580 (worked out by INCOSE PLE Group)
- pure::variants as a PLE tool comes with Connectors for ALM / MBSE / Functional Safety Tools, Code, …

Thank you!

www.pure-systems.com