Model Based Systems Engineering (MBSE) Myth & Reality

IEEE Lecture/Webinar
15 December 2016

Rick Steiner, INCOSE Fellow, ESEP, OCSMP Advanced
Co-Chair, SysML 1.5 Revision Task Force
Skygazer Consulting
858.260.9520
rick@ricksteiner.net

Topics

- Description of MBE and MBSE
  - What it is
  - What it isn’t
- Elements of Successful MBSE deployment
  - Pitfalls
  - Success strategies

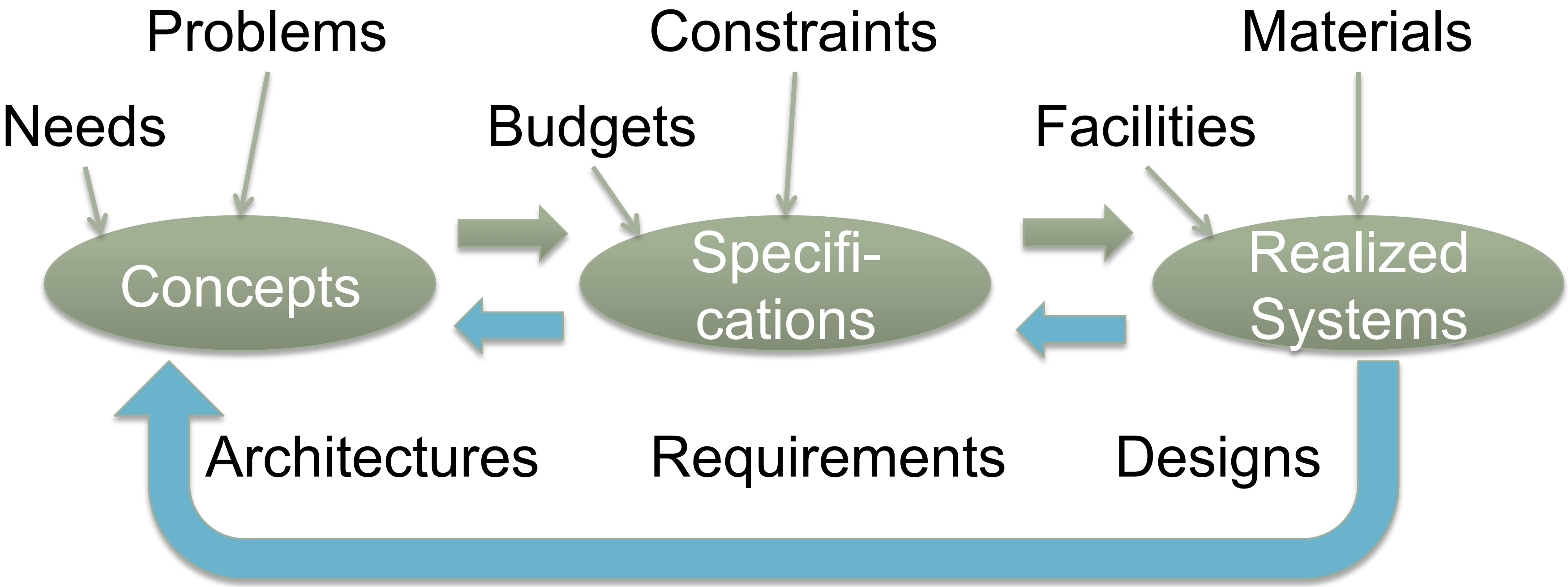
What IS MBE?   What is MBSE?

- Steiner's definition: An *Enabler* with *Implications*
  - Model Based Engineering (MBE) is the consistent application of information science, computer modeling and simulation technology to **enable engineers to be both more productive and more effective**.
  - Model Based Systems Engineering (MBSE) is the specific application of MBE principles to make **systems engineers** more productive and more effective.
  - Models must therefore unambiguously and clearly **represent and communicate applicable (systems) engineering concepts.**
  - In MBSE, models must take on "a **central and governing role** in the specification, design, integration, validation, and operation of a system" [Estefan]

- National Defense Industrial Association* (NDIA) Definition:
  - MBE "uses **models** as an integral part of the **technical baseline**".
  - "Core to MBE is the **integration** of descriptive/design models with computational models".

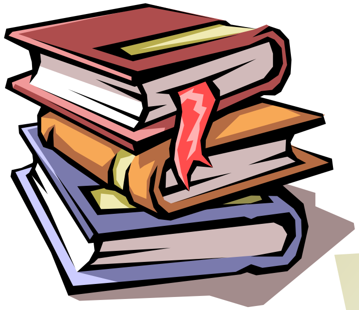
*NDIA Systems Engineering Division, M&S Committee, MBE Subcommittee Final Report, Feb 2011

Engineering's Role in Development

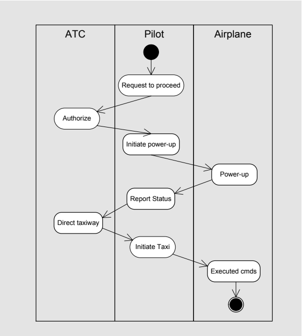


Key challenge:
Ensure the solution continues to fit the problem.
Each arrow introduces potential loss of data...
This is where MBE/MBSE might help!

Basis of MBE:
Documents, Diagrams, and *Descriptive* Models

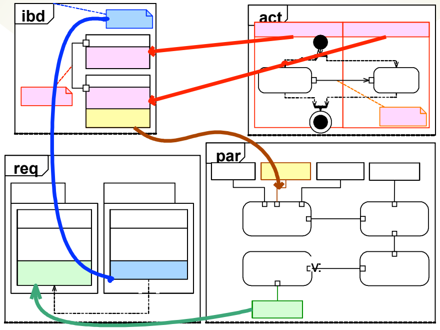


Can be used to describe, or to express imperatives, for a concept, mission, or system



Document: A primarily textual vehicle, possibly with some embedded graphics.

Diagram (Drawing): A primarily graphical vehicle, possibly with some embedded text.



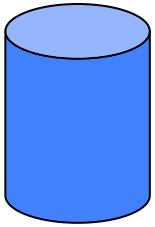
Model: an organized collection of concepts, obeying certain syntactical and methodological rules, from which documents or diagrams can be expressed.

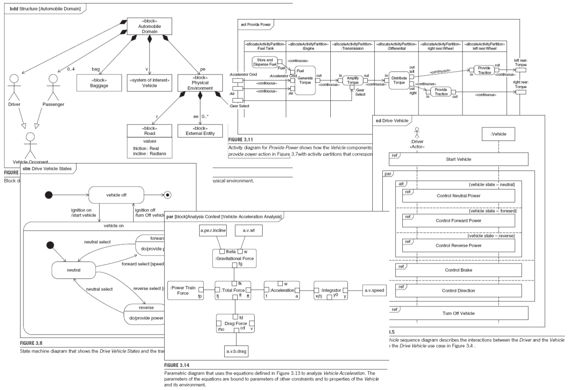
Model vs. Diagrams

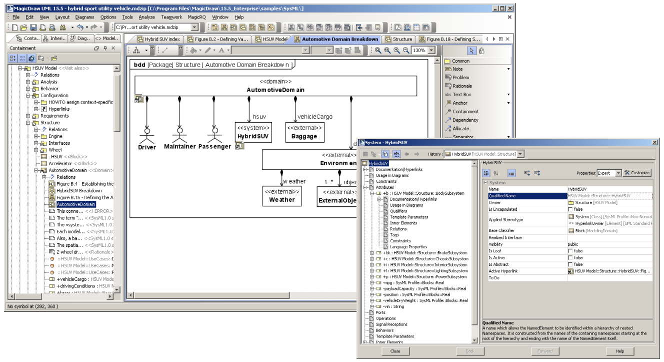
Reality
- Envisioned or actual

Model
- Computer-oriented
- Master repository
- Complete for intended scope

Diagrams
- Human-oriented
- Subset views







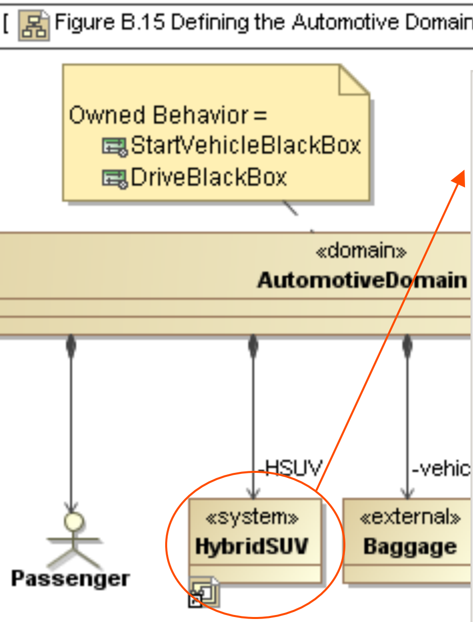
Tools
- Authoring, viewing, executing, ...

Source: SysML 101/201 course material – Russell.Peak@gatech.edu

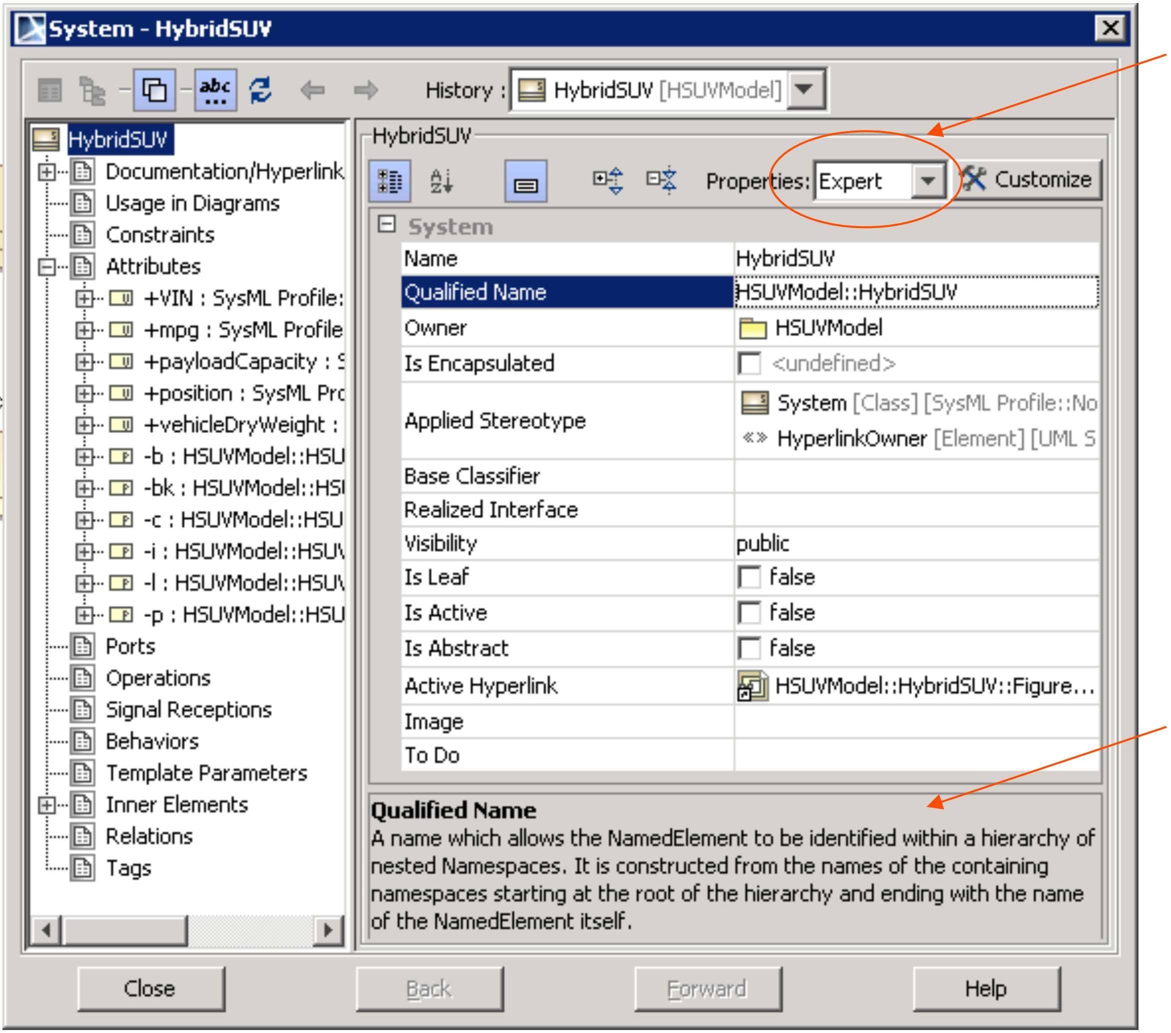
Acknowledgements: Selected portions from Friedenthal et al. 2008 and MagicDraw samples.

Beyond Pretty Pictures: Rich Modeling Attributes (Metadata) in each SysML Block

Hybrid SUV [Appendix B - SysML spec]



Specification Window



- HybridSUV
  - Documentation/Hyperlink
  - Usage in Diagrams
  - Constraints
  - Attributes
    - +VIN : SysML Profile:
    - +mpg : SysML Profile
    - +payloadCapacity : S
    - +position : SysML Pro
    - +vehicleDryWeight :
    - -b : HSUVModel::HSU
    - -bk : HSUVModel::HSI
    - -c : HSUVModel::HSU
    - -i : HSUVModel::HSUV
    - -l : HSUVModel::HSUV
    - -p : HSUVModel::HSU
  - Ports
  - Operations
  - Signal Receptions
  - Behaviors
  - Template Parameters
  - Inner Elements
  - Relations
  - Tags

HybridSUV

System

Name	HybridSUV
Qualified Name	HSUVModel::HybridSUV
Owner	HSUVModel
Is Encapsulated	□ <undefined>
Applied Stereotype	System [Class] [SysML Profile::No «» HyperlinkOwner [Element] [UML S
Base Classifier	
Realized Interface	
Visibility	public
Is Leaf	□ false
Is Active	□ false
Is Abstract	□ false
Active Hyperlink	HSUVModel::HybridSUV::Figure...
Image	
To Do	

Qualified Name
A name which allows the NamedElement to be identified within a hierarchy of nested Namespaces. It is constructed from the names of the containing namespaces starting at the root of the hierarchy and ending with the name of the NamedElement itself.

Filters to add/remove detail

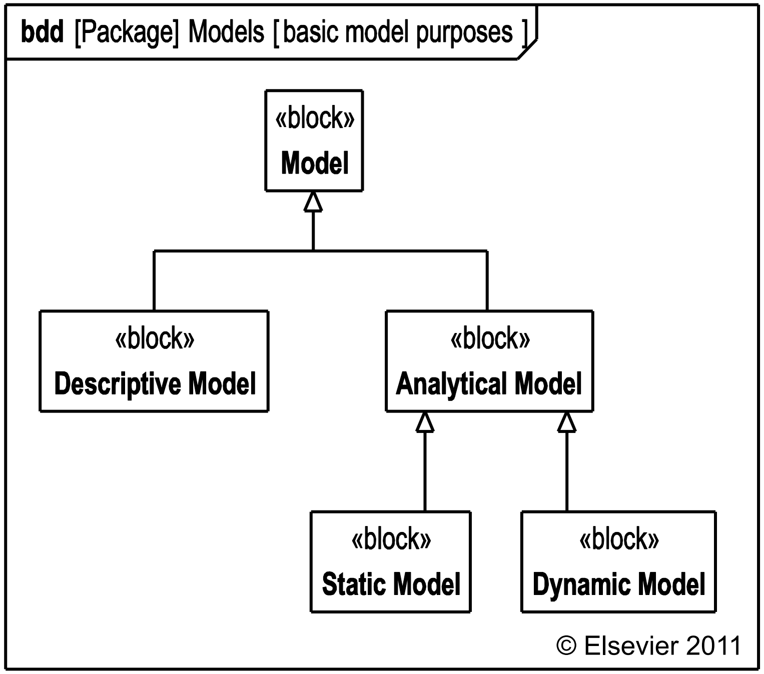
Built-in documentation on each model attribute

Consistent and comprehensive access to many model aspects (attributes, meta-attributes, ...)

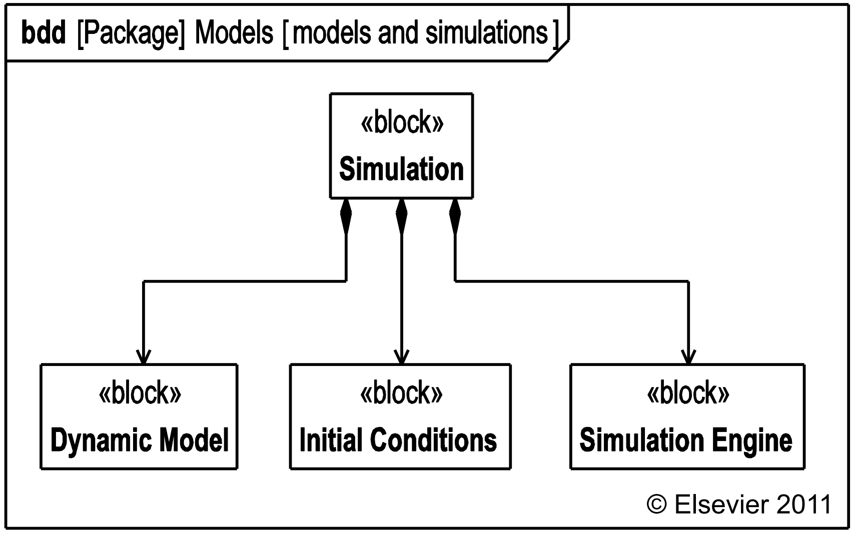
Source: SysML 101/201 course material – Russell.Peak@gatech.edu

[MagicDraw: hybrid sport utility vehicle.mdzip]

Models and Simulations – A Broader Set of Definitions



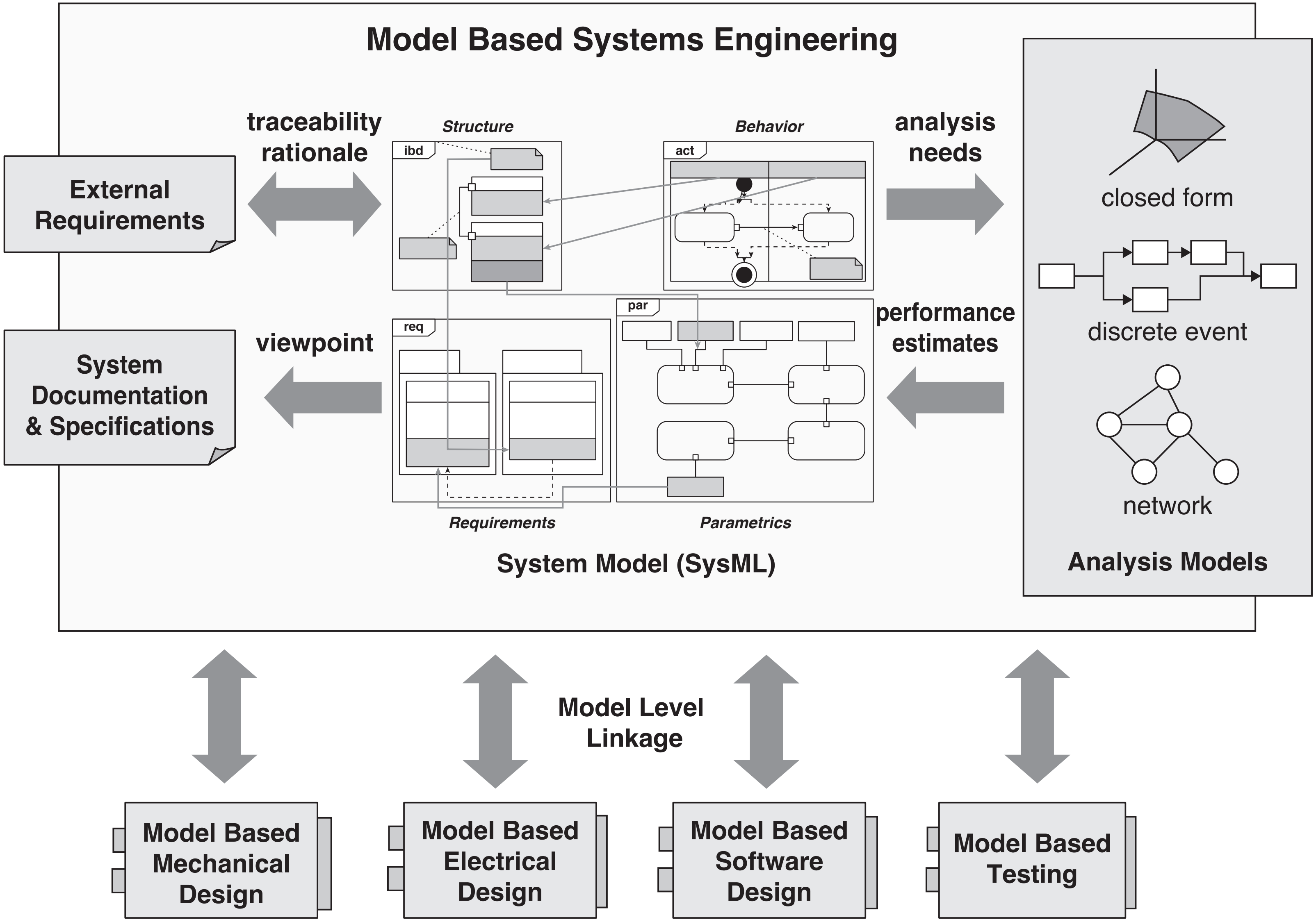
© Elsevier 2011


bdd [Package] Models [models and simulations]
«block» Simulation
«block» Dynamic Model
«block» Initial Conditions
«block» Simulation Engine

© Elsevier 2011

- Models can be ***descriptive*** or ***analytical***
  - Descriptive models are human interpretable, and ***can*** tolerate ambiguity
  - Analytical models are machine interpretable (computational), and ***cannot*** tolerate ambiguity
- Simulations are built from dynamic (time-aware) analytical models.

Model Based Systems Engineering (MBSE) as a Framework for MBE



Model Based Systems Engineering

External Requirements

traceability rationale

System Documentation & Specifications

viewpoint

System Model (SysML)

analysis needs

performance estimates

Analysis Models

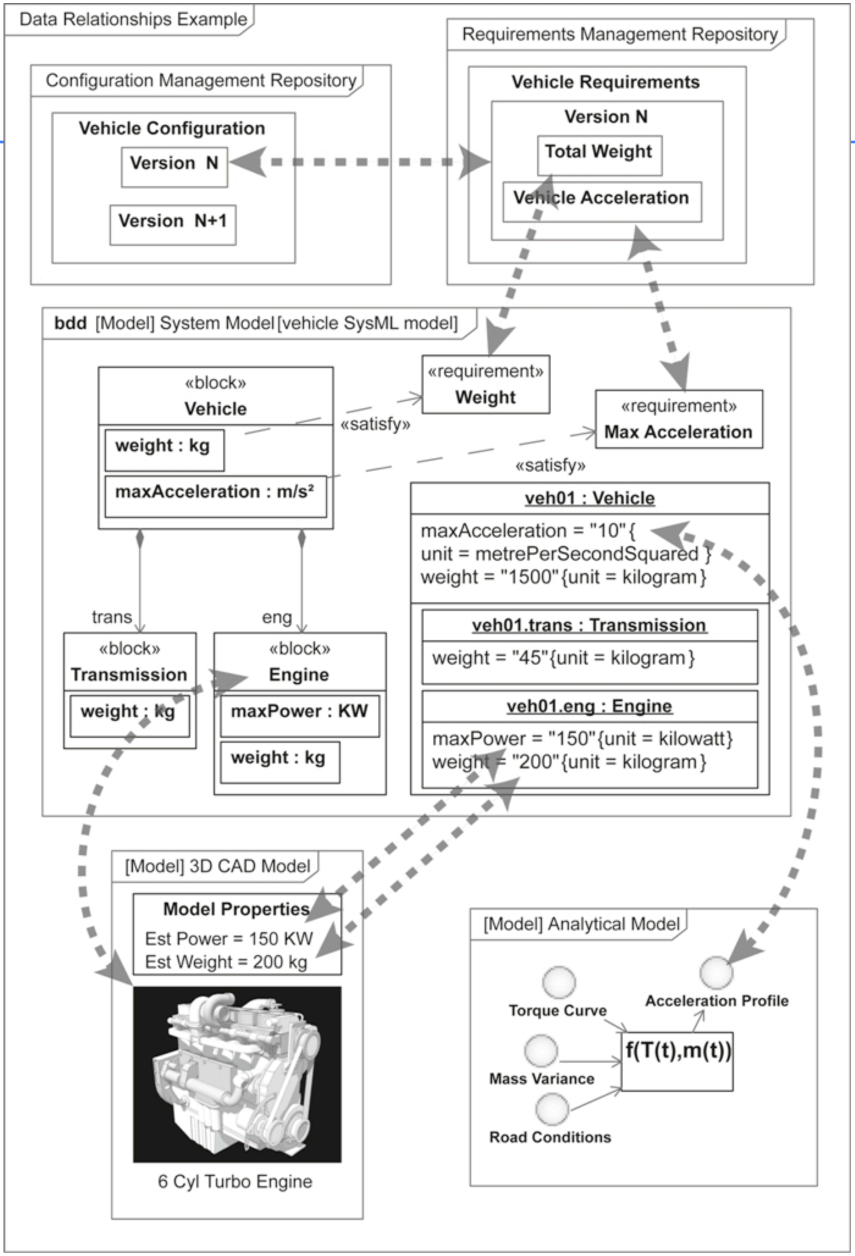
Model Level Linkage

Model Based Mechanical Design

Model Based Electrical Design

Model Based Software Design

Model Based Testing



© 2015 Elsevier

Cross-Model Linkage

- Requirements Management
  - DOORs
- Configuration Management
  - SVN, TeamCenter
- PDM/CAD Repositories
  - Windchill
- Analysis/Optimization Environments
  - Phoenix Model Center

Reality −
MBSE Has Potential to Increase Efficiency by:

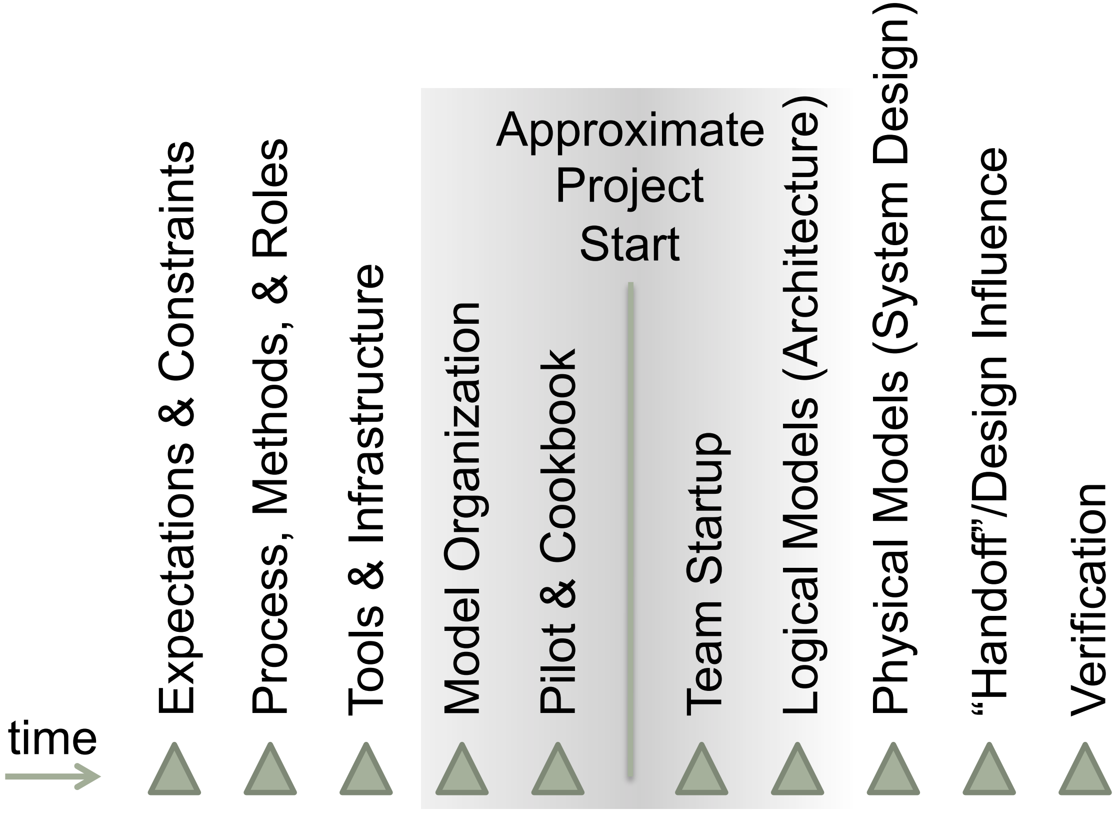
- Minimizing Redundancy (& maximizing consistency)
  - Same information in multiple specifications is captured once in the model
  - Design review information/charts generated from the model
- Automating Tedious Tasks
  - Building and checking verification matrices
  - Linking design models to verification models
  - Generating frequent updates to specifications or design documents
    - Documents generated from the model need toolsmith involvement
- Identifying and Minimizing Waste (especially if downstream process are also model based)
  - Analysis – appropriately focused on performance characteristics
  - Design – consolidating specification artifacts, tracing to design.
  - Testing, Manufacturing – durable visibility into rationale
- Appropriately Reusing Parts of Previous System Models
  - Libraries of needs/scenarios, design patterns, product lines
  - In each case, “fitness for use” must be carefully assessed

Myth - MBSE WILL NOT:

- Recover schedule/budget for a program in trouble.
  - Management & estimating practices are largely unaffected
- Make your engineering labor rates competitive.
  - Your organization will need to invest to grow skillsets
- Instantly make engineers more productive (with a *single* tool)
  - Processes, methods, and tools need to be tailored to company/program needs, then deliberately deployed
  - Effectiveness & productivity may increase, if managed
- Save money in the early phases of a program.
  - Big payback realized in sustaining and follow-on phases
- Automatically invigorate all your engineering staff.
  - By making MBSE relevant & meaningful, engineers want to participate

"A fool with a tool still remains a fool" [Fuller]

Preparing for and Executing MBSE



time

Expectations & Constraints

Process, Methods, & Roles

Tools & Infrastructure

Model Organization

Pilot & Cookbook

Approximate Project Start

Team Startup

Logical Models (Architecture)

Physical Models (System Design)

"Handoff"/Design Influence

Verification

- Significant decisions must be made before the start of system modeling on a project

Expectations & Constraints

- Role and scope of MBSE within the broader program
  - Not all areas of the project need to or will embrace MBSE
  - **Choose the high-payoff areas first**
  - **Get MBSE on the critical path**… somewhere!
    - A new process that isn’t critical will be marginalized.
- Type and nature of SE deliverables
  - Generated directly from the model(s)?
  - Include material from the model(s), manually updated?
  - Maintained independently from the model(s)?
- Anticipated nature of stakeholder interaction
  - Can stakeholders tolerate churn, or need to see only final products?
  - The need for rapid, responsive document/view generation from the model(s) cannot be overemphasized!

Process, Methods and Roles

- Establishing the SE team
  - Ordain a **Methodologist**!
    - This one person must have the authority to establish and tailor methods, guidelines, tool settings, etc.. The go-to person for what to do next SE/MBSE wise on the program
    - Should be a senior systems engineer with a flair for MBSE, and a level head for problem solving
  - Establish a **Toolsmith**
    - This is a SW person who loves scripting suited to generating custom output from the model(s)/tools, and can handle arcane APIs & OSLC
- Select and tailor a methodology
  - Methodologist, Chief Engineer must agree on something that meets the goals of the program
- Be clear on roles, and *who owns what in the model*
- Involve stakeholders from IPTs/other disciplines
- Invest in developing modelers… but not too many at first
  - ***A model is not a whiteboard***… ‘the model is the design’!

Tools and Infrastructure

- Methodologist helps develop a ‘**Tool Spec**’ for the project
  - – May need assistance of an IT architect
- Identify the value-added information flow desired across tools
- Be willing to leverage existing development environments
- Focus on **value added** analysis, such as parametric modeling and optimization
- Specifically ask **how each simulation contributes** to the robustness of the design information
- Identify which pre-exiting models need to be reused, and how integrated they need to be
- Cannot over-emphasize the need for consistent, reliable, adaptable **document and view generation**!

Model Organization

- Much like a shared drive or CM system, a model can become unmanageable very quickly if it doesn't have a **clear organization structure**.
- If there aren't corporate guidelines, pick a structure that most people with and run a pilot
- It's up to the methodologist and chief engineer to **enforce model discipline**! Be ready for it.
- The organization structure of the model may change after key milestones, as more people are brought onto the program.

Pilot and Cookbook

- Bring a small team of systems engineers and modelers up to speed on the implemented methodology, infrastructure, and regain familiarity with the applicable language concepts.
- Exercise the tools in the target development environment, with active interfaces between them
- Exercise and refine the model organization principles
- Exercise report/document generation, data extraction, and formatting tools to validate the work in store for the toolsmith(s)
- Quickly develop a **cookbook** (lightweight guidelines or notes) for model development on the program, specific to the tool, environment, interfaces, and model organization.

Team Startup

- Training should be **as close to project start as possible**.
  - – ‘factory’ tool training goes stale in a few months
  - – Training using a project-related problem can pique interest and get folks enthused quickly
  - – The modeling cookbook assumes a trained modeler. Train to the cookbook and get modelers up to speed quickly.
- Not every SE needs to be a model builder… initially.
  - – But every SE should be able to read the model, and use it to do their job!
  - – Consider leveraging folks with OCSMP certification.

Organizational Responsibilities for MBSE

- Infrastructure
  - Environment, tool licenses
  - **Don’t get too locked in to a specific tool**… they change!
- Standard processes
  - Turn good cookbooks into company guidelines – build a library
- Standard model templates
  - Criteria for model organization, alternatives
- Standard model libraries
  - Share what works across programs
- System Modeling User Groups
- INCOSE, OMG participation
- OCSMP certification
  - Build a cadre of L1-L3 certified modeling professionals!

Backup Material

Definitions

- **Model Based Engineering** – the elevation of models in the engineering process to a central and governing role in the specification, design, integration, validation, and operation of a system [Estefan]

- **Model Based Systems Engineering** – the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases. [INCOSE]

- **Model Driven Software Development** – a kind of domain engineering, specific to software systems, involving a platform independent model translation into a platform specific model using an domain specific language or general purpose language [Balcer et al]

- **Innovation** – the implementation of an idea, concept, research, or invention; activities that refine or simplify and organization’s processes (e.g., efficiency gains or waste reduction) [Kowalenko]

MBE Definition

- Model-Based Engineering (MBE): An approach to engineering that uses models as an integral part of the technical baseline that includes the requirements, analysis, design, implementation, and verification of a capability, system, and/or product throughout the acquisition life cycle
- Model: A physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process. (DoD 5000.59 -M 1998)
- Preferred MBE Practices:
  - Models are scoped to purpose/objectives
  - Models are appropriate to the context (e.g., application domain, life cycle phase)
  - The models represent the technical baseline that is delivered to customers, suppliers, and partners
  - Models are integrated or interoperable across domains and across the lifecycle
- Core to MBE is the integration of descriptive/design models with the computational models

NDIA Systems Engineering Division, M&S Committee, MBE Subcommittee Final Report, Feb 2011

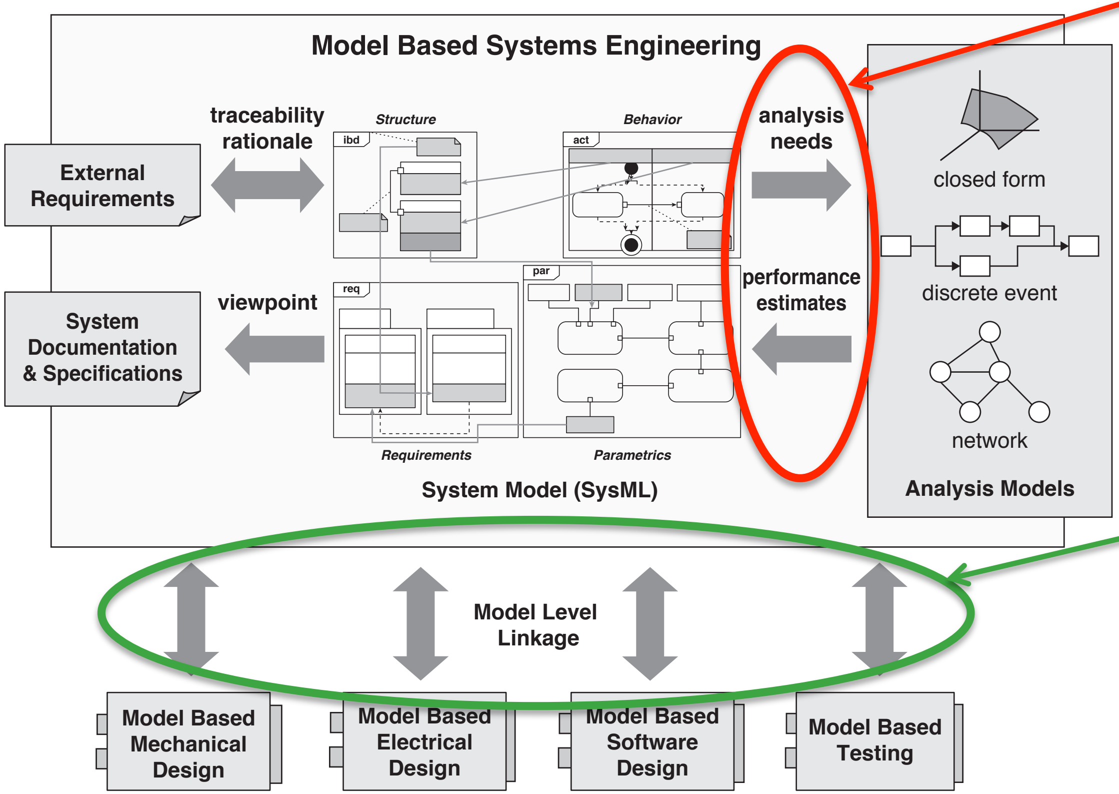
Why Worry About Model Based Engineering?

- Each engineering discipline is pursuing productivity gains through model based engineering, in one form or another:
  - Product Realization (Electrical)
  - Platform Based Development (Semiconductor Industry, AUTOSAR)
  - Model Based Development / Model Based Enterprise (Mechanical)
  - Model Driven Software Development / Model Driven Architecture
  - Model Based Systems Engineering / Model Driven System Design
- Industry segments are shifting to model based approaches for complex, high volume products (Automotive, Semi-conductor)
- DoD is investigating deployment of model based engineering on a broader scale
  - DARPA META I and META II
  - DoD Systems 2020
- Product Data Management (PDM) makes sense in a model-driven environment, and Product Lifecycle Management (PLM) may not be possible without it.
- INCOSE SE Vision 2020 included MBSE, SE Vision 2025 extends it

Model Based - A Distinct Approach to Systems Engineering

- Distinction between approaches
  - **Document Driven**
    - Traditional approach, manage design by a limited set of dense specification documents
  - **Requirement Driven**
    - Manage design using a database of requirements (an interconnected set of declarative statements)
  - **Model Driven**
    - Manage design using highly interconnected descriptive objects as a framework for performance, function, and interface information
- Process implications of MBSE
  - SE products/artifacts shift from being documents to being models
  - customer view into design shifts from being specifications to being report routines/viewers into design database, or scenarios in which models execute
- It’s clear we want model based approach (or is it?)
  - System Model could include entire “body of knowledge” developed as part program - this can be a BIG DEAL, and CM nightmare!
  - is not stand-alone set of simulations
  - is not independent from Product Data Management (PDM)

Model Based Systems Engineering (MBSE) as a Framework for MBE



This linkage can be accomplished through API/ commercial toolsets:

- InterCAX
- Phoenix MBSEPak

This linkage is still being developed… need to work with PLM vendors

- Mentor Graphics
- PTC/Windchill
- IBM

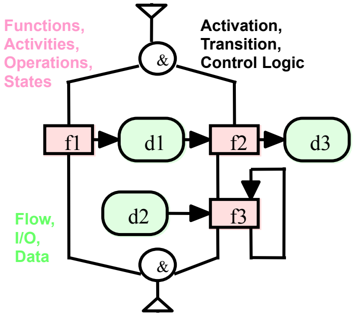
Characteristics of Programs that can Benefit from MBSE

- Systems Engineering is already highly collaborative across disciplines, including cross-discipline internal reviews
- Customer has strong interest and/or POC for SE, and expects frequent working-level interaction
- Geographic distribution of team conducive to model collaboration
- SE metrics and SEMP support model-based milestones & maturity, rather than document based metrics
- Customer expecting model-based deliveries, rather than only documents
- SE ≥ 15% of the total program budget

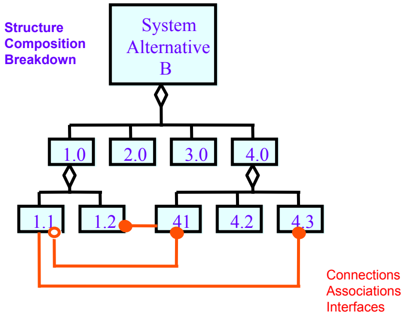
A System Model is ***More*** Than a Database Because:

A Model has a ***stronger abstract syntax***, and can be used to articulate systems engineering concepts, such as

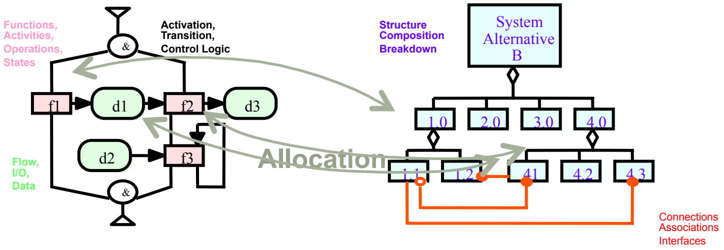
- Causal Chain Behavior



- State Based Behavior
- Structural Hierarchy



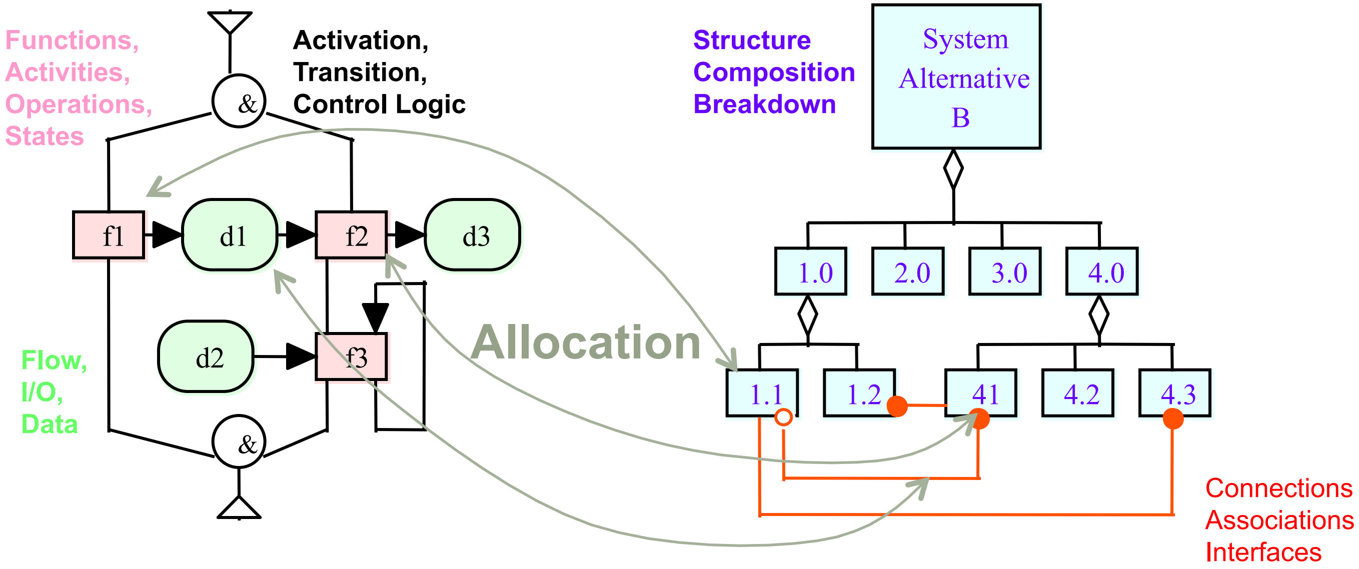
- Allocation of Function onto Form



- Message-based Interfaces

System Modeling Capability: Segregating, Mapping Function & Form

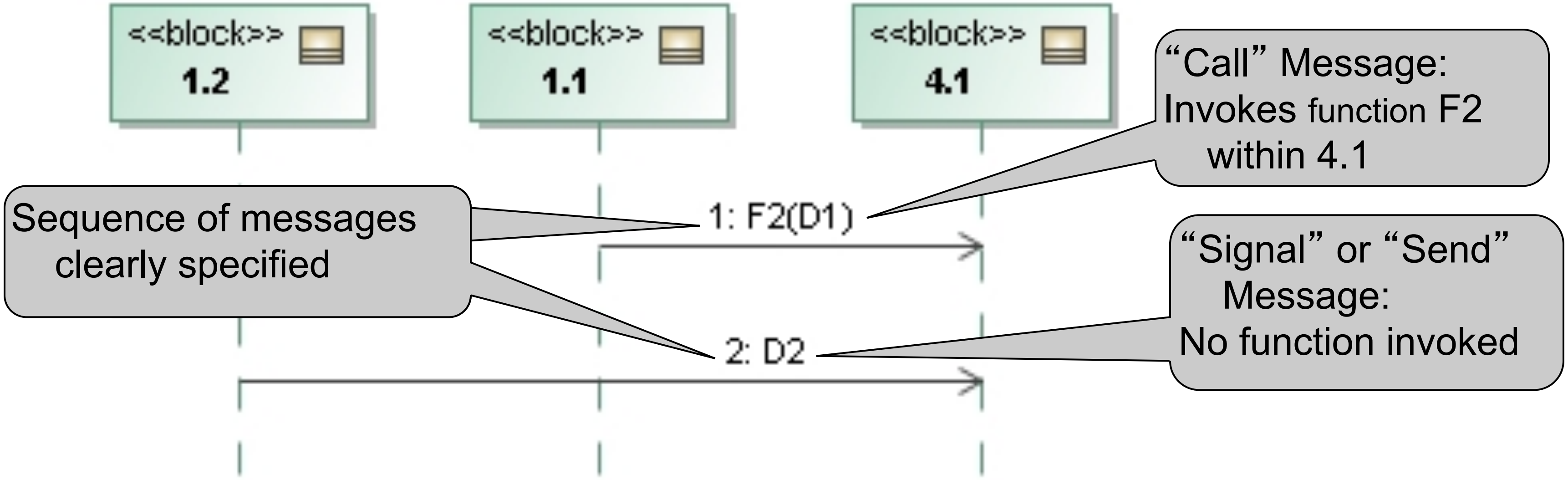
- System Model needs to support Behavior, Structure, and Allocation
  - Behavior has to do with action: functions (verbs), inputs & outputs (messages, material)
    - Functional flow (example below) is only ONE way to represent behavior; others include sequence, state, IDEF, etc.
  - Structure has to do with “pieces” or components: objects (nouns)
  - Allocation describes what piece/component exercises which function
- Segregating behavior from structure is key to trade space, making room for innovation!
  - Reallocation of functions to different objects, aggregating functions a different way for efficiency
  - Reuse of functions or objects in different places
  - **Consistent with “late binding”** philosophy in software



Behavior/Function **Structure/Form**

Integrating Function & Form

- Service Oriented Architecture approach defines services as structure, and encapsulates function within.
- Messages & Service Requests are visible externally via Sequence Diagrams (Interactions).
  - – No functional allocation is necessary: it is implied by sequence, or explicitly invoked via call messages.
- Internal behavior of services is typically modeled with State Machines.

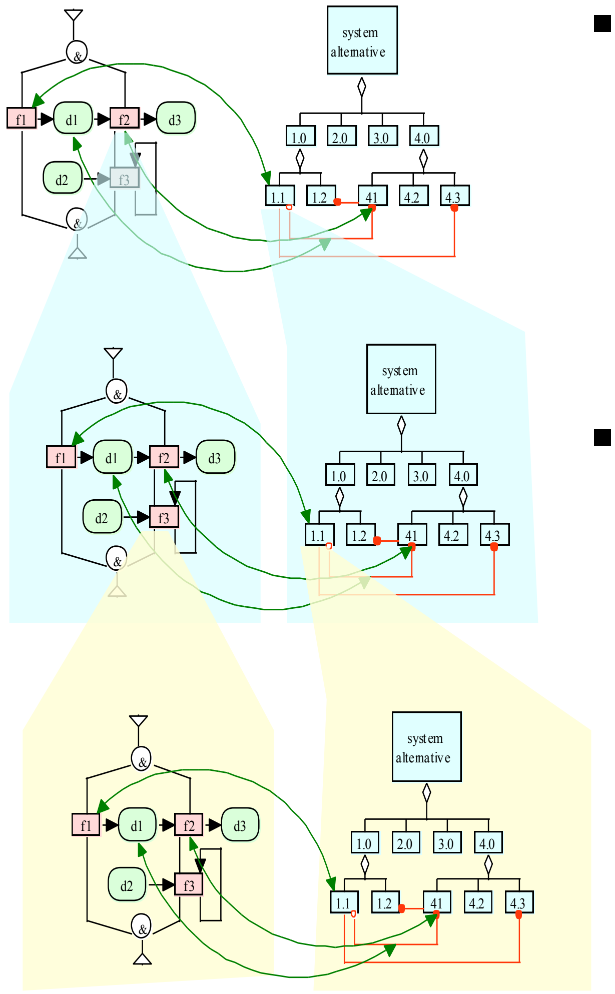


"Call" Message: Invokes function F2 within 4.1

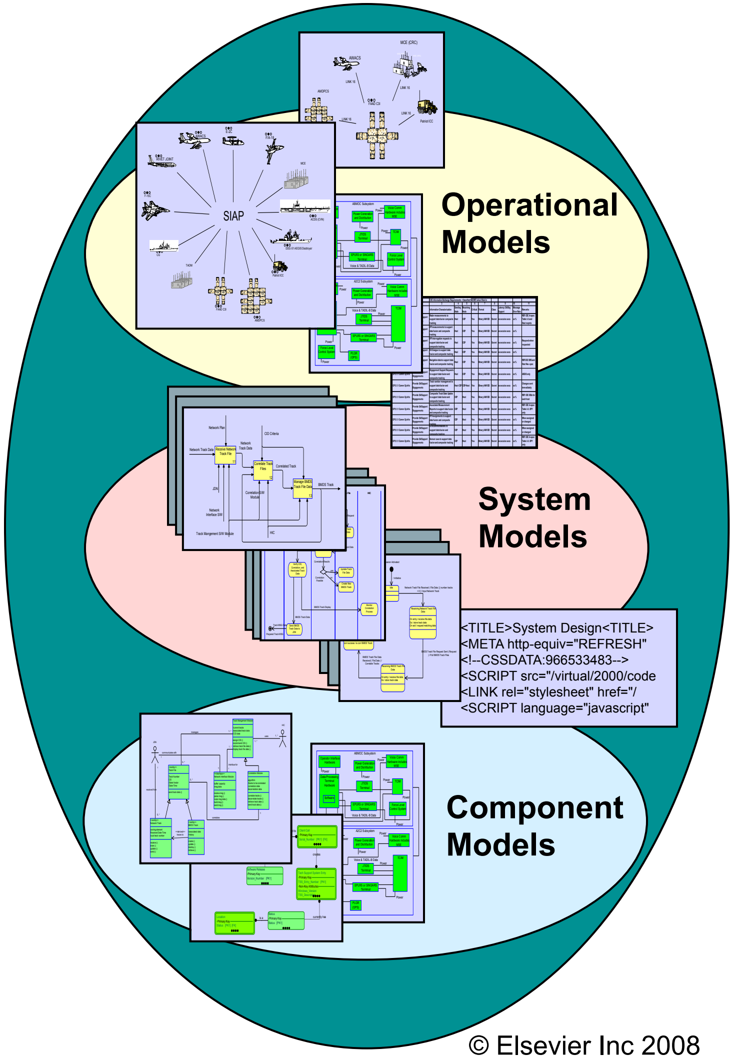
Sequence of messages clearly specified

"Signal" or "Send" Message: No function invoked

A System Model Should be Hierarchical: Each Element Needs to Fit in a Higher Context



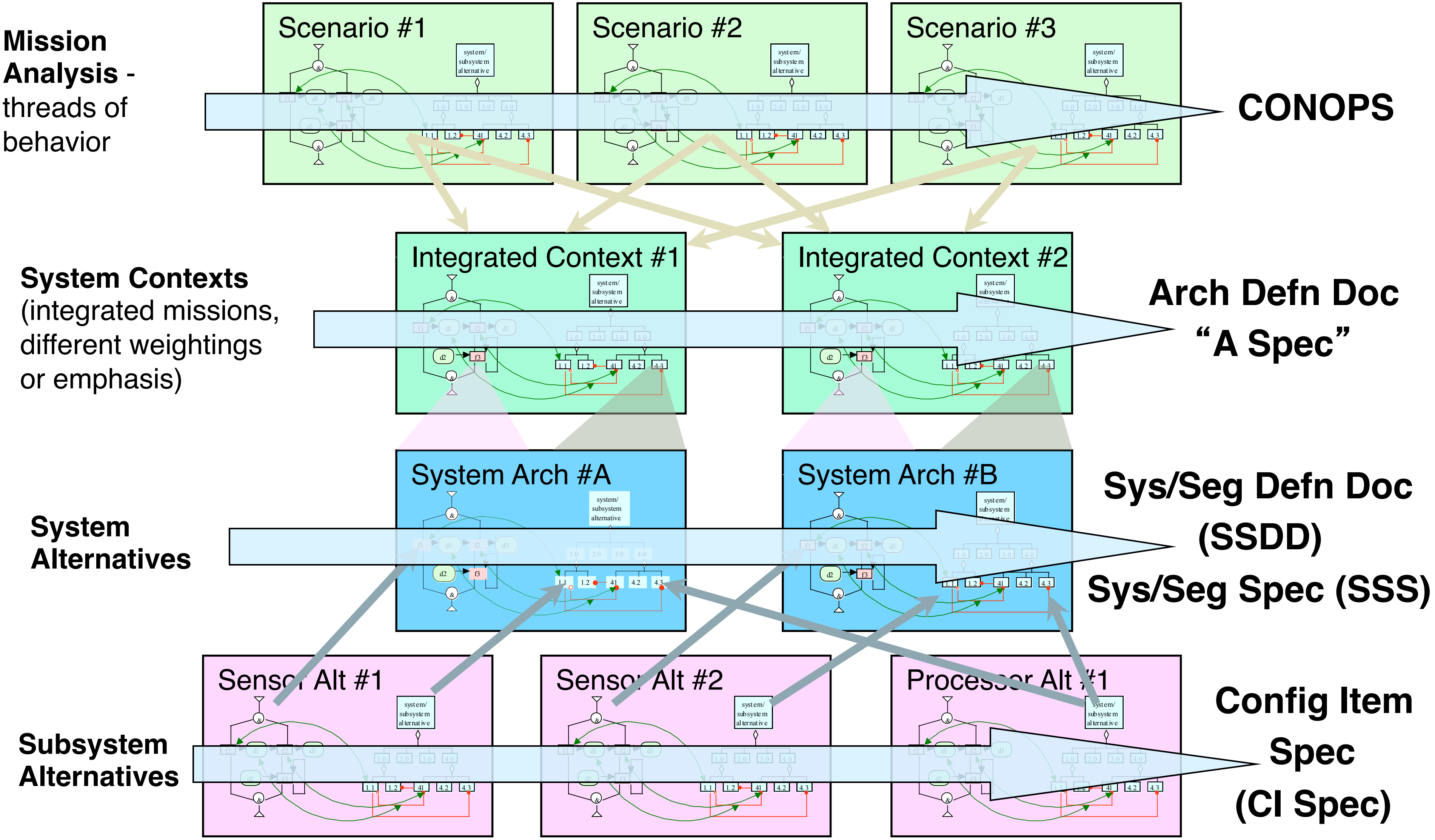
- Each level of decomposition forms the context for the next level down
  - models at each level rely on inputs & outputs from higher level
- Integration of models occurs periodically
  - some design may occur at lower levels without needing to “roll it all up”
  - integration is critical part of change management process



```
<TITLE>System Design<TITLE>
<META http-equiv="REFRESH"
<!--CSSDATA:966533483-->
<SCRIPT src="/virtual/2000/code
<LINK rel="stylesheet" href="/
<SCRIPT language="javascript"
```

© Elsevier Inc 2008

A System Model Should Map to Accepted Levels of Abstraction



Mission Analysis - threads of behavior

Scenario #1

Scenario #2

Scenario #3

CONOPS

System Contexts (integrated missions, different weightings or emphasis)

Integrated Context #1

Integrated Context #2

Arch Defn Doc “A Spec”

System Alternatives

System Arch #A

System Arch #B

Sys/Seg Defn Doc (SSDD)

Sys/Seg Spec (SSS)

Subsystem Alternatives

Sensor Alt #1

Sensor Alt #2

Processor Alt #1

Config Item Spec (CI Spec)

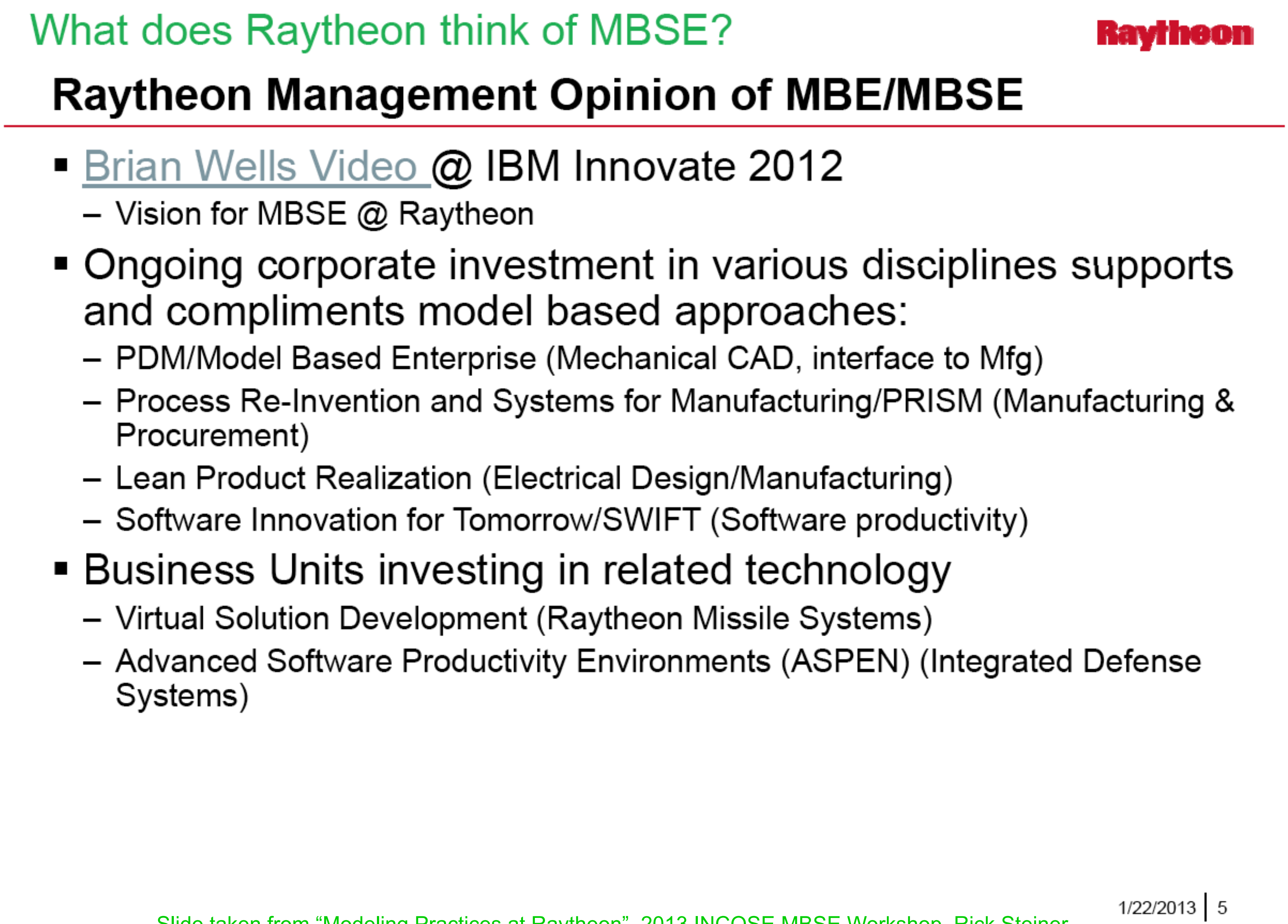
MBSE Industry Perspective:

- 2009 Object Management Group (OMG) Request for Info re. MBSE
  - 128 responders across industry, mostly large scale mil/aero companies
  - MBSE Benefit to project (on scale of 5.0)
    - Value to SE (4.2), Satisfactory method (3.8), Overall benefit (3.9)
    - Most benefit (4.3) when introduced **pre/proposal**, but some benefit (3.4) after project underway
  - MBSE **currently used on smaller projects** (<100 people, <10 modelers)
    - 55% of responders currently using MBSE to improve quality of systems engineering
    - Only 5% have currently incorporated it into policy, but >70% intend to in future
  - Biggest inhibitors to implementing MBSE: **culture, learning curve**
    - Training the team seems to partially mitigate inhibitors, increase internal & customer support
- Opportunities in MBSE application
  - Performance analysis integration with system design/architecture
    - Rapidly growing field, great potential for accelerating system development & increasing quality
  - Intelligent transition from system model to software model (not "flip a switch"!)
    - System model perceived as "requirements", different emphasis than SW architecture.
  - Test engineering has not yet embraced MBSE
    - potentially **huge** benefit (test scenarios, integrated test architectures, etc.)

Open Literature application of MBSE

- INCOSE MBSE Wiki
- Defense
  - Raytheon
  - Lockheed Martin
  - Boeing
  - Others to varying degrees
- Space
  - JPL: Europa project
  - European Space Agency: ESO Telescope Modeling
- Automotive
  - Ford

What does Raytheon think of MBSE?



Raytheon Management Opinion of MBE/MBSE

- Brian Wells Video @ IBM Innovate 2012
  - Vision for MBSE @ Raytheon
- Ongoing corporate investment in various disciplines supports and compliments model based approaches:
  - PDM/Model Based Enterprise (Mechanical CAD, interface to Mfg)
  - Process Re-Invention and Systems for Manufacturing/PRISM (Manufacturing & Procurement)
  - Lean Product Realization (Electrical Design/Manufacturing)
  - Software Innovation for Tomorrow/SWIFT (Software productivity)
- Business Units investing in related technology
  - Virtual Solution Development (Raytheon Missile Systems)
  - Advanced Software Productivity Environments (ASPEN) (Integrated Defense Systems)

Slide taken from “Modeling Practices at Raytheon”, 2013 INCOSE MBSE Workshop, Rick Steiner

What does Raytheon think of MBSE?

The Model is the [System] Design

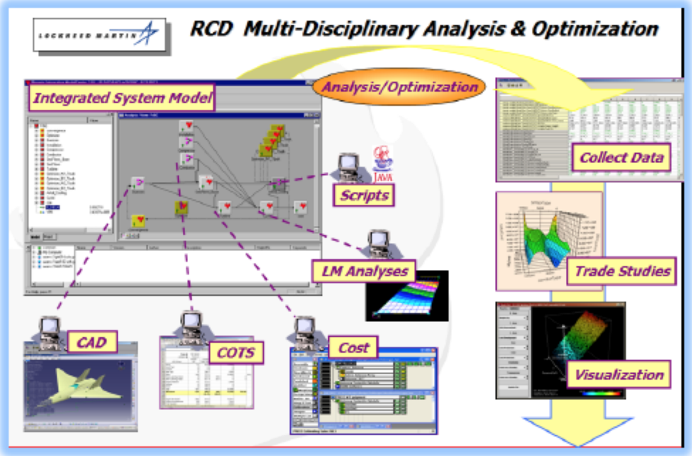
- Capture the information once
- Models are the design, there is no other documentation
  - Replace the documents and drawings with a model based description of all aspects of the design
- All aspects of the design are captured and integrated
- Models are automatically cross connected and compatible
- Design changes and updates are applied only to the models
- The Design is maintained by maintaining the models

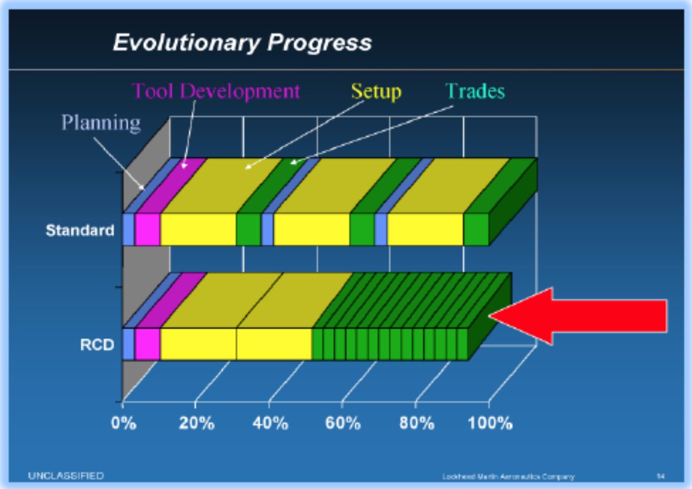
Revolutionize the methods for defining designs

Slide taken from “The Challenges of MBE and MBSE”, Brian Wells, INCOSE New England MBSE Workshop 2011

Lockheed Martin Case Study
Aeronautics Rapid Conceptual Design

- Integrated analysis framework used to maximize the number of alternatives analyzed
- Leverages COTS and heritage homegrown analysis tools
- “Design point runs were conducted in a 20 hour period whereas using conventional methods, these trades would have taken weeks. The end result was a vehicle whose size was reduced by 33%”

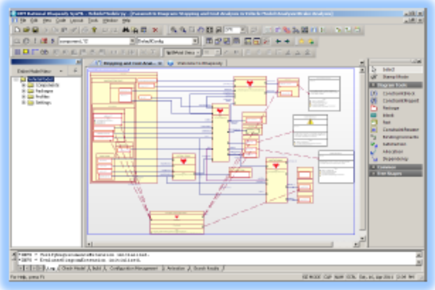


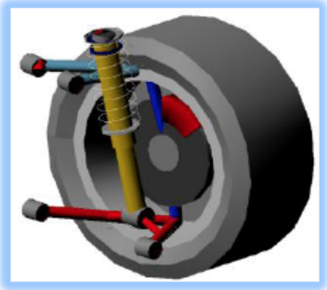


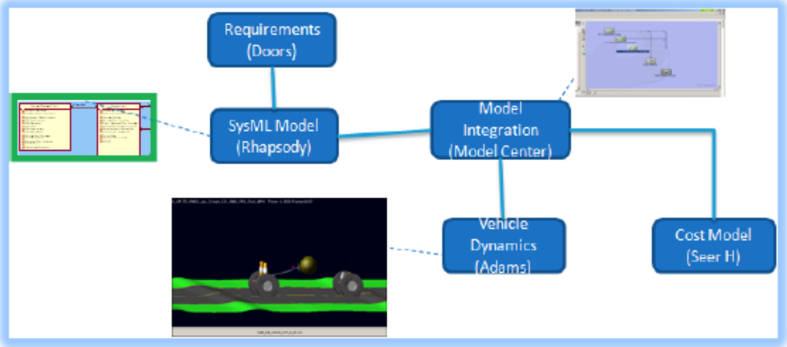
Slide taken from “Evolving Lockheed Martin’s Engineering Practices through the Creation of a Model-Centric Digital Tapestry”, 2011 Frontiers in MBSE Workshop (Georgia Tech), Christopher Oster

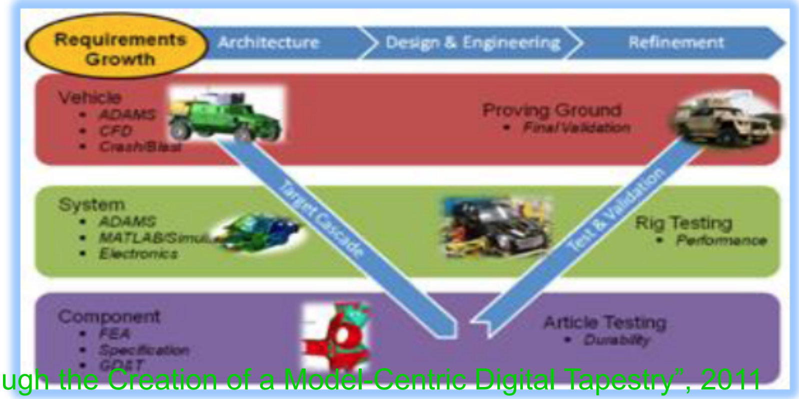
Lockheed Martin Case Study
SysML Driven Subsystem Optimization

- Pilot Objective
  - Elaborate the modeling capability to demonstrate robust integration of requirements, analysis and design
  - Leverage Design of Experimentation techniques to analyze the design trade space
- Pilot Challenges
  - Developing a Suspension System that meets ride and handling qualities under the full range of loading conditions at minimum cost and weight
  - Considering conventional, adjustable, active and position dependent shock absorber designs
- Demonstrated integration of Rhapsody, SEER, Matlab, Excel and MSC.Adams using Model Center
  - Focused on verifying an enhanced ability to investigate performance









Slide taken from "Evolving Lockheed Martin's Engineering Practices through the Creation of a Model-Centric Digital Tapestry", 2011 Frontiers in MBSE Workshop (Georgia Tech), Christopher Oster

LOCKHEED MARTIN
ENGINEERING

Boeing Enterprise Approach to MBSE Implementation

The Boeing Company

- **Education on the topic of MBSE**
  - Reestablishment of basic SE beyond requirements management
  - Importance of functional models (Carson & Sheeley, INCOSE 2013)
- **Development of MBSE capability for programs to use**
- **Development of guidance for how to use the MBSE developed capability**
- **A core group that provides support to all programs**
- **Means to capture and share successes and lessons learned**

Slide taken from "MBSE Implementation Across Diverse Domains at The Boeing Company", 2014 INCOSE MBSE Workshop, Barbara Sheeley, Robert Malone, John Palmer, Ron Carson

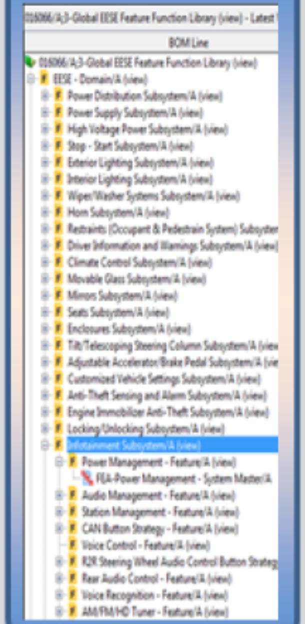
Systems Engineering MB Framework

ONE FORD
ONE TEAM • ONE PLAN • ONE GOAL

Systems Engineering Work Products

- FMEA
- P-Diagrams
- Boundary Diagrams
- Interface Matrix
- Noise Factor Mangmt Strategy
- DVP
- Safety Plan
- Prelim Hazard Analysis
- Functional Safety Concept

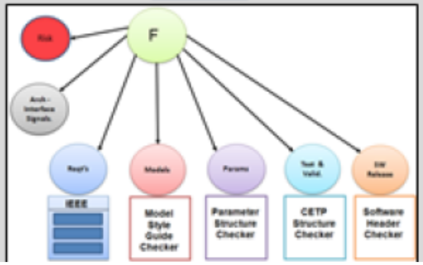
Feature Dictionary



016066/A;3-Global EESE Feature Function Library (view) - Latest

BOM Line

```
016066/A;3-Global EESE Feature Function Library (view)
  EESE - Domain/A (view)
    Power Distribution Subsystem/A (view)
    Power Supply Subsystem/A (view)
    High Voltage Power Subsystem/A (view)
    Stop - Start Subsystem/A (view)
    Exterior Lighting Subsystem/A (view)
    Interior Lighting Subsystem/A (view)
    Wiper/Washer Systems Subsystem/A (view)
    Horn Subsystem/A (view)
    Restraints (Occupant & Pedestrain System) Subsystem
    Driver Information and Warnings Subsystem/A (view)
    Climate Control Subsystem/A (view)
    Movable Glass Subsystem/A (view)
    Mirrors Subsystem/A (view)
    Seats Subsystem/A (view)
    Enclosures Subsystem/A (view)
    Tilt/Telescoping Steering Column Subsystem/A (view
    Adjustable Accelerator/Brake Pedal Subsystem/A (vie
    Customized Vehicle Settings Subsystem/A (view)
    Anti-Theft Sensing and Alarm Subsystem/A (view)
    Engine Immobilizer Anti-Theft Subsystem/A (view)
    Locking/Unlocking Subsystem/A (view)
    Infotainment Subsystem/A (view)
      Power Management - Feature/A (view)
        FEA-Power Management - System Master/A
      Audio Management - Feature/A (view)
      Station Management - Feature/A (view)
      CAN Button Strategy - Feature/A (view)
      Voice Control - Feature/A (view)
      R2R Steering Wheel Audio Control Button Strateg
      Rear Audio Control - Feature/A (view)
      Voice Recognition - Feature/A (view)
      AM/FM/HD Tuner - Feature/A (view)
```



Functional Architecture (Implementation Independent)

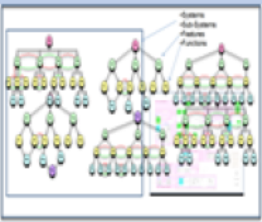
Virtual, Model Based Analysis;

Dependencies;

Coupling; Cohesion

& Arch Quality

Attribute Studies



Logical Architecture (Re-usable Reference Architectures)

Commodity Specific Reference Architectures

Sync Gen 2 Ref Arch V1.0

BCM Ref Arch V1.x

Physical Architectures (Vehicle Program Specific)

Vehicle Program Specific Architectures

Global Signal Database (GSDB)

- Signal Definitions - Interface Specs
  - -CAN Message relationship
- Global Device Transmittal Database (GDT)
  - SW-HW interfaces
  - EE Devices – Sensors - Actuators

Slide taken from “Automotive Software Systems Complexity”, 2013 INCOSE MBSE Workshop, Chris Davey

Process, Method, Methodology

- A **process*** describes a sequence of tasks to achieve an objective, e.g. Company Standard Process (Stage 1, Stage 2, etc.)
- A **method*** describes techniques for performing one or more tasks, e.g. program specific work instructions, design guides
- A **tool*** facilitates one or more method, using one or more *languages*, e.g. IBM Rhapsody, System Architect, RS(X), Sparx Enterprise Architect
- A **methodology*** is a collection of related processes, methods and tools, e.g. IBM Harmony, Rational Unified Process, OPM****, INCOSE Object Oriented Systems Engineering Method (OOSEM)

Note: Tools are Enablers, NOT MBSE Drivers…

Tool decisions should come AFTER process and method decisions.

* Survey of MBSE Methodologies, J. Estefan, INCOSE-TD-2007-003-01

Language

- A **language** establishes basic **rules** (syntax) for communicating…
  - The rules for graphically expressing these concepts (on diagrams) are called the **Concrete Syntax** of the language
    - This is like constraints on a rendering engine, including standard symbology
  - The rules for constructing and relating **concepts** are called the **Abstract Syntax** of the language
    - This is like constraints on an ERA database, including allowable relationships
    - Languages with weak abstract syntax are inappropriate for modeling
  - Examples for MBSE include SDL**, OMT***, OPL****, IDEF, UML, and SysML
- Limitations of the language will inhibit communication, and obscure inconsisencies

** Model Based Systems Engineering, Wymore ISBN 084938012X

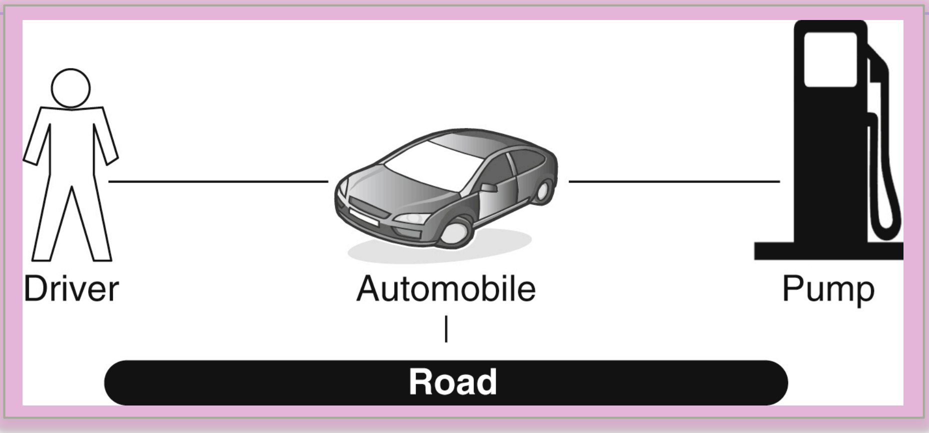
*** Object Oriented Modeling and Design, Rumbagh et al ISBN 0136298419

**** Object Process Methodology, Dori ISBN 3540654712

From Concept to Specification/Model

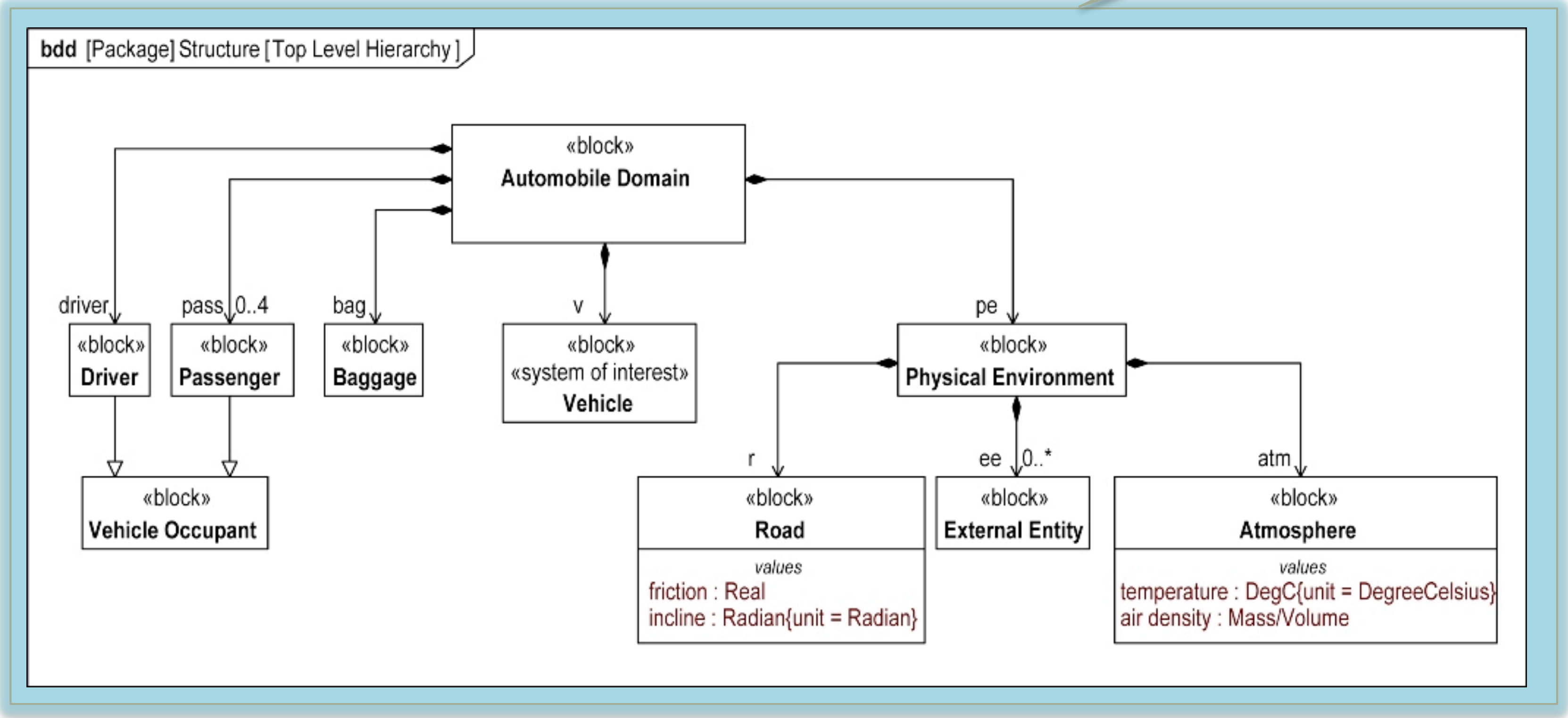
- Concept art (e.g. Powerpoint) and abstract requirements usually exist before specifications or models
- The codification of concepts into a system model provides a framework for analysis, specification, verification data
  - Automotive example follows this chart…
- Once system data is in this model framework, it can
  - Serve as a technical baseline for generating specifications
  - Serve as a basis for generating and evaluating trade studies
  - Provide contextualized parameters for more detailed system analysis
  - Provide context and specification for each component to be developed

Establishing a Domain/Context

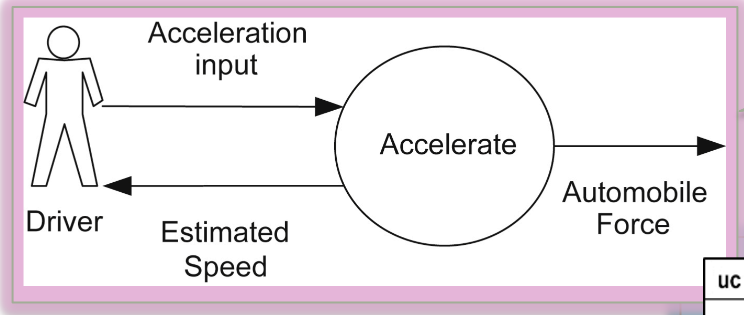


Concept Art/ Cartoon

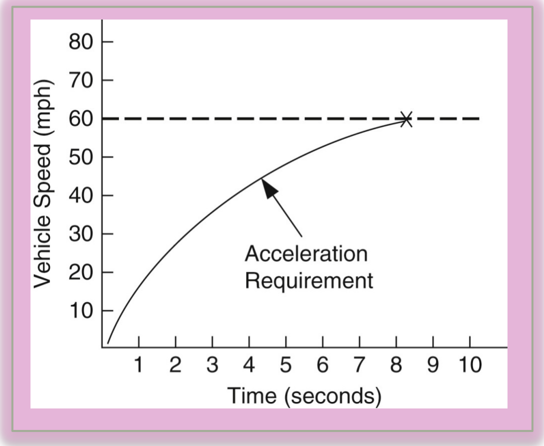
System Model



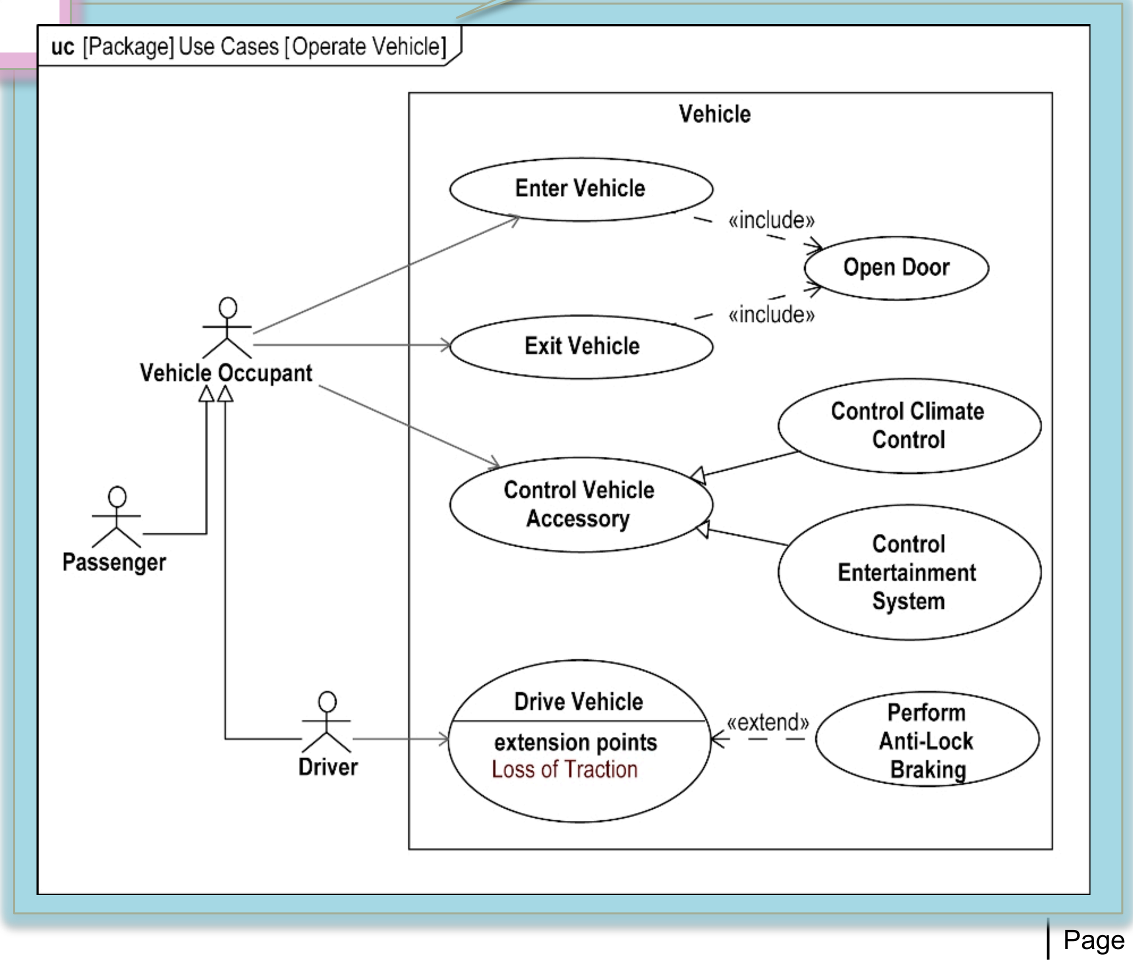
Setting System Goals & Needs



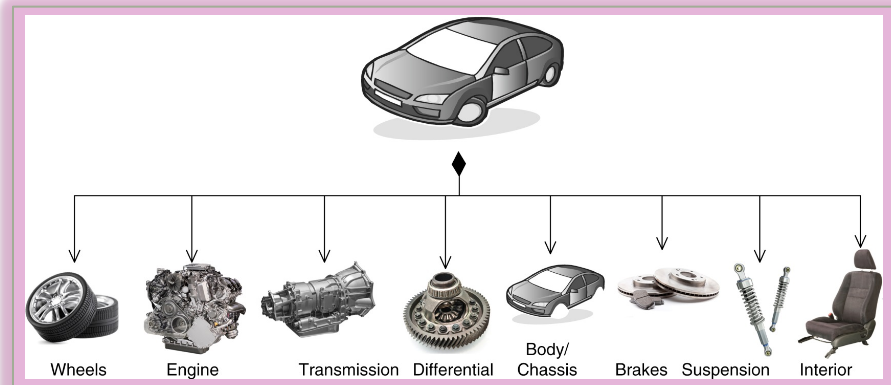
Concept Art/ Cartoon



System Model

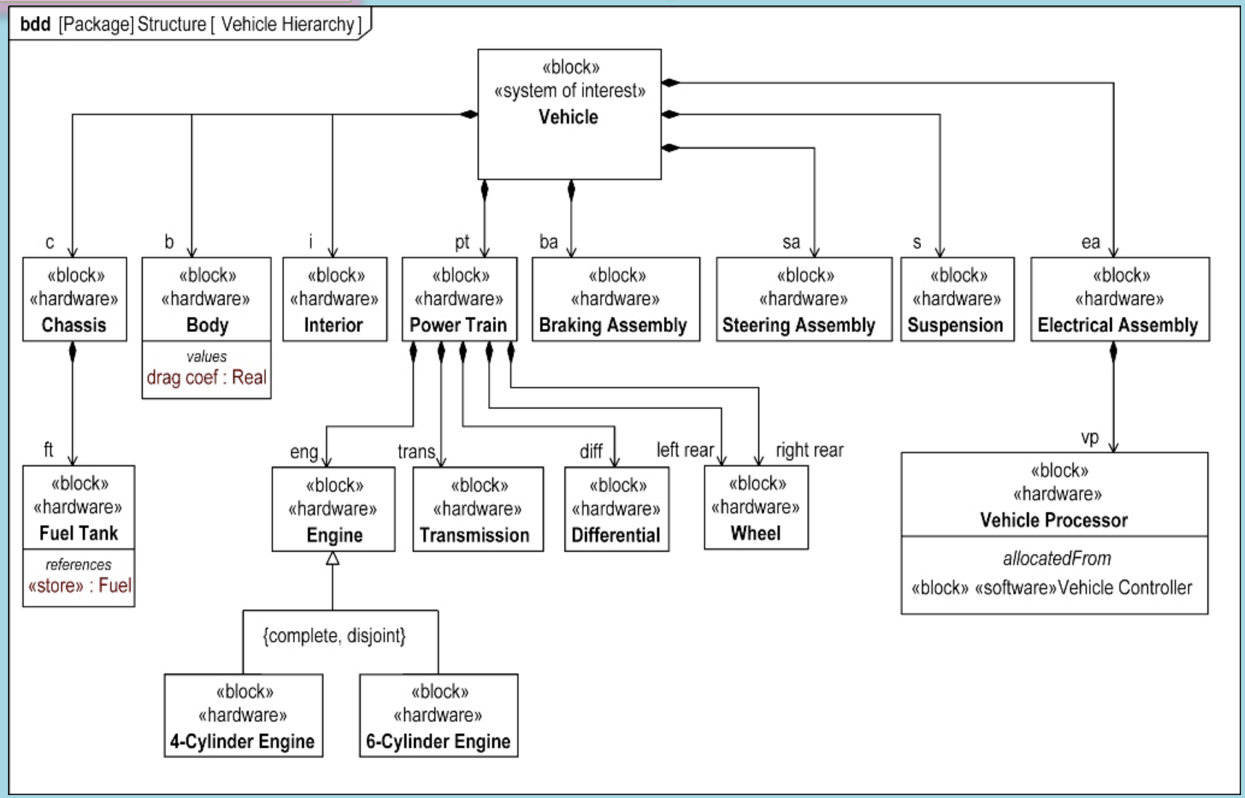


Establishing Structural Hierarchy



Concept Art/
Cartoon

System
Model



bdd [Package] Structure [Vehicle Hierarchy]

«block»
«system of interest»
Vehicle

c
«block»
«hardware»
Chassis

b
«block»
«hardware»
Body
values
drag coef : Real

i
«block»
«hardware»
Interior

pt
«block»
«hardware»
Power Train

ba
«block»
«hardware»
Braking Assembly

sa
«block»
«hardware»
Steering Assembly

s
«block»
«hardware»
Suspension

ea
«block»
«hardware»
Electrical Assembly

ft
«block»
«hardware»
Fuel Tank
references
«store» : Fuel

eng
«block»
«hardware»
Engine

trans
«block»
«hardware»
Transmission

diff
«block»
«hardware»
Differential

left rear
right rear
«block»
«hardware»
Wheel

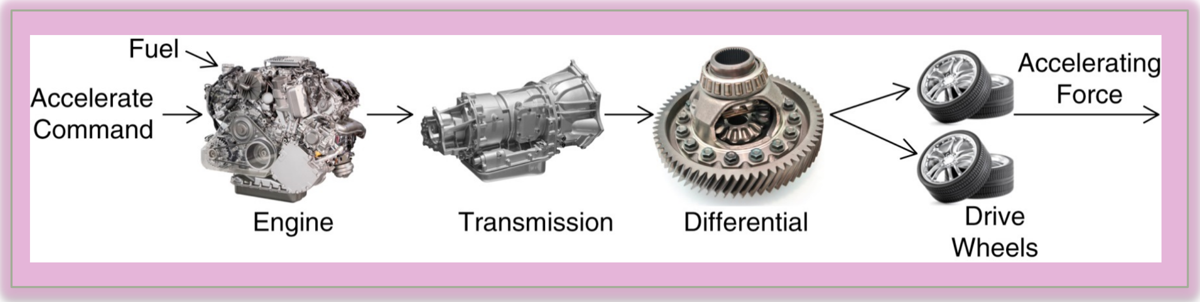
vp
«block»
«hardware»
Vehicle Processor
allocatedFrom
«block» «software»Vehicle Controller

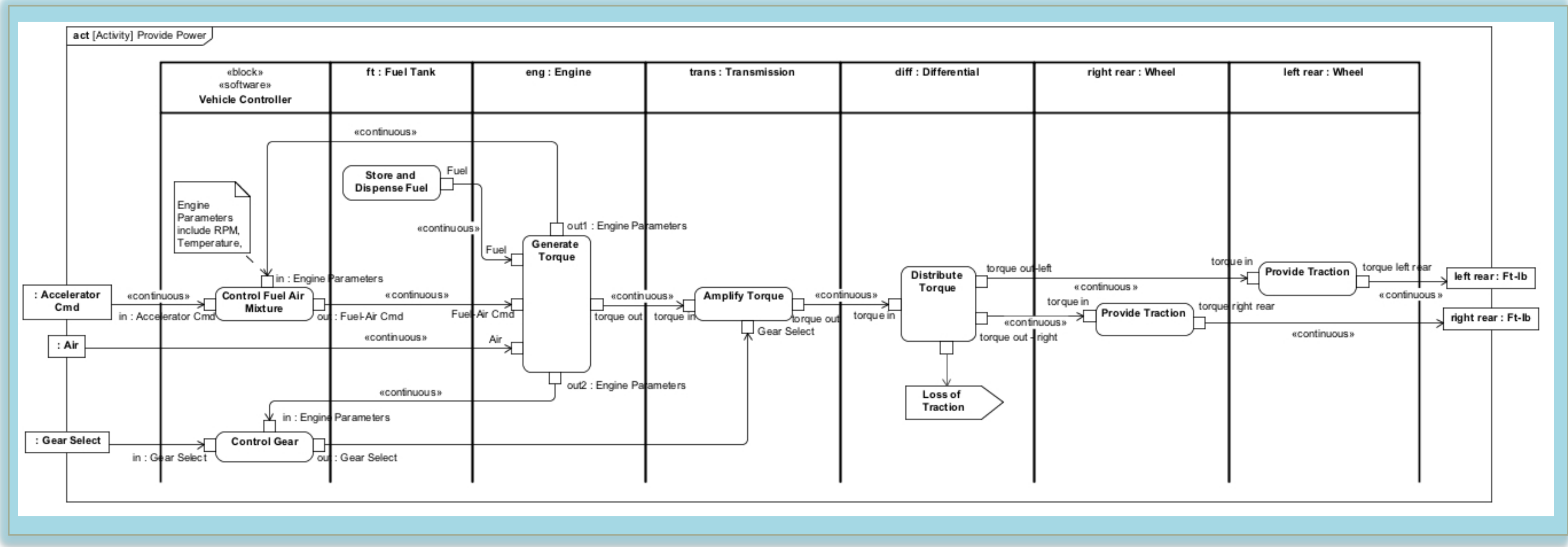
{complete, disjoint}

«block»
«hardware»
4-Cylinder Engine

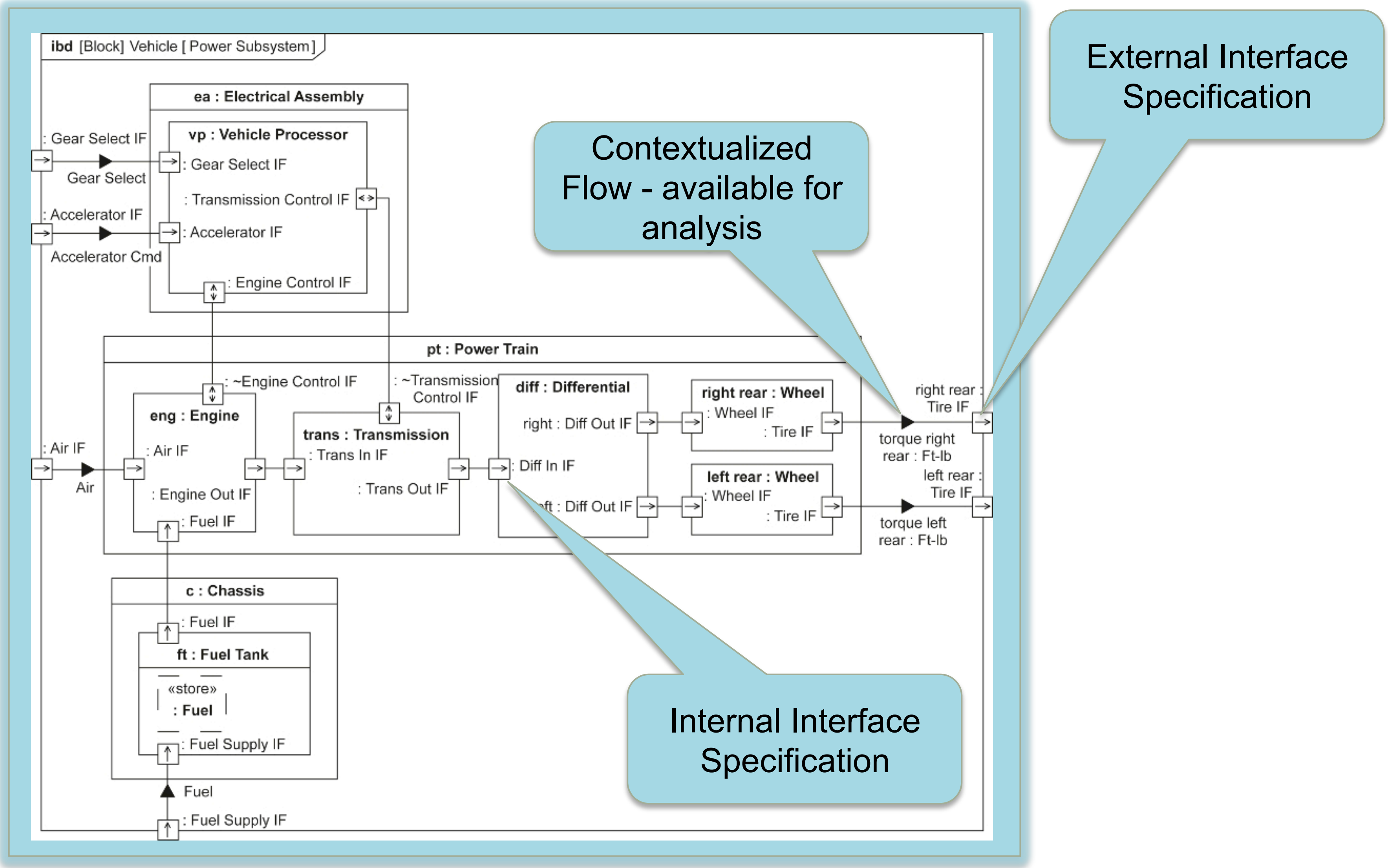
«block»
«hardware»
6-Cylinder Engine

Analyzing Initial Functionality





Specifying Top Level Internal Interfaces

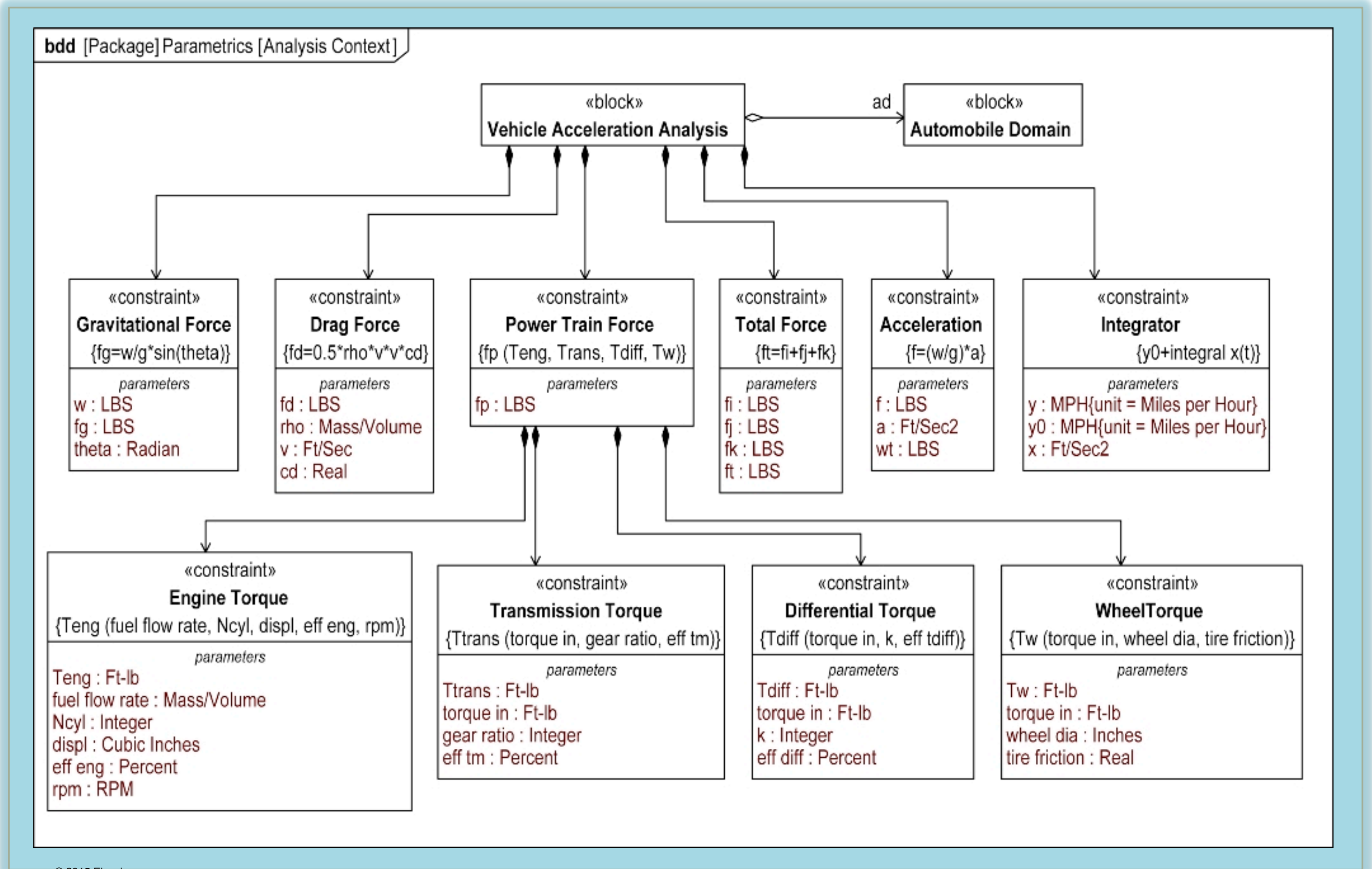


External Interface Specification

Contextualized Flow - available for analysis

Internal Interface Specification

Establishing Analysis Context & Key Constraints



bdd [Package] Parametrics [Analysis Context]

«block» Vehicle Acceleration Analysis

ad

«block» Automobile Domain

«constraint» Gravitational Force {fg=w/g*sin(theta)}
parameters
w : LBS
fg : LBS
theta : Radian

«constraint» Drag Force {fd=0.5*rho*v*v*cd}
parameters
fd : LBS
rho : Mass/Volume
v : Ft/Sec
cd : Real

«constraint» Power Train Force {fp (Teng, Trans, Tdiff, Tw)}
parameters
fp : LBS

«constraint» Total Force {ft=fi+fj+fk}
parameters
fi : LBS
fj : LBS
fk : LBS
ft : LBS

«constraint» Acceleration {f=(w/g)*a}
parameters
f : LBS
a : Ft/Sec2
wt : LBS

«constraint» Integrator {y0+integral x(t)}
parameters
y : MPH{unit = Miles per Hour}
y0 : MPH{unit = Miles per Hour}
x : Ft/Sec2

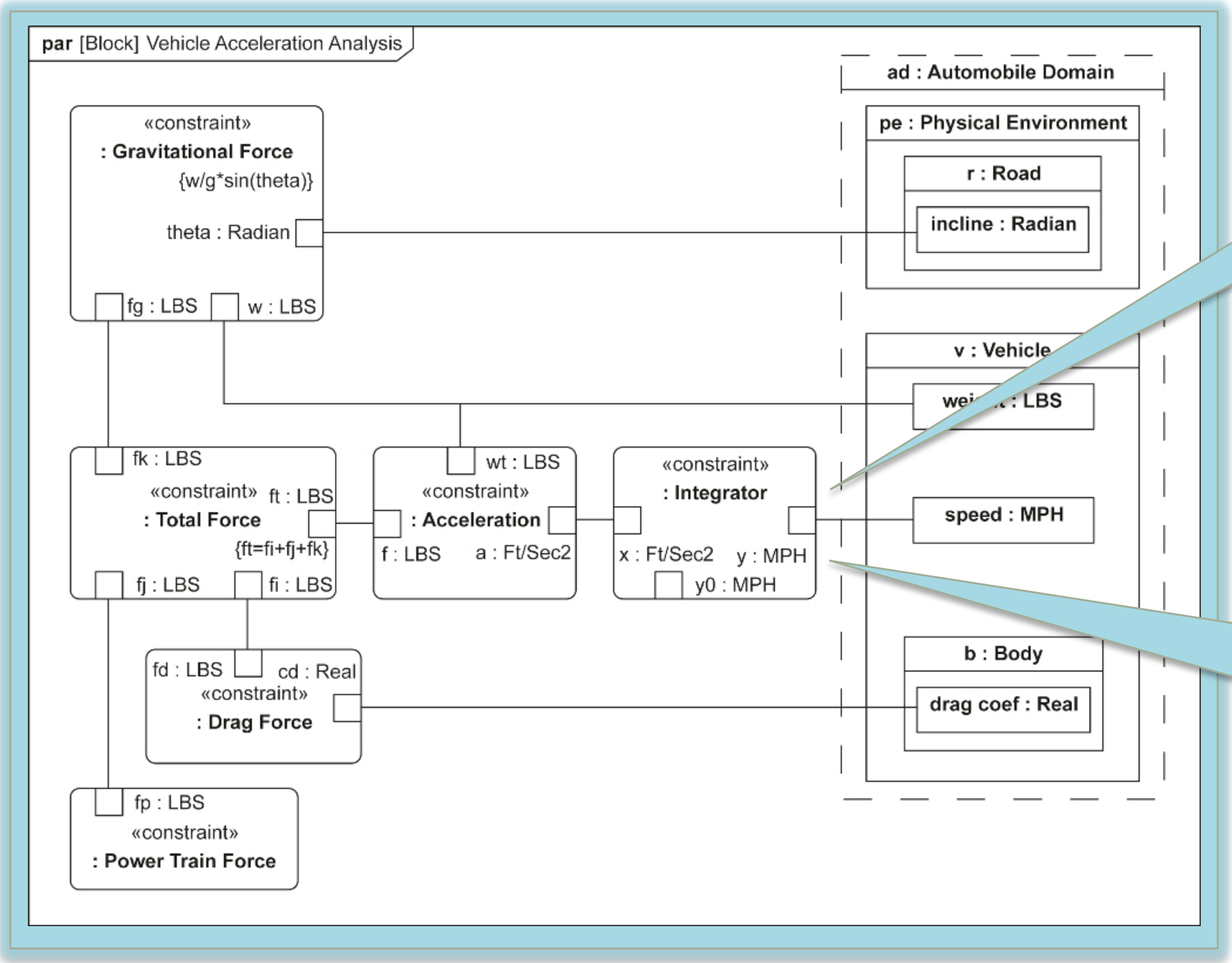
«constraint» Engine Torque {Teng (fuel flow rate, Ncyl, displ, eff eng, rpm)}
parameters
Teng : Ft-lb
fuel flow rate : Mass/Volume
Ncyl : Integer
displ : Cubic Inches
eff eng : Percent
rpm : RPM

«constraint» Transmission Torque {Ttrans (torque in, gear ratio, eff tm)}
parameters
Ttrans : Ft-lb
torque in : Ft-lb
gear ratio : Integer
eff tm : Percent

«constraint» Differential Torque {Tdiff (torque in, k, eff tdiff)}
parameters
Tdiff : Ft-lb
torque in : Ft-lb
k : Integer
eff diff : Percent

«constraint» WheelTorque {Tw (torque in, wheel dia, tire friction)}
parameters
Tw : Ft-lb
torque in : Ft-lb
wheel dia : Inches
tire friction : Real

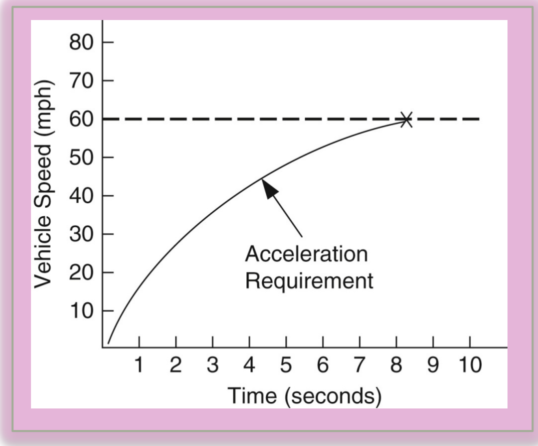
Binding Key Equations to Key Parameters



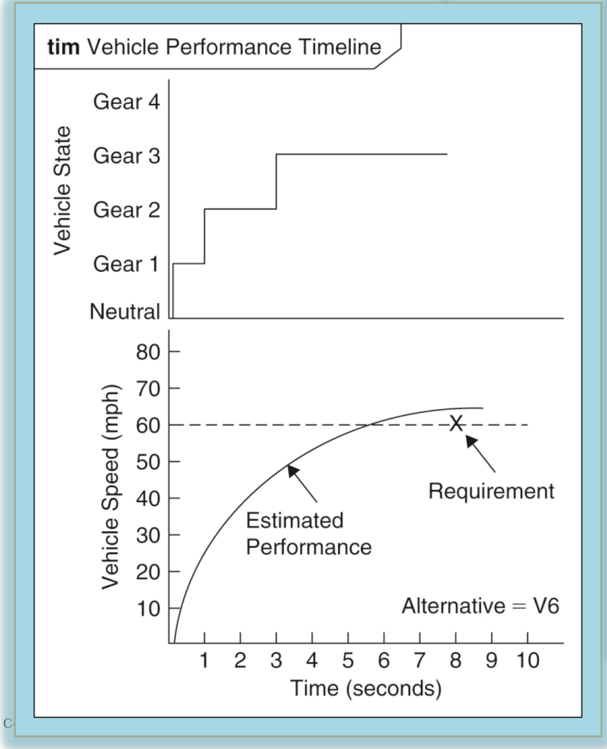
SysML is not intended to solve these sets of equations…

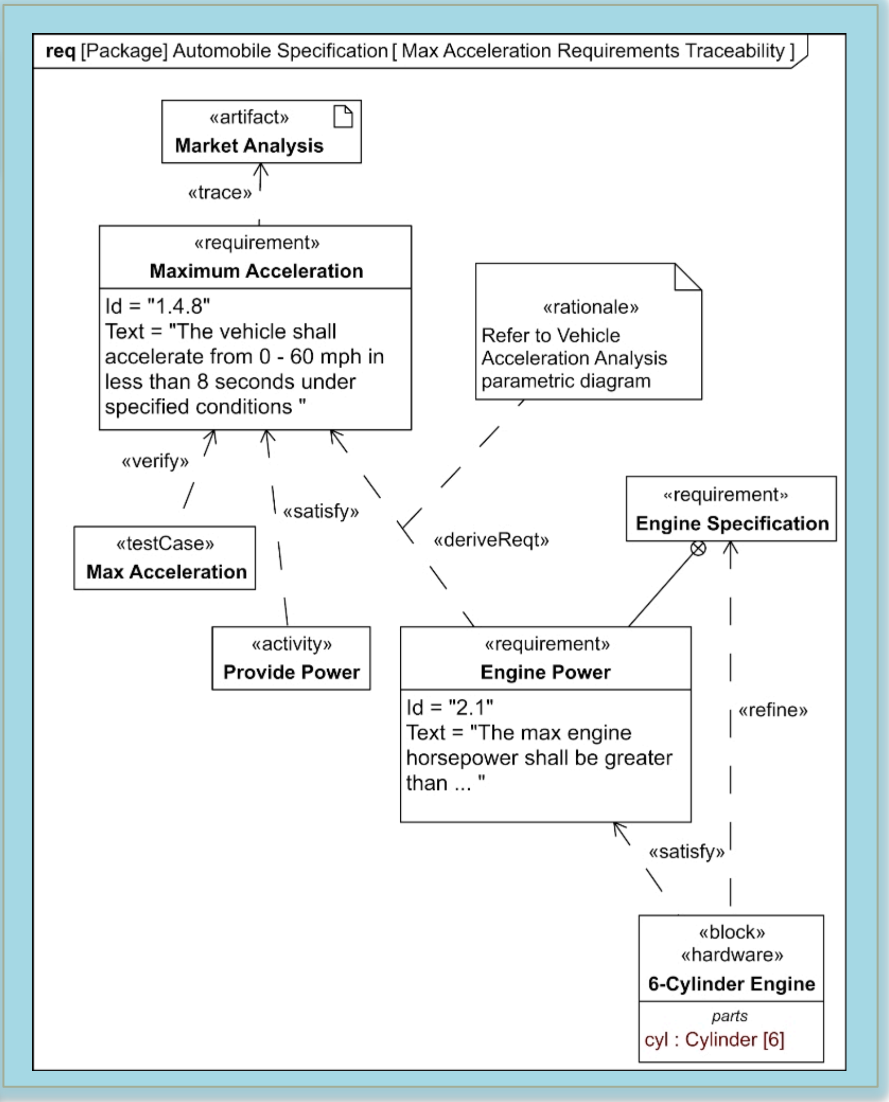
But SysML tools interface to standard solvers (Matlab, Mathematica, …)

Evaluating Performance to Requirements



Results of solving equations/ simulations





req [Package] Automobile Specification [Max Acceleration Requirements Traceability]

«artifact»
Market Analysis

«trace»

«requirement»
Maximum Acceleration

Id = "1.4.8"
Text = "The vehicle shall accelerate from 0 - 60 mph in less than 8 seconds under specified conditions "

«rationale»
Refer to Vehicle Acceleration Analysis parametric diagram

«verify»

«satisfy»

«deriveReqt»

«requirement»
Engine Specification

«testCase»
Max Acceleration

«activity»
Provide Power

«requirement»
Engine Power

Id = "2.1"
Text = "The max engine horsepower shall be greater than ... "

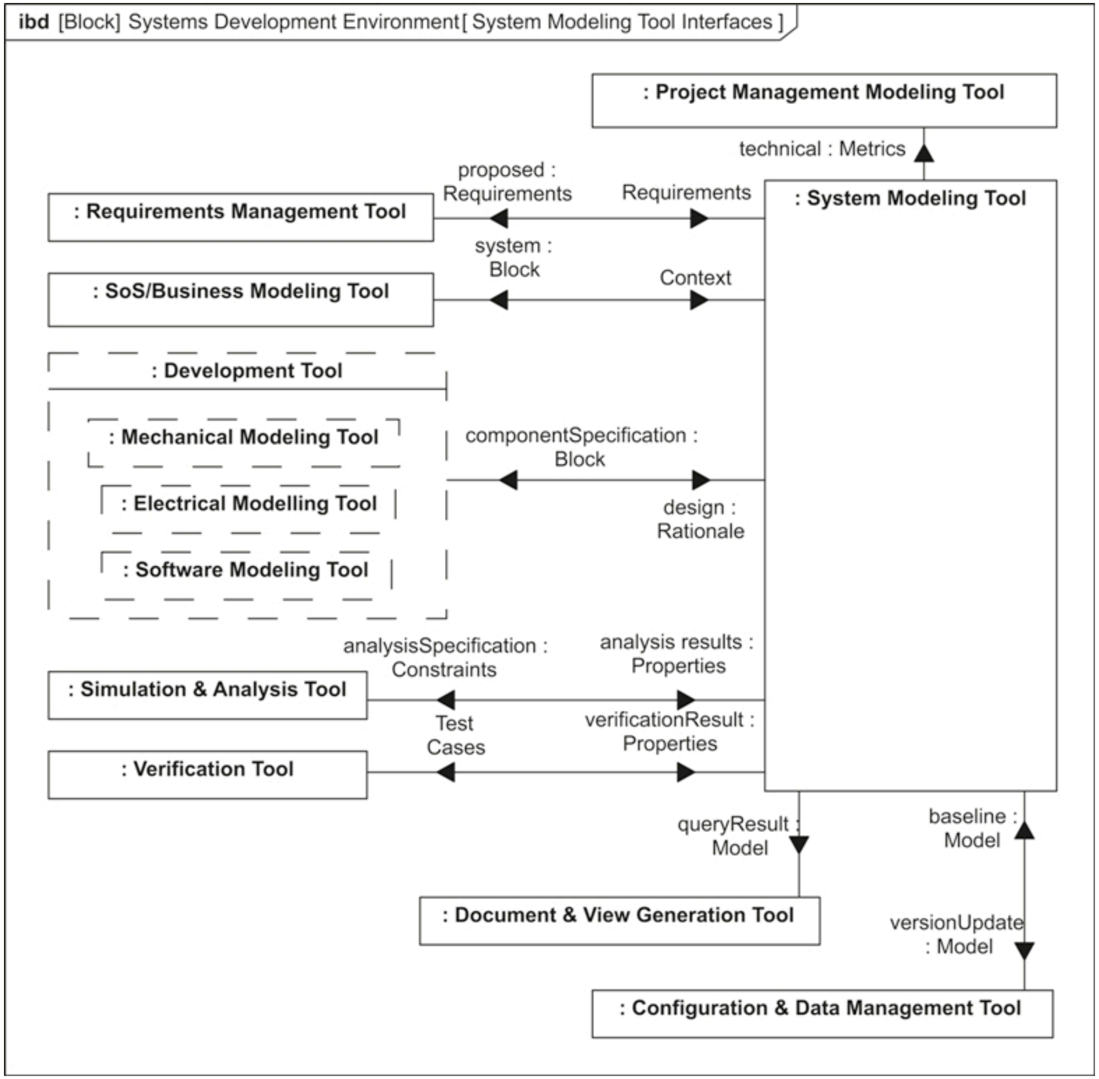
«refine»

«satisfy»

«block»
«hardware»
6-Cylinder Engine

parts
cyl : Cylinder [6]

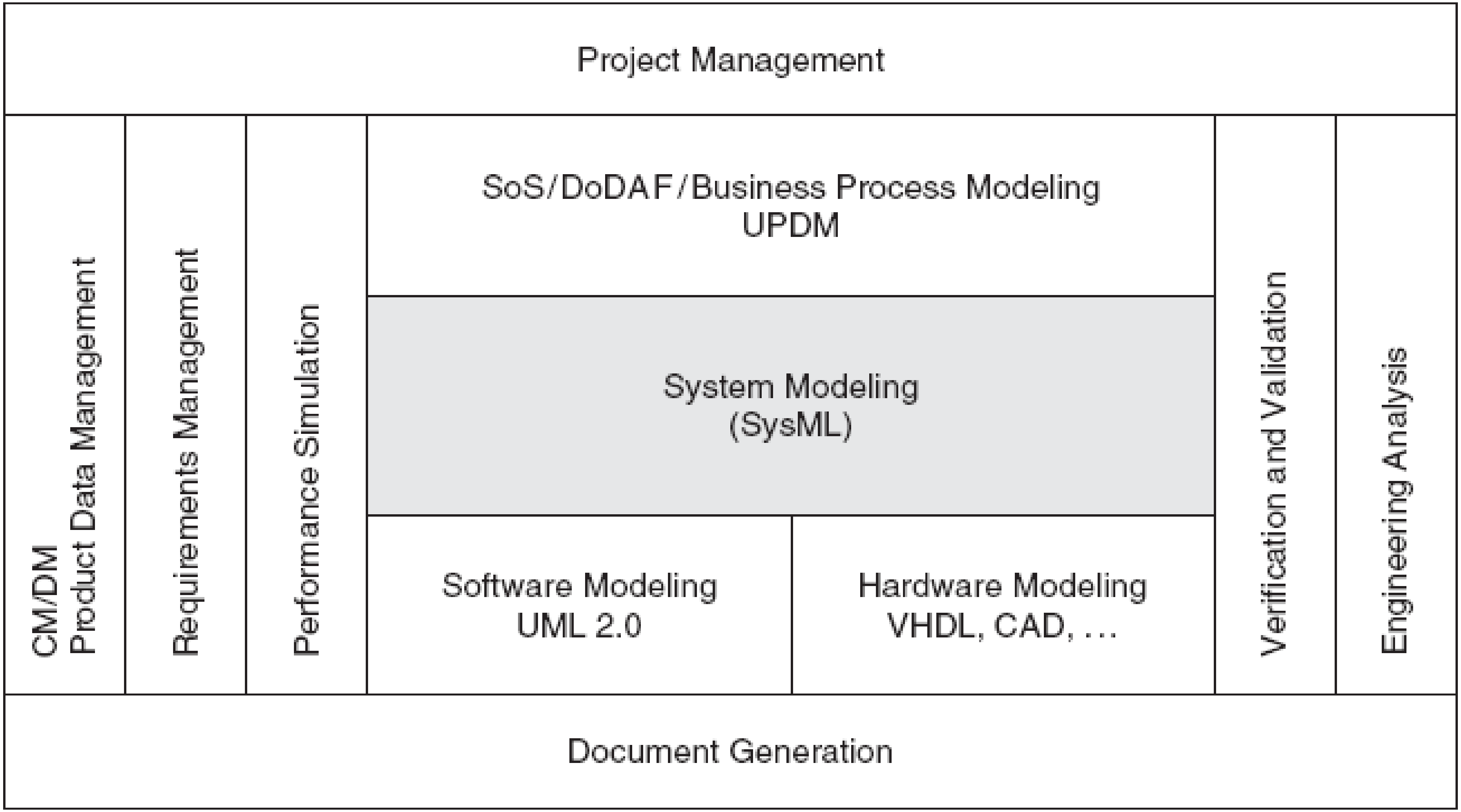
Every MBSE Deployment will need to invest in infrastructure!



ibd [Block] Systems Development Environment[System Modeling Tool Interfaces]
: Project Management Modeling Tool
technical : Metrics
proposed : Requirements
Requirements
: System Modeling Tool
: Requirements Management Tool
system : Block
Context
: SoS/Business Modeling Tool
: Development Tool
: Mechanical Modeling Tool
componentSpecification : Block
: Electrical Modelling Tool
design : Rationale
: Software Modeling Tool
analysisSpecification : Constraints
analysis results : Properties
: Simulation & Analysis Tool
Test Cases
verificationResult : Properties
: Verification Tool
queryResult : Model
baseline : Model
: Document & View Generation Tool
versionUpdate : Model
: Configuration & Data Management Tool

- xmi-based data exchange
- Diagram Interchange standard
- Query, View, Transformation (QVT) standard
- SysML-Modelica transformation spec
- OSLC web-based referencing
- Functional Mockup Interface (FMI)

System Modeling Must Fit With Other Key Project Activities



Project Management

CM/DM
Product Data Management

Requirements Management

Performance Simulation

SoS/DoDAF/Business Process Modeling
UPDM

System Modeling
(SysML)

Software Modeling
UML 2.0

Hardware Modeling
VHDL, CAD, …

Verification and Validation

Engineering Analysis

Document Generation

© Elsevier Inc 2008