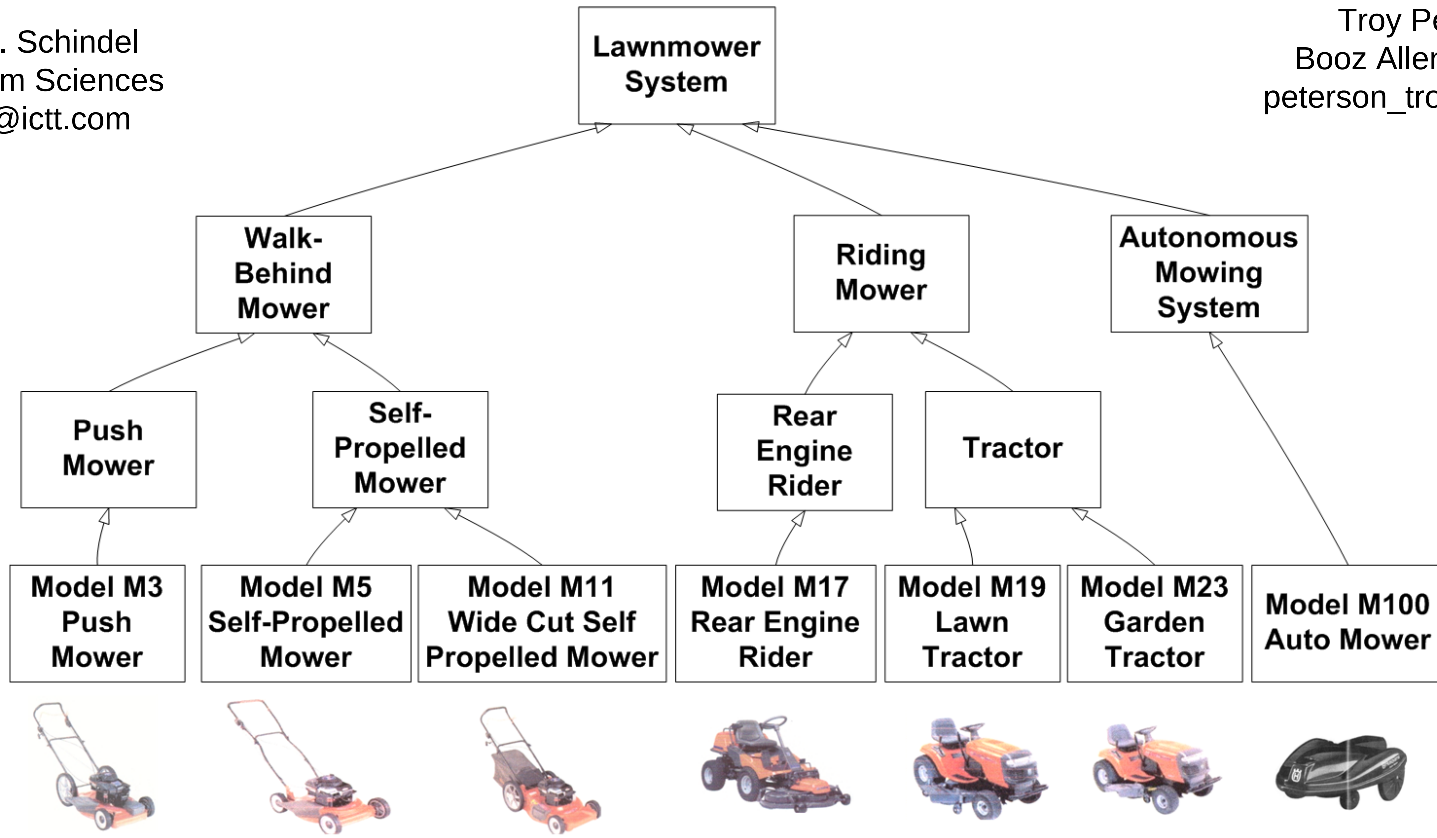
An Overview of Pattern-Based Systems Engineering (PBSE): Leveraging MBSE Techniques

William D. Schindel
ICTT System Sciences
schindel@ictt.com

Troy Peterson
Booz Allen Hamilton
peterson_troy@bah.com



Abstract

Abstract: This tutorial is a practitioner's brief overview of Pattern-Based Systems Engineering (PBSE), including some specific system domain illustrations.

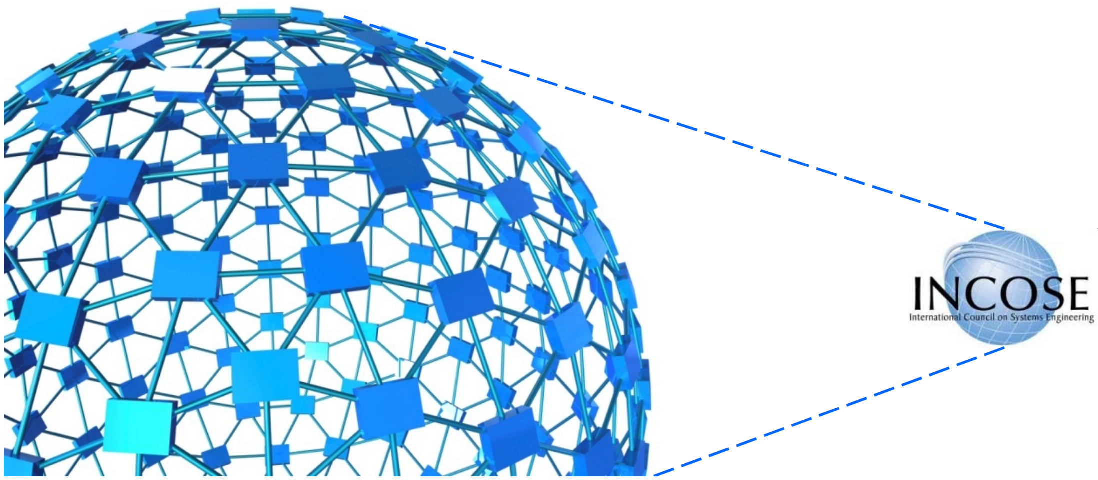
INCOSE thought leaders have discussed the need to address 10:1 more complex systems with 10:1 reduction in effort, using people from a 10:1 larger community than the "systems expert" group INCOSE currently reaches. Through the PBSE Challenge Team of the INCOSE/OMG MBSE Initiative, the team aims to enable 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 only limited INCOSE community awareness, through IS tutorials, and most recently the start- up of the PBSE Challenge Team at the IW2014 LA meeting in January.

Contents

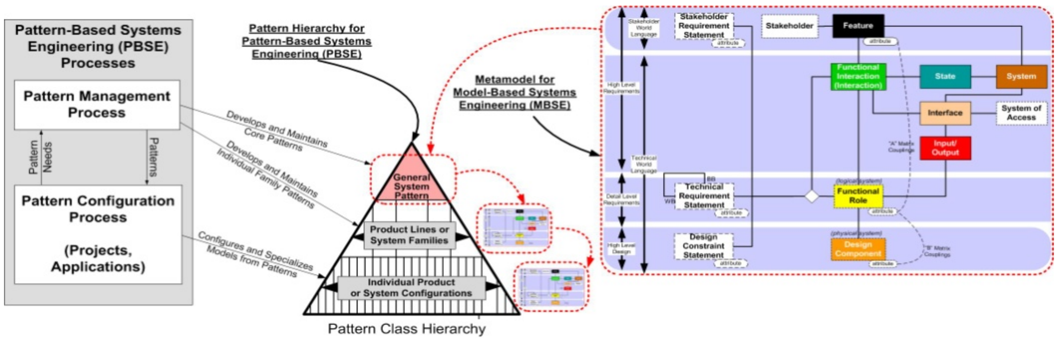
The Need, Call-to-Arms, and Vision

- INCOSE thought leaders have discussed the growing need to address 10:1 more complex systems with 1:10 reduction in effort, using people from a 10:1 larger community than the “systems expert” group INCOSE currently reaches.
- 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.
- This talk is about a way to achieve this order-of-magnitude improvement:
  - Using Pattern-Based Systems Engineering (PBSE)—an extension of Model-Based Systems Engineering (MBSE).



Concept Summary: Pattern-Based Systems Engineering (PBSE)

- The PBSE approach respects the systems engineering tradition, body of knowledge, and historical lessons, while providing a high-gain path forward.
- An S* Pattern is a configurable, re-usable S* Model (S*Metamodel compliant). It is an extension of the idea of a Platform (which is a configurable, re-usable design). The Pattern includes not only the Platform, but all the extended system information (e.g., requirements, risk analysis, design trade-offs & alternatives, decision processes, etc.):



- By including the appropriate S* Metamodel concepts, these can readily be managed in (SysML or other) preferred modeling languages and 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.

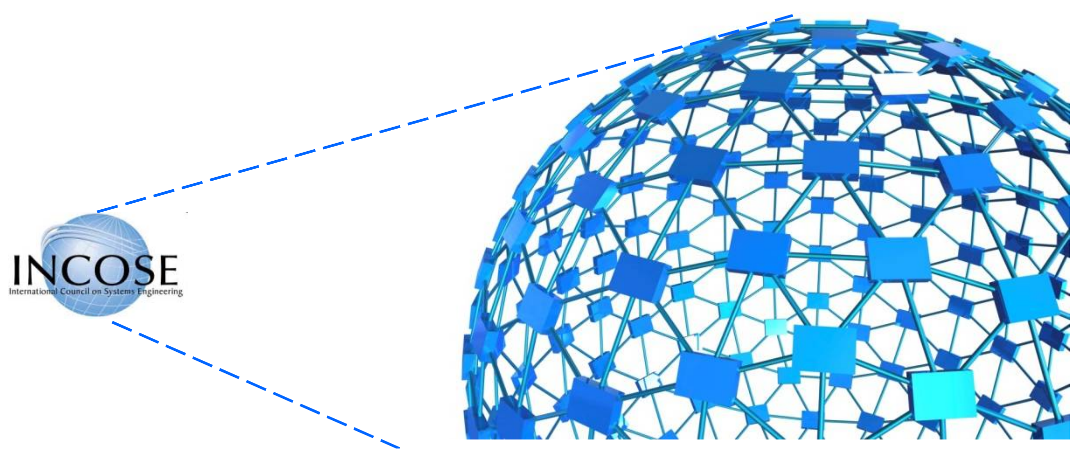
Status of PBSE

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

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

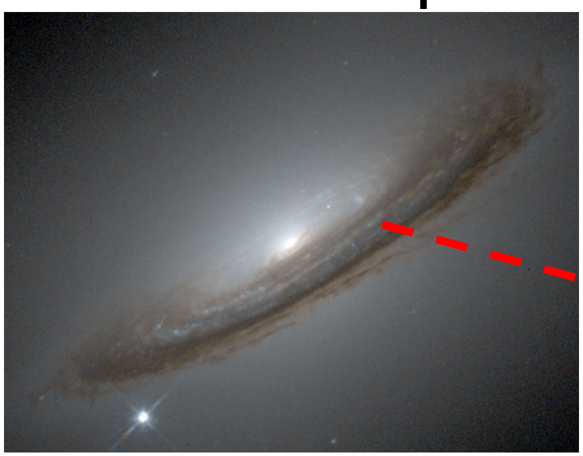
– This talk is more about INCOSE community awareness and capability than about technically establishing a new method—although it will look new to INCOSE practitioners.

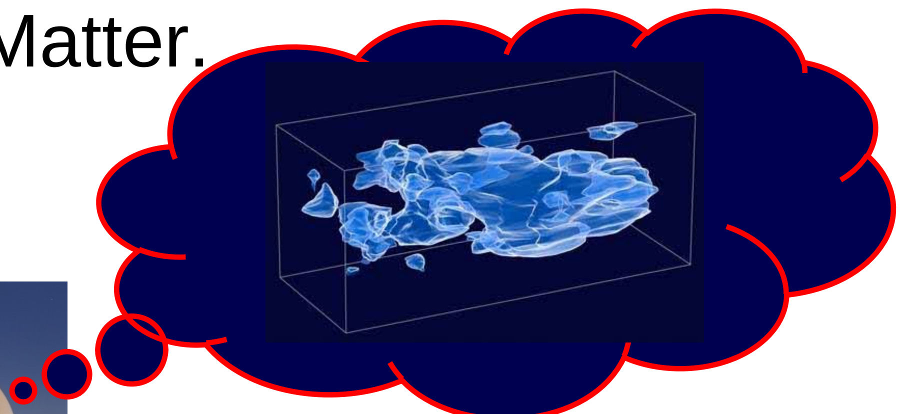
– We recognize that the human change aspect can be the most challenging – but are not suggesting that we also have to create new technical methods. We are <u>introducing PBSE to a larger community</u>.



Dark Matter

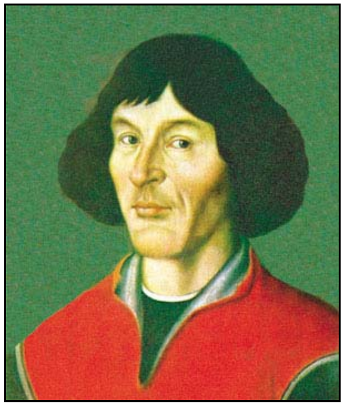
- Some cosmologists believe "Dark Matter" exists:
  - Is invisible (optically),
  - Exerts gravitational force on the rest of matter,
  - Is a major & widespread component of the universe.
- Otherwise unexplained behavior of the universe seems explained by Dark Matter.

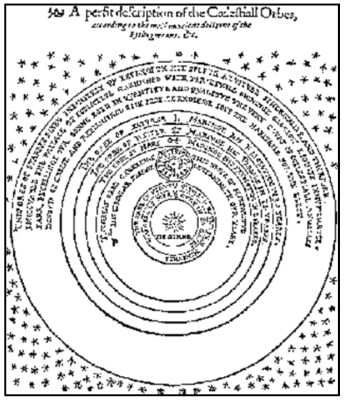


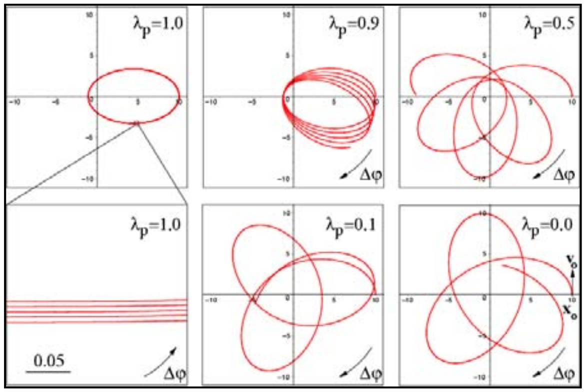


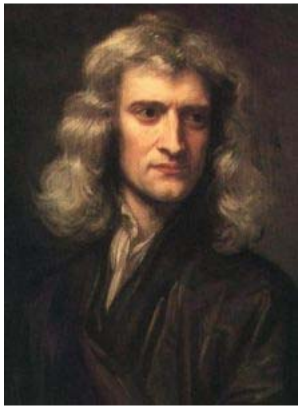
Science Seeks Models

- Will Dark Matter become fully accepted by the scientific community to explain the patterns of observed behavior?
  – It is still early to say.
- Earlier, the Copernican Revolution required <u>generations</u> to be the accepted explanation of planetary behavior patterns:
  – Others after Copernicus (Galileo, Brahe, Kepler, Newton) were aided by finding <u>improved methods of representation</u> to better explain <u>regularities of observed behavior</u>.



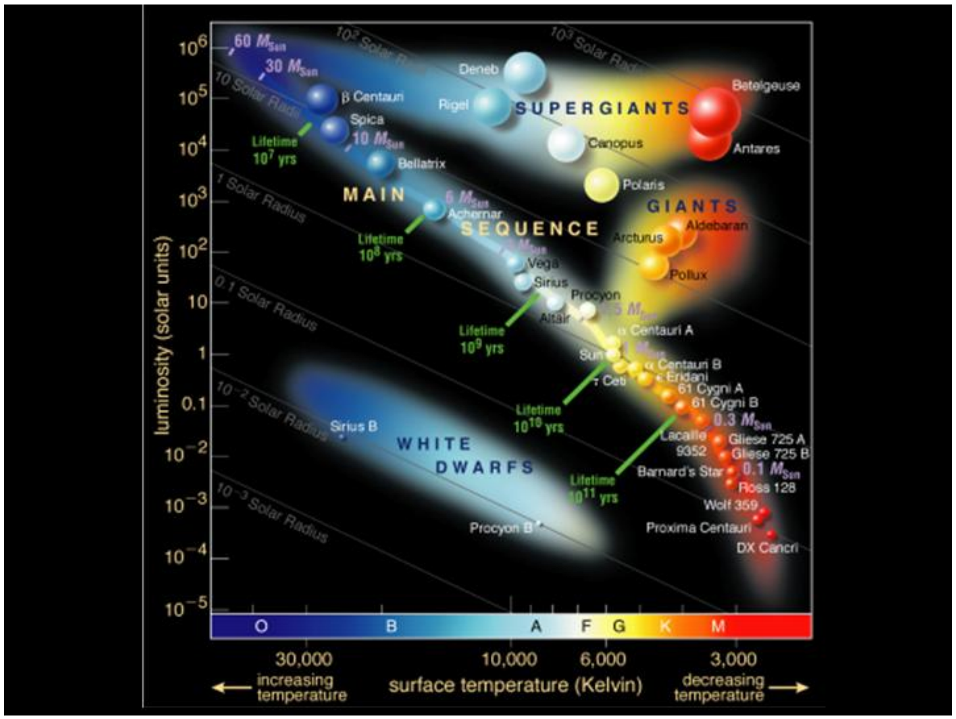






One-off Models versus Family Patterns

- These improved understandings depended not just on better Models of individual situations, but also on . . .
- Representation of Families, helping to understand different types of behavior, organizing the universe further into “types”:



The systems engineering connection

- *Discovering regularities and how to represent them* has been at the heart of science and engineering progress:
  - The INCOSE System Sciences Working Group (SSWG) bridges the interests of engineering and science.
  - Next meeting will be at IW2012.
  - https://sites.google.com/site/syssciwg/
- *Ability to manage risk and adapt* are related to our *awareness* and *understanding* of the *regularities* (patterns) around us:
  - Whether in the systems we engineer, or the markets and operational environments in which their life cycle unfolds.
  - They exert "forces" on us, whether are aware of them or not.

What repeating regularities are of interest?

- Smaller-Scale Regularities:
  - Patterns of Stakeholder Features (e.g., in vehicles, energy systems, etc.)
  - Patterns of Requirements
  - Patterns of Design Solutions
  - Patterns of Failure Modes and Effects
  - Patterns of Functional Roles, Interactions, States
  - Patterns of Interfaces, Input-Outputs, and Access
  - Patterns of Technologies

- Larger-Scale Regularities:
  - Patterns of how all the above are related to each other
  - Patterns in couplings across systems, domains, SOS's
  - Systems of Material Handling, Production, Distribution, Sustainment
  - Systems of Innovation
  - Patterns of Systems Pathologies

INCOSE SSWG

Is this “just of academic interest”?

- Hardly! Lack of awareness of these regular patterns leaves products, programs, enterprises at serious risk:
  - Re-experiencing the same mis-steps and reworks;
  - Just because we have made one system work, how do we know what will happen when we deploy more of them, as markets, conditions, & technologies evolve?
  - Just because our system has human experts on hand today, how do we know what will happen when they move on?
- Example cases and responses:
  - FDA push to the pharmaceutical manufacturing industry to improve the science-based understanding of underlying process transformations, provable ranges, and control strategies, etc.
  - The generation of system requirements families for globally-deployed product families and their production, distribution, and support systems.
  - The generation of system verification plans from underlying patterns of system requirements.
  - The use of System Patterns to generate Risk Analyses (e.g., FMEAs, etc.) for a variety of domain systems.

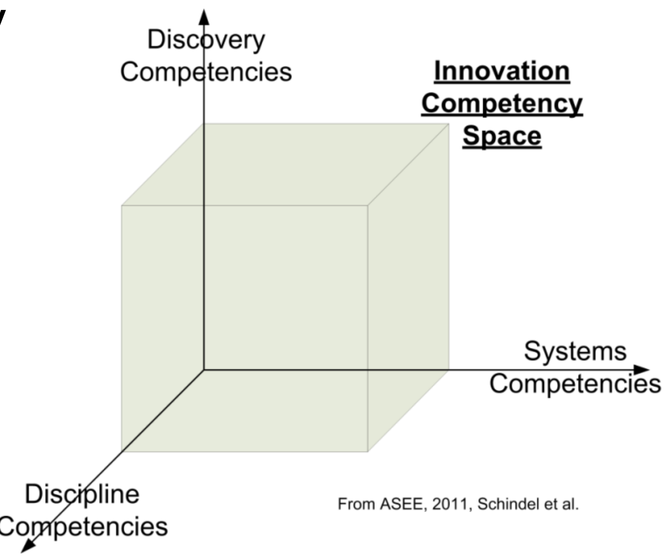
Benefits of applying system patterns

- 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
- Practice PBSE with a goal in mind: What benefits seem most important?

"Chance favors the prepared mind"

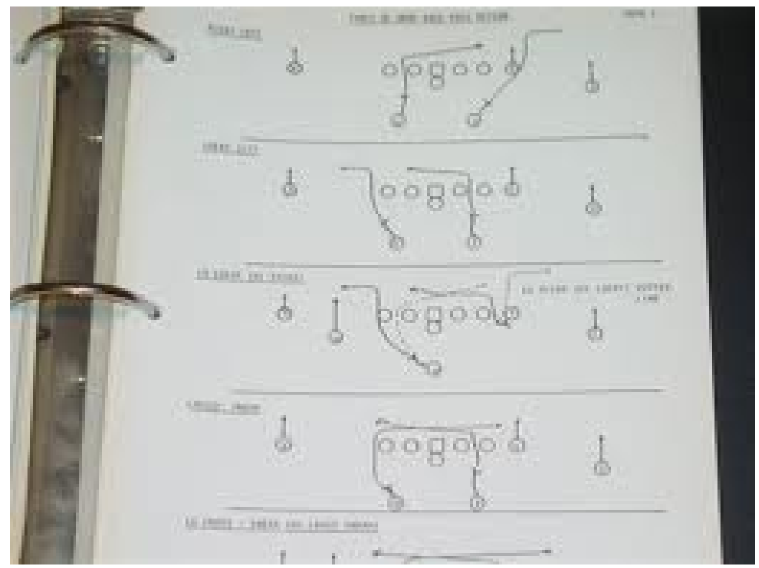
- Louis Pasteur

- Explicit patterns help us organize what we know--as well as what we don't.
- Explicit preparation for:
  - System & program risks
  - Market & competitive shifts
  - New science & technology
  - Life cycle extensions
- Adaptability!



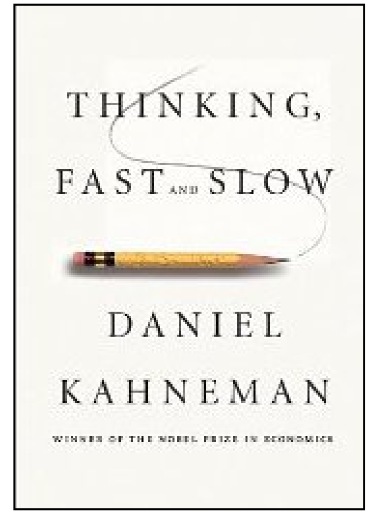
Adaptation Response Time

- Explicit pattern awareness helps us to:
  - Recognize the situation has changed.
  - Know the best alternate pattern configuration.

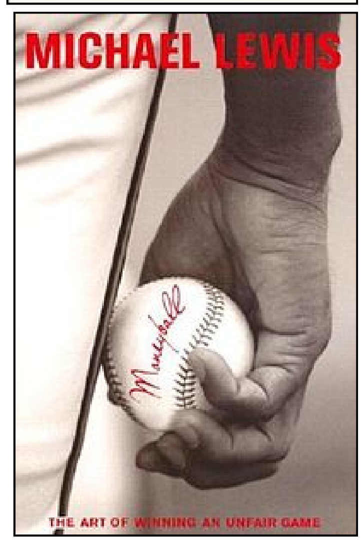


Irrationality: Human beings' behaviorally-preferred mode?

- A broad issue across human life:
  - The science of irrationality
  - Daniel Kahneman, Nobel Laureate, "Thinking, Fast and Slow")
  - "Moneyball", Oakland A's, Billy Beane.



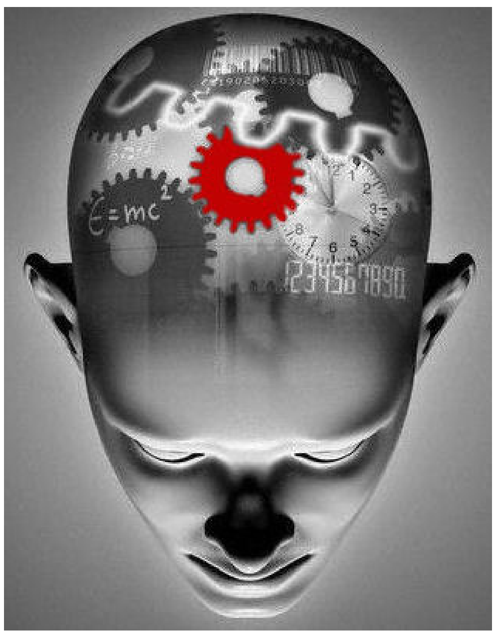
- Engineering teams more rational than others?
  - Ever encounter a bad decision?
  - A significant fraction of requirements are left unstated



- Patterns existing in Nature do not mean the patterns are recognized by humans

One way people cope . . .

- “Domain experts” internalize patterns:
  - These human experts influence our projects, using their experience, intuition, informed judgment.



System Patterns: Dark and Visible

- The regularities are "out there", whether we represent them or not:
  - In particular, they impact our ability to deal with uncertainty and adaptability.
- We use the term <u>Dark Pattern</u> to refer to system regularities that have <u>not</u> been explicitly represented:
  - They are in a sense "invisible", but still impact our systems, customers, programs, enterprises, institutions, and society.
- By contrast, when we represent those System Patterns formally, they become "visible", as <u>Explicit Patterns</u>:
  - Our method for doing this is Pattern-Based Systems Engineering (PBSE);
  - PBSE is an extension of Model-Based Systems Engineering (MBSE);
  - PBSE creates and applies configurable, re-usable models, called <u>Patterns</u>;
  - They typically include much more than just the "subject system".

How many patterns are Dark?

- Most systems programs involve Patterns, such as:
  - Patterns of available technologies and parts
  - Patterns of candidate solution architectures
  - Patterns of interfaces
  - Patterns of system states or modes
  - Patterns of customers, or market expectations
  - Patterns of competitive offerings
  - Patterns of system failures modes and effects
- Most systems engineering efforts—even model-based--still occur without use of explicit Pattern-Based methods:
  - This is the world of Dark Patterns.
  - Example: Nearly universally missed requirements.
- Explicit Patterns prepare us to adapt by describing key objects, relationships, and variables—including multiple types of risk.

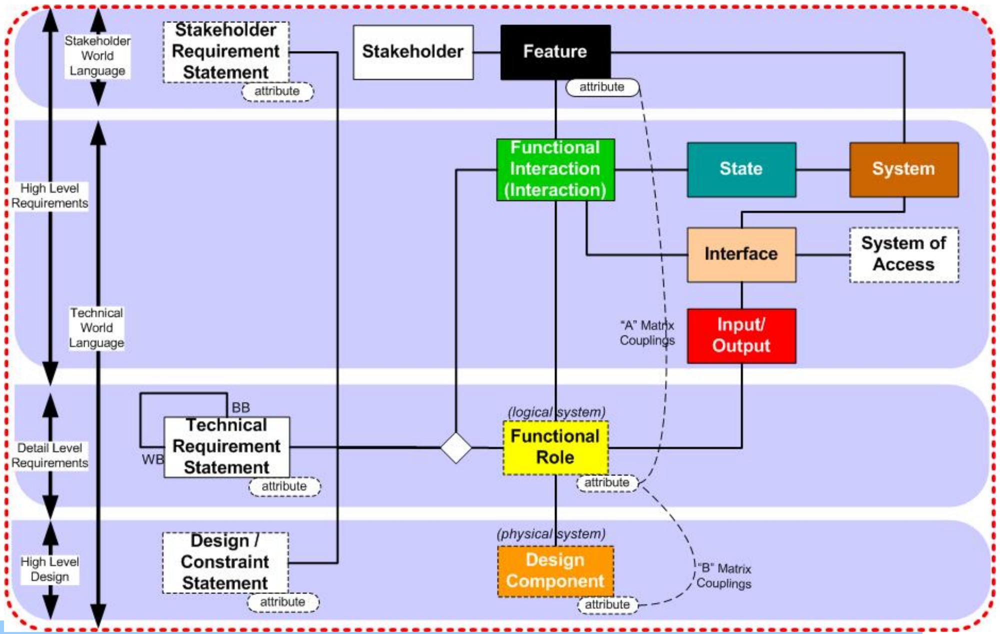
Representing System Patterns

- What is the smallest amount of information we need to represent these regularities?
  - Some people have used prose to describe system regularities.
  - This is better than nothing, but usually not enough to deal with 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.

Constructing an efficient representation

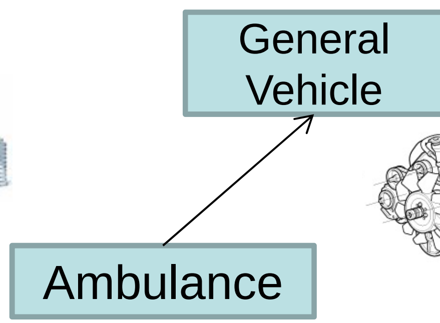
- A <u>metamodel</u> is a model of other models;
  - Sets forth how we will represent Requirements, Designs, Verification, Failure Analysis, Trade-offs, etc.;
  - We utilize the (language independent) S* Metamodel from Systematica™ Methodology:
- The resulting system models may be expressed in SysML™,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

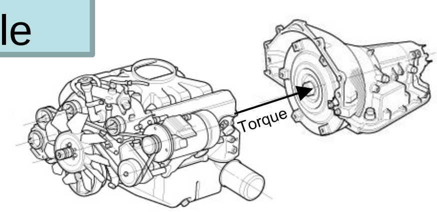
Simple summary of detailed S* Metamodel.

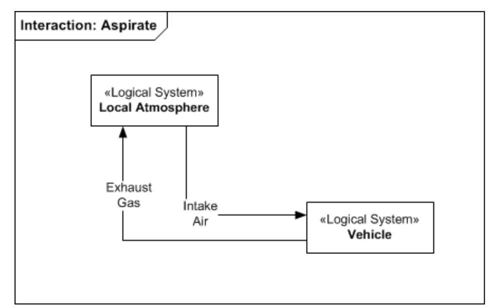


PBSE Enablers: Definitions of some S* Metamodel Classes

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





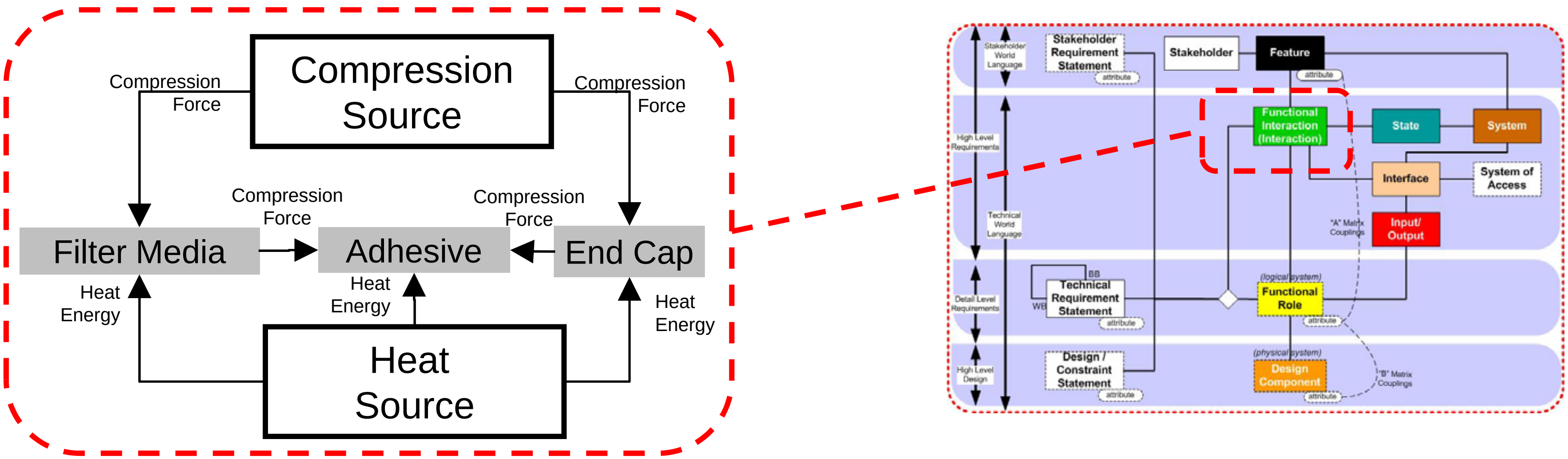


PBSE Enablers: Definitions of some S* Metamodel Classes

- **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

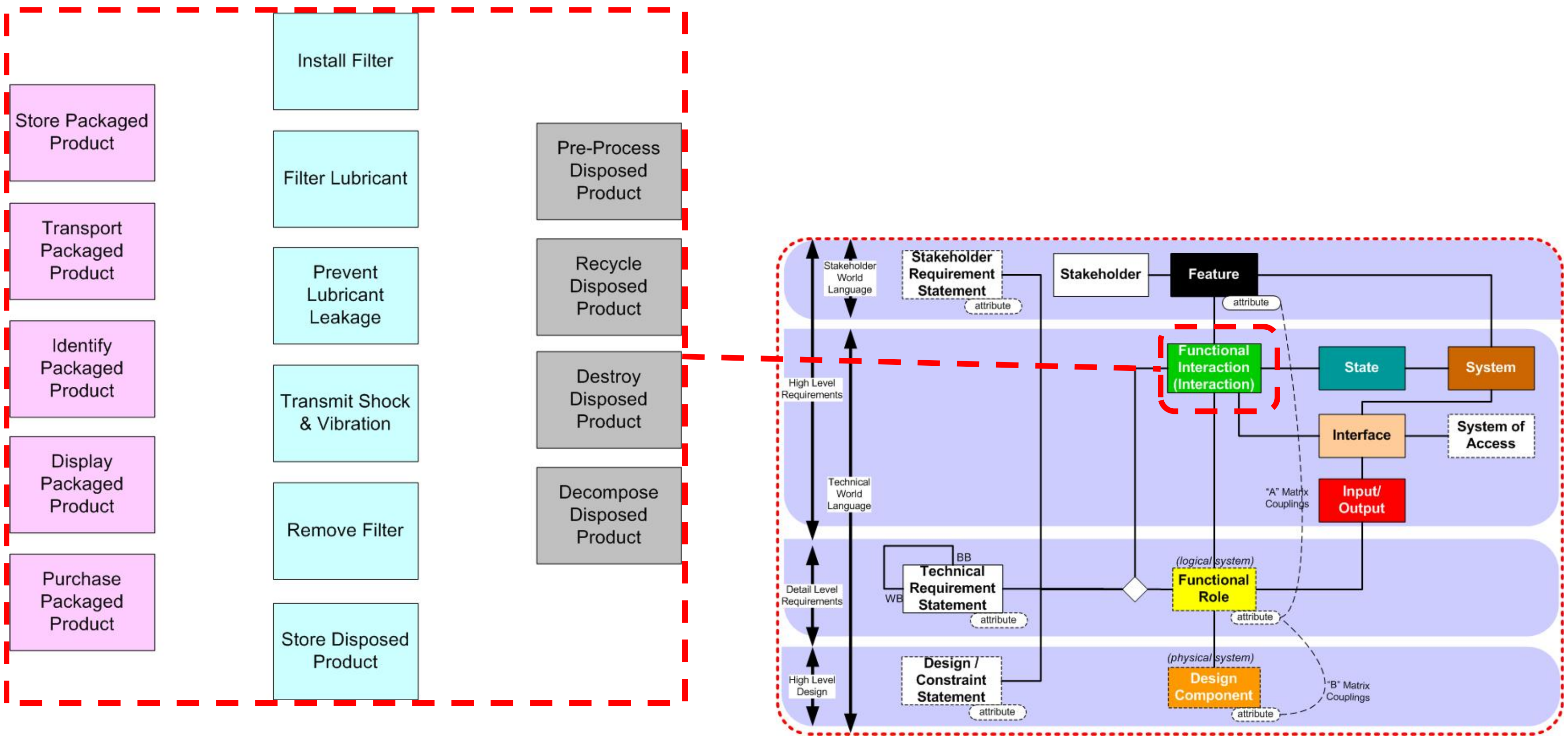
- S* models represent <u>Interactions </u>as explicit objects:
  - Goes to the heart of 300 years of natural science of systems as a foundation for engineering, including emergence.
  - All functional requirements are revealed as external interactions [Schindel].



- Other Metamodel parts: See the References.

Physical Interactions: At the heart of S* models

- S* models represent Physical Interactions as explicit objects:
  - Example: Pattern of Oil Filter Interactions:



Pattern-based systems engineering (PBSE)

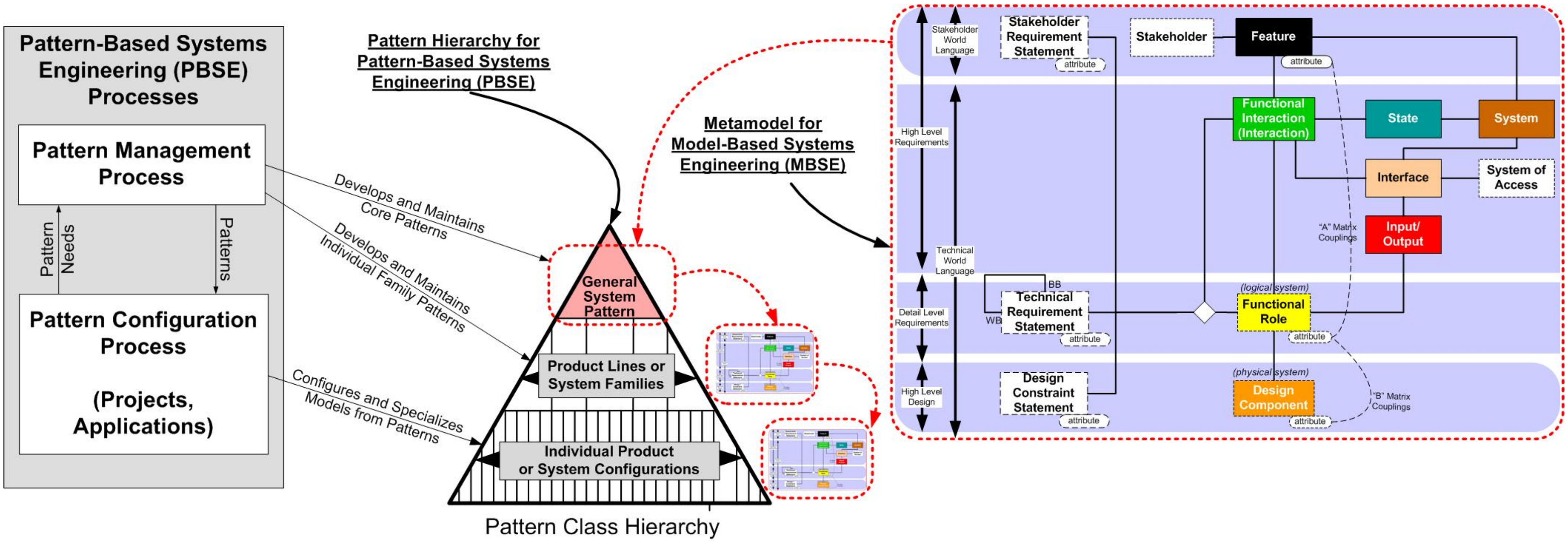
- Model-based Patterns:
  - In this approach, Patterns are reusable, configurable S* models of families (product lines, sets, ensembles) of systems.
- These Patterns are ready to be configured to serve as Models of individual systems in projects.
- Configured here is specifically limited to mean that:
  - 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 are based on the same Metamodel as “ordinary” Models

Pattern-based systems engineering (PBSE)

- Pattern-Based Systems Engineering (PBSE) has two overall processes:
  - **Pattern Management Process**: Generates the underlying family model, and periodically updates it based on application project discovery and learning;
  - **Pattern Configuration Process**: Configures the pattern into a specific model for application in a project.

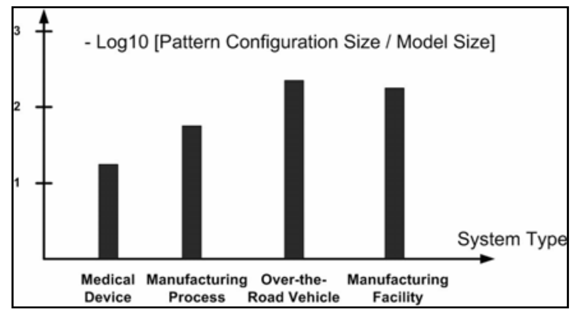


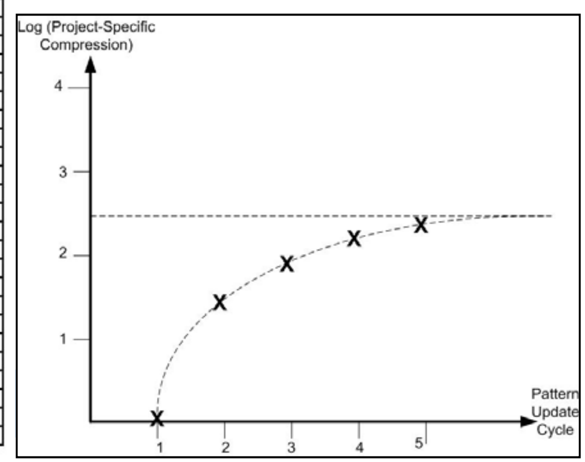
Pattern configurations

- 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

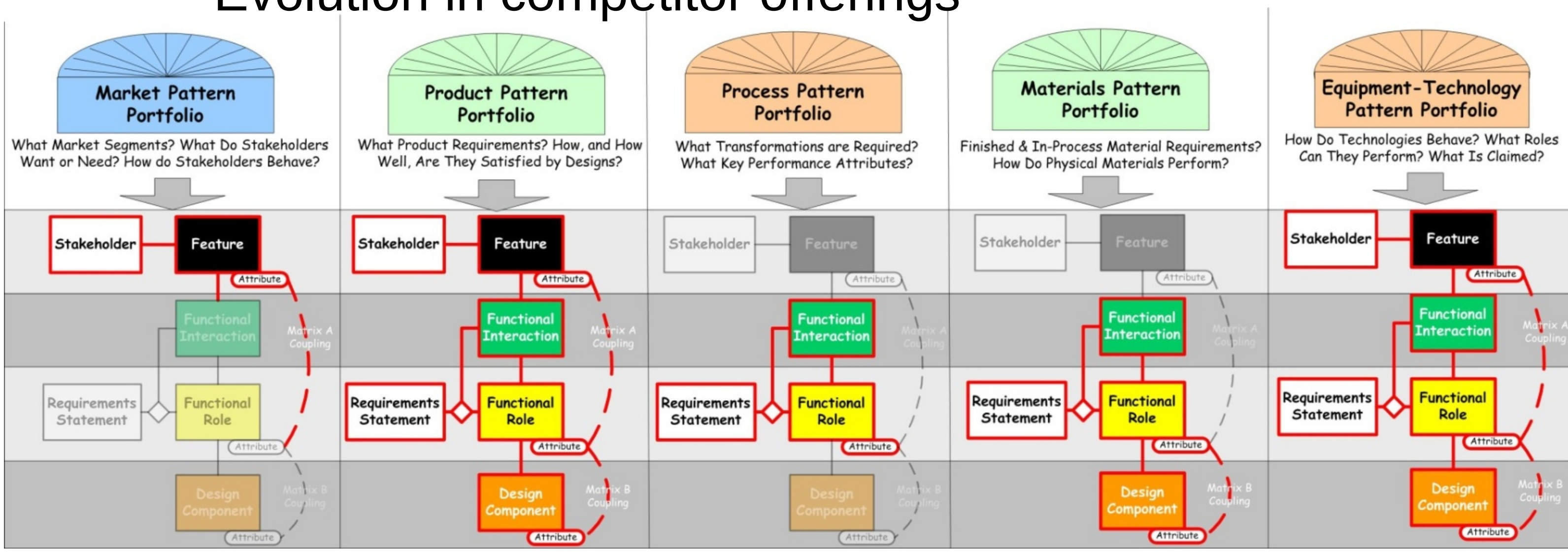
		Units	Walk-Behind	Walk-Behind	Walk-Behind	Riding	Riding	Riding Mower	Autonomous
			Push Mower	Mower	Self-Propelled	Rider	Tractor	Tractor	Autonomous
			Push Mower	Self-Propelled	Wide Cut	Rider	Lawn	Garden	Auto Mower
	Model Number		M3	M5	M11	M17	M19	M23	M100
	Market Segment		Sm Resident	Med Resident	Med 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	





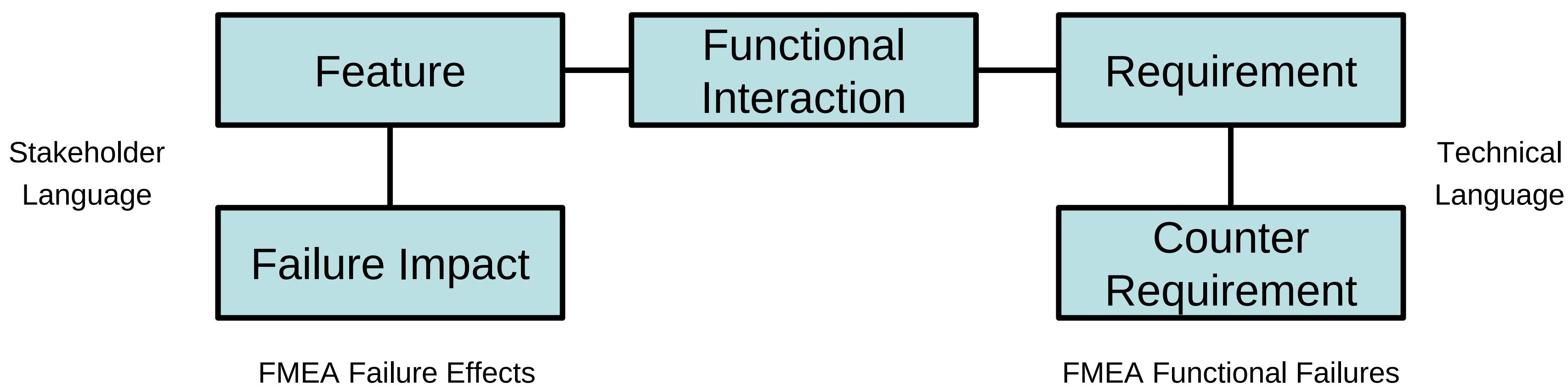
Patterns organize portfolios

- Patterns express “envelopes” around “point situations”.
- Patterns help us discover, explore, and record what we may have to adapt to, along with adaptation plans:
  - Evolution in available technologies and parts
  - Evolution in system requirements, interfaces, modes, etc.
  - Evolution in the larger systems in which we operate
  - Evolution in customer or market expectations
  - Evolution in competitor offerings



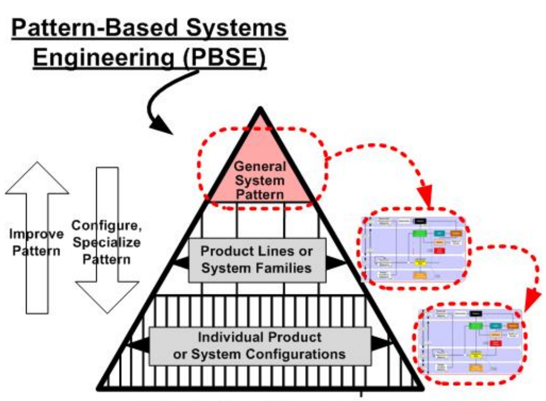
Patterns for managing risk

- Patterns also express risks and mitigations for:
  - Patterns of system failure modes and effects (d-FMEA)
  - Patterns of operator failure modes and effects (a-FMEA)
  - Patterns of production & distribution failures (p-FMEA)

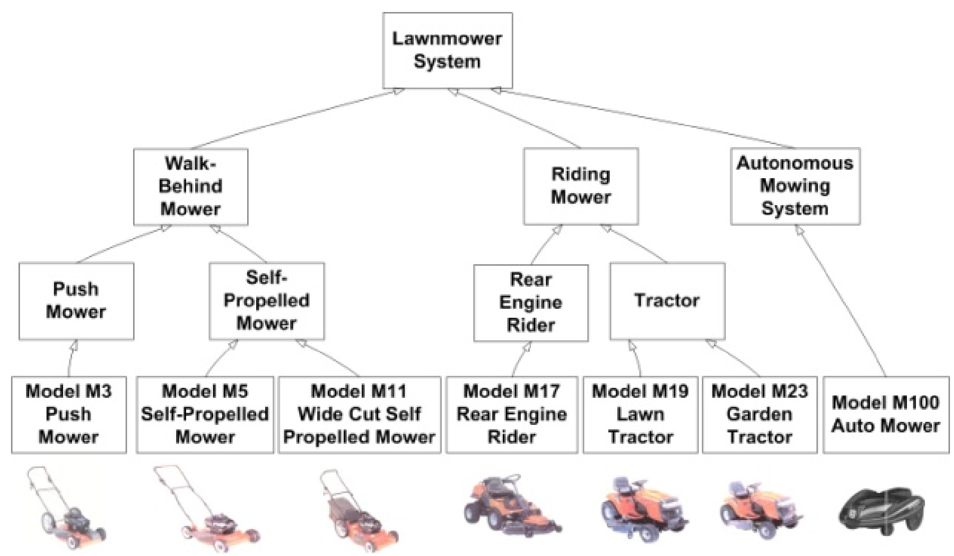


PBSE helps make Platform Management a discipline

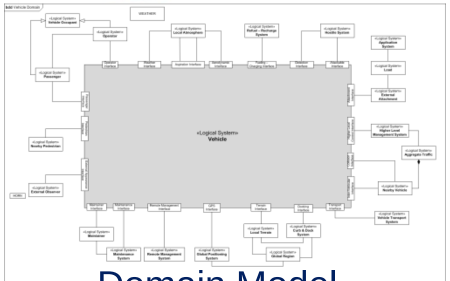
- Descriptions of SE processes typically appear to describe engineering a “new” system “from scratch” [e.g., ISO 15288, INCOSE SE Handbook]:
  - However, real projects are often concerned with engineering similar (but different) systems across different product generations, applications, configurations, or market segments.
  - Patterns provide the IP basis to make Platform Management a discipline, not just an attractive idea:



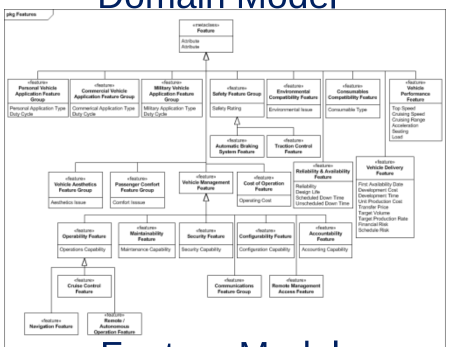
Pattern Class Hierarchy



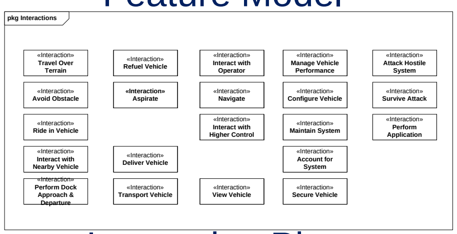
Vehicle Pattern Example: SysML Model, Organized in Packages



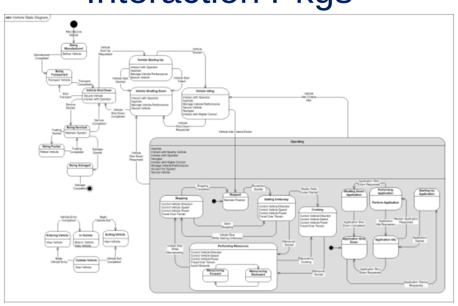
Domain Model



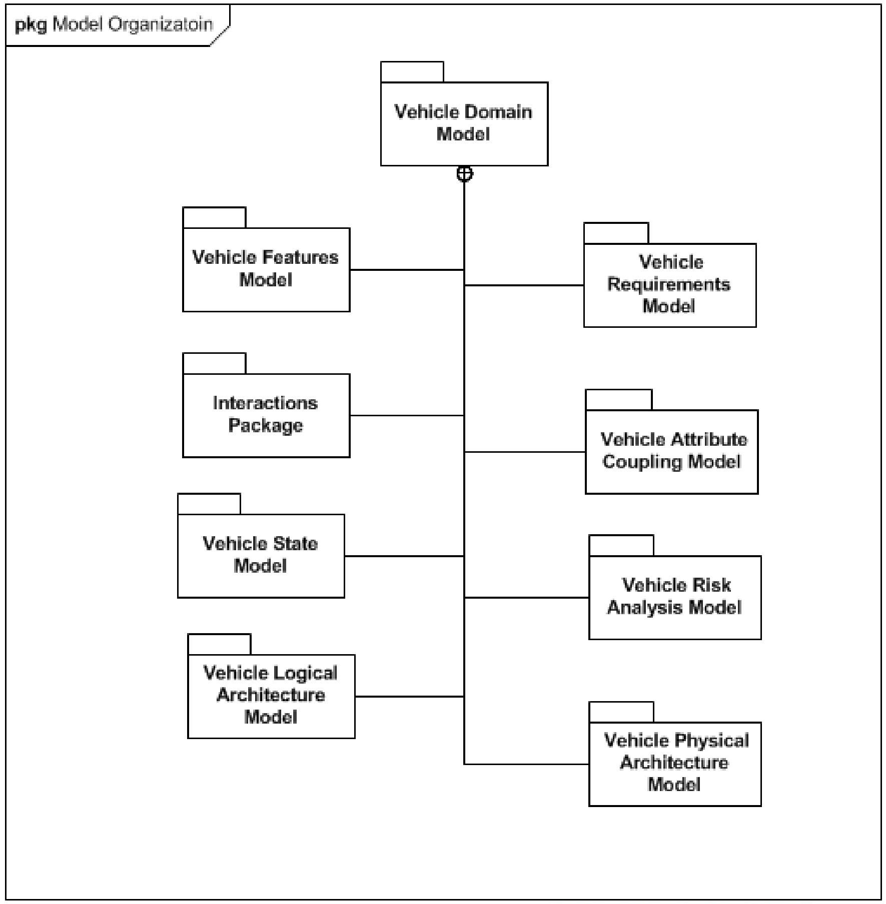
Feature Model

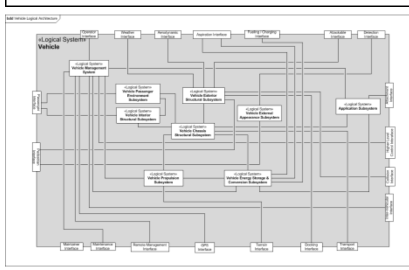


Interaction Pkgs



State

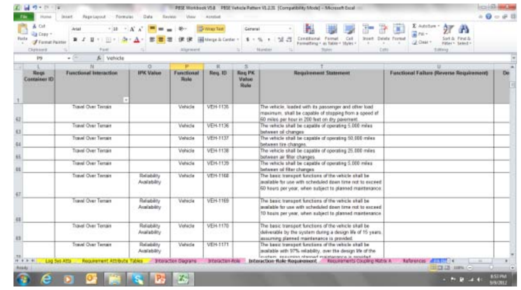




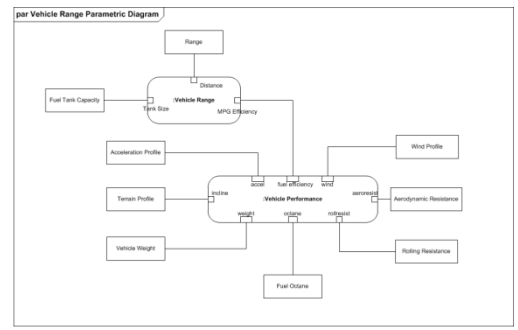
Logical Architecture

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

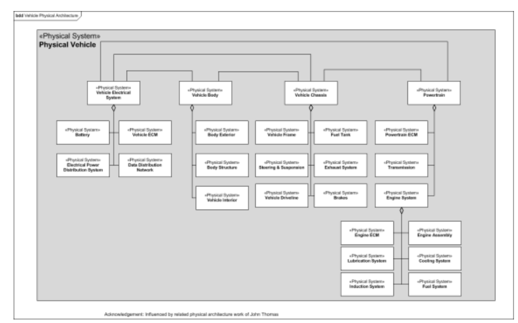
FMEA



Integrated Tables/Models



Parametrics



Physical Architecture

New INCOSE Patterns Challenge Team

- Part of the MBSE Initiative
- First meeting was at IW2014 in January, 2014
- Next meeting is at IS2014
- See http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns

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. MBSE comes first—Patterns without Models is like orbital mechanics before Newton.
5. PBSE provides a number of identified benefits.
6. We've had good success applying pattern-based methods in mil/aerospace, automotive, medical/health care, advanced manufacturing, and consumer product domains.

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, T. Peterson, “Introduction to Pattern-Based Systems Engineering: Leveraging MBSE”, INCOSE IS2013 Tutorial, Philadelphia, 2013.

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).

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. Eric Berg, “Affordable Systems Engineering: An Application of Model-Based System Patterns to Consumer Packaged Goods Products, Manufacturing, and Distribution”, INCOSE IW2014 MBSE Workshop, Los Angeles, CA, January 2014.

12. INCOSE Patterns Challenge Team of MBSE Initiative web site, with link to downloadable refs: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns

***Systems Engineering in Innovation*:**

13. 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

14. 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).

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

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.

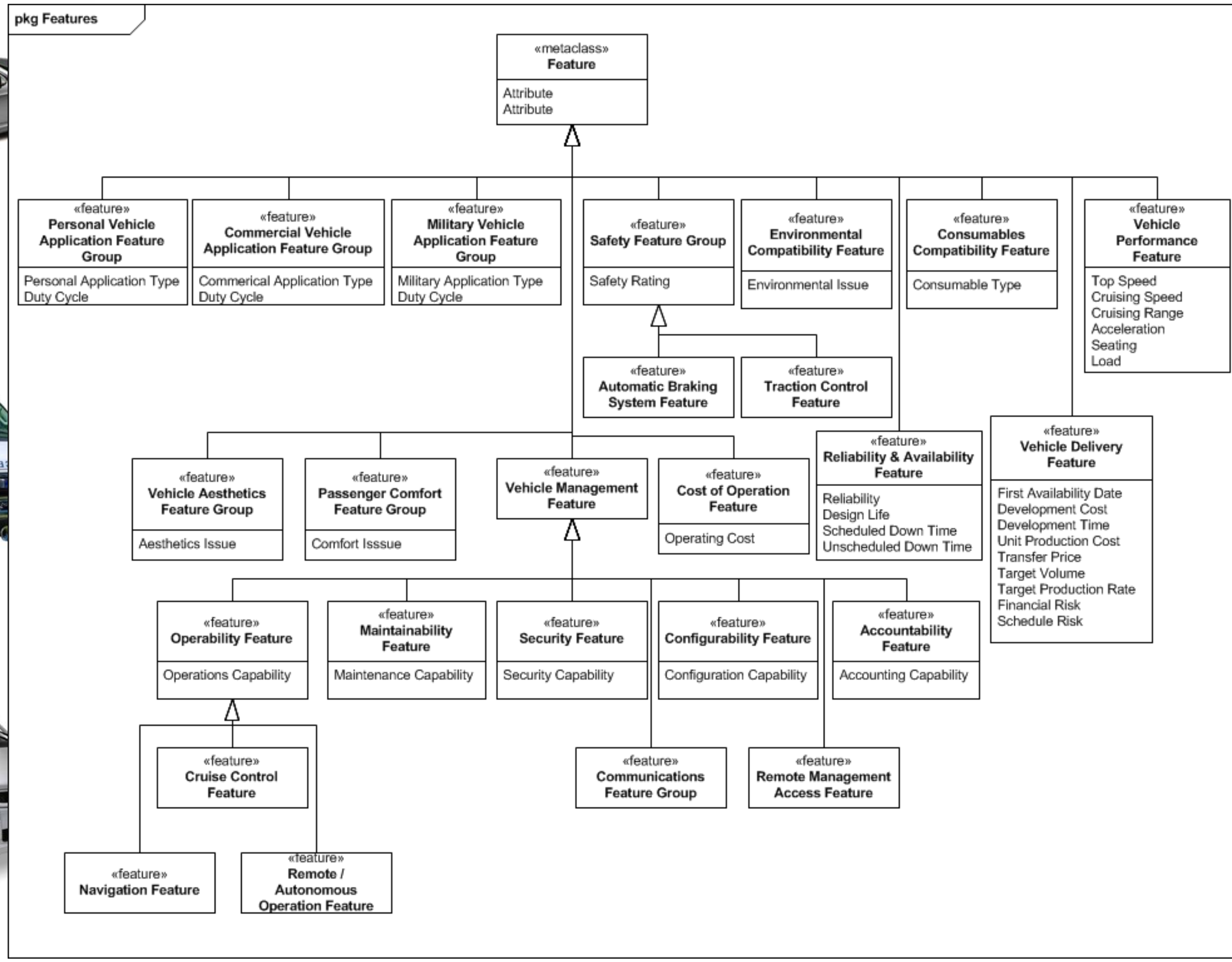
Speakers

Presenter: 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 he co-leads the Patterns Challenge Team of the INCOSE / OMG MBSE Initiative. Bill is an INCOSE CSEP, and president of the Crossroads of America INCOSE chapter.

Presenter: Troy Peterson is a Booz Allen Fellow leading a portion of the firm's Systems Engineering capability and a portfolio of work in Michigan. His areas of expertise include systems engineering and management, strategy and enterprise development. He has led several teams in the delivery of complex systems and has instituted numerous methodologies to speed development cycles. His experience spans academic, commercial and government environments across all product life cycle phases. Troy completed advanced graduate studies at Massachusetts Institute of Technology in System Design and Management, obtained a MS in Technology Management from Rensselaer Polytechnic Institute and a BS in Mechanical Engineering from Michigan State University. Troy an INCOSE Michigan Chapter Past President and CSEP, PMI PMP, and ASQ CSSBB. He co-leads the Patterns Challenge Team of the INCOSE / OMG MBSE Initiative.

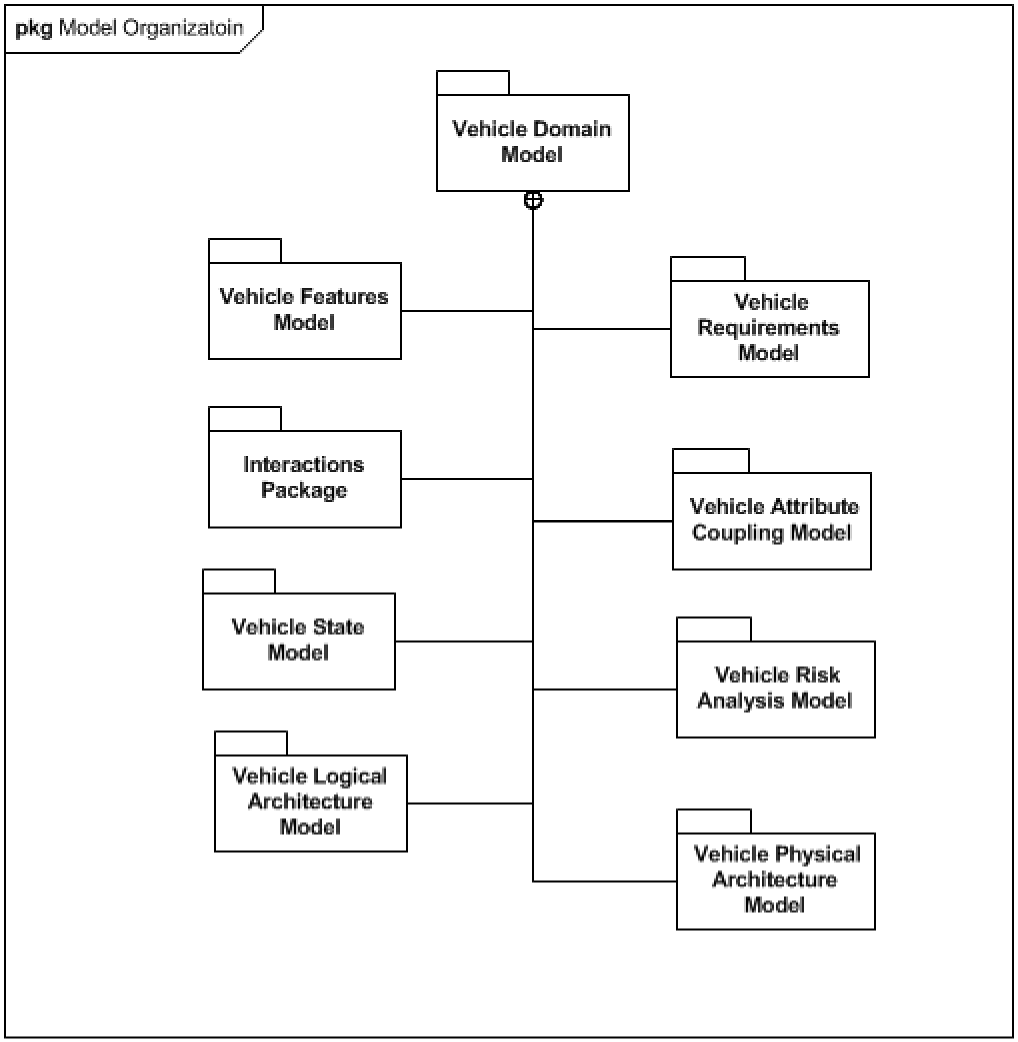
The Vehicle Pattern: Extracts from an example S*Pattern

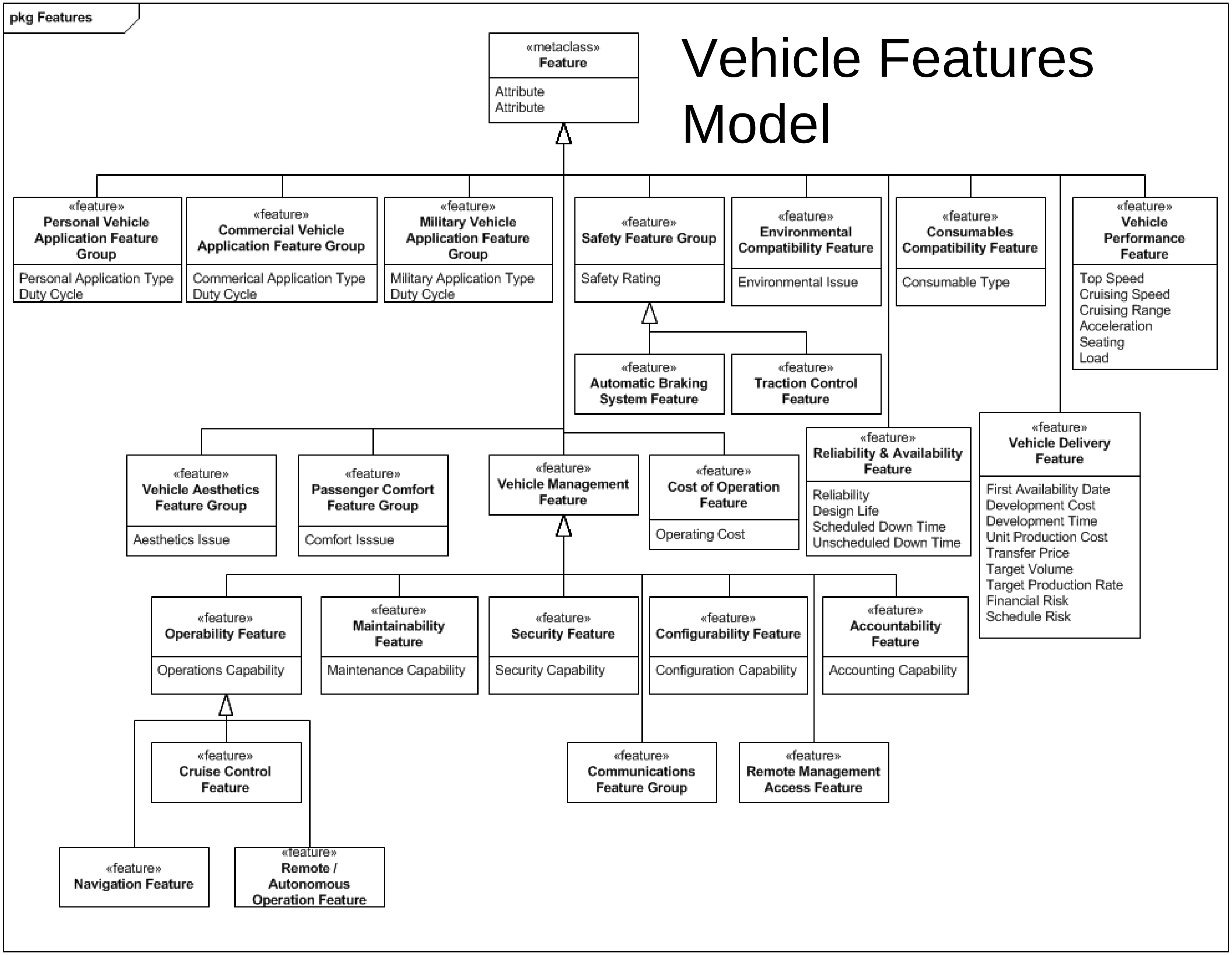




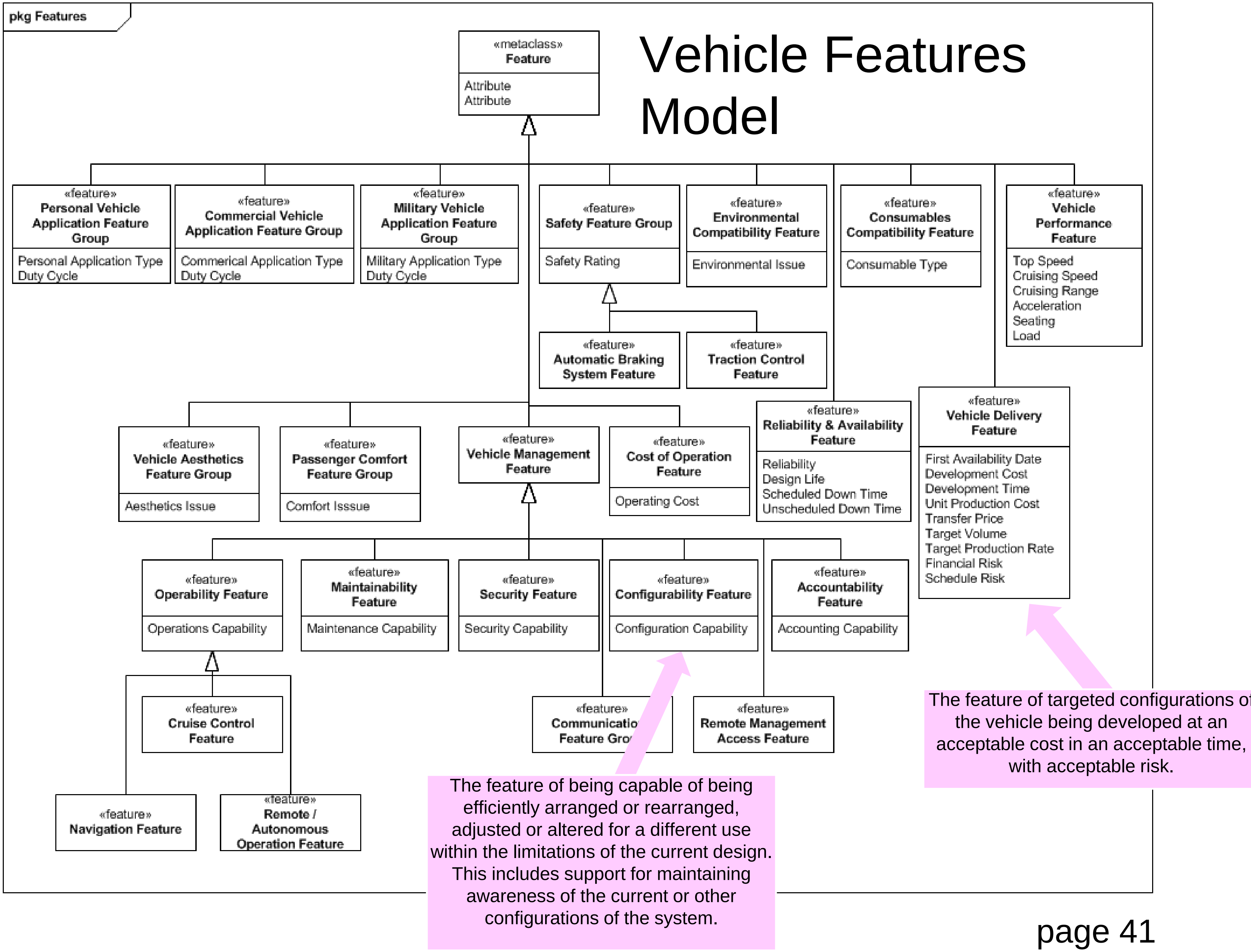


Vehicle Pattern: Model Organization (Packages)

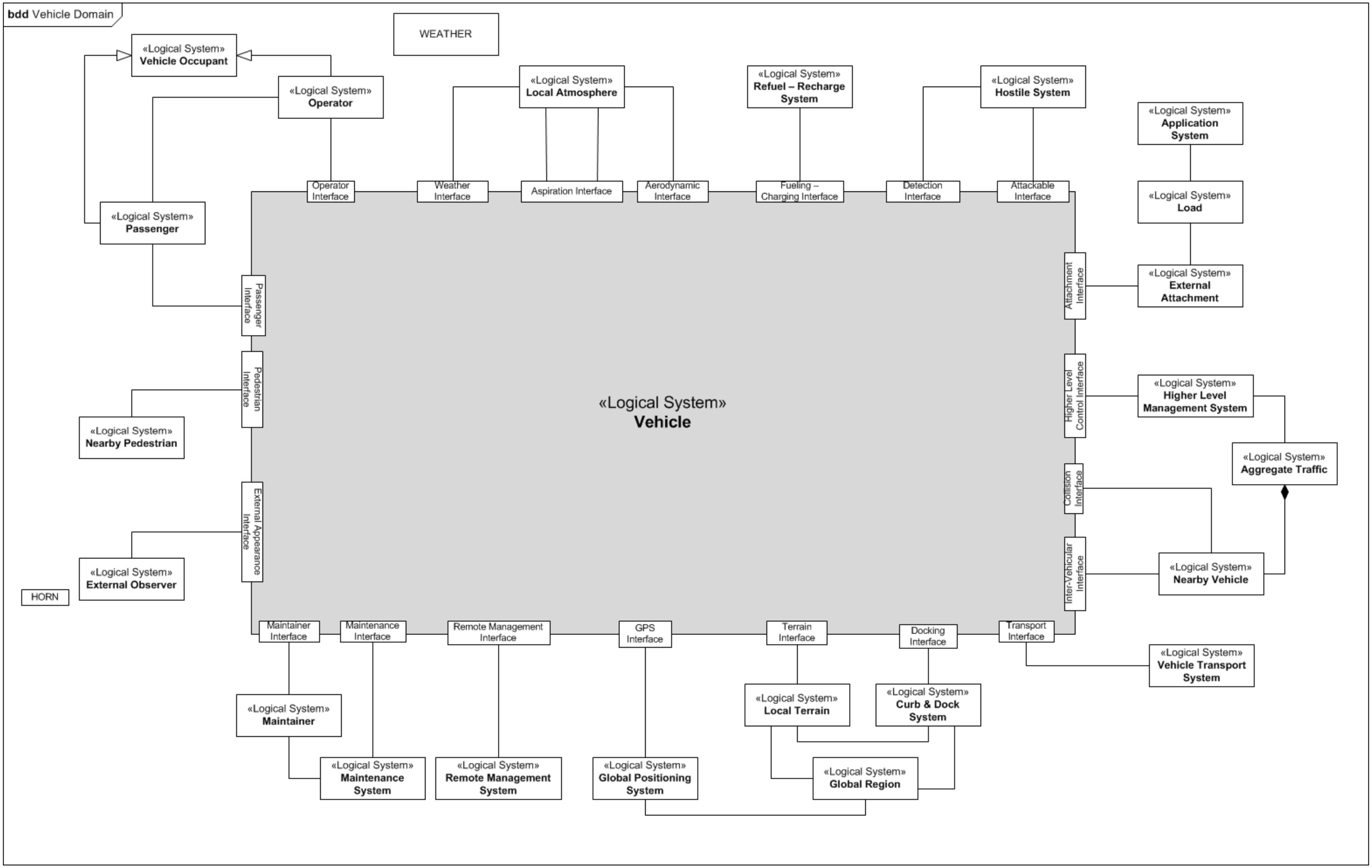



pkg Features
Vehicle Features Model
«metaclass» Feature
Attribute
Attribute
«feature» Personal Vehicle Application Feature Group
Personal Application Type
Duty Cycle
«feature» Commercial Vehicle Application Feature Group
Commerical Application Type
Duty Cycle
«feature» Military Vehicle Application Feature Group
Military Application Type
Duty Cycle
«feature» Safety Feature Group
Safety Rating
«feature» Environmental Compatibility Feature
Environmental Issue
«feature» Consumables Compatibility Feature
Consumable Type
«feature» Vehicle Performance Feature
Top Speed
Cruising Speed
Cruising Range
Acceleration
Seating
Load
«feature» Automatic Braking System Feature
«feature» Traction Control Feature
«feature» Vehicle Aesthetics Feature Group
Aesthetics Issue
«feature» Passenger Comfort Feature Group
Comfort Isssue
«feature» Vehicle Management Feature
«feature» Cost of Operation Feature
Operating Cost
«feature» Reliability & Availability Feature
Reliability
Design Life
Scheduled Down Time
Unscheduled Down Time
«feature» Vehicle Delivery Feature
First Availability Date
Development Cost
Development Time
Unit Production Cost
Transfer Price
Target Volume
Target Production Rate
Financial Risk
Schedule Risk
«feature» Operability Feature
Operations Capability
«feature» Maintainability Feature
Maintenance Capability
«feature» Security Feature
Security Capability
«feature» Configurability Feature
Configuration Capability
«feature» Accountability Feature
Accounting Capability
«feature» Cruise Control Feature
«feature» Communications Feature Group
«feature» Remote Management Access Feature
«feature» Navigation Feature
«feature» Remote / Autonomous Operation Feature

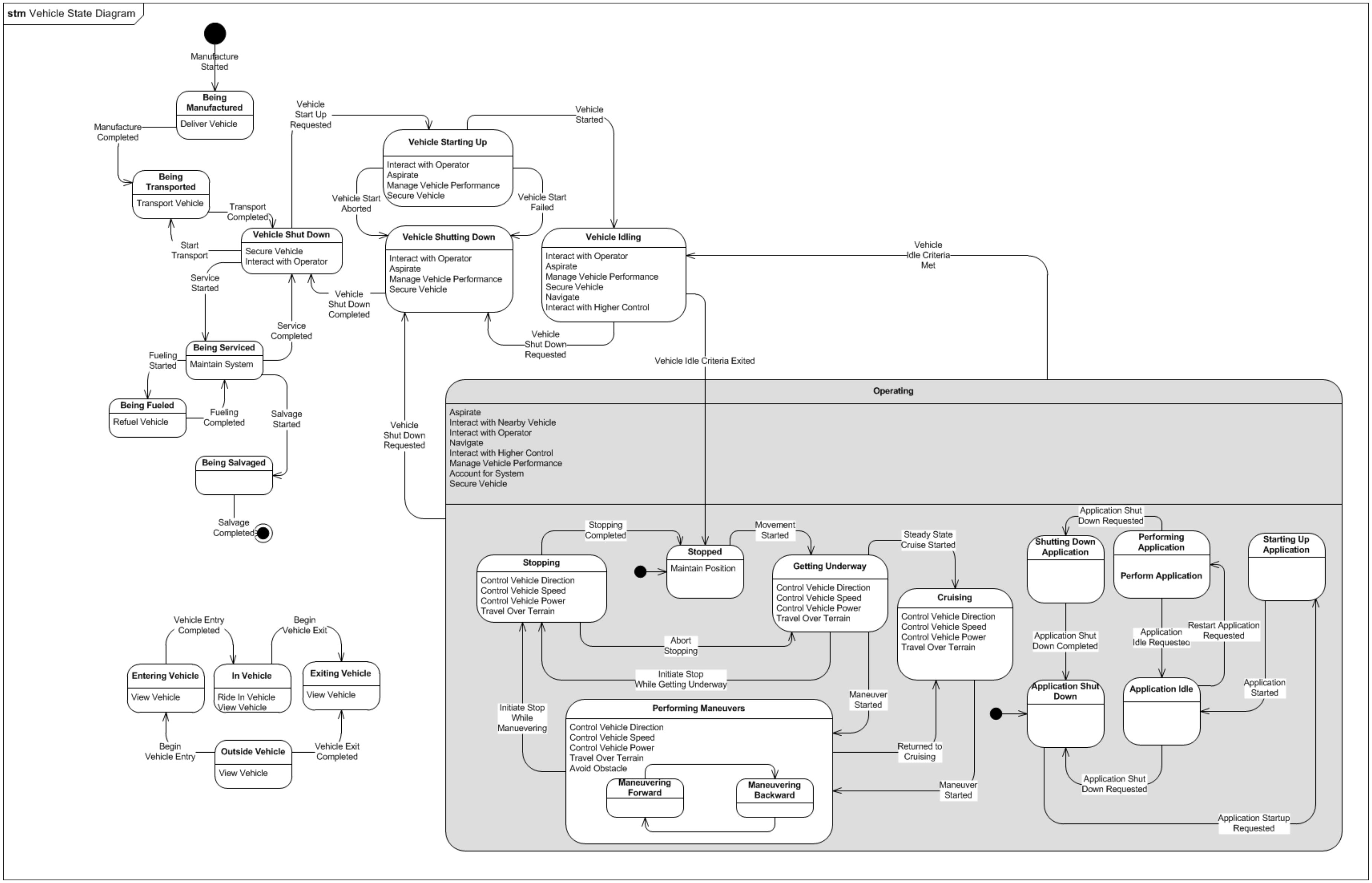
Vehicle Features Model



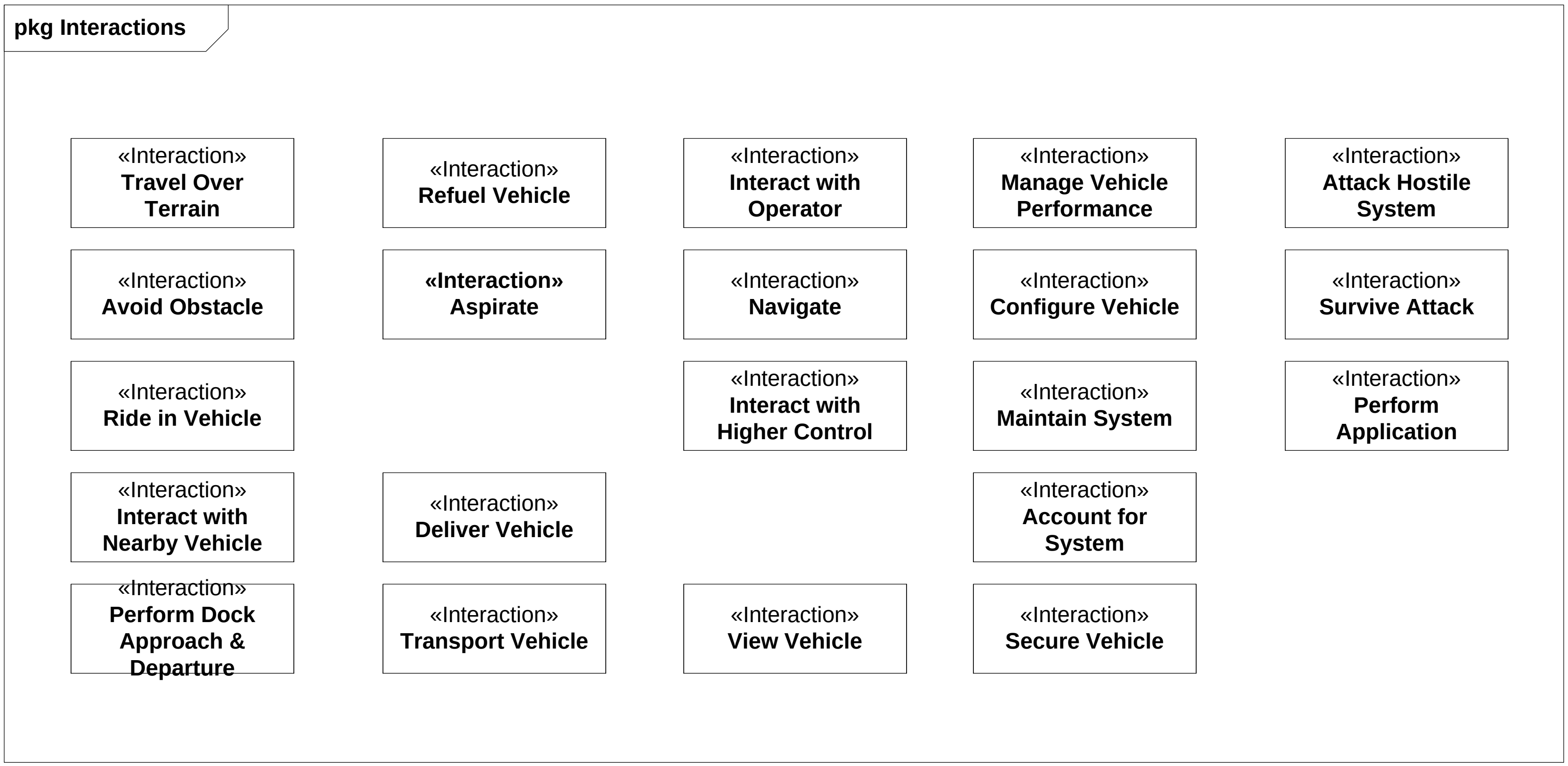
Vehicle Domain Model



Vehicle State Model

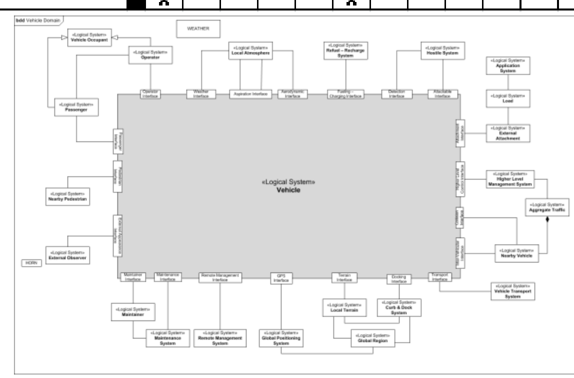


Vehicle Interaction Model

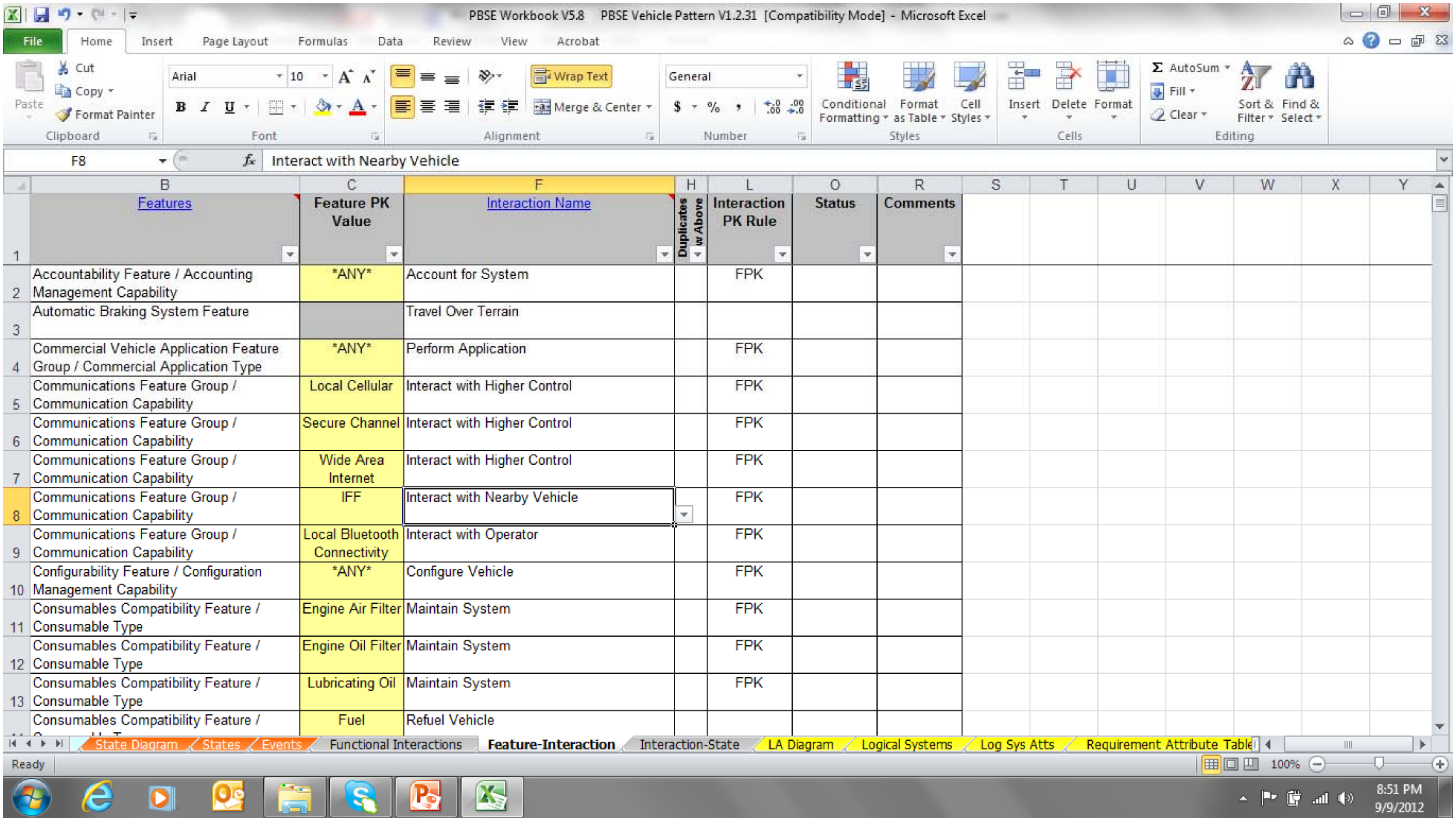


Vehicle Interactions:
Which Actors Participate in Interaction?

1			Actors																					
2	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
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																

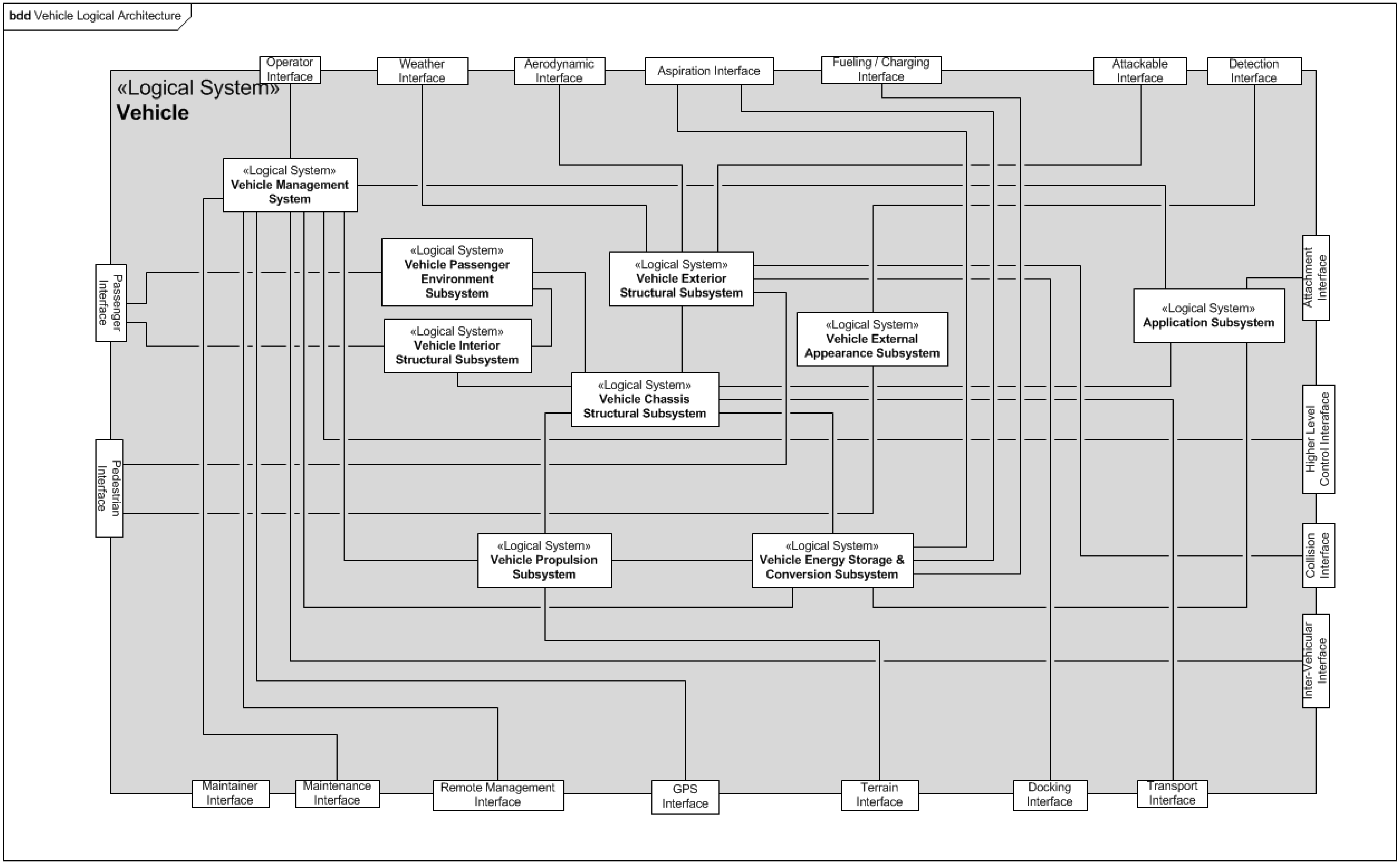


Vehicle Feature-Interaction Associations

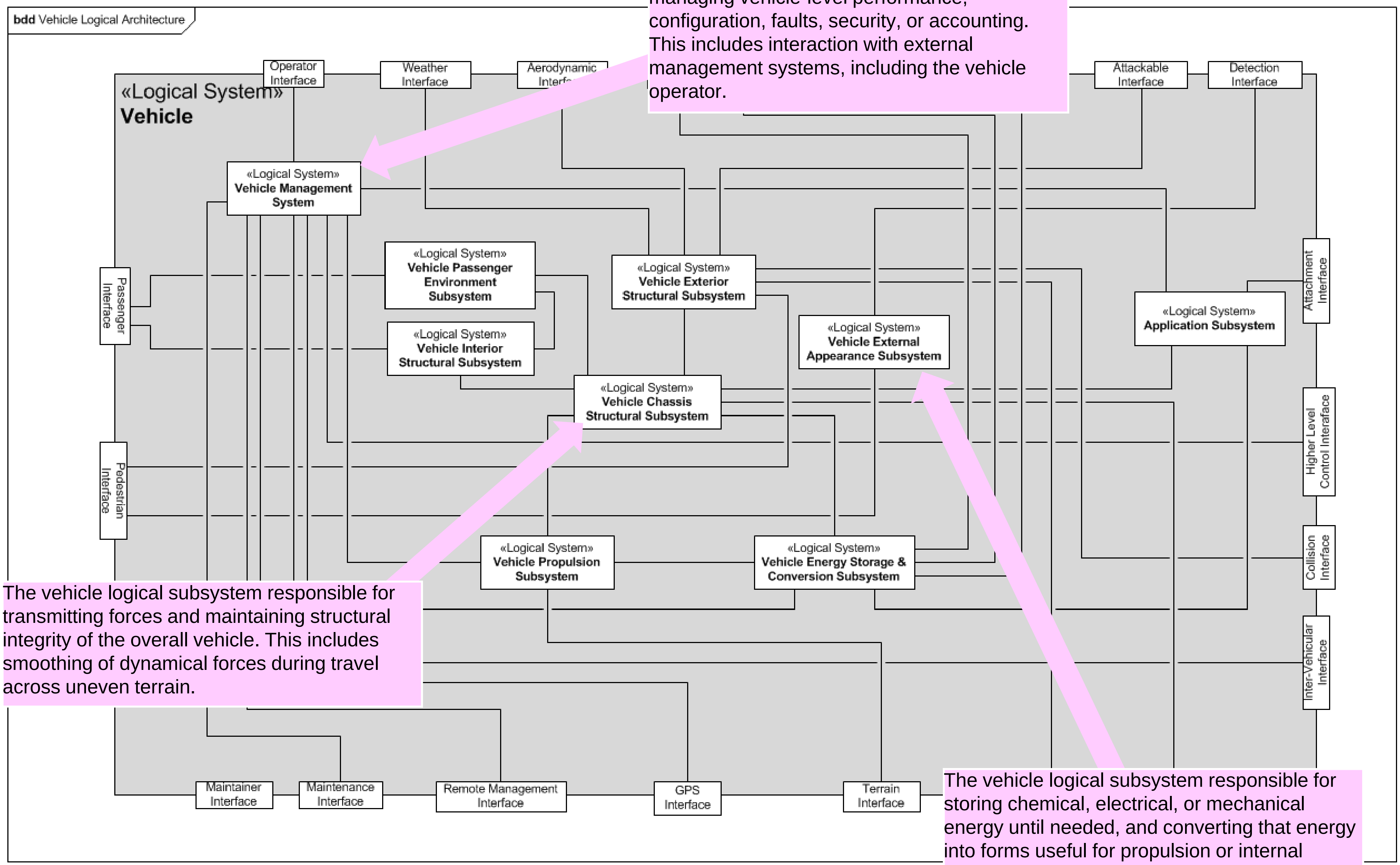


	B	C	F	H	L	O	R
1	Features	Feature PK Value	Interaction Name	Duplicates w Above	Interaction PK Rule	Status	Comments
2	Accountability Feature / Accounting Management Capability	*ANY*	Account for System		FPK		
3	Automatic Braking System Feature		Travel Over Terrain				
4	Commercial Vehicle Application Feature 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				

Logical Architecture Model



Logical Architecture Model

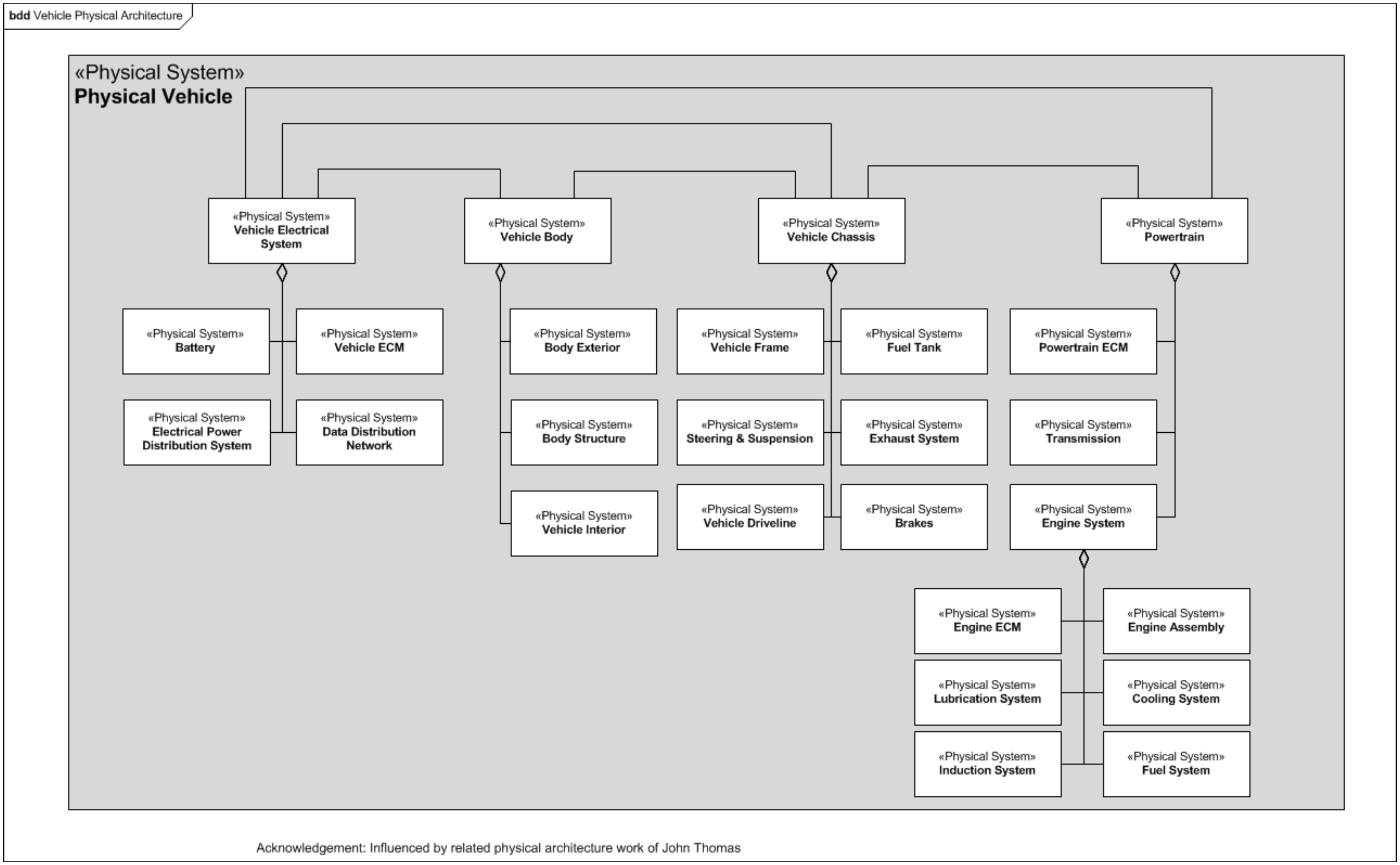


The vehicle logical subsystem responsible for managing vehicle-level performance, configuration, faults, security, or accounting. This includes interaction with external management systems, including the vehicle operator.

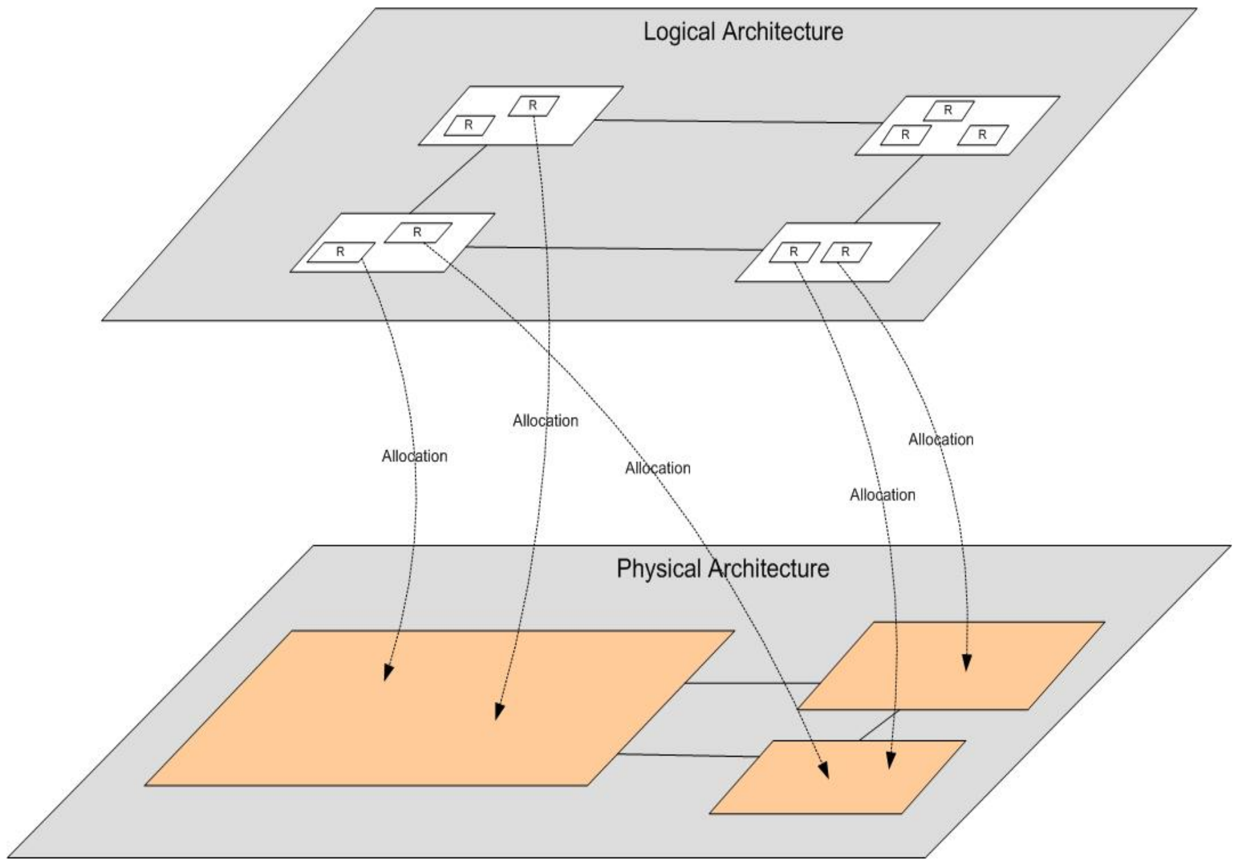
The vehicle logical subsystem responsible for transmitting forces and maintaining structural integrity of the overall vehicle. This includes smoothing of dynamical forces during travel across uneven terrain.

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

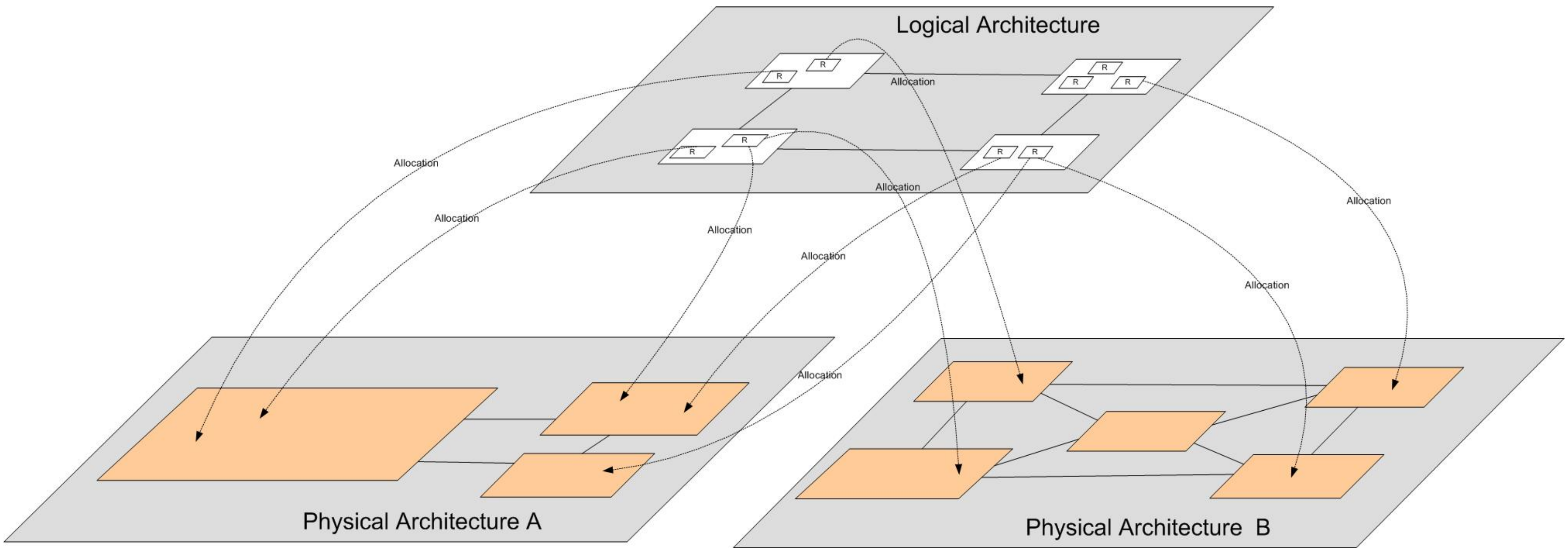


Allocation of Logical Roles to Physical Architecture

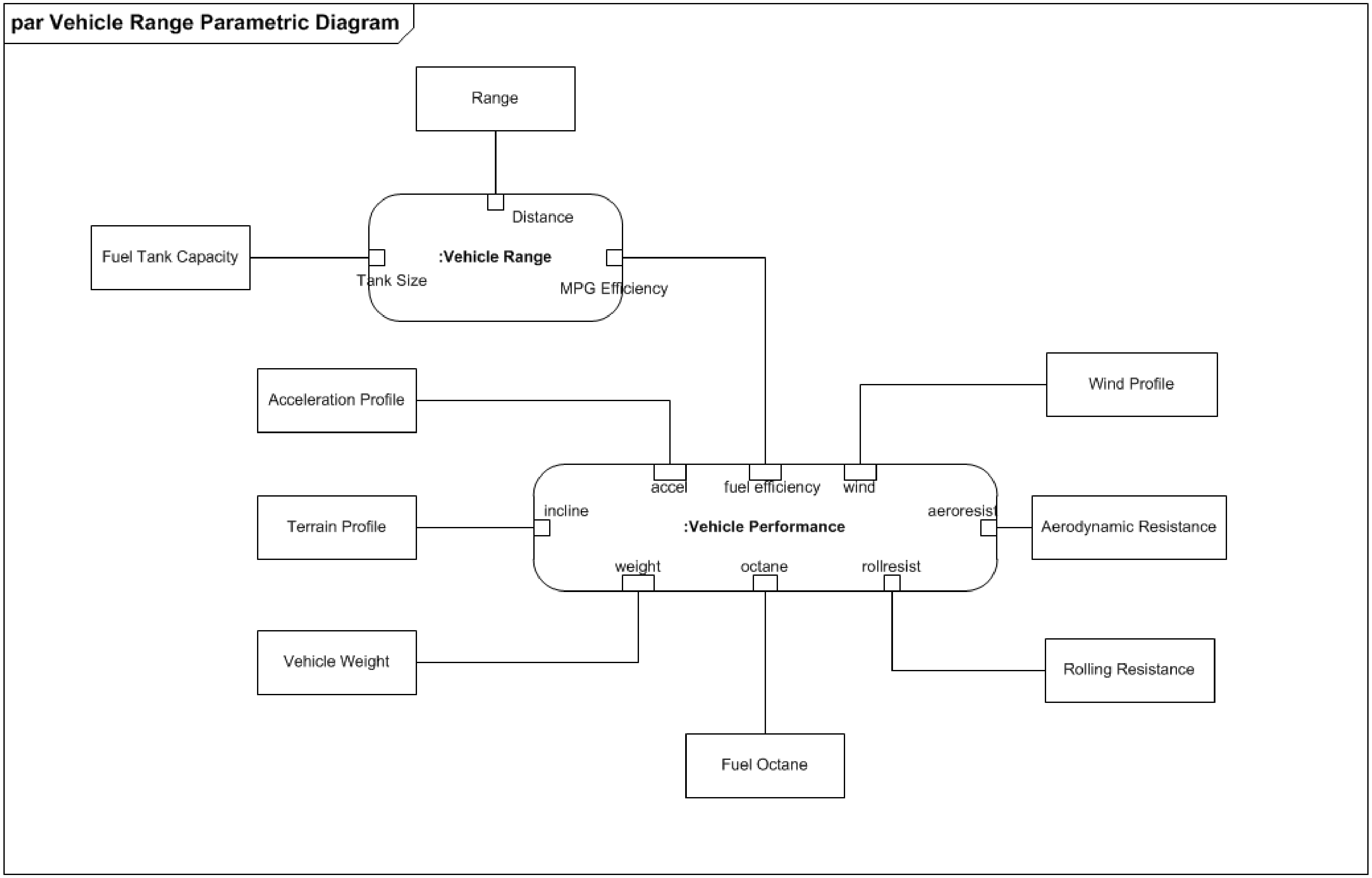


Allocation of Logical Roles to Physical Architecture

- Same Logical Architecture covers many Physical Architectures:

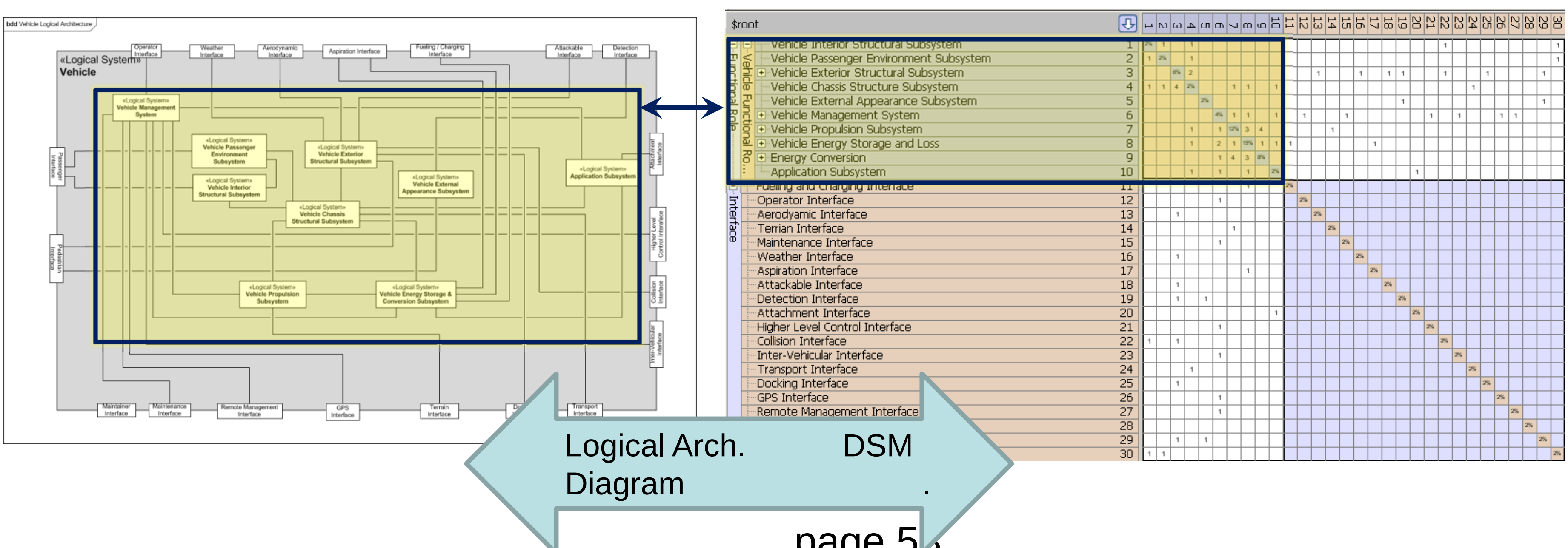


Attribute Coupling Model



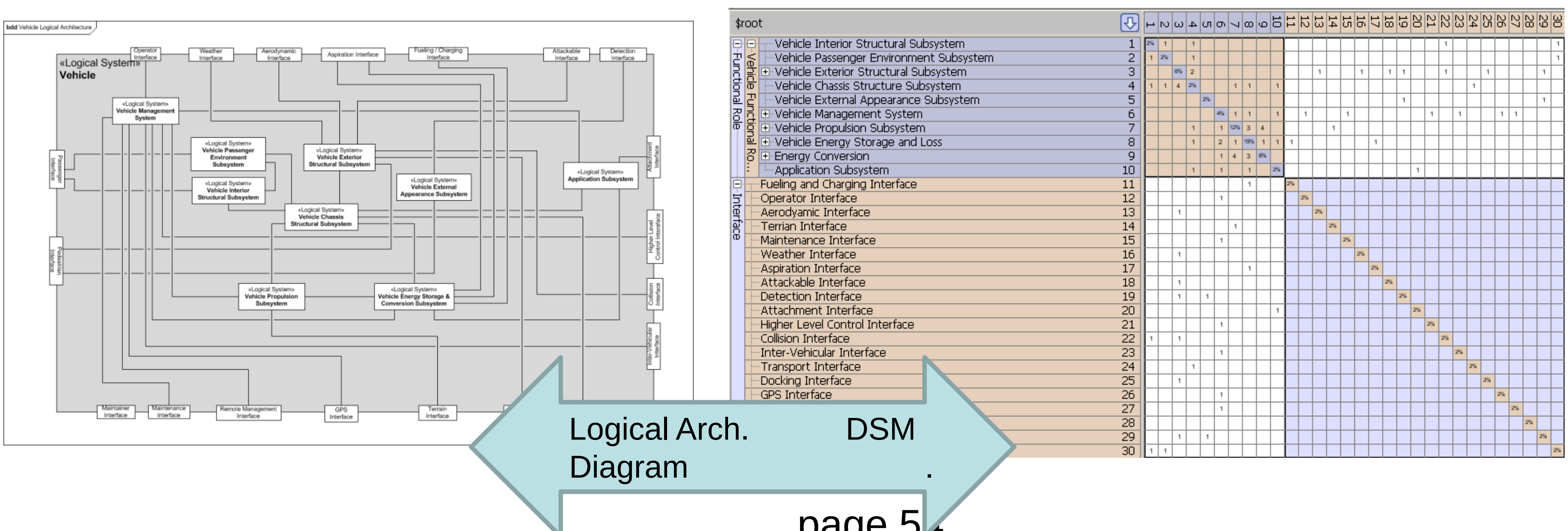
Logical Architecture Views
Block Diagram and Design Structure Matrix (DSM)

- 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)

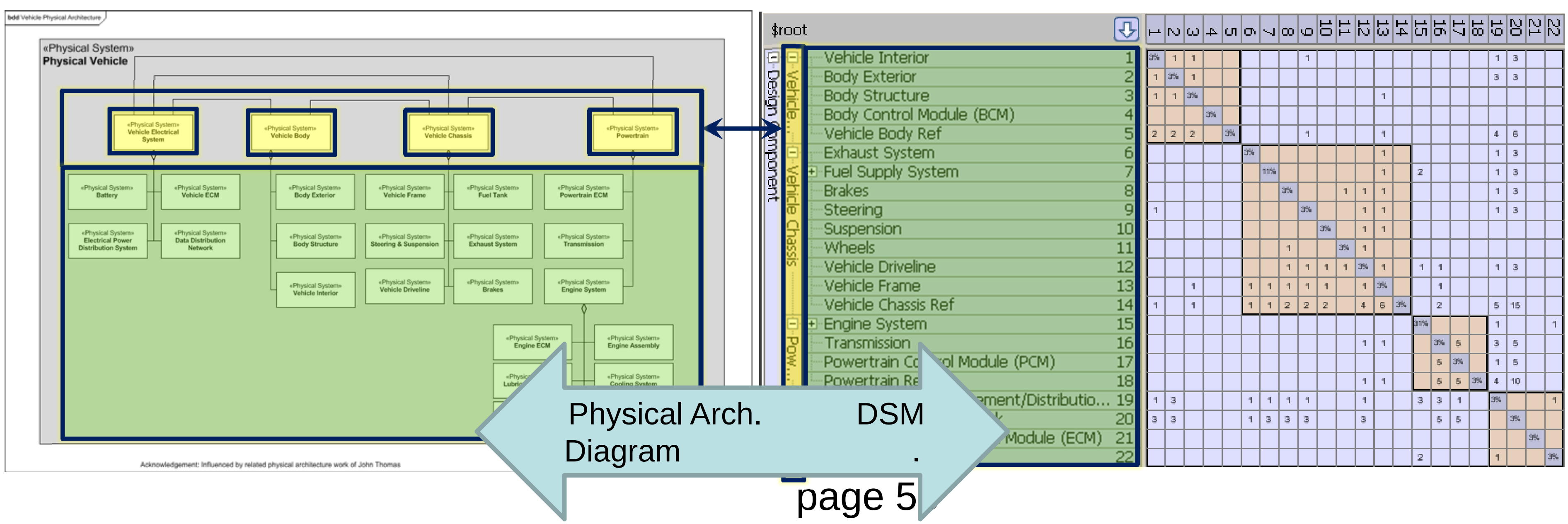
- In the case of Logical Architecture:
  - The blocks in the LA diagram become rows and columns of the DSM
  - The connection lines in the LA diagram become marks in the DSM
- Both views are visualizations of the same information:
  - However the functionality has been partitioned into interacting subsets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



Physical Architecture Views

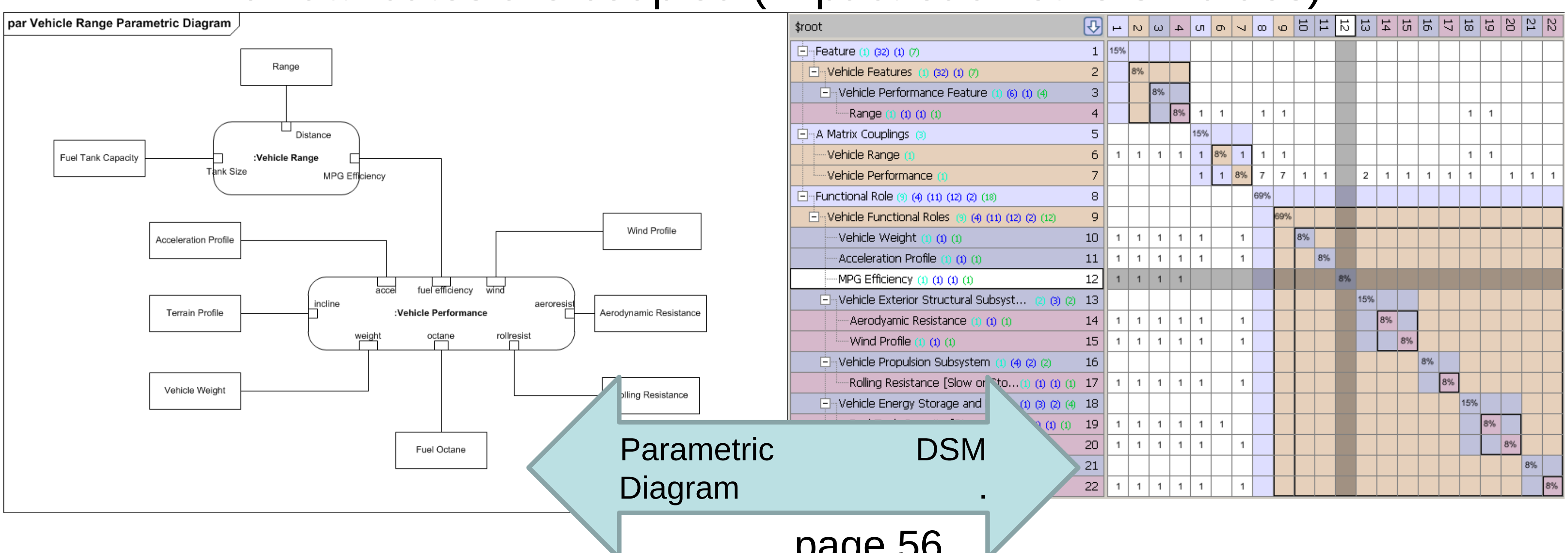
Block Diagram and Design Structure Matrix (DSM)

- 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



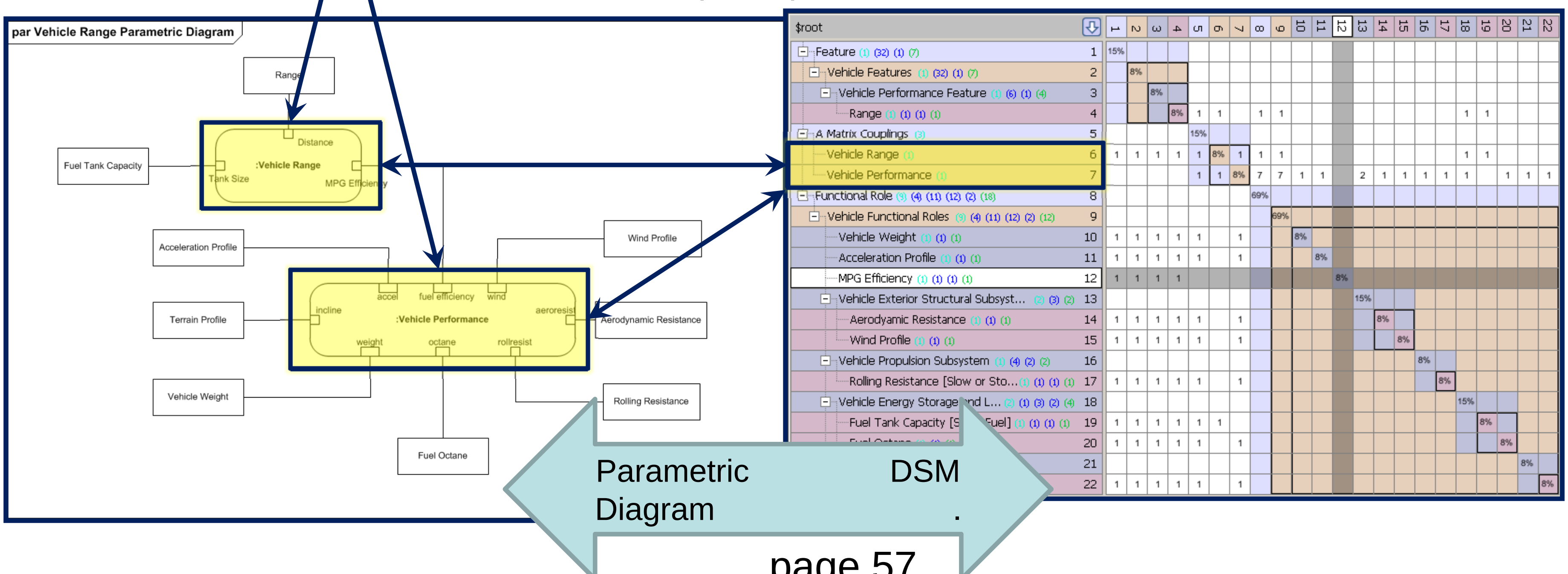
Domain Structure Matrix (DSM) View of Same

- 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)

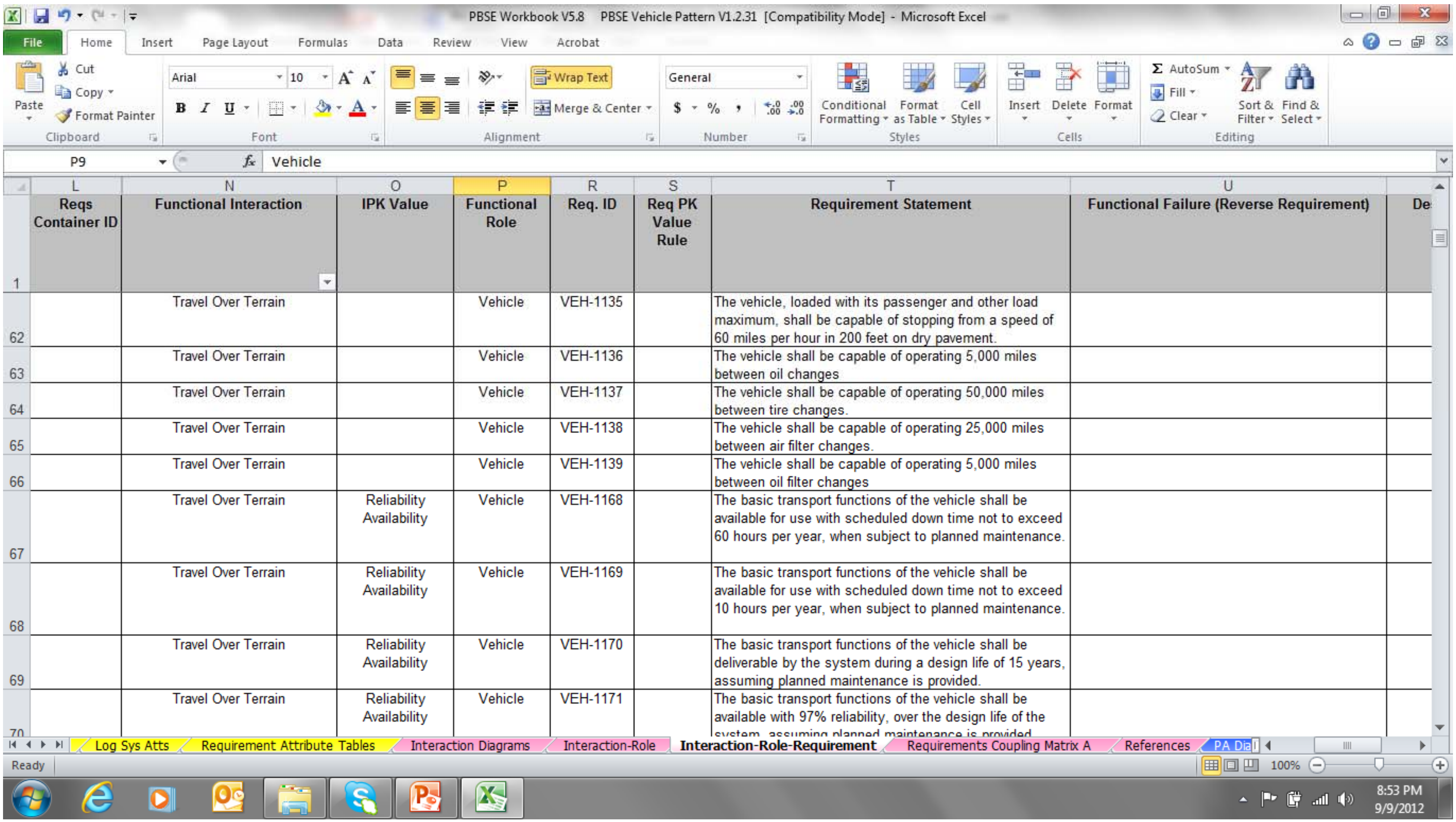


Domain Structure Matrix (DSM) View of Same

- 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

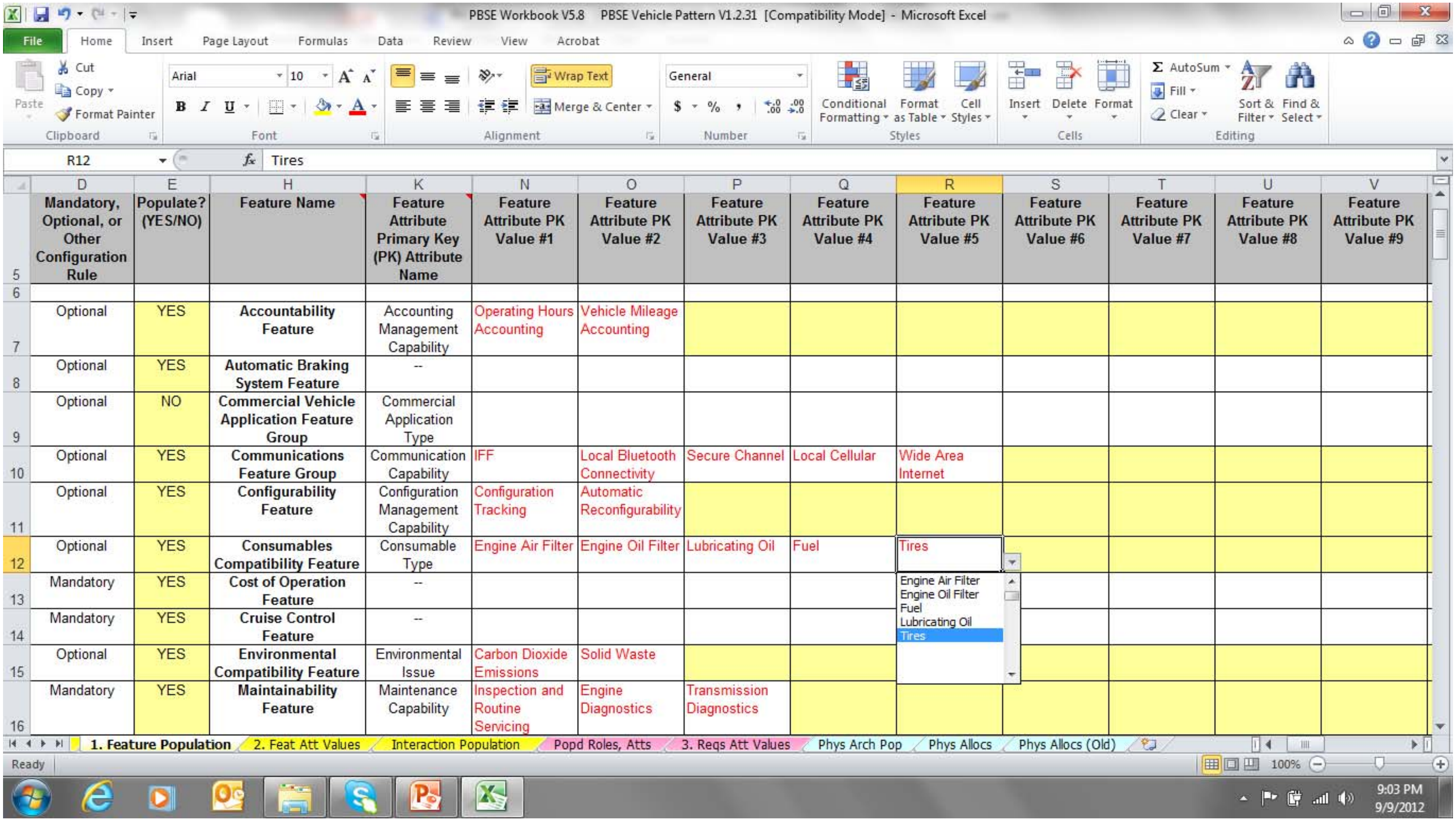


	Reqs Container ID	Functional Interaction	IPK Value	Functional Role	Req. ID	Req PK Value Rule	Requirement Statement	Functional Failure (Reverse Requirement)
62		Travel Over Terrain		Vehicle	VEH-1135		The vehicle, loaded with its passenger and other load maximum, shall be capable of stopping from a speed of 60 miles per hour in 200 feet on dry pavement.	
63		Travel Over Terrain		Vehicle	VEH-1136		The vehicle shall be capable of operating 5,000 miles between oil changes	
64		Travel Over Terrain		Vehicle	VEH-1137		The vehicle shall be capable of operating 50,000 miles between tire changes.	
65		Travel Over Terrain		Vehicle	VEH-1138		The vehicle shall be capable of operating 25,000 miles between air filter changes.	
66		Travel Over Terrain		Vehicle	VEH-1139		The vehicle shall be capable of operating 5,000 miles between oil filter changes	
67		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.	
68		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.	
69		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.	
70		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

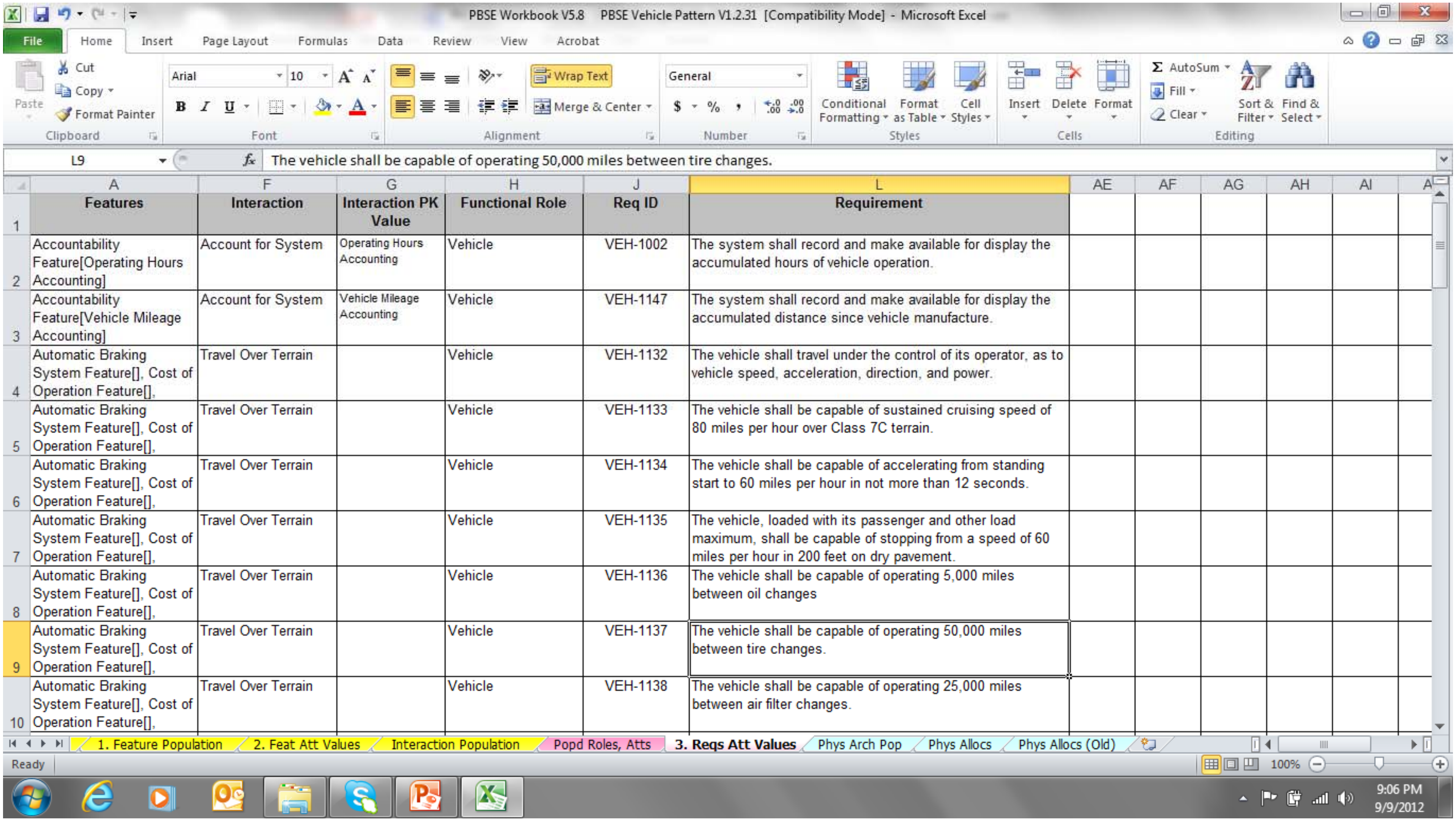
Physical Entity	Failure Mode
Vehicle ECM	Dead ECM
Vehicle ECM	Network Connector Open
Vehicle ECM	Network Connector Short
Vehicle ECM	Erratic ECM
Battery	Discharged Battery
Battery	Battery Cell Short
Battery	Battery Cell Open
Battery	Battery Leak
Panel Display	Fractured Display
Panel Display	Illuminator 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.