Integrating Descriptive Models with an Analytical Model Culture – Lessons Learned at Ford

Kyle Post, George Walley and Judy Che

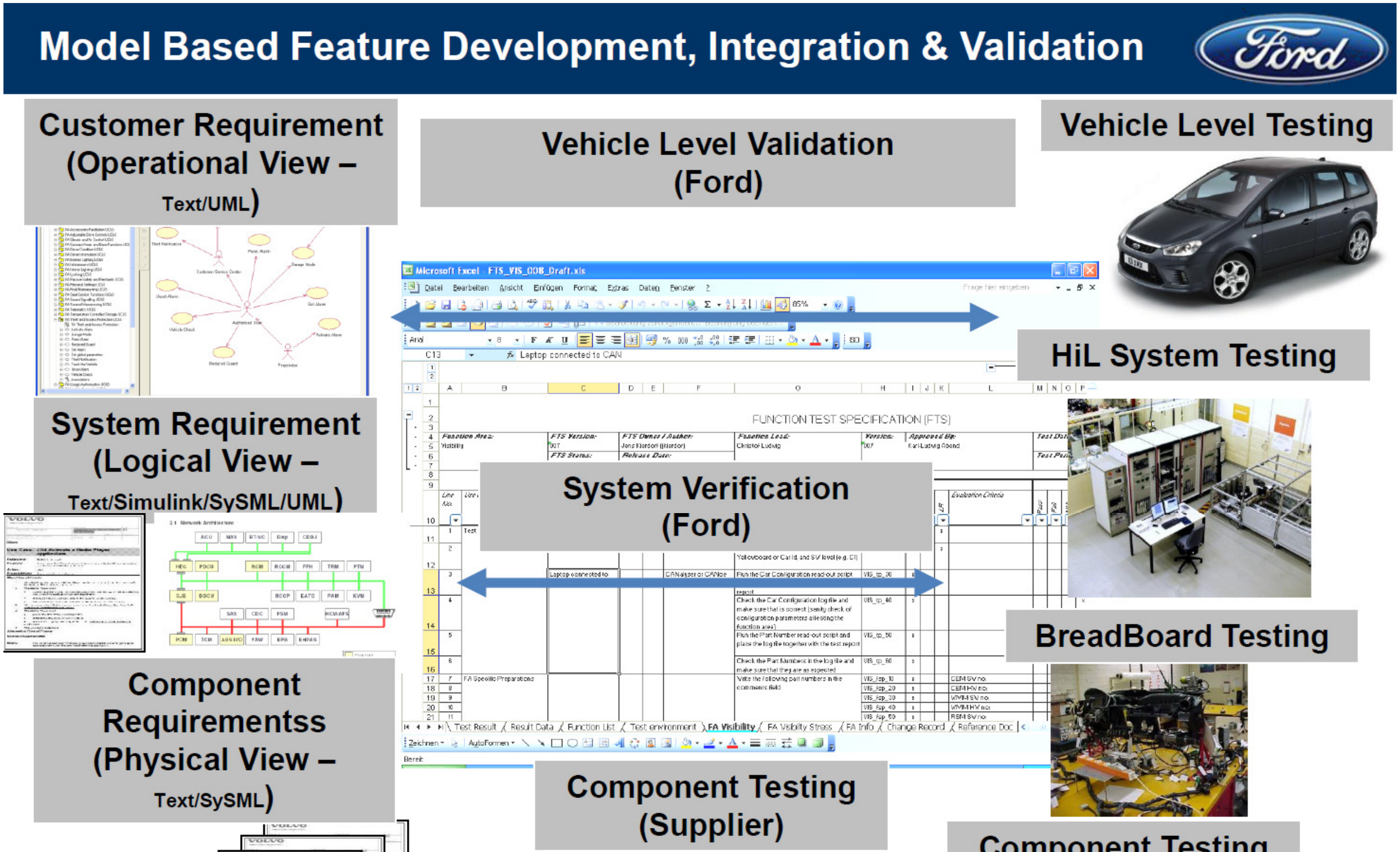
January 25, 2014 – INCOSE IW 2014 / MBSE Workshop

Ford Motor Company

In this Presentation…

- Motivation
- Descriptive & Analytical models at Ford
- Understand role for and line between descriptive (sysml) and analytical or implementation models
- Modeling and eliciting requirements through response diagrams tied to system models
- Exploring Model Based Failure Mode Avoidance
- Experiences in integration SysML and PLM systems for requirements and feature/function breakdowns
- Leveraging SysML in an Integrated Vehicle Analysis process

Motivation

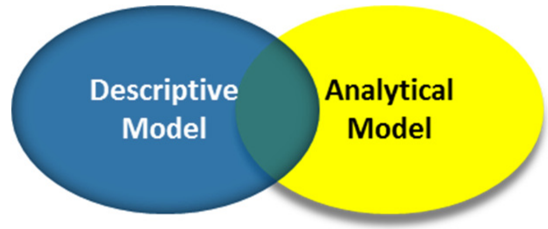


As presented by Ford colleagues in the past, an increase and influx of models and model-based approaches are being used to develop, integrate, test, and manage our increasing complex vehicle systems

Source: Davey, C., 2013, "Automotive Software Systems Complexity: Challenges & Opportunities", INCOSE IW 2013/MBSE WS

DESCRIPTIVE MODELS VS. ANALYTICAL MODELS

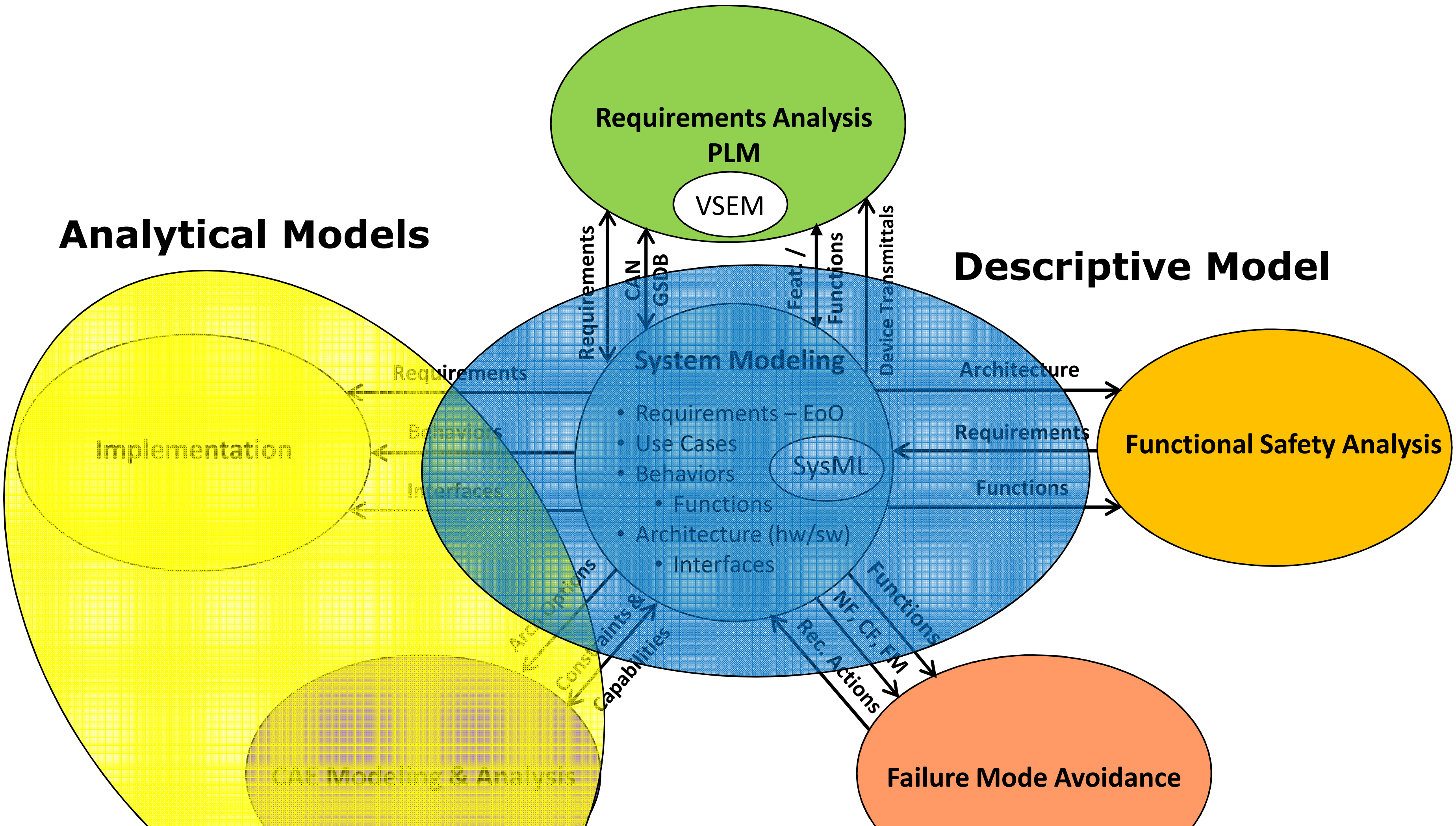
Using SysML for descriptive modeling as a useful addition to analytical models



Descriptive Models

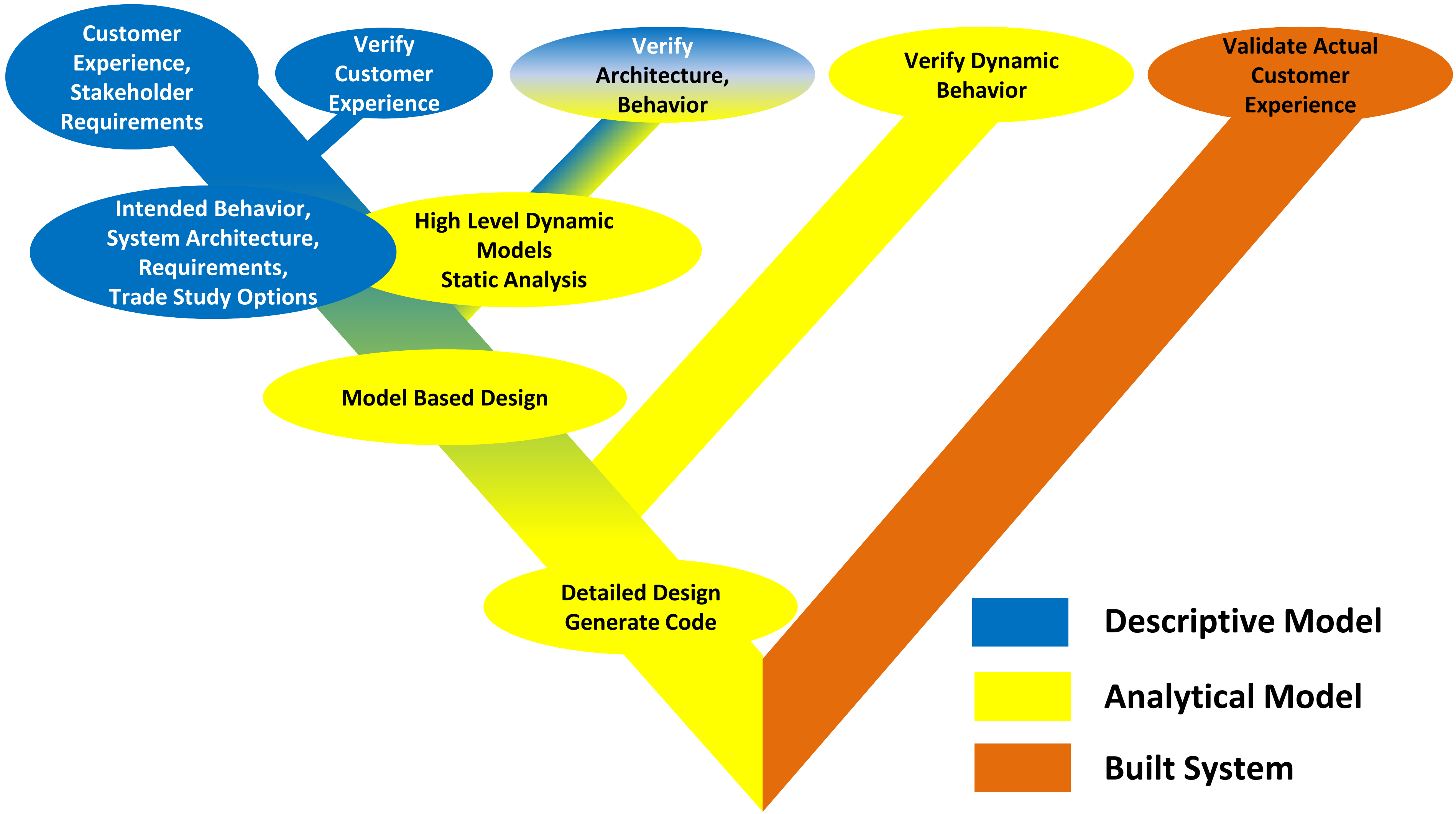
- Enable the transition from a document-based system engineering approach to a model-based approach
- Provide an abstract view of the system that can be analyzed before building more time consuming analytical models and implementation
- Is not restricted to a closed form mathematical equation as analytical models are
- Convey multiple viewpoints (e.g. Structural view vs. Functional view vs. Physical)
- Used as a master to coordinate and connect engineering toolsets (e.g. analytical models, PLM systems, test benches, etc…)

Descriptive / Analytical Model Breakdown



Distinguishing the purpose and value of different types of models in an overall MBSE strategy helped alleviate concerns that SysML was yet another modeling language to compete with existing analytical modeling languages.

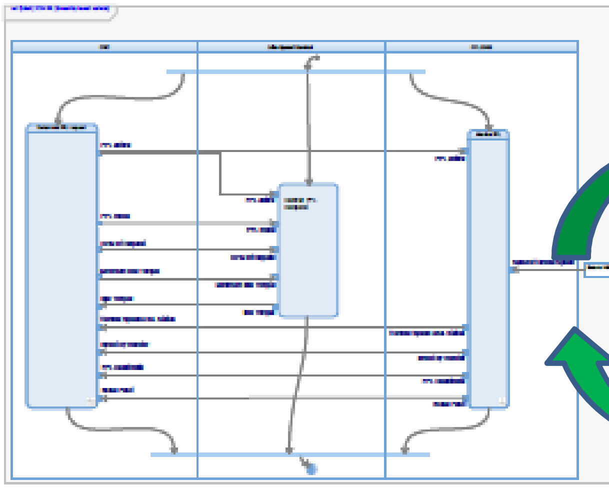
DESCRIPTIVE / ANALYTICAL PROCESS BREAKDOWN



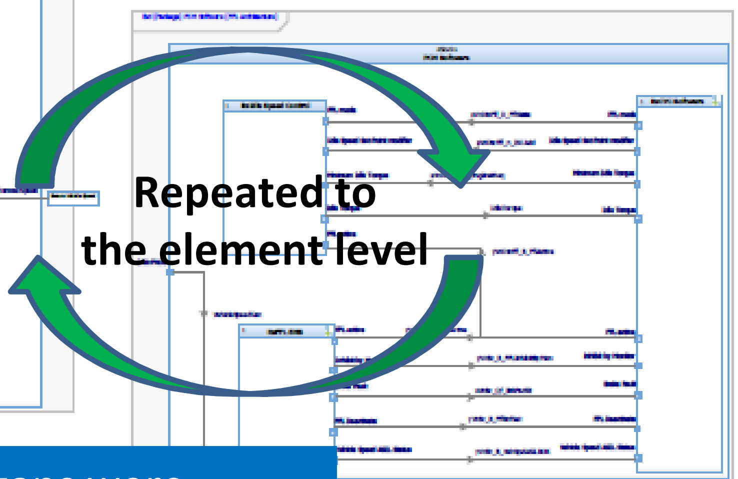
Reference: **SysML role in Model Driven Requirements Engineering (MoDRE) & MBSE**, Kyle Post and George Walley, 2013

Descriptive Modeling using SysML

System high level functional requirements are modeled in SysML which are then decomposed into lower level behavior models

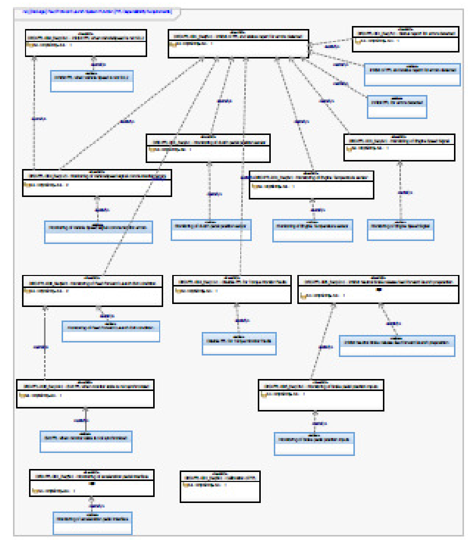


Behaviors are partitioned into a logical architecture and/or physical architecture



Repeated to the element level

The functional system requirements and derived requirements are linked to the SysML behaviors using a Requirements Diagram



Requirements gaps were identified when modeling the behaviors which were not apparent from the textual requirements alone

Descriptive Model Examples

Customer Experience, Stakeholder Requirements

Verify
itecture,
ehavior

Verify Dynamic Behavior

Validate Actual Customer Experience

System Model used to define the features to be developed

Compared to a more textual document approach the models generate more constructive feedback on the proposed features which are quickly iterated on during review meetings.

In one case a proposed project, which was originally not kicked off due to questions on the ability to meet timing, was approved on the spot after creating a SysML model and walking the decision makers through the diagrams. None of the decision makers involved had ever heard or seen SysML before the meeting.

Descriptive Model Examples

Intended Behavior, System Architecture, Requirements, Trade Study Options

Verify Architecture, Behavior

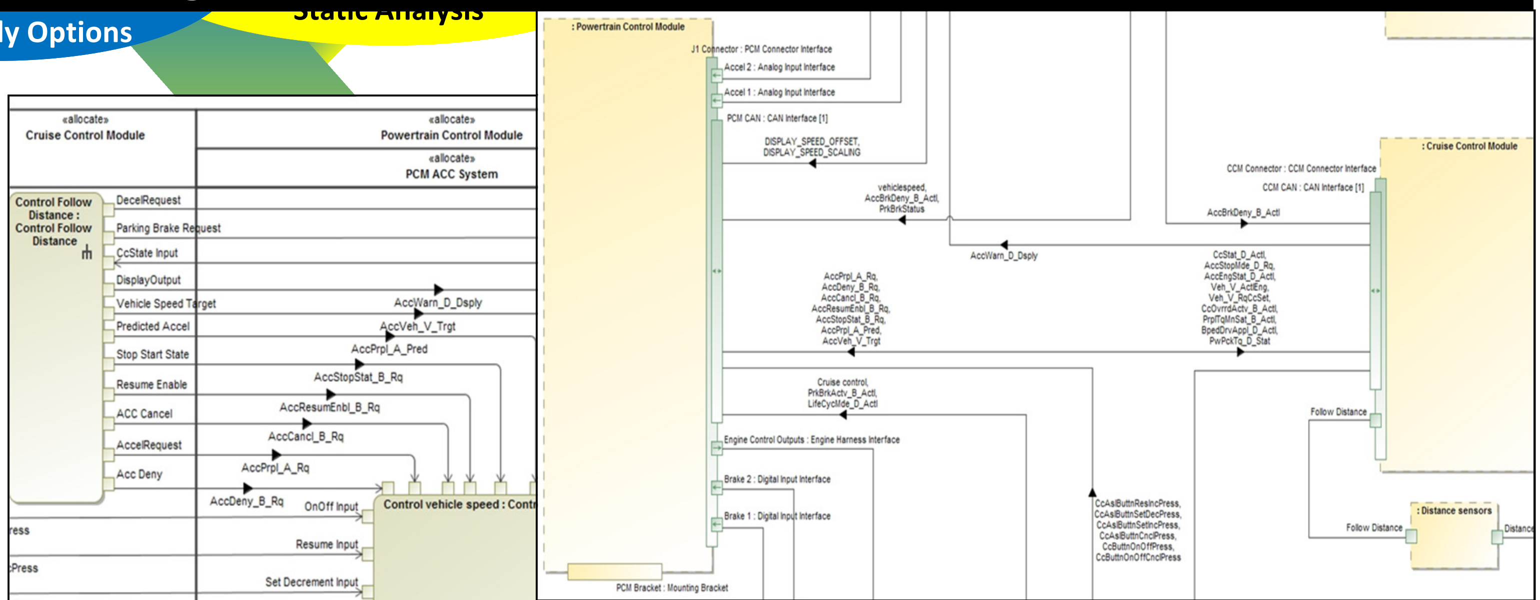
Verify Dynamic Behavior

Validate Actual Customer Experience

System Model used to define the behaviors that the features perform along with the logical architecture

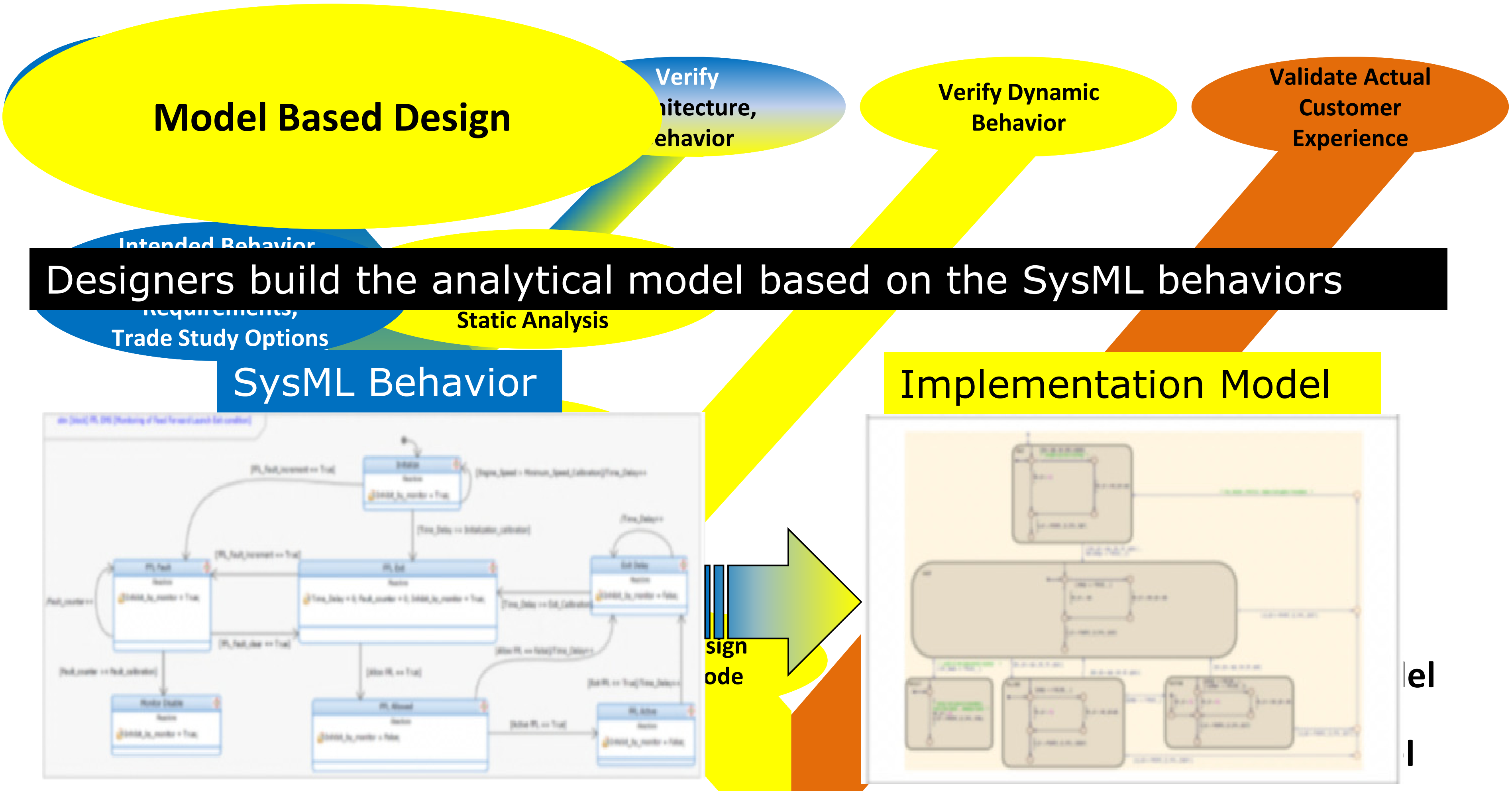
Static Analysis

Trade Study Options



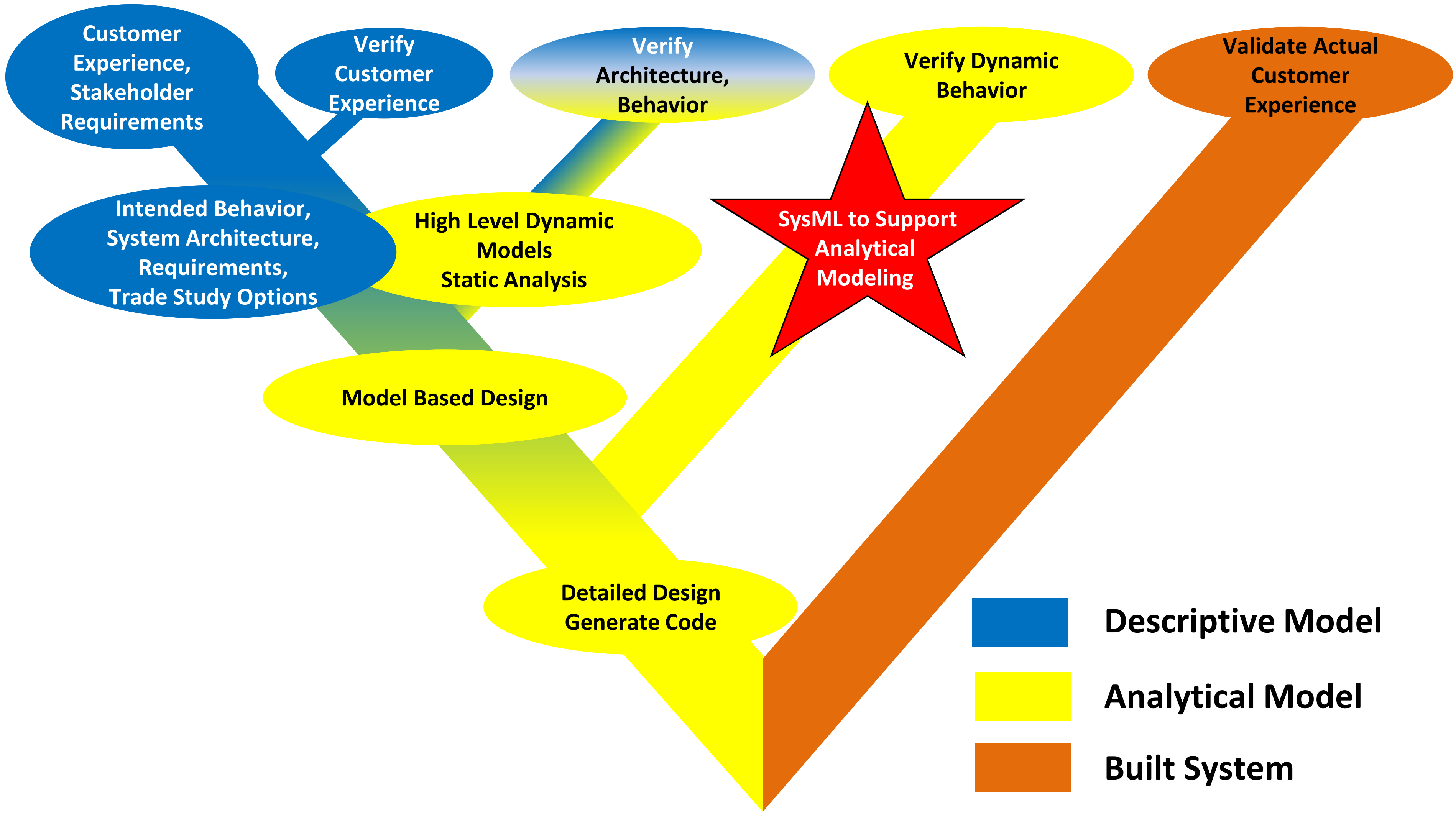
There is more engagement and debate of the system behavior and architectures earlier in the development cycle as people are able to interpret the diagrams easier than with textual requirements alone.

Descriptive Model Examples



Implementation of the model based functional specification resulted in more consistent behaviors, even when given to multiple suppliers, compared with similar traditional document based experiences

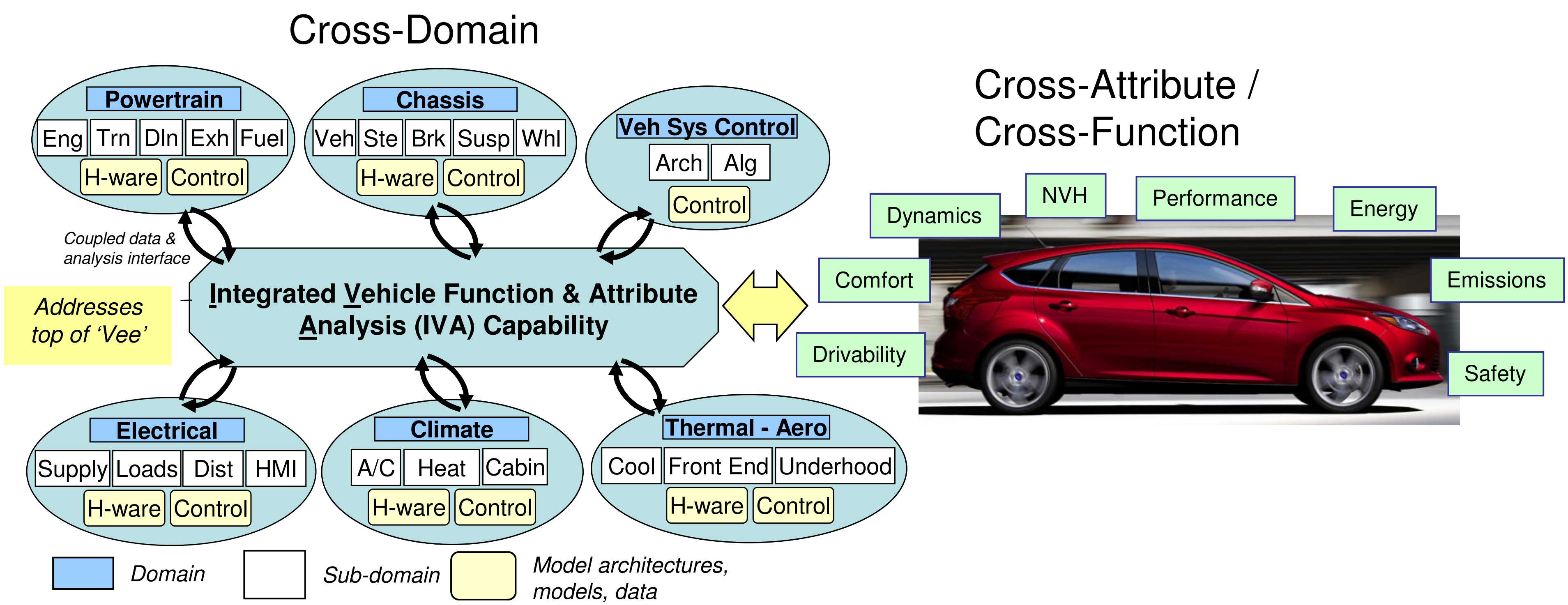
DESCRIPTIVE / ANALYTICAL MODEL BREAKDOWN



Reference: **SysML role in Model Driven Requirements Engineering (MoDRE) & MBSE**, Kyle Post and George Walley, 2013

Integrated Vehicle Analysis

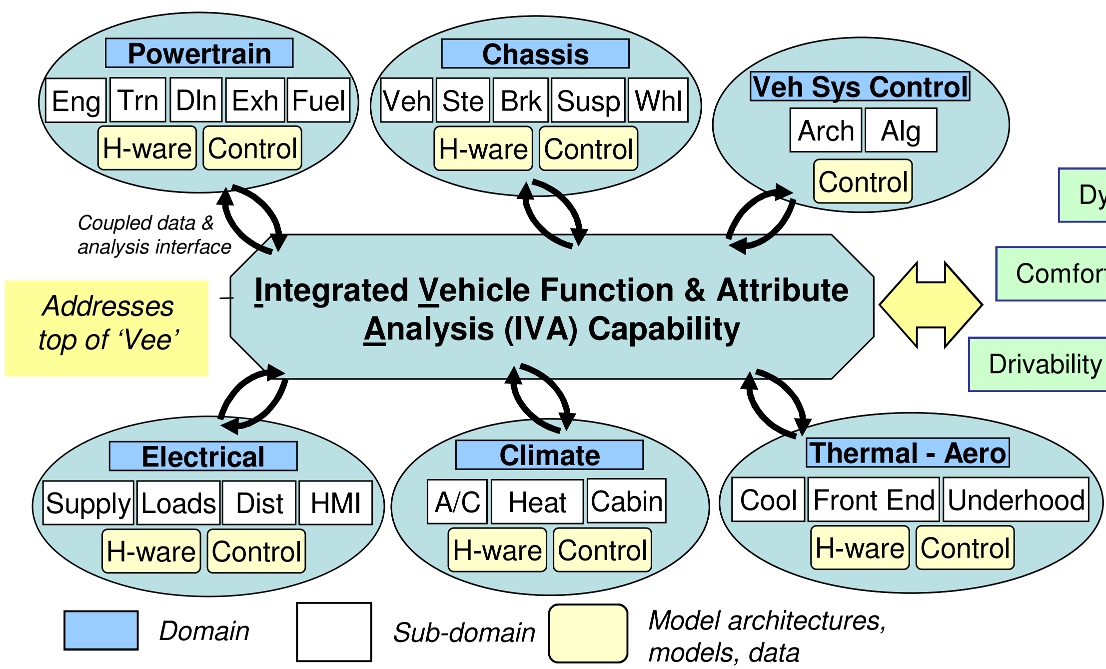
- Fully integrated vehicle system models are needed to simulate vehicle-level, cross-functional attributes (e.g. Fuel Economy, Performance, etc.)
- Vehicle models are built up of sub-system models from various domains created by subject-matter-experts
- Objective of Integrated Vehicle Analysis (IVA): Develop and optimize system design for critical vehicle level attributes



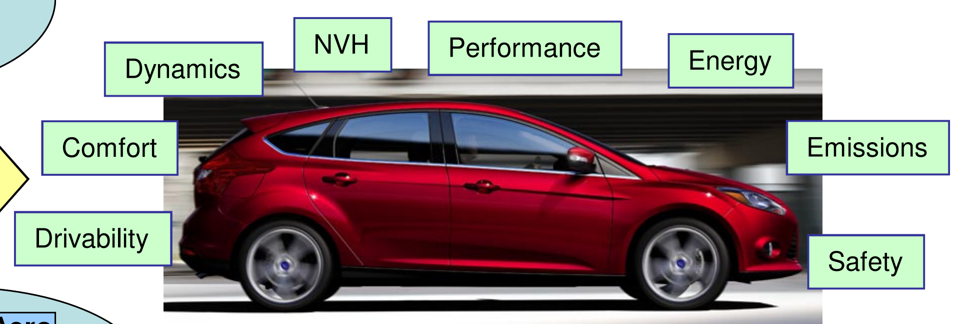
Integrated Vehicle Analysis

Challenges:
- Vehicle system models can be very complex
- Simulations of different attributes require different models (i.e. fidelity, operating range, etc) – requires planning and coordination
- Numerous domains & dozens model developers/integrators participate
- Communication across domain areas – vocabulary, processes, tools, interfaces
- Integration & testing of vehicle models is a tremendous task

Cross-Domain



Cross-Attribute / Cross-Function

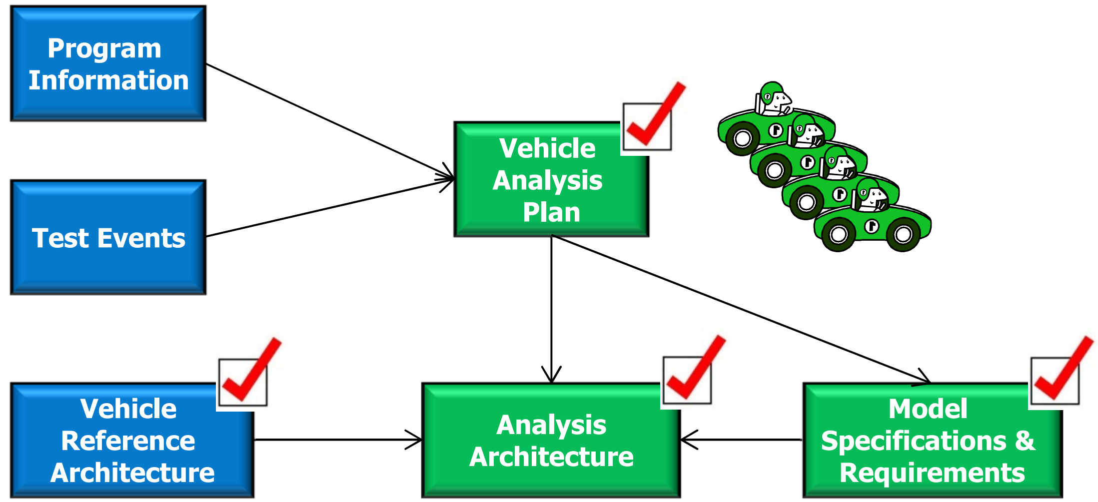


Vehicle System Analysis Process

ANALYSIS PLANNING

MODEL DEVELOPMENT

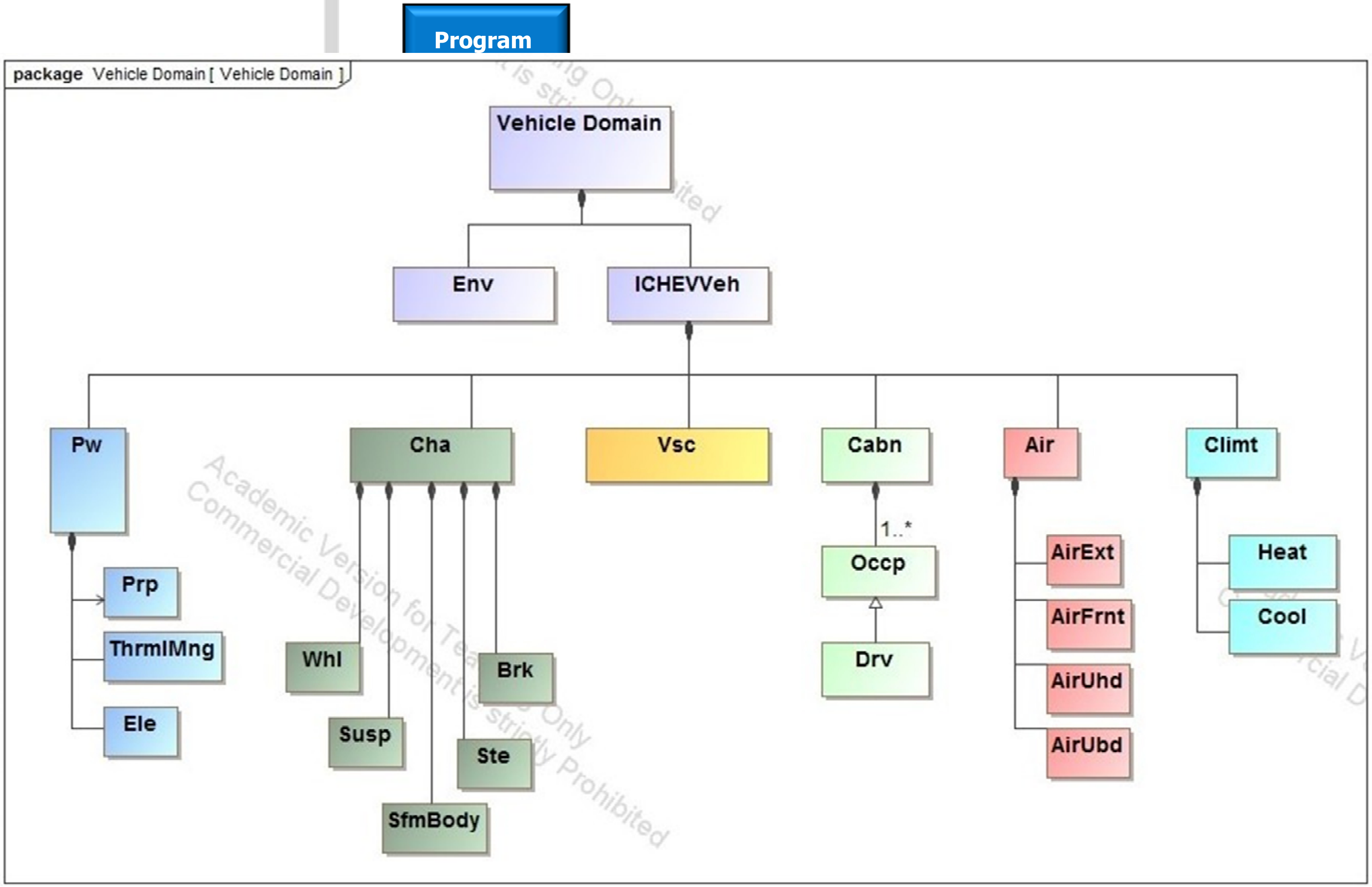
DESIGN ANALYSIS




SysML Model Representation

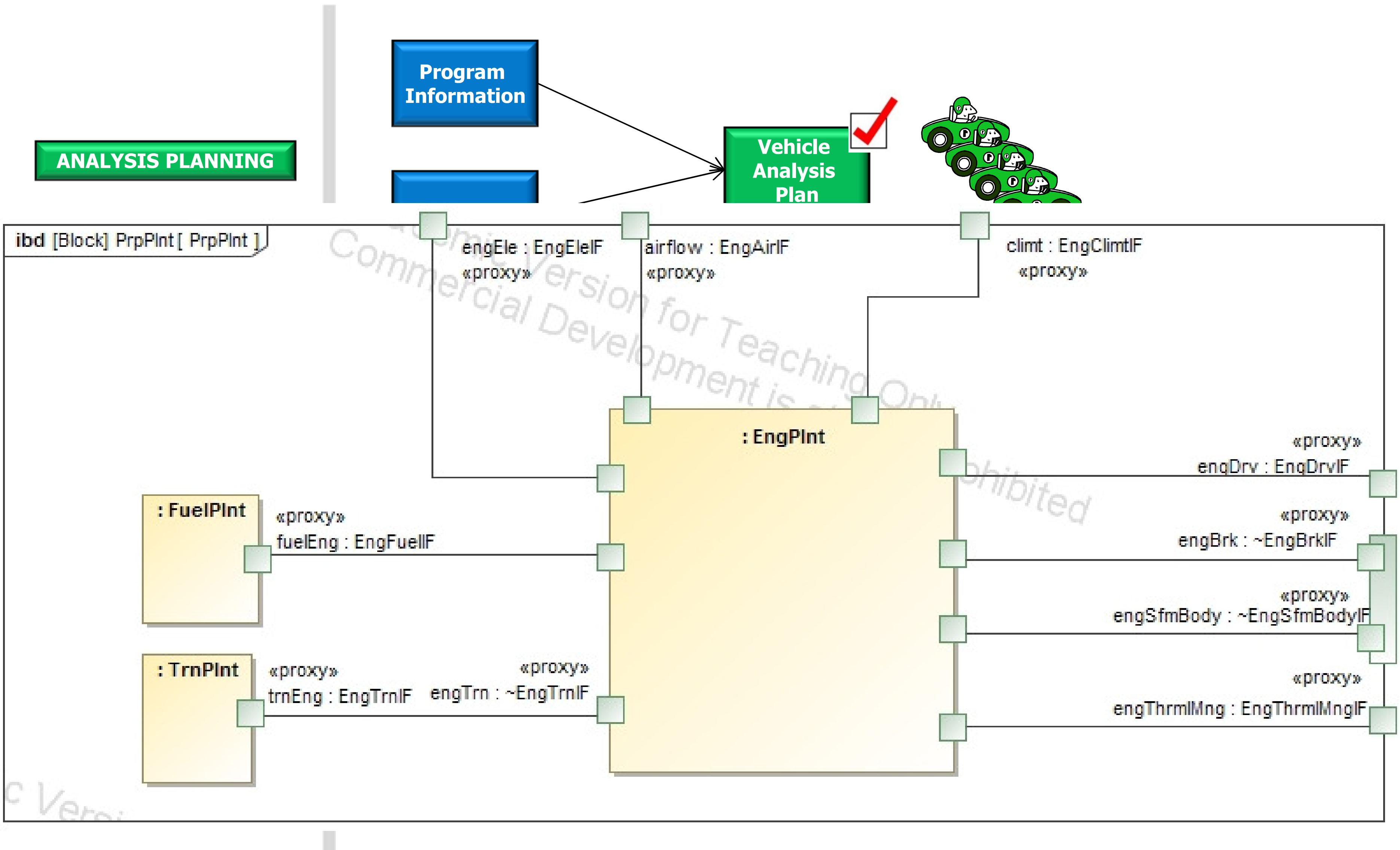
Source: William C. Bailey, et al., "Using model-based methods to support vehicle analysis planning," Conference on Systems Engineering Research (CSER 2014)

Vehicle System Analysis Process



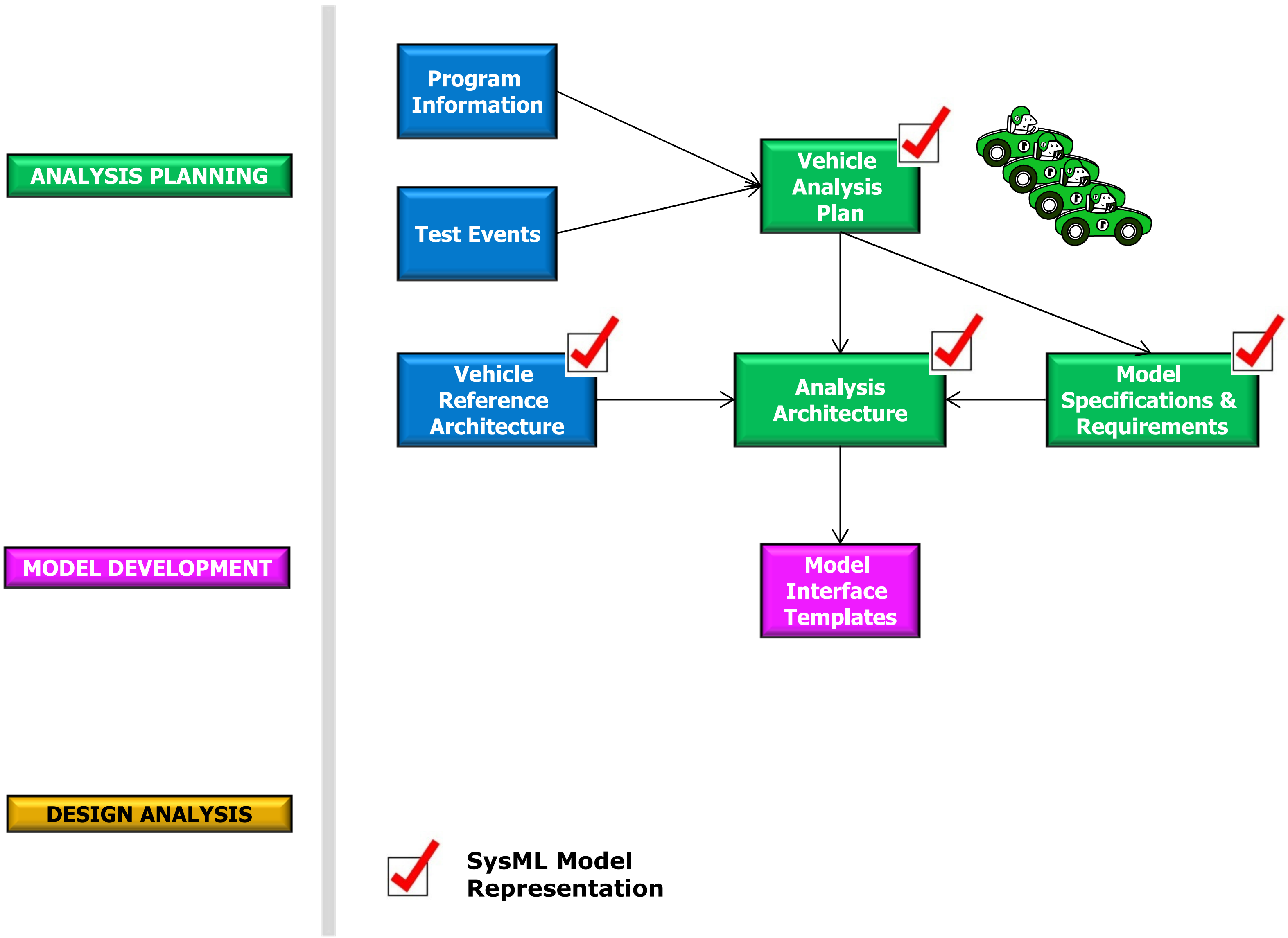
Source: William C. Bailey, et al., "Using model-based methods to support vehicle analysis planning," Conference on Systems Engineering Research (CSER 2014)

Vehicle System Analysis Process



Source: William C. Bailey, et al., "Using model-based methods to support vehicle analysis planning," Conference on Systems Engineering Research (CSER 2014)

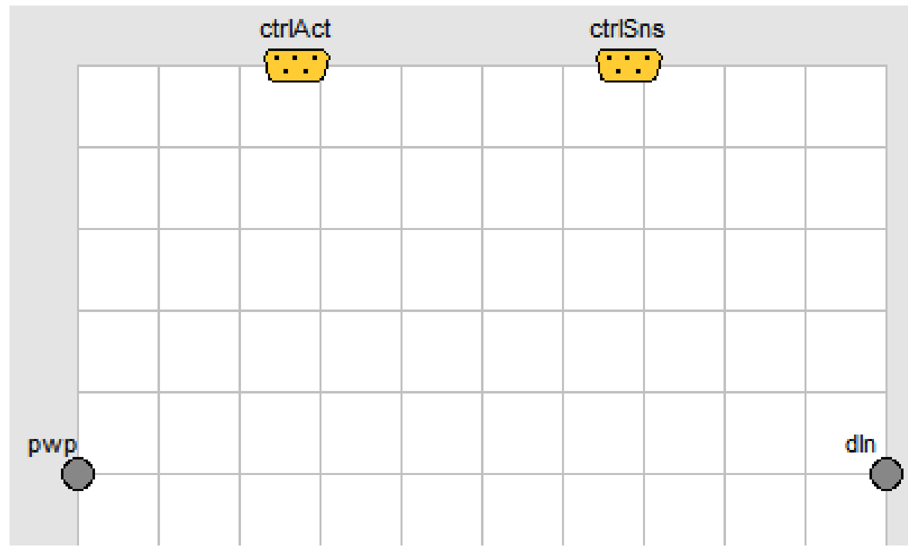
Vehicle System Analysis Process



Model Transformations and Interface Templates

Simulink Model Transformation :omb *et. al*

– Georgia Tech student with Chris Paredis



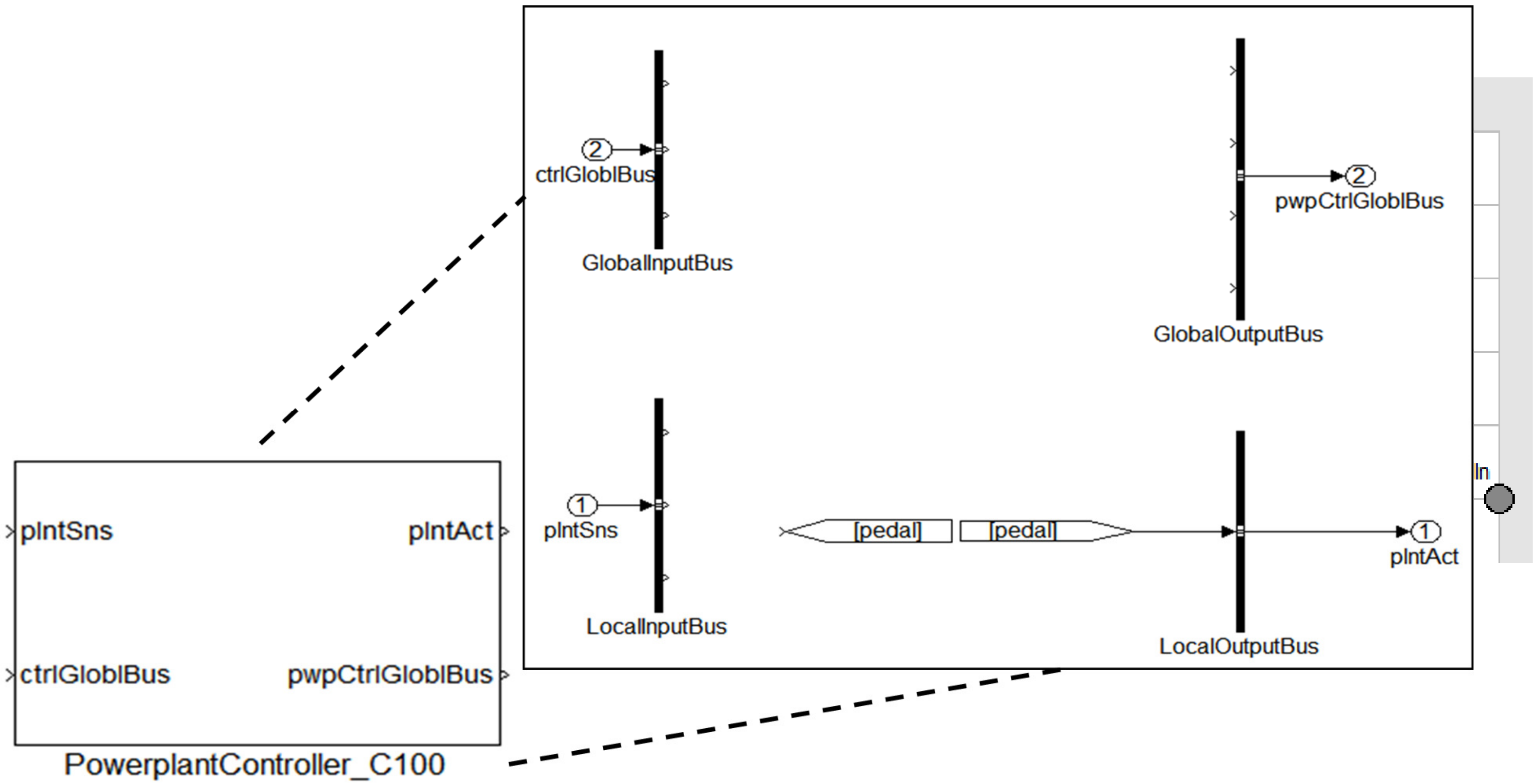

```
partial model TransmissionPlant_C100

  Vehicle_0To100kph_C100.ControlBus_C100 ctrlSns
  annotation (Placement(
        transformation(
        extent={{-16,-16},{16,16}},
        rotation=0,
        origin={36,100})));
  Vehicle_0To100kph_C100.ControlBus_C100 ctrlAct
  annotation (Placement(
        transformation(
        extent={{-16,-16},{16,16}},
        rotation=0,
        origin={-46,100})));
  Modelica.Mechanics.Rotational.Interfaces.Flange_a pwp
  annotation (Placement(transformation(extent={{-110,-10},{-90,10}})));
  Modelica.Mechanics.Rotational.Interfaces.Flange_a dln
  annotation (Placement(transformation(extent={{90,-10},{110,10}})));
  annotation (Icon(graphics={  Rectangle(extent = {{-64, 52}, {72, -36}},
        lineColor = {0, 0, 255}, fillColor = {135, 135, 135},
            fillPattern = FillPattern.Solid), Text(extent = {{-58, 68}, {66, 54}},
            lineColor = {0, 0, 255}, fillColor = {135, 135, 135},
            fillPattern = FillPattern.Solid, textString = "%TransmissionPlant")}), Diagram(graphics));
end TransmissionPlant_C100;
```

Source: Branscomb, J. et al., 2013, "Supporting Multidisciplinary Vehicle Analysis Using a Vehicle Reference Architecture Model in SysML," *Procedia Computer Science*

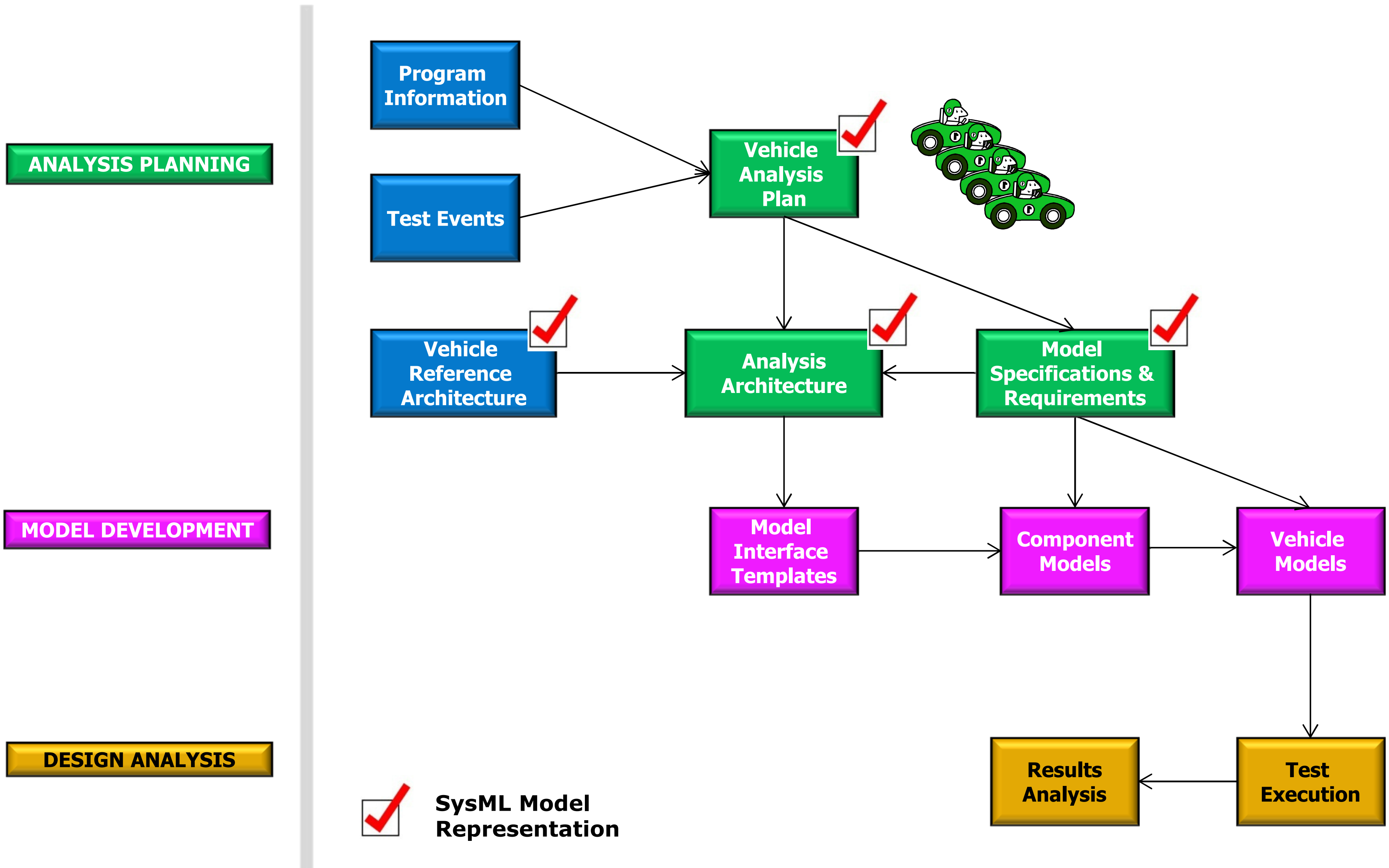
Model Transformations and Interface Templates

Simulink Model Transformation



Source: Branscomb, J. et al., 2013, "Supporting Multidisciplinary Vehicle Analysis Using a Vehicle Reference Architecture Model in SysML," *Procedia Computer Science*

Vehicle System Analysis Process



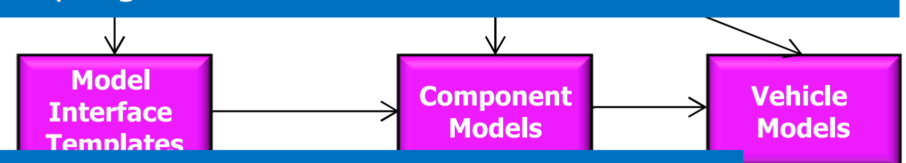
Vehicle System Analysis Process

ANALYSIS PLAI

Overall Benefits:

- Improved communication across domains – model interfaces and requirements
- Common understanding of simulation goals
- Upfront planning of model/data needs
- SysML acts as master source of information – live/not static
- Process automation avoids errors
- SysML for model development and integration:
  - Reduces time better support upfront program decisions
  - Reduces effort support more programs

MODEL DEVELOPMENT

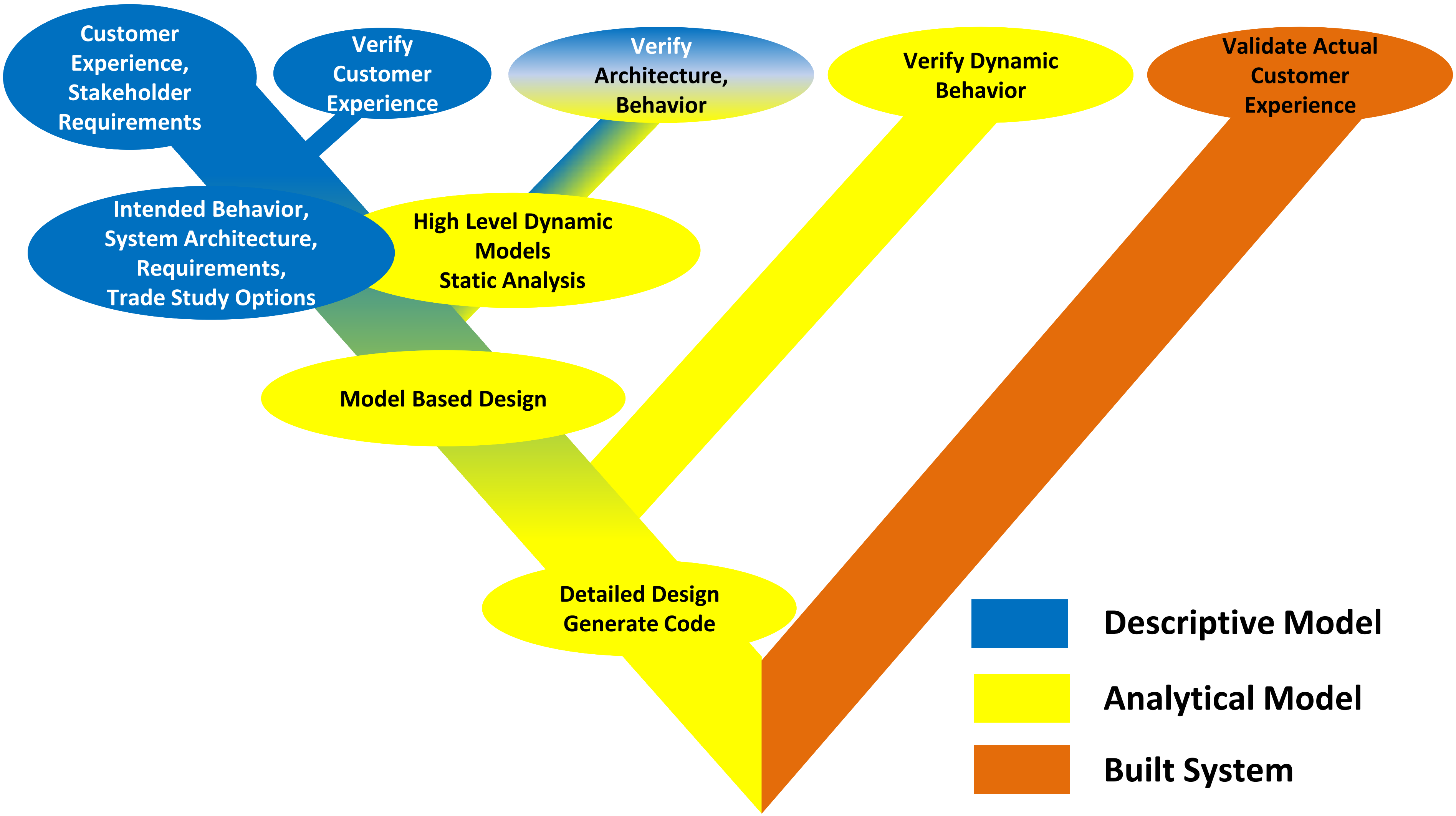


Observations:

- Version control and collaboration tools and methods essential
- Many of our users are not SysML experts; methods needed to support novice users
- Scalability SysML not an efficient way to store lots information
- Formal SysML semantics to manage variants would be useful

Test Execution

DESCRIPTIVE / ANALYTICAL PROCESS BREAKDOWN



Reference: **SysML role in Model Driven Requirements Engineering (MoDRE) & MBSE**, Kyle Post and George Walley, 2013

Connecting SysML Models to other models and data

Goals

1. Improve efficiency in writing & managing requirements
   Connect to and reuse requirements from corporate repositories in SysML model elements & synchronize updates

2. Associate parts & assemblies with the functions they deliver
   Allocate functions and physical parts, which are managed in corporate repositories, in a visually oriented way

3. Realize new relationships and improve reporting & efficiency
   Connect data sources that aren't currently integrated via SysML models to reduce redundant or stale data, ease access to information for engineers, and improve reporting

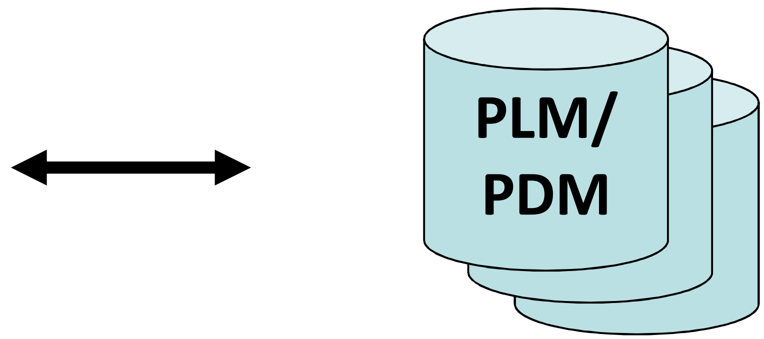
4. Improve requirements and design work
   Tie requirements to specific instances of real and simulation data points and data sets to a requirement's context

PLM System Integration

- As we began expanding our use of SysML models, we quickly found that SysML was not where we wanted to manage or even author details of requirements

- The duplication in effort required to do initial requirements work in SysML and then transfer and manually synchronize them with corporate requirements repositories was non-value-add and cast a limiting view on the potential use for such models.

- In 2012 & 2013 we worked with InterCAX to begin exploring connections to some internal data systems, including PLM systems like Teamcenter[2], where requirements, signal databases, and traditional CAD data/parts are managed



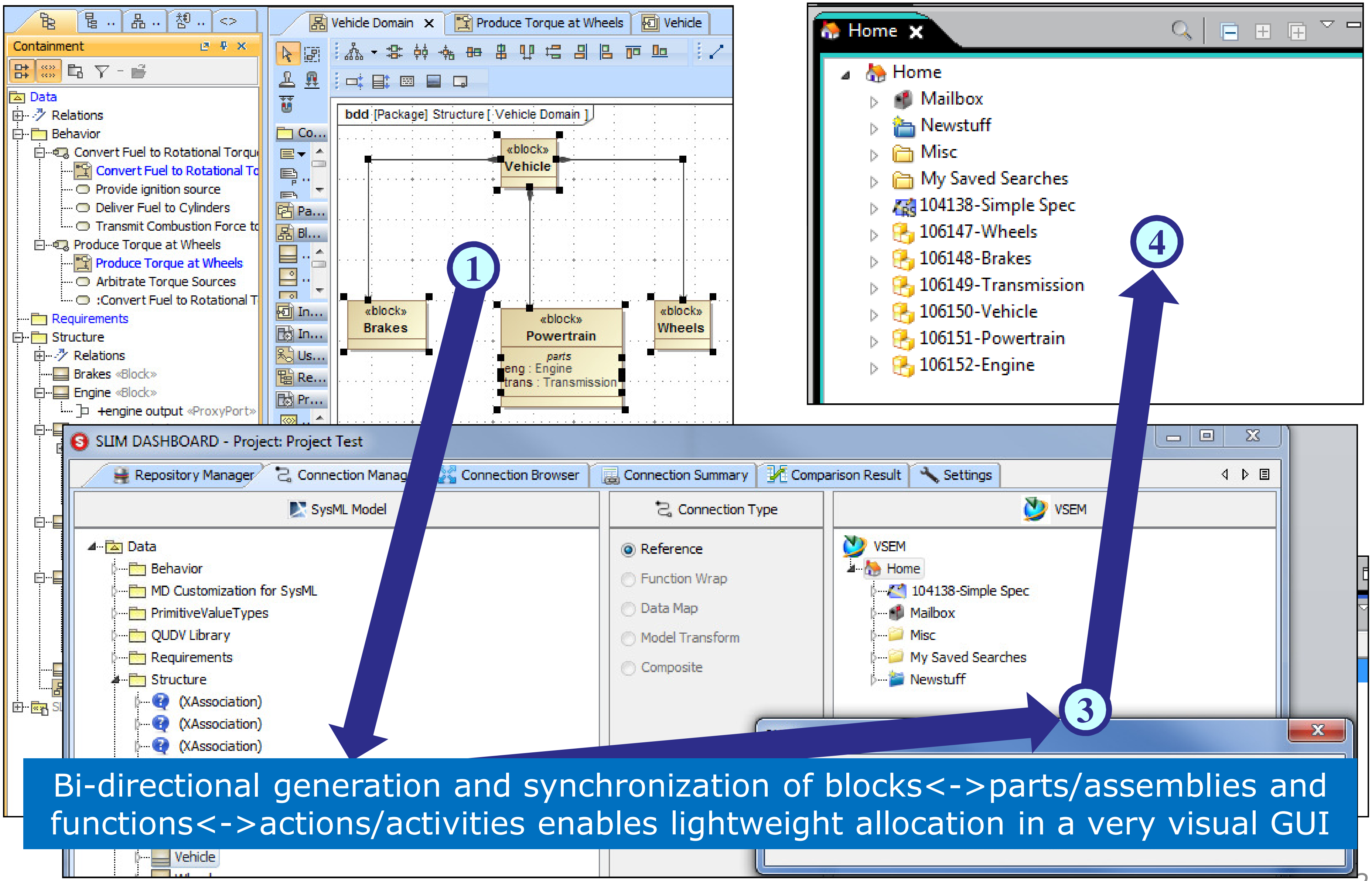
[1] *OMG SysML is a trademark of The Object Management Group*

[2] *Teamcenter is a trademark of Siemens Product Lifecycle Management Software Inc.*

PLM System Integration Examples

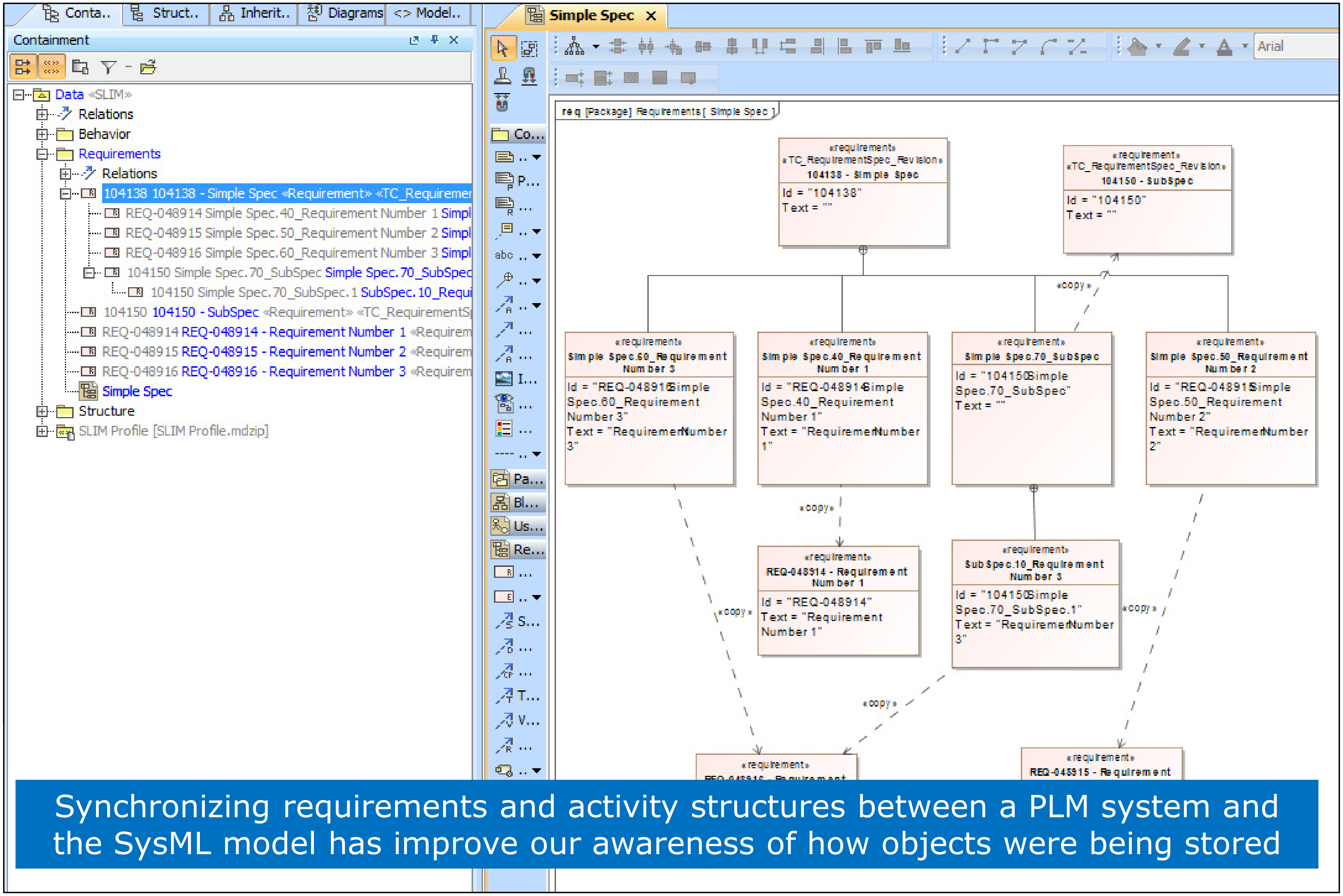
SysML Model

PLM System



Bi-directional generation and synchronization of blocks<->parts/assemblies and functions<->actions/activities enables lightweight allocation in a very visual GUI

PLM System Integration Examples



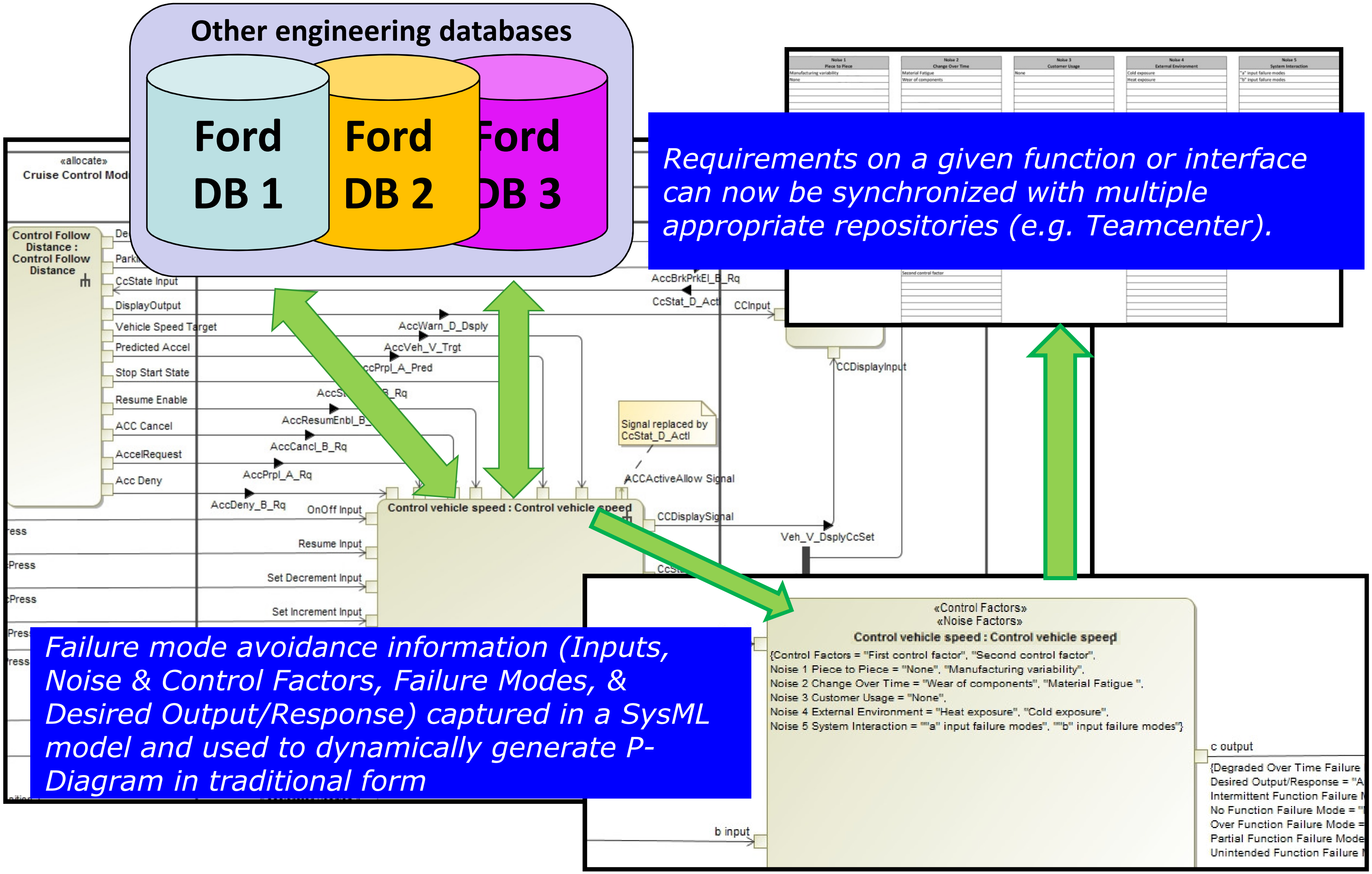
Synchronizing requirements and activity structures between a PLM system and the SysML model has improve our awareness of how objects were being stored

PLM Integration Lessons Learned

- Opportunities:
  - Reuse (and potentially storage) of shared requirements, parts, functions, etc…
  - Provide standards based, visually oriented environment for linking and displaying PLM-managed elements like requirements, functions, and parts

- Challenges:
  - Traditional physical CAD / parts & part assemblies data model doesn't naturally mesh with how certain SysML elements are defined and stored
  - Determining where the SysML model provides a richer managing environment and vice-versa
  - Vendor tool and language support (model transformations, model management, APIs, import/export mechanisms)

Model-Based Failure Mode Avoidance (MBFMA)



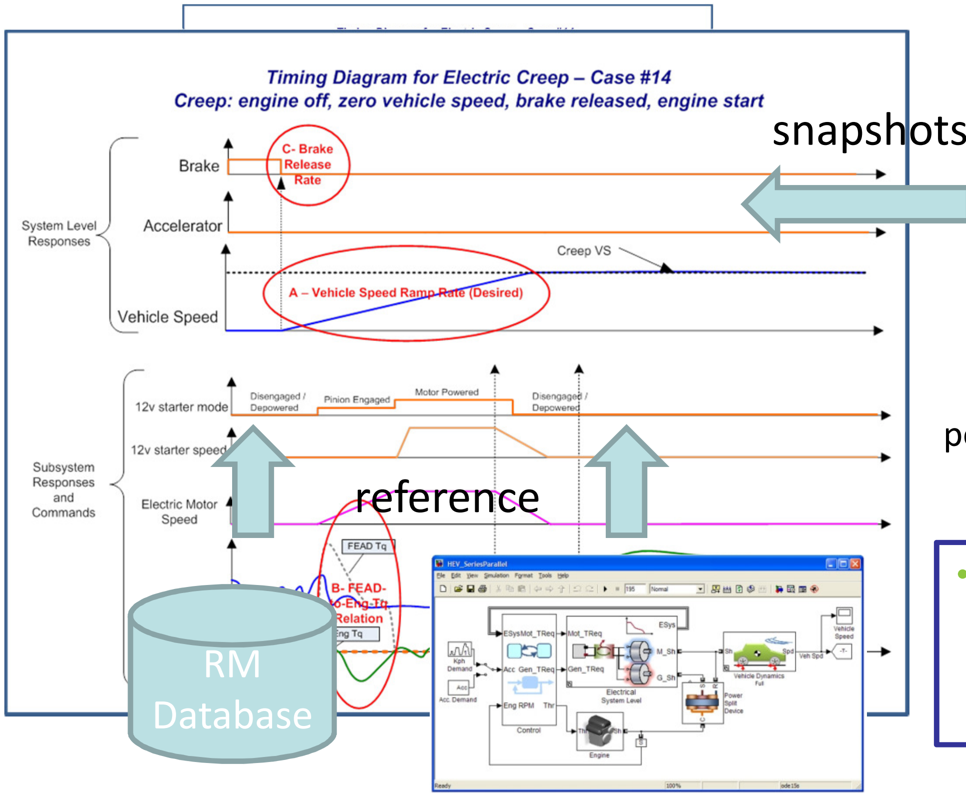
Requirements on a given function or interface can now be synchronized with multiple appropriate repositories (e.g. Teamcenter).

Failure mode avoidance information (Inputs, Noise & Control Factors, Failure Modes, & Desired Output/Response) captured in a SysML model and used to dynamically generate P-Diagram in traditional form

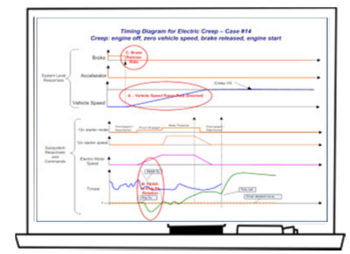
Example from an Adaptive Cruise Control System model

Connecting Descriptive & Analytical Models: Preserving Context

Response plots are great for collaboration and requirements discovery & elicitation, stacking system responses to be viewed in context to each other



or



SysML model preserves context of requirements and permits dynamic linking of requirements , regions, & models

- Principle challenges of current methods:
  - Static Plots - changes to one don't drive changes in others
  - Static Regions - Call-outs have no underlying tie to data points, ranges, systems involved in given traces

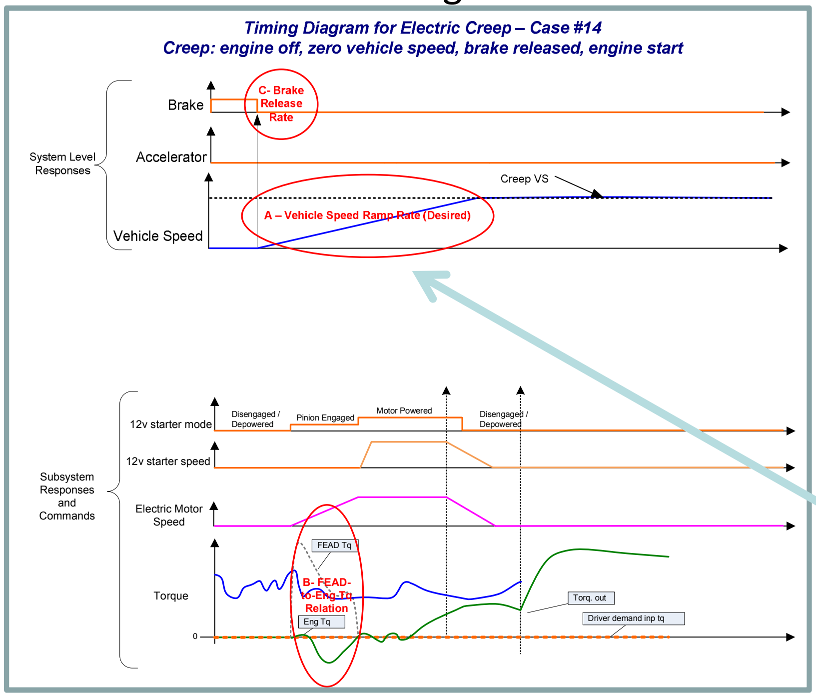
Labeled callouts can be made on either physical or digital (e.g. whiteboard vs. Visio) forms of the diagrams, against which requirements can be written, executable models created, and loose traceability established by manual reference to these call-outs (e.g. "Region A").

Adapted from: G.Walley, *Eliciting Requirements In Context with System Response Plots & SysML*, Engineering Design Conference 2012, May 2012

Pilot Exploration of Response-Based Requirements

The team developed an initial prototype via extension scripts (DXL) in DOORS to import a bitmap of the response diagram from Visio into a top-level DOORS object, extract specific region call-out layer elements, and create/update separate database objects for each (e.g. 3.1 A, 3.2 B, & 3.3 C).

Visio Diagram



DOORS Module

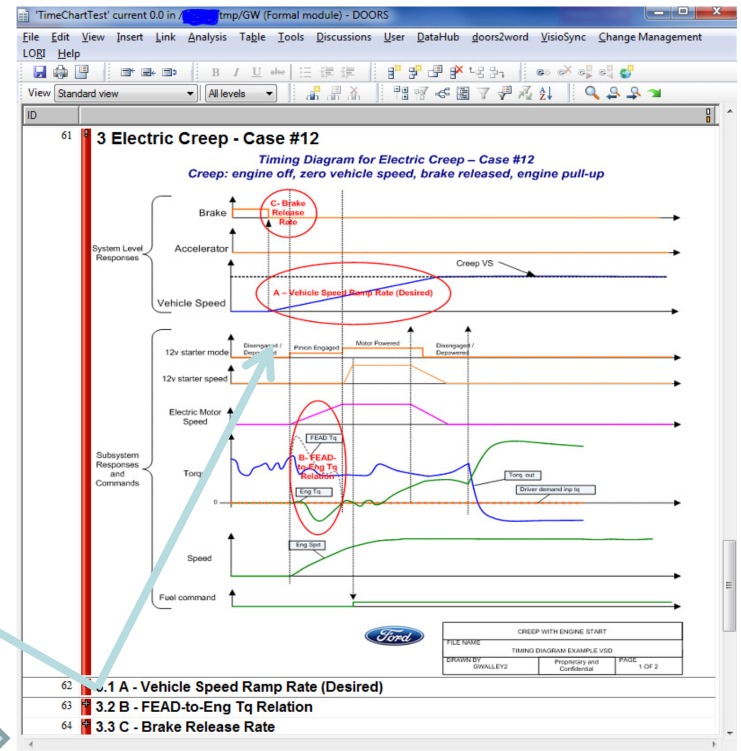


Diagram edited natively in Visio and stored by team on shared resource (share drive, SharePoint, etc)

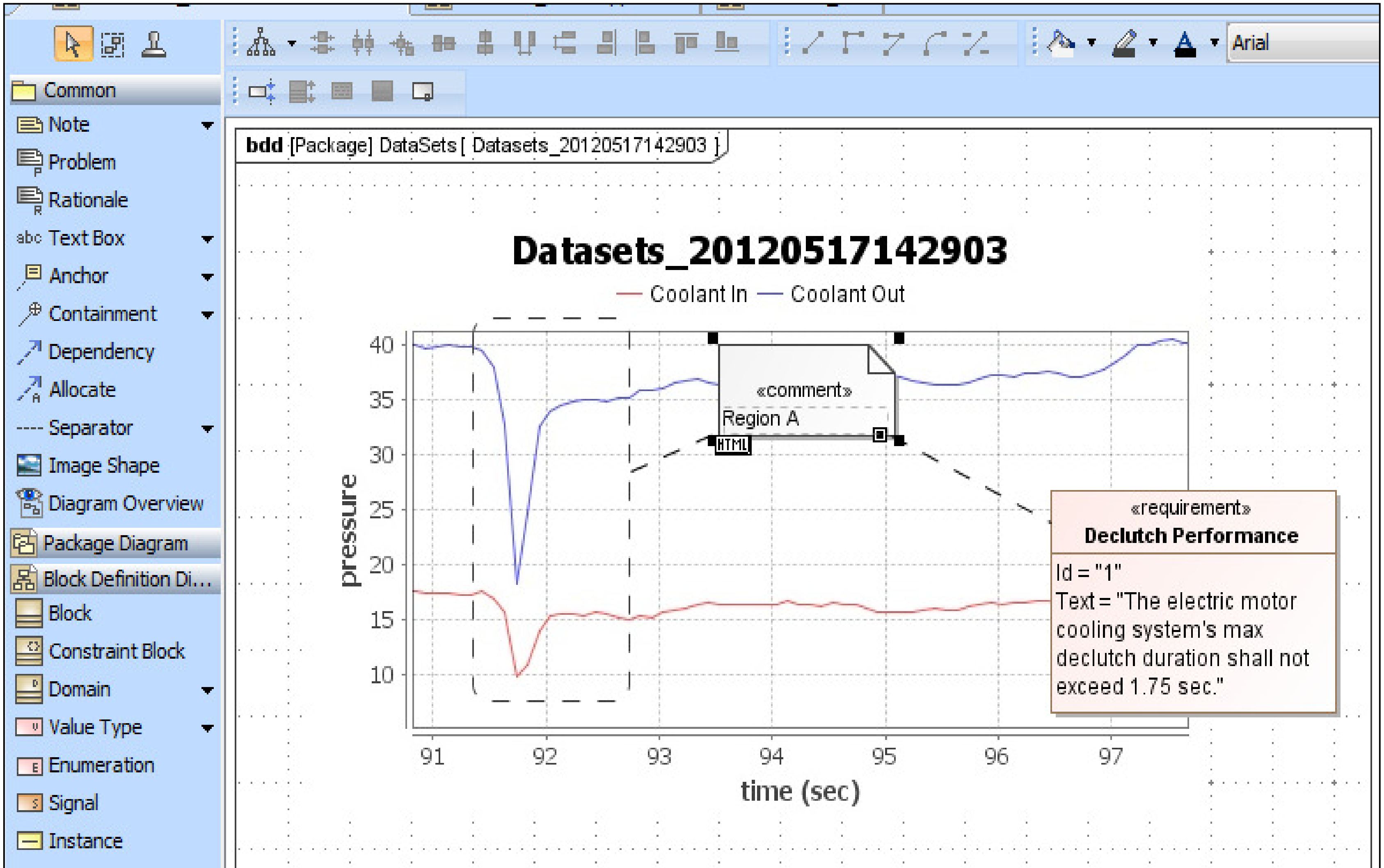
Shared Storage

Updates pulled programmatically upon request or scheduled periodically

Automating the synchronization of familiar-format diagrams with a requirements database, minimized tool changes while enhancing traceability

Source: G.Walley, *Eliciting Requirements In Context with System Response Plots & SysML*, Engineering Design Conference 2012, May 2012

Pilot Exploration of Response-Based Requirements in SysML



Utilizing the call-out notation to link requirements to static of dynamic plots of system responses improves contextual awareness when reading requirements

Source: G.Walley, *Eliciting Requirements In Context with System Response Plots & SysML*, Engineering Design Conference 2012, May 2012

Thank you!

Questions?

Authorization to use OMG SysML logo

Walley, George (G.E.)

From:	Lana Orlova <svetlana@omg.org>
Sent:	Tuesday, January 14, 2014 1:22 PM
To:	Walley, George (G.E.)
Subject:	Re: Request to Use OMG Trademarks
Attachments:	OMG-logo.jpg; OMG-SysML-logo.jpg

Hello,

You have OMG's permission to use the logos (attached). Let me know if you need further assistance.

Best regards,

Lana

At 01:02 PM 1/14/2014, you wrote:

```
Organization_Individual: Ford Motor Company
Description_Licensee:    Automotive OEM
Trademark_Logo:          OMG
SysML logo
Name:
George Edmund Walley III
Title:
Technical Expert, Control Systems Engineering
Work_Phone:
313-594-2912
Email:
gwalley2@ford.com
CODE:
OMG621
B1:
Submit
AGREEMENT:
By clicking on the "Submit" button below, I am agreeing to the
terms of the license set out above.
Remote
Name:
ncfmccx1-ext.nb.ford.com
Remote
User:
HTTP User Agent:
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/31.0.1650.63 Safari/537.36

Proposed_Use:

We would like to use the OMG SysML logo in a presentation by Ford Motor
Company at the 2014 INCOSE International Workshop regarding our growing
use of SysML in conjunction with established analytical and
implementation modeling efforts.
```