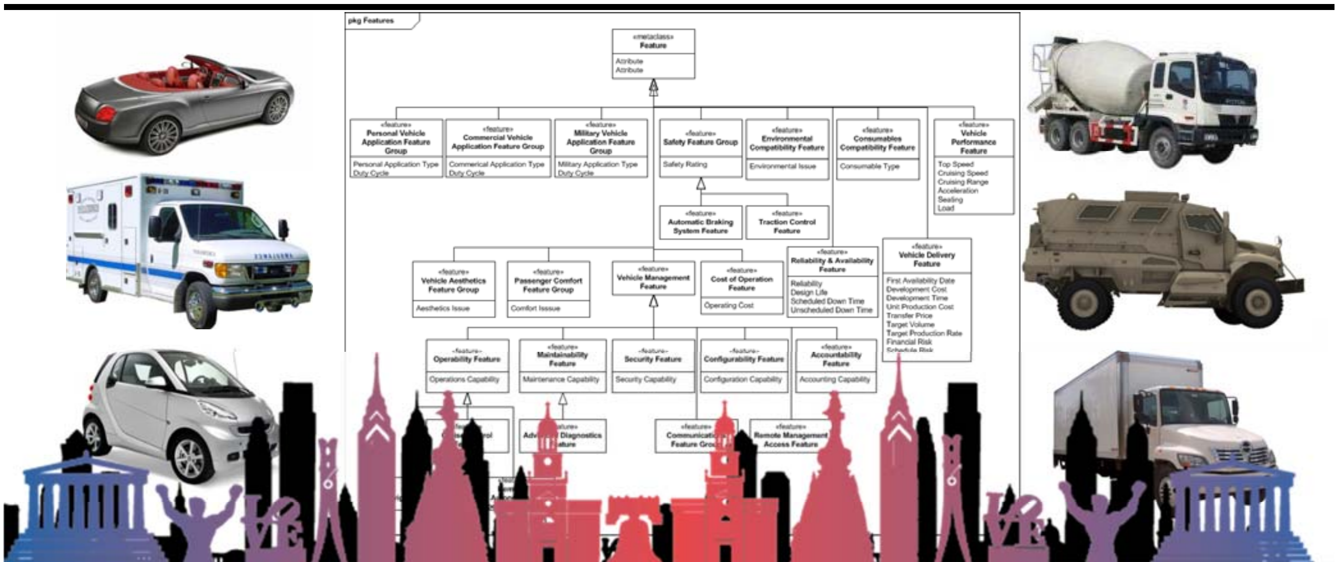
ICTT System Sciences
Bill Schindel
schindel@ictt.com

INCOSE International Symposium
Philadelphia, PA
June 24-27, 2013

Booz | Allen | Hamilton
Troy Peterson
peterson_troy @ bah.com

Introduction to Pattern-Based Systems Engineering (PBSE): Leveraging MBSE Techniques



1.5.3

Abstract

INCOSE International Symposium
Philadelphia, PA
June 24-27, 2013

- This tutorial is a practitioner's introduction to Pattern-Based Systems Engineering (PBSE), including a specific system domain illustration suitable for educational use.
- INCOSE thought leaders have discussed the need to address 10:1 more complex systems with 10:1 reduction in effort, using people from a 10:1 larger community than the "systems expert" group INCOSE currently reaches. Through the PBSE Project, the project team proposes to enable INCOSE membership, and the larger systems community beyond INCOSE, to achieve such order-of-magnitude improvements.
- PBSE leverages the power of Model-Based Systems Engineering (MBSE) to rapidly deliver benefits to a larger community. Projects using PBSE get a "learning curve jumpstart" from an existing Pattern, gaining the advantages of its content, and improve that pattern with what they learn, for future users. The major aspects of PBSE have been defined and practiced some years across a number of enterprises and domains, but with limited INCOSE community awareness.

Contents—Summary

- The need, call-to-arms, and vision
- Conceptual summary of PBSE
- PBSE applications to date
- Representing system patterns: An example
- Applying system patterns: Example uses and benefits
- Challenges and opportunities
- Conclusions

For electronic copy of IS2013 June 26 Tutorial Version:
Contact the authors—email addresses on cover.

- References

Contents—Detail & Timeline

10:00 – 12:10

- The need, call-to-arms, and vision
- Conceptual summary of PBSE
- PBSE applications to date
- Representing system patterns: An example
  - S*Metamodel framework
  - A Vehicle Pattern in SysML
  - A practice exercise

Lunch Break

13:30 – 14:55

- Applying system patterns: Examples of uses and benefits
  1. Stakeholder Features and Scenarios: Better stakeholders alignment sooner
  2. Pattern Configuration: Generating better requirements faster
  3. Selecting Solutions: More informed trades

Coffee Break

15:30 – 16:55

  4. Design for Change: Analyzing and improving platform resiliency
  5. Risk Analysis: Pattern-enabled FMEAs
  6. Verification: Generating better tests and reviews faster
- Challenges and opportunities
  - Human nature & organizations
  - Approaches to my situation
  - Exercise and discussion
- Conclusions

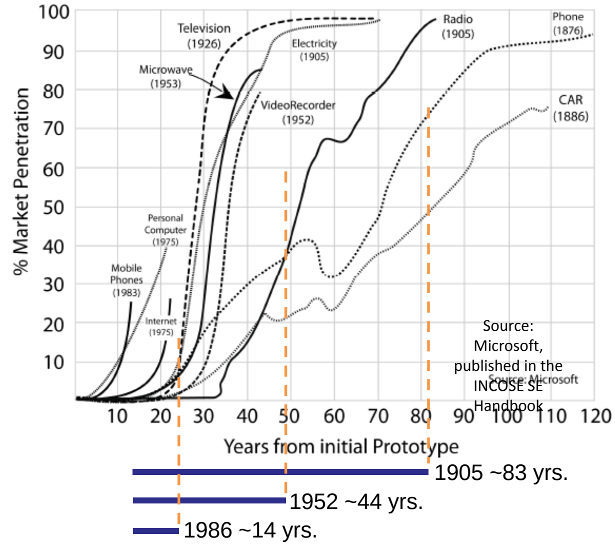
- References

PBSE Addresses Speed, Leverage, Knowledge

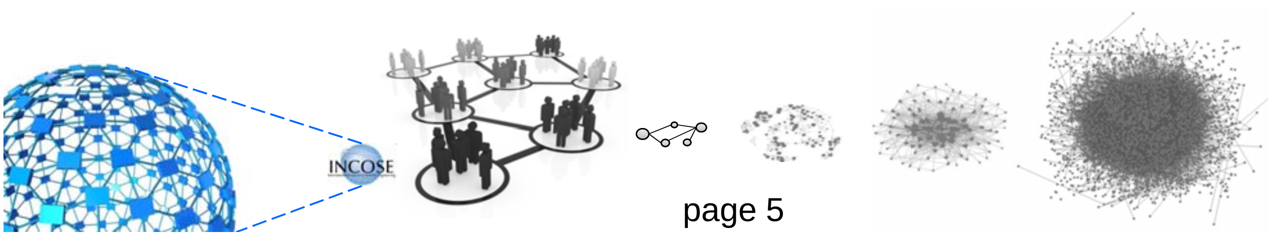
– INCOSE thought leaders have discussed the growing need to address 10:1 more complex systems with 1:10 reduction in time and effort, using people from a 10:1 larger community than the “systems expert” group

– Many SE efforts are in some way concerned with growing complexity, but none give evidence of the sweeping order-of-magnitude improvements demanded by this call-to-arms.

– PBSE is a methodical way to achieve this order-of-magnitude improvement



Rates of system proliferation decreased by 4:1 over 50 years

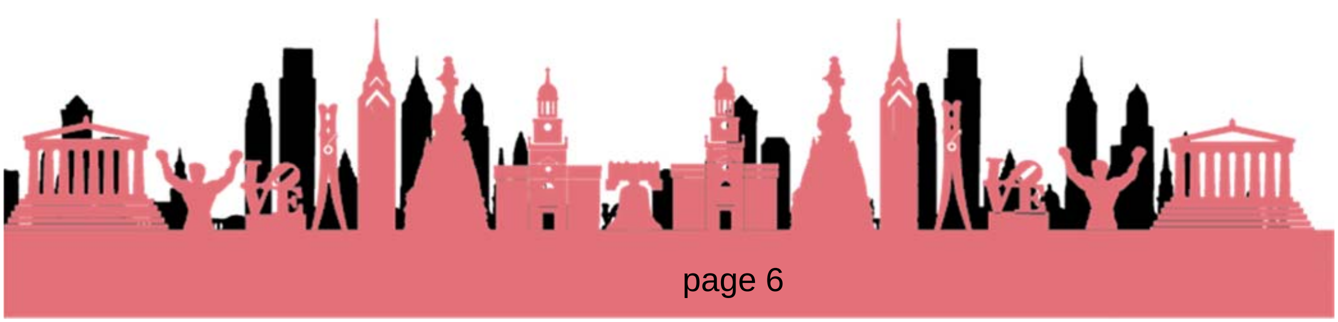


Pattern-Based Systems Engineering (PBSE)

INCOSE International Symposium
Philadelphia, PA
June 24-27, 2013

- What <u>are</u> System Patterns?

- What are System Patterns <u>for</u>?



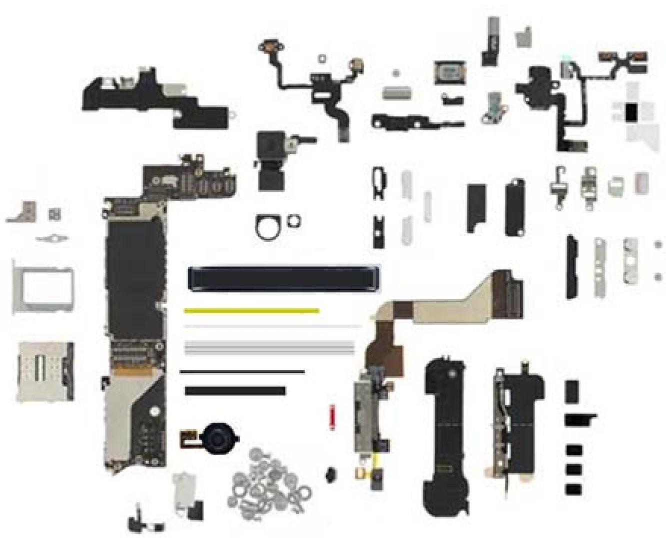
Pattern-Based Systems Engineering (PBSE)

Philadelphia, PA
June 24-27, 2013

- Standard Parts have been a great aid to progress:

- The same part type can be used to make many things!

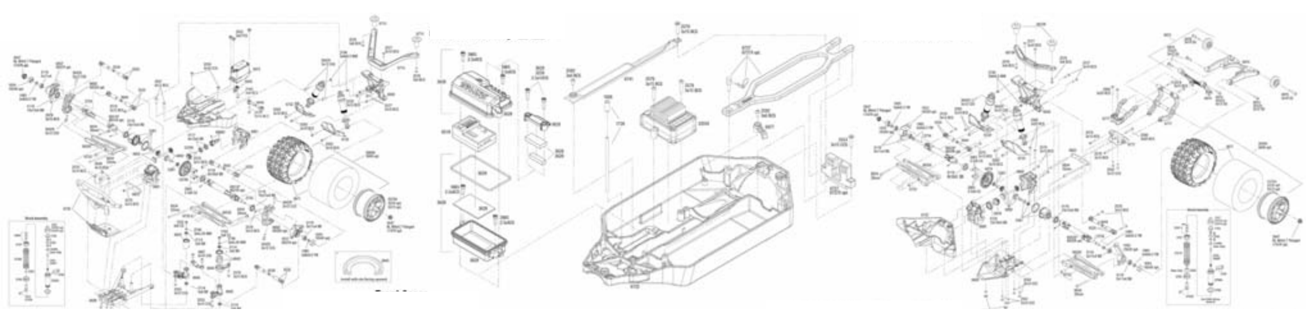
Quick Exercise: Can you recognize this system?

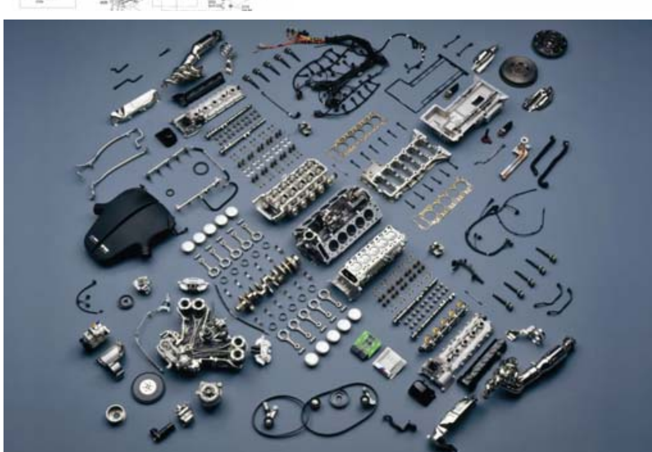


Using different views helps improve recognition: Does rotating the parts improve recognition?

Showing parts in relationship helps recognition

Philadelphia, PA
June 24-27, 2013



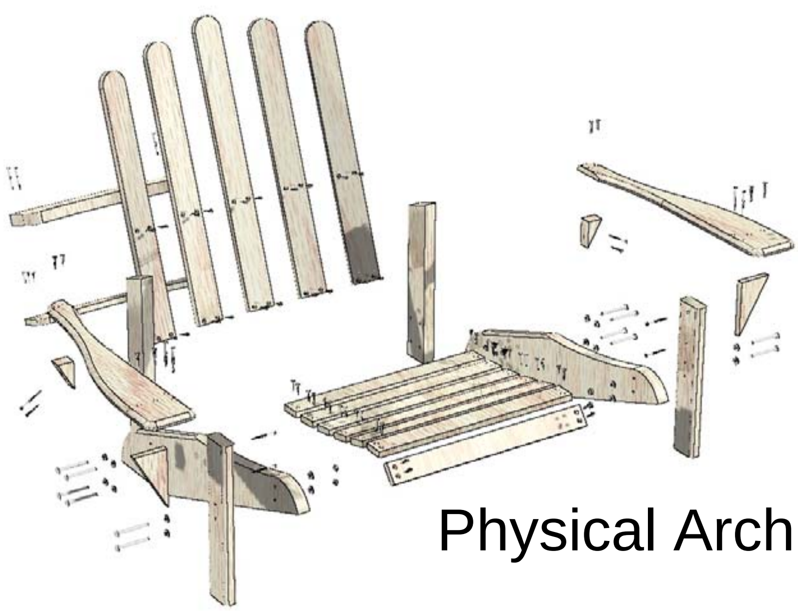


Can we identify a system from its parts alone?

Obviously not in many cases—and in all cases, the parts list alone lacks critical information . . .

Any systems engineer will tell you . . .

- We need to know the relationships between the parts to understand what the "system" they create.



Physical Architecture

INCOSE International Symposium
Philadelphia, PA
June 24-27, 2013

But . . .

we are interested in much more than Physical Architecture:

- Stakeholders
- Requirements
- Design
- Interfaces
- Modes
- Performance
- Failure Modes & Effects
- Verification Plans
- Alternatives
- Configurability
- Manufacturability
- Maintainability
- Operability
- Reliability
- Risks
- etc., etc., etc.

INCOSE International Symposium
Philadelphia, PA
June 24-27, 2013

And, in an "information sense", . . .

we can still think of all these as kinds of "parts"—not just physical parts of a system, but parts of a system model:

- Stakeholders
- Requirements
- Design
- Interfaces
- Modes
- Performance
- Failure Modes & Effects
- Verification Plans
- Alternatives
- Configurability
- Manufacturability
- Maintainability
- Operability
- Reliability
- Risks
- etc., etc., etc.

And, once again, it turns out that . . .

the relationships between these information components is just as important as the lists of them, taken alone:

- Stakeholders
- Requirements
- Design
- Interfaces
- Modes
- Performance
- Failure Modes & Effects
- Verification Plans

- Alternatives
- Configurability
- Manufacturability
- Maintainability
- Operability
- Reliability
- Risks
- etc., etc., etc.

Physical Architecture

Information Architecture

??

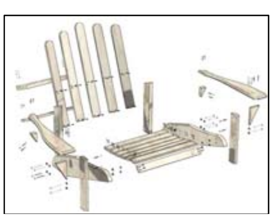
And, once again, it turns out that . . .

the relationships between these information components is just as important as the lists of them, taken alone:

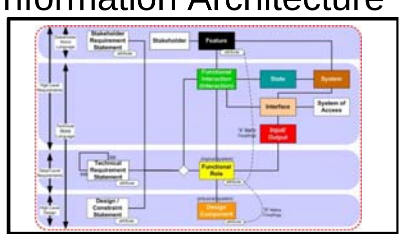
- Stakeholders
- Requirements
- Design
- Interfaces
- Modes
- Performance
- Failure Modes & Effects
- Verification Plans

- Alternatives
- Configurability
- Manufacturability
- Maintainability
- Operability
- Reliability
- Risks
- etc., etc., etc.

Physical Architecture

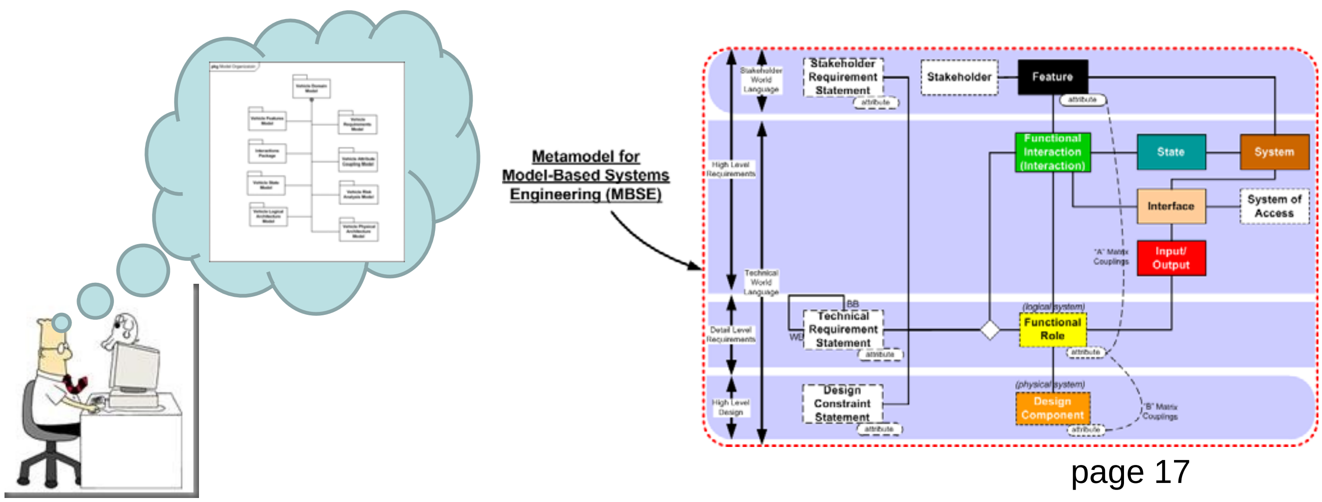


Information Architecture



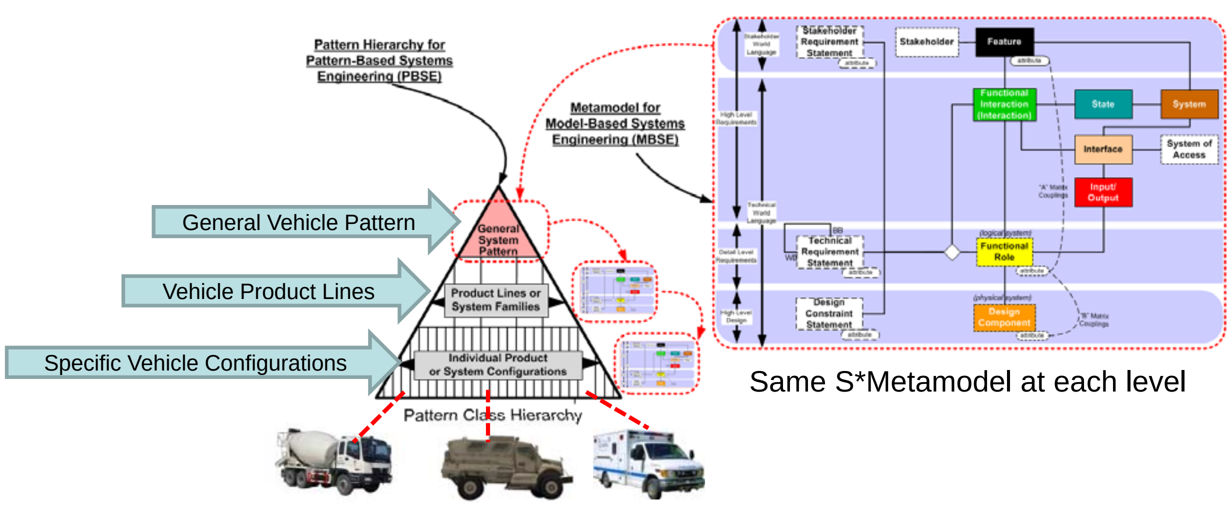
Taking advantage of Model-Based Systems Engineering (MBSE)

- **An S* Model** is a description of all those important things, and the relationships between them.
- Typically expressed in the "views" of some modeling language (e.g., SysML™).
- The S* Metamodel: The smallest set of information sufficient to describe a system for systems engineering purposes.
- Includes not only the physical Platform information, but all the extended system information (e.g., requirements, risk analysis, design trade-offs & alternatives, decision processes, etc.):



Extending the Concept to Patterns, and Pattern-Based Systems Engineering (PBSE)

- An **S* Pattern** is a configurable, re-usable S* Model. It is an extension of the idea of a Platform (which is a configurable, re-usable design) or Enterprise / Industry Framework.
- The Pattern includes not only the physical Platform information, but all the extended system information (e.g., pattern configuration rules, requirements, risk analysis, design trade-offs & alternatives, decision processes, etc.):

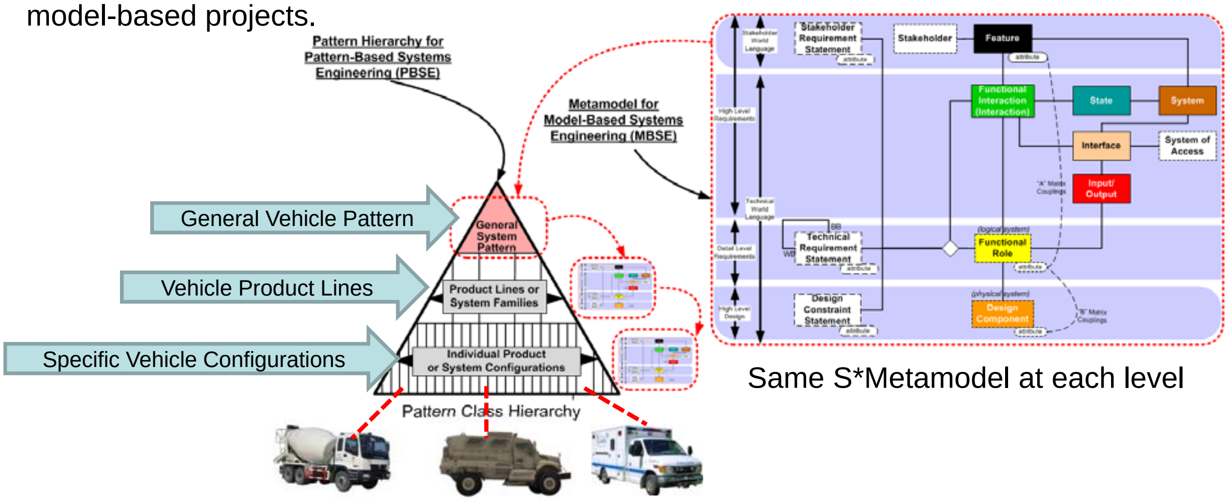


Concept Summary:
Pattern-Based Systems Engineering (PBSE)

– By including the appropriate S* Metamodel concepts, these can readily be managed in (SysML or other) preferred modeling languages and MBSE tools—the ideas involved here are not specific to a modeling language or specific tool.

– The order-of-magnitude changes have been realized because projects that use PBSE rapidly start from an existing Pattern, gaining the advantages of its content, and feed the pattern with what they learn, for future users.

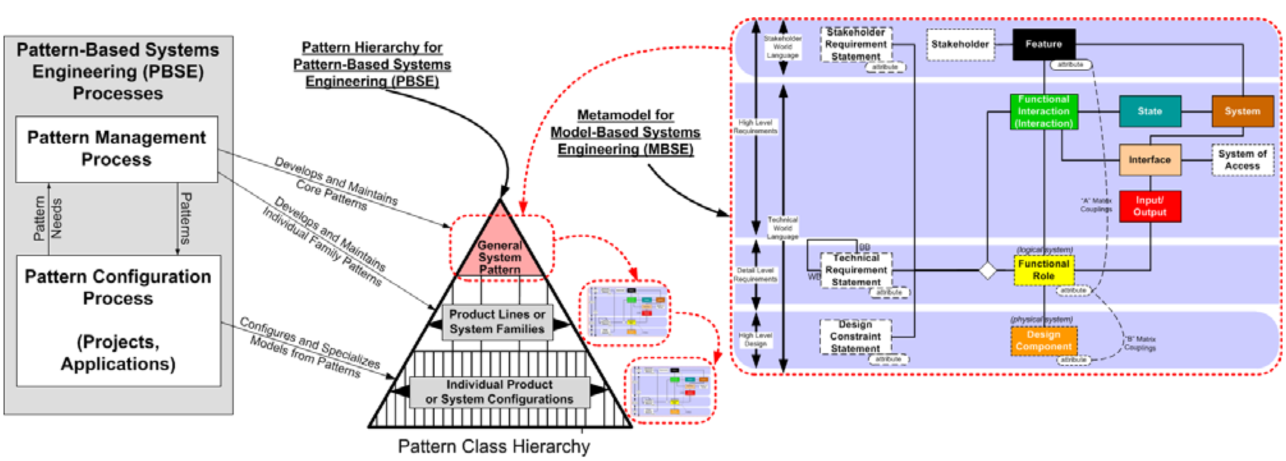
– The "game changer" here is the shift from "learning to model" to "learning the model", freeing many people to rapidly configure, specialize, and apply patterns to deliver value in their model-based projects.



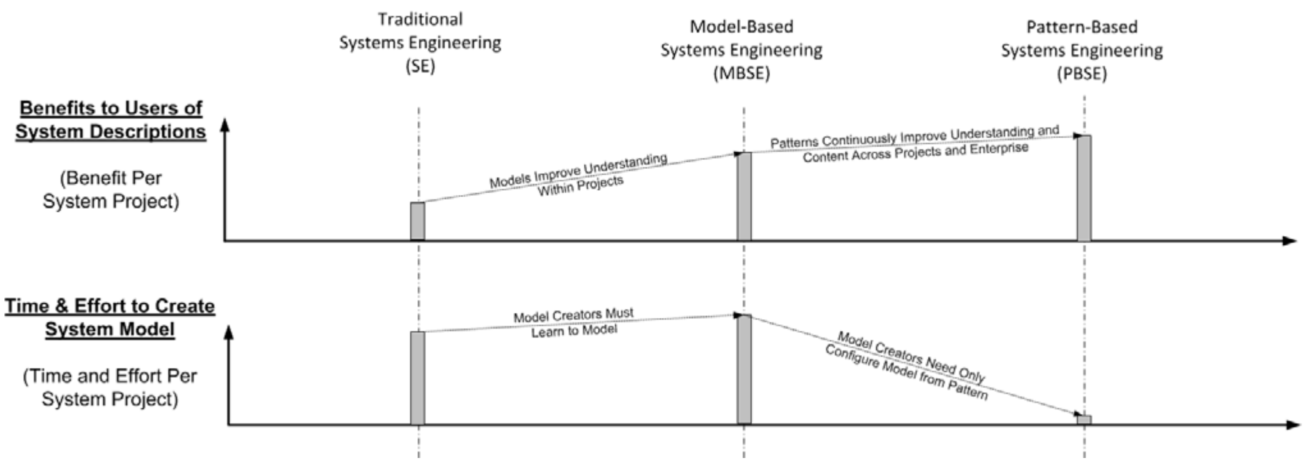
Same S*Metamodel at each level

Concept Summary:
Pattern-Based Systems Engineering (PBSE)

- PBSE provides a specific technical method for implementing:
  – Platform Management
  – Enterprise or Industry Frameworks
  – System Standards
  – Experience Accumulation for Systems of Innovation
  – Lean Product Development & IP Asset Re-use



Comparative Benefits and Costs Summary



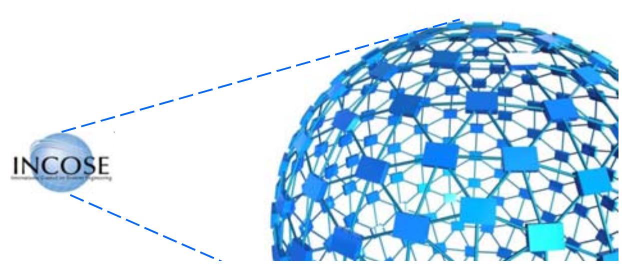
"Learn to Model" "Learn the Model"

Status of PBSE

- The major aspects of PBSE have been defined and practiced for years across a number of enterprises and domains, but with limited integration or awareness within INCOSE community:

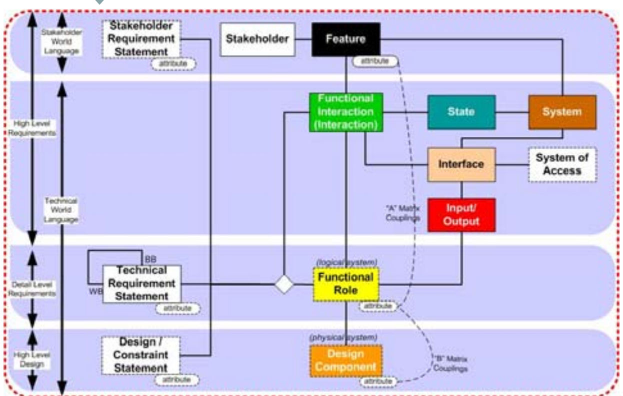
Medical Device Patterns	Construction Equipment Patterns	Commercial Vehicle Patterns	Space Tourism Pattern
Manufacturing Process Patterns	Vision System Patterns	Packaging System Patterns	Lawnmower Pattern
Embedded Intelligence Patterns	Systems of Innovation (SOI) Pattern	Baby Product Pattern	Orbital Satellite Pattern
Development Process Patterns	Production Material Handling Patterns	Engine Controls Patterns	Military Radio Systems Pattern

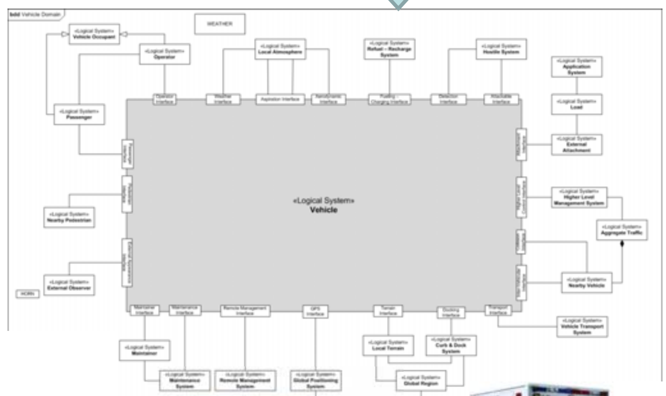
- The PBSE Workshop is more about integration of proven methods and INCOSE community awareness and capability than about technically establishing a new method—although it may look new to INCOSE practitioners.
- We recognize that the human change aspect can be the most challenging – but are not suggesting that we also have to create new technical methods. We are introducing PBSE to a larger community.



Representing system patterns: An example

- S*Metamodel framework
- A Vehicle Pattern in SysML
- An Exercise





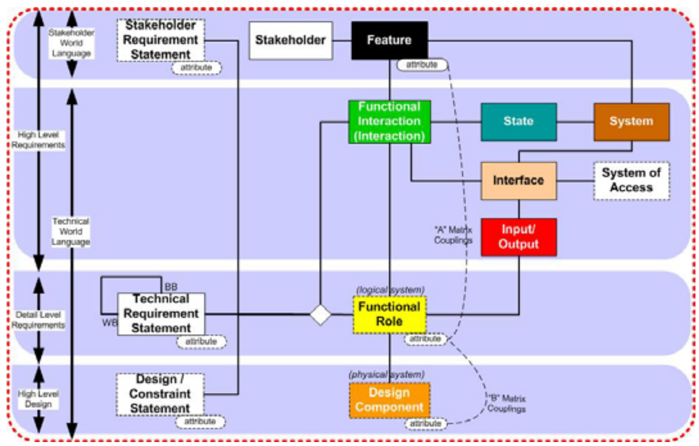
Representing System Patterns: The S* Metamodel Framework

- What is the smallest amount of information we need to represent pattern regularities?
  - Some people have used prose to describe system regularities.
  - This is better than nothing, but usually not enough to deal with the spectrum of issues in complex systems.
- We use S* Models, which are the minimum model-based information necessary:
  - This is not a matter of modeling language—your current favorite language and tools can readily be used for S* Models.
  - The minimum underlying information classes are summarized in the S* Metamodel, for use in any modeling language.
- The resulting system model is made configurable and reusable, thereby becoming an S* Pattern.

Representing System Patterns: The S* Metamodel Framework

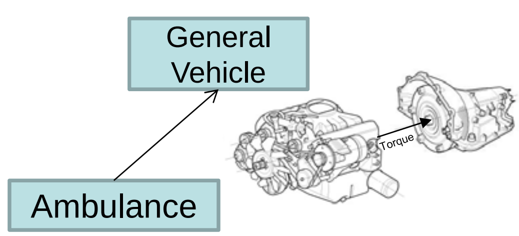
- A <u>metamodel</u> is a model of other models;
  - Sets forth how we will represent Requirements, Designs, Verification, Failure Analysis, Trade-offs, etc.;
  - We utilize the (language independent) S* Metamodel from Systematica™ Methodology:
- The resulting system models may be expressed in SysML™, other languages, DB tables, etc.
- Has been applied to systems engineering in aerospace, transportation, medical, advanced manufacturing, communication, construction, other domains.

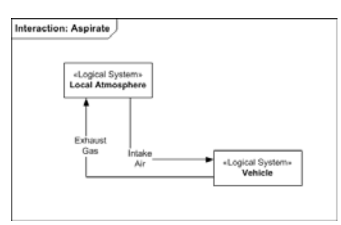
Simple summary of detailed S* Metamodel.



Definitions of some S* Metamodel Classes

- **<u>System</u>**: A collection of interacting components. Example: Vehicle; Vehicle Domain System.
- **<u>Stakeholder</u>**: A person or other entity with something at stake in the life cycle of a system. Example: Vehicle Operator; Vehicle Owner; Pedestrian
- **<u>Feature</u>**: A behavior of a system that carries stakeholder value. Example: Automatic Braking System Feature; Passenger Comfort Feature Group
- **<u>Functional Interaction (Interaction):</u>** An exchange of energy, force, mass, or information by two entities, in which one changes the state of the other. Example: Refuel Vehicle; Travel Over Terrain
- **<u>Functional Role (Role):</u>** The behavior performed by one of the interacting entities during an Interaction. Example: Vehicle Operator; Vehicle Passenger Environment Subsystem
- **<u>Input-Output:</u>** That which is exchanged during an interaction (generally associated with energy, force, mass, or information). Example: Fuel, Propulsion Force, Exhaust Gas



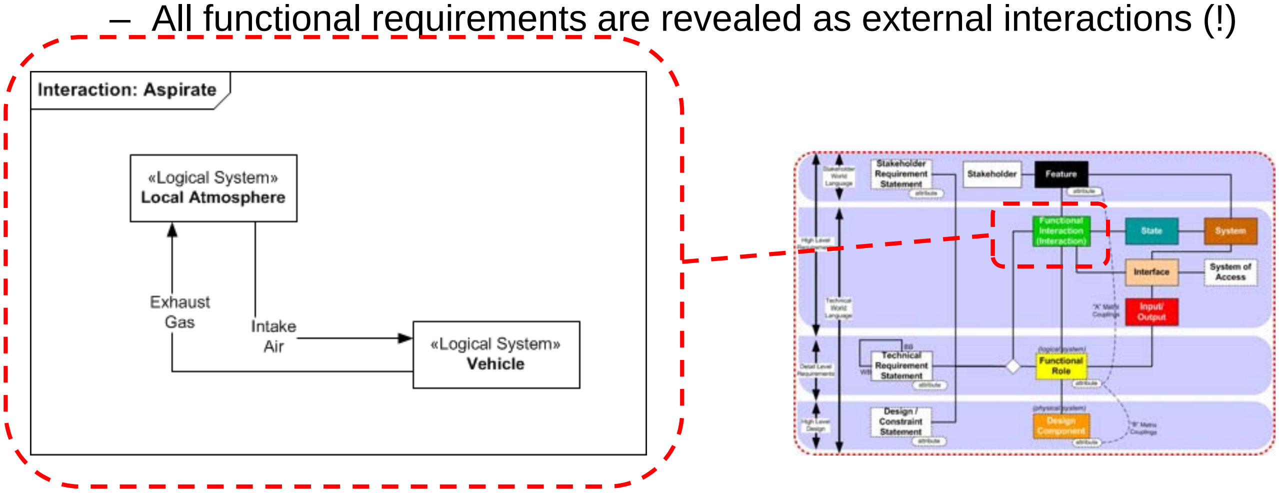


Definitions of some S* Metamodel Classes

- **System of Access:** A system which provides the means for physical interaction between two interacting entities. Examples: Fueling Nozzle-Receptacle; Grease Gun Fitting; Steering Wheel; Dashboard; Brake Peddle
- **Interface:** The association of a System (which "has" the interface), one or more Interactions (which describe behavior at the interface), the Input-Outputs (which pass through the interface), and a System of Access (which provides the means of the interaction). Examples: Operator Interface; GPS Interface
- **State:** A mode, situation, or condition that describes a System's condition at some moment or period of time. Example: Starting; Cruising; Performing Maneuvers
- **Design Component:** A physical entity that has identity, whose behavior is described by Functional Role(s) allocated to it. Examples: Garmin Model 332 GPS Receiver; Michelin Model 155 Tire
- **Requirement Statement:** A (usually prose) description of the behavior expected of (at least part of) a Functional Role. Example: "The System will accept inflow of fuel at up to 10 gallons per minute without overflow or spillage."

Physical Interactions: At the heart of S* models

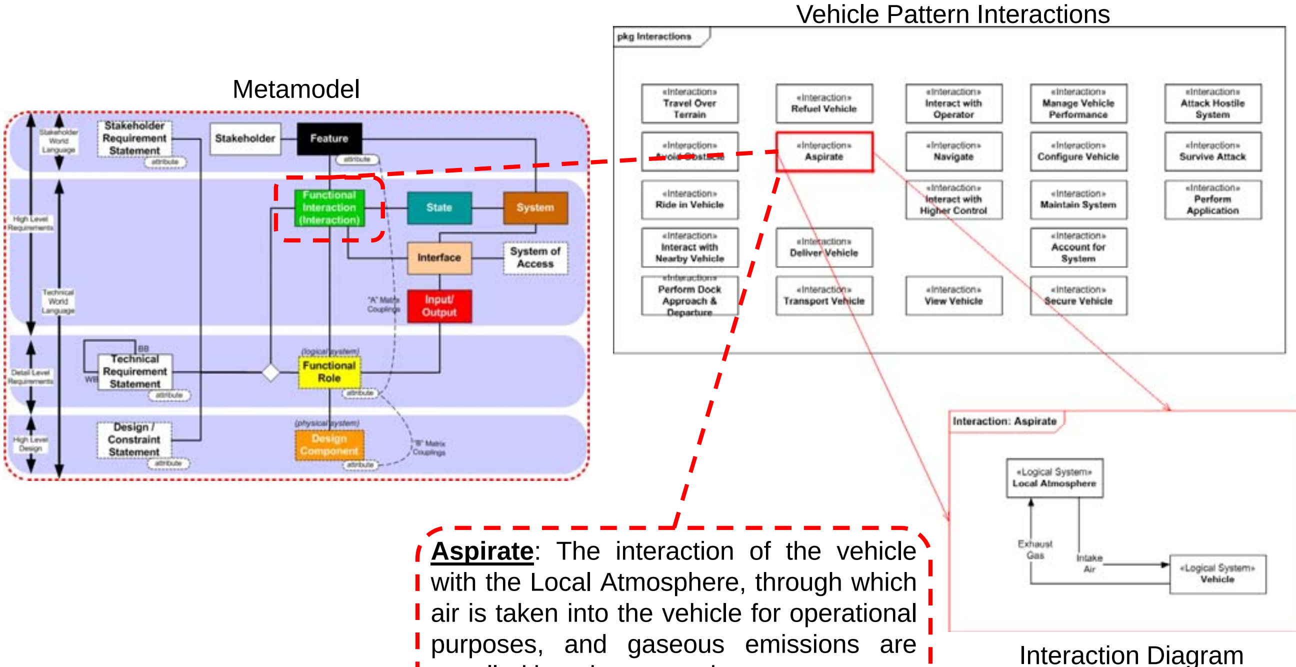
- S* models represent Interactions as explicit objects:
  - Goes to the heart of 300 years of natural science of systems as a foundation for engineering, including emergence.
  - All physical laws of science are about interactions in some way.
  - All functional requirements are revealed as external interactions (!)



- Other Metamodel parts: See the Vehicle Pattern example.

Physical Interactions: At the heart of S* models

- S* models represent Physical Interactions as explicit objects:



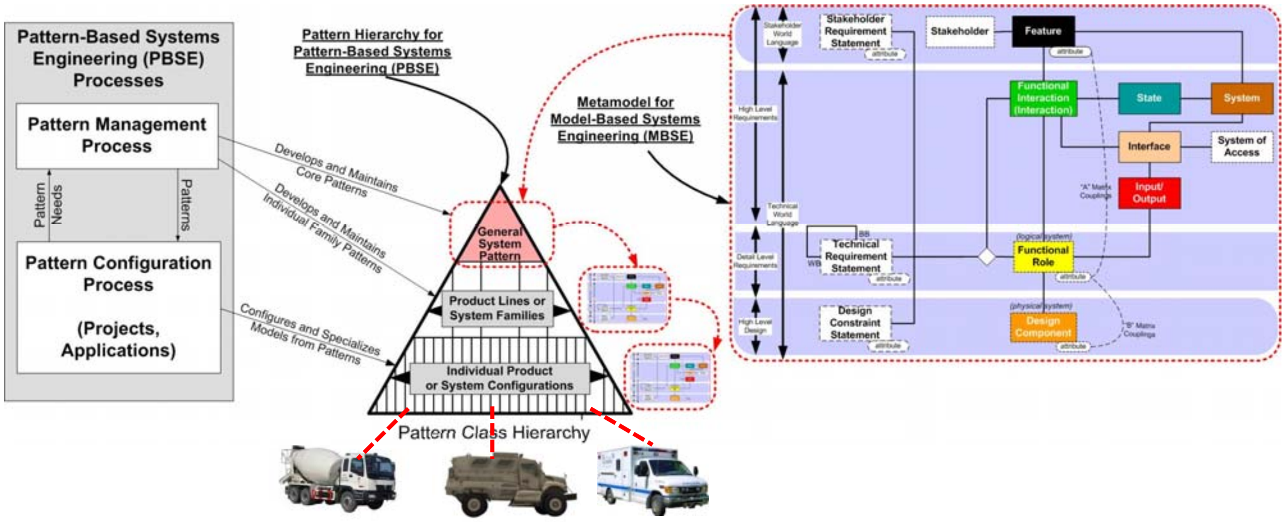
Aspirate: The interaction of the vehicle with the Local Atmosphere, through which air is taken into the vehicle for operational purposes, and gaseous emissions are expelled into the atmosphere.

Pattern-based systems engineering (PBSE)

- Model-based Patterns:
  - In this approach, Patterns are reusable, configurable S* models of families (product lines, sets, ensembles) of systems.
  - A Pattern is not just the physical product family—it includes its behavior, decomposition structure, failure modes, and other aspects of its model.
- These Patterns are ready to be configured to serve as Models of individual systems in projects.
- Configured here is specifically limited to mean that:
  - Pattern model components are populated / de-populated, and
  - Pattern model attribute (parameter) values are set
  - both based on Configuration Rules that are part of the Pattern.
- Patterns based on the same Metamodel as “ordinary” Models

Pattern-based systems engineering (PBSE)

- Pattern-Based Systems Engineering (PBSE) has two overall processes:
  - **<u>Pattern Management Process</u>**: Creates the general pattern, and periodically updates it based on application project discovery and learning;
  - **<u>Pattern Configuration Process</u>**: Configures the pattern into a specific model configuration (e.g., a new product) for application in a project.



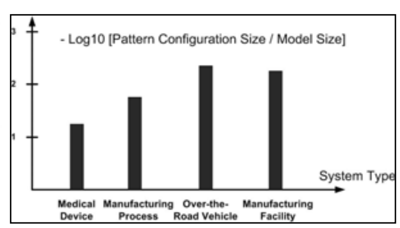
We'll discuss examples from both processes in this tutorial.

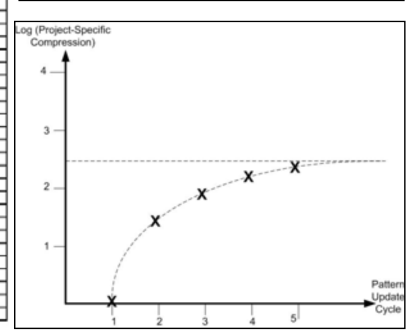
Pattern configurations

- A table of configurations illustrates how patterns facilitate compression;
- Each column in the table is a compressed system representation with respect to ("modulo") the pattern;
- The compression is typically very large;
- The compression ratio tells us how much of the pattern is variable and how much fixed, across the family of potential configurations.

Lawnmower Product Line: Configurations Table

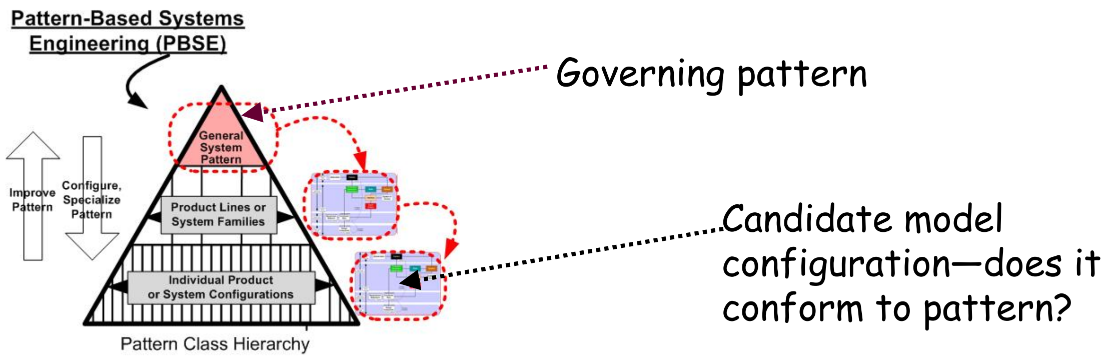
		Units	Walk-Behind	Walk-Behind	Walk-Behind	Riding	Riding	Riding Mower	Autonomous
			Push Mower	Mower	Self-Propelled	Rider	Tractor	Tractor	Autonomous
			Push Mower	Self-Propelled	Wide Cut	Rider	Lawn	Garden	Auto Mower
	Model Number		M3	M5	M11	M17	M19	M23	M100
	Market Segment		Sm Resident	Med Resident	Med Resident	Lg Resident	Lg Resident	Home Garden	High End Suburban
Power	Engine Manufacturer		B&S	B&S	Tecumseh	Tecumseh	Kohler	Kohler	Elektroset
	Horsepower	HP	5	6.5	13	16	18.5	22	0.5
Production	Cutting Width	Inches	17	19	36	36	42	48	16
	Maximum Mowing Speed	MPH	3	3	4	8	10	12	2.5
	Maximum Mowing Productivity	Acres/Hr			1.6				
	Turning Radius	Inches	0	0	0	0	126	165	0
	Fuel Tank Capacity	Hours	1.5	1.7	2.5	2.8	3.2	3.5	2
	Towing Feature						x	x	
	Electric Starter Feature				x	x	x	x	
	Basic Mowing Feature Group		x	x	x	x	x	x	x
Mower	No. of Anti-Scalping Rollers		0	0	1	2	4	6	0
	Cutting Height Minimum	Inches	1	1.5	1.5	1.5	1	1.5	1.2
	Cutting Height Maximum	Inches	4	5	5	6	8	10	3.8
	Operator Riding Feature					x	x	x	
	Grass Bagging Feature		Optional	Optional	Optional	Optional	Optional	Optional	
	Mulching Feature		Standard	Factory Installed	Dealer Installed				
	Aerator Feature					Optional	Optional	Optional	
	Autonomous Mowing Feature								x
	Dethatching Feature					Optional	Optional	Optional	
Physical	Wheel Base	Inches	18	20	22	40	48	52	16
	Overall Length	Inches	18	20	23	58	56	68	28.3
	Overall Height	Inches	40	42	42	30	32	36	10.3
	Width	Inches	18	20	22	40	48	52	23.6
	Weight	Pounds	120	160	300	680	705	1020	15.6
	Self-Propelled Mowing Feature			x	x	x	x	x	x
	Automatic TransmFeature							x	
Financials	Retail Price	Dollars	360	460	1800	3300	6100	9990	1799
	Manufacturer Cost	Dollars	120	140	550	950	1800	3500	310
Maintenance	Warranty	Months	12	12	18	24	24	24	12
	Product Service Life	Hours	500	500	600	1100	1350	1500	300
	Time Between Service	Hours	100	100	150	200	200	250	100
Safety	Spark Arrest Feature		x	x	x	x	x	x	





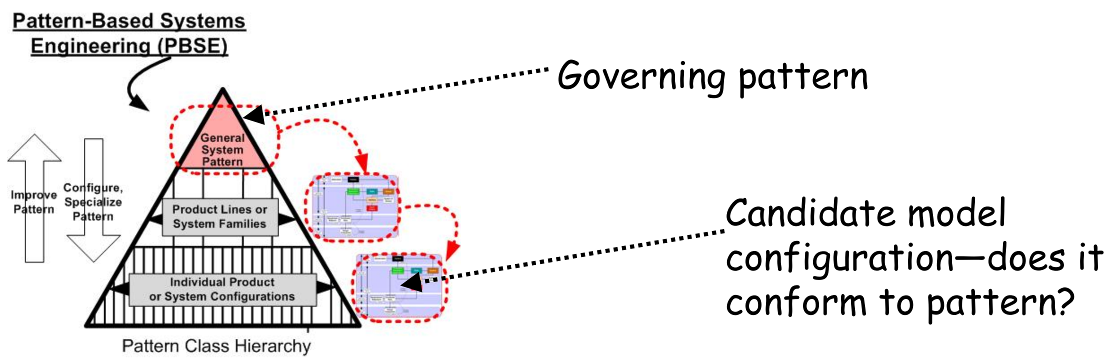
Checking holistic alignment to a pattern

- <u>Gestalt Rules</u> express what is meant by holistic conformance to a pattern:
  - Expressing regularities of whole things, versus same "parts"

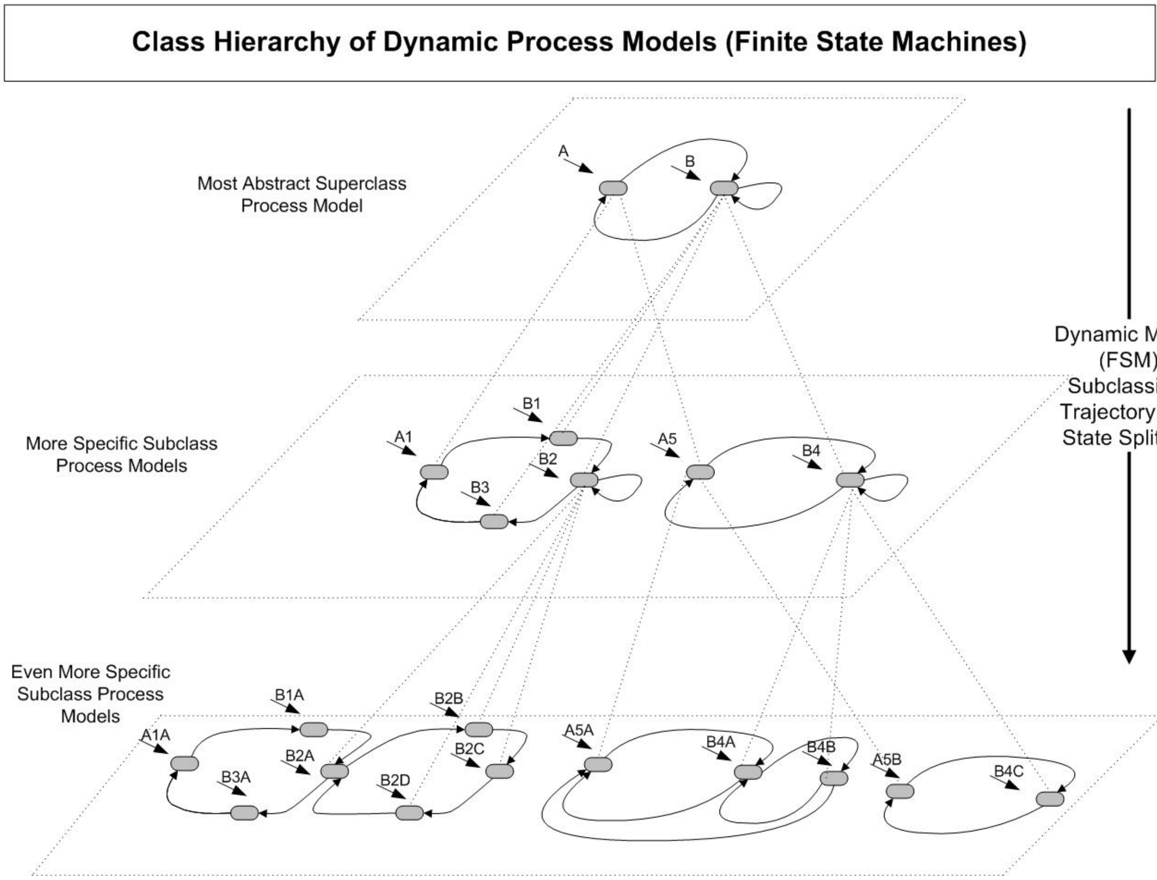


The Gestalt Rules

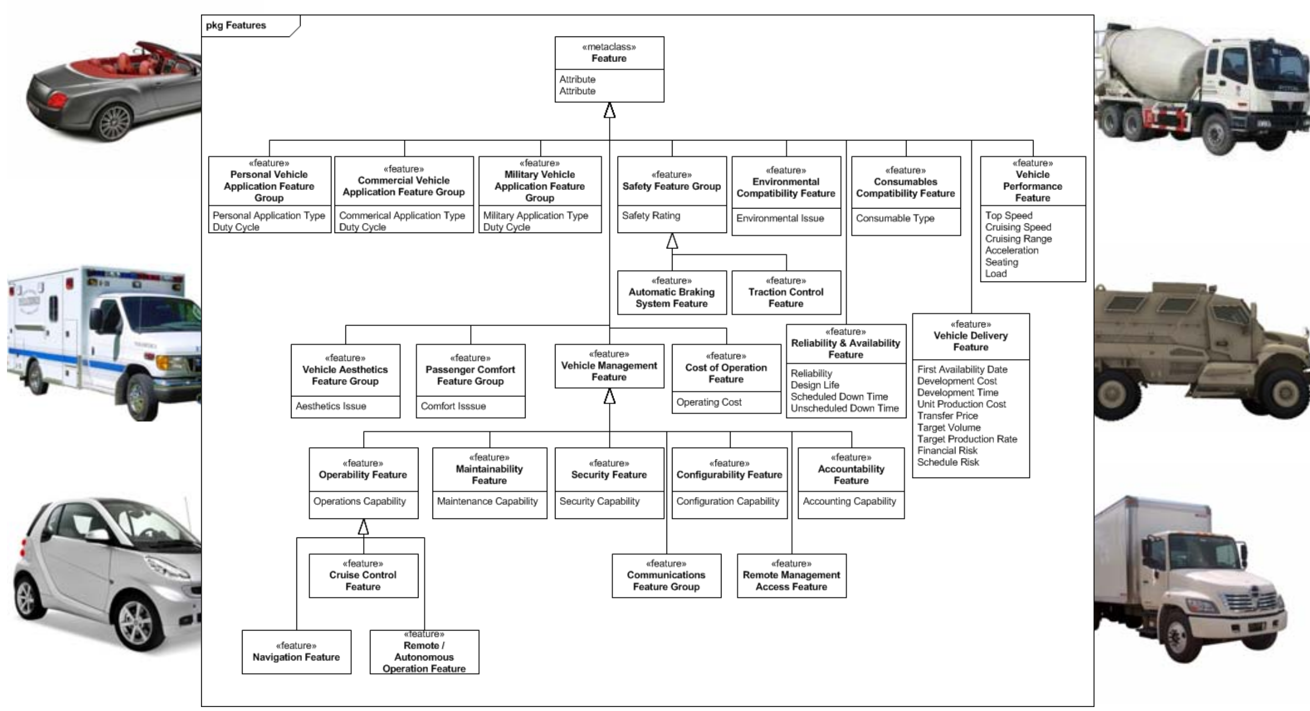
1. Every component class in the candidate model must be a subclass of a parent superclass in the pattern—no "orphan classes".
2. Every relationship between component classes must be a subclass of a parent relationship in the pattern, and which must relate parent superclasses of those same component classes—no "orphan relationships".
3. Refining the pattern superclasses and their relationships is a permissible way to achieve conformance to (1) and (2).



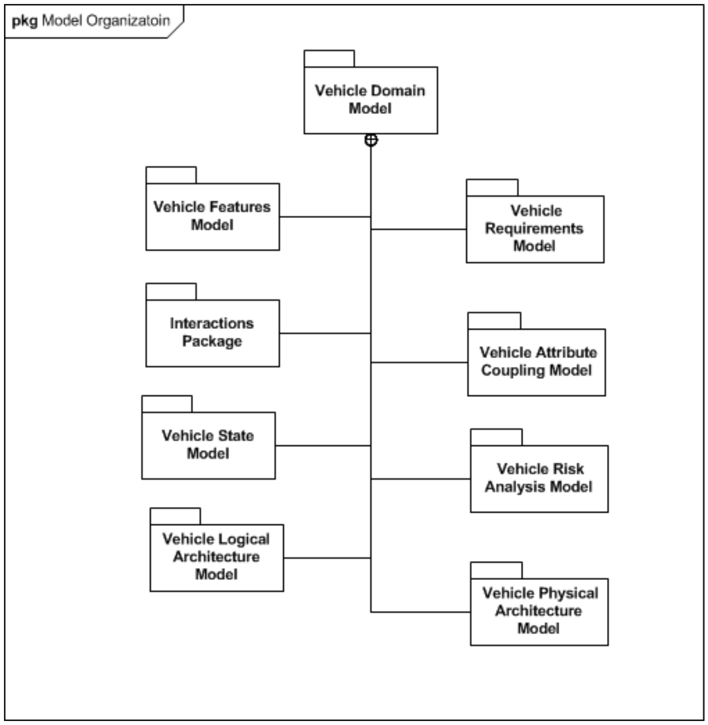
Example: State Model Pattern—illustrates how *visual* is the "class splitting" and "relationship rubber banding" of the Gestalt Rules

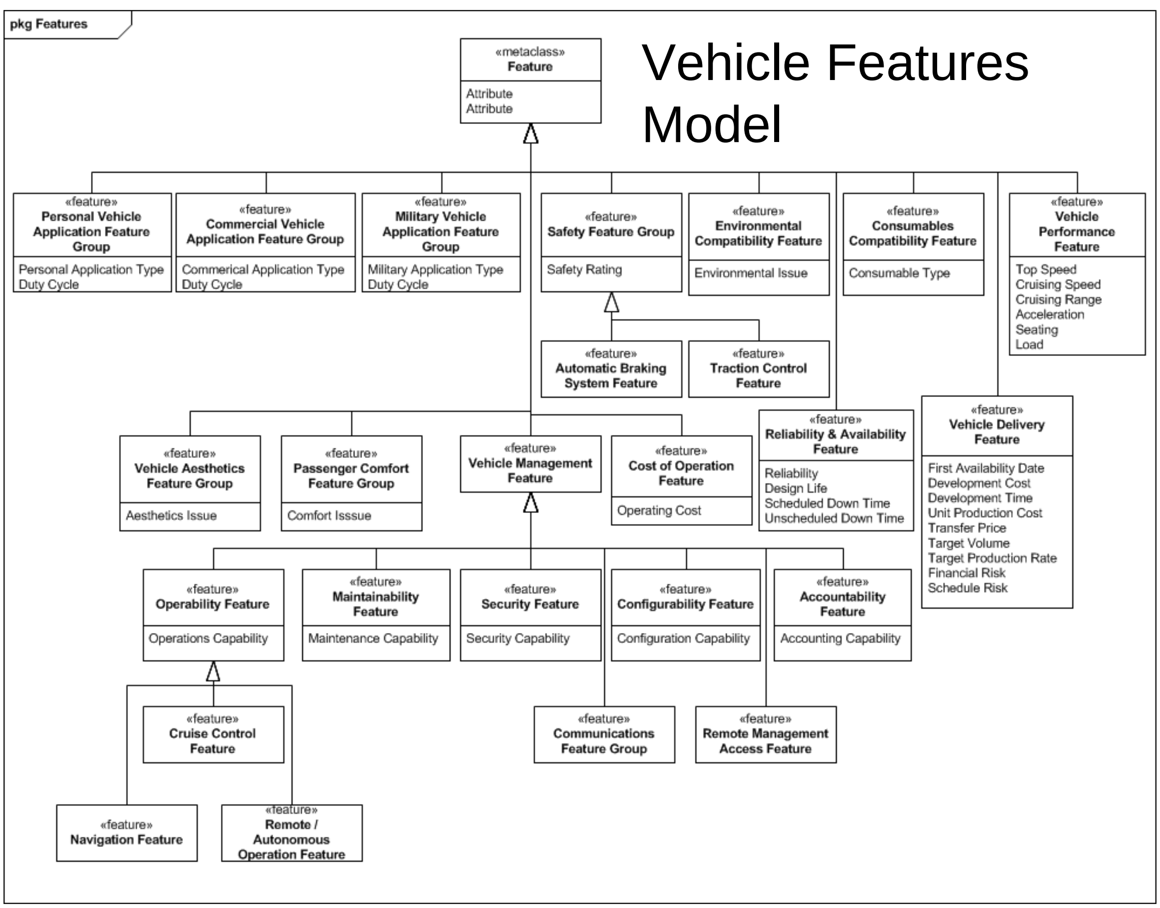


A vehicle pattern in SysML

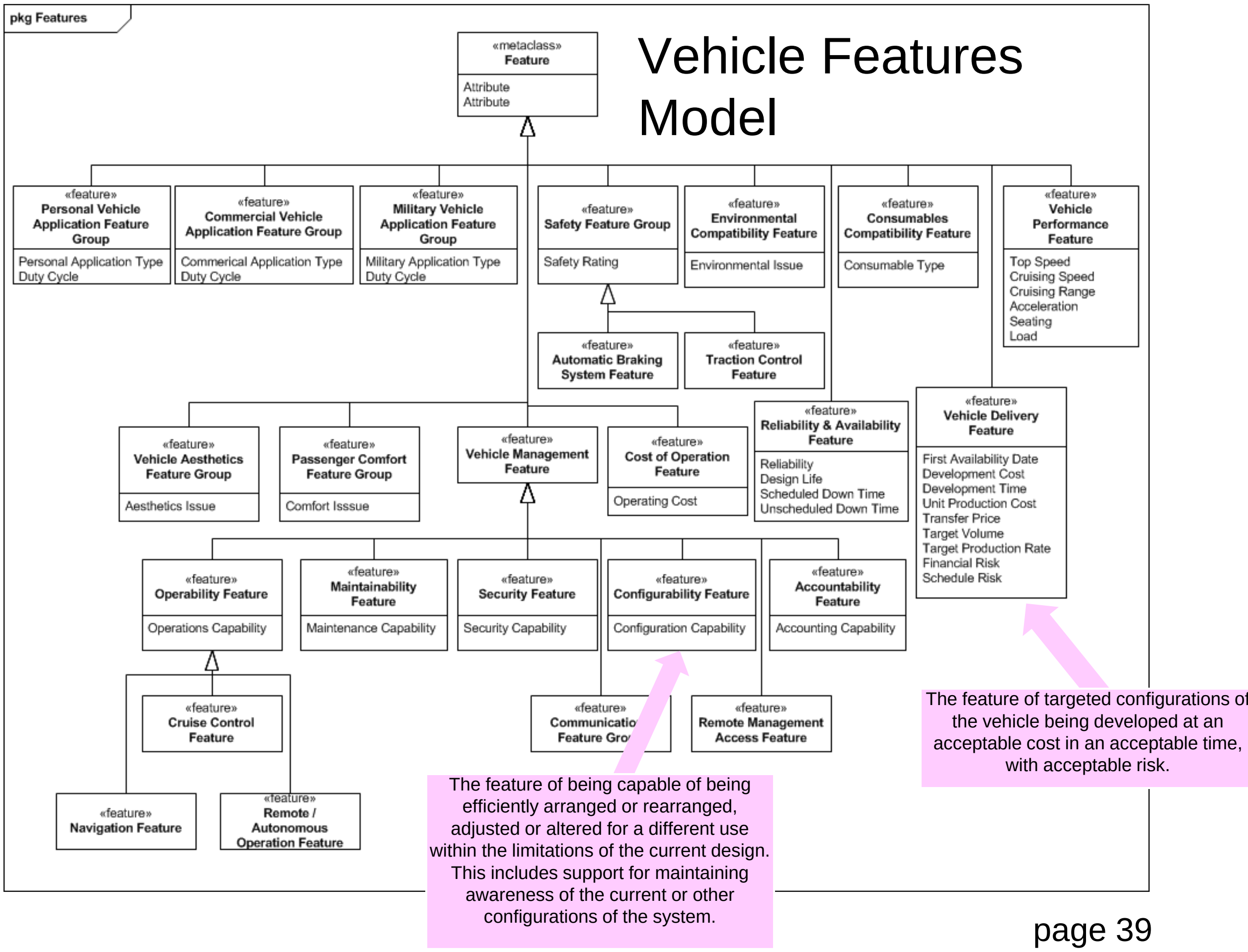


Vehicle Pattern:
Model Organization (Packages)

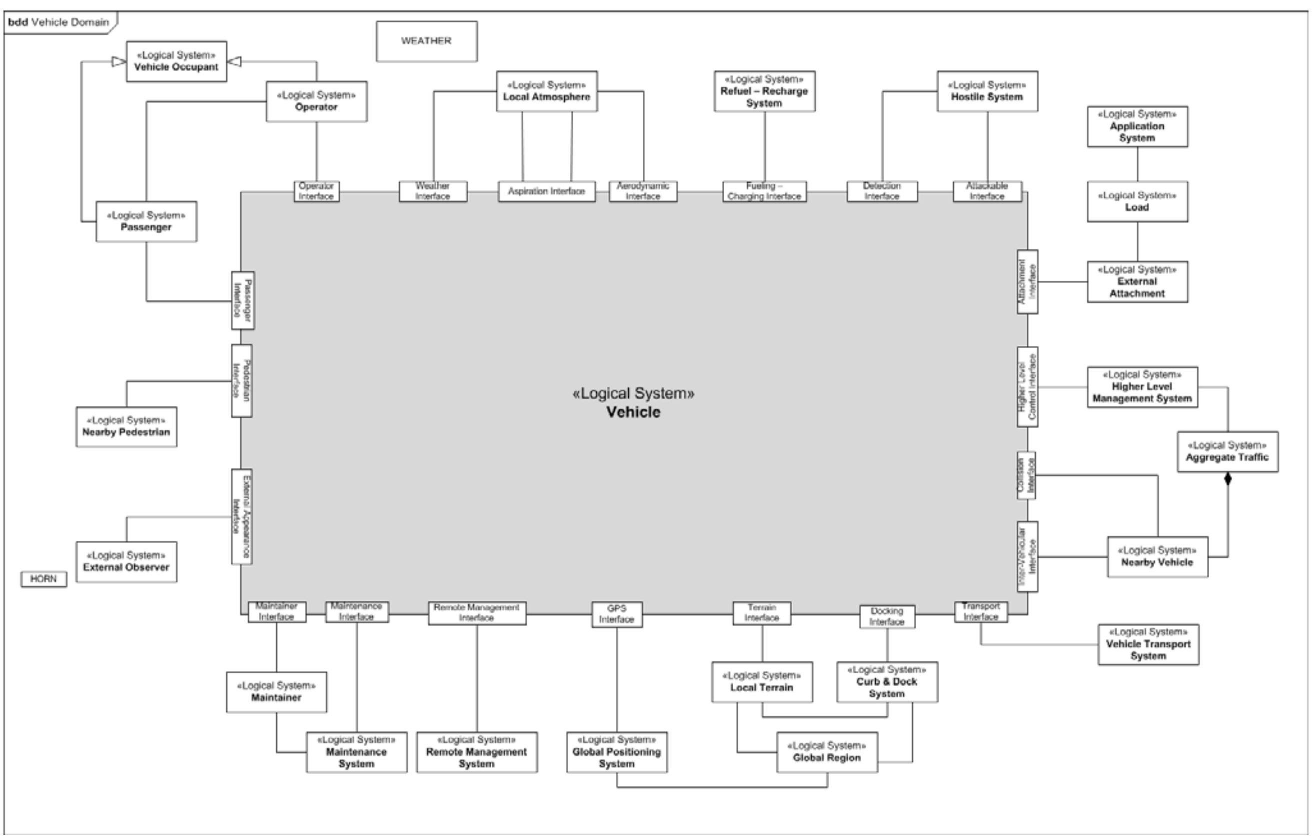




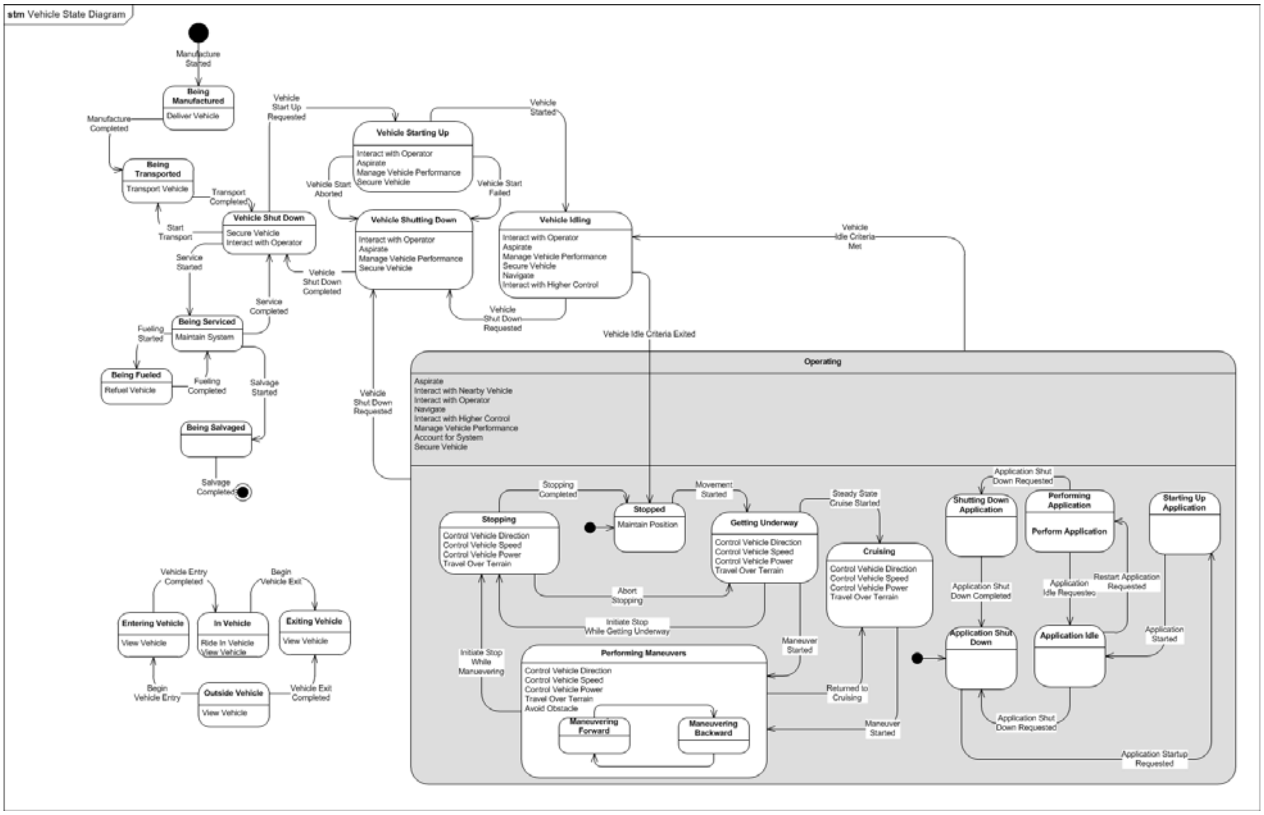
Vehicle Features Model



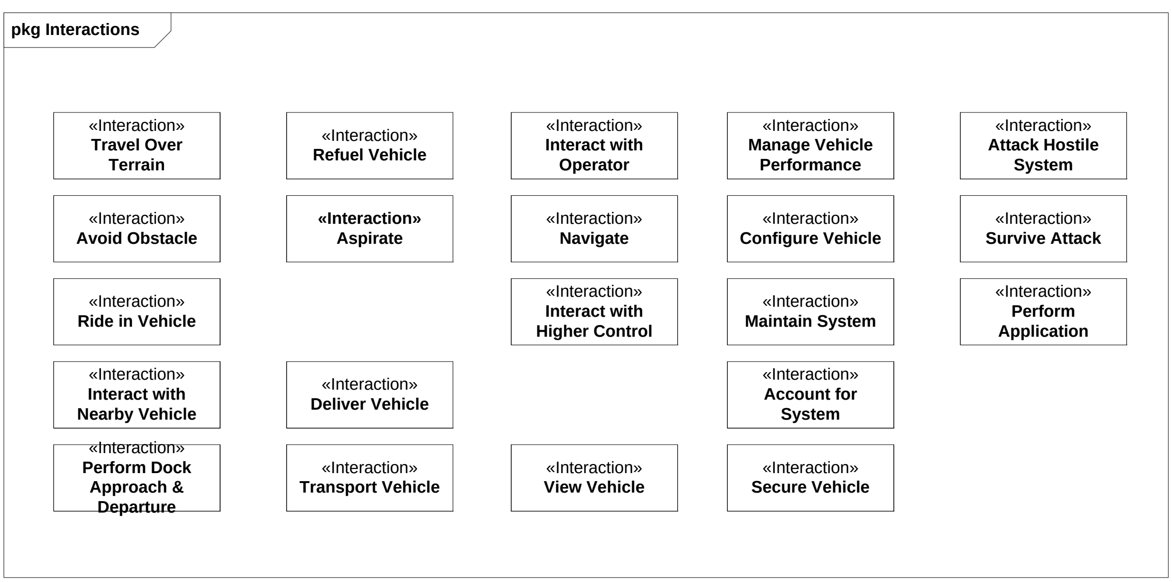
Vehicle Domain Model



Vehicle State Model

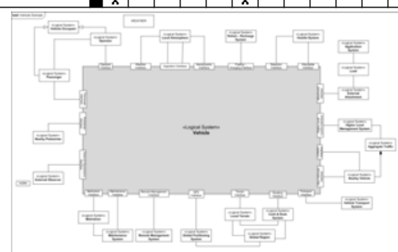


Vehicle Interaction Model



Vehicle Interactions:
Which Actors Participate in Interaction?

Row	Interaction Name	Interaction Definition	Vehicle	Operator	Passenger	Vehicle Occupant	Nearby Pedestrian	External Observer	Maintainer	Maintenance System	Local Atmosphere	Refuel System	Hostile System	External Attachment	Load	Application System	Higher Level Management	Nearby Vehicle	Vehicle Transport	Curb & Dock System	Local Terrain	Global Region	Remote Management System	Global Positioning System
1			Actors																					
3	Account for System	The interaction of the vehicle with its external managers, in which it accounts for vehicle utilization.	x	x					x	x							x						x	
4	Aspirate	The interaction of the vehicle with the Local Atmosphere, through which air is taken into the vehicle for operational purposes, and gaseous emissions are expelled into the atmosphere.	x								x													
5	Attack Hostile System	The interaction of the vehicle with an external hostile system, during which the vehicle projects an attack onto the hostile system's condition.	x										x											
6	Avoid Obstacle	The interaction of the vehicle with an external object, during which the vehicle minimizes contact with or proximity to the object.	x				x																	
7	Configure	The interaction of the vehicle with people or systems that manage its arrangement or configuration for intended use.	x						x	x														
8	Deliver Vehicle	The interaction of the vehicle with the process of its delivery, including manufacture, distribution, and development. This includes delivery of each configured version and update of the vehicle product line or family.																						
9	Interact with Higher Control	The interaction of the vehicle with an external higher level management system, along with the vehicle operator, through which the vehicle is fit into larger objectives.	x														x							
10	Interact with Nearby Vehicle	The intearction of the vehicle with another vehicle, in which information is exchanged to identify one vehicle to another.																						
11	Interact with Operator	The interaction of the vehicle with its operator.																						
12	Maintain System	The interaction of the vehicle with a maintainer and/or maintenance system, through which faults in the vehicle are prevented or corrected, so that the intended qualified operating state of the vehicle is maintained.	x						x	x														
13	Manage Vehicle Performance	The interaction of the vehicle with its operator and/or external management system, through which the performance of the vehicle is managed to achieve its operational purpose and objectives.	x	x																				
14	Navigate	The interaction of the vehicle with the Global Positioning System, by which the Vehicle tracks is position on the Earth.	x																					x
15	Perform Application	The interaction of the vehicle with an external Application System, through which the vehicle performs a specialized application.	x													x								
16	Perform Dock Approach & Departure	The interaction of the vehicle with an external docking system, through which the vehicle arrives at, aligns with, or departs from a loading / unloading dock.	x																	x				
17	Refuel Vehicle	The interaction of the vehicle with a fueling system and its operator, through which fuel is added to the vehicle.	x									x												
18	Ride In Vehicle	The interaction of the vehicle with its occupant(s) during, before, or after travel by the vehicle.	x	x	x	x																		
19	Secure Vehicle	The interaction of the vehicle with external actors that may or may not have privileges to access or make use of the resources of the vehicle, or with actors managing that vehicle security.	x	x																				
20	Survive Attack	The interaction of the vehicle with an external hostile system, during which the vehicle protects its occupants and minimizes damage to itself.	x										x											
21	Transport	The interaction of the vehicle with a Vehicle Transport System, through which the Vehicle is transported to an intended destination.	x																x					
22	Travel Over Terrain	The interaction of the vehicle with the terrain over which it travels, by means of which the vehicle moves over the terrain.	x																		x			
23	View Vehicle	The interaction of the vehicle with an external viewer, during which the viewer observes the vehicle.	x					x																



Vehicle Feature-Interaction Associations

PBSE Workbook V5.8 PBSE Vehicle Pattern V1.2.31 [Compatibility Mode] - Microsoft Excel

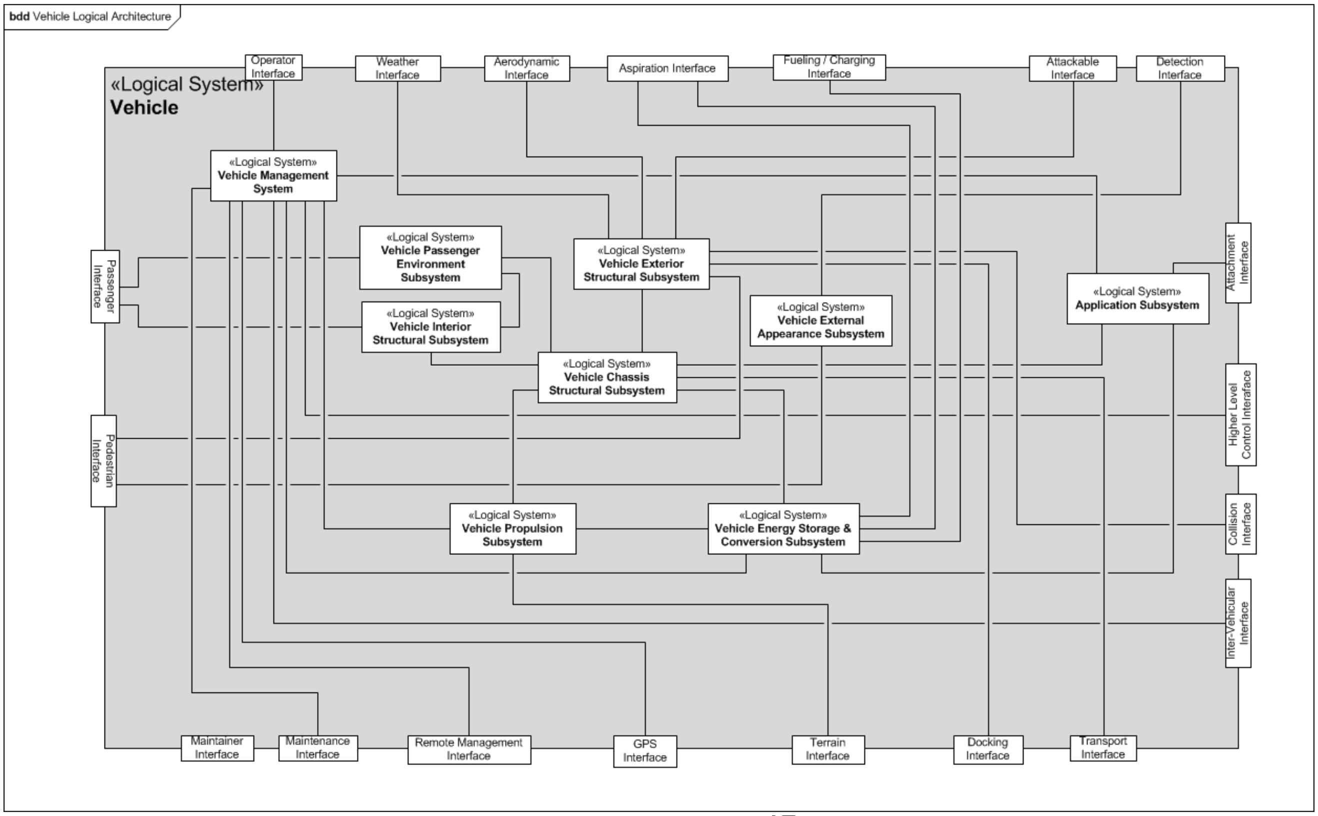
F8 *fx* Interact with Nearby Vehicle

Row	Features (B)	Feature PK Value (C)	Interaction Name (F)	Duplicates w/ Above (H)	Interaction PK Rule (L)	Status (O)	Comments (R)
2	Accountability Feature / Accounting Management Capability	*ANY*	Account for System		FPK		
3	Automatic Braking System Feature		Travel Over Terrain				
4	Commercial Vehicle Application Feature Group / Commercial Application Type	*ANY*	Perform Application		FPK		
5	Communications Feature Group / Communication Capability	Local Cellular	Interact with Higher Control		FPK		
6	Communications Feature Group / Communication Capability	Secure Channel	Interact with Higher Control		FPK		
7	Communications Feature Group / Communication Capability	Wide Area Internet	Interact with Higher Control		FPK		
8	Communications Feature Group / Communication Capability	IFF	Interact with Nearby Vehicle		FPK		
9	Communications Feature Group / Communication Capability	Local Bluetooth Connectivity	Interact with Operator		FPK		
10	Configurability Feature / Configuration Management Capability	*ANY*	Configure Vehicle		FPK		
11	Consumables Compatibility Feature / Consumable Type	Engine Air Filter	Maintain System		FPK		
12	Consumables Compatibility Feature / Consumable Type	Engine Oil Filter	Maintain System		FPK		
13	Consumables Compatibility Feature / Consumable Type	Lubricating Oil	Maintain System		FPK		
14	Consumables Compatibility Feature /	Fuel	Refuel Vehicle				

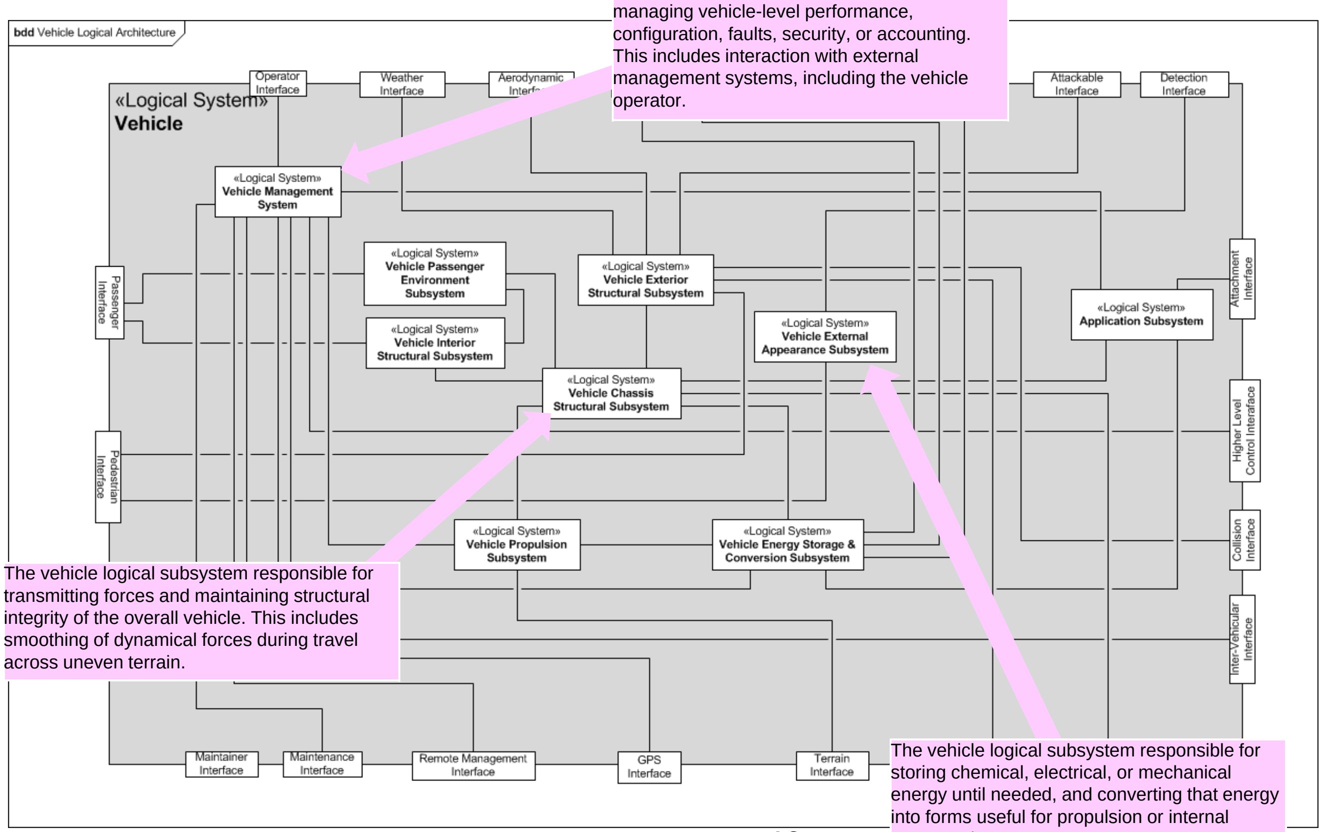
State Diagram | States | Events | Functional Interactions | Feature-Interaction | Interaction-State | LA Diagram | Logical Systems | Log Sys Atts | Requirement Attribute Table

Ready 100% 8:51 PM 9/9/2012

Logical Architecture Model



Logical Architecture Model

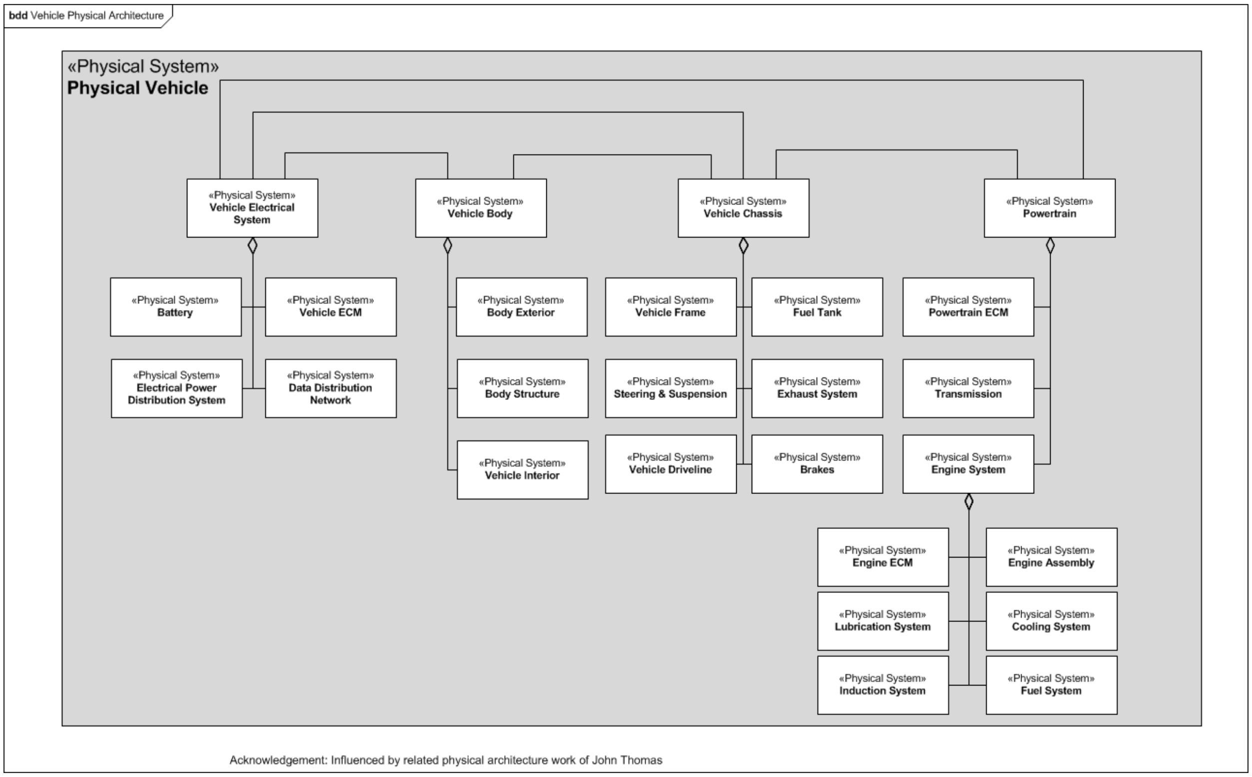


The vehicle logical subsystem responsible for managing vehicle-level performance, configuration, faults, security, or accounting. This includes interaction with external management systems, including the vehicle operator.

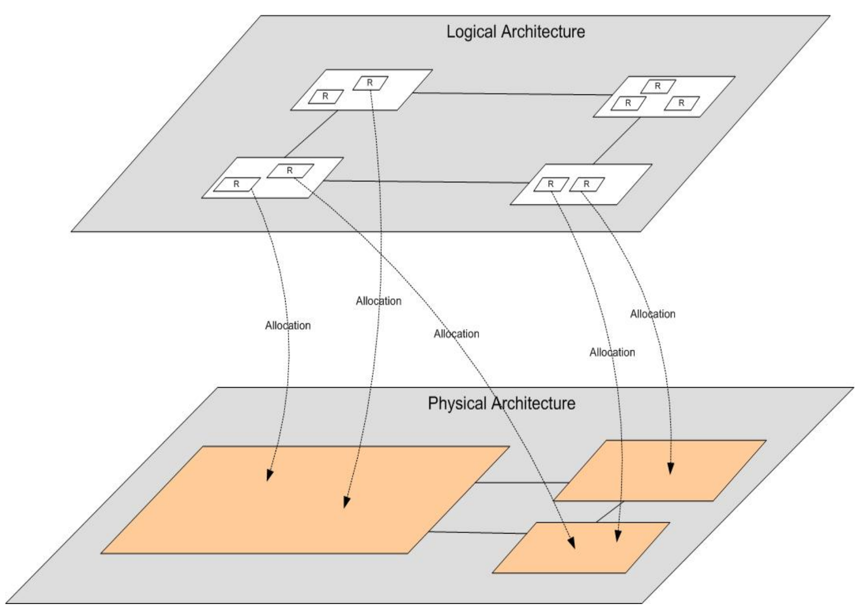
The vehicle logical subsystem responsible for transmitting forces and maintaining structural integrity of the overall vehicle. This includes smoothing of dynamical forces during travel across uneven terrain.

The vehicle logical subsystem responsible for storing chemical, electrical, or mechanical energy until needed, and converting that energy into forms useful for propulsion or internal consumption.

Physical Architecture Model

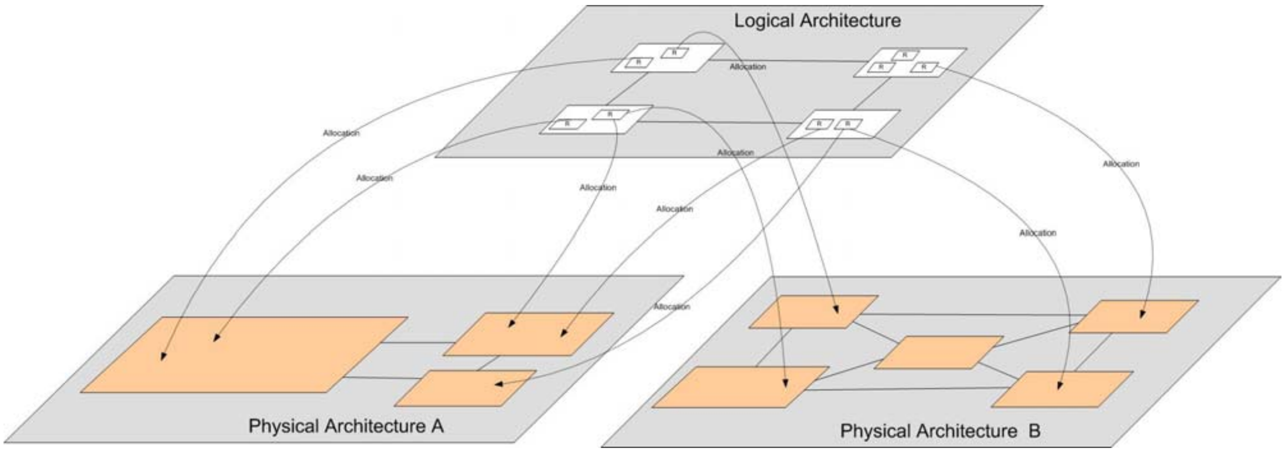


Allocation of Logical Roles to Physical Architecture

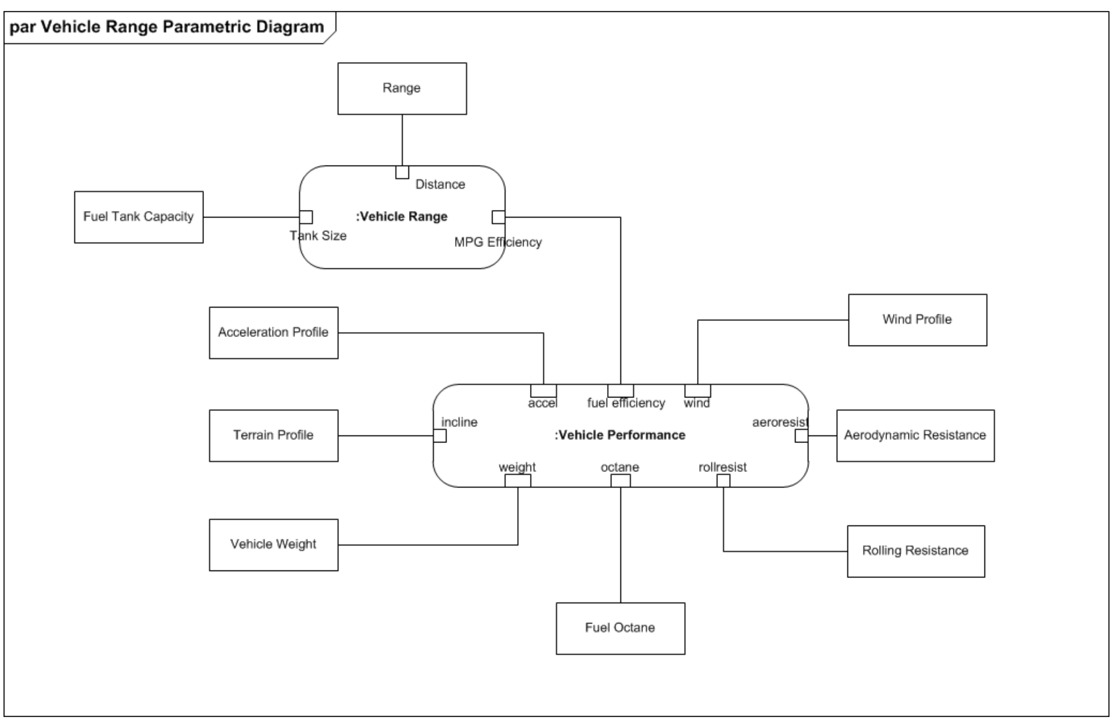


Allocation of Logical Roles to Physical Architecture

- Same Logical Architecture covers many Physical Architectures:



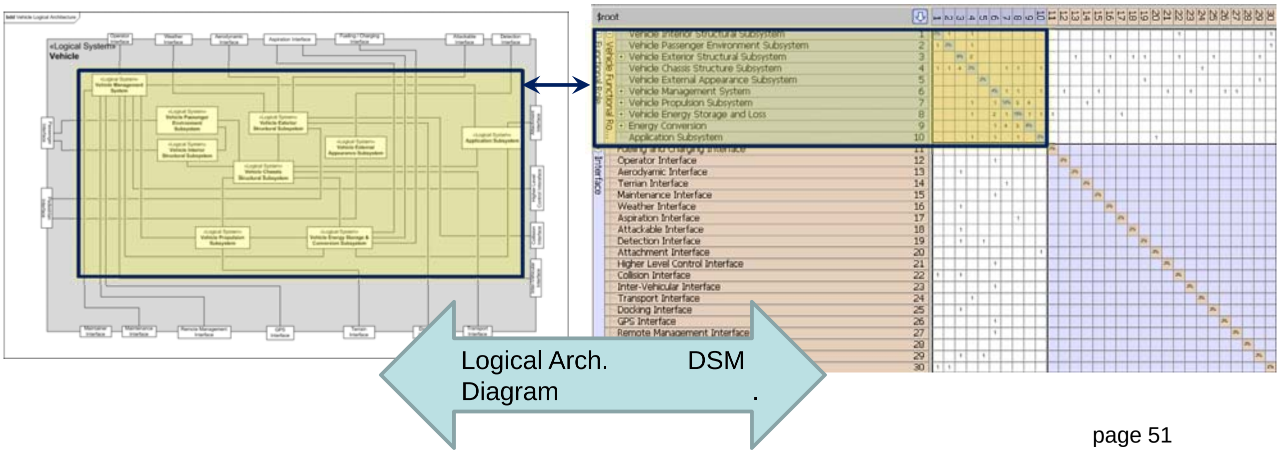
Attribute Coupling Model



Logical Architecture Views

Block Diagram and Design Structure Matrix (DSM)

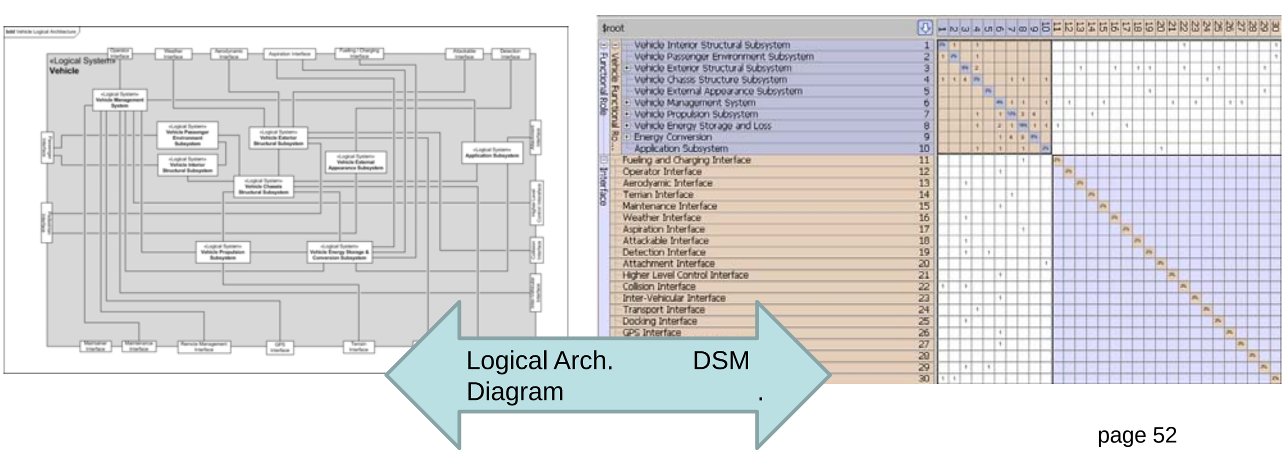
- The structure shown in these architectural diagrams can also be expressed in matrix form
  - These matrices are known as: N^2 matrices, Adjacency Matrices and Design or Dependency Structure Matrices (DSMs)
  - N^2 because their column and row headings are identical, with the matrix cells showing “marks” indicating relationships between components.



Logical Architecture Views

Block Diagram and Design Structure Matrix (DSM)

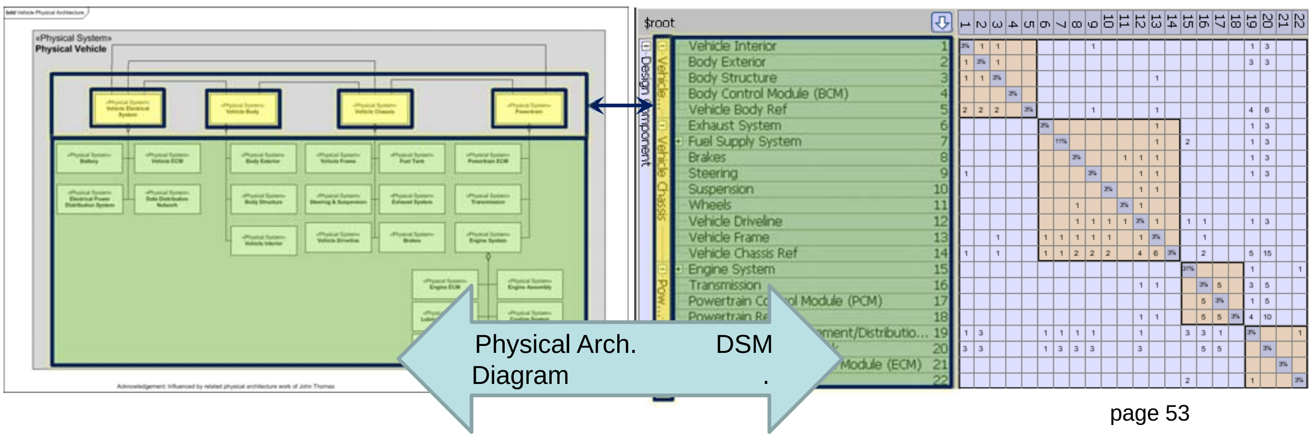
- In the case of Logical Architecture:
  - The blocks in the LA diagram become rows and columns of the DSM
  - The connection lines in the LA diagram become marks in the DSM
- Both views are visualizations of the same information:
  - However the functionality has been partitioned into interacting subsets – Vehicle Functional Roles and Interfaces in this case.



Physical Architecture Views

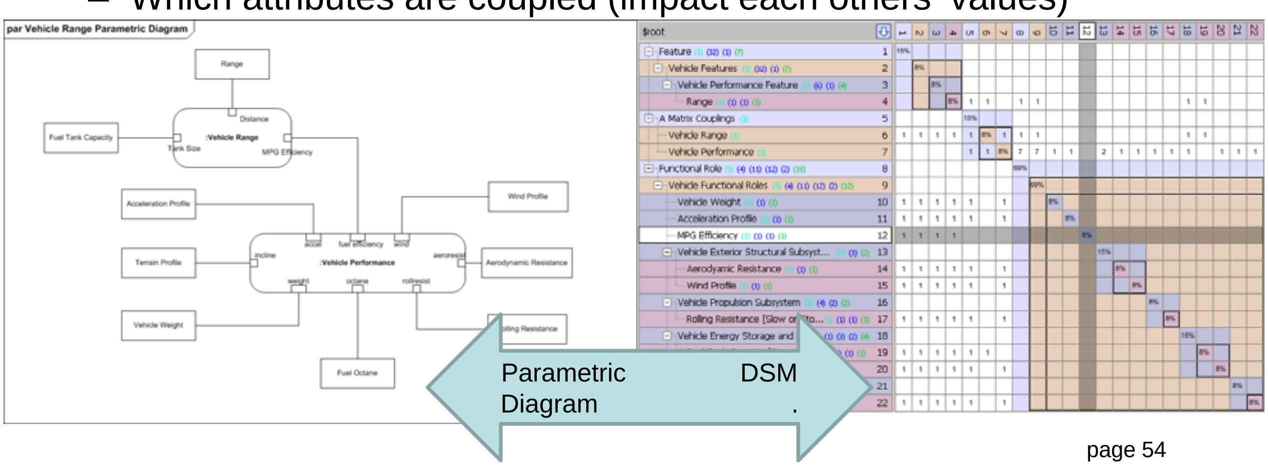
Block Diagram and Design Structure Matrix (DSM)

- In the case of Physical Architecture:
  - The blocks in the LA diagram become rows and columns of the DSM
  - The connection lines in the LA diagram become subsystems or components in the DSM shown in rows and columns
- Both views provide visualizations of hierarchy
  - How the physical system has been partitioned into physical sub-systems that are physically related (connected, contained, adjacent, etc.)
  - The DSM additionally shows the interactions of subsystems



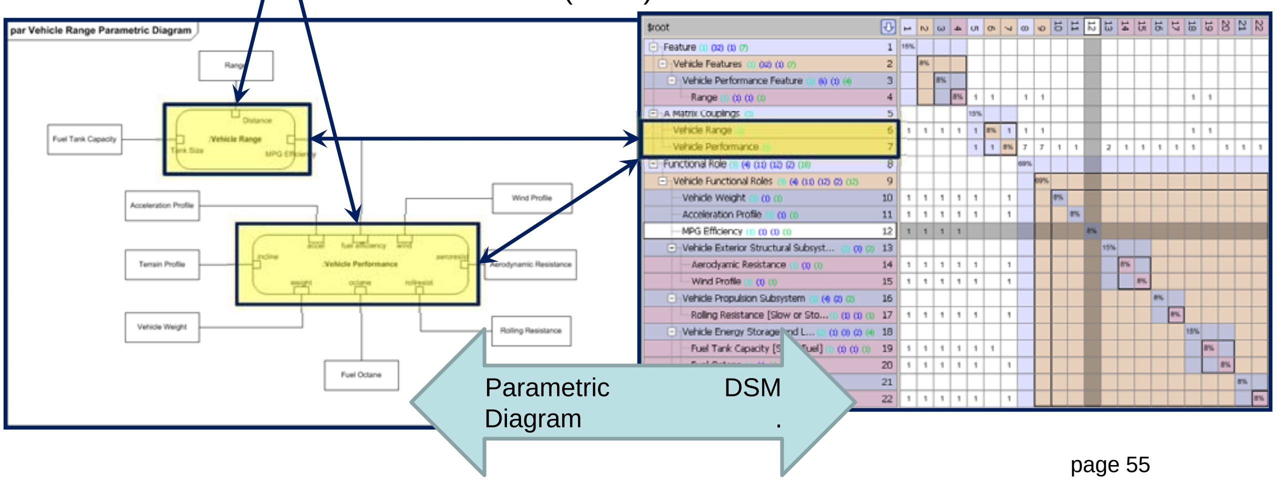
Domain Structure Matrix (DSM) View of Same

- In the case of Coupled Parameters (attributes):
  - Attributes become row and column headings in the DSM
  - This includes adding rows and columns to the Logical Architecture DSM, showing attributes of the Logical Subsystems
  - Connection lines in the drawing become marked cells in the DSM
- Both views convey the same information:
  - Which attributes are coupled (impact each others' values)

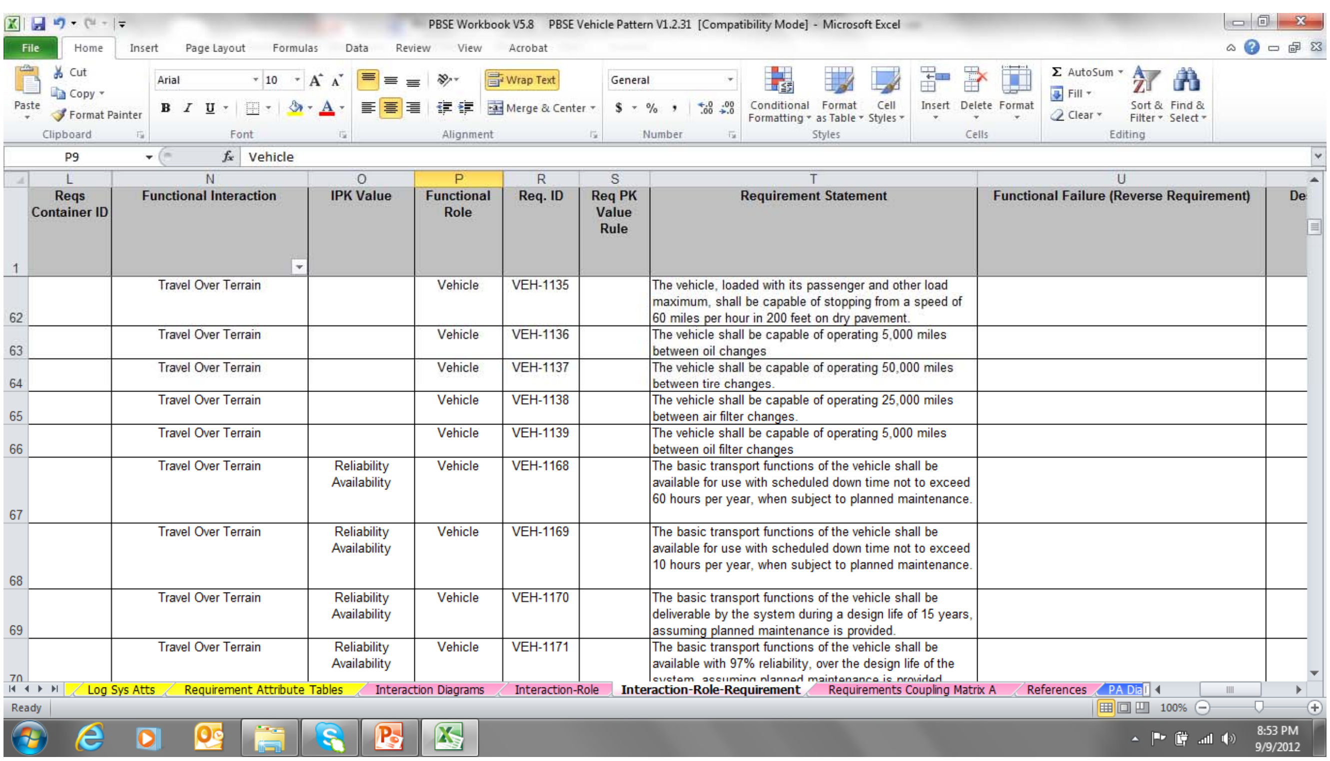


Domain Structure Matrix (DSM) View of Same

- Instead of just showing which attributes are coupled, the DSM (like the Parametric Diagram) can also symbolize the named Coupling that connects them:
  - This provides a reference to a (separately documented) quantitative coupling description.
- The names of the couplings can be introduced as row and column headings, separate from the rows and columns that list the attribute names:
  - This becomes a Multi-Domain Matrix (MDM):



Requirement Statements

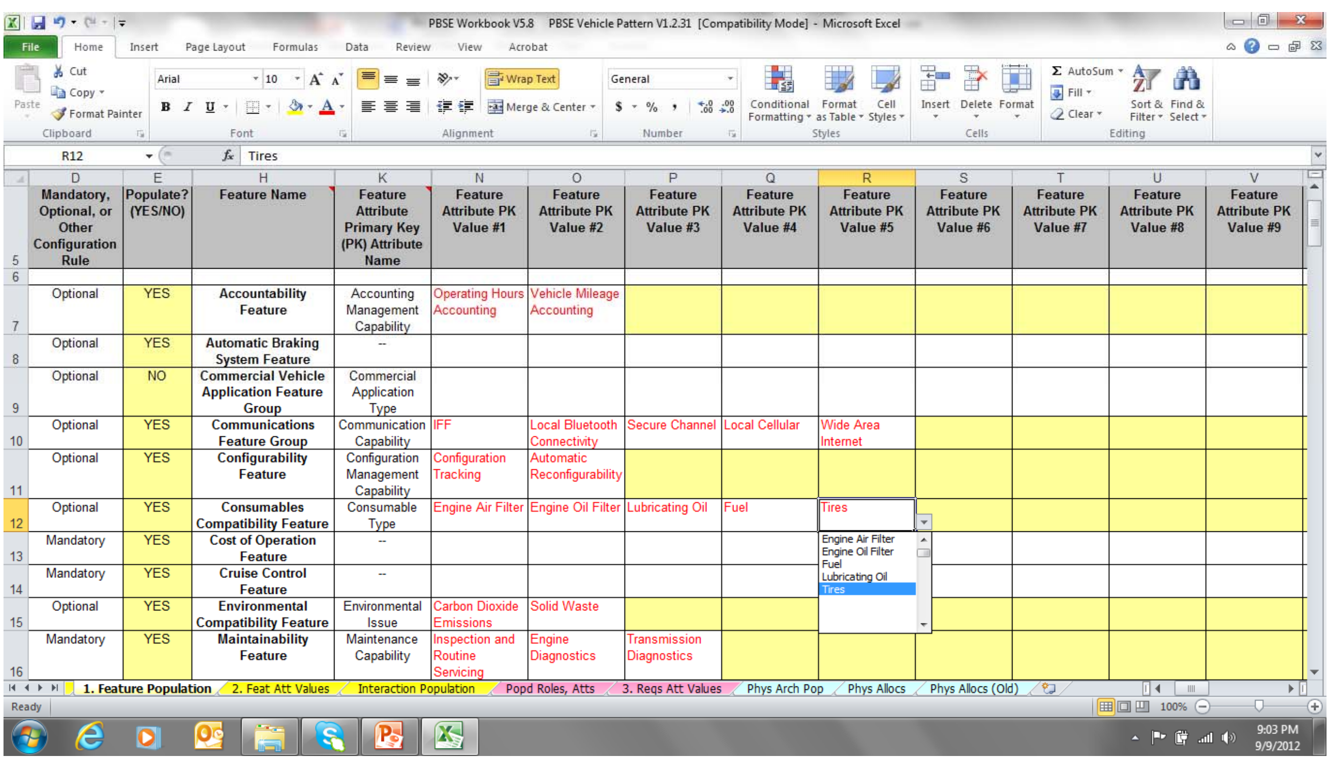


Reqs Container ID	Functional Interaction	IPK Value	Functional Role	Req. ID	Req PK Value Rule	Requirement Statement	Functional Failure (Reverse Requirement)	De
	Travel Over Terrain		Vehicle	VEH-1135		The vehicle, loaded with its passenger and other load maximum, shall be capable of stopping from a speed of 60 miles per hour in 200 feet on dry pavement.		
	Travel Over Terrain		Vehicle	VEH-1136		The vehicle shall be capable of operating 5,000 miles between oil changes		
	Travel Over Terrain		Vehicle	VEH-1137		The vehicle shall be capable of operating 50,000 miles between tire changes.		
	Travel Over Terrain		Vehicle	VEH-1138		The vehicle shall be capable of operating 25,000 miles between air filter changes.		
	Travel Over Terrain		Vehicle	VEH-1139		The vehicle shall be capable of operating 5,000 miles between oil filter changes		
	Travel Over Terrain	Reliability Availability	Vehicle	VEH-1168		The basic transport functions of the vehicle shall be available for use with scheduled down time not to exceed 60 hours per year, when subject to planned maintenance.		
	Travel Over Terrain	Reliability Availability	Vehicle	VEH-1169		The basic transport functions of the vehicle shall be available for use with scheduled down time not to exceed 10 hours per year, when subject to planned maintenance.		
	Travel Over Terrain	Reliability Availability	Vehicle	VEH-1170		The basic transport functions of the vehicle shall be deliverable by the system during a design life of 15 years, assuming planned maintenance is provided.		
	Travel Over Terrain	Reliability Availability	Vehicle	VEH-1171		The basic transport functions of the vehicle shall be available with 97% reliability, over the design life of the system, assuming planned maintenance is provided.		

Failure Modes Model

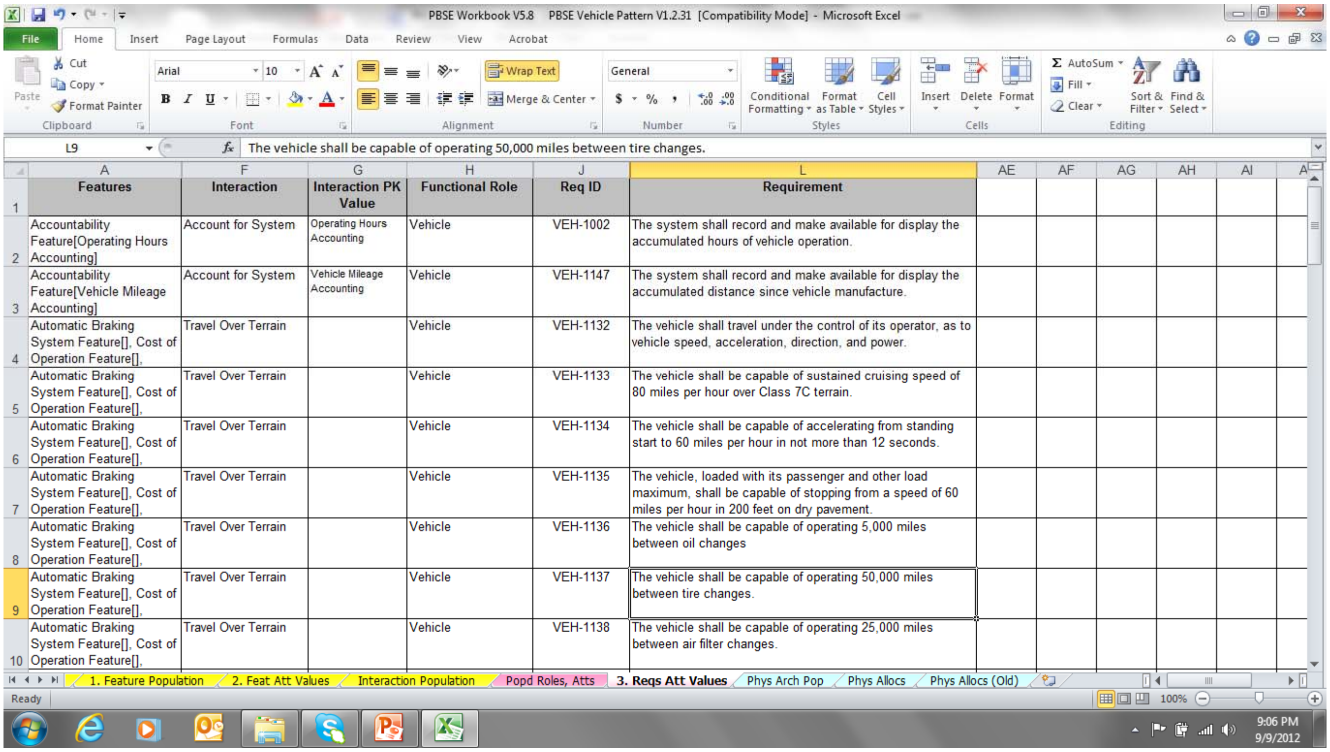
Physical Entity	Failure Mode
Vehicle ECM	Dead ECM
Vehicle ECM	Network Connector Open
Vehicle ECM	Network Connector Short
Vehicle ECM	Erratic ECM
Battery	Discharged Battery
Battery	Battery Cell Short
Battery	Battery Cell Open
Battery	Battery Leak
Panel Display	Fractured Display
Panel Display	Illuminator Fail
Bluetooth Module	Module Hard Fail
Bluetooth Module	Transmitter Fail
Bluetooth Module	Receiver Fail

Filling in the Feature Population Form—with Stakeholder Needs



	D	E	H	K	N	O	P	Q	R	S	T	U	V
5	Mandatory, Optional, or Other Configuration Rule	Populate? (YES/NO)	Feature Name	Feature Attribute Primary Key (PK) Attribute Name	Feature Attribute PK Value #1	Feature Attribute PK Value #2	Feature Attribute PK Value #3	Feature Attribute PK Value #4	Feature Attribute PK Value #5	Feature Attribute PK Value #6	Feature Attribute PK Value #7	Feature Attribute PK Value #8	Feature Attribute PK Value #9
6													
7	Optional	YES	Accountability Feature	Accounting Management Capability	Operating Hours Accounting	Vehicle Mileage Accounting							
8	Optional	YES	Automatic Braking System Feature	--									
9	Optional	NO	Commercial Vehicle Application Feature Group	Commercial Application Type									
10	Optional	YES	Communications Feature Group	Communication Capability	IFF	Local Bluetooth Connectivity	Secure Channel	Local Cellular	Wide Area Internet				
11	Optional	YES	Configurability Feature	Configuration Management Capability	Configuration Tracking	Automatic Reconfigurability							
12	Optional	YES	Consumables Compatibility Feature	Consumable Type	Engine Air Filter	Engine Oil Filter	Lubricating Oil	Fuel	Tires				
13	Mandatory	YES	Cost of Operation Feature	--									
14	Mandatory	YES	Cruise Control Feature	--									
15	Optional	YES	Environmental Compatibility Feature	Environmental Issue	Carbon Dioxide Emissions	Solid Waste							
16	Mandatory	YES	Maintainability Feature	Maintenance Capability	Inspection and Routine Servicing	Engine Diagnostics	Transmission Diagnostics						

Resulting Auto-Populated Requirements



	Features	Interaction	Interaction PK Value	Functional Role	Req ID	Requirement
2	Accountability Feature[Operating Hours Accounting]	Account for System	Operating Hours Accounting	Vehicle	VEH-1002	The system shall record and make available for display the accumulated hours of vehicle operation.
3	Accountability Feature[Vehicle Mileage Accounting]	Account for System	Vehicle Mileage Accounting	Vehicle	VEH-1147	The system shall record and make available for display the accumulated distance since vehicle manufacture.
4	Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1132	The vehicle shall travel under the control of its operator, as to vehicle speed, acceleration, direction, and power.
5	Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1133	The vehicle shall be capable of sustained cruising speed of 80 miles per hour over Class 7C terrain.
6	Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1134	The vehicle shall be capable of accelerating from standing start to 60 miles per hour in not more than 12 seconds.
7	Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1135	The vehicle, loaded with its passenger and other load maximum, shall be capable of stopping from a speed of 60 miles per hour in 200 feet on dry pavement.
8	Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1136	The vehicle shall be capable of operating 5,000 miles between oil changes
9	Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1137	The vehicle shall be capable of operating 50,000 miles between tire changes.
10	Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1138	The vehicle shall be capable of operating 25,000 miles between air filter changes.

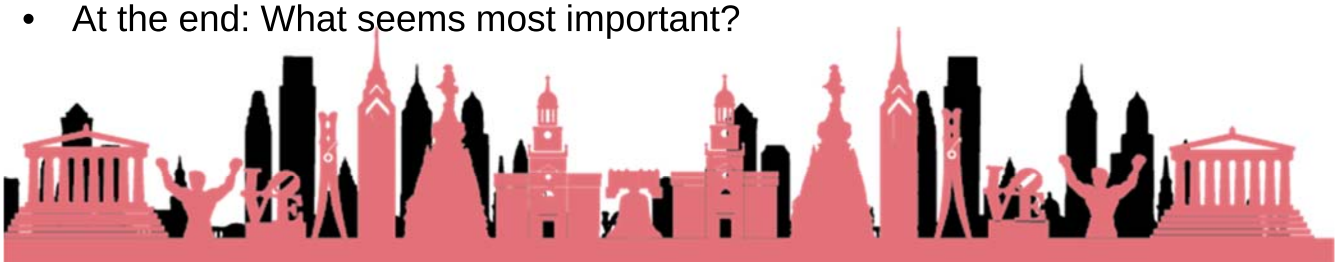
Break out: Practice exercise

- For the Vehicle Pattern:
  - Think of some Vehicle Application
  - Fill in the Feature Configuration Form for your application
  - Did you need any new Features not in the Vehicle Pattern?
- For your own Pattern: Interactions
  - Think of a new Interaction between the Vehicle and some Actor (you can add a new Actor)
  - Create an Interaction Diagram
  - Write requirements on the Vehicle for this Interaction
- Group Discussion of Exercise

Applying system patterns

- Example Uses and Benefits:
  1. Stakeholder Features and Scenarios: Better stakeholder alignment sooner
  2. Pattern Configuration: Generating better requirements faster
  3. Selecting Solutions: More informed trade-offs
  4. Design for Change: Analyzing and improving platform resiliency
  5. Risk Analysis: Pattern-enabled FMEAs
  6. Verification: Generating better tests faster

- At the end: What seems most important?

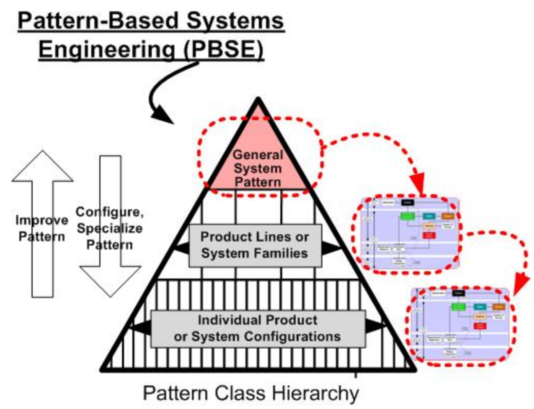


1. Stakeholder Features and Scenarios: Better stakeholders alignment sooner

- Alignment with stakeholders is critical to program success.
- That alignment can be achieved earlier and maintained stronger using:
  - Stakeholder Feature Pattern: Aligns understanding of system capabilities (base as well as options) and the nature of their value to stakeholders
  - Scenario Pattern: Aligns understanding of the concepts of operations, support, manufacture, distribution, other life cycle situations; accelerates alignment of system documentation, training, and communication.
- Both of these are “pattern configurations” directly generated from the System Pattern—not separate and unsynchronized information.

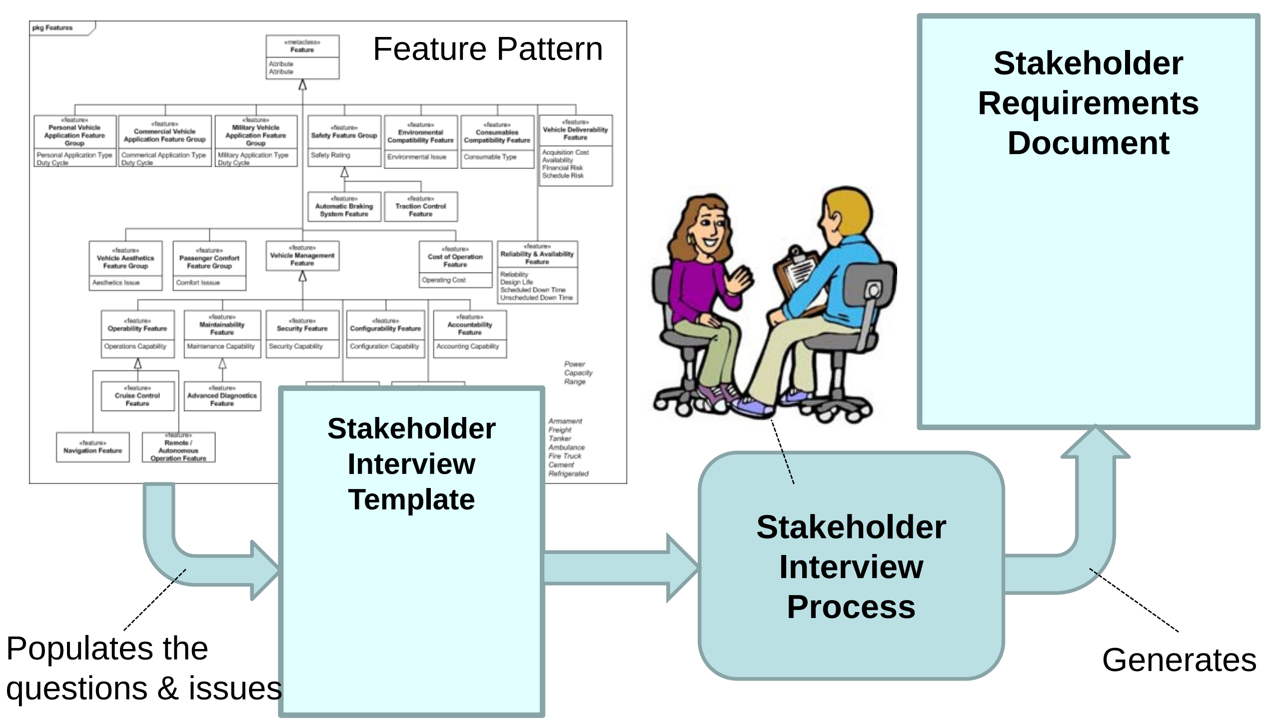
1. Using the Feature Pattern to Rapidly Capture & Validate Stakeholder Requirements: An Example

- Concept: The Feature Pattern is a powerful tool for establishing Stakeholder Requirements—as a “configuration” of Feature Pattern.
- By “configuration”, we mean that individual Features from the Pattern are (1) either populated or de-populated, and (2) their Feature Attributes (parameters) are given values:

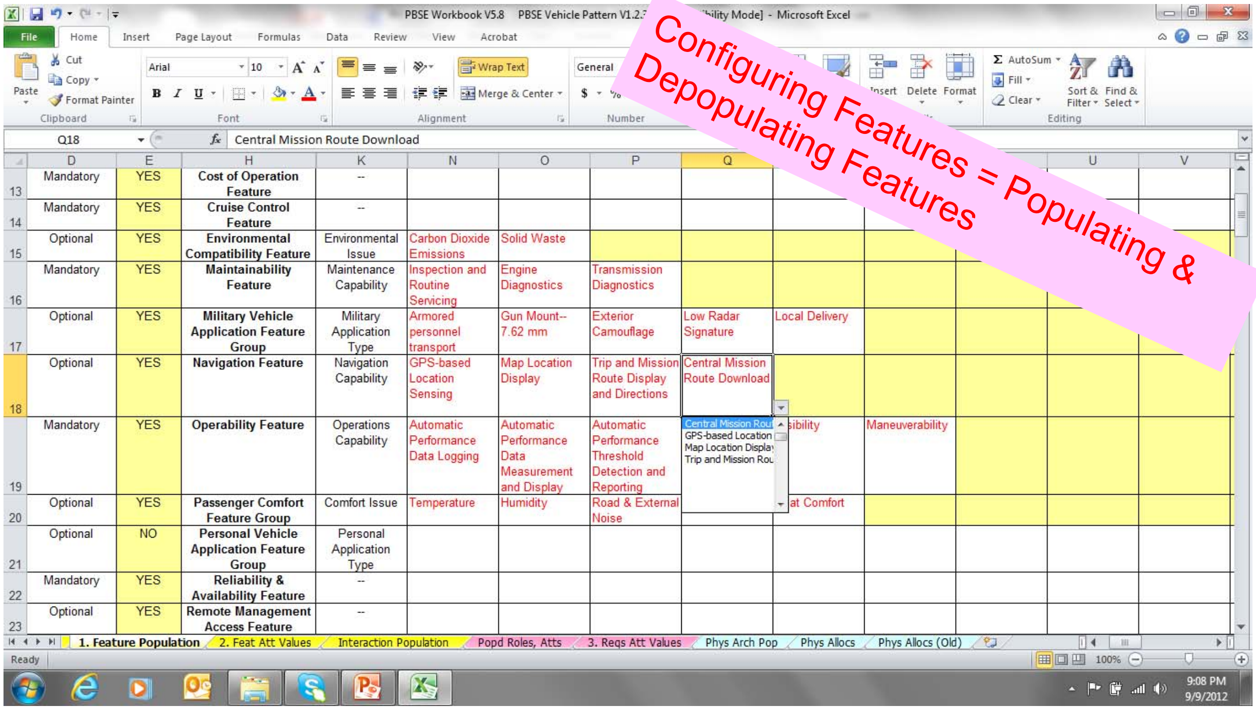


- These can be expressed (1) as configured Feature objects and their attribute values or (2) as sentence-type statements if desired, but in any case the degrees of freedom (stakeholder choices) are brought into clear focus.

Using the Feature Pattern to Rapidly Capture & Validate Stakeholder Requirements: An Example



1. Using the Feature Pattern to Rapidly Capture & Validate Stakeholder Requirements: An Example



Row	D	E	H	K	N	O	P	Q	R	S
13	Mandatory	YES	Cost of Operation Feature	--						
14	Mandatory	YES	Cruise Control Feature	--						
15	Optional	YES	Environmental Compatibility Feature	Environmental Issue	Carbon Dioxide Emissions	Solid Waste				
16	Mandatory	YES	Maintainability Feature	Maintenance Capability	Inspection and Routine Servicing	Engine Diagnostics	Transmission Diagnostics			
17	Optional	YES	Military Vehicle Application Feature Group	Military Application Type	Armored personnel transport	Gun Mount-- 7.62 mm	Exterior Camouflage	Low Radar Signature	Local Delivery	
18	Optional	YES	Navigation Feature	Navigation Capability	GPS-based Location Sensing	Map Location Display	Trip and Mission Route Display and Directions	Central Mission Route Download		
19	Mandatory	YES	Operability Feature	Operations Capability	Automatic Performance Data Logging	Automatic Performance Data Measurement and Display	Automatic Performance Threshold Detection and Reporting	Central Mission Rou / GPS-based Location / Map Location Display / Trip and Mission Rou	...ibility	Maneuverability
20	Optional	YES	Passenger Comfort Feature Group	Comfort Issue	Temperature	Humidity	Road & External Noise		...at Comfort	
21	Optional	NO	Personal Vehicle Application Feature Group	Personal Application Type						
22	Mandatory	YES	Reliability & Availability Feature	--						
23	Optional	YES	Remote Management Access Feature	--						

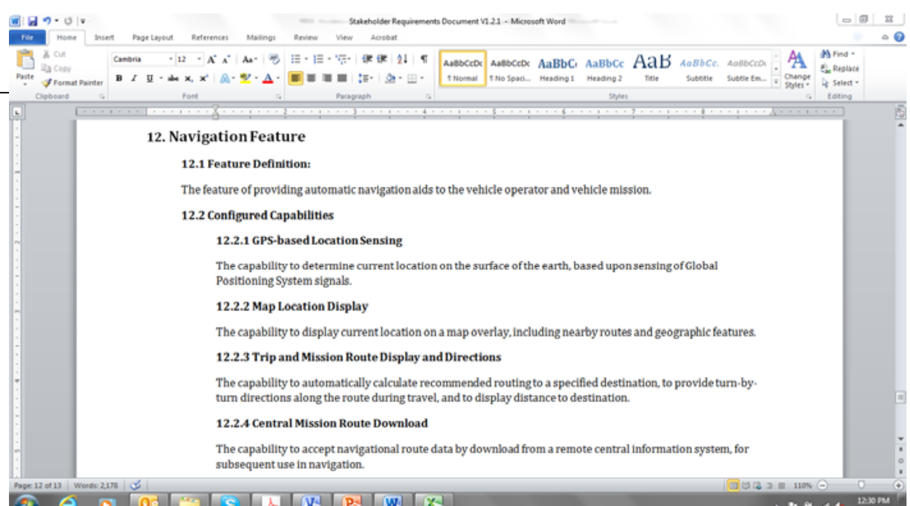
1. Using the Feature Pattern to Rapidly Capture & Validate Stakeholder Requirements: An Example

Stakeholder Requirements Document

Military Vehicle Configuration Baker

Version: 1.2.1

03 May 2012



12. Navigation Feature

12.1 Feature Definition:

The feature of providing automatic navigation aids to the vehicle operator and vehicle mission.

12.2 Configured Capabilities

12.2.1 GPS-based Location Sensing

The capability to determine current location on the surface of the earth, based upon sensing of Global Positioning System signals.

12.2.2 Map Location Display

The capability to display current location on a map overlay, including nearby routes and geographic features.

12.2.3 Trip and Mission Route Display and Directions

The capability to automatically calculate recommended routing to a specified destination, to provide turn-by-turn directions along the route during travel, and to display distance to destination.

12.2.4 Central Mission Route Download

The capability to accept navigational route data by download from a remote central information system, for subsequent use in navigation.

1. Using the Feature Pattern to Rapidly Capture & Validate Stakeholder Requirements

- Benefits:
  - A more complete set of stakeholder requirements—reduce omissions;
  - Stronger alignment with stakeholders, sooner—surface issues earlier;
  - Pattern identifies classes of stakeholders that might have been missed;
  - Pattern makes very clear the difference between Stakeholder Requirements versus Design Constraints or Technical Requirements;
  - The Pattern provides a clear place to accumulate new learning (e.g., additional Features);
  - Sets up subsequent uses of Feature Pattern in support of Trade Space, Risk Management, and other applications.
- No free lunch:
  - Interviewer needs to be knowledgeable about the Features;
  - Stakeholders won't have all the answers—find the right representative;
  - Stakeholder representatives need know they are formal representatives;
  - The Feature Pattern needs to be relatively complete.

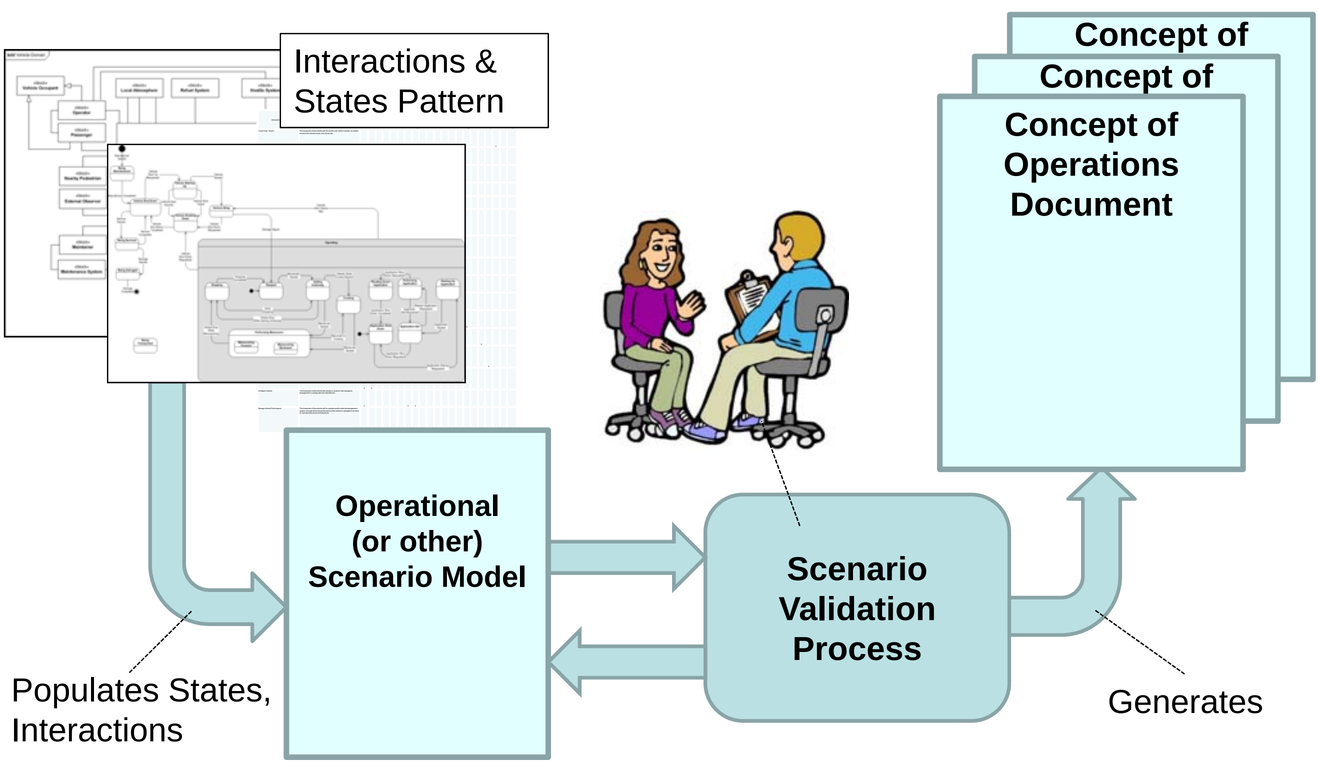
How do I know whether I have all the Features?

- This is why we use a Pattern!
  - Moves problem to the builder of the original pattern.
- Related key points for the builder of the Feature Pattern:
  - First, identify all the Stakeholder classes
  - Then, all the Features for each Stakeholder class
  - Validate the Features with their Stakeholders
  - Then, make sure all the Interactions are reviewed for associated Feature value
  - There are well-known abstract Feature classes (e.g., Maintainability)
- Every time we discover another Feature, we add it to the Pattern; for example:
  - Every argument / decision should invoke trade space Features as its ultimate rationale – a new one might appear during an argument.
  - Every impactful Failure Mode should cause Feature impacting Effects – a new one might appear while discussing a Failure Mode.

1. Using the Interactions & States Pattern to Rapidly Generate & Validate Scenarios: An Example

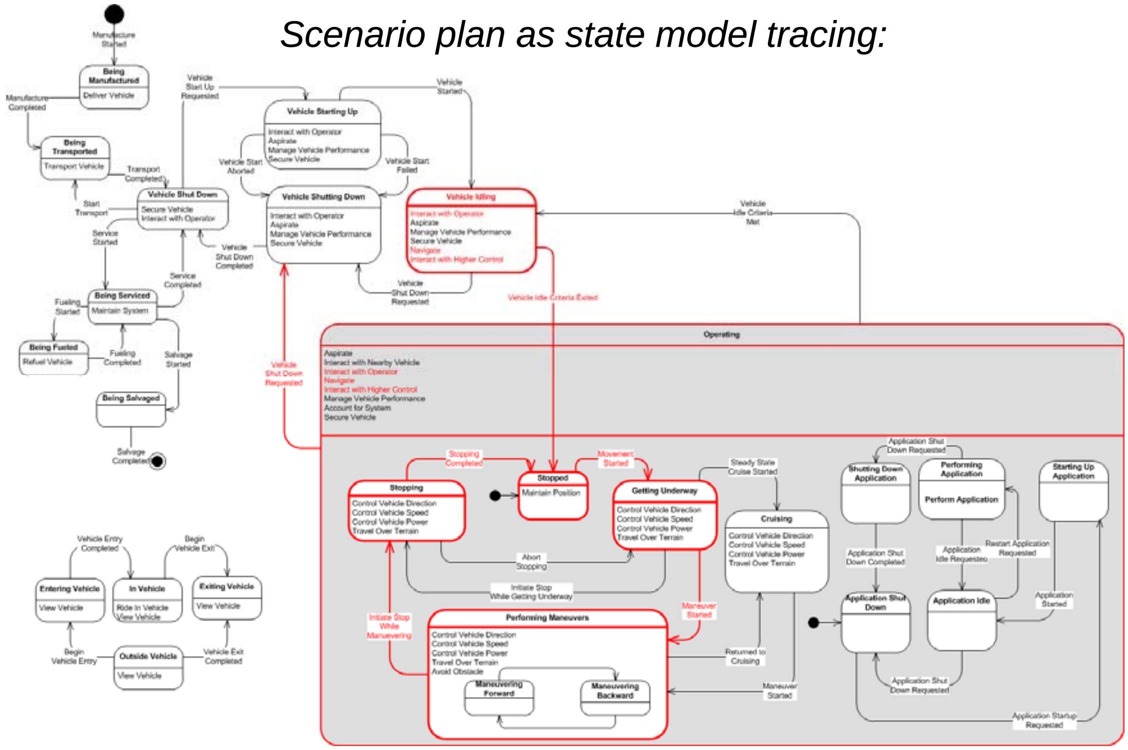
- Concept: Scenarios can be efficiently generated, as single thread tracings through the configured pattern State Model;
- Each scenario "tells a story" within the system's life cycle—operations, maintenance, or other CONOPS type view;
- Early in life cycle: Stakeholders validate (or give feedback) scenario;
- Later in life cycle: Generates base data for training and documentation, as well as test plans;
- Akin to typical Use Case process, but easier maintained ongoing as a part of the configured pattern;
- Reference: Operational Views (OV)

1. Using the Interactions & States Pattern to Rapidly Generate & Validate Scenarios: An Example



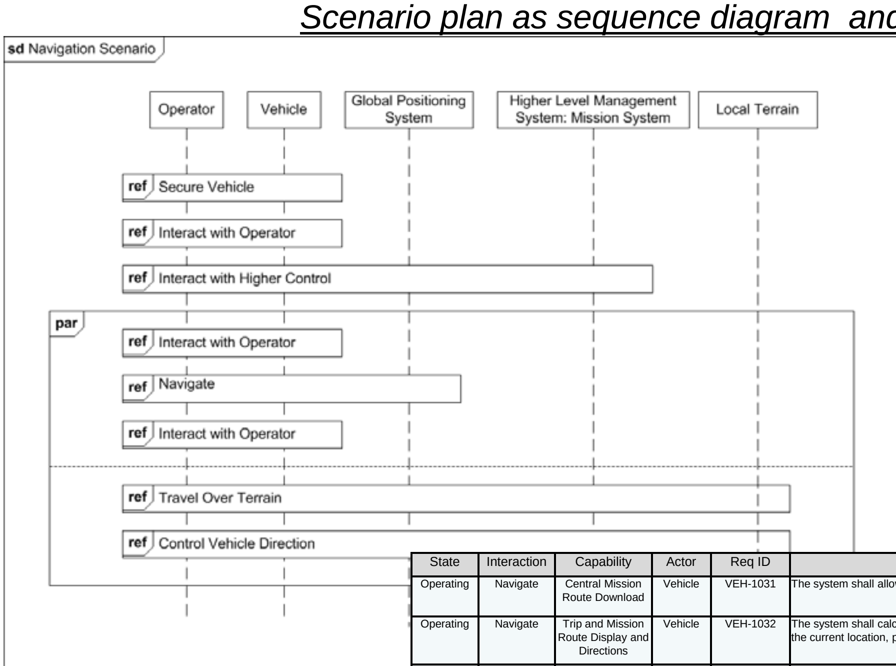
1. Using the Interactions & States Pattern to Rapidly Generate & Validate Scenarios: An Example

Scenario plan as state model tracing:



1. Using the Interactions & States Pattern to Rapidly Generate & Validate Scenarios: An Example

Scenario plan as sequence diagram and requirements:



State	Interaction	Capability	Actor	Req ID	Requirement
Operating	Navigate	Central Mission Route Download	Vehicle	VEH-1031	The system shall allow the operator to select a pre-stored route for travel on a mission.
Operating	Navigate	Trip and Mission Route Display and Directions	Vehicle	VEH-1032	The system shall calculate and display a recommended route to an operator-specified destination from the current location, providing turn-by-turn en route directions and progress tracking.
Operating	Navigate	GPS-based Location Sensing	Vehicle	VEH-1029	The system shall sense the location of the vehicle by accessing the Global Positioning System (GPS) satellite constellation and computing location on the surface of the earth, accurate to 10 feet.
Operating	Navigate	Map Location Display	Vehicle	VEH-1030	The system shall display position of the vehicle on a pre-stored graphic map presentation, including major road and geographic features, updating while enroute to reflect travel of the vehicle.
Operating	Navigate	GPS-based Location Sensing	Vehicle	VEH-1033	The system shall display to the vehicle operator a location confidence indicator, signaling whether accurate GPS location sensing is currently available.

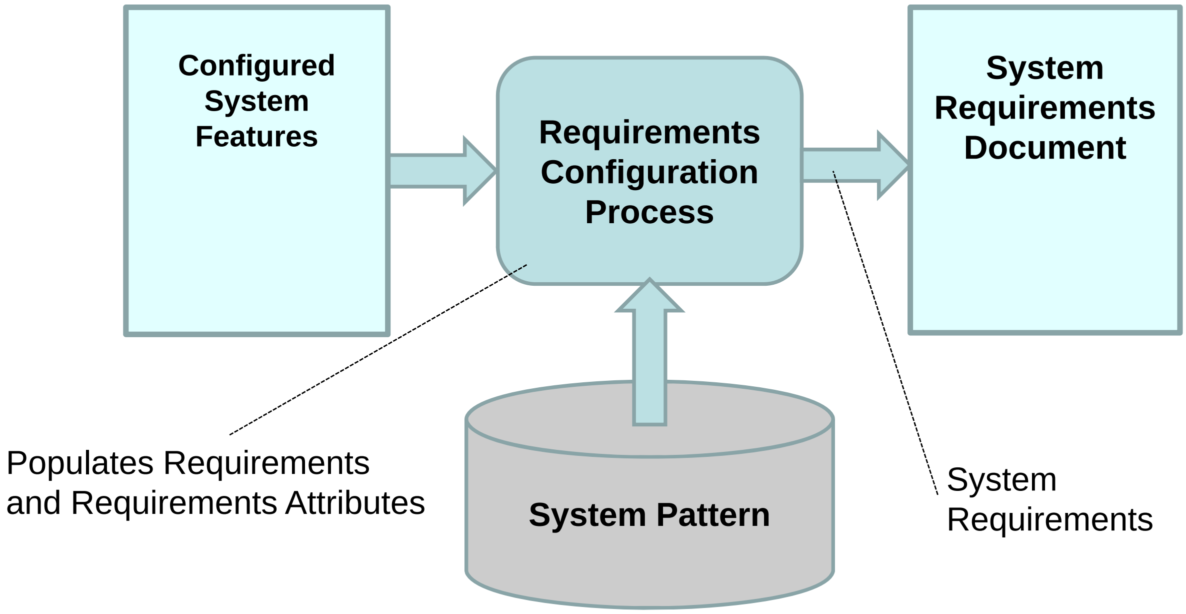
1. Using the Interactions & States Pattern to Rapidly Generate & Validate Scenarios

- Benefits:
  - A more complete set of scenarios—reduces omissions;
  - Easier to generate from pattern;
  - Easier to keep consistent with configured system model as it evolves over the delivery and life cycle;
  - Valuable not only for initial validation, but also as seed information for generation of system training, documentation, SOPs;
  - As system requirements are configured, becomes progressively more detailed;
  - The Pattern provides a clear place to accumulate new learning (e.g., additional Scenarios);
- No free lunch:
  - The State and Interaction Pattern needs to be relatively complete.

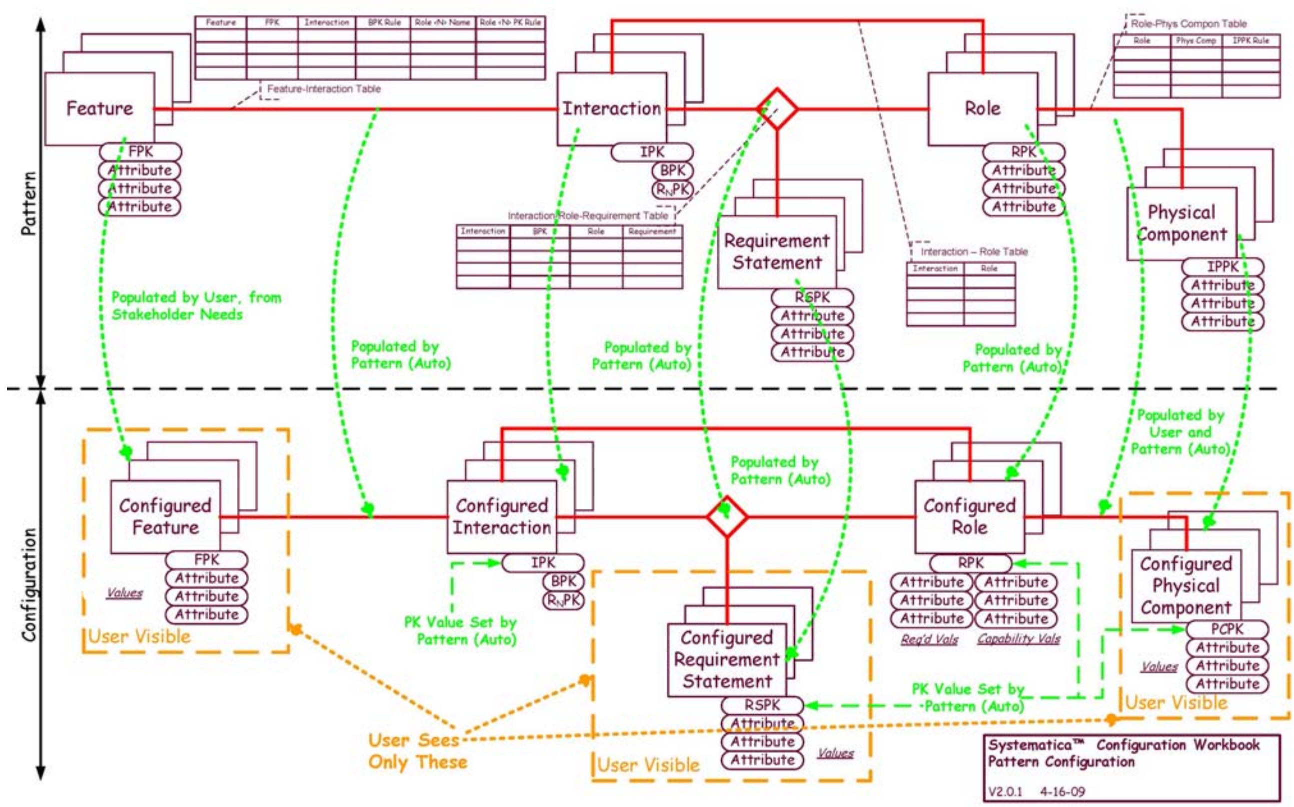
2. Using Pattern Configuration to generate better System Requirements faster: Example

- Concept: Configured System Requirements can be semi-automatically generated from Configured Features, using the System Pattern;
- Low dimensionality / degrees of freedom choices in Feature stakeholder space imply higher dimensionality / degrees of freedom choices in Requirements space:
  - The difference is made up by relationships encoded in the Pattern.

2. Using Pattern Configuration to generate better System Requirements faster: Example

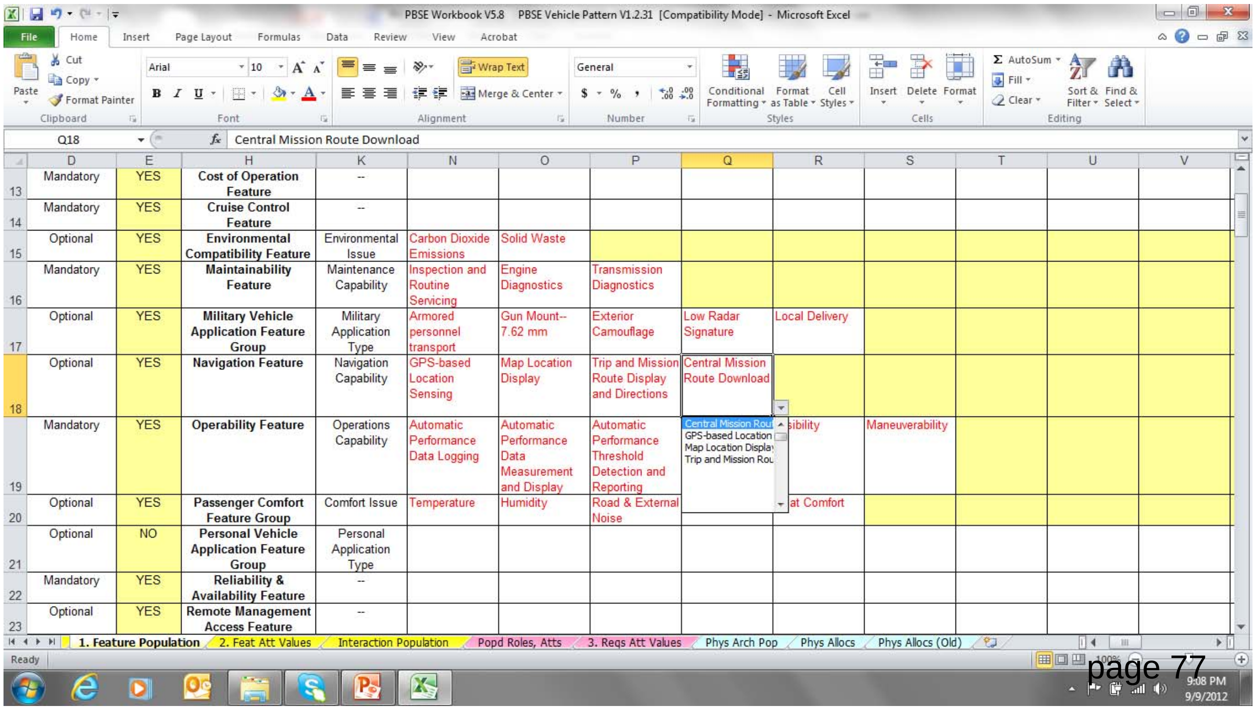


- The S*Pattern links Features to Requirements:
  - This means that populating a configuration of Features can automatically populate a configuration of Requirements--



2. Using the Feature Pattern to Rapidly Capture & Validate Stakeholder Requirements: An Example

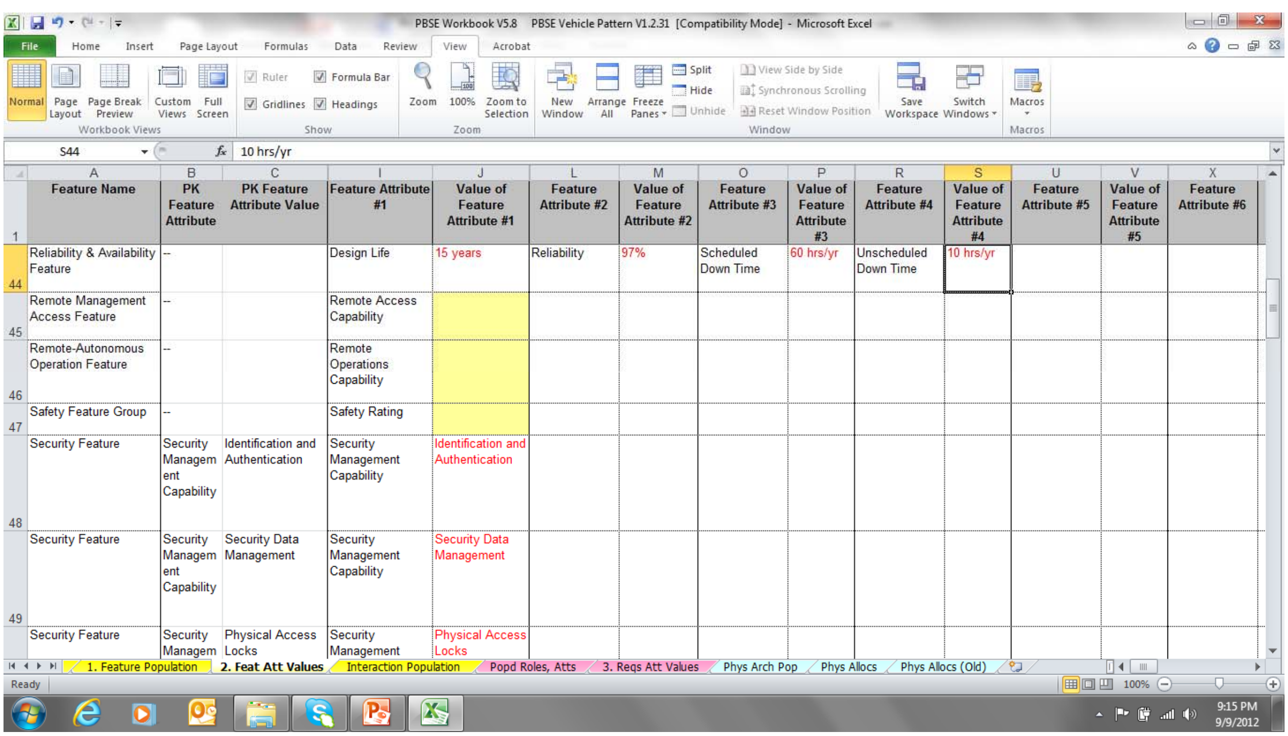
Populating / depopulating Features:



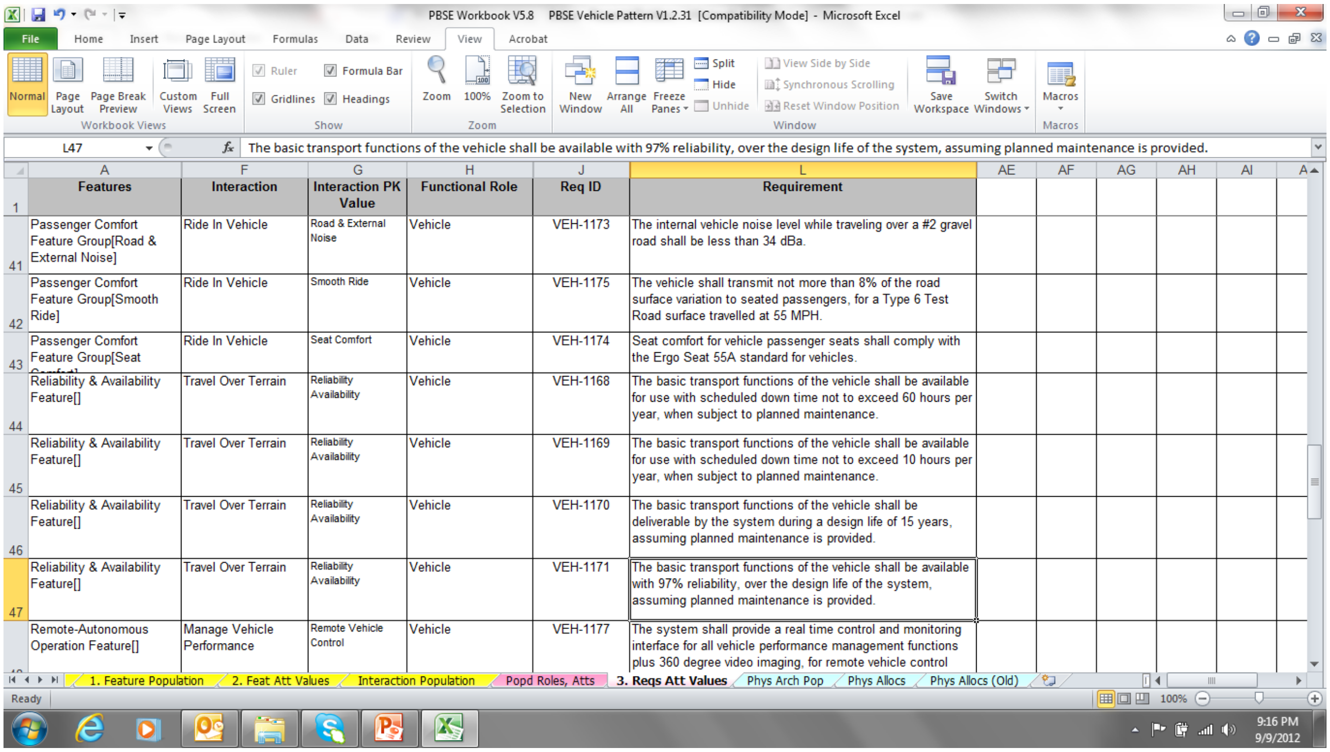
Row	D	E	H	K	N	O	P	Q	R	S	T	U	V
13	Mandatory	YES	Cost of Operation Feature	--									
14	Mandatory	YES	Cruise Control Feature	--									
15	Optional	YES	Environmental Compatibility Feature	Environmental Issue	Carbon Dioxide Emissions	Solid Waste							
16	Mandatory	YES	Maintainability Feature	Maintenance Capability	Inspection and Routine Servicing	Engine Diagnostics	Transmission Diagnostics						
17	Optional	YES	Military Vehicle Application Feature Group	Military Application Type	Armored personnel transport	Gun Mount-- 7.62 mm	Exterior Camouflage	Low Radar Signature	Local Delivery				
18	Optional	YES	Navigation Feature	Navigation Capability	GPS-based Location Sensing	Map Location Display	Trip and Mission Route Display and Directions	Central Mission Route Download					
19	Mandatory	YES	Operability Feature	Operations Capability	Automatic Performance Data Logging	Automatic Performance Data Measurement and Display	Automatic Performance Threshold Detection and Reporting	Central Mission Rout GPS-based Location Map Location Displa Trip and Mission Rou	...ibility	Maneuverability			
20	Optional	YES	Passenger Comfort Feature Group	Comfort Issue	Temperature	Humidity	Road & External Noise		...at Comfort				
21	Optional	NO	Personal Vehicle Application Feature Group	Personal Application Type									
22	Mandatory	YES	Reliability & Availability Feature	--									
23	Optional	YES	Remote Management Access Feature	--									

2. Using the Feature Pattern to Rapidly Capture & Validate Stakeholder Requirements: An Example

Configuring Features: Setting Feature Attribute Values



Row	Feature Name	PK Feature Attribute	PK Feature Attribute Value	Feature Attribute #1	Value of Feature Attribute #1	Feature Attribute #2	Value of Feature Attribute #2	Feature Attribute #3	Value of Feature Attribute #3	Feature Attribute #4	Value of Feature Attribute #4	Feature Attribute #5	Value of Feature Attribute #5	Feature Attribute #6
44	Reliability & Availability Feature	--		Design Life	15 years	Reliability	97%	Scheduled Down Time	60 hrs/yr	Unscheduled Down Time	10 hrs/yr			
45	Remote Management Access Feature	--		Remote Access Capability										
46	Remote-Autonomous Operation Feature	--		Remote Operations Capability										
47	Safety Feature Group	--		Safety Rating										
48	Security Feature	Security Management Capability	Identification and Authentication	Security Management Capability	Identification and Authentication									
49	Security Feature	Security Management Capability	Security Data Management	Security Management Capability	Security Data Management									
	Security Feature	Security Managem	Physical Access Locks	Security Management	Physical Access Locks									



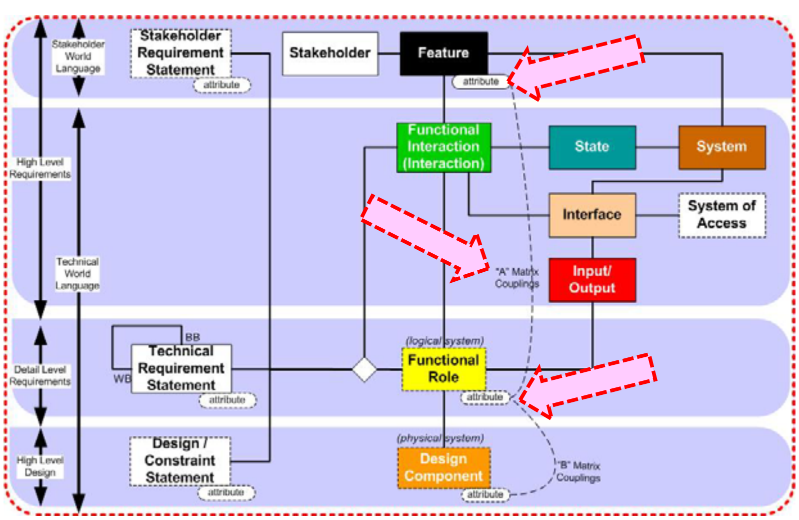
Row	Features	Interaction	Interaction PK Value	Functional Role	Req ID	Requirement
41	Passenger Comfort Feature Group[Road & External Noise]	Ride In Vehicle	Road & External Noise	Vehicle	VEH-1173	The internal vehicle noise level while traveling over a #2 gravel road shall be less than 34 dBa.
42	Passenger Comfort Feature Group[Smooth Ride]	Ride In Vehicle	Smooth Ride	Vehicle	VEH-1175	The vehicle shall transmit not more than 8% of the road surface variation to seated passengers, for a Type 6 Test Road surface travelled at 55 MPH.
43	Passenger Comfort Feature Group[Seat	Ride In Vehicle	Seat Comfort	Vehicle	VEH-1174	Seat comfort for vehicle passenger seats shall comply with the Ergo Seat 55A standard for vehicles.
44	Reliability & Availability Feature[]	Travel Over Terrain	Reliability Availability	Vehicle	VEH-1168	The basic transport functions of the vehicle shall be available for use with scheduled down time not to exceed 60 hours per year, when subject to planned maintenance.
45	Reliability & Availability Feature[]	Travel Over Terrain	Reliability Availability	Vehicle	VEH-1169	The basic transport functions of the vehicle shall be available for use with scheduled down time not to exceed 10 hours per year, when subject to planned maintenance.
46	Reliability & Availability Feature[]	Travel Over Terrain	Reliability Availability	Vehicle	VEH-1170	The basic transport functions of the vehicle shall be deliverable by the system during a design life of 15 years, assuming planned maintenance is provided.
47	Reliability & Availability Feature[]	Travel Over Terrain	Reliability Availability	Vehicle	VEH-1171	The basic transport functions of the vehicle shall be available with 97% reliability, over the design life of the system, assuming planned maintenance is provided.
	Remote-Autonomous Operation Feature[]	Manage Vehicle Performance	Remote Vehicle Control	Vehicle	VEH-1177	The system shall provide a real time control and monitoring interface for all vehicle performance management functions plus 360 degree video imaging, for remote vehicle control

- Resulting Requirements:

  Attribute values can also be set, in line or in tables . . . .

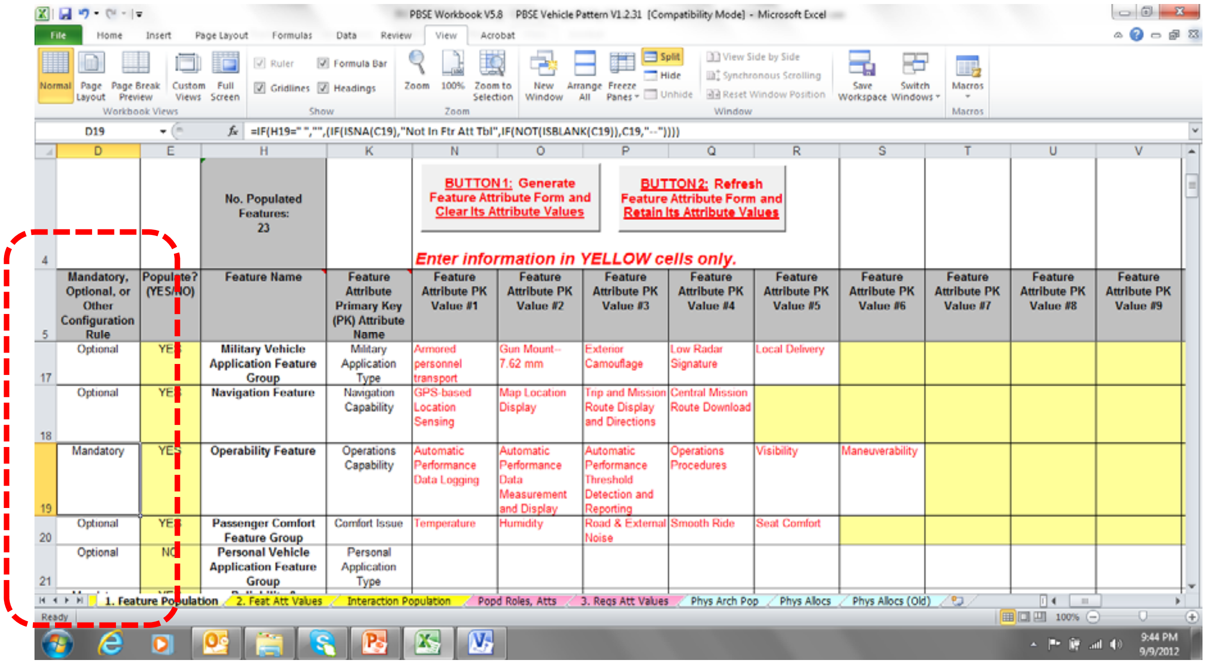
2. Using Pattern Configuration to generate better System Requirements faster: Example

- Requirements Attribute Value Setting:
  - A part of the configuration process
  - Example: Cruise Control Speed Stability
  - In PBSE, requirements attribute value setting can be manual, semi-automatic, or automatic—in all cases, driven by Feature Attribute Values and Attribute Couplings:



2. Using Pattern Configuration to generate better System Requirements faster: Example

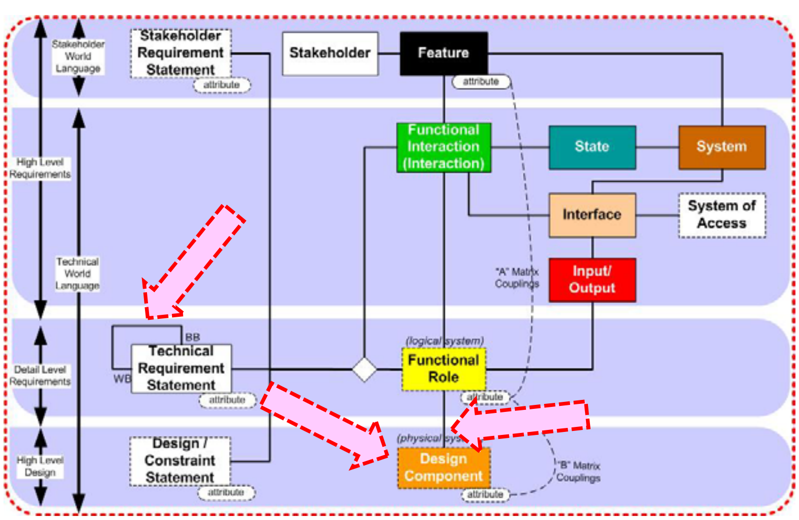
In general, Configuration Rules are found in the Relationships that associate the model Classes, and also those that associate the model Attributes:



Mandatory, Optional, or Other Configuration Rule	Populate? (YES/NO)	Feature Name	Feature Attribute Primary Key (PK) Attribute Name	Feature Attribute PK Value #1	Feature Attribute PK Value #2	Feature Attribute PK Value #3	Feature Attribute PK Value #4	Feature Attribute PK Value #5	Feature Attribute PK Value #6	Feature Attribute PK Value #7	Feature Attribute PK Value #8	Feature Attribute PK Value #9
Optional	YES	Military Vehicle Application Feature Group	Military Application Type	Armored personnel transport	Gun Mount--7.62 mm	Exterior Camouflage	Low Radar Signature	Local Delivery				
Optional	YES	Navigation Feature	Navigation Capability	GPS-based Location Sensing	Map Location Display	Trip and Mission Route Display and Directions	Central Mission Route Download					
Mandatory	YES	Operability Feature	Operations Capability	Automatic Performance Data Logging	Automatic Performance Data Measurement and Display	Automatic Performance Threshold Detection and Reporting	Operations Procedures	Visibility	Maneuverability			
Optional	YES	Passenger Comfort Feature Group	Comfort Issue	Temperature	Humidity	Road & External Noise	Smooth Ride	Seat Comfort				
Optional	NO	Personal Vehicle Application Feature Group	Personal Application Type									

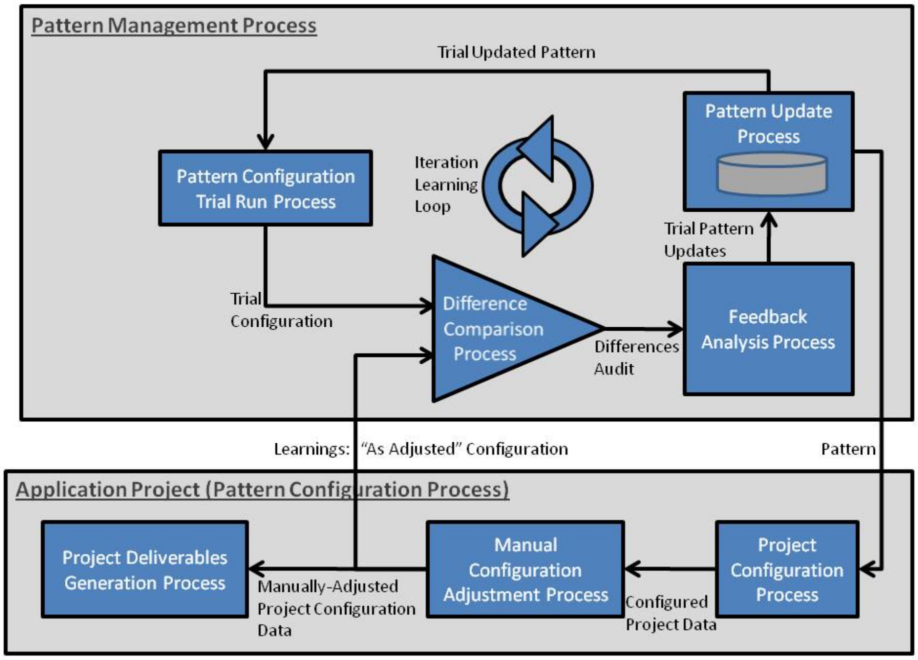
2. Using Pattern Configuration to generate better System Requirements faster

- The scope of a System Pattern can include more than Requirements:
  - Design Patterns include Physical Architecture, Requirements Decomposition, Requirements Allocations:



2. Using Pattern Configuration to generate better System Requirements faster

- PBSE processes continuously improve the content of the pattern, accumulating lessons for use in future projects:



3. Selecting Solutions

More Informed Trade-offs

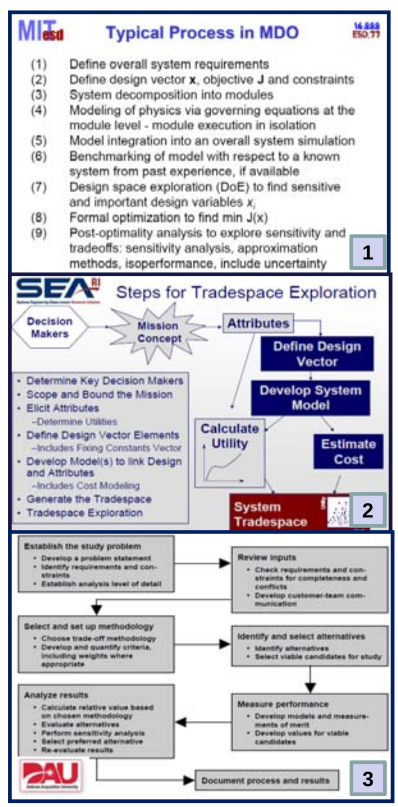
Introduction:

Understanding trade-offs are an essential and critical part of engineering systems

Trades include many formalized methodologies to make informed decisions

Trade-offs seek to:

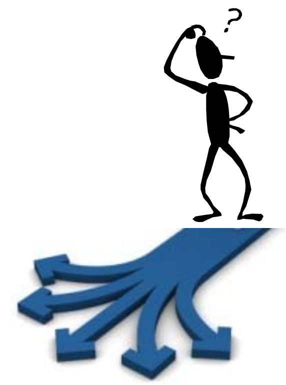
- Identify practical alternatives / optimal solutions
- Resolve conflicting objectives
- Account for the full spectrum of stakeholder needs to ensure a balanced system solution
- Methods incorporate identifying/defining stakeholders, requirements, values, attributes, metrics, costs, governing equations, interactions etc.



1. *Bullets from MIT, ESD.77 MDO Course, Oli deWeck*
2. *SEARI Ref: http://seari.mit.edu/short_courses.php#value*
3. *Defense Acquisition University SE Handbook Trades Studies process*

3. Selecting Solutions:

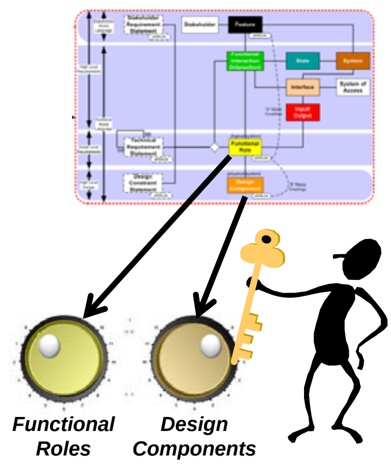
More Informed Trade-offs



Concept:

Patterns provide a very quick and explicit way to perform trades

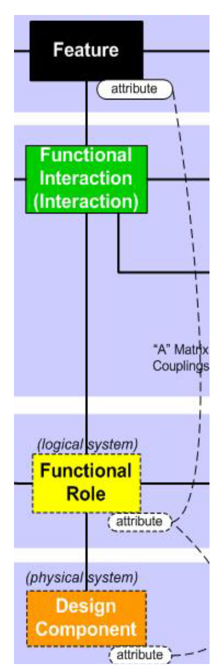
- Patterns contain the essential information to identify and assess systems solutions
- Enable the rapid creation and comparison of multiple system configurations
- Patterns save time in collection, integration and structuring of the required information to perform trade-offs
- Patterns provide leverage across programs and promote consistency
- PBSE enables feature space optimization through the turning of knobs in the logical and design component space



3. Selecting Solutions

More Informed Trade-offs

PBSE and Trades



Feature Space

- Makes explicit all stakeholder needs
- Quantifies value impact through attributes
- Contains the entire trade space

Functional Role / Logical Architecture

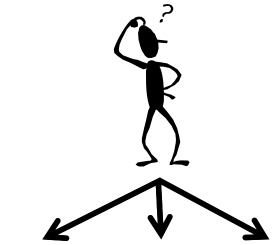
- Logical, independent of design
- Describes the system's behavioral structure
- Formally models subsystems/design components
- Houses performance data (range, cost, weight etc.)
- Supports modeling of multiple physical architectures

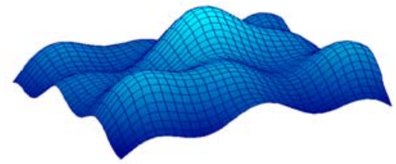
Design Components

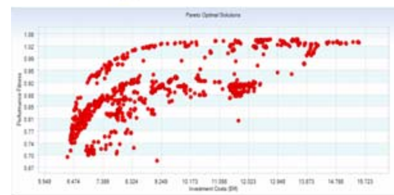
- Contains subsystem and technology options
- Design component options populate the logical architecture to create system configurations
- Contains part numbers, option names etc.
- Models the physical architecture

3. Selecting Solutions:
More Informed Trade-offs

Vehicle Trades Example



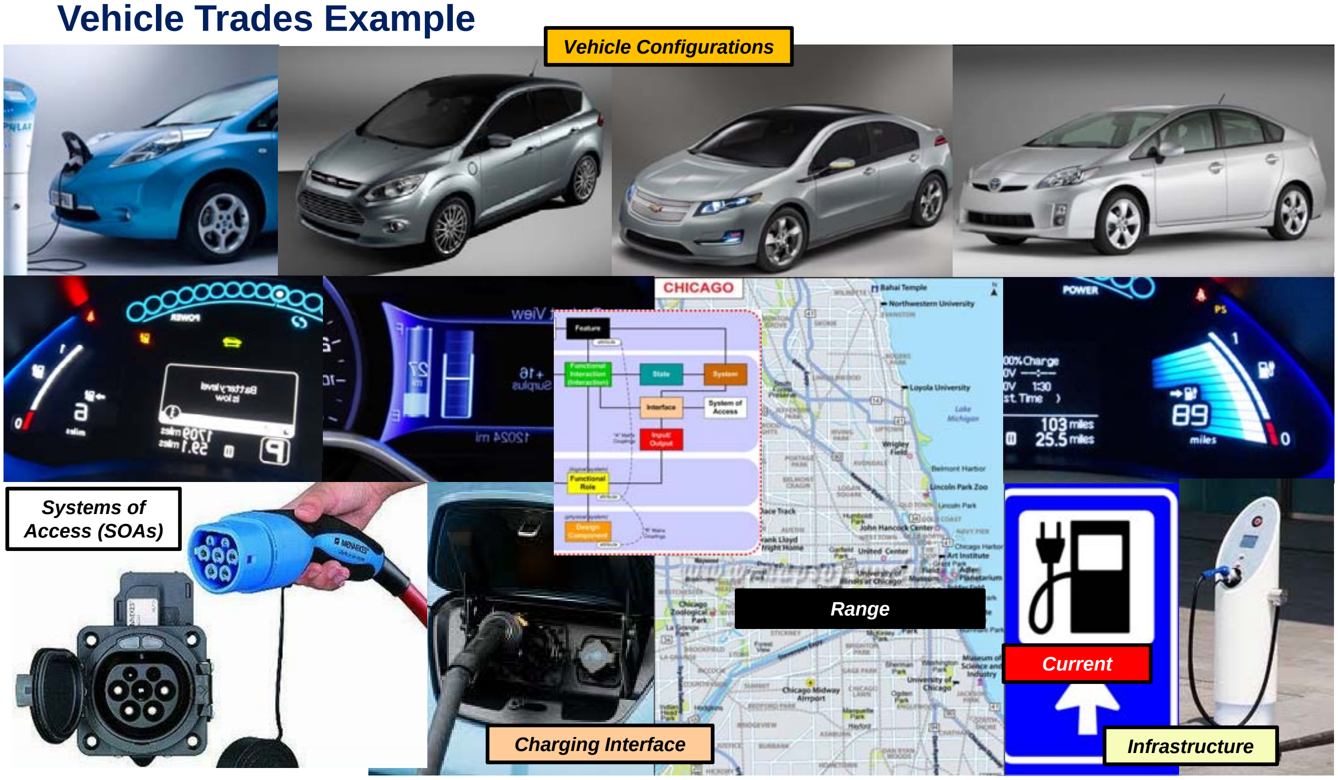




- **Buyer Sample Features:**
  - Sufficient ***range*** to make it to work and back - without going into Flintstone mode
  - Low ***operating costs*** i.e. fuel economy
  - Reasonable ***acceleration*** – 0-60 mph in 2.8 sec.
  - Affordability / purchase price / ***cost***
- **Producer Sample Features:**
  - To develop product lines which meet a broad portfolio of user requirements
  - To meet ambitious fuel economy standards - CAFÉ 54.5 mpg by 2025
  - Provide a return on investment
  - Leverage existing assets and capital structure

3. Selecting Solutions
More Informed Trade-offs

Vehicle Trades Example

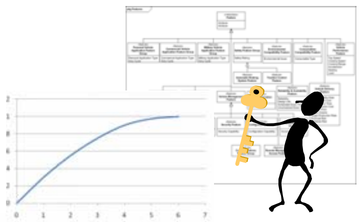


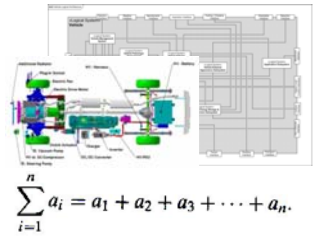
3. Selecting Solutions

More Informed Trade-offs

Vehicle Trades Example

- Using patterns a table of multiple configurations is easily created
- The table enables many different configurations to be easily compared
- Provides the ability to generate many repeatable views and models of value, gaps, utility, sensitivity etc.





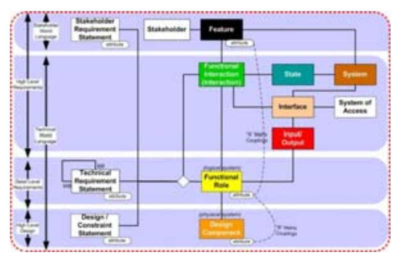
Vehicle		Feature				Functional Role				Design Component			
Configuration	Variant	Range (miles)	Purchase Price ($)	Operating Costs (mpg)	Acceleration 0-60 mph (sec)	Weight	Fuel Tank Capacity (gal)	Battery Full Charge Range	Battery kWh	Fuel Tank	Battery	IC Engine	Regen. Braking Sys.
Vehicle 1	Hybrid Plug In	640	$ 38,712	62	8.9	3781	12	35	16.5	PN# 1	Batty PN#1	I4	Yes
Vehicle 2	Hybrid Plug In	620	$ 32,950	108	8.9	3899	14	20	7.6	PN# 2	Batty PN#1	4 EFF	Yes
Vehicle 3	Hybrid	570	$ 25,200	47	9.4	2906	13.5	10	1.4	PN# 3	Batty PN#2 Batty PN#3	I4	Yes
Vehicle 4	Hybrid Plug In	540	$ 33,000	95	10.2	3165	10.6	11	4.4	PN# 4	Batty PN#4 Batty PN#5	I4	Yes
Vehicle 5	IC Engine Enhanced	496	$ 20,780	40	11.1	2800	12.4	N/A	N/A	PN# 5	N/A	4 EFF	No
Vehicle 6	IC Engine Base	446	$ 16,200	36	7.2	2800	12.4	N/A	N/A	PN# 6	N/A	I4	No
Vehicle 7	Electric Engine	73	$ 28,800	116	7.9	3291	N/A	90-100	24	N/A	Batty PN#5	N/A	Yes

3. Selecting Solutions

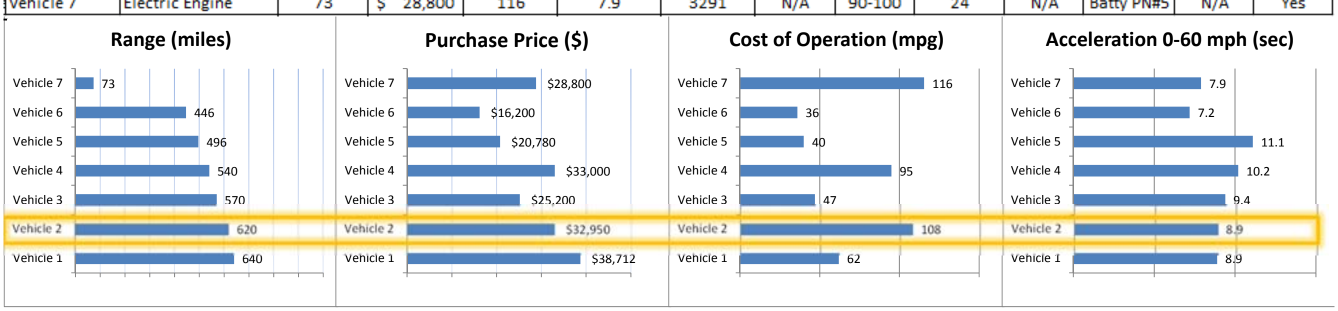
More Informed Trade-offs

Vehicle Trades Example

- Selecting design components populates performance criteria within the logical space and value impact within feature space providing a basis to measure the value of any potential system configuration

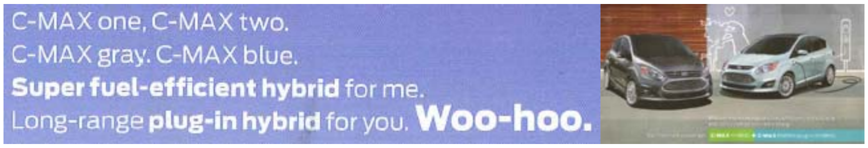


Vehicle		Feature				Functional Role				Design Component			
Configuration	Variant	Range (miles)	Purchase Price ($)	Operating Costs (mpg)	Acceleration 0-60 mph (sec)	Weight	Fuel Tank Capacity (gal)	Battery Full Charge Range	Battery kWh	Fuel Tank	Battery	IC Engine	Regen. Braking Sys.
Vehicle 1	Hybrid Plug In	640	$ 38,712	62	8.9	3781	12	35	16.5	PN# 1	Batty PN#1	I4	Yes
Vehicle 2	Hybrid Plug In	620	$ 32,950	108	8.9	3899	14	20	7.6	PN# 2	Batty PN#1	4 EFF	Yes
Vehicle 3	Hybrid	570	$ 25,200	47	9.4	2906	13.5	10	1.4	PN# 3	Batty PN#2 Batty PN#3	I4	Yes
Vehicle 4	Hybrid Plug In	540	$ 33,000	95	10.2	3165	10.6	11	4.4	PN# 4	Batty PN#4 Batty PN#5	I4	Yes
Vehicle 5	IC Engine Enhanced	496	$ 20,780	40	11.1	2800	12.4	N/A	N/A	PN# 5	N/A	4 EFF	No
Vehicle 6	IC Engine Base	446	$ 16,200	36	7.2	2800	12.4	N/A	N/A	PN# 6	N/A	I4	No
Vehicle 7	Electric Engine	73	$ 28,800	116	7.9	3291	N/A	90-100	24	N/A	Batty PN#5	N/A	Yes



For Fun…

Highlighted in the table



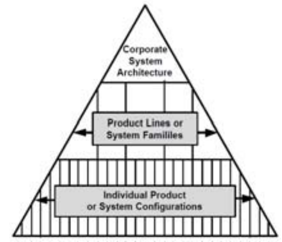
Configuration	Ford C-Max Energi
Variant	Hybrid Plug In
Range (miles)	620
Operating Costs (mpg)	108
Acceleration 0-60 mph (sec)	8.9
Cost (dollars)	$32,950
Top speed (mph)	102

Not in the table

A whole different kind of

Woo-hoo.

Configuration	Porsche 918
Variant	Hybrid Plug In
Range (miles)	952
Operating Costs (mpg)	78
Acceleration 0-60 mph (sec)	2.8
Cost (dollars)	$845,000
Top speed (mph)	202



As wildly different as these two are can you think of pattern aspects they share?

3. Selecting Solutions

More Informed Trade-offs

Summary / Benefits

- Patterns provide a rapid way to investigate configuration options and the impact of subsystem selections on stakeholder value impact
- Patterns provide an established and well documented knowledge base for making decisions
- Patterns translate discrete design component selections into system level value impact through attribute couplings
- Provides a way to develop heuristics, design rules and platform strategies

If you drive 20 miles or less a day, the Energi plug-in version is for you. It costs more, but you'd probably go to the dentist more often than the gas station.

If your daily driving much exceeds 30 miles, the regular hybrid is the better choice. You'll save about two grand and you'll still get 40-plus mpg, which is stellar.

Dan Neil, The Wall Street Journal
May 31, 2013

4. Design for Change

Improving System Resiliency

Concept: System Resiliency/ Platform Evolution

Challenge:

To design and build systems which overcome constraints and vulnerabilities of the global supply chain, *rapidly changing* user needs, and an *uncertain operational future*[1].

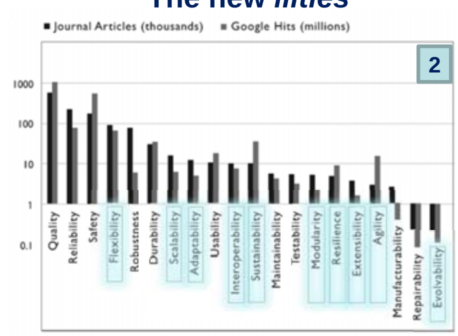
Goal:

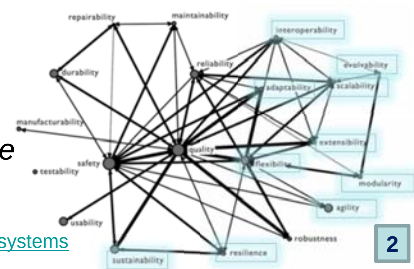
Significantly *transform traditional engineering* practices to develop and adapt systems to *address dynamic needs* and risks[1].

Assertions:

- *Clean sheet design is extremely rare*
- *Rapid change is normative, keeping pace is required*
- *Systems often require lifecycle extension i.e. upgrades*
- *System resilience provides significant competitive advantage*

The new *ilities*





1. *DoD Engineering Resilient Systems* http://www.acq.osd.mil/chieftechnologist/areas/ers.html
2. *Engineering Systems*: de Weck, Ross and Magee, 2011 - http://mitpress.mit.edu/books/engineering-systems

4. Design for Change

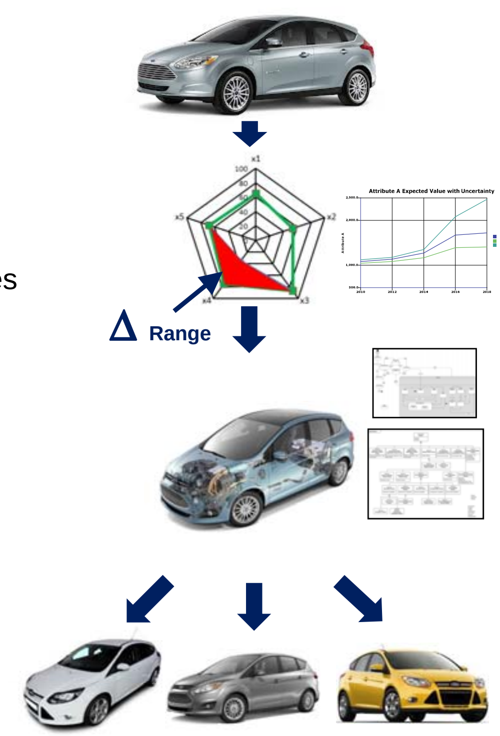
Improving System Resiliency

Uncertainty Management:

- Understanding how requirements might change
- Eliminating the physical cause of the uncertainty
- Delaying design decisions until uncertain variables are known

Architecture Management:

- Reducing the system sensitivity to uncertainties
- Purposefully isolating anticipated change
- Planning for subsystem and technology insertion
- Leveraging platform engineering methodologies



We can't solve problems by using the same kind of thinking we used when we created them. -- Albert Einstein --

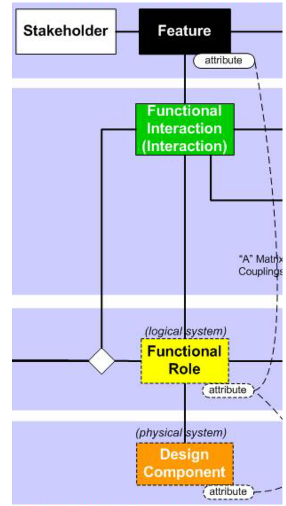
4. Design for Change
Improving System Resiliency

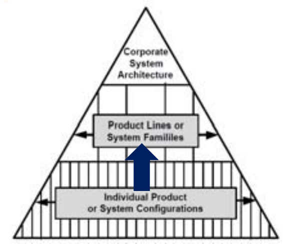
Uncertainty Management:

- Should be viewed across all Stakeholders
- Is performed in Feature space
- Assigns value and measures to new ilities
- Must go beyond best guess or average estimates

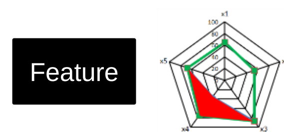
Architecture Management:

- Extends beyond the end product alone – flexible manufacturing etc.
- Is performed in functional and physical space
- Accommodates new *ilities* within product lines/families to improve leverage. ***Move up*** resilient design principles where appropriate





4. Design for Change
Uncertainty Management



Uncertainty Management Includes:

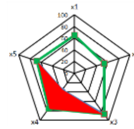
- Clarifying Issues
  - Envisioning alternate futures for operational context, mission, technologies etc.
  - Identifying key issues and categorizing them as Criteria, Chances, Choices & Constituencies
  - *Clarifying Issues Tools: War gaming, Brainstorming, Delphi, Affinity Diagrams…*
- Describing the potential uncertainties, decisions and criteria
  - Assessing probability of occurrence and how that probability changes over time
  - Understanding how uncertainties may be driven by more fundamental ones
  - For each criteria perform Five Whys to infer the primary criteria/needs
  - *Identifying Uncertainties Tools: SME and Stakeholder Interviews, Five Whys, Root Cause Analysis…*
- Identifying the contextual drivers of potential change
  - Define a deterministic multi-objective measure of performance
  - Relate multi-objective measure to the uncertainties and decisions (Influence Diagrams)
  - Analyze the end-point uncertainties of the influence diagram to determine which uncertainties, when varied over their range, cause the greatest change in value
  - *Identifying Drivers Tools: Influence Diagrams, Sensitivity Analysis, DOEs, Pareto Charting…*

For all of its uncertainty, we cannot flee the future. - Barbara Jordan

4. Design for Change

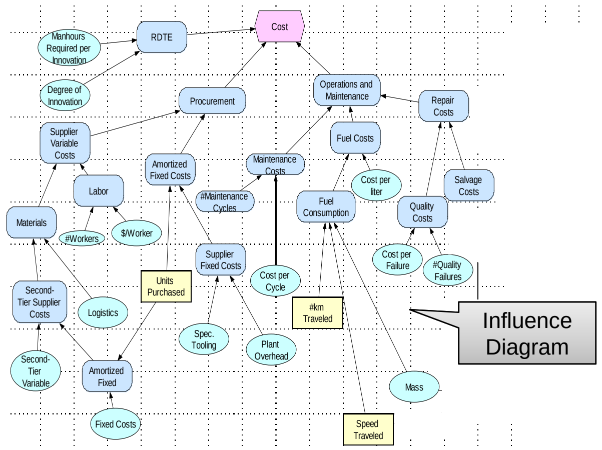
Uncertainty Management

Feature



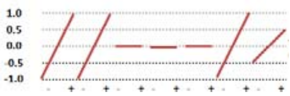
Influence Diagrams

- The adjacent example models cost as the relevant criteria
- Great tool for identifying potential drivers of change in complex systems
- Sensitivity - With this model we can conduct a sensitivity analysis, via a DOE, to identify the impact and interaction effects
- This DOE also allows for the estimation of Criticality - Use a tornado chart (two-sided vertical Pareto chart) to identify the most critical uncertainties



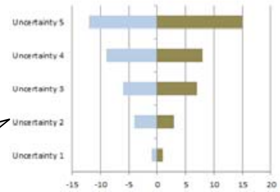
Influence Diagram

Symbol	Element	Description
What do we do?	Decision	A variable that can be modified directly
What's the outcome?	Chance Variable	A value which cannot be controlled directly, is uncertain
What's the situation?	General Variable	A deterministic fuction of the quantities is depends on
How do we like it?	Objective	A measure of satisfaction with an outcome, utility
→	Arrow	An influence



Design Of Experiments

Tornado Chart

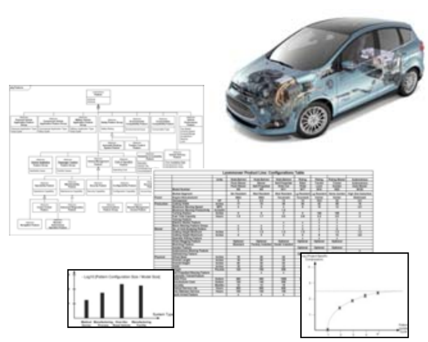


4. Design for Change

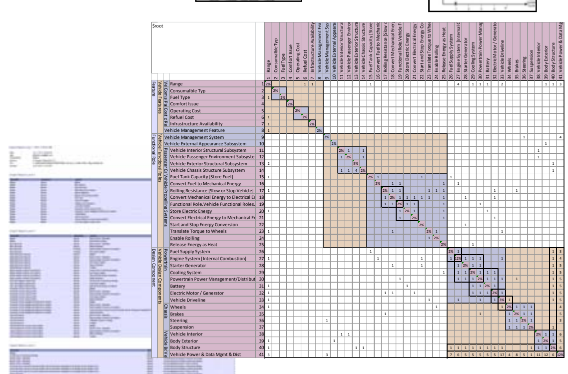
Architecture Management

Architecture Management Includes

- Informing system designers through analysis
  - Provide rigor around how system elements interact – pattern contains this key information
  - Understanding how system elements and interactions are affected by change
  - Modifying architectures to decrease sensitivity to change



- Architectural analysis of:
  - Modularity & System Partitioning
  - Accommodating New Technology
  - Change Propagation and Impact

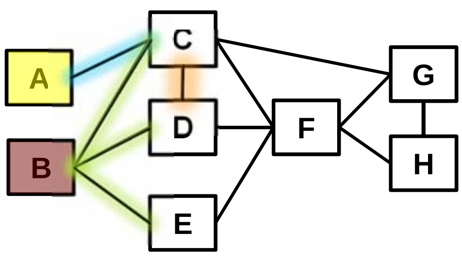


Curiosity begins as an act of tearing to pieces or analysis. - Samuel Alexander

Graph Theory & Design Structure Matrix
Systems Analysis

Powerful methods to analyze architectures

- The diagrams below provide two different views of a generic system with interrelationships as shown
- These interrelationships could be physical, informational, energy transfer or material/mass exchange
- Such diagrams are necessary to gain a better understanding of how systems elements interact



Network Graph
Lines indicate connectivity between elements

	A	B	C	D	E	F	G	H
A			X					
B			X	X	X			
C	X	X		X		X	X	
D		X	X			X		
E		X				X		
F			X	X	X		X	X
G			X			X		X
H							X	

Matrix View
X's indicate connectivity between elements

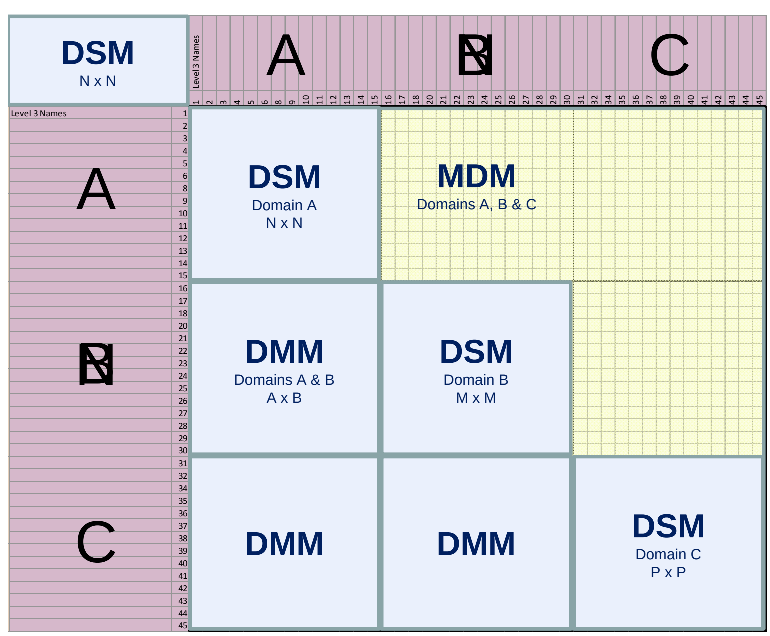
The benefit of the matrix is that it provides a compact visual of the system and it enables holistic systems modeling, analysis and optimization

Design Structure Matrix Overview

Design Structure Matrix (DSM)
- Square matrix- N x N or N^2
- Analyze dependencies within a domain
- Used for products, process and Organizations
- Binary marks "(1" or "X") show existence of a relation
- Numerical entries are weights of relation strength
- Can be directed or undirected (symmetrical)

Multi Domain Matrix (MDM)
- Square matrix - N x N or N^2
- Analyze dependencies across domain
- Combination of DSMs and DMMs
- Especially helpful for DSMs > 1000 elements

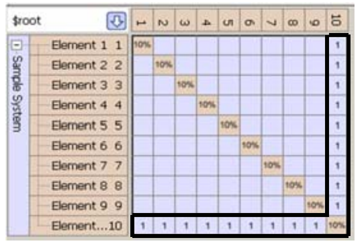
Domain Mapping Matrix (DMM)
- Normally rectangular matrix – N x M
- Mapping between two domains



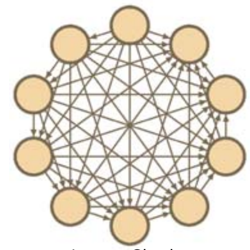
Example Network Graphs and DSM Patterns

Understanding Architecture, Dependency and Related Patterns

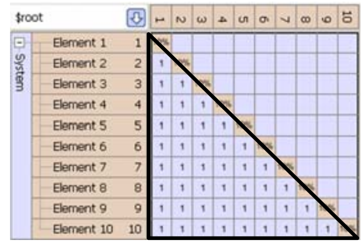
Layout: Concentric



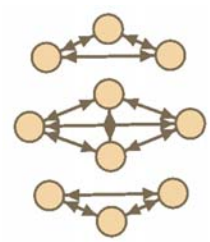
- Symmetrical
- Layered System – every systems uses every system below it



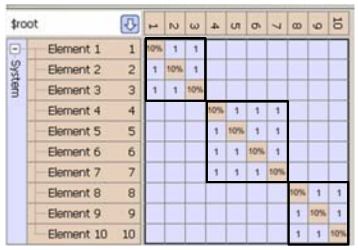
Layout: Circular



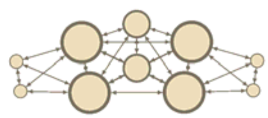
- Non symmetrical
- Layered System – every systems uses every system below it



Layout: ForceAtlas2



- Symmetrical
- Non-Overlapping clusters



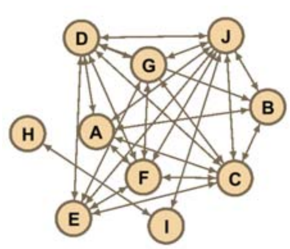
Layout: Yifan Hu

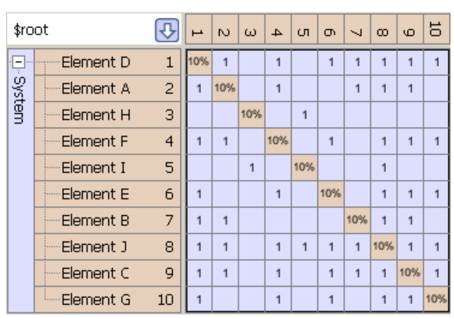
- Symmetrical
- Overlapping clusters

Example Network and DSM Patterns

Understanding Architecture, Dependency and Related Patterns

Unorganized





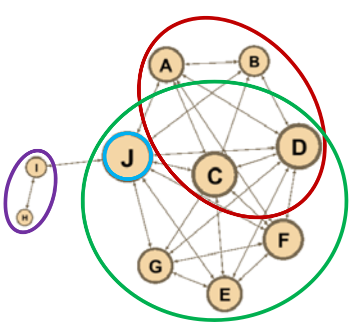
Network Graph

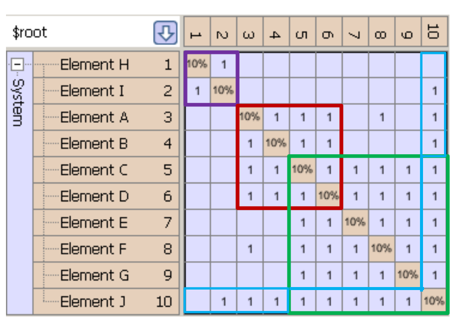
- Randomly generated

DSM

- Randomly ordered

Organized





Network Graph

- Nodes sized by degree
- Arranged by cluster

DSM

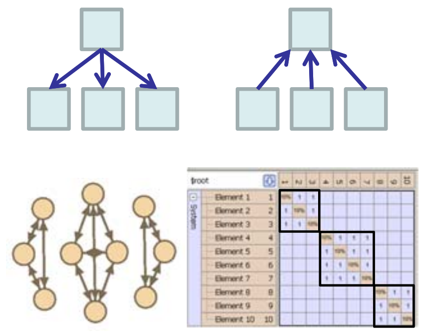
- Layered
- Change propagator, Element 10, clearly shown at the bottom
- Clustered, showing both overlapping non-overlapping and clusters

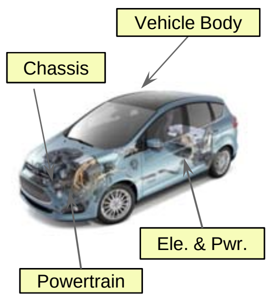
4. Design for Change

Architecture Management

Modularization & System Partitioning

- Modularization is the grouping of system elements that are mutually exclusive or minimally interacting subsets (absorb interactions internally).
- It eliminates redundancy, minimizes external connections
- It minimizes change propagation, enables technology insertion and platform based engineering methods making systems less sensitive to the uncertainties





Design Component	Vehicle Design Components				26	27	28	29	33	34	35	36	37	38	39	40	41
		Powertrain	Fuel Tank	26		1										1	3
			IC Engine System	27	1		1	1	1							1	4
			Starter Generator	28		1		1								1	5
			Electric Drive	29		1	1		1		1						
			Vehicle Driveline	33		1		1		1						1	5
		Chassis	Wheels	34					1		1	1	1				4
			Brakes	35				1		1		1	1				5
			Steering	36						1	1		1				3
			Suspension	37						1	1	1				1	
		Vehicle Bc	Vehicle Interior	38											1	1	6
			Body Exterior	39										1		1	5
			Body Structure	40	1	1	1		1				1	1	1		6
		Vel	Vehicle Power & Data Mgmt & Dist	41	7	6	5		17	4	8	5	1	11	12	6	

Power-train

Power/ Electrical

Chassis

Vehicle Body

4. Design for Change

Architecture Management

Accommodating New Technologies / Subsystems

- Patterns enable in depth analysis of design component selection
- Combining system and subsystem matrixes permits:
  - Analysis of subsystem and technology integration complexity and risk
  - Identification of potential cost drivers
  - Further pattern recognition, development and refinement

Identify which technology elements affect multiple system level elements

Element Number		1	3	2	5	4	6	8	7	9	10	12	13	11	14	15	16	17	18	19	20	21	22	23	24	25
Body - Exterior	1		1	1											3	3										
Body - Structure	3	1		1					1										1	1	1	1	1	1	1	1
Body - Interior	2	1	1								1				3	1										
Powertrain - Powertrain Control Module	5					5	5								5	1			5							5
Powertrain - Transmission	4				5		3	1	1						5	3										
Powertrain - Engine	6				5	3			1			1	1		5	3										
Chassis - Driveline	8					1			1	1	1			1	3	1	1	1								
Chassis - Frame	7		1			1	1	1		1	1	1	1	1												
Chassis - Suspension	9							1	1																	
Chassis - Steering	10			1				1	1						3	1									1	
Chassis - Fuel Supply System	12						1		1						3	1										
Chassis - Exhaust System	13						1		1						3	1										
Chassis - Brakes	11							1	1						3	1			3	1	1	1	1			
Electrical - Data System	14	3		3	5	5	5	3			3	3	1	3		3		3	5					5	5	5
...ical - Power Distribution	15	3		1	1	3	3	1			1	1	1	1	3			1	1	1	1	1	1	1	1	1
Chassis ...kes - Exciter	16							1										1								
Chassis - Brakes - Speed Sens...	17							1							3	1	1		1							3
Chassis - Brakes - ABS Control Module	18		1		5									3	5	1		1		3	3	3	3			5
Chassis - Brakes - ABS Pump	19		1											1		1			3							3
Chassis - Brakes - ABS Modulator Valves	20		1											1		1			3			1	1			3
Traction Control Solenoid Valve	21		1											1		1			3		1		1			
Modulator Valves	22		1											1		1			3		1	1				3
Acceleration Sensor (Yaw,R,L)	23		1												5	1										5
Steering Angle/Position Sensor	24		1								1				5	1										5
Electronic Controller Module & Data Bus	25		1		5										5	1		3	5	3	3	3	3	5	5	

Identify high impact areas to a particular system element

Assess multiple technologies to determine Technology Invasiveness (*Technology Infusion – Oli de Weck*)

4. Design for Change

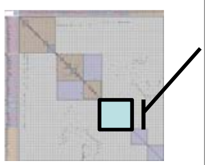
Architecture Management

Change Propagation

- Realized uncertainties often drive engineering changes which can easily balloon in an uncontrolled fashion
- Knowing how changes propagate so 2nd, 3rd, and 4th order impacts are known is very powerful
- Early discovery of "propagation paths" can have a significant impact on total life cycle cost.[1]
- Architectural analysis and understanding of system patterns helps control change propagation

Multipliers	Generate more changes than they absorb
Carriers	Absorb a similar number of changes to those they cause
Absorbers	Absorb more change they themselves cause
Constants	Unaffected by change 1

1. Eckert C, (2004) Change and Customization in Complex Engineering Domains, Research in Eng. Design



Design Component / Vehicle Design Components	Component		26	27	28	29	33	34	35	36	37	38	39	40	41	# of Elements	# Dependencies
Powertrain	Fuel Tank	26		1										1	3	3	5
Powertrain	IC Engine System	27	1		1	1	1							1	4	6	11
Powertrain	Starter Generator	28		1		1								1	5	4	10
Powertrain	Electric Drive	29		1	1		1		1							4	8
Powertrain	Vehicle Driveline	33		1		1		1						1	5	5	11
Chassis	Wheels	34					1		1	1	1				4	5	8
Chassis	Brakes	35				1		1		1	1				5	5	10
Chassis	Steering	36						1	1		1				3	4	6
Chassis	Suspension	37						1	1	1				1		4	4
Vehicle Bc	Vehicle Interior	38											1	1	6	3	8
Vehicle Bc	Body Exterior	39										1		1	5	3	7
Vehicle Bc	Body Structure	40	1	1	1		1				1	1	1		6	8	17
Vel	Vehicle Power & Data Mgmt & Dist	41	7	6	5		17	4	8	5	1	11	12	6		11	102

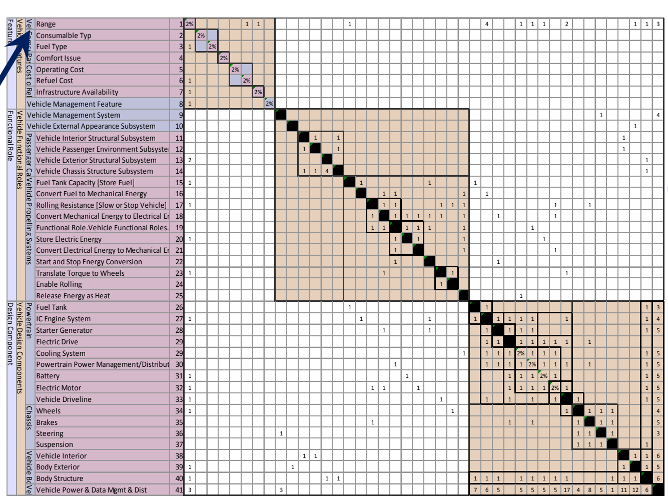
All change is not growth, as all movement is not forward. - Ellen Glasgow

4. Design for Change

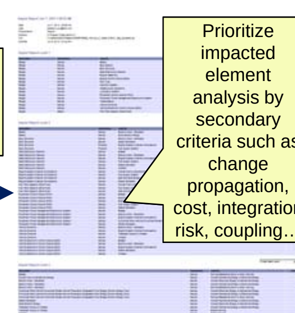
Architecture Management

Impact Analysis

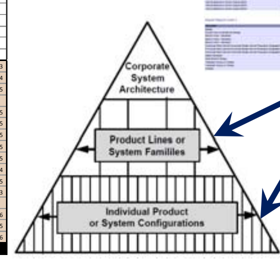
- ***Product Line/System Families/Platforms***: The common system pattern which enable rapid specialization or configuration of individual products / systems configurations i.e. product variants. Change impact analysis can aid in determining which elements remain a part of the family pattern, which are unique and which should become flexible.



Generate impact report of realized / modeled uncertainties



Prioritize impacted element analysis by secondary criteria such as change propagation, cost, integration risk, coupling…

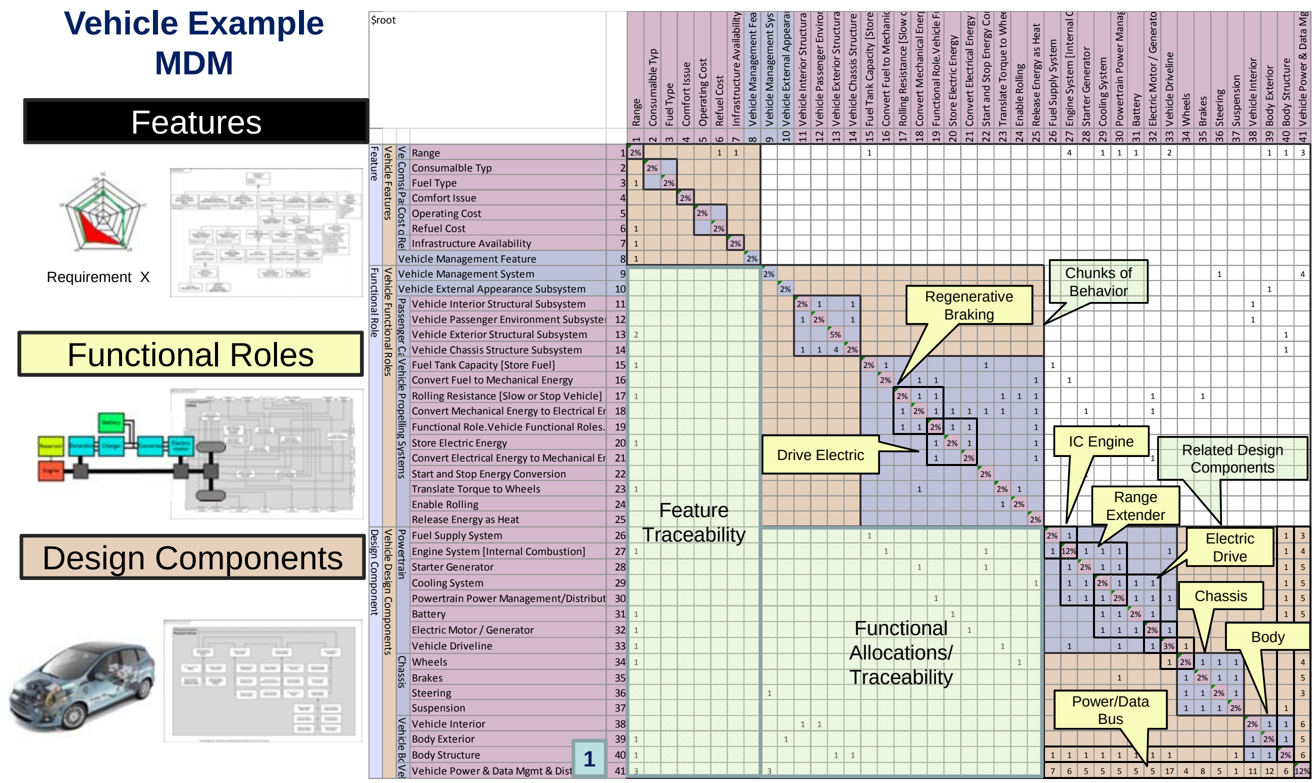


Address uncertainty as high up in the pattern as possible to leverage across the portfolio

1. deWeck, Oli, Strategic Engineering: Designing Systems for an Uncertain Future, Flexible Product Platforms: Framework and Case Study
2. Kalligeros K., de Weck O., de Neufville R., Luckins A., "Platform Identification using Design Structure Matrices", *Sixteenth Annual International Symposium of the International Council On Systems Engineering (INCOSE)*, Orlando, Florida, 8 - 14 July 2006

4. Design for Change

Architecture Management

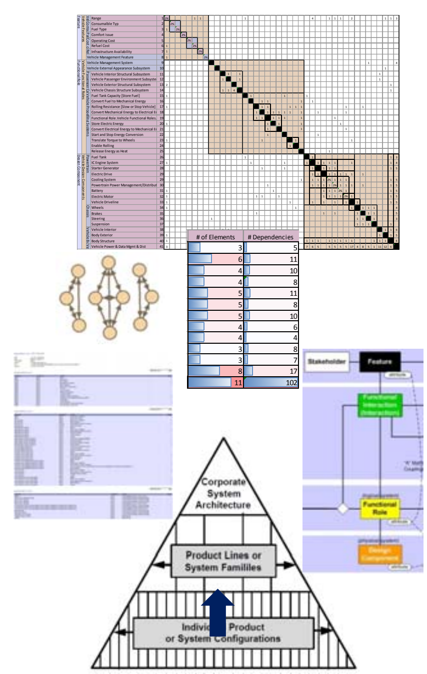


4. Design for Change

Improving System Resiliency

Designing for Change Benefits:

- Provide a means to accommodate rapidly changing needs
- Measure change impact and improve pattern management evolution and leverage
- Improve new ility system characteristics
- Supports platform methods reducing total life cycle cost
- Avoids the Flaw of Averages
  - Assuming that evaluation of accommodating an uncertainty based upon average conditions gives a correct result[1].



1. *Flexibility in Engineering Design*: de Neufville and Scholtes, 2011 - http://mitpress.mit.edu/books/flexibility-engineering-design

5. Using Patterns to Improve Risk Analysis: Example

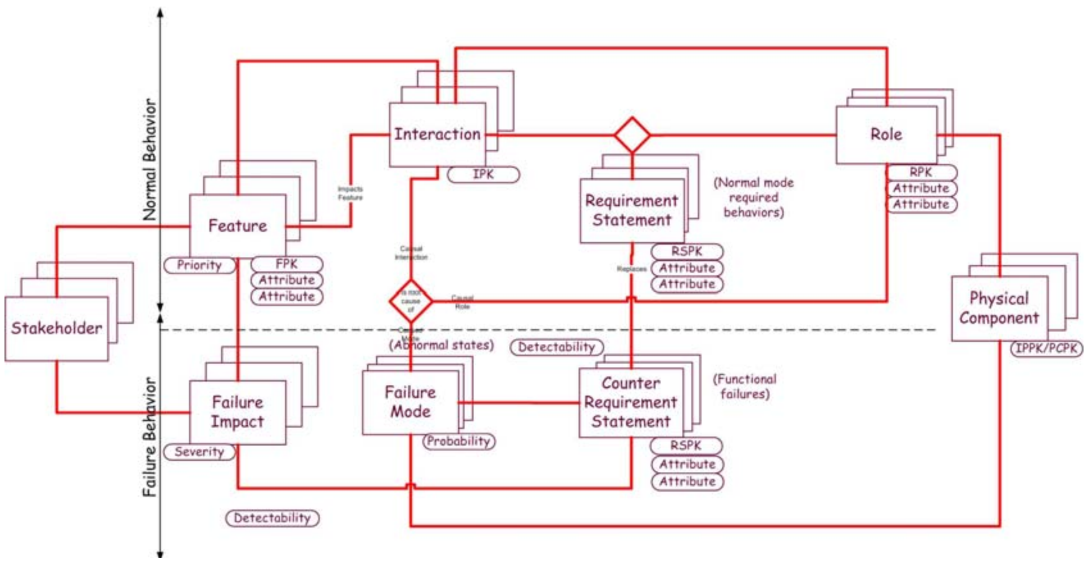
- Concept: A System Pattern can be used to generate more complete risk analyses, and with less effort;
- Because the Feature Pattern by intention represents the stakeholder level concerns of all classes of stakeholders:
  - Features are the only things that can possibly be at risk!
- For example, in an FMEA, the only possible “Effects” at risk are the system Features:
  - The System Pattern can provide a pre-stored library of Impacts of non-delivery / non-performance of each Feature, even before a design exists.
- Similarly, analysis and management of Project Risks, Technology Risks, doing a Preliminary Hazard Analysis (PHA), Fault Tree Analysis, integrating Technology Readiness Levels (TRLs), or other forms of risk analysis can all be viewed through the integrated lens of Stakeholder Features
- This has a nice integration effect—for example, project “top level” risk reports or views can be expressed in the form of master risk views . . . .

5. Using Patterns to Improve Risk Analysis: Example

Physical Entity	Failure Mode
Vehicle ECM	Dead ECM
Vehicle ECM	Network Connector Open
Vehicle ECM	Network Connector Short
Vehicle ECM	Erratic ECM
Battery	Discharged Battery
Battery	Battery Cell Short
Battery	Battery Cell Open
Battery	Battery Leak
Panel Display	Fractured Display
Panel Display	Illuminator Fail
Bluetooth Module	Module Hard Fail
Bluetooth Module	Transmitter Fail
Bluetooth Module	Receiver Fail

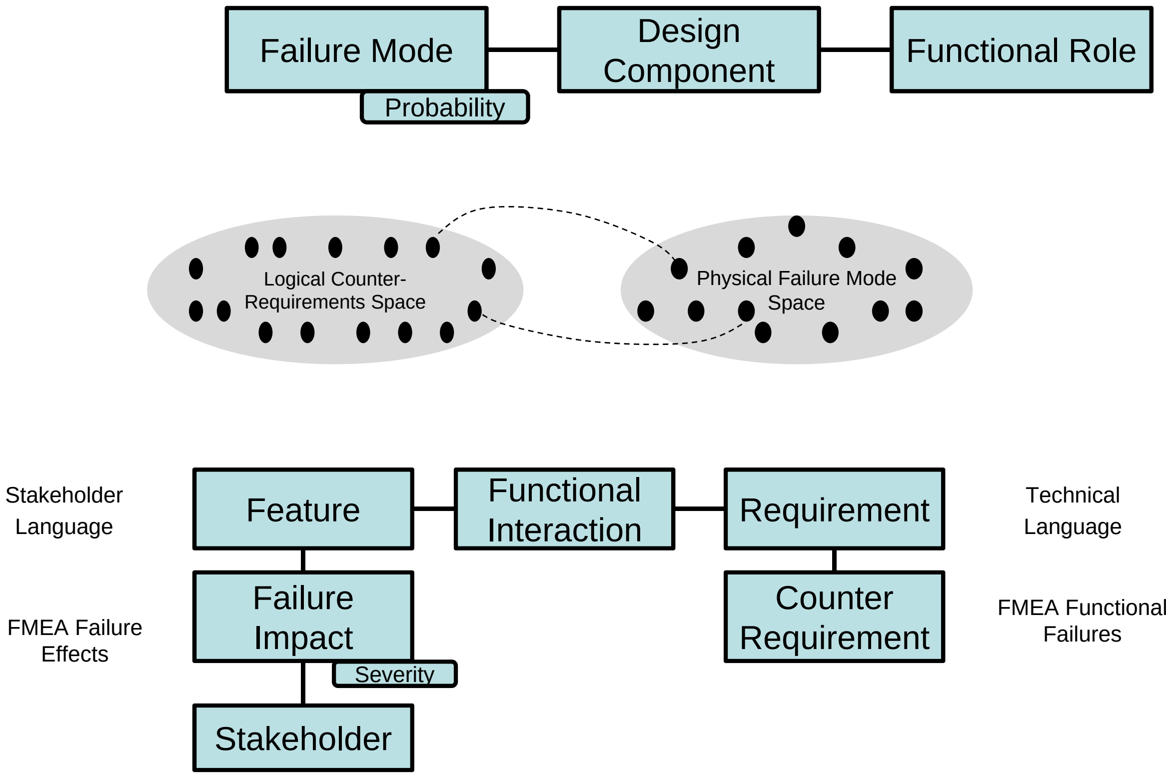
Using Patterns to Improve Risk Analysis: Failure Modes

- The pattern is used to accumulate experience in the following Risk Model areas:
  - Feature Impacts: The stakeholder impact of non-delivery of a Feature
  - Counter-Requirements: An (abnormal) behavior violating a System Requirement
  - Failure Mode: A state of an entity in which its behavior includes at least one Counter Requirement

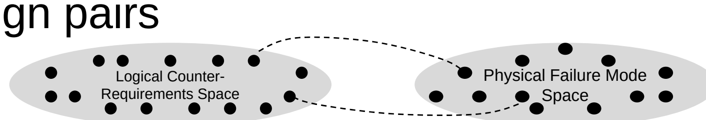


Using Patterns to Improve Risk Analysis: Example

Feature	Effect (Failure Impact)	Severity	Functional Failure (Counter Requirement)	Component	Failure Mode	Probability	Mitigation (Control)
Navigation Feature [GPS-based Location Sensing]	No Confidence in Displayed Position	Serious (4)	The system displays a location that is not accurate to 10 feet.	Vehicle ECM	Erratic ECM	0.0015	Nav Backup Mode: External Nav Module
Navigation Feature [GPS-based Location Sensing]	False Confidence in High Error Displayed Position	Critical (5)	The system displays a location confidence indicator that is not correct.	Vehicle ECM	Erratic ECM	0.0015	None
Navigation Feature [GPS-based Location Sensing]	No Displayed Location	Serious (4)	The system does not display the graphic map presentation.	Panel Display	Fractured Display	0.0003	Nav Backup Mode: External Nav Module

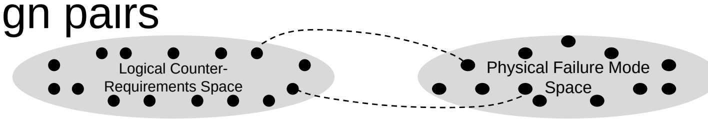


Combinatorial “matching up” of requirements-design pairs



- The Functional Failures (counter requirements) and Failure Effects (feature failure impact) data can be pre-populated independent of the system’s internal design, and the Failure Mode data for standard component roles can be pre-populated independent of the system’s external requirements.
  - So, when both the requirements and a candidate design have become known, how do these two halves of the failure analysis model get connected to each other?
  - This turns out to be a combinatorial algorithm.
- First, it turns out that the counter-requirements (functional failures) obtained by reversing the requirements statements may describe some hypothetical external behaviors that are never (or with probability too small to matter) caused by component failure modes.
  - This will cause some pre-populated functional failures to be dropped.
  - For example, a requirement that a product weigh less than one pound has a counter-requirement that it weighs more than one pound.
  - It may be determined that there is no component failure mode that impacts weight, so that this functional failure is dropped from the list.
  - Notice that even this failure mode could happen for some products—for example, a hazard protection suit that becomes wet weighs more.
- Second, it turns out that some failure modes of a physical component have no consequence on the product’s required behavior, because the failure mode goes with a role not allocated to the part in this particular product design.
  - For example, an integrated circuit may have built-in circuitry for performing certain functions which are not used by a certain product’s design, even though other portions of that chip are used.
- The connection of the requirements half of the failure analysis to the design half of the failure analysis is made by matching up “mating” pairs, and discarding what is left as not applicable (after checking for missed cases this approach also helps us find—another benefit) . . .

Combinatorial "matching up" of requirements-design pairs



- The "matching up" is accomplished through the matching of counter-requirements with failure modes.
  - Each failure mode causes some abnormal behavior.
  - All abnormal behavior is described by counter requirements. When we find a counter-requirement belonging to a failure impact is equal to a counter-requirement for a failure mode, that pair is associated together, completing two major sections of a row in a failure analysis table.
  - Some failure modes may connect to multiple counter requirements and some counter requirements may connect to multiple failure modes.
- This process may use two levels of requirements, in the form of system black box requirements and their decomposed white box requirements (allocated to physical parts), in which case counter-requirements may be developed at both levels.
  - A simpler alternate method is to use only one level of counter-requirements, with the component failure modes associated directly with the resulting abnormal behavior at the black box level—in which case the association of failure modes with abnormal behavior is dependent upon knowing the system level design.
  - Likewise, the states discussed above may be at two levels, representing states (and failure modes) of system components and the whole system, or simplified to states of the whole system, in which case the failure modes are modes of the whole system and again dependent upon its design.
- The discussion above assumes failure modes originate in <u>internal</u> system components, typical of analyses such as a Design FMEA (D-FMEA).
  - Also discussed later below are failure modes of external people or processes (actors) that impact upon the subject system, as seen in an Application FMEA (A-FMEA) or a Process FMEA (P-FMEA).
  - The counter-requirements and physical mode matching-up approach is substantially the same in these cases.

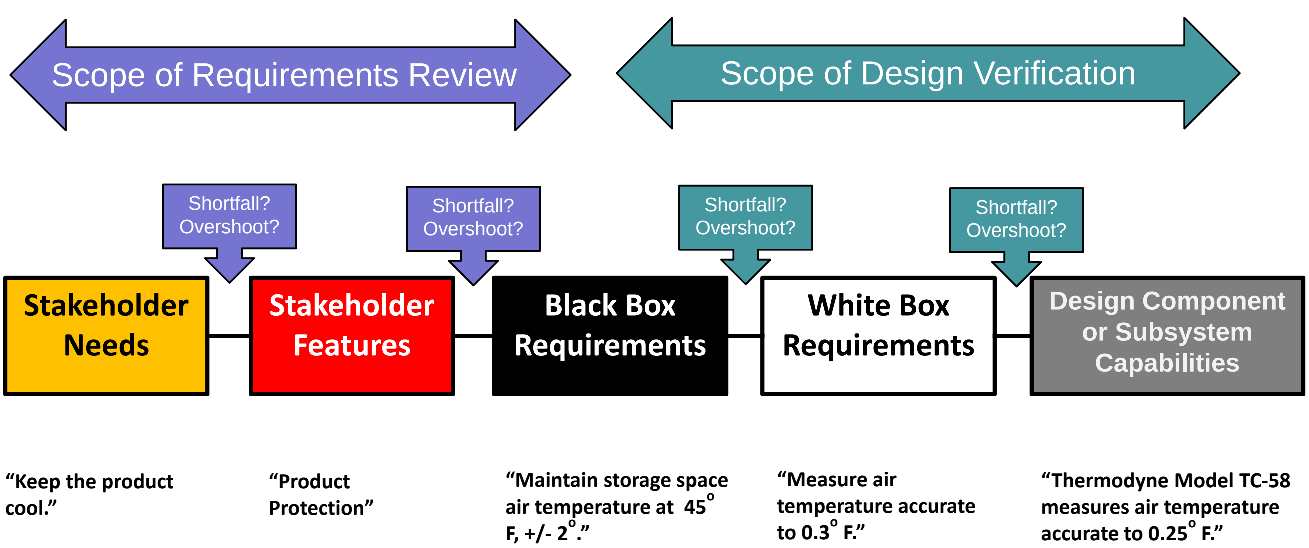
5. Using Patterns to Improve Risk Analysis: Example

- Benefits:
  - Generate initial FMEA or other risk analyses with less initial effort;
  - More complete—reduces omissions;
  - Feels more systematic than the usual FMEA process;
  - Generates the "normal" FMEA view
  - Easier to generate from pattern;
  - Stages—without failure modes versus with failure modes
  - The Pattern provides a clear place to accumulate new learning (e.g., additional Requirements);
- No free lunch:
  - Analysis should still pass through normal SME review—this is just a way to generate the first draft faster and in more complete form;
  - Incomplete models of features, requirements, or failure modes means incomplete failure risk analysis.

6. Using Patterns to Improve Verification

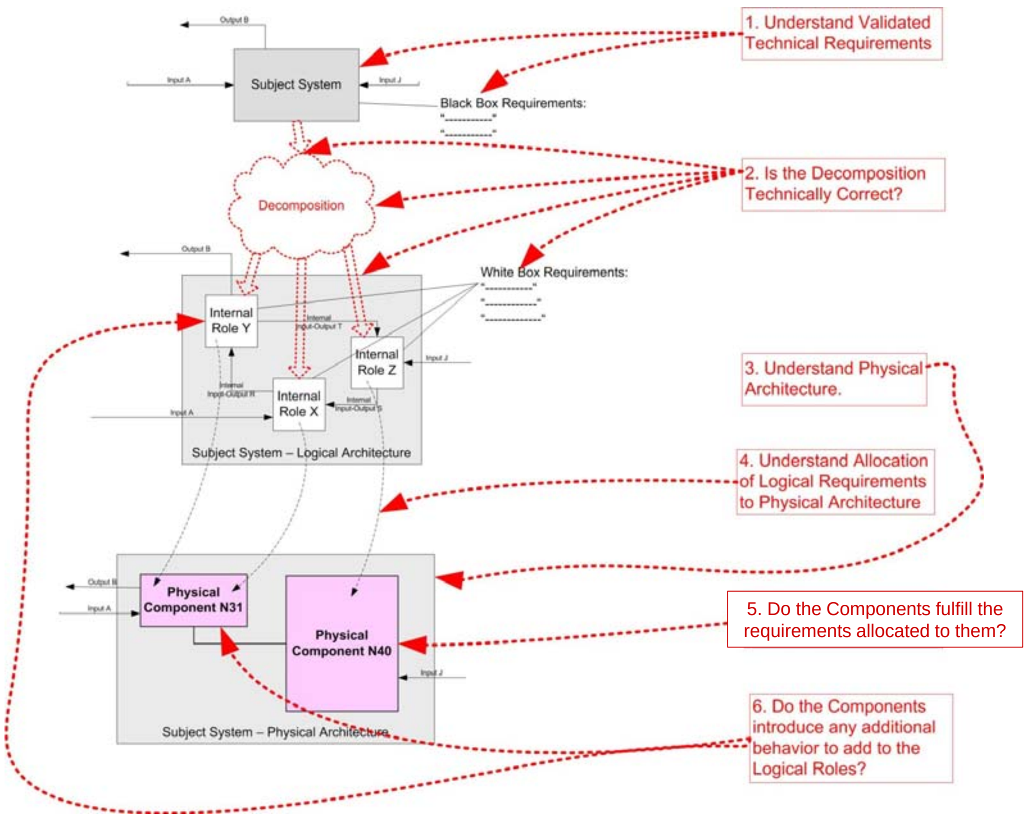
- Concept: Patterns help generate better Verification Plans faster—including plans for Design Review, Simulation, System Test, etc.
- Verification is concerned with confirming that a candidate design will meet requirements;
- In some domains (medicine, flight, etc.), verification represents a high fraction of large costs and time investment—patterns can help reduce this;
- Patterns represent: Requirements, Design, and connecting relationships—including the degree of their consistency with each other, as well as the means of verifying it.

There are a <u>limited number of types</u> of potential misalignments to check and close



(All these misalignments are *<u>ultimately</u>* measured in terms of their *<u>impact on Features</u>*.)

Six questions for Design Review:



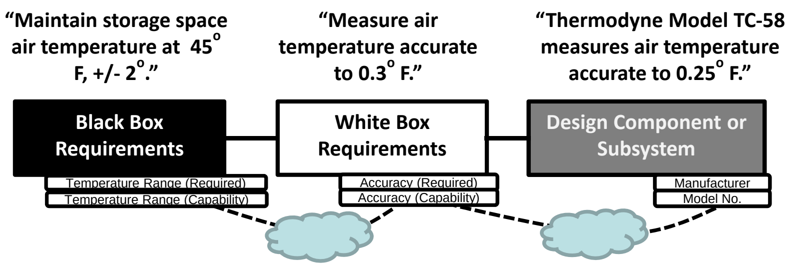
6. Using Patterns to Improve Verification: An Example

- Using the System Pattern, configuring its Features not only configures the Requirements, it also populates the Verification Approach (plan):

	Features	Interaction	Interaction PK Value	Functional Role	Req ID	Requirement	Test/Review Approach
68	Vehicle Performance Feature[]	Travel Over Terrain	Accel 0-60	Vehicle	VEH-1054	The system shall be capable of accelerating to 60 MPH from a standing start to 8 seconds, on level dry pavement.	Test Track 2, Certified Test Driver, Max Acceleration, Average of Four Runs
69	Vehicle Performance Feature[]	Travel Over Terrain	Braking Distance	Vehicle	VEH-1055	The system shall be capable of decelerating from 60 MPH to a stop in 150 feet, on level dry pavement.	Test Track 1, Certified Test Driver, Max Over-Braking Maximum of Five Runs
70	Vehicle Performance Feature[]	Travel Over Terrain	Cruising Range	Vehicle	VEH-1056	The system shall be capable of traveling 300 miles on one full tank of fuel, on level terrain in nominal conditions.	Test Oval 1, Certified Test Driver, 60 MPH, Cruise Control Engaged, Average 10 Runs on Standard Summer Conditions.
71	Vehicle Performance Feature[]	Travel Over Terrain	Cruising Speed	Vehicle	VEH-1057	The system shall be capable of extended travel at 55 MPH over nominal terrain.	Terrain Course 3, Certified Test Driver, Cruise Control Engaged, Post Run Operator Interview, Vehicle Inspection
72	Vehicle Performance Feature[]	Travel Over Terrain	Load	Vehicle	VEH-1058	The system shall be capable of carrying 800 pounds of load in addition to a full load of passengers.	Standard 800 pound load in Transport Bed, four 180 pound average passengers, 60 mile course, Certified Test Driver, Post Run Passengers Interview, Vehicle Inspection
73	Vehicle Performance Feature[]	Travel Over Terrain	Seating	Vehicle	VEH-1059	The system shall be capable of transporting 4 people in seated positions.	Four 180 pound average passengers, passenger heights between 5'8" and 6'4", 60 mile course, Certified Test Driver, Post Run Passengers Interview
74	Vehicle Performance Feature[]	Travel Over Terrain	Top Speed	Vehicle	VEH-1060	The system shall be capable of reaching a stop speed of 85 MPH on smooth level terrain.	Test Track 2, Certified Test Driver, Average of Four Runs, Laser Speed Instrumentation

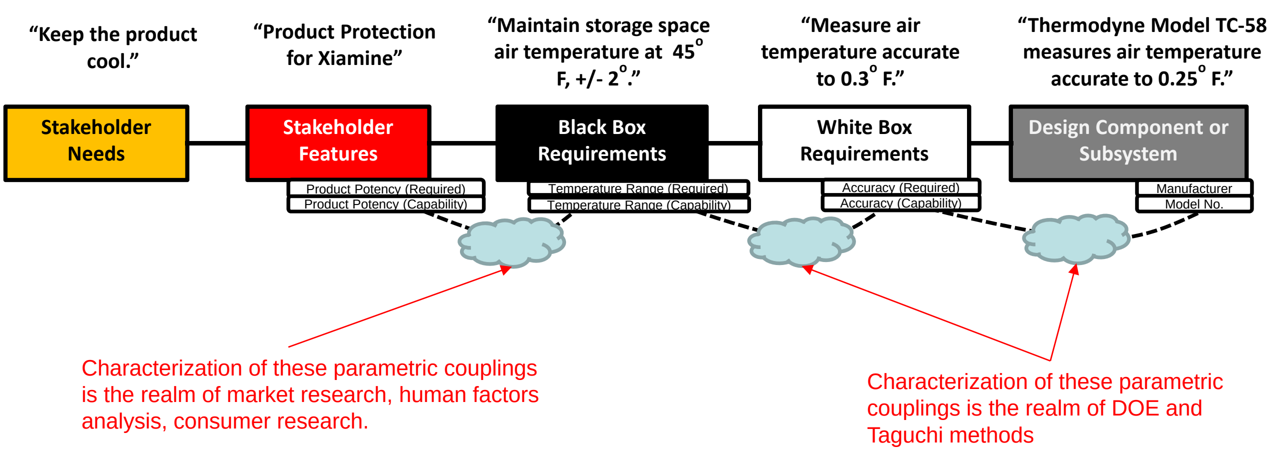
6. Using Patterns to Improve Verification: An Example

- Configuring both the Requirements, as well as the High Level Design, also configures the Decomposition and related Verification:



6. Using Patterns to Improve Verification

- "Test" includes not just functional testing, but also characterization testing, such as planned in the methods of DOE and Taguchi:



6. Using Patterns to Improve Verification

- Benefits:
  - Accumulation of good test methods reduces re-invention of the testing “wheel”.
  - Accumulation of known design review trace information reduces effort to generate paper design review analysis.
  - The Pattern provides a place to accumulate this learning.
- No Free Lunch:
  - Just because we are re-using these assets does not mean we don’t have to think.
  - For example, we need to assure ourselves that previous test methods and design review decompositions really do apply in the next case at hand.

Challenges and Opportunities

Philadelphia, PA
June 24-27, 2013

1. Human hurdles: Inventing from scratch, expertise

2. Organizational hurdles: Better business models are nevertheless unfamiliar

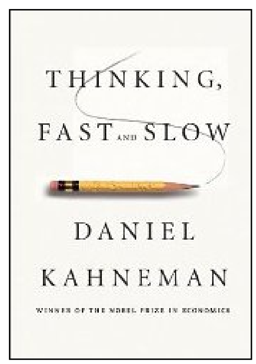
Exercise / group discussion: Approaches to my situation

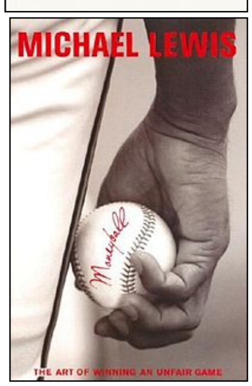
Human hurdles

- Engineers and other designers enjoy creating things—sometimes even if the thing has been created before:
  - This may lead to re-traveling paths, sometimes re-discovering things the hard way (e.g., overlooking requirements, using over-simplifications, etc.)
  - In any case, it can expend time and effort in re-generating, re-validating, and re-verifying what others had already done.
- In other cases, human subject matter experts provide great expertise:
  - but it is accessible only in the form of the presence of the SME, and after accumulating years of experience.
  - Seemingly more a craft of journeymen experts than a discipline based upon teachable principles.
- All these challenges can be viewed as resistance to expressing and applying explicit patterns.

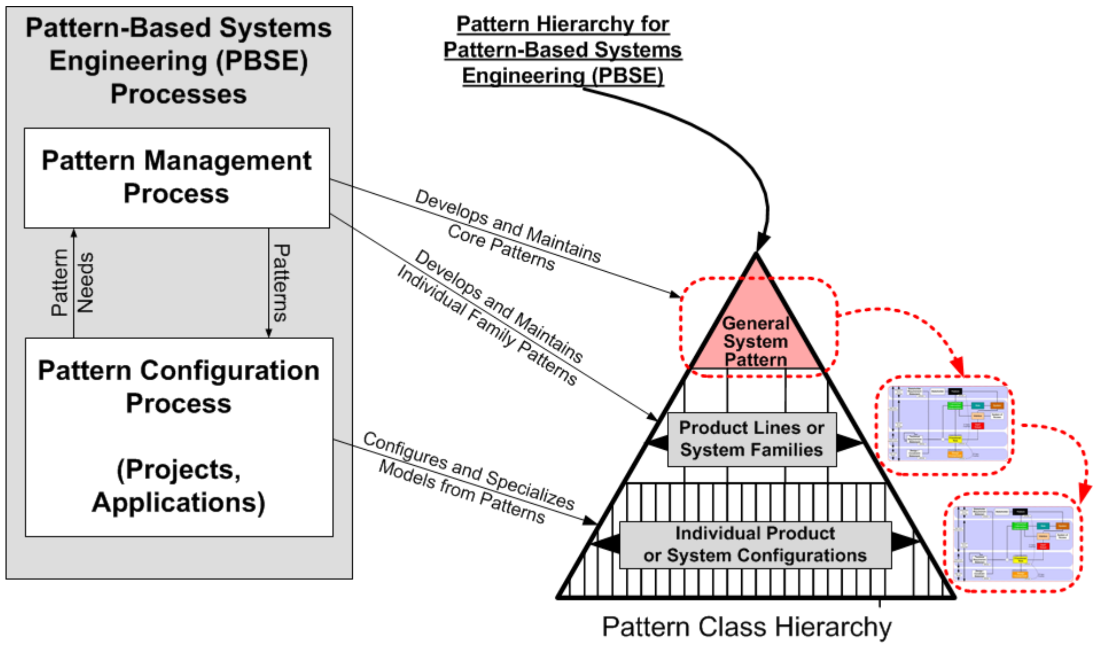
Human hurdles

- A broad issue across human life:
  - The science of irrationality
  - Daniel Kahneman, Nobel Laureate, “Thinking, Fast and Slow”)
  - “Moneyball”, Oakland A’s, Billy Beane.
- Engineering teams more rational than others?
  - Ever encounter a bad decision?
  - A significant fraction of requirements are left unstated
- Patterns existing in Nature do not mean the patterns are recognized by humans



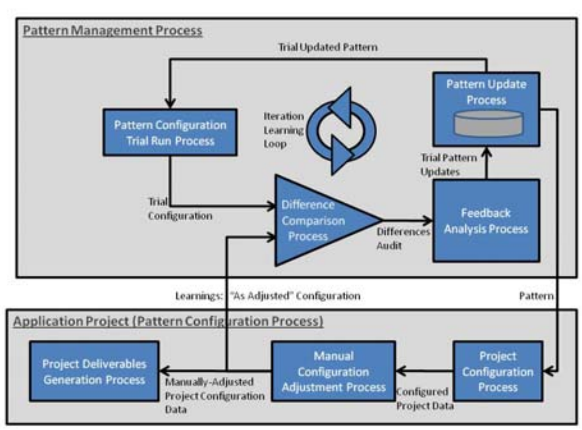


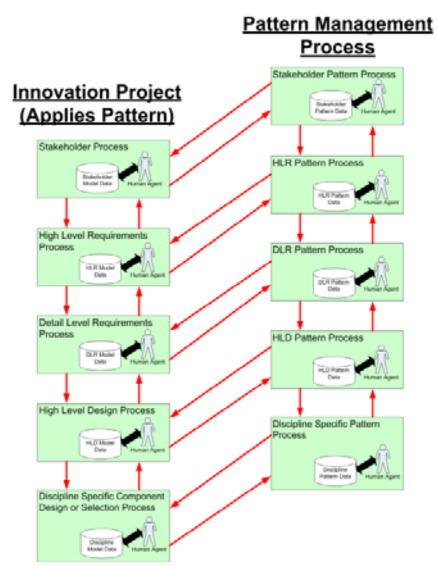
Organizational hurdles: Better business processes are nevertheless unfamiliar



Challenges and Opportunities: Organizational hurdles

- Better business processes may nevertheless be unfamiliar;
- Some familiar organizational paradigms can be leveraged in explaining to others: e.g.:
  - Standards groups, change control boards
  - Platform management processes
  - Standard parts processes





Exercise: What seems most important? What seems most actionable?

INCOSE International Symposium
Philadelphia, PA
June 24-27, 2013

Pattern Applications & Benefits	Importance	Actionable
1. Stakeholder Features and Scenarios: Better stakeholder alignment sooner		
2. Pattern Configuration: Generating better requirements faster		
3. Selecting Solutions: More informed trade-offs and design reviews		
4. Design for Change: Analyzing and improving platform resiliency		
5. Risk Analysis: Pattern-enabled FMEAs		
6. Verification: Generating better verifications, tests faster		

- Rank importance (1-6 ; 1 = most important)
- Rank actionable (1-6 ; 1 = most actionable)

Exercise / Group Discussion: Approaches to my situation

INCOSE International Symposium
Philadelphia, PA
June 24-27, 2013

- Write your ideas about what you could do next, in these areas:
  - Learn more:
  - Try an experiment:
  - Build a pattern:
  - Apply PBSE to:
  - Take a class:
  - Other:
- What questions are on your mind at the end of this workshop?
  -
- Group discussion

Conclusions

1. Patterns abound in the world of systems engineering.
2. These patterns extensively impact our projects, whether we take advantage of them as Explicit Patterns, or we are negatively impacted by Dark Patterns.
3. Pattern-Based Systems Engineering (PBSE) offers specific ways to extend MBSE to exploit Patterns.
4. Patterns provide benefits across many SE areas, through better models available at lower costs per project.
5. MBSE comes first—Patterns without Models is like orbital mechanics before Newton: useful but not as powerful as it could be.
6. We've had good success applying pattern-based methods in mil/aerospace, automotive, medical/health care, advanced manufacturing, and consumer product domains.
7. In site of the net benefits, change is difficult, so both MBSE and PBSE are not without challenges.

Survey

Please take the time to rate this presentation by submitting the web survey found at:

www.incose.org/symp2013/survey

References

Representing Systems:

1. W. Schindel, “Requirements statements are transfer functions: An insight from model-based systems engineering”, *Proceedings of INCOSE 2005 International Symposium*, (2005).

2. W. Schindel, “What Is the Smallest Model of a System?”, *Proc. of the INCOSE 2011 International Symposium*, International Council on Systems Engineering (2011).

3. B. Van Fraassen, *Scientific Representation: Paradoxes of Perspective*, Oxford U Press, 2008.

Patterns; Pattern-Based Systems Engineering:

4. W. Schindel, “Pattern-Based Systems Engineering: An Extension of Model-Based SE”, INCOSE IS2005 Tutorial TIES 4, (2005).

5. J. Bradley, M. Hughes, and W. Schindel, “Optimizing Delivery of Global Pharmaceutical Packaging Solutions, Using Systems Engineering Patterns” Proceedings of the INCOSE 2010 International Symposium (2010).

6. W. Schindel, and V. Smith, “Results of applying a families-of-systems approach to systems engineering of product line families”, SAE International, Technical Report 2002-01-3086 (2002).

7. D. Williams, “How Concepts of Self-Regulation Explain Human Knowledge”, *The Bent of Tau Beta Pi*, Winter (2011) 16-21.

8. W. Schindel, “The Impact of ‘Dark Patterns’ On Uncertainty: Enhancing Adaptability In The Systems World”, INCOSE Great Lakes 2011 Conference, Dearborn, MI, 2011.

9. Kahneman, D., *Thinking, Fast and Slow*, Farrar, Straus and Giroux, Publishers, 2011, ISBN**-10:** 0374275637.

10. Lewis, Michael, *Moneyball: The Art of Winning an Unfair Game*, Norton, New York, 2004.

11. W. Schindel, “Introduction to Pattern-Based Systems Engineering (PBSE)”, INCOSE Finger Lakes Chapter Webinar, April 26, 2012.

***Systems Engineering in Innovation*:**

12. W. Schindel, “Innovation as Emergence: Hybrid Agent Enablers for Evolutionary Competence” in *Complex Adaptive Systems*, Volume 1, Cihan H. Dagli, Editor in Chief, Elsevier, 2011

13. W. Schindel, S. Peffers, J. Hanson, J. Ahmed, W. Kline, “All Innovation is Innovation of Systems : An Integrated 3-D Model of Innovation Competencies ”, Proc. of ASEE 2011 Conference (2011).

14. W. Schindel, “Systems of Innovation II: The Emergence of Purpose”, *Proceedings of INCOSE 2013 International Symposium* (2013).

15. INCOSE System Sciences Working Group, Systems of Innovation Project web site: https://sites.google.com/site/syssciwg/projects/o-systems-of-innovation

Analysis of Architecture, Changeability, Modularity:

16.Carlos A. Osorio, Dov Dori, Joseph Sussman, “COIM: An Object-Process Based Method for Analyzing Architectures of Complex, Interconnected, Large-Scale Socio-Technical Systems”, *Systems Engineering* Vol. 14, No. 4, pp. 364-382, 2011.

17.Jason E. Bartolomei, Daniel E. Hastings, Richard de Neufville, and Donna H. Rhodes, “Engineering Systems Multiple-Domain Matrix: An Organizing Framework for Modeling Large-Scale Complex Systems”, *Systems Engineering,* Vol. 15, No. 1, pp. 41- 61, 2012.

18.Azad M. Madni, “Adaptable Platform-Based Engineering: Key Enablers and Outlook for the Future”, *Systems Engineering,* Vol. 15, No. 1, pp. 95-107, 2012.

19.Tobias K.P. Holmqvist and Magnus L. Persson, “Analysis and Improvement of Product Modularization Methods: Their Ability to Deal with Complex Products”, *Systems Engineering*, Vol. 6, No. 3, pp. 195-209, 2003.

20.Ernst Fricke, and Armin P. Schulz, “Design for Changeability (DfC): Principles To Enable Changes in Systems Throughout Their Entire Lifecycle”, *Systems Engineering*, Vol. 8, No. 4, pp. 342-359, 2005

21.Avner Engel, * and Tyson R. Browning, “Designing Systems for Adaptability by Means of Architecture Options”, *Systems Engineering*, Vol. 11: pp. 125–146, 2008.

22. David M. Sharman and Ali A. Yassine, “Characterizing Complex Product Architectures”, *Systems Engineering*, Vol. 7, No. 1, pp. 35-60, 2004.

23.Adam M. Ross, Donna H. Rhodes, and Daniel E. Hastings, “Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining System Lifecycle Value”, *Systems Engineering*, 11:3, 2008.

24. “Engineering Resilient Systems”, US DoD, http://www.acq.osd.mil/chieftechnologist/areas/ers.html

25. de Weck, Ross and Magee, *Engineering Systems*, MIT Press, 2011 .

26. de Neufville and Scholtes, *Flexibility in Engineering Design,* MIT Press, 2011.

Other Systems Engineering References:

27. ISO/IEC 15288: Systems Engineering—System Life Cycle Processes. International Standards Organization (2008).

28.INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, Version 3.2, International Council on Systems Engineering (2010).

29.NASA Systems Engineering Handbook, NASA/SP-2007-6105, Rev 1, U.S. National Aeronautics and Space Administration (2007).

30.W. Schindel, “Failure Analysis: Insights from Model-Based Systems Engineering”, *Proceedings of INCOSE 2010 Symposium*, July 2010.

About the presenters

Troy Peterson is a Senior Associate at Booz Allen Hamilton and his expertise is in strategy, systems engineering and management. He has led several distributed teams in delivery of large-scale complex systems and has instituted numerous organizational processes to improve efficiency and effectiveness. His consulting experience spans academic, commercial and government sectors as well as all lifecycle phases of program and product development. Troy obtained a BS in Mechanical Engineering from Michigan State University, a MS in Business and Technology Management from Rensselaer Polytechnic Institute and completed advanced graduate studies at Massachusetts Institute of Technology in System Design and Management. Troy is also the Past President of the INCOSE Michigan Chapter and an INCOSE CSEP, PMI PMP, and ASQ CSSBB.

Bill Schindel is president of ICTT System Sciences (www.ictt.com), a systems engineering company. His 40-year engineering career began in mil/aero systems with IBM Federal Systems, Owego, NY, included service as a faculty member of Rose-Hulman Institute of Technology, and founding of three commercial systems-based enterprises. He has led and consulted on improvement of engineering processes within automotive, medical/health care, manufacturing, telecommunications, aerospace, and consumer products businesses. Schindel earned the BS and MS in Mathematics. At the 2005 INCOSE International Symposium, he was recognized as the author of the outstanding paper on Modeling and Tools, and currently co-leads a research project on the science of Systems of Innovation within the INCOSE System Science Working Group. Bill is an INCOSE CSEP, and president of the Crossroads of America INCOSE chapter.