Maplesoft Engineering Solutions

PHOENIX INTEGRATION

Model-Driven Innovation

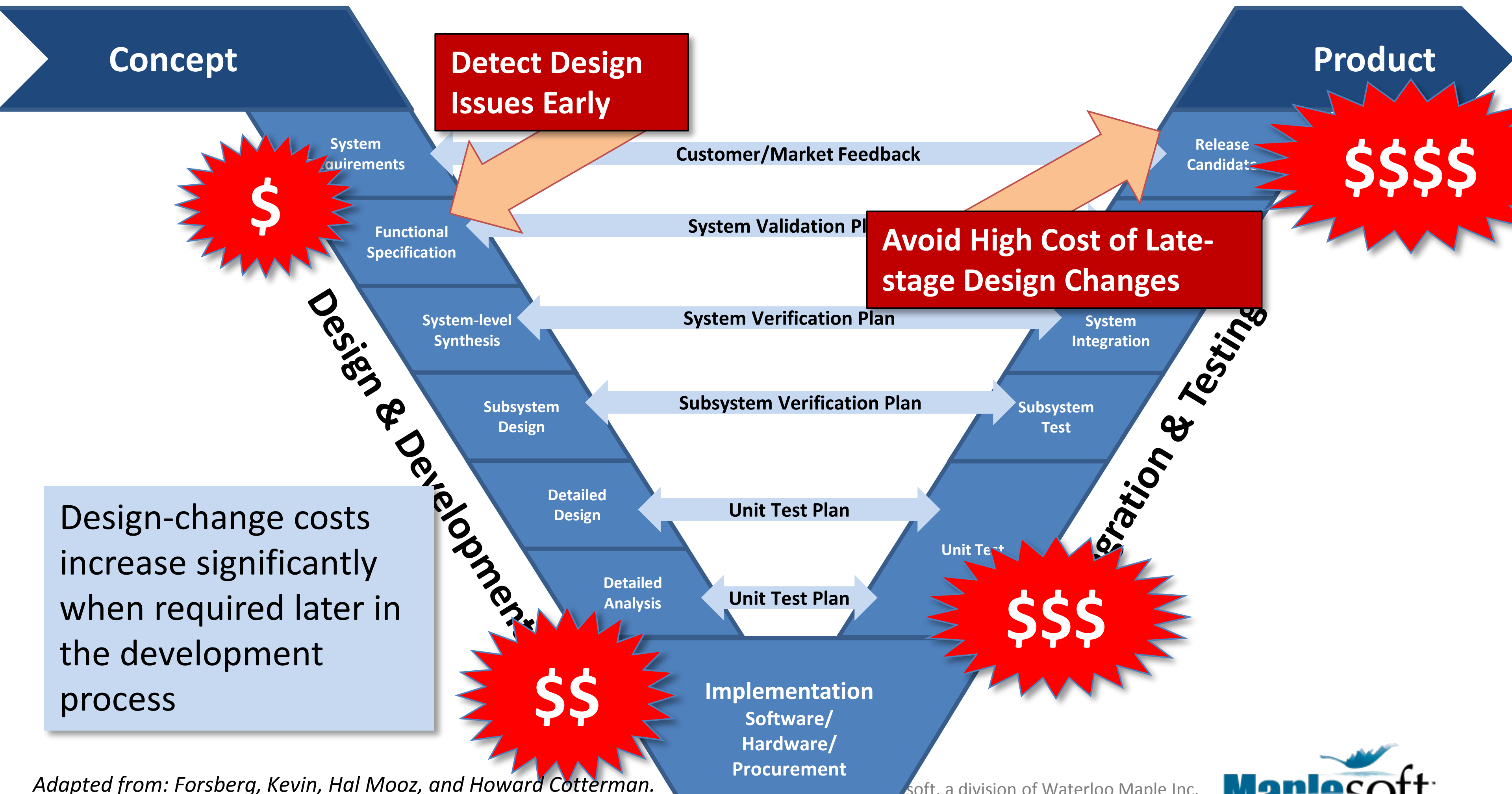
The Role of Multi-domain Dynamic Models for Functional Verification in Model-based Systems Engineering (MBSE)

Paul Goossens, VP, Engineering Solutions, Maplesoft
Joydeep Banerjee, Application Engineer, Maplesoft
Andy Ko, Ph.D., Manager of Engineering Services, Phoenix Integration

NIST MBE Summit 2017, Gaithersburg, MD

Maplesoft
Mathematics • Modeling • Simulation
A Cybernet Group Company

Systems Design & Development Process



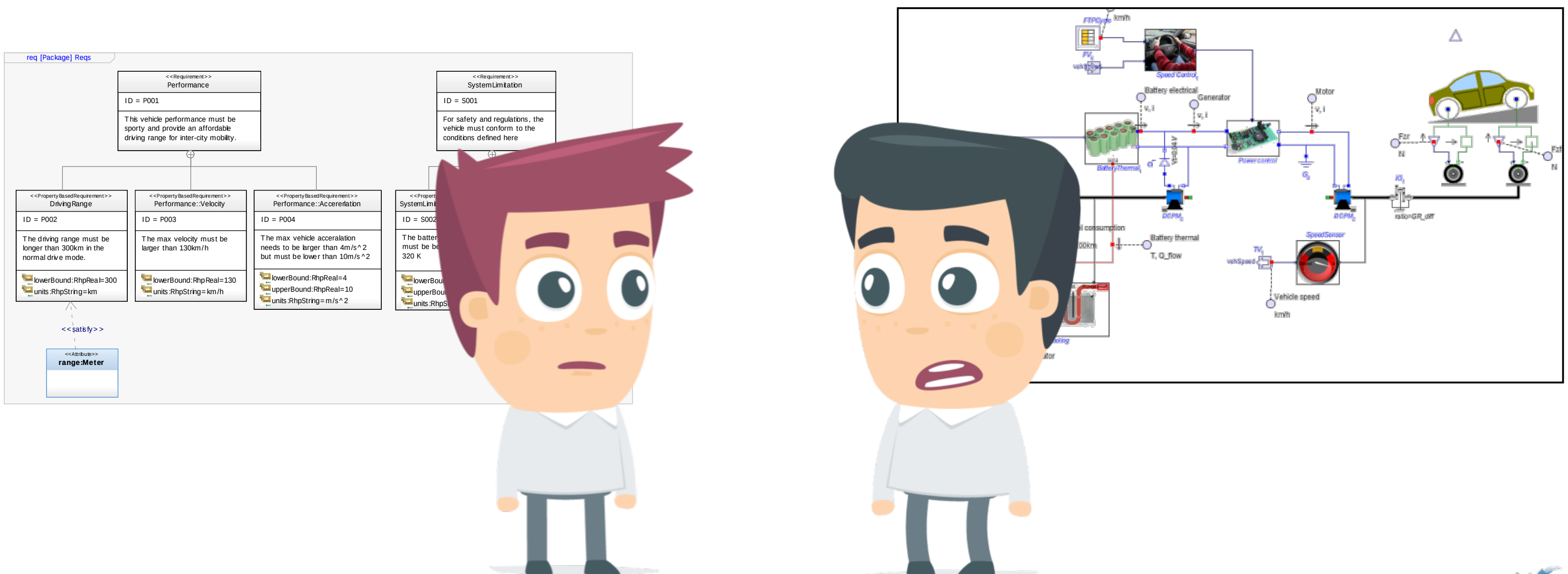
Design-change costs increase significantly when required later in the development process

Adapted from: Forsberg, Kevin, Hal Mooz, and Howard Cotterman. Visualizing Project Management. John Wiley & Sons. Hoboken, NJ. 2005.

Model-based Systems Engineering vs Model-based Engineering

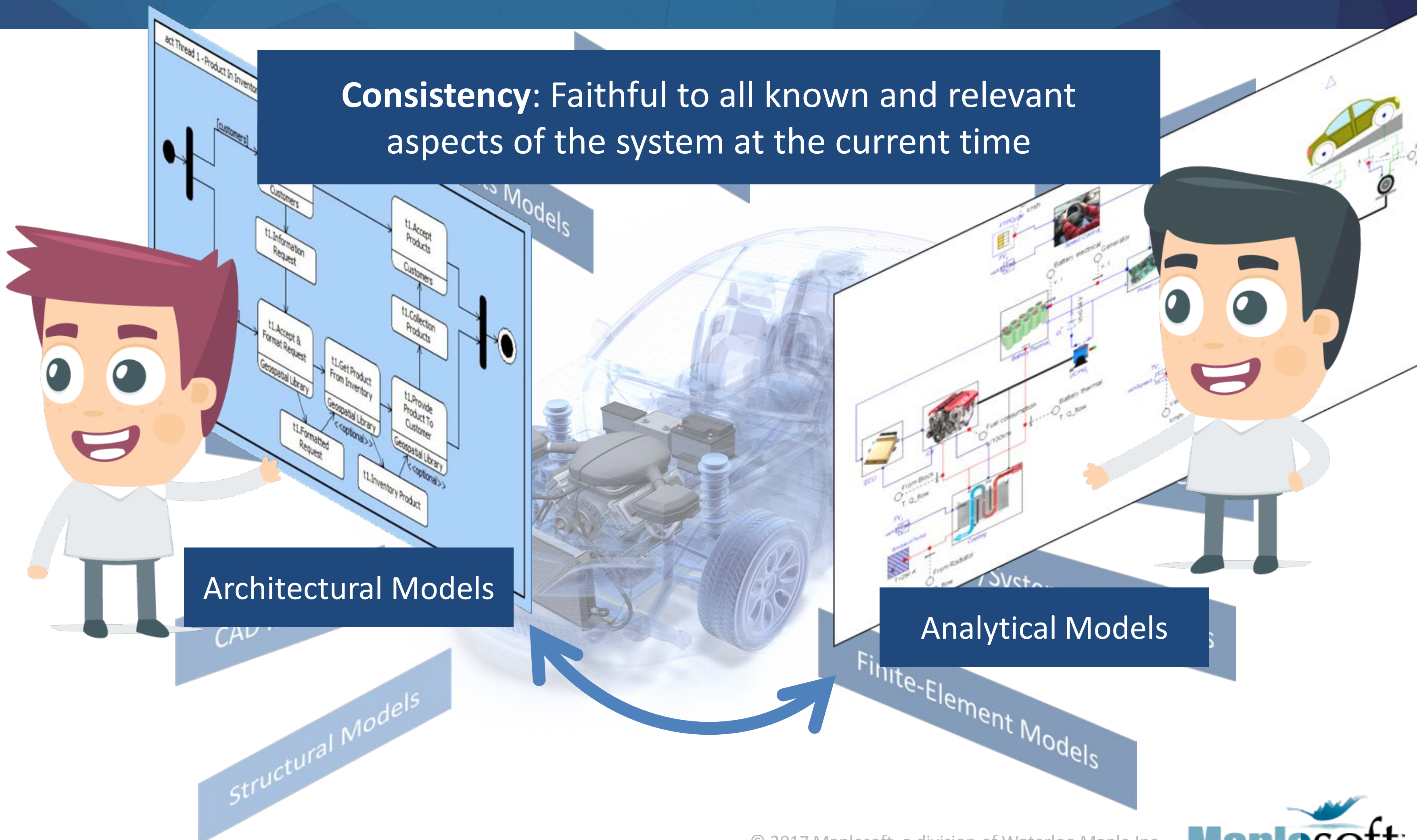
With apologies to George and Ira…

I say "system engineering", you say "systems engineering". …Let's call the whole thing off…



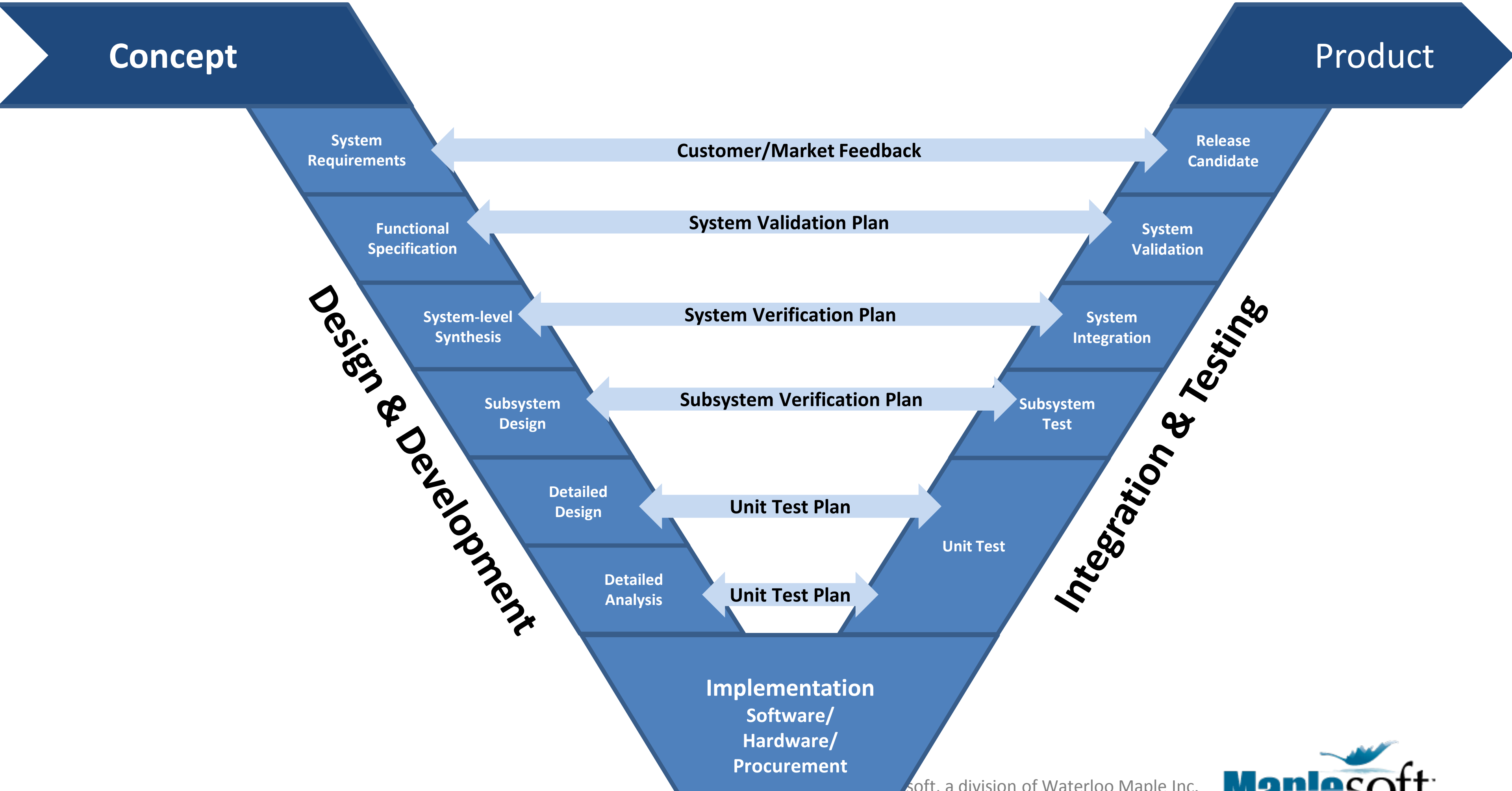
Model-based Systems Engineering vs Model-based Engineering

Consistency: Faithful to all known and relevant aspects of the system at the current time

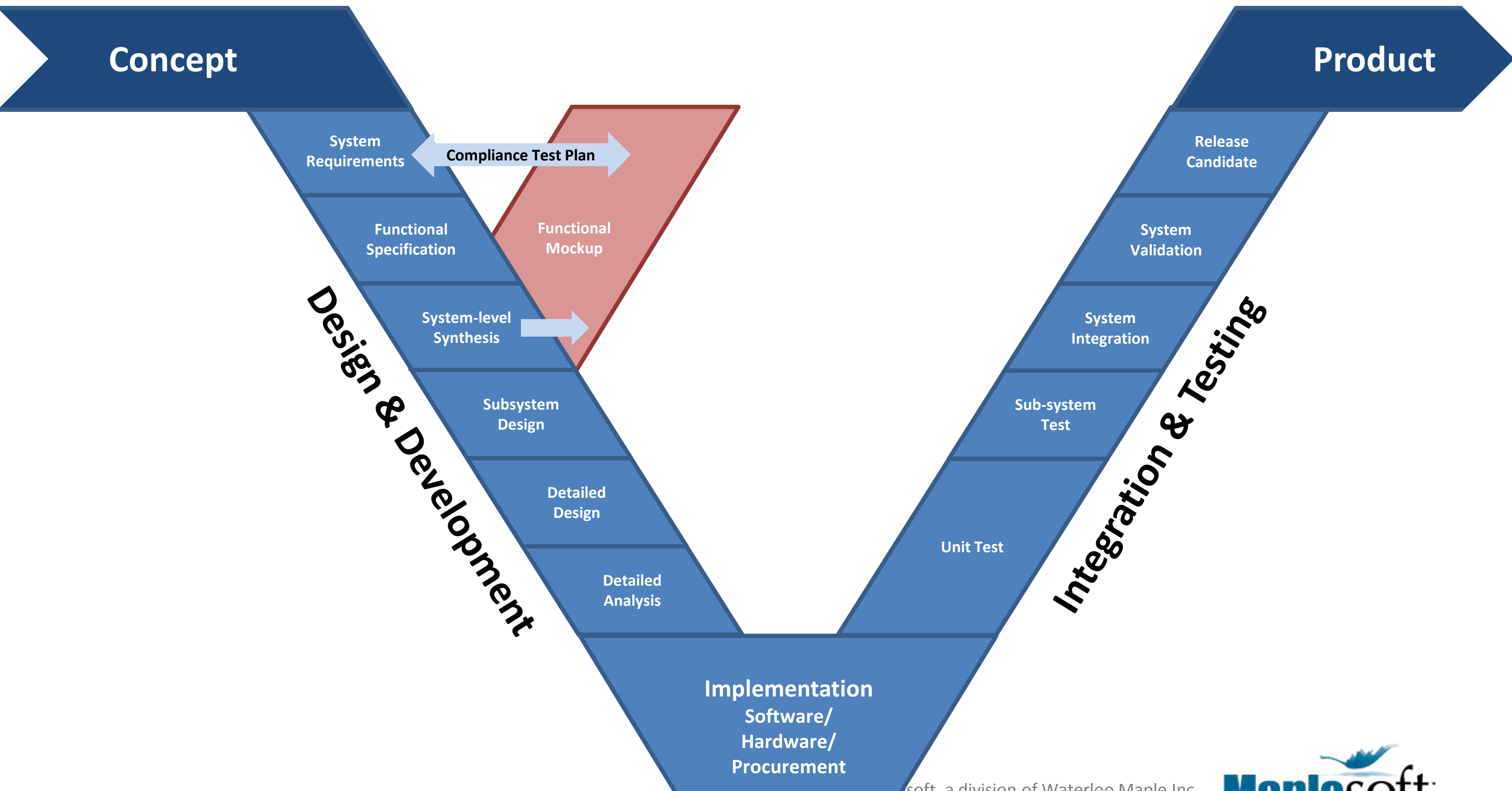


Maplesoft

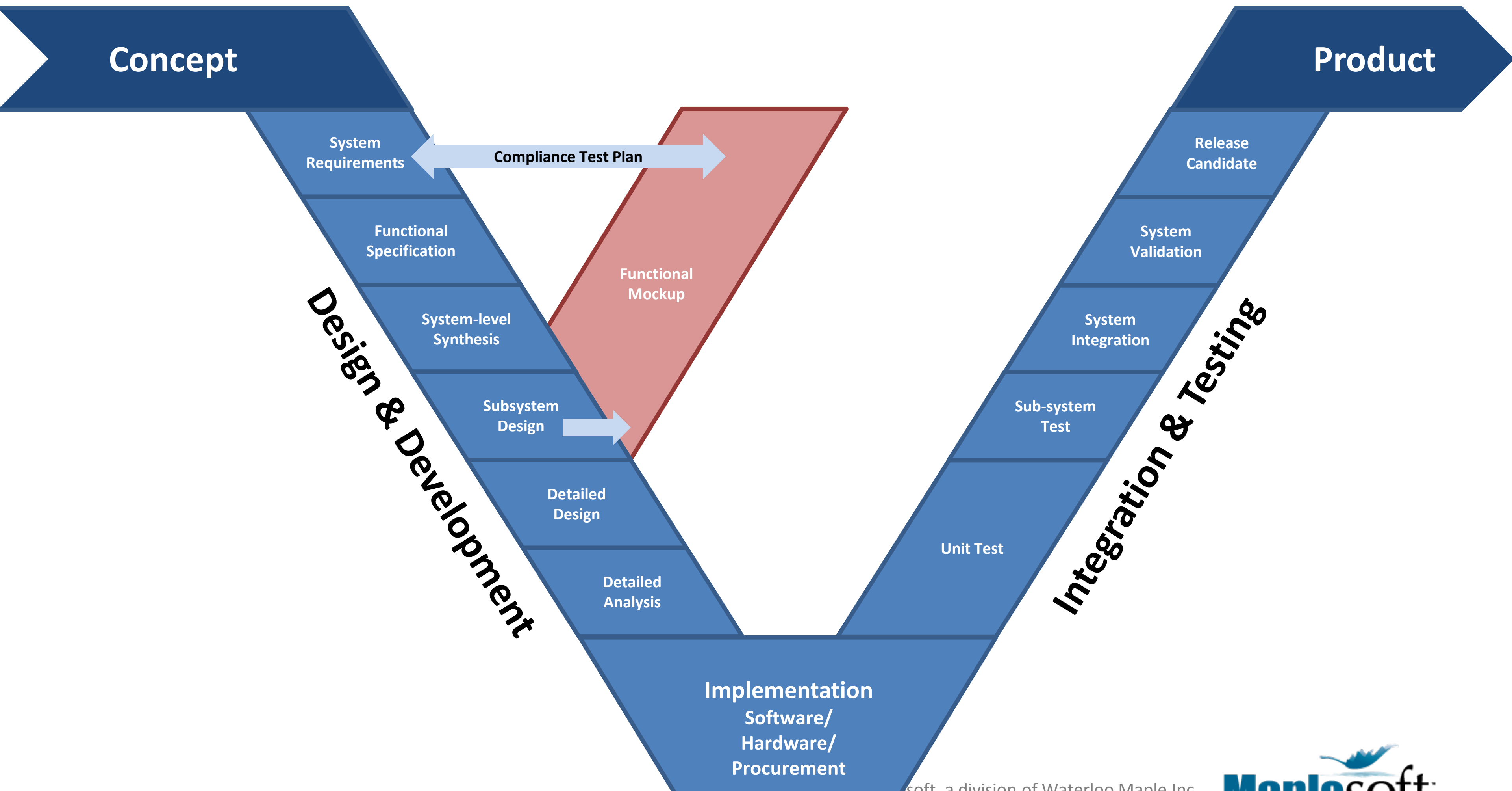
Systems Design & Development Process
Functional Verification



Systems Design & Development Process
Functional Verification



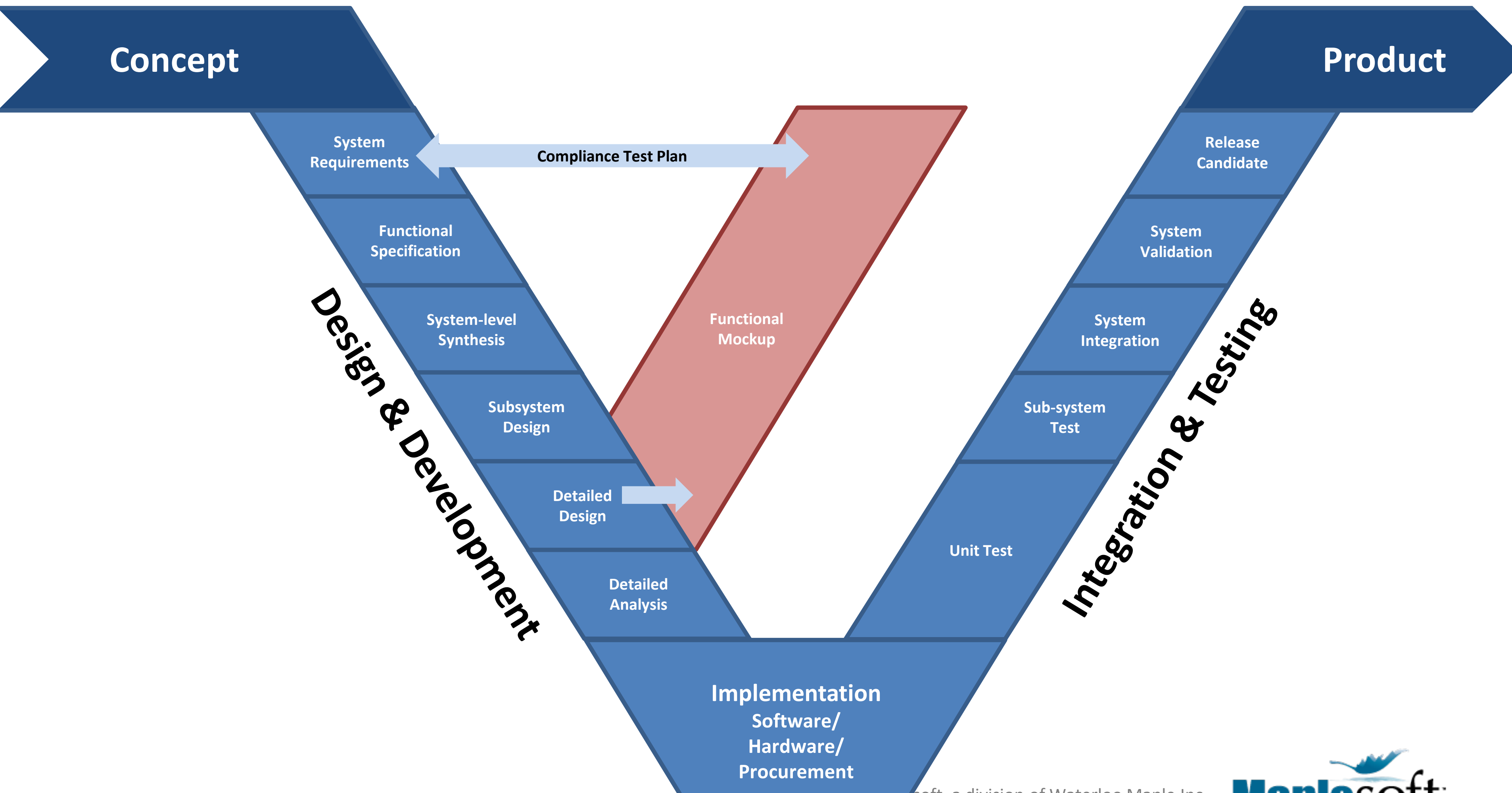
Systems Design & Development Process
Functional Verification



Systems Design & Development Process
Functional Verification

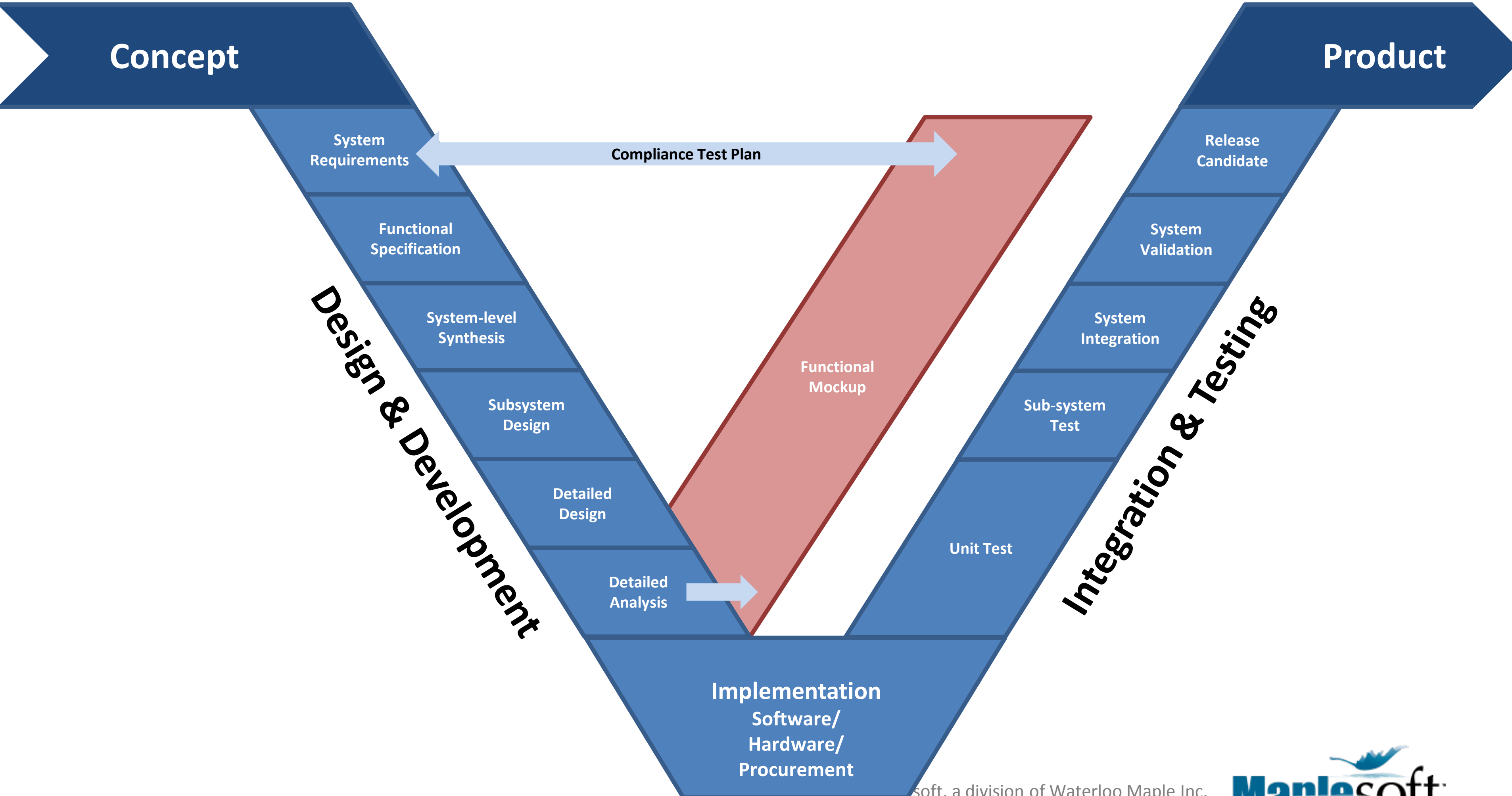
Maplesoft Engineering Solutions

PHOENIX INTEGRATION

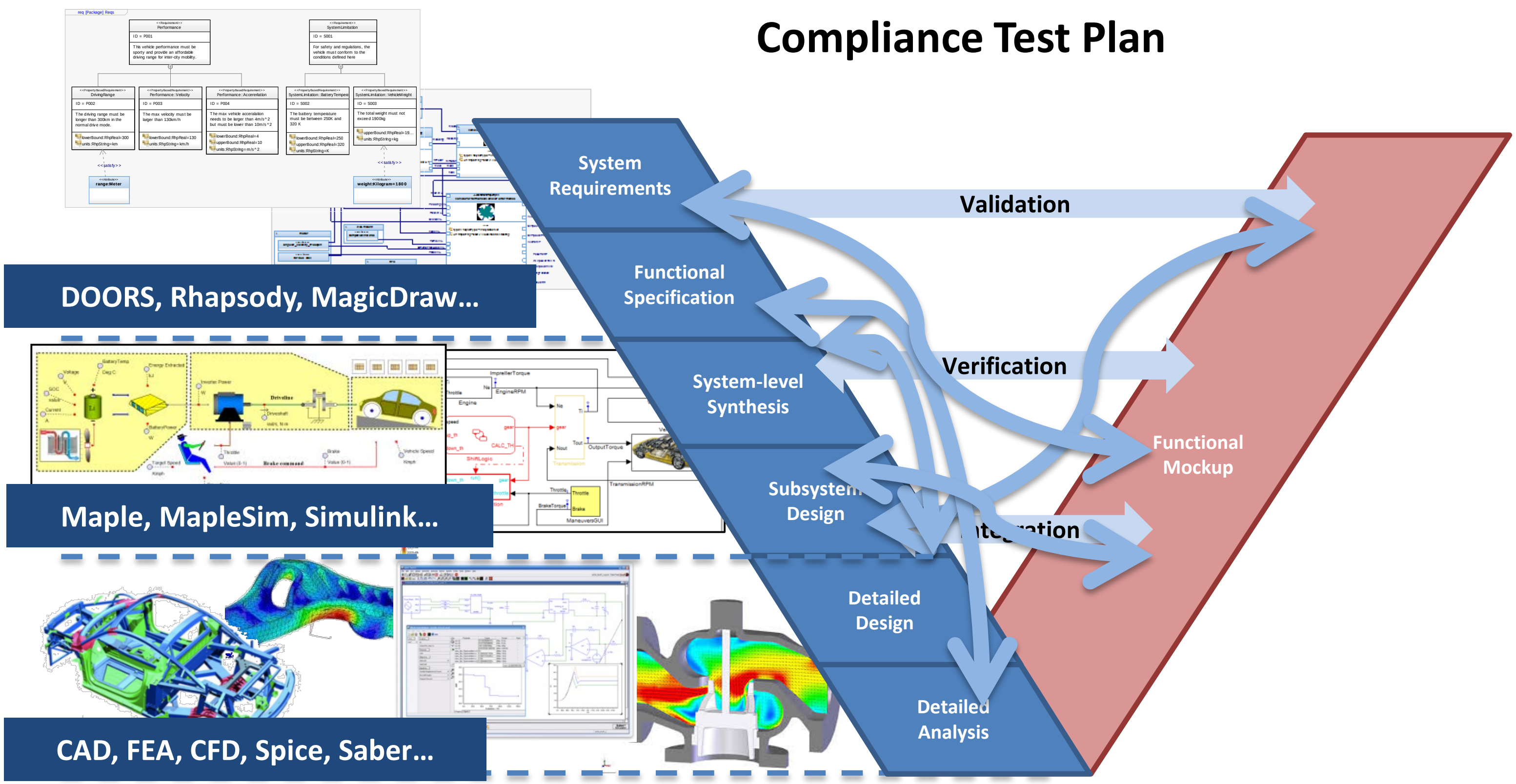


Maplesoft

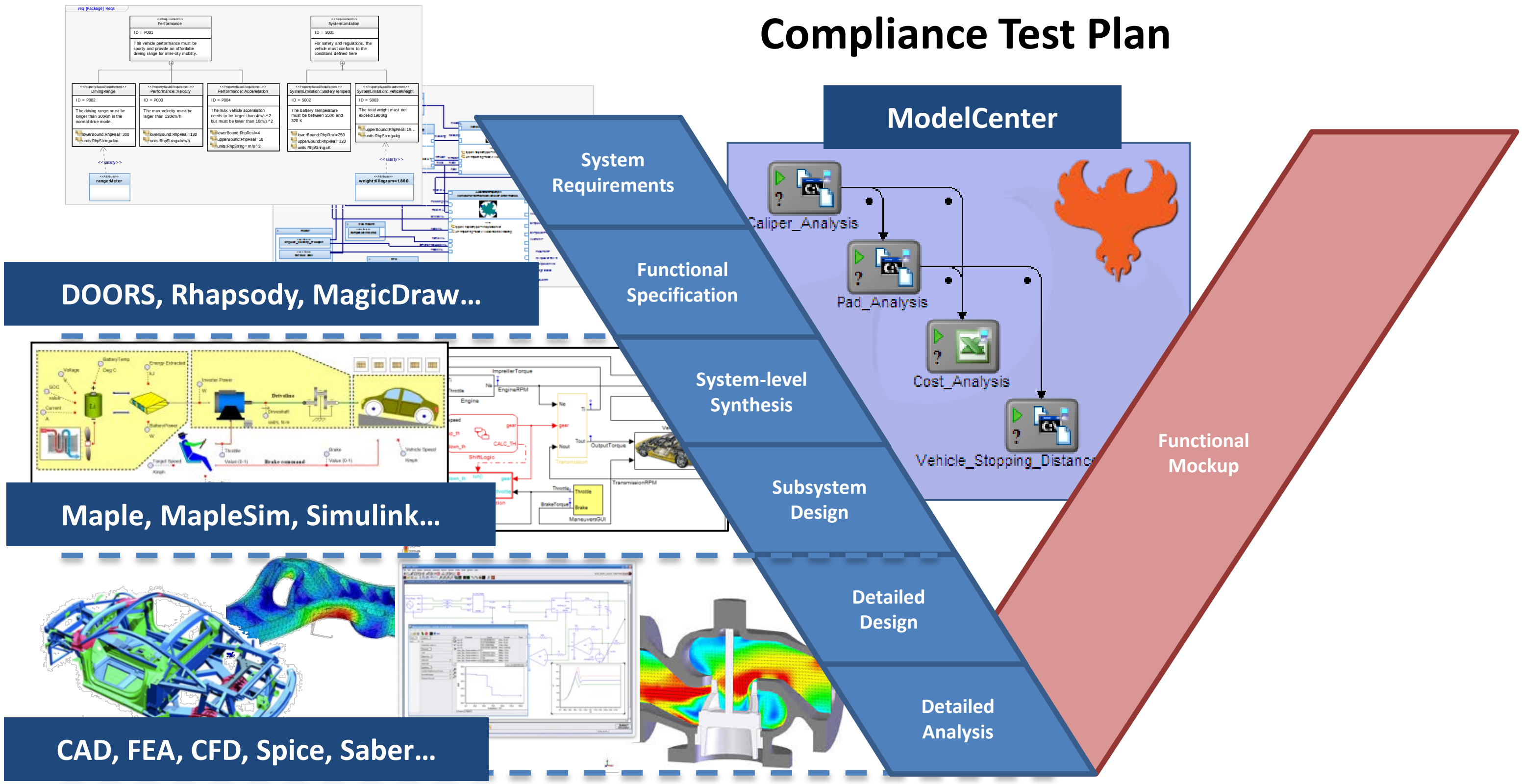
Systems Design & Development Process
Functional Verification



Systems Design & Development Process
Functional Verification



Systems Design & Development Process
Functional Verification

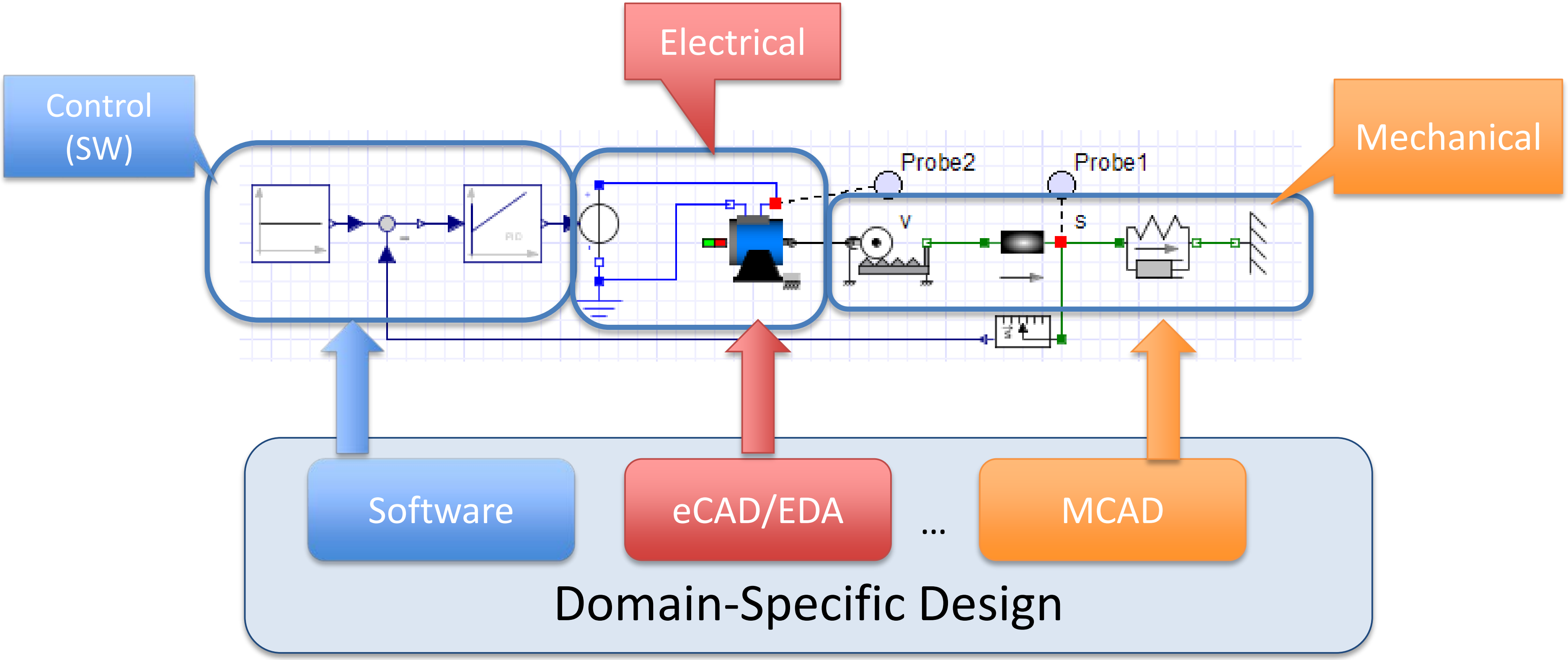


MapleSim

Maplesoft Engineering Solutions

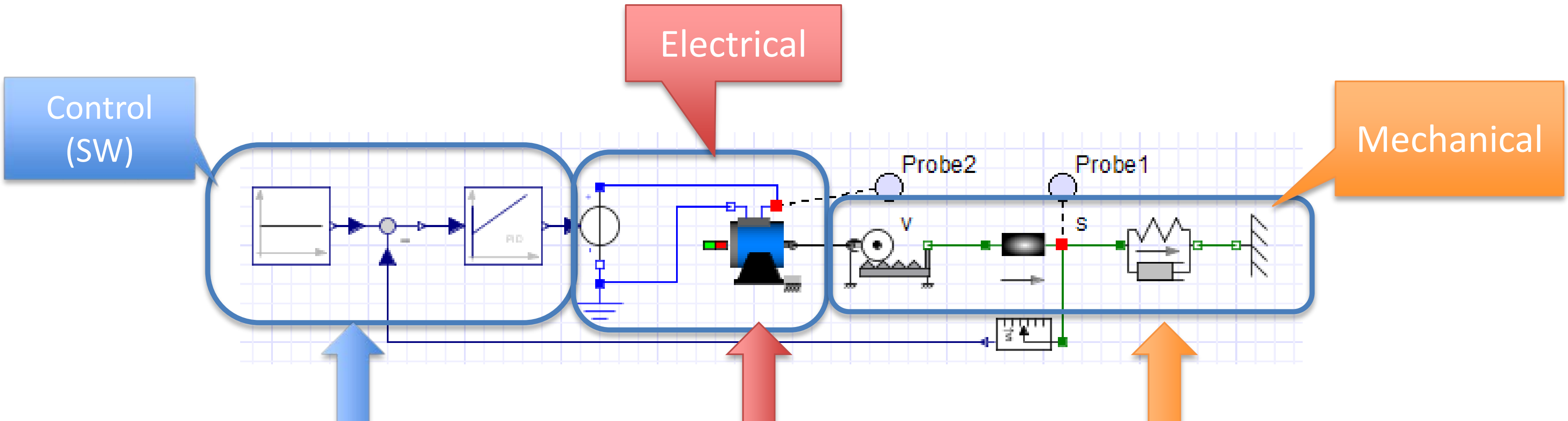
PHOENIX INTEGRATION

Multi-domain Systems Design



Maplesoft

Modelica: Object-oriented Physical Modeling

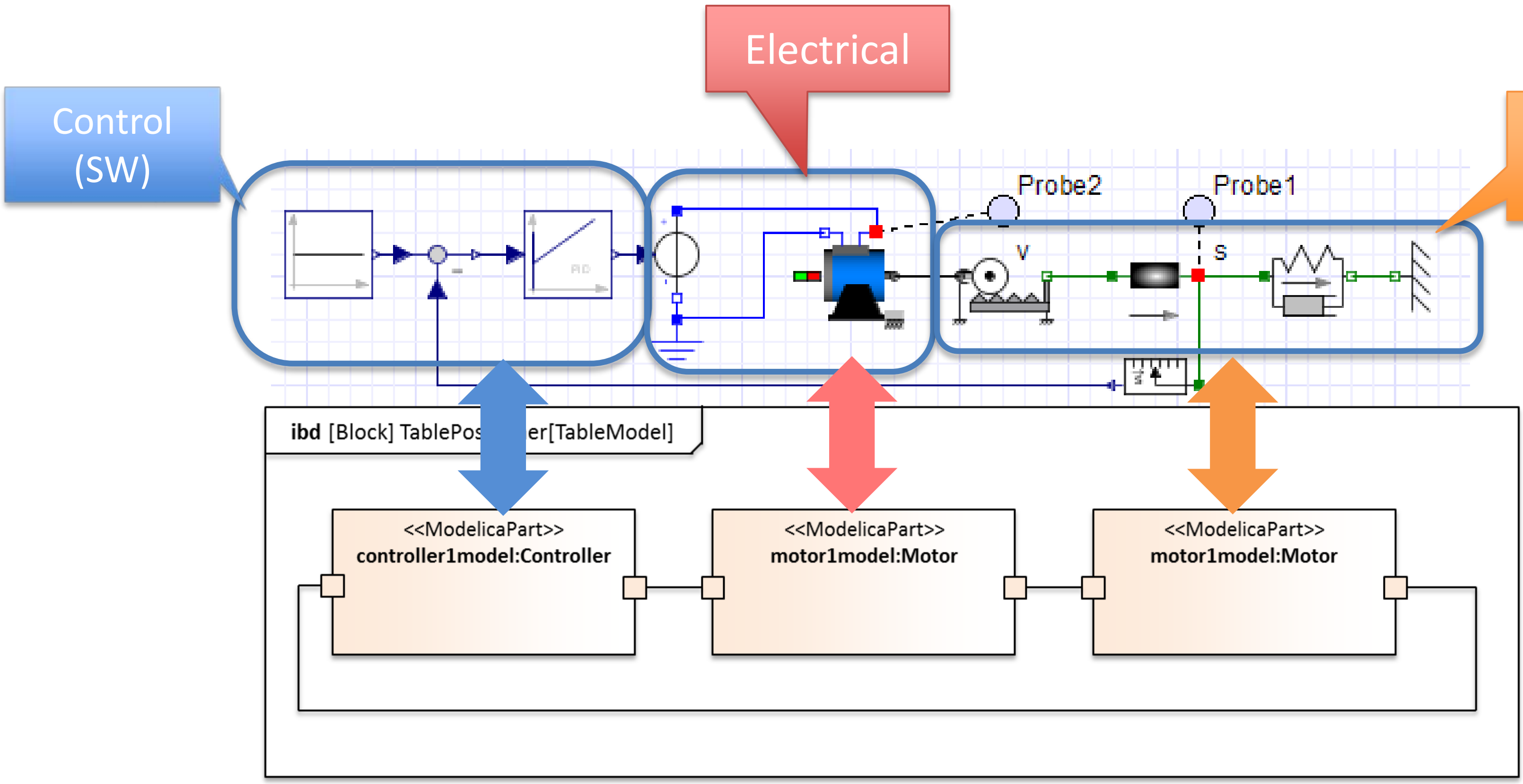



```
model Controller
   import Modelica.Constants.inf ;
   import Pi=Modelica.Constants.pi ;
   import Modelica.Constants.pi ;
   import Maplesoft.Constants.I ;

   public Maplesoft.SignalBlocks.Controllers.PID   PID1(sign
annotation (Placement (transformation (origin={190.0,60.0},ex
   public Maplesoft.SignalBlocks.Math.Operators.Feedback
   public Maplesoft.SignalBlocks.Sources.Real.Step   S1(heig
   public Modelica.Blocks.Interfaces.RealInput   u1 annotatio
   public Modelica.Blocks.Interfaces.RealOutput   y1 annotat
equation
   connect(F1.y, PID1.u) annotation(Line(points={{148.0,60
   connect(S1.y, F1.u1) annotation(Line(points={{92.0,60.0
   connect(PID1.y, y1) annotation(Line(points={{212.0,60.0
   connect(F1.u2, u1) annotation(Line(points={{130.0,44.0
   annotation(
         Diagram(coordinateSystem(preserveAspectRatio =true
         Icon(coordinateSystem(preserveAspectRatio =true,
   );
end Controller;
```

```
model Motor
   import Modelica.Constants.inf ;
   import Pi=Modelica.Constants.pi ;
   import Modelica.Constants.pi ;
   import Maplesoft.Constants.I ;

   public Maplesoft.Mechanics.Rotational.Common.Inertia   I4(J=1,
   public Maplesoft.Electrical.Analog.Sources.Voltage.SignalVoltag
   protected Modelica.Mechanics.Rotational.Interfaces.Flange_a
   public Maplesoft.Electrical.Analog.Passive.Resistors.Resistor
)));
   public Maplesoft.Electrical.Analog.Passive.Inductors.Inductor
   public Maplesoft.Electrical.Analog.Passive.Ground   G2 annotat
   public Maplesoft.Mechanics.Rotational.Sensors.AngleSensor
   public Maplesoft.Mechanics.Rotational.Sensors.RotationalSpeedSe
rotation=0)));
   public Maplesoft.Electrical.Analog.Passive.EMF   EMF2(useSuppo
   public Modelica.Blocks.Interfaces.RealInput   InputSource
   public Modelica.Mechanics.Rotational.Interfaces.Flange_b
   public Modelica.Blocks.Interfaces.RealOutput   PositionOutput
   public Modelica.Blocks.Interfaces.RealOutput   RO3 annotation
equation
   connect(R2.n, I7.p) annotation(Line(points={{128.0,268.0},
   connect(EMF2.n, G2.p) annotation(Line(points={{228.0,158.0
```

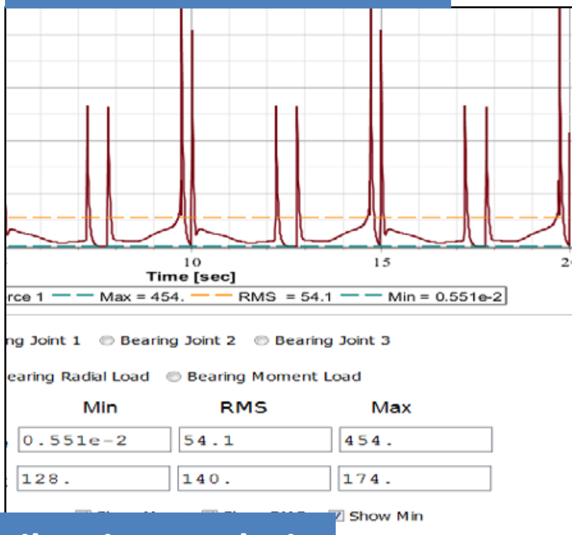
```
model Table
   import Modelica.Constants.inf ;
   import Pi=Modelica.Constants.pi ;
   import Modelica.Constants.pi ;
   import Maplesoft.Constants.I ;

   public Maplesoft.Mechanics.Rotational.Common.Inertia   I3(J=0.
   public Maplesoft.Mechanics.Translational.Common.Mass   SM2(m=2
   public Maplesoft.Mechanics.Rotational.BearingsGears.IdealGearR2T
   public Modelica.Mechanics.Rotational.Interfaces.Flange_a   a1
   public Modelica.Mechanics.Translational.Interfaces.Flange_b   b1
equation
   connect(IGRT2.flangeT, SM2.flange_a) annotation(Line(points=
   connect(I3.flange_b, IGRT2.flangeR) annotation(Line(points=
   connect(SM2.flange_b, b1) annotation(Line(points={{220.0,60.
   connect(I3.flange_a, a1) annotation(Line(points={{50.0,60.0},
   annotation(
         Diagram(coordinateSystem(preserveAspectRatio =true, exte
         Icon(coordinateSystem(preserveAspectRatio =true, extent
   );
end Table;
```

Functional Verification against formal requirements models

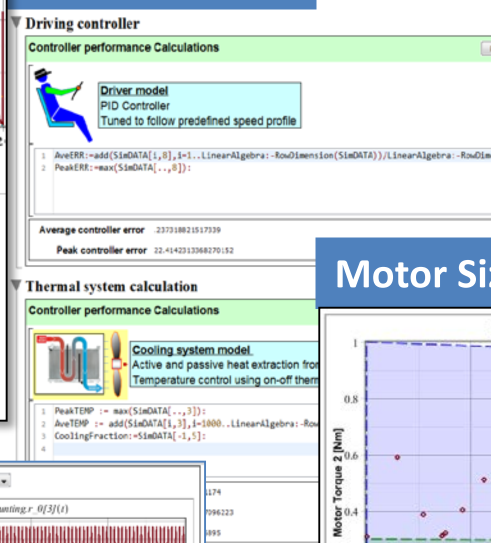


Symbolic Tools for Design-space Exploration

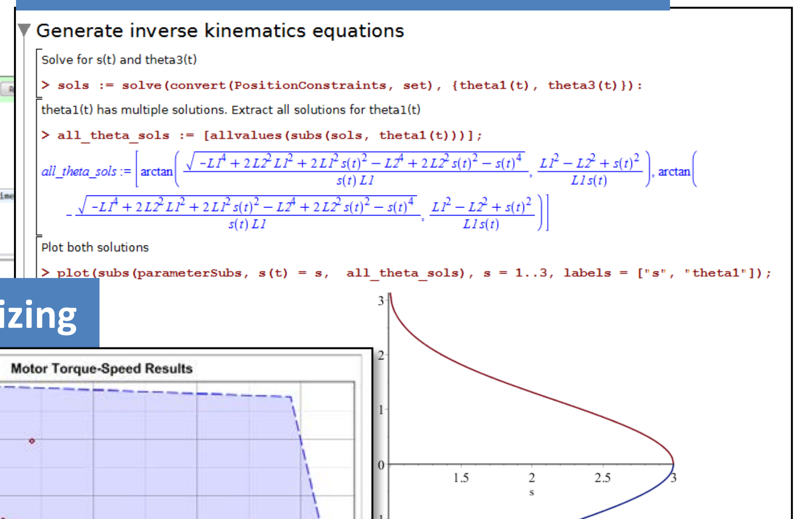
Dynamic Load Analysis



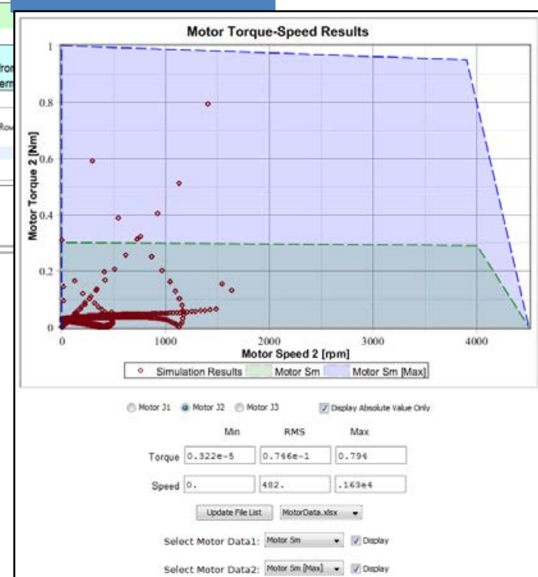
Post-Processing



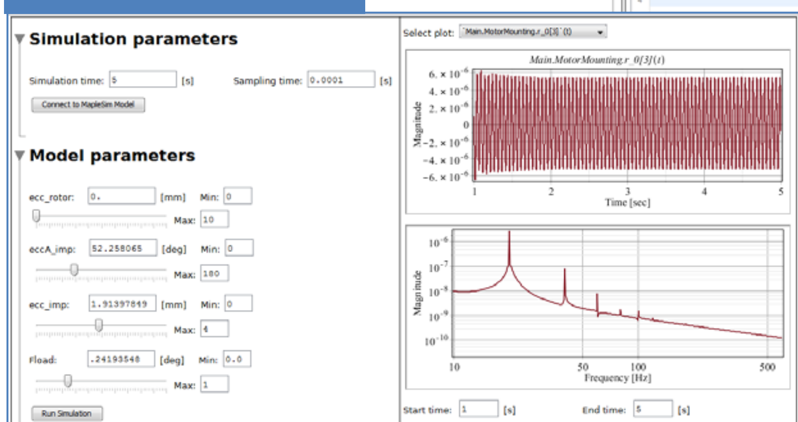
Inverse Kinematics and Dynamics



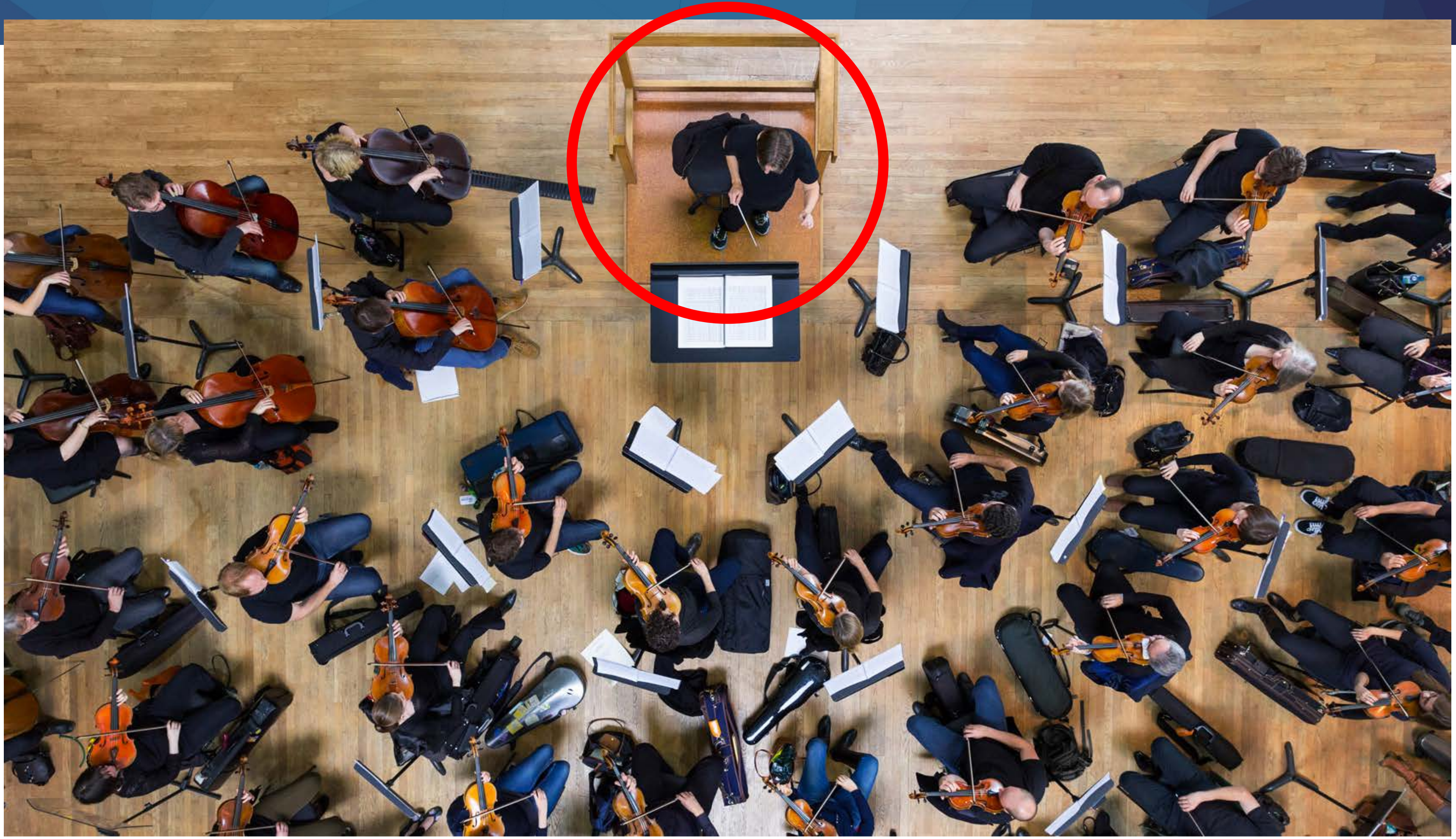
Motor Sizing



Vibration Analysis

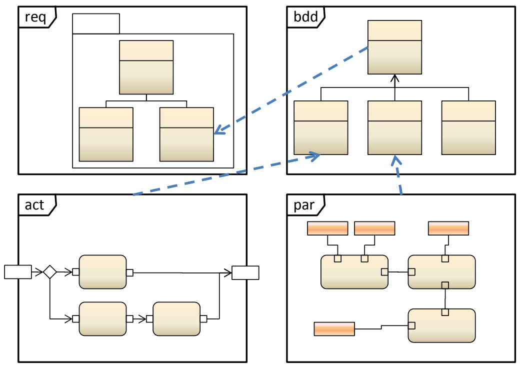


PHX ModelCenter

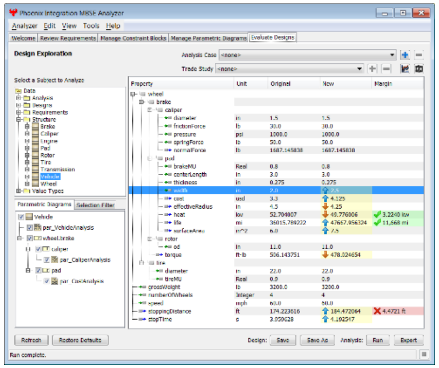


PHX Systems Engineering Integration Module

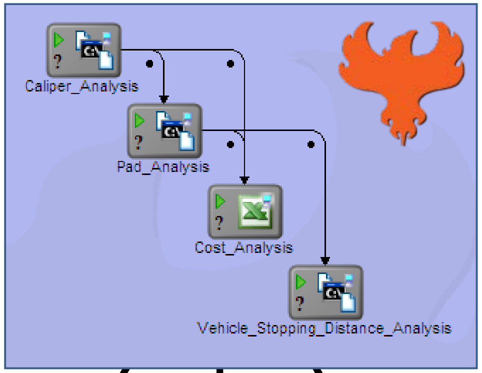
Systems Engineering: Architectural Model

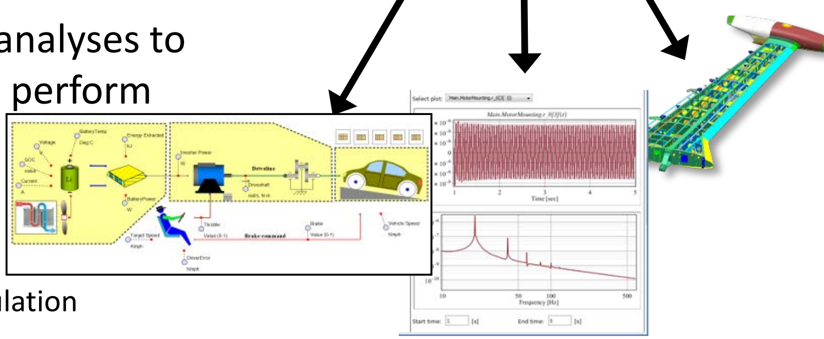


MBSE Analyzer



Domain Engineering: Executable Analysis Model

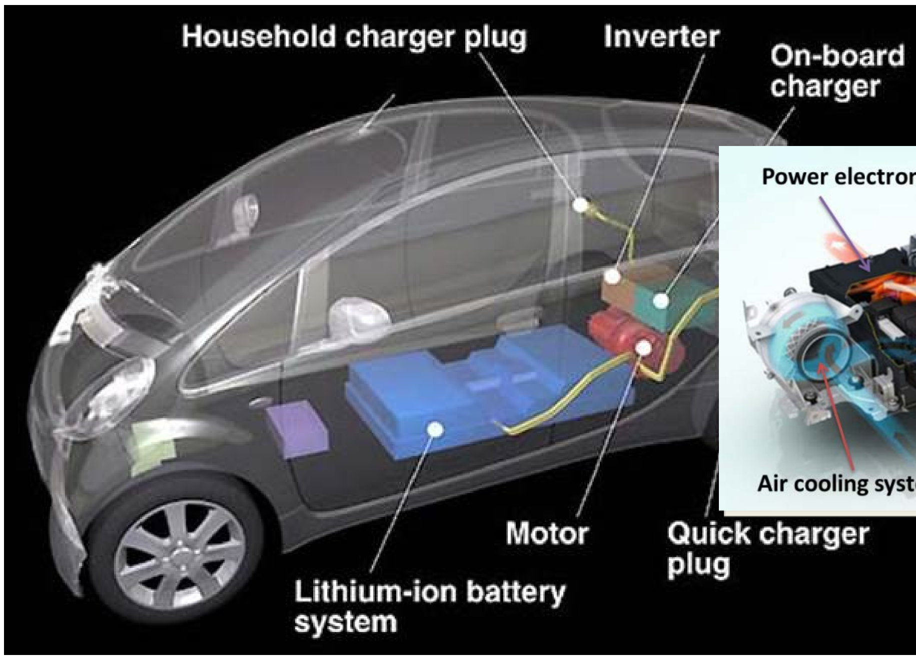


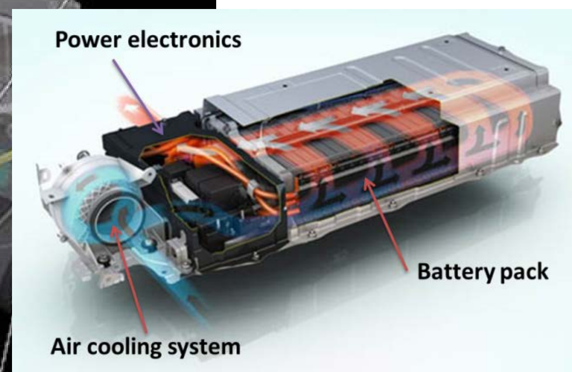


Engineering Analysis

- Connect systems architecture models with engineering analyses to calculate system performance, check requirements, and perform design trade-offs
- Capabilities
  - Execute SysML parametric diagrams to evaluate designs
  - Perform requirements compliance analysis using modeling and simulation
  - Perform design trade-off studies
  - Update SysML models with analysis results
  - Import engineering analyses into a SysML model

Electric Vehicle: Battery System Thermal Performance





Battery Power Electronics and Cooling

Stored Energy

- Battery Capacity
- State-of-Charge
- Affects driving range

Temperature Control

- Heat flow to/from battery
- Thermal effects on battery performance
- Active/passive cooling system

Electric Vehicle: Battery System Thermal Performance

Safety requirements

- Battery must operate in a safe temperature range
- Roll / pitch acceleration must be under a certain target
- Stopping distance should not be more than a specific target

Performance requirements

- Maximum acceleration / speed should be more than designated targets.
- Must be operable within a designated range

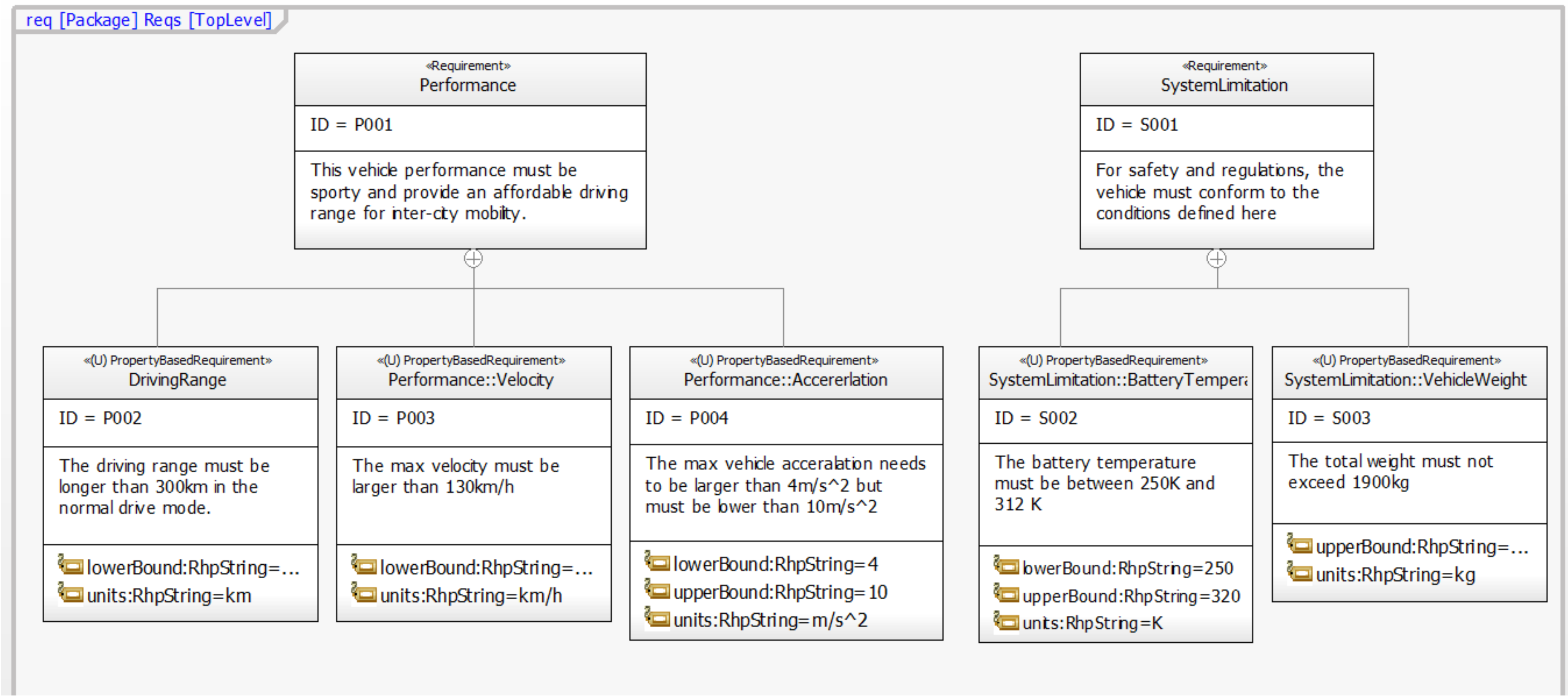
Battery requirements

- Battery mass, energy density
- Max/min operating temperature
- Max/min peak temperature
- Efficiency vs SOC characteristics
- State of Health characteristics

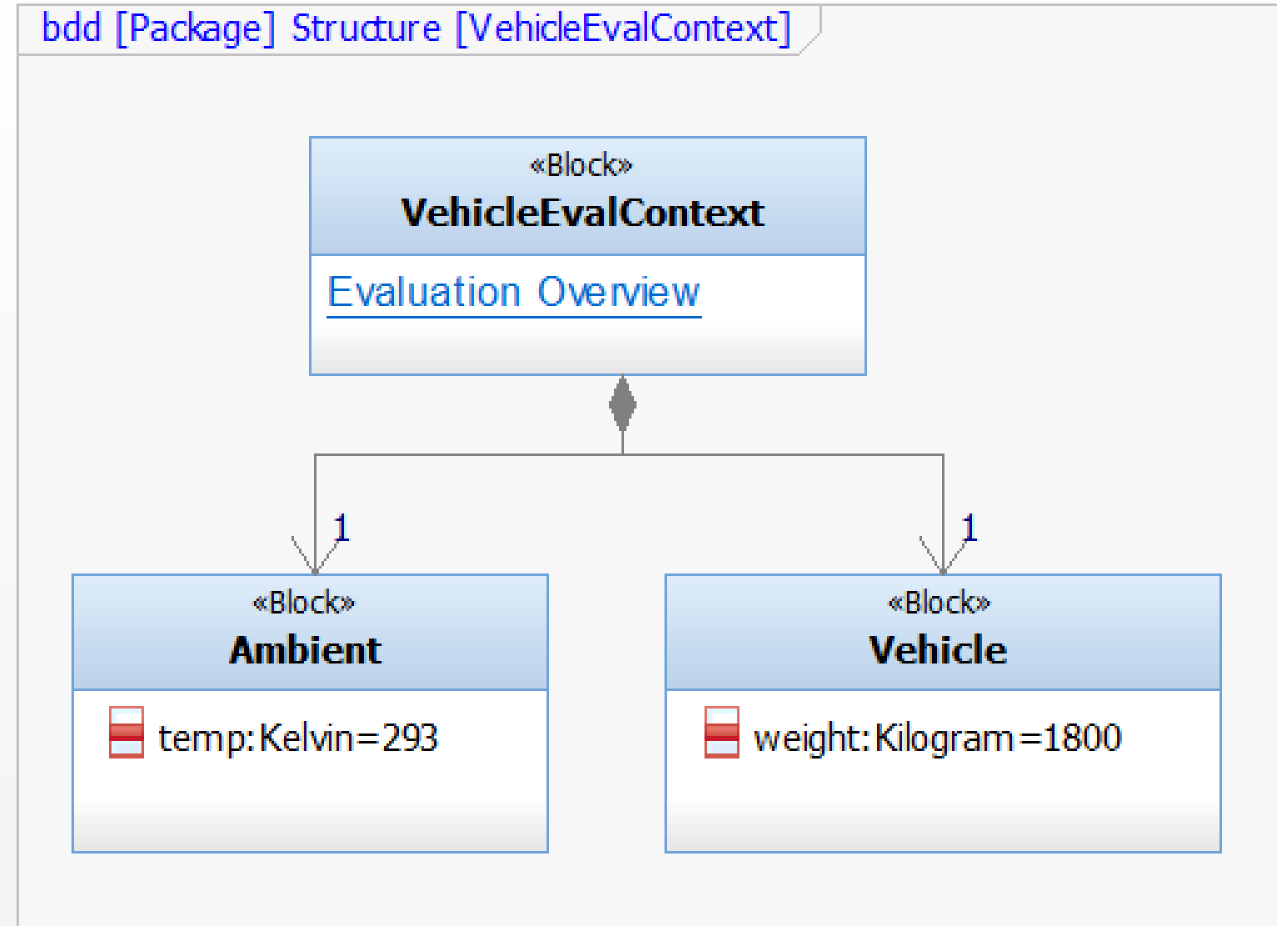
Cooling system requirements

- Heat transfer characteristics
- Maximum heat transfer rate
- Temperature control system
- Multi-component cooling
- Maximum weight of the system
- Critical temperature detection

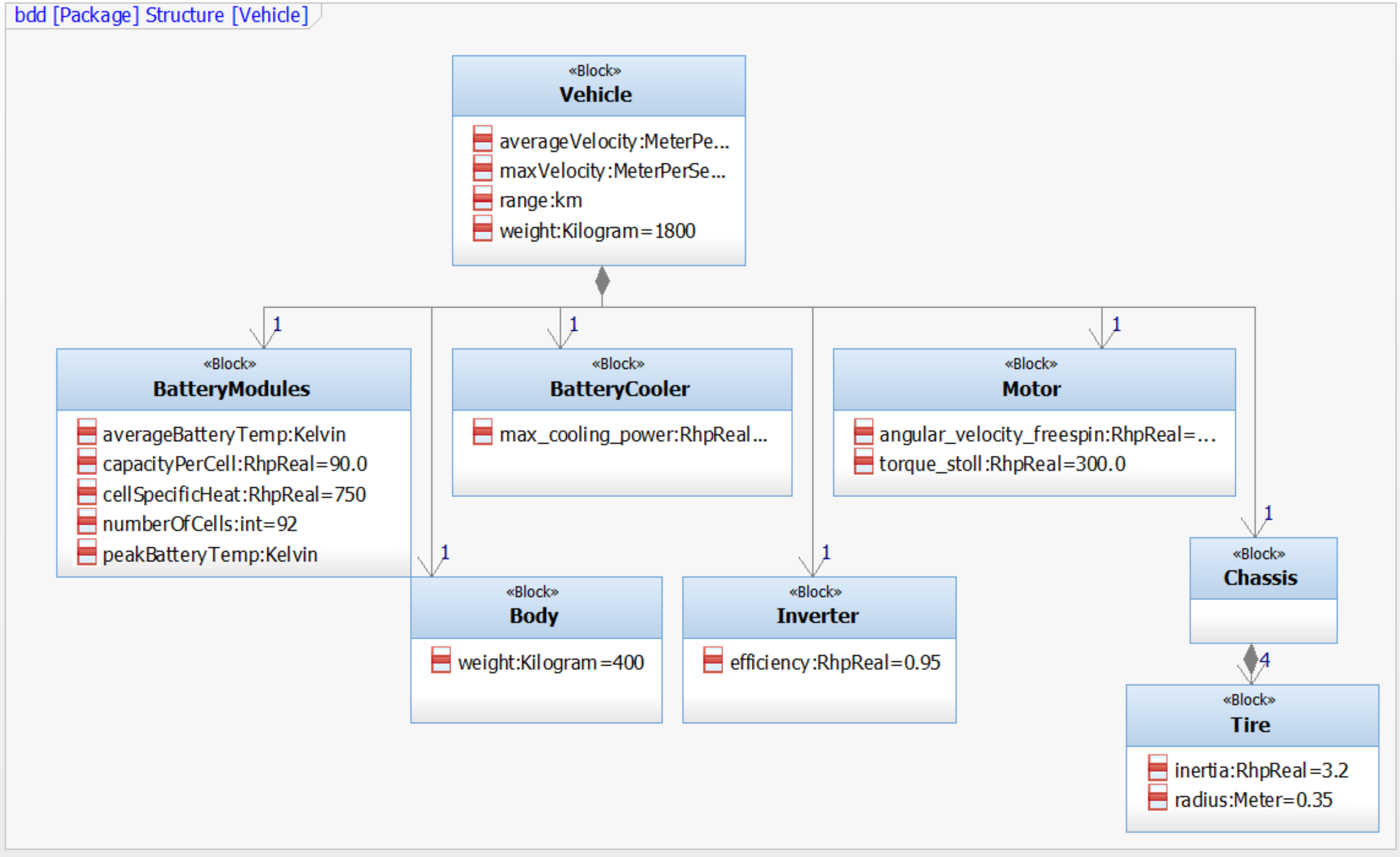
Architectural Model Requirement Diagram



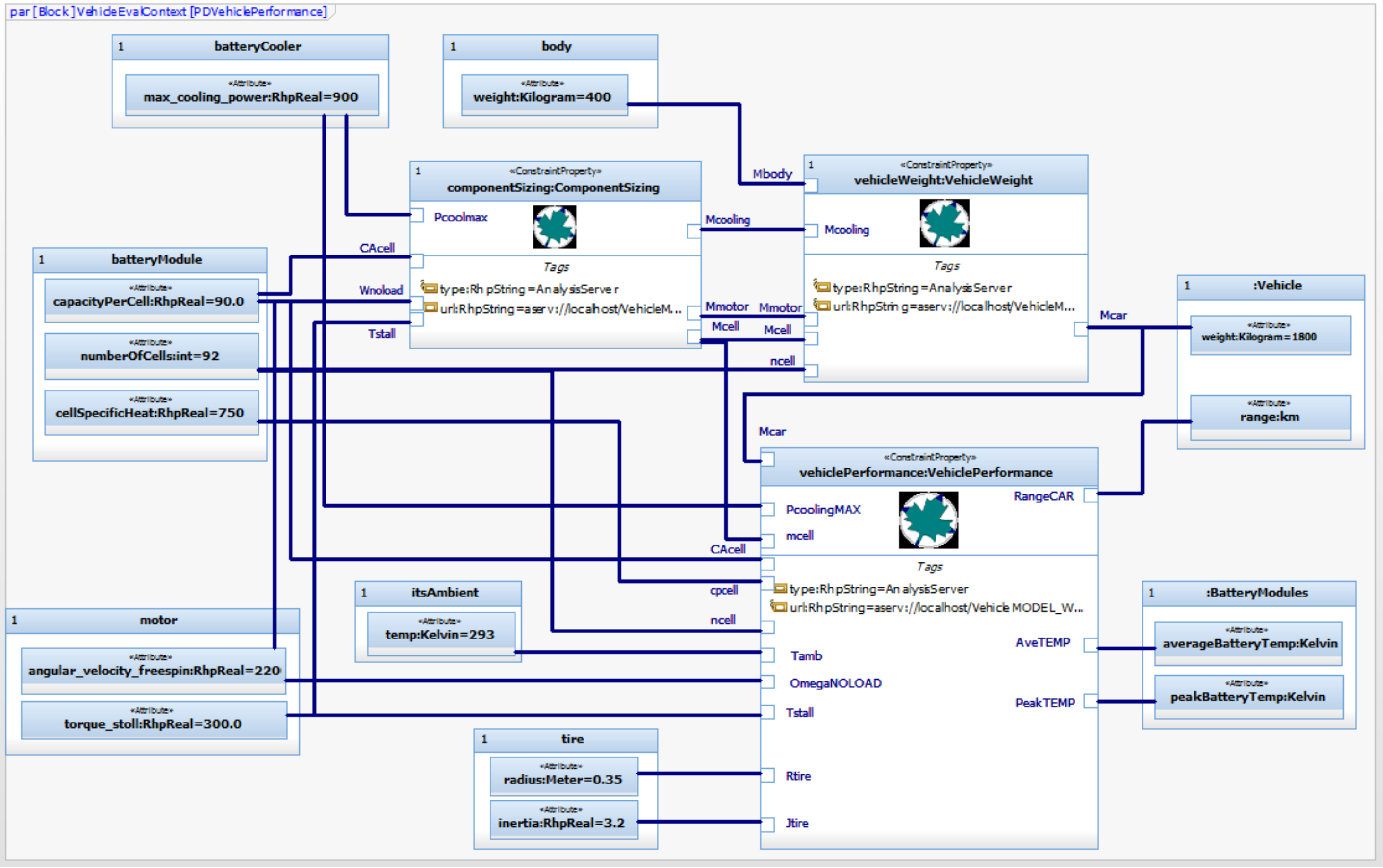
Block Definition Diagram (1)
Context Def.



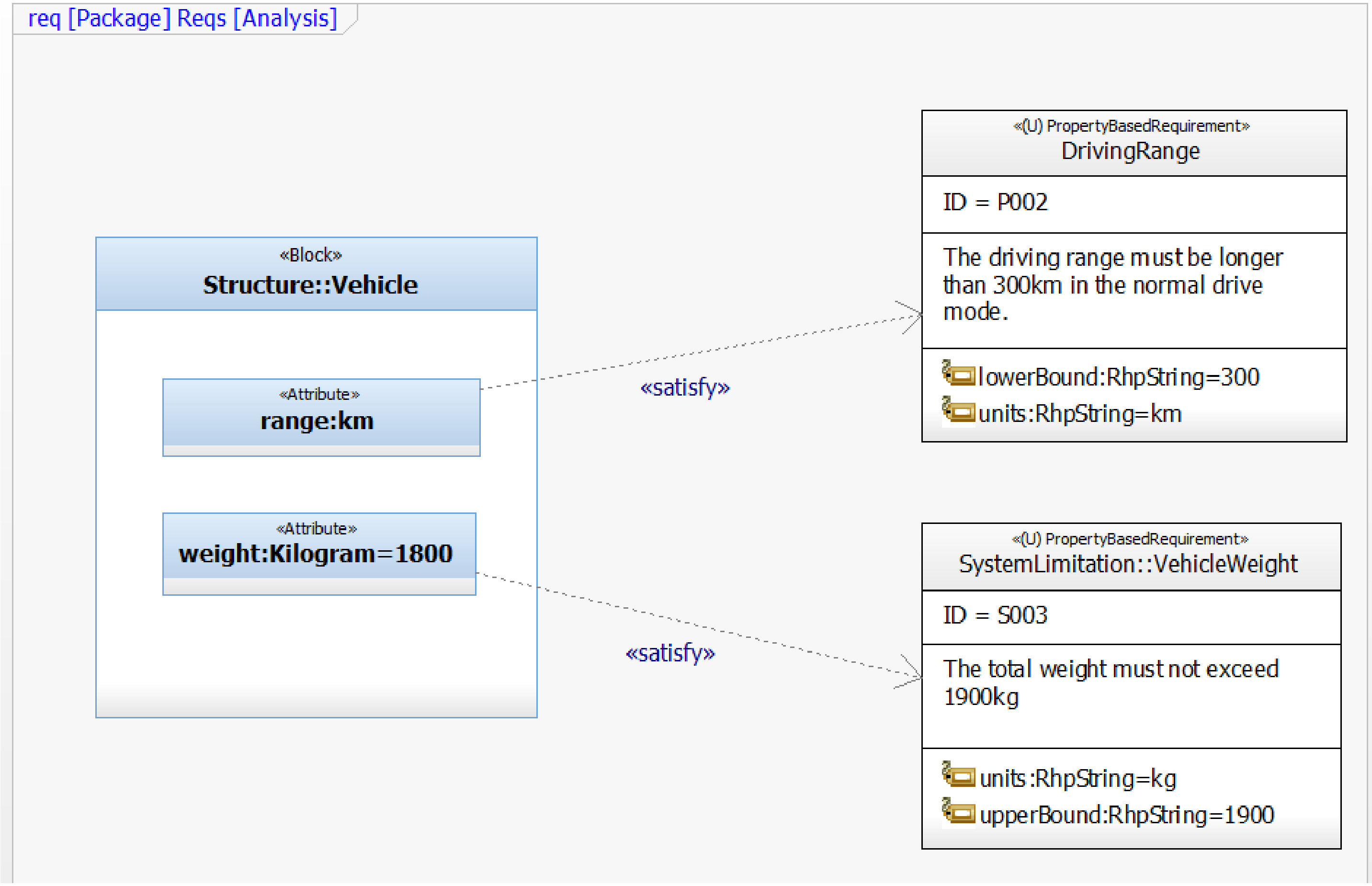
Block Definition Diagram (2)



Architectural Model Parametric Diagram



The Satisfy Relationship with Requirements



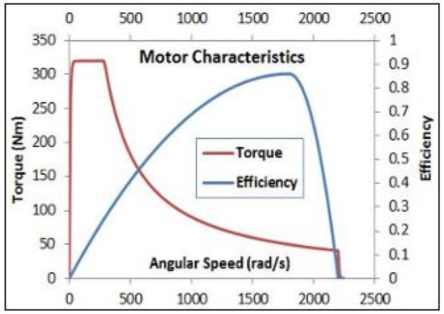
Analytical Model
Multi-domain System-level Dynamics

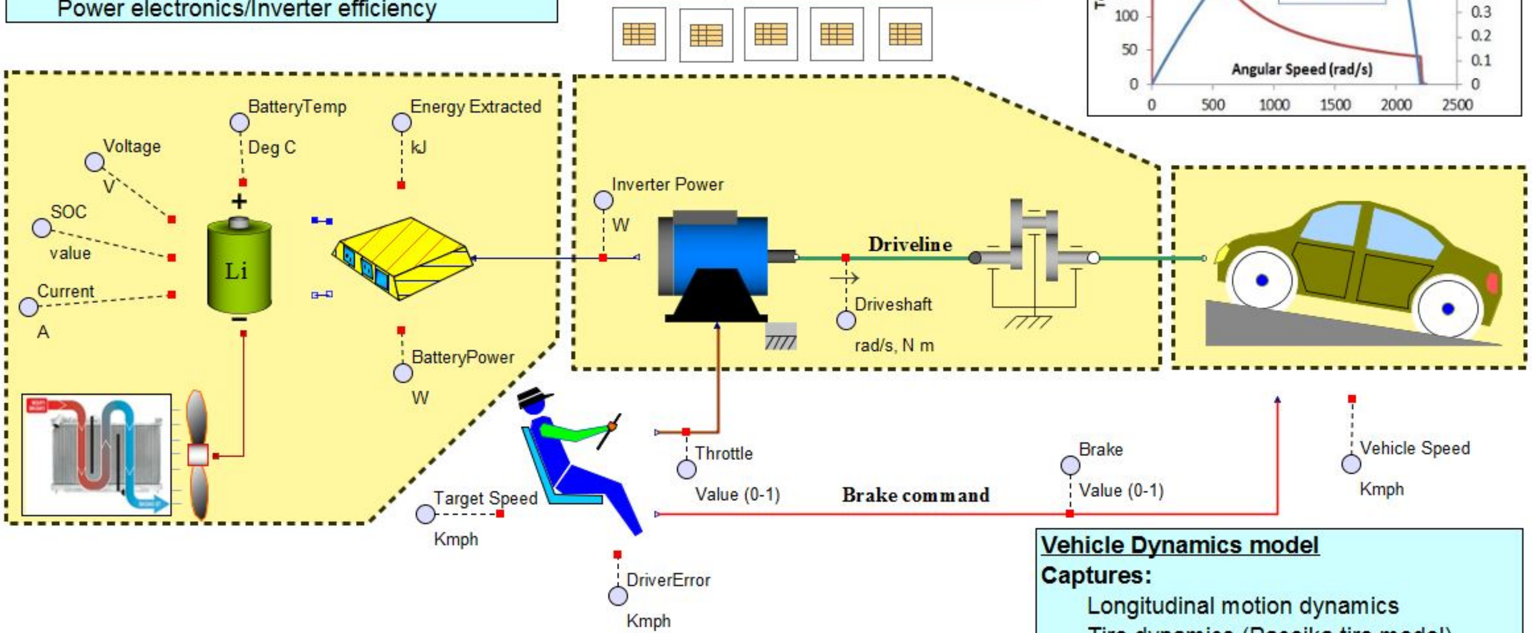
Maplesoft Engineering Solutions

PHOENIX INTEGRATION

Li-ion Battery model
Graphite - Lithium Cobult Oxide electrodes
Captures:
- Thermal effects on the battery chemistry
- Temperature dependent internal resistance
- Degradation effects
- Power electronics/Inverter efficiency

DC motor
Model decribed by
- A torque - speed curve
- A Motor efficiency map
- Rated power and angular speed





Vehicle Dynamics model
Captures:
- Longitudinal motion dynamics
- Tire dynamics (Pacejka tire model)
- Fixed reduction gear from the motor
- Effects of aerodynamic drag forces
- Effects of tire sizing (radius and inertia)
- Effects of variable road grade

Cooling system model
Active and passive heat extraction from battery
Temperature control using on-off thermostat

Driver model
PID Controller
Tuned to follow predefined speed profile

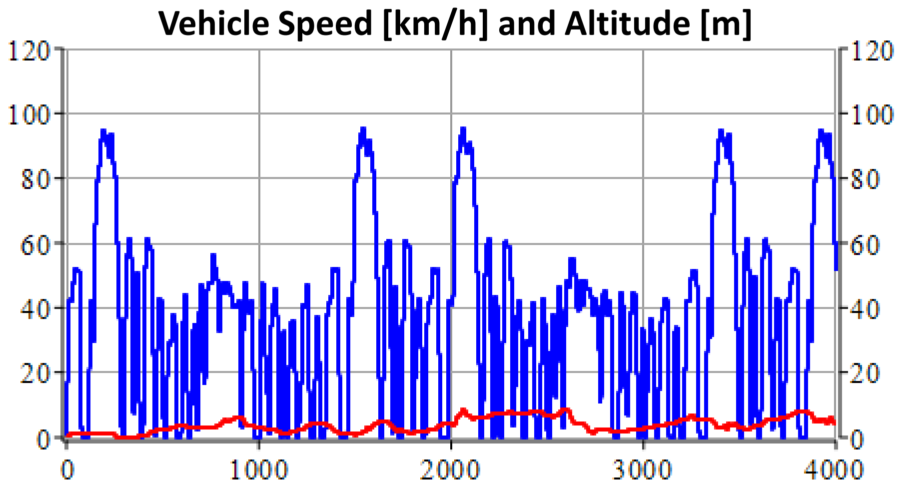
Maplesoft

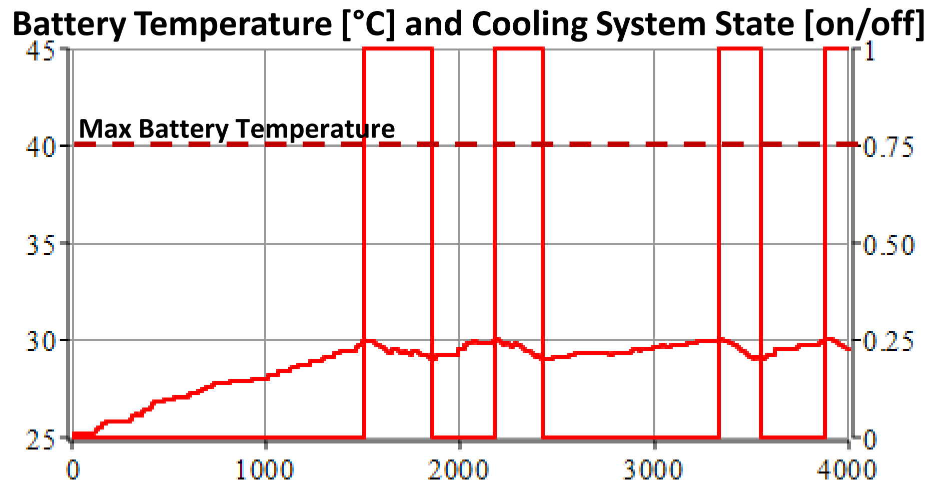
Normal Loading

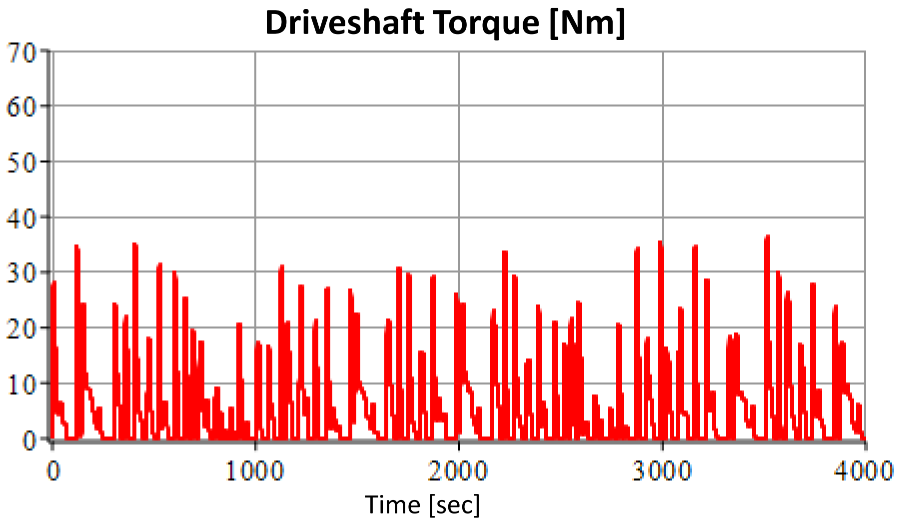
25°C Ambient, Gentle Grades

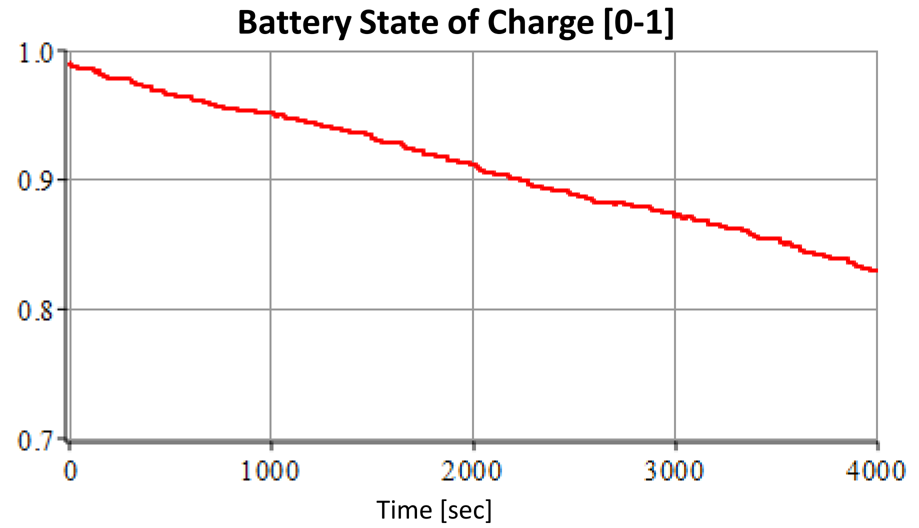
Maplesoft Engineering Solutions

PHOENIX INTEGRATION



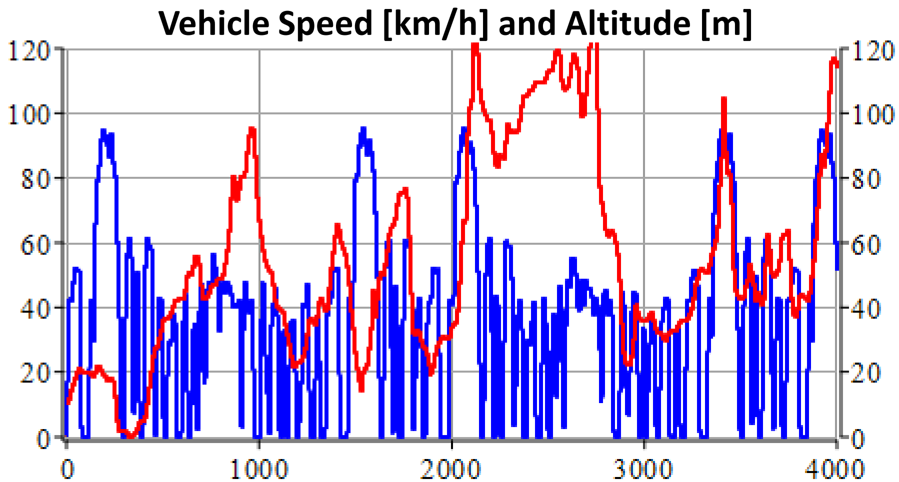


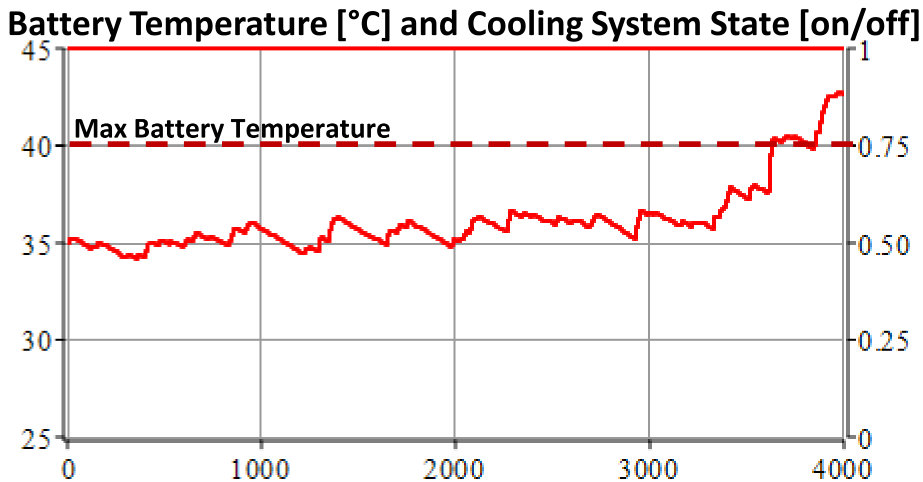


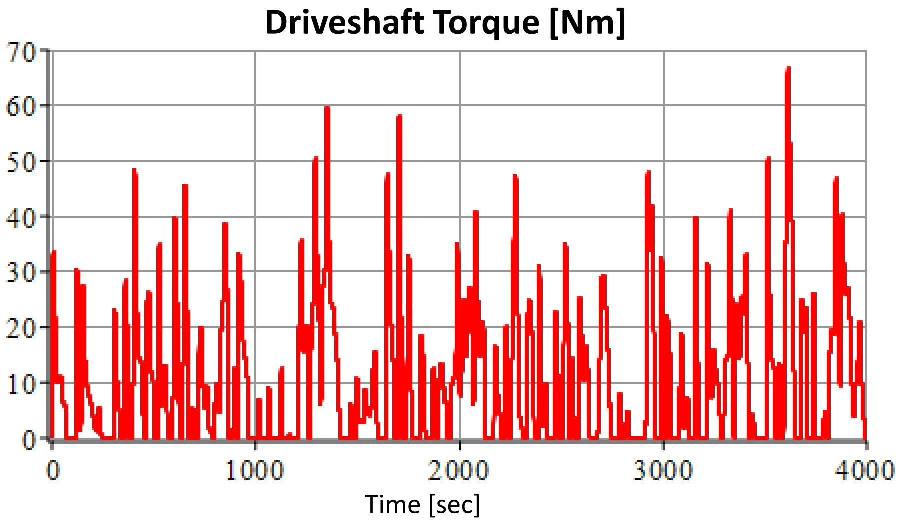


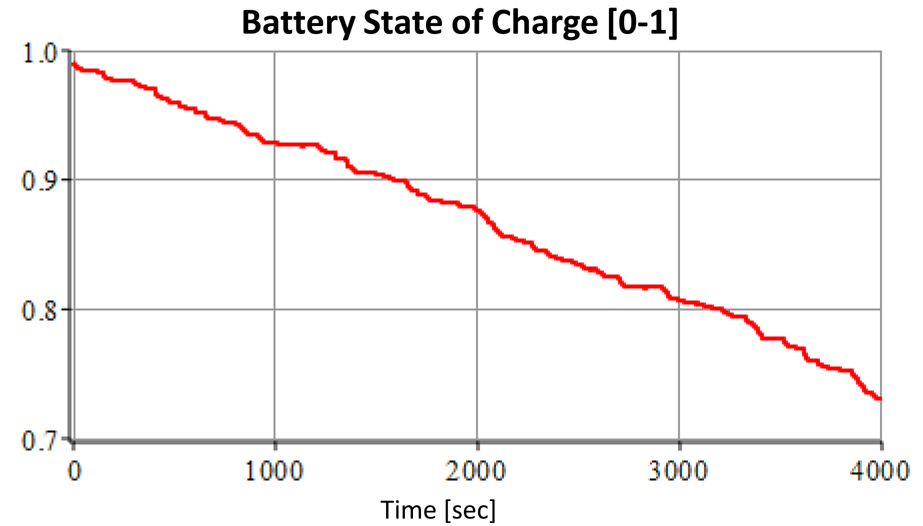
Extreme Loading

35°C Ambient, Steep Grades

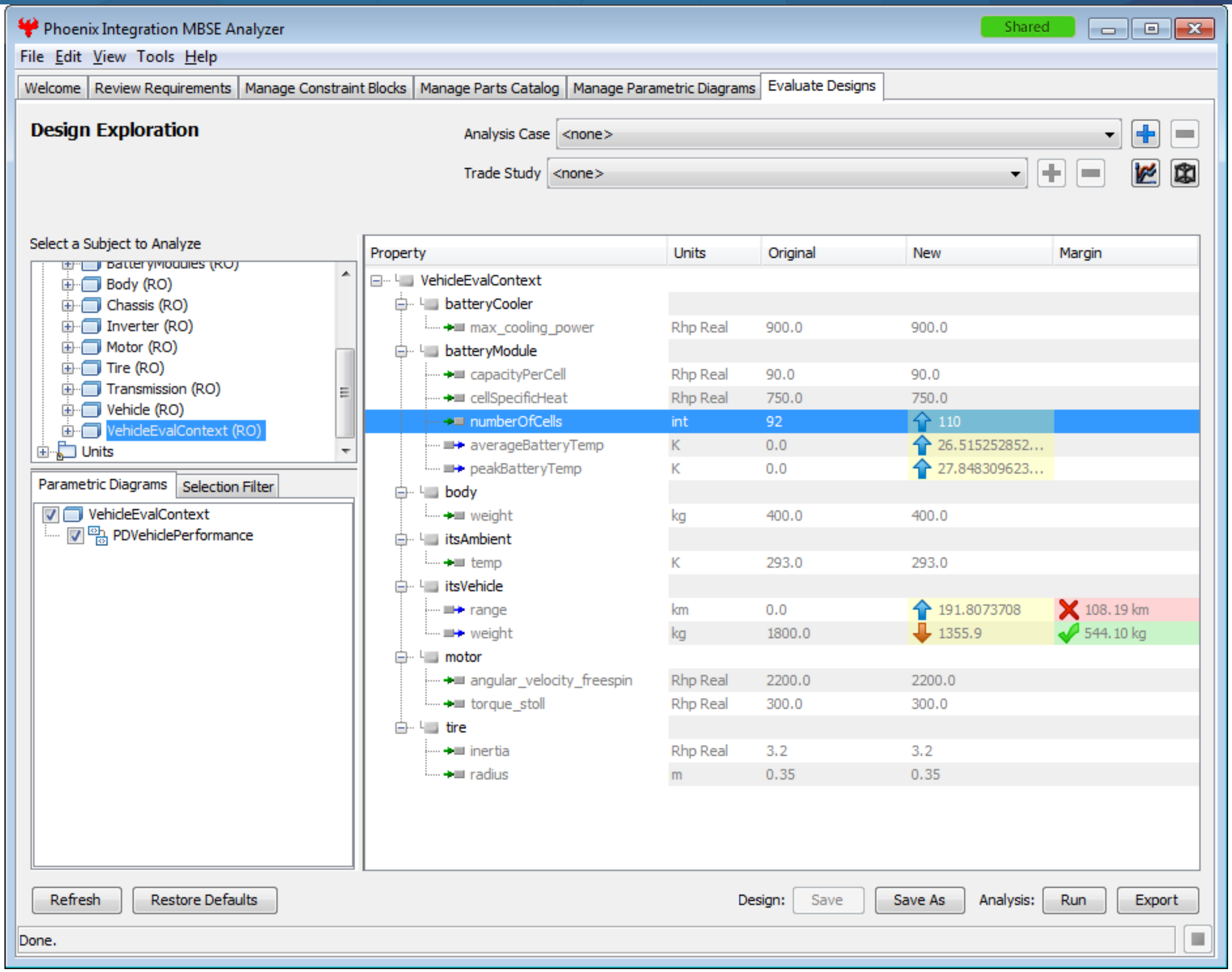




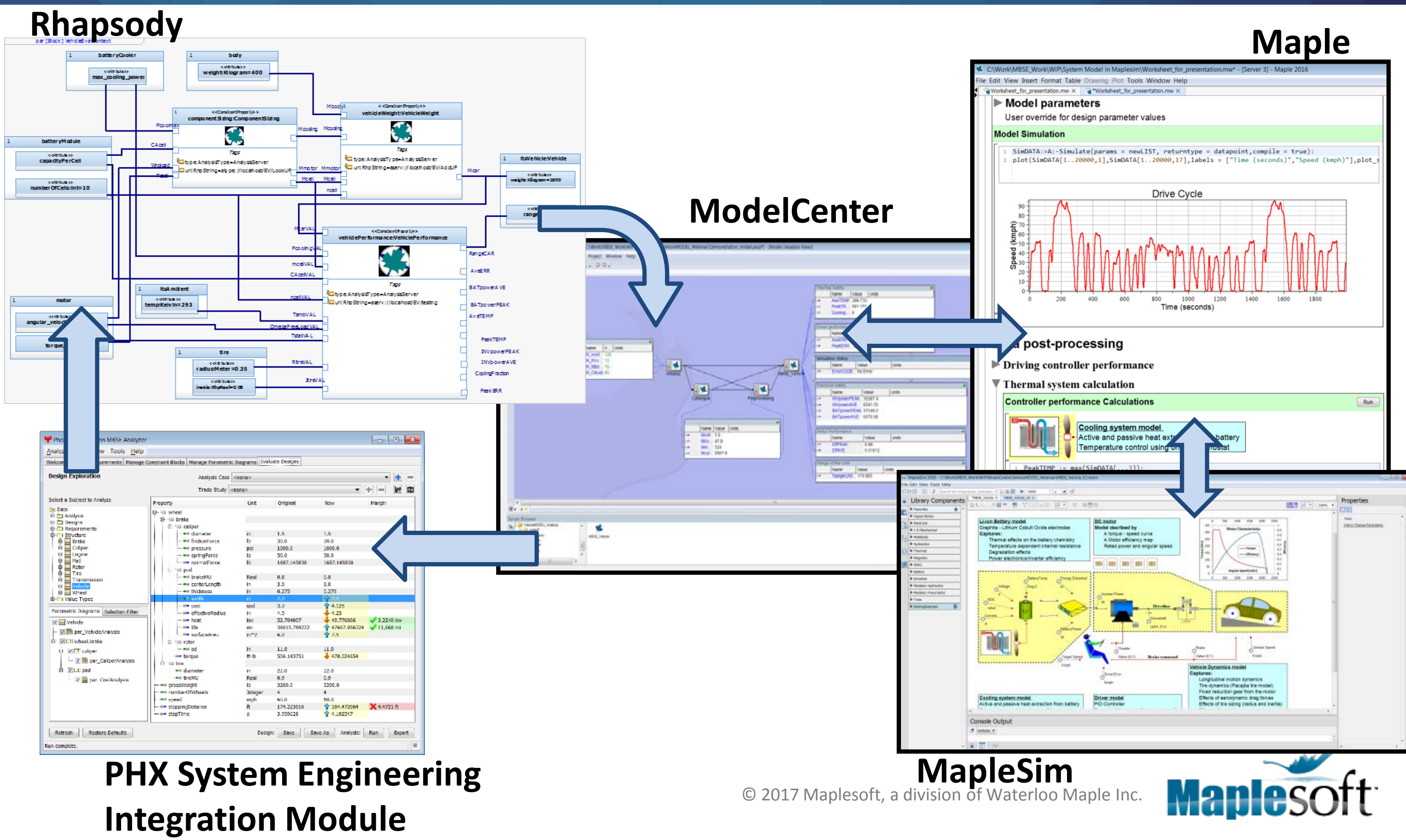




Requirements Compliance Testing



Requirements Compliance Testing



Summary

- MBSE: Process is being increasingly automated through architectural modeling tools
- MapleSim provide rapid functional verification of complex multidomain dynamic systems
- Maple provides powerful environment for data pre- and post-processing as well as managing executable requirements
- ModelCenter brings everything together for rapid requirements-compliance testing, trade-off studies, and impact analysis due to changes in design requirements
- **Convergence of tools helps realize the Systems Design (“V“) process**

Thank You

Questions?

Maplesoft Engineering Solutions

Maplesoft
Mathematics • Modeling • Simulation
A Cybernet Group Company