INCOSE/OMG MBSE Initiative
PBSE Patterns Challenge Team

Meetings: July 12 & 13, 2015 (during IS2015)

Team web site: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns

Our MBSE Patterns Challenge Team will meet twice during IS2015, on site in Seattle and on line:

- **Sunday, July 12 (15:00-17:00 Pacific Time) and**
- **Monday, July 13 (13:30 - 15:00 Pacific Time)**

We expect to move through different content on these two days, although there could be some repetition where it is of value to our membership.

Agenda . . .

Meeting(s) Agenda: INCOSE PBSE Patterns Challenge Team of MBSE Initiative (at IS2015 and remote web conferenced)

- **Sunday, July 12, 2015 15:00-17:00 PST, IS2015 on site location: Laurel Room**
- **Monday, July 13, 2015 13:30-15:00 PST, IS2015 on site location: Laurel Room**
- **Access for REMOTE participants in EITHER of the above meetings:**
  - **Meeting Web Address: https://incose.pgimeet.com/GlobalmeetTwelve**
  - **Join as GUEST**
  - **Audio--USA /Canada: 1-888-619-1583 (toll free) or 1-719-457-1414 or use computer (other country numbers on separately emailed meeting invitations for whole team)**

Pre-reading and Background: Team web site: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns

Materials of meetings of July 12-13, 2015: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns_challenge_team_mtg_07.12-13.15

(For those interested in catching up on past PBSE material, refer to the May 19 meeting materials: "PBSE Methodology Summary" document, plus meeting slides on:

- *Brief review of HLR (high level requirements framework) portion of S*Metamodel*
- *Criticality of Interactions to the heart of MBSE and PBSE, science and engineering*
- *Viewing Requirements Statements as non-linear Transfer Functions*
- *Gestalt Rules in PBSE – and their connections to the above and applications in understanding system patterns*

Those materials may be found at: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns_challenge_team_mtg_05.19.15)

MEETING #1 OF 2 MEETINGS: Sunday, July 12	15:00 – 17:00 PST
Meeting start up: • Review of meeting objectives and agenda • Introductions of participants—on site and remote • Why the Patterns Challenge Team exists: Goals and approach	15:00 – 15:15
Announcements and updates: • INCOSE Great Lakes Regional Conference (GLRC9) 2015: Cleveland, October 23-25, 2015: https://www.incose.org/newsevents/currentevents/2015/01/14/incose-great-lakes-9th-regional-conference-2015-(glrc9) • Look for five or our team's papers at IS2015, Seattle, July: Pickard (best paper award); Cook; Peterson; Sanyal; Schindel • Agile SE Life Cycle Model (ASELCM) Project (joint w/Agile WG) host enterprise workshops to begin August; five orgs in pipeline • Our team's co-chair, Troy Peterson, named INCOSE Asst. Director for SE Transformation to MBSE • Status of our team's PBSE Methodology Summary posting by John Watson, INCOSE • Other announcements or updates?	15:15 – 15:25
PBSE technical subjects—discussion of: • What is in an S*Pattern, and why? • Our team's "Wave 1 projects" IS2015 authors (if present)—brief summary of their papers to be presented this week: ○ Improving Automated Test of Safety Critical Aerospace Systems -- Dave Cook, Moog Aircraft, et al ○ Reducing Error Escapes in the Development Process – Andy Pickard, Rolls Royce, et al ○ Autonomous Vehicle Pattern – Troy Peterson, Booz Allen Hamilton, et al ○ Improving Product Life Cycle Management – Saumya Sanyal, K2 Firm, et al ○ Representing and Improving System Life Cycle Trajectories – Bill Schindel, ICTT System Sciences	15:25 – 16:30

- Our team's "Wave 2 project" leaders—brief summary of their work now underway, and participation interest from others:
  - Systems Engineering Social Network Pattern – Chris Hoffman, Cummins, Inc.
  - Health Care High Fidelity Transcription System Pattern – Vijay Thukral, Cientive Group
  - Improving the Connection to Systems Value – Troy Peterson, Booz Allen Hamilton, et al
  - Agile Systems Engineering Life Cycle Model Project & Pattern – Bill Schindel, ASELCM Team, and Agile Systems WG
  - Strengthening Metamodel Support for MBSE: The Systems Phenomenon – Bill Schindel

(*For those interested in catching up on past PBSE material, refer to the May 19 team meeting materials on:*
"PBSE Methodology Summary" document, plus meeting slides on:

- *Brief review of HLR (high level requirements framework) portion of S*Metamodel*
- *Criticality of Interactions to the heart of MBSE and PBSE, science and engineering*
- *Viewing Requirements Statements as non-linear Transfer Functions*
- *Gestalt Rules in PBSE – and their connections to the above and applications in understanding system patterns*

Those materials may be found at: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns_challenge_team_mtg_05.19.15)

15:25 – 16:30 cont'd

(Depending on which authors and leaders attend Sunday vs. Monday meeting—will continue in Monday meeting)

Planning discussion:

- Future (Third Wave) Projects Pipeline Candidates:

Mapping PBSE to COTS Tools and Information Systems	Example SOS Pattern (Joint with SoS WG)
Mapping to ISO 15288; Processes vs. Data (Maps vs. Itineraries)	Supporting INCOSE objective for SE model-based; Case for Stronger Model Semantics
PBSE Implementation Strategies	Other interests from team members
Example Product Line Engineering (PLE) Pattern (Joint w/PLE WG)	

- Future meetings schedule: Pace, rate, calendar
- Outreach: Who else should be involved? Example—other INCOSE WGs that are natural Patterns applications. Ideas?
- Adjourn Sunday meeting

16:30 – 17:00

MEETING #2 OF 2 MEETINGS: Monday, July 13 — **13:30 – 15:00 PST**

Meeting start up:

- Review of meeting objectives and agenda; summary of Sunday's meeting outcomes
- Introductions for new participants—on site and remote
- Why the Patterns Challenge Team exists: Goals and approach

13:30 – 13:45

***Reminder* of Sunday's announcements and updates** — 13:45 – 13:55

***Continue* discussion of PBSE technical subjects begun on Sunday:**

- What is in an S*Pattern, and why?
- Our team's "Wave 1 projects" IS2015 authors (if present)—brief summary of their papers to be presented this week:
  - Improving Automated Test of Safety Critical Aerospace Systems -- Dave Cook, Moog Aircraft, et al
  - Reducing Error Escapes in the Development Process – Andy Pickard, Rolls Royce, et al
  - Autonomous Vehicle Pattern – Troy Peterson, Booz Allen Hamilton, et al
  - Improving Product Life Cycle Management – Saumya Sanyal, K2 Firm, et al
  - Representing and Improving System Life Cycle Trajectories – Bill Schindel, ICTT System Sciences
- Our team's "Wave 2 project" leaders—brief summary of their work now underway, and participation interest from others:
  - Systems Engineering Social Network Pattern – Chris Hoffman, Cummins, Inc.
  - Health Care High Fidelity Transcription System Pattern – Vijay Thukral, Cientive Group

13:55 – 14:50

<table>
<tr><td>

Improving the Connection to Systems Value – Troy Peterson, Booz Allen Hamilton, et al
Agile Systems Engineering Life Cycle Model Project & Pattern – Bill Schindel, ASELCM Team, and Agile Systems WG
Strengthening Metamodel Support for MBSE: The Systems Phenomenon – Bill Schindel

</td><td></td></tr>
<tr><td>
<u>Continuation of Sunday's planning discussion</u>:

Future (Third Wave) Projects Pipeline Candidates:

<table>
<tr><td>Mapping PBSE to COTS Tools and Information Systems</td><td>Example SOS Pattern (Joint with SoS WG)</td></tr>
<tr><td>Mapping to ISO 15288; Processes vs. Data (Maps vs. Itineraries)</td><td>Supporting INCOSE objective for SE model-based; Case for Stronger Model Semantics</td></tr>
<tr><td>PBSE Implementation Strategies</td><td>Other interests from team members</td></tr>
<tr><td>Example Product Line Engineering (PLE) Pattern (Joint w/PLE WG)</td><td></td></tr>
</table>

Future meetings schedule: Pace, rate, calendar
Outreach: Who else should be involved? Example—other INCOSE WGs that are natural Patterns applications. Ideas?
Adjourn Monday meeting

</td><td>15:40 – 15:00</td></tr>
</table>

For more information, contact-- Bill Schindel schindel@ictt.com Troy Peterson peterson_troy@bah.com

The MBSE Initiative Patterns Challenge Team: ***<u>Who are we</u>?***

- Our most active members come from across diverse domains:
  - Automotive
  - Advanced Manufacturing
  - Aerospace
  - Consumer Products
  - Defense
  - Health Care, Medical Devices, Pharmaceuticals
  - Others
  - Today's attendees?

- During the last 18 months, over 100 colleagues have participated in Patterns Challenge Team activities:
  - Team meetings, work sessions, and tutorials
  - Construction of system patterns
  - Writing related papers for IS, IW, and regional INCOSE conferences
  - Invited presentations of our team's work to INCOSE chapter meetings

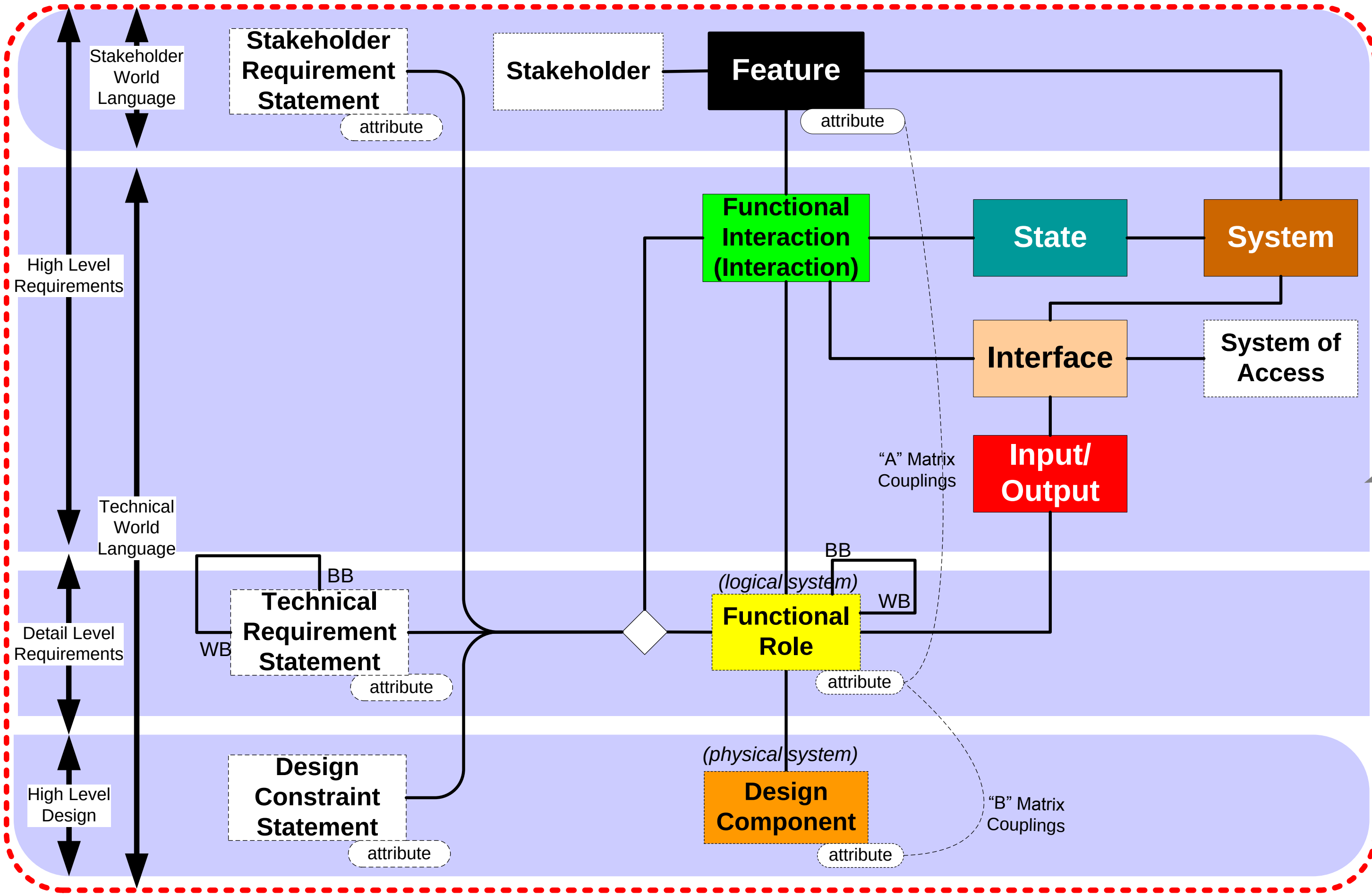
Introductions

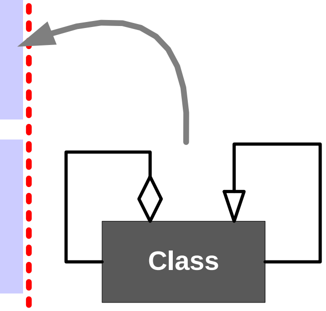
- Introductions of Participants
  - Sunday session
  - Monday session

What does the Patterns Challenge Team *do*?

- **This Challenge Team is concerned with *configurable, re-usable system models*, called "S*Patterns":**
  1. Models containing a certain minimal set of elements are called S*Models (S is short for "Systematica")
  2. Those underlying elements are called the S*Metamodel, which was inspired by the physical sciences
  3. S*Models using those elements may be expressed in any modeling language (e.g., SysML, or other languages)
  4. S*Models can be created and managed in many different COTS modeling tools.
  5. Re-usable, configurable S*Models are called S*Patterns
  6. By "Pattern-Based Systems Engineering" (PBSE) we mean MBSE enhanced by these generalized assets
  7. These are system-level patterns (models of whole managed platforms), not just smaller-scale component design patterns

Summary of some major S*Metamodel classes and relationships—the underlying semantics of all S*Models (Refer to S*Glossary for definitions)

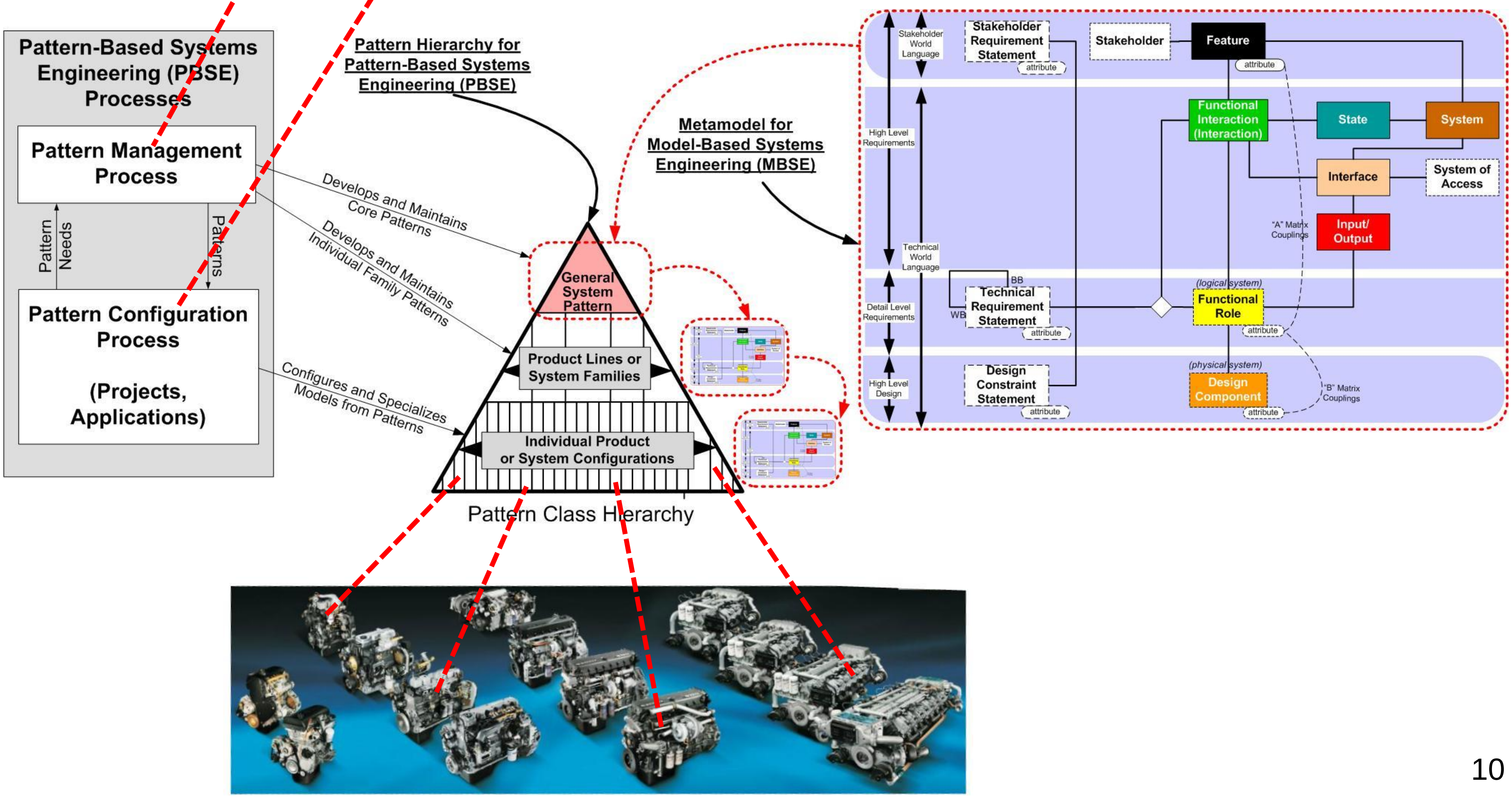




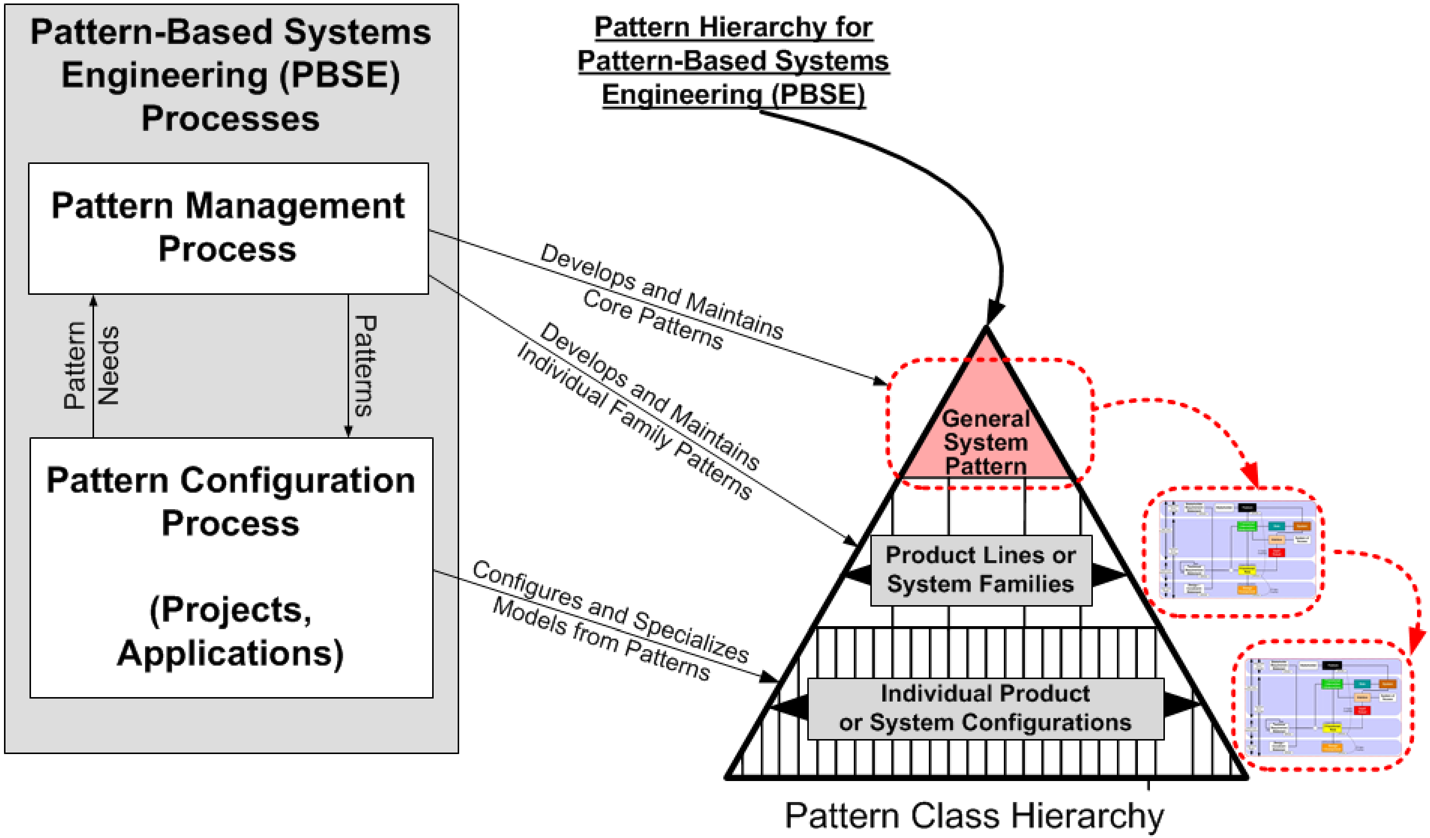
Every S*Metaclass shown is embedded in both an aggregation (whole-part) hierarchy and an abstraction (general-special) hierarchy, connected by the relationship types shown.

Summary of some major S*Metamodel elements

- Pattern-Based Systems Engineering (PBSE) has two overall processes:
  - **Pattern Management Process**: Generates the general pattern, 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.



Business process optimized for PBSE fulfill a different vision:

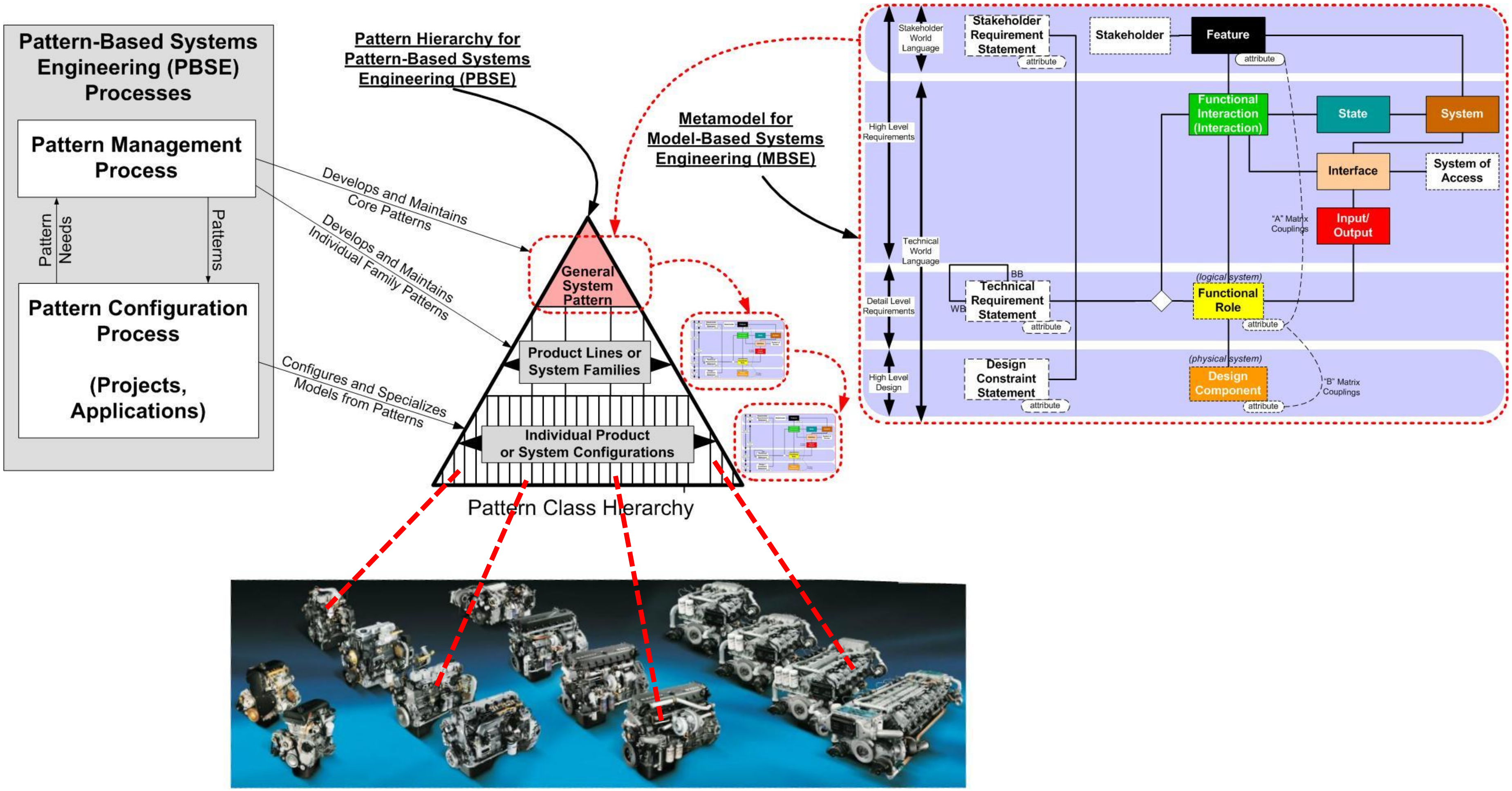


Why do most representations of the systems engineering process appear to assume starting from no formal knowledge about the system of interest & its domain?

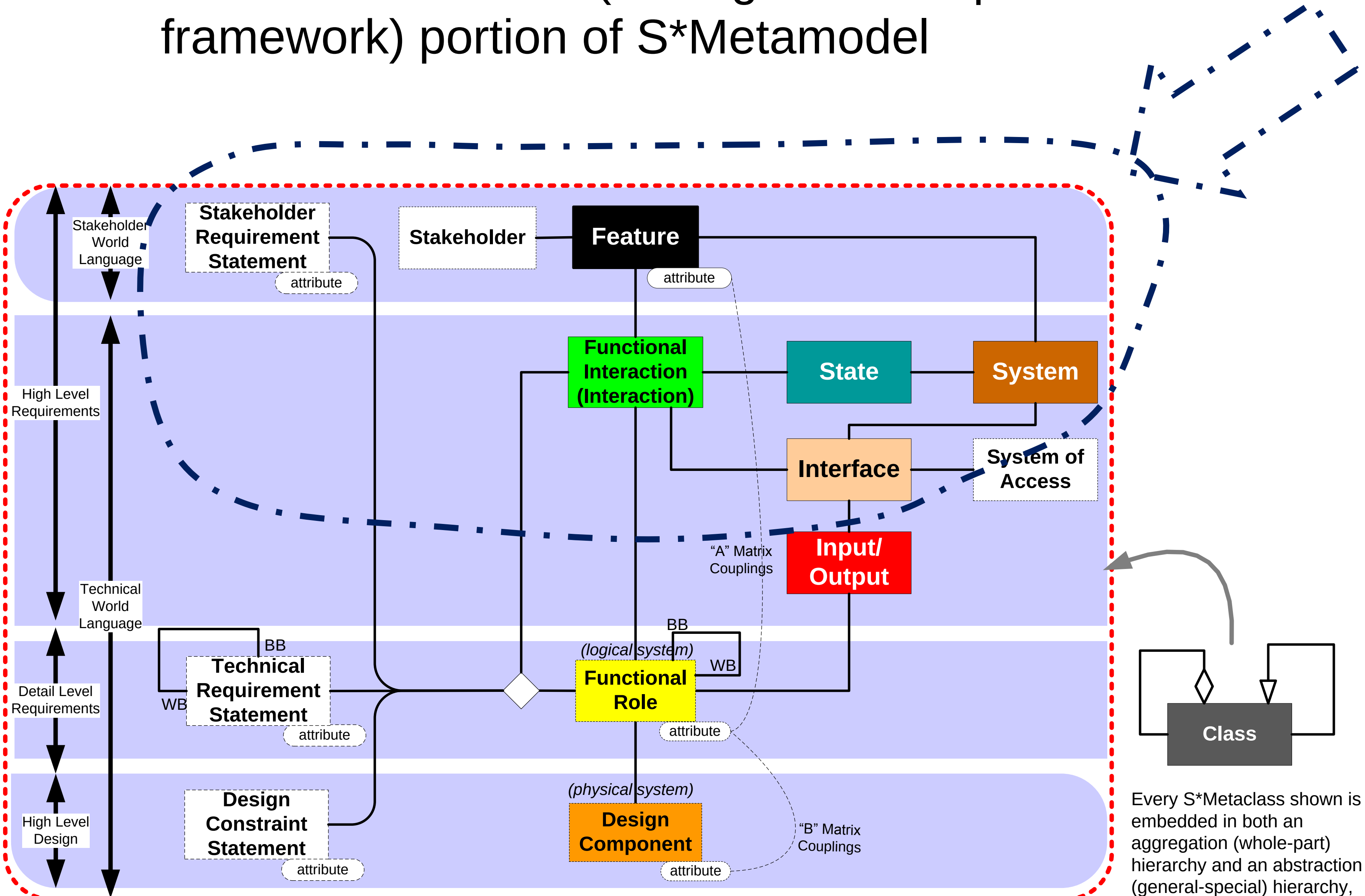
Team Announcements and Updates

- INCOSE Great Lakes Regional Conference (GLRC9) 2015: Cleveland, October 23-25, 2015: https://www.incose.org/newsevents/currentevents/2015/01/14/incose-great-lakes-9th-regional-conference-2015-(glrc9)
  - Will include presentations on S*Patterns from several of our members
- Look for five or our team's papers at IS2015, Seattle: Pickard (best paper award); Cook; Peterson; Sanyal; Schindel
- Agile SE Life Cycle Model (ASELCM) Project (joint w/Agile WG) host enterprise workshops to begin August; five orgs in pipeline (more on this is on agenda for our team's meeting)
- Our team's co-chair, Troy Peterson, named INCOSE Asst. Director for SE Transformation to MBSE
  - More on this Monday; MBSE Teams become formal Working Groups
- Status of our team's PBSE Methodology Summary posting
- Other announcements or updates?

What is in an S*Pattern, and why?



Brief review of HLR (the high level requirements framework) portion of S*Metamodel

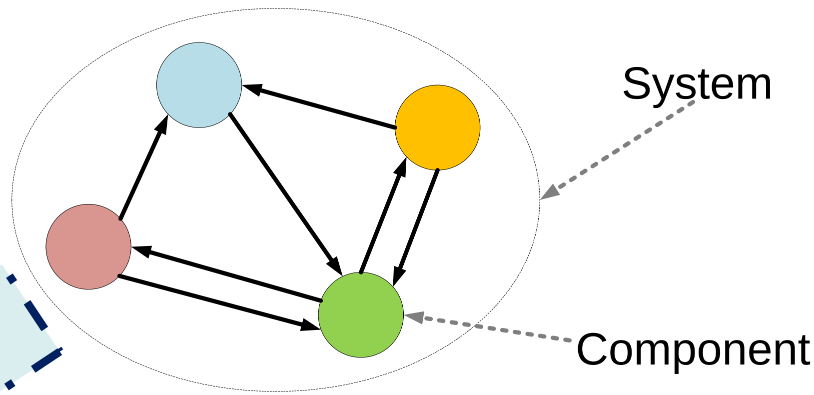


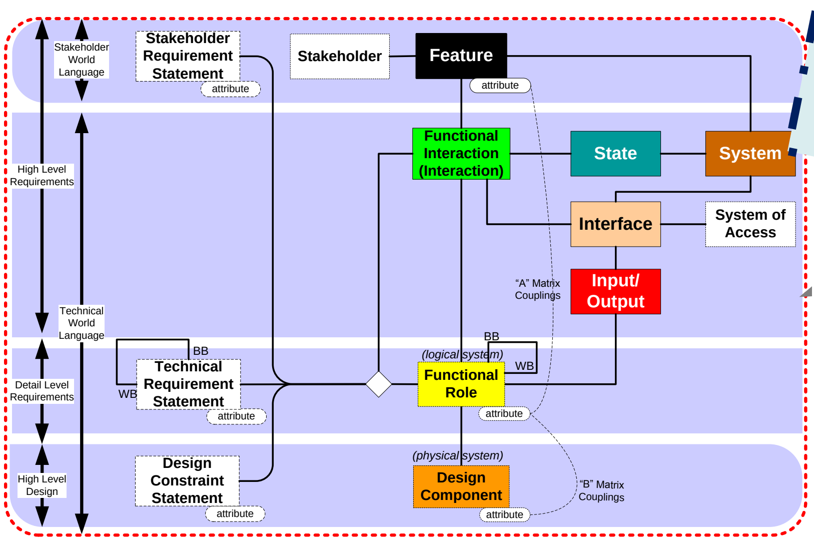
Every S*Metaclass shown is embedded in both an aggregation (whole-part) hierarchy and an abstraction (general-special) hierarchy, connected by the relationship types shown.

- A <u>System</u> is a collection of interacting Components.

- By “interact”, we mean exchanges of energy, force, mass, or information, so that one component changes the state of another component.

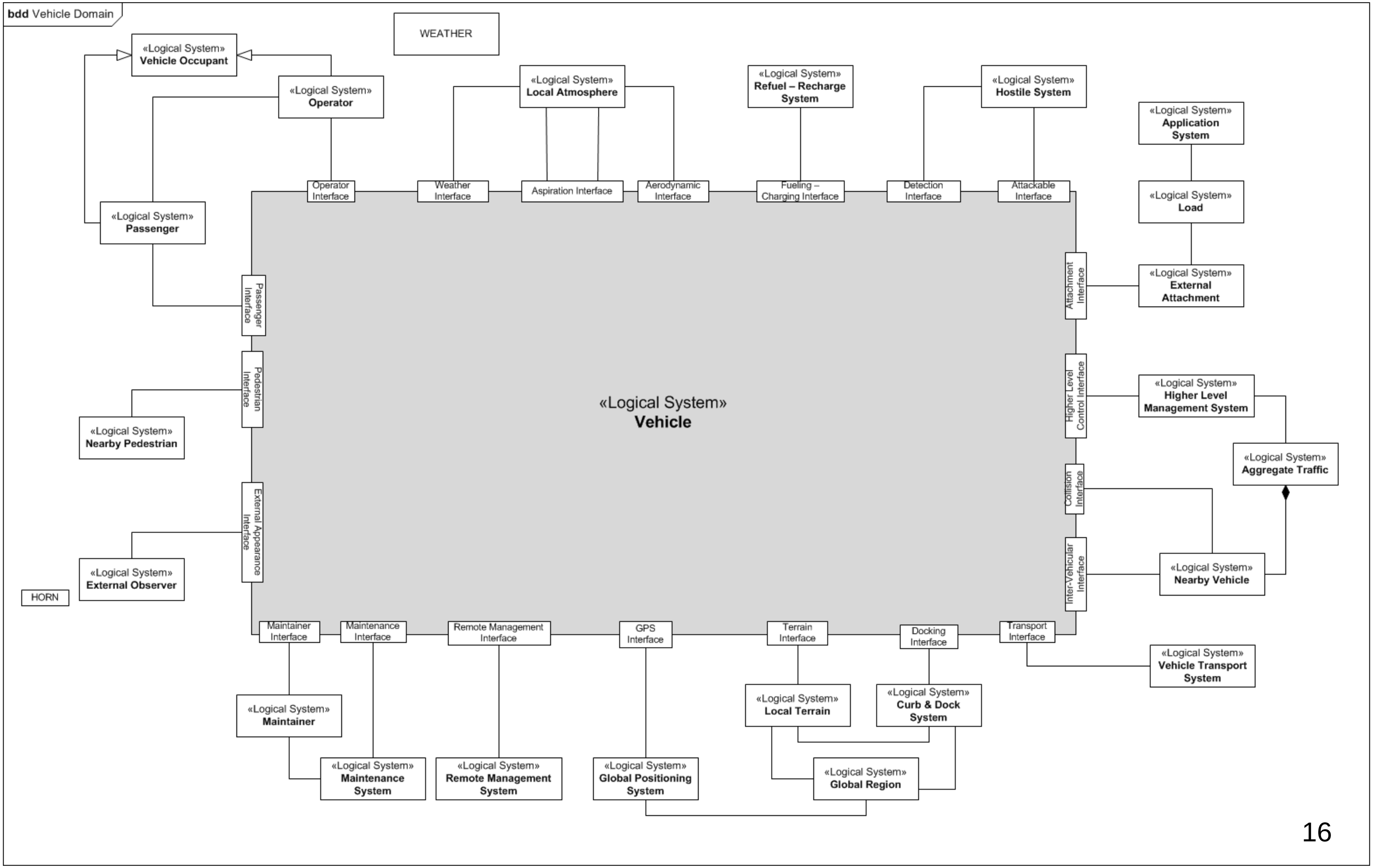
- A Component can be a System.



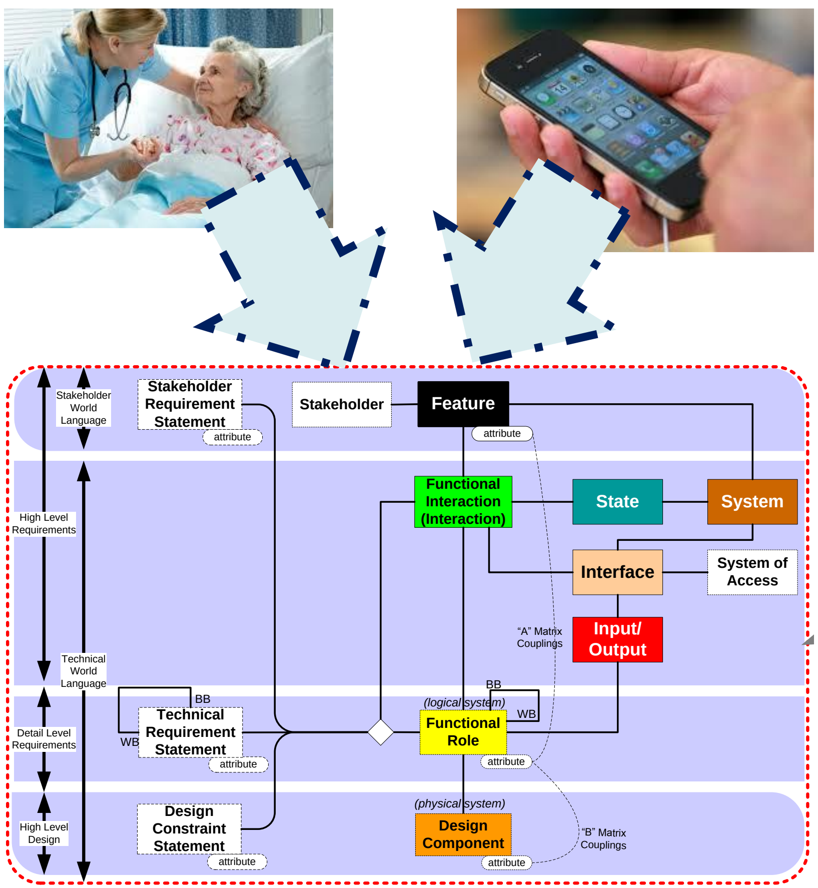


Systems: Engine, Vehicle, Manufacturing Line, Medical Device, Consumer Product, Aircraft, Engine, etc.

In S*Models, Domain Systems are described by Domain Diagrams (showing interacting components of the domain)



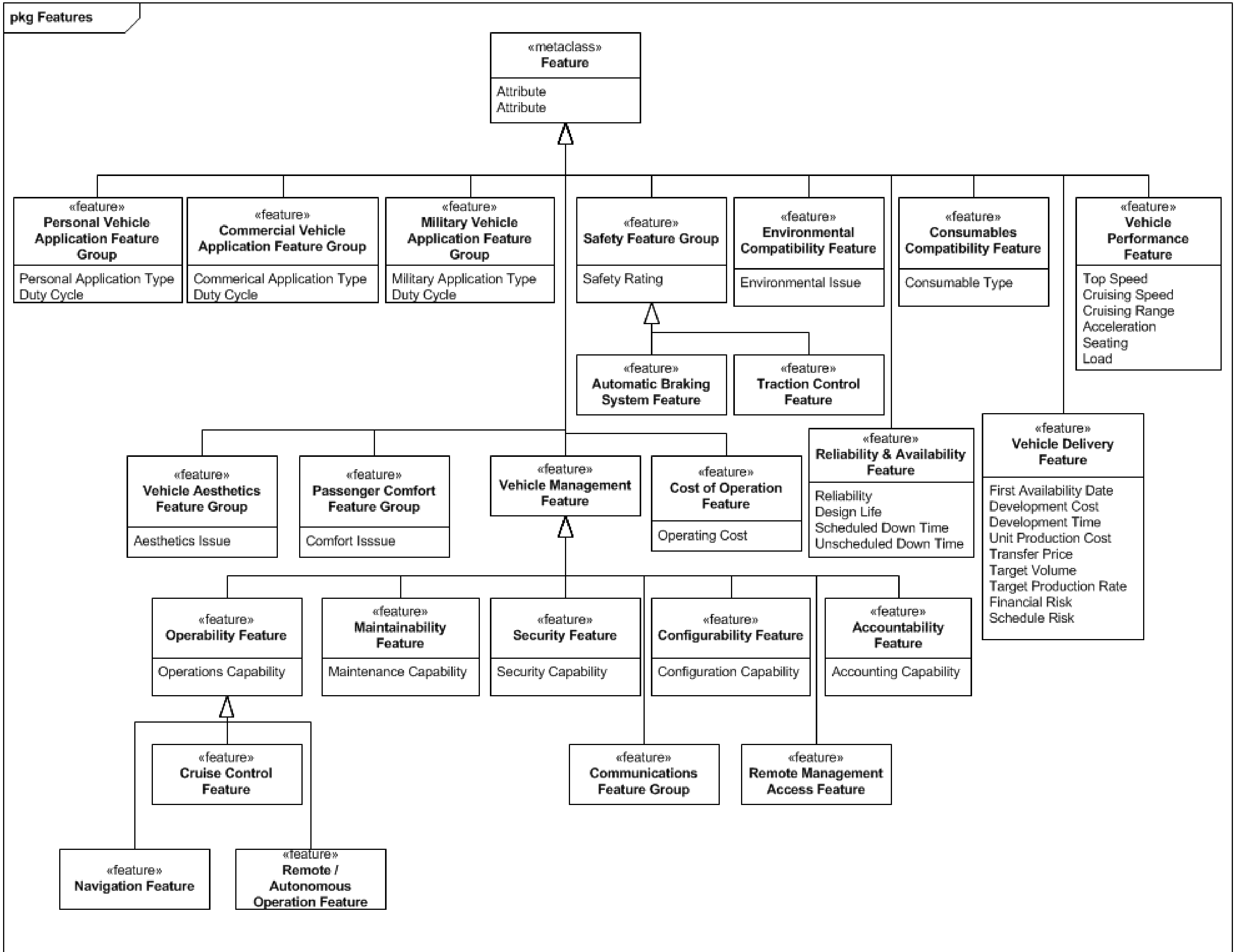
- A Stakeholder is a person, organization, community, or other entity with a stake in the behavior of a system.
- A Feature is a system behavior or capability having value to a Stakeholder, described in Stakeholder concepts & language.
- Features are the basis of Stakeholder selection of systems.



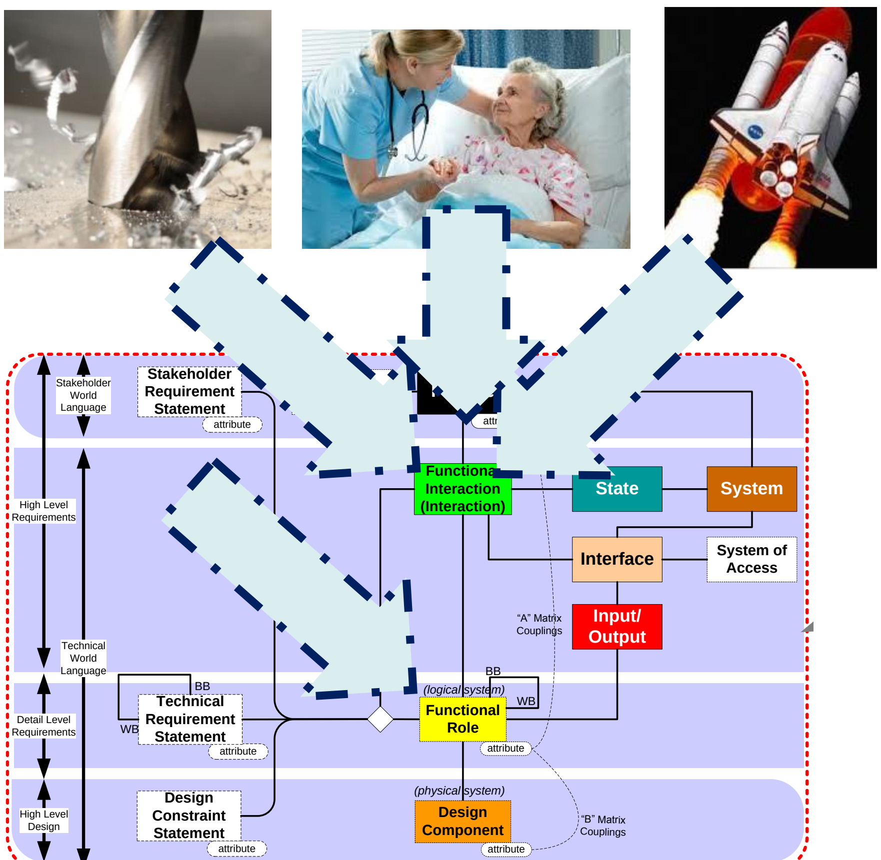
- Features are parameterized by Feature Attributes.
- These measures of effectiveness are in Stakeholder terms, so are frequently subjective and non-technical.

Features: Commercial Transportation, Cruise Control, Navigation, Consumables Compatibility, etc.

In S*Models, Feature models are summarized by Stakeholder Models and associated Feature Frameworks (Including Feature Attributes, Definitions, and Stakeholder associations with the Features)



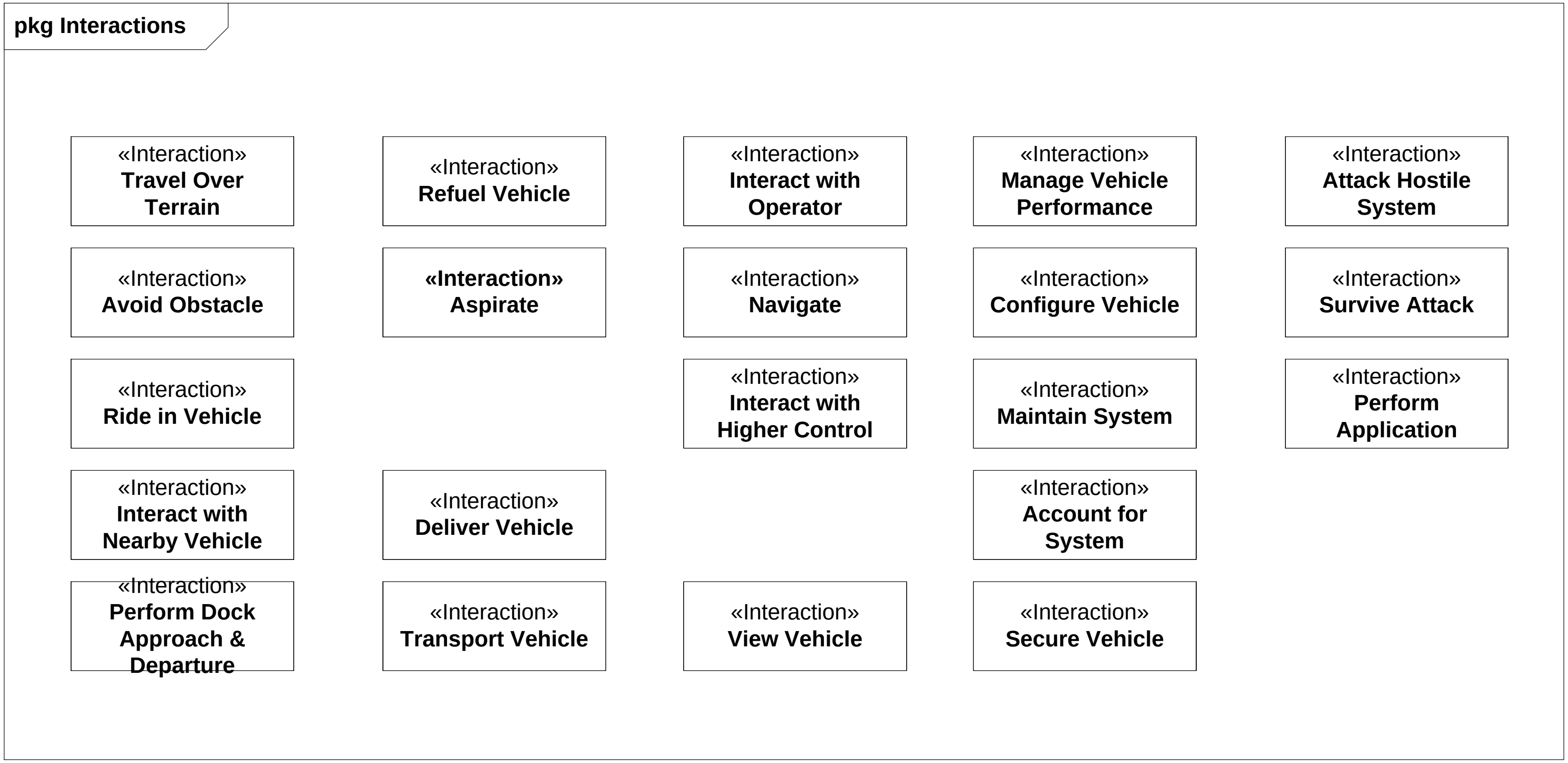
- A (Functional) Interaction is an exchange of energy, force, mass, or information, by two or more entities, said to play (Functional) Roles in the Interaction.
- All behavior occurs in the context of Interactions.



- Functional Role behaviors are parameterized by (technical) Role Attributes.
- These describe behavior variables in objective, technical terms—the language of science and engineering.

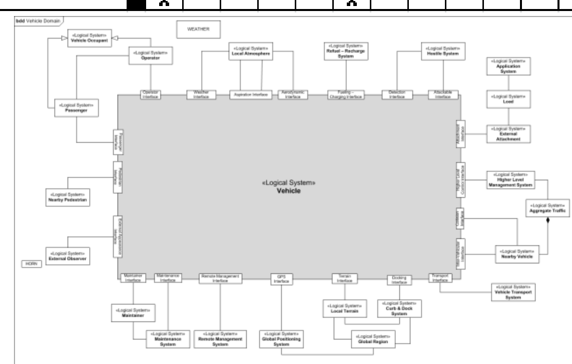
Interactions: Travel over Terrain, Ride in Vehicle, Maintain System, etc.

- In the High Level Requirements (HLR) framework subset of an S*Model, the Interactions are summarized by name, definition, and active role-players.
- The HLR framework provides a place to associate each Interaction with related Actors, Features, and States.
- In the Detail Level Requirements (DLR) subset of an S*Model, each Interaction can be detail modeled, leading to detail Requirements and other aspects.

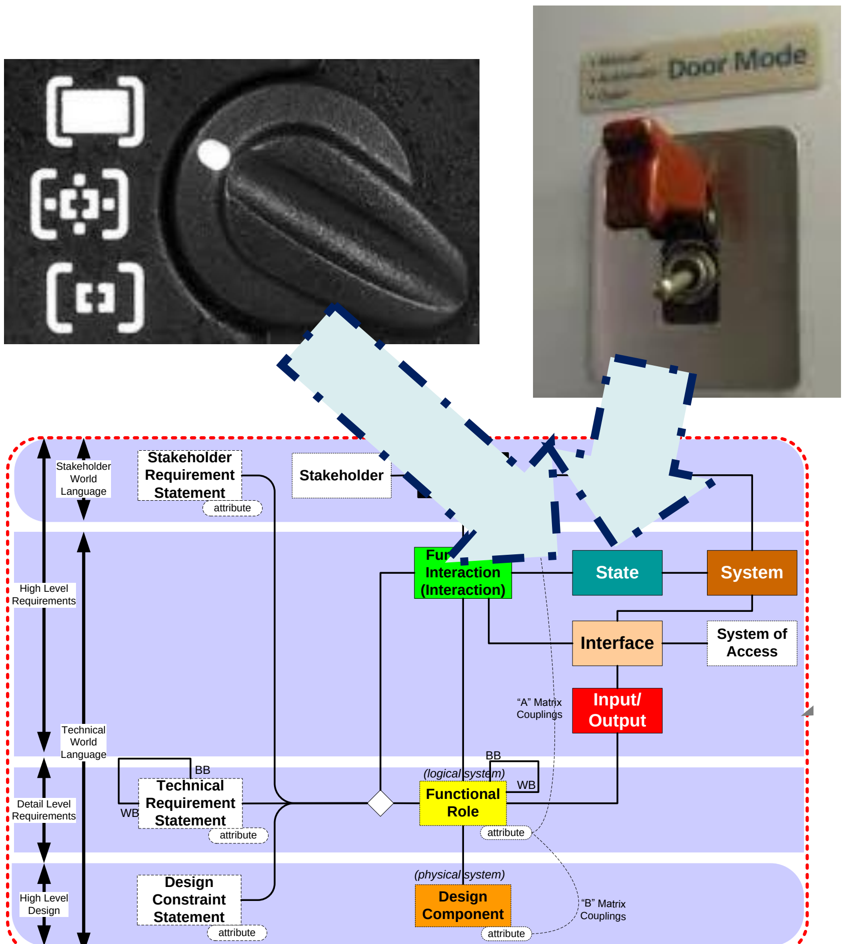


Vehicle Interactions:
Which Actors Participate in Interaction?

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																



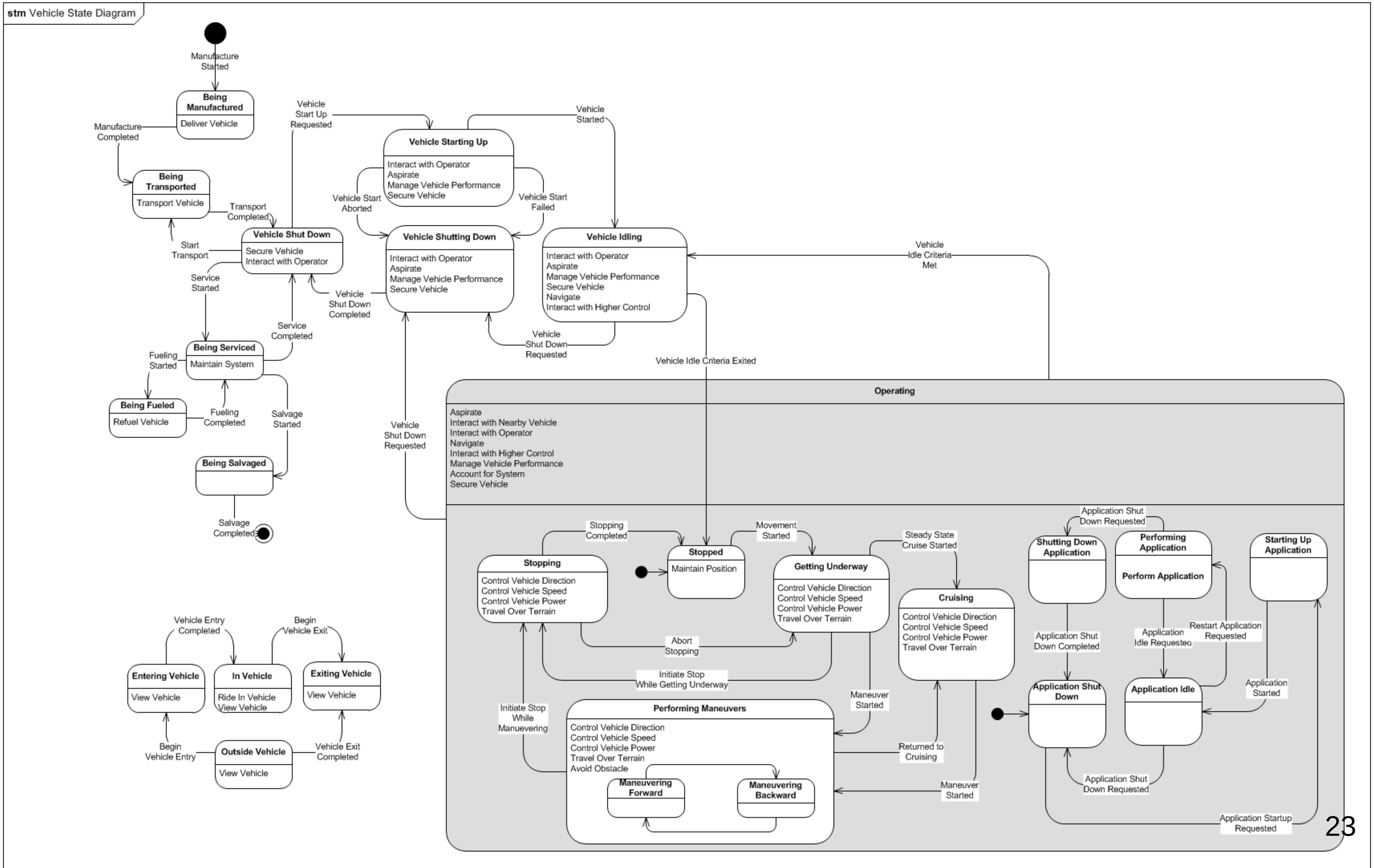
- A <u>State</u> is a condition of a system that determines its future behavior.
- Some state variables are continuous (e.g., position, velocity), and others are discrete (e.g., operational states).
- For the discrete case, Finite State Machine models are used.



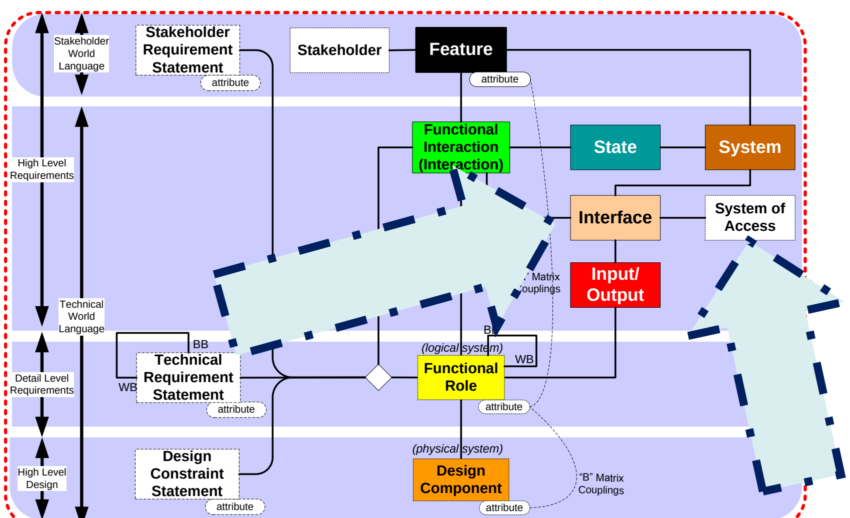
- The fact that different behavior is expected in the different (finite) states is represented by associating different Interactions with different (finite) States.

Finite States: Being Serviced, Stopped, Stopping, Cruising, Performing Maneuvers, etc.

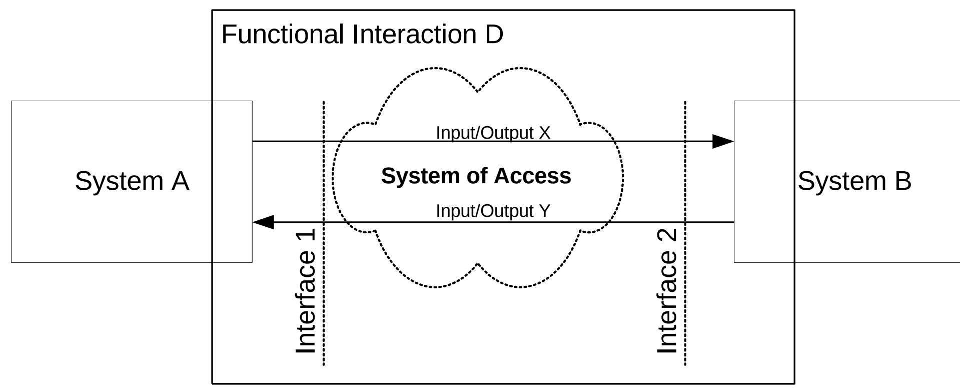
- In the High Level Requirements (HLR) subset of an S*Model, the State Model establishes a high level temporal (time) model of the system.
- The scope of such a State Model may be the entire System Life Cycle, an Operational Cycle, or other time scope.



An Interface is an association of (1) a system (which has the interface), (2) a set of Input-Outputs (which pass through the Interface), (3) a set of Interactions (which describe behavior at the Interface), and (4) a System of Access (which provides the physical transport at the Interface).

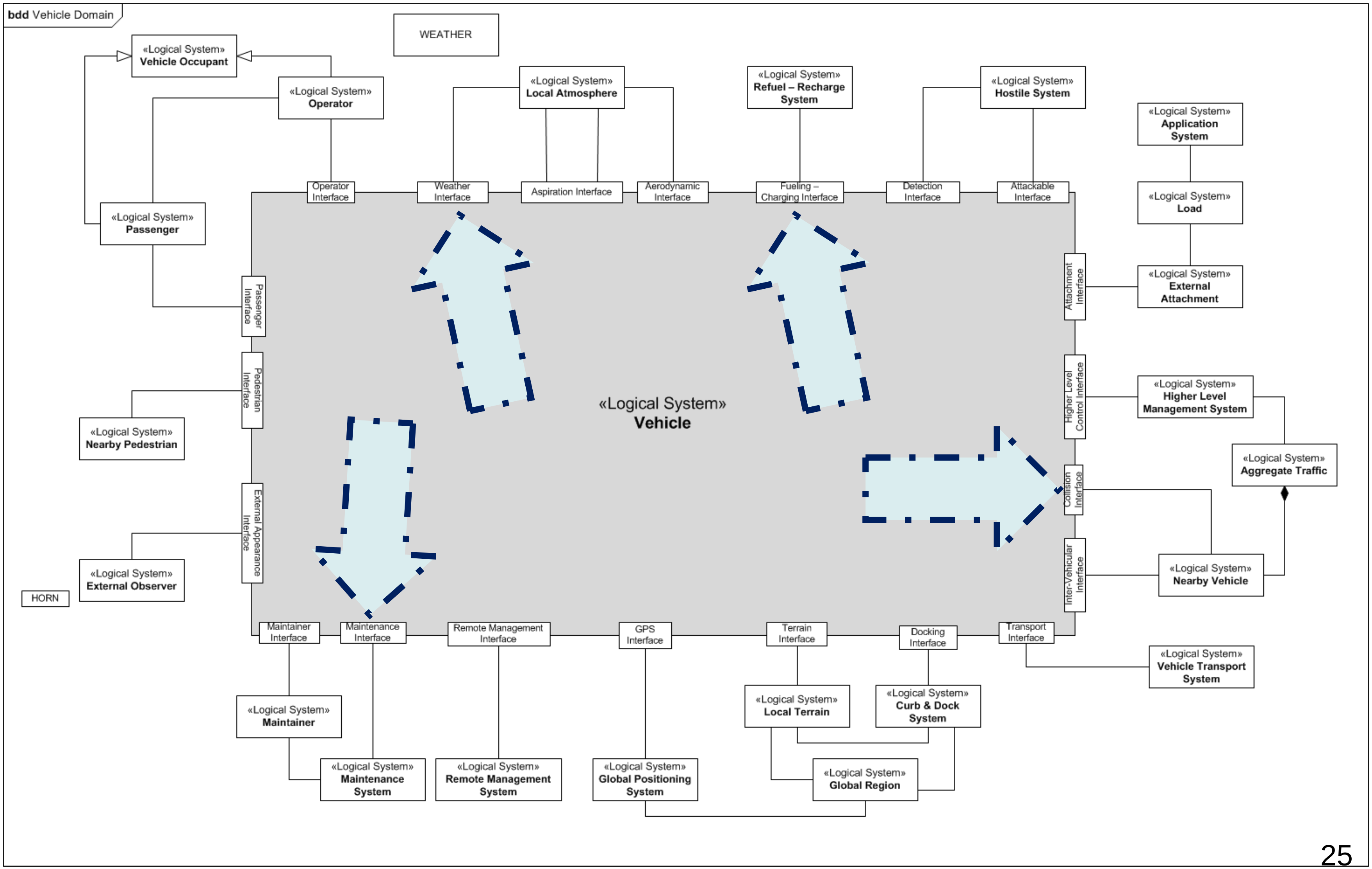


S*Models show that there are multiple interfaces between systems:

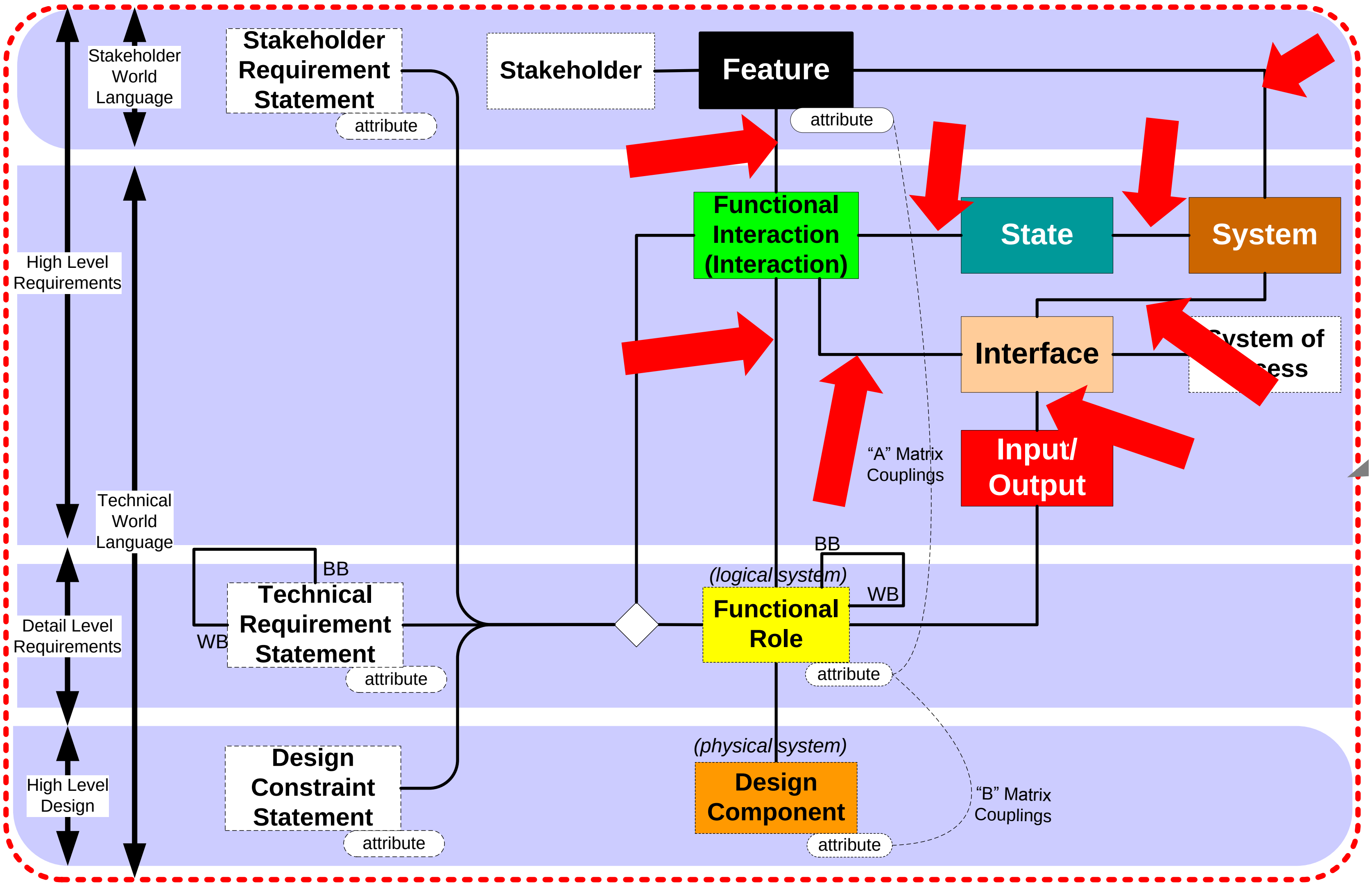


Systems of Access: Steering Wheel, Dashboard Indicator, Pedal, Headlights, etc.

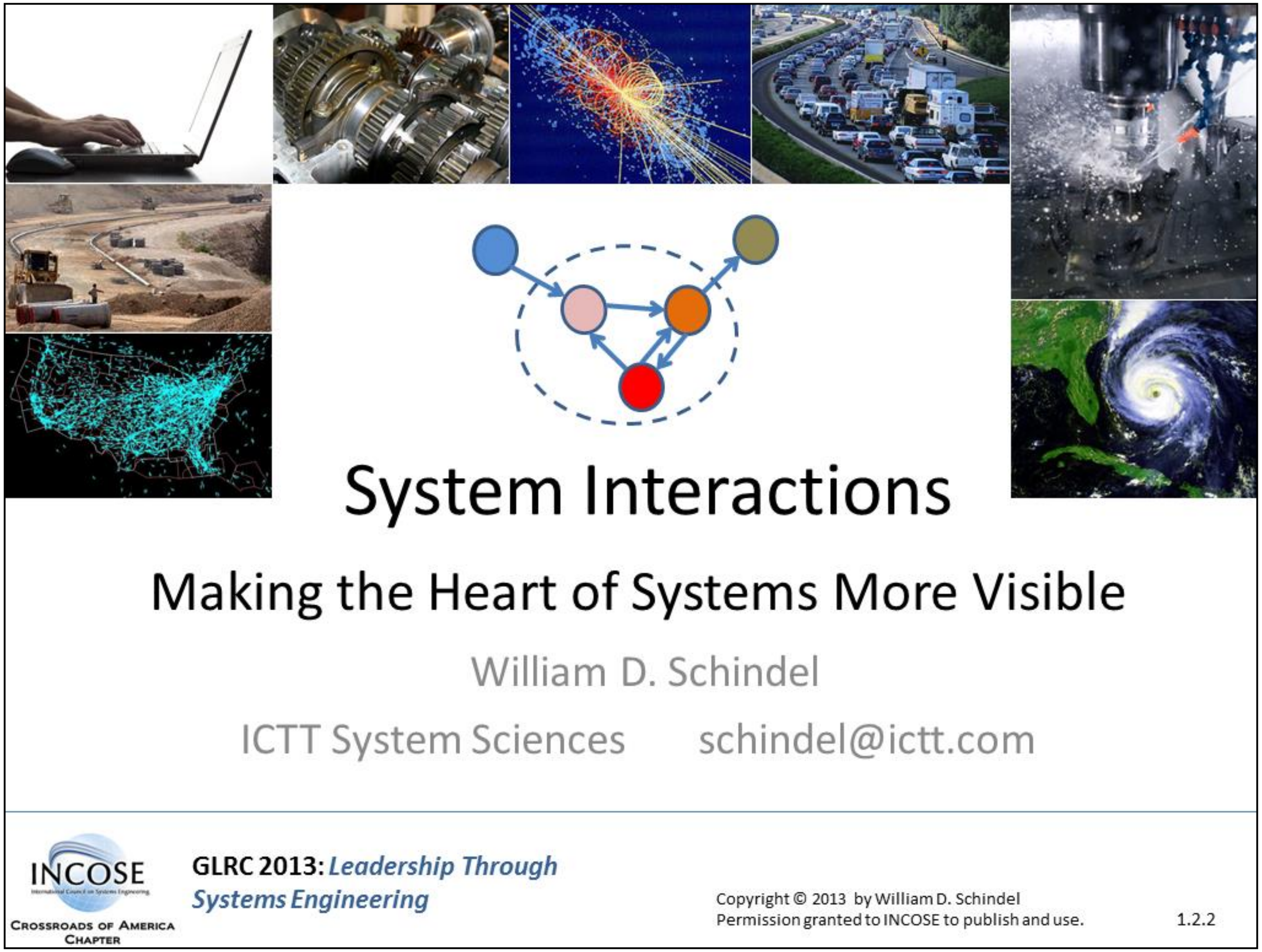
In S*Models, external Interfaces can appear at the edge of systems (i.e., in Domain Diagrams), and queries can be used to generate Interface Control Document (ICD) views.



To be ready for a later section below, it is important to be very aware of the web of S*Relationships linking the classes we have been discussing (the lines in the S*Metamodel):



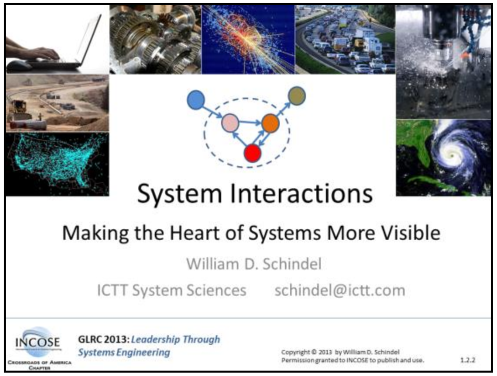
Criticality of modeled Interactions to the heart of MBSE, PBSE, science and engineering



Downloadable from: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns

Criticality of modeled Interactions to the heart of MBSE, PBSE, science and engineering

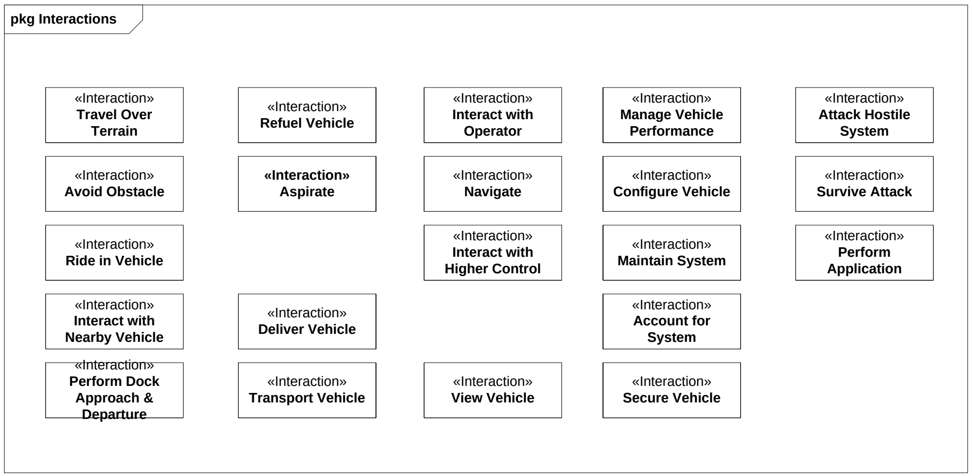
For more detail, see -->

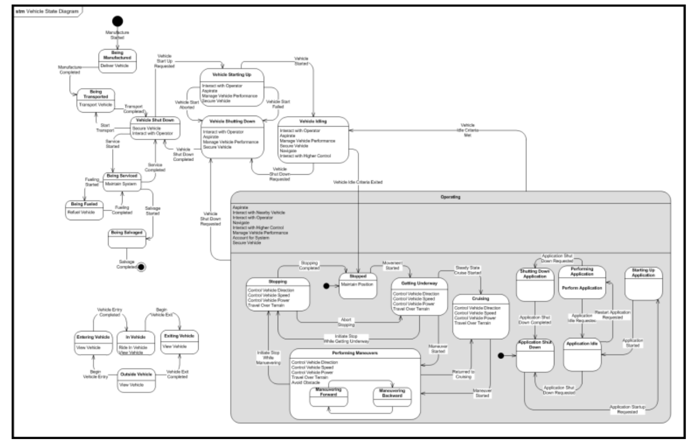


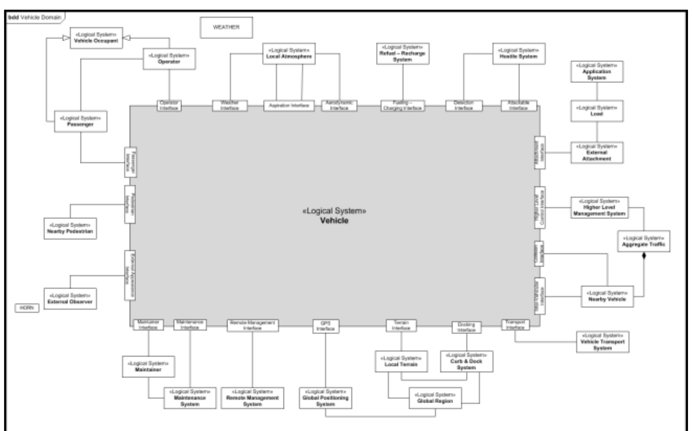
- In a nutshell:
  - Physical interaction models provide the context for all the laws of the hard sciences (Newton, Maxwell, Boltzmann, etc.).
  - Explicit models of physical interactions are perfectly legal in MBSE models (collaboration, activity, etc.), but are frequently under-emphasized in them.
  - All physical behavior occurs in the context of interactions—there is no behavior we know except behavior in interactions.
  - All system "black box (BB) requirements" are descriptions of "one side" of behavior – what a subject system does during interactions.
  - Engineers frequently model only "one side"—what "my system does", but not the overall interactions it has with its (equally active) environment.
  - This leads to missed assumptions and requirements.
  - To find all system BB requirements, find all system external Interactions.
  - These Interactions can be systematically discovered through three independent relational paths—through associated Interfaces (Actors), States (Modes), and (Stakeholder) Features; this enhances ability to discover more Interactions.
  - "White box interactions" are equally powerful representations of *design*.

Criticality of modeled Interactions to the heart of MBSE, PBSE, science and engineering

- The HLR model identifies (names, defines) Interactions, who participates in them, when they occur, and why (the stakeholder Features they support):



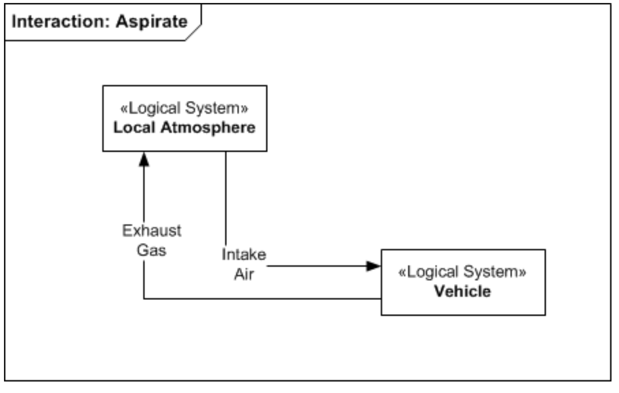


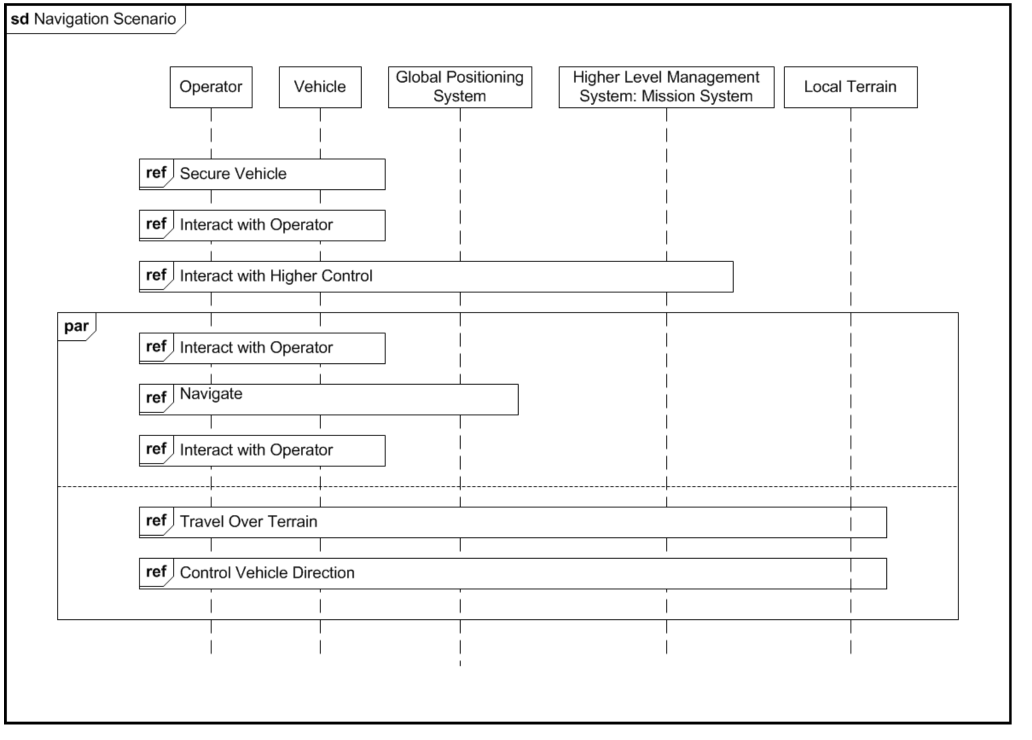


1			Actors																					
2	Interaction Name	Interaction Definition	Vehicle	Operator	Passenger	Vehicle Occupant	Nearby Pedestrian	External Observer	Maintainer	Maintenance System	Local Atmosphere	Refuel System	Hostile System	External Attachment	Load	Application System	Higher Level Management	Nearby 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																

Criticality of modeled Interactions to the heart of MBSE, PBSE, science and engineering

- The DLR model identifies what occurs during an individual Interaction, as an exchange of Energy, Force, Mass, or Information between interacting functional roles.
- Typical DLR model views include Collaboration Diagrams, Activity Diagrams, Timing Diagrams, FFBDs, etc.:





Understanding Requirements Statements as non-linear Transfer Functions

Session 11.2.1

INCOSE 2005 ROCHESTER, NY

Systems Engineering: Bridging Industry, Government and Academia

Requirements Statements Are Transfer Functions: An Insight from Model-Based Systems Engineering

Bill Schindel,

ICTT, Inc. and System Sciences, LLC

INCOSE
INTERNATIONAL COUNCIL ON SYSTEMS ENGINEERING

1

INCOSE 2005 Symposium "Best Paper" Award in Modeling and Tools

Requirements Statements Are Transfer Functions: An Insight from Model-Based Systems Engineering

William D. Schindel
ICTT, Inc., and System Sciences, LLC
100 East Campus Drive, Terre Haute, IN 47802
812-232-2062 schindel@ictt.com

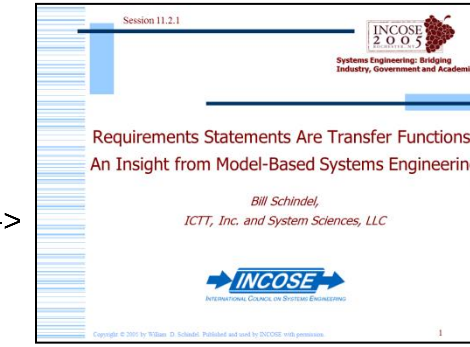
Abstract. Traditional systems engineering pays attention to careful composition of prose requirements statements. Even so, prose appears less than what is needed to advance the art of systems engineering into a theoretically-based engineering discipline comparable to Electrical, Mechanical, or Chemical Engineering. Ask three people to read a set of prose requirements statements, and a universal experience is that there will be three different impressions of their meaning. The rise of Model-Based Systems Engineering might suggest the demise of prose requirements, but we argue otherwise. This paper shows how prose requirements can be productively embedded in and a valued formal part of requirements models. This leads to the practice-impacting insight that requirements statements can be non-linear extensions of linear transfer functions, shows how their ambiguity can be further reduced using ordinary language, how their completeness or overlap more easily audited, and how they can be "understood" more completely by engineering tools.

Systems Engineering Prose

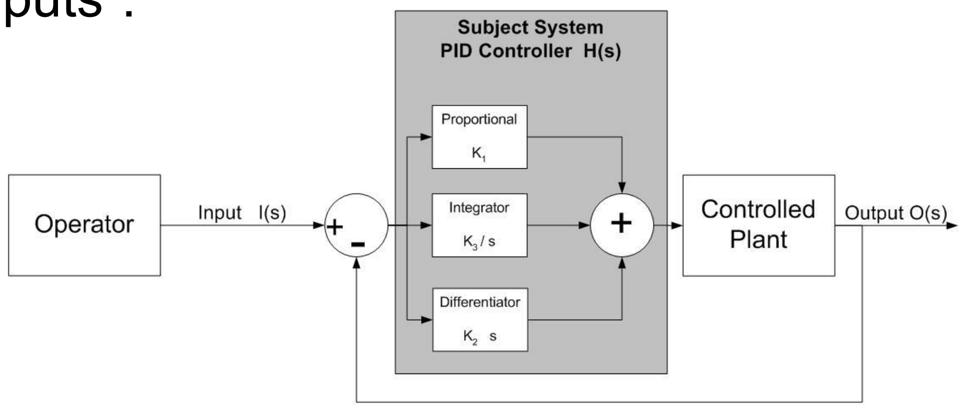
Traditional Requirements Discipline. Composing good requirements statements prose has a

Understanding Requirements Statements as non-linear Transfer Functions

For more detail, see -->



- In a nutshell:
  - The "Transfer Function" perspective of signals and systems fully characterizes the (externally visible) behavior of a system, as a sort of "ratio of outputs to inputs":



$$H(s) = (K_1 + (K_2 \; s) + (K_3 / s))$$

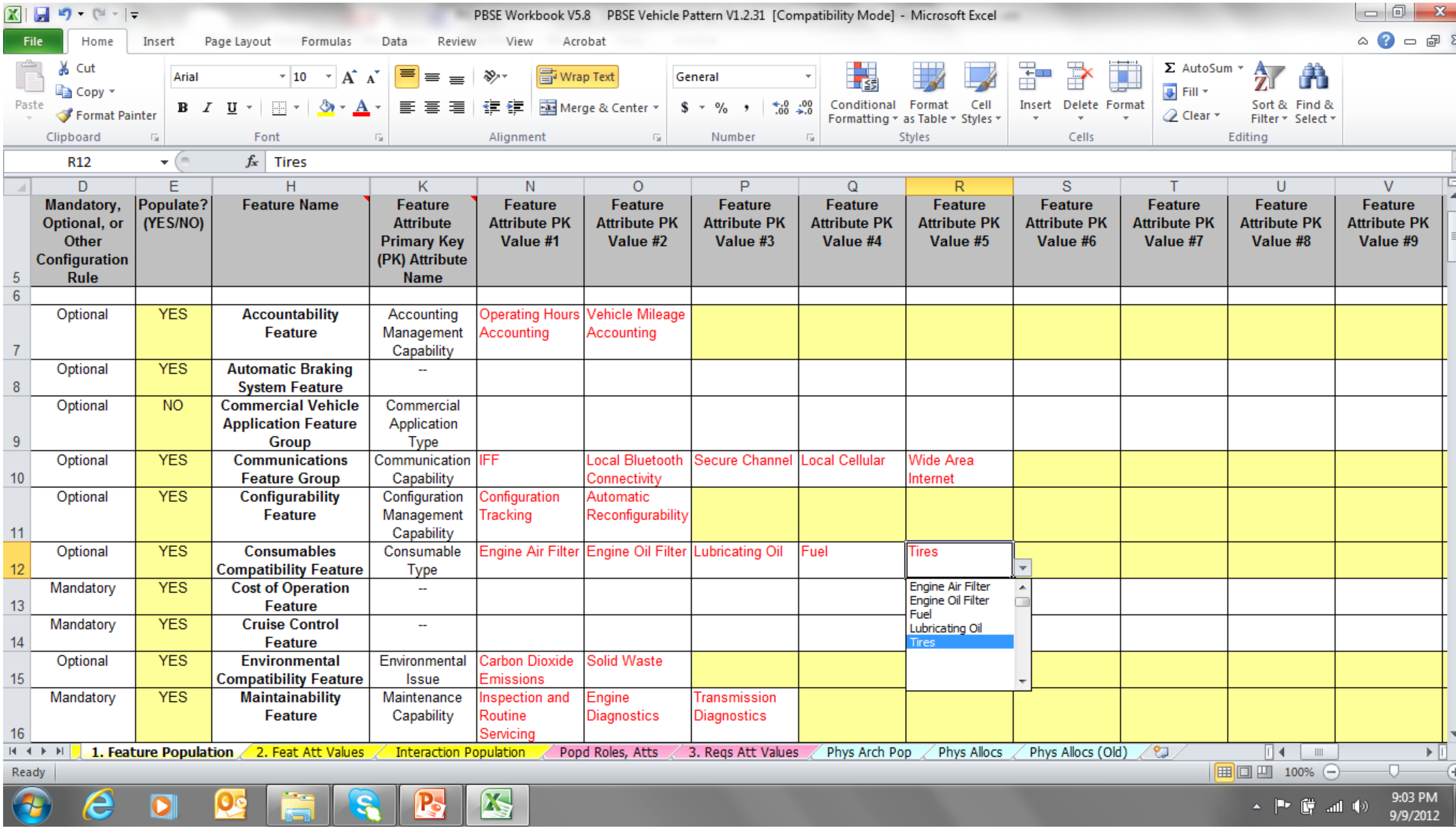
  - However, Transfer Functions are limited to linear systems, and describes their behavior in the frequency domain. Systems generally are not linear, and frequently not described by available mathematical equations!
  - However, for general systems we can extend the idea of (linear) Transfer Functions, as a way to understand Requirement Statements . . .

Understanding Requirements Statements as non-linear Transfer Functions

- We can <u>borrow a key idea</u> from the "Transfer Function perspective":
  - Characterizing a system's behavior by stating the externally visible <u>relationships between its inputs and outputs</u>
  - In words, and only infrequently as equations, and often not in the frequency domain, and usually not linear.

- All Requirement Statements then become descriptions of relationships (quantitative, temporal, functional, statistical, etc.) between system inputs and outputs:
  - Offers a powerful way to understand that the only thing Requirements Statements can describe are those relationships, parameterized by requirements parameters (efficiency, delay, yield, reliability, capacity, etc.)

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

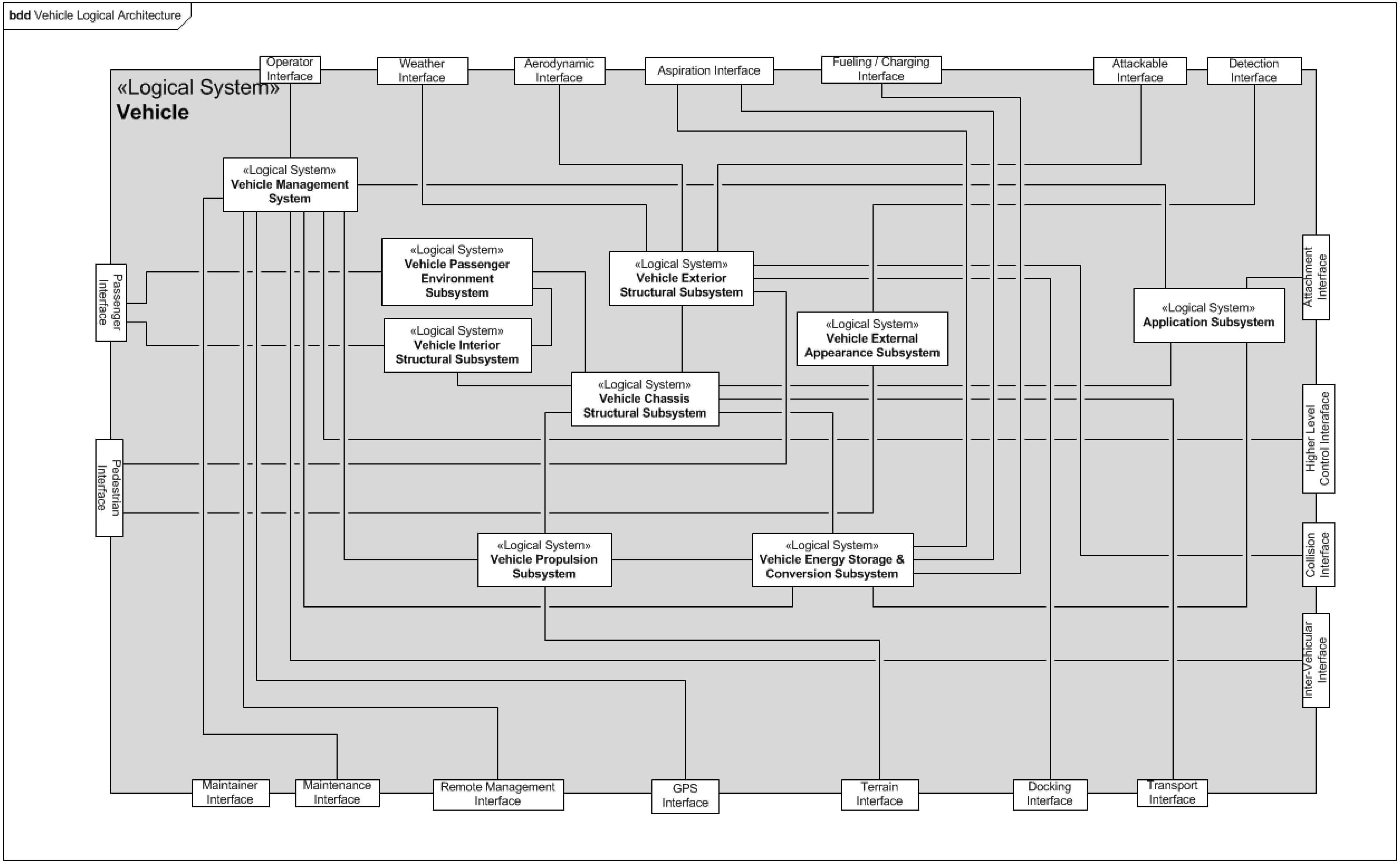
PBSE Workbook V5.8 PBSE Vehicle Pattern V1.2.31 [Compatibility Mode] - Microsoft Excel

L9 | The vehicle shall be capable of operating 50,000 miles between tire changes.

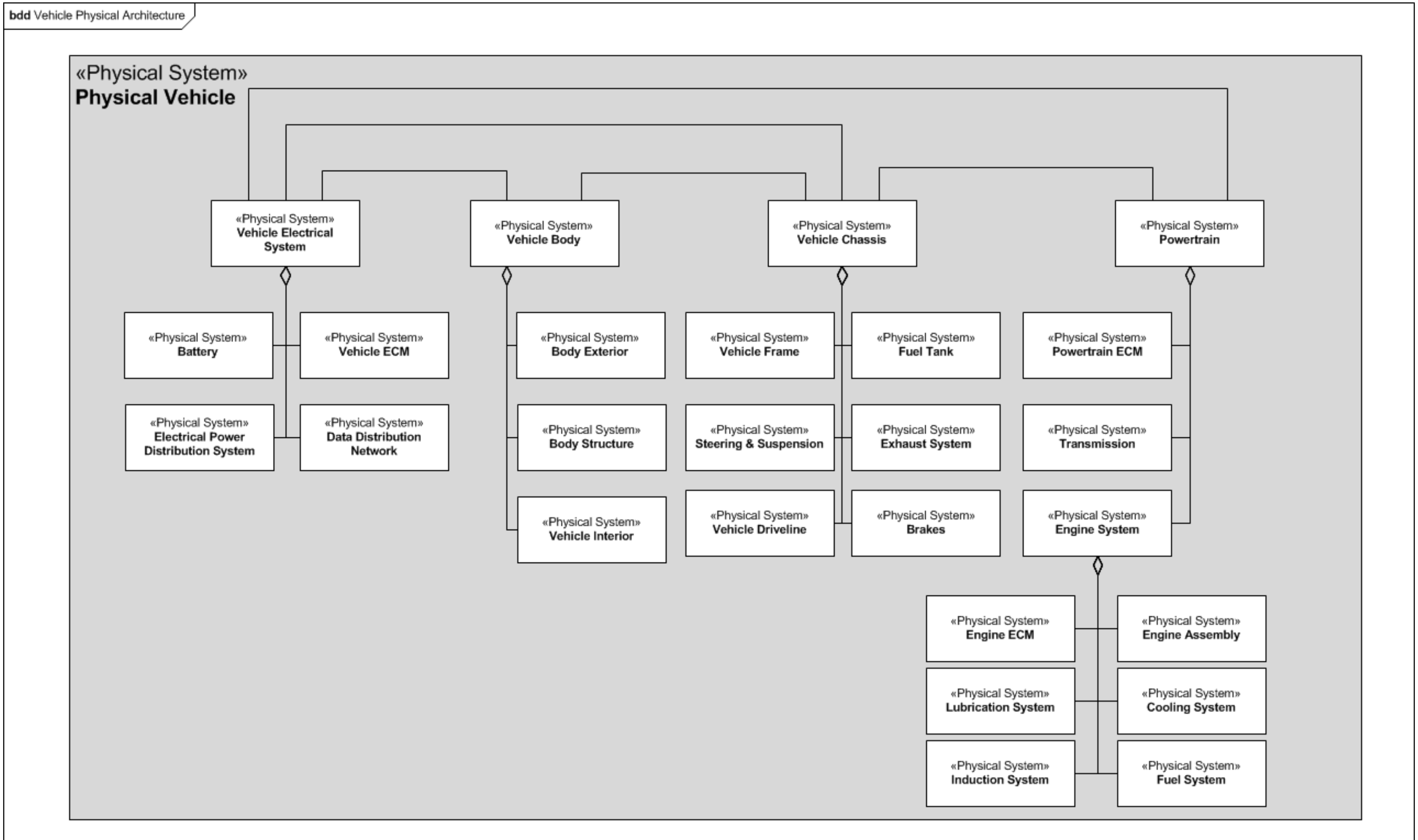
	A	F	G	H	J	L
1	Features	Interaction	Interaction PK Value	Functional Role	Req ID	Requirement
2	Accountability Feature[Operating Hours Accounting]	Account for System	Operating Hours Accounting	Vehicle	VEH-1002	The system shall record and make available for display the accumulated hours of vehicle operation.
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.

1. Feature Population | 2. Feat Att Values | Interaction Population | Popd Roles, Atts | 3. Reqs Att Values | Phys Arch Pop | Phys Allocs | Phys Allocs (Old)

Logical Architecture Model

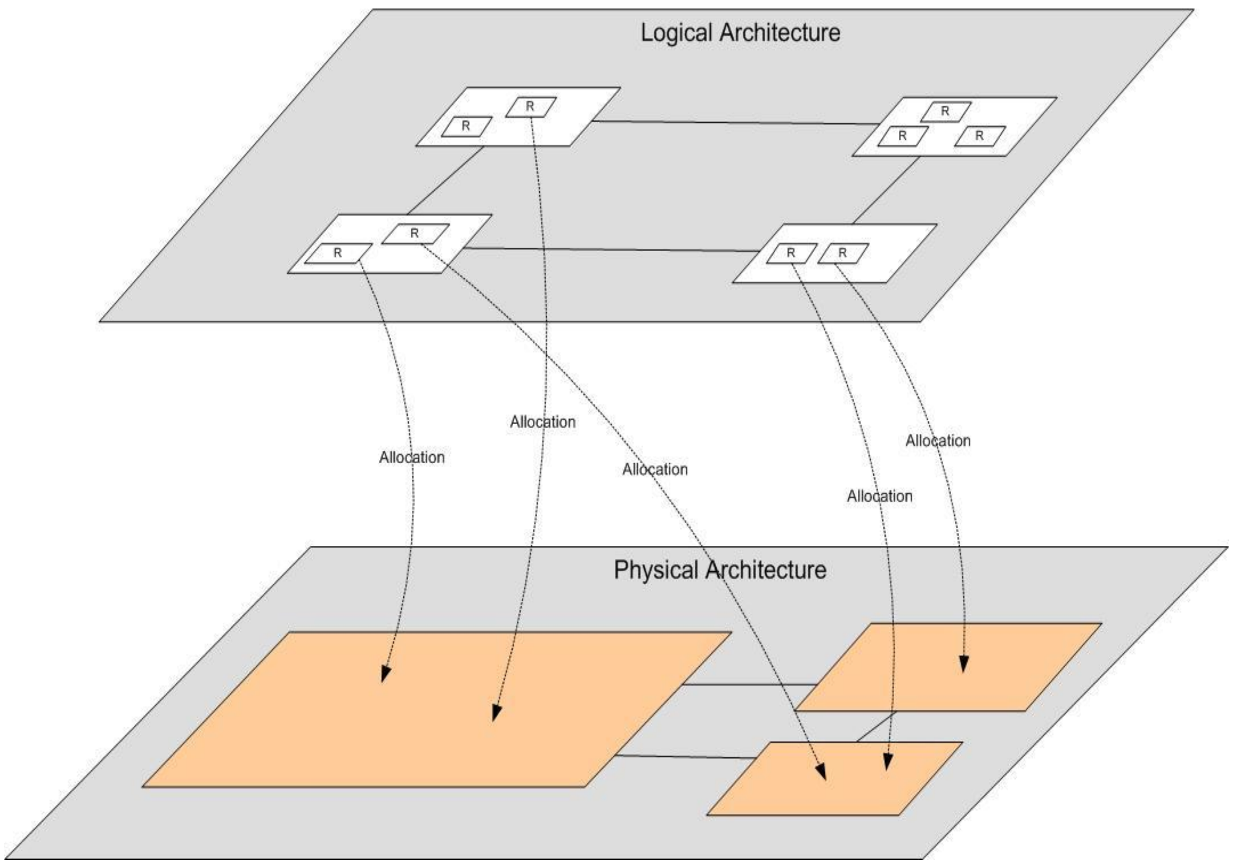


Physical Architecture Model



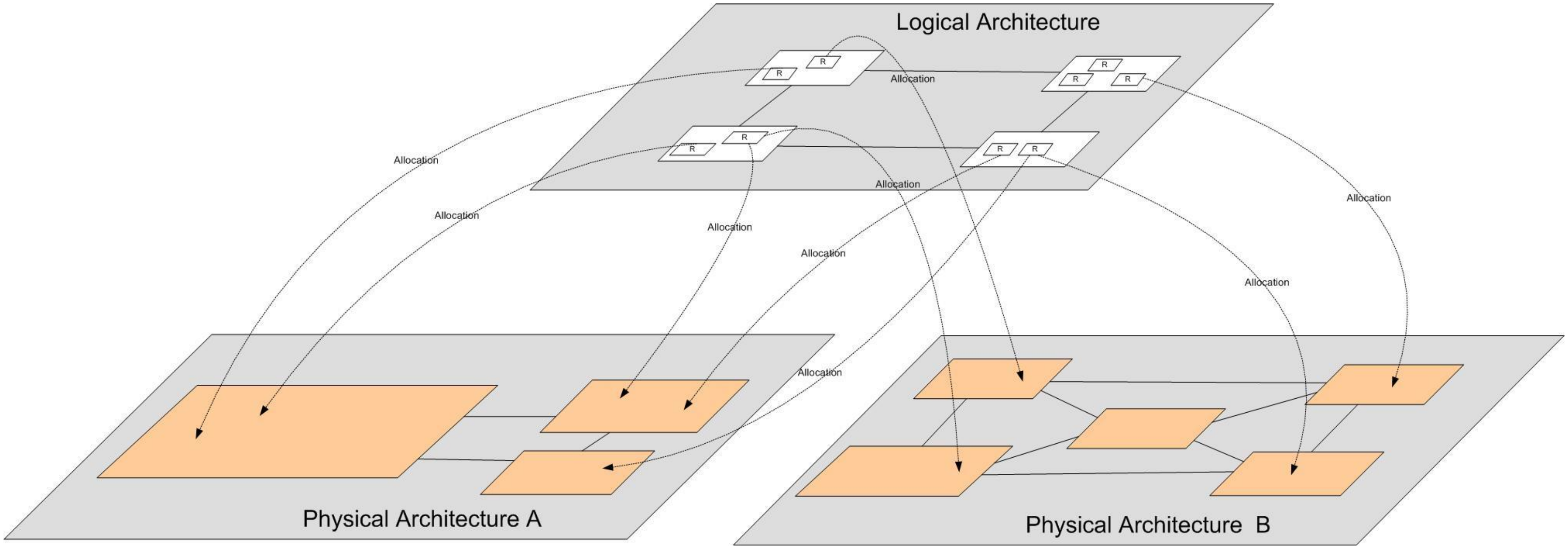
Acknowledgement: Influenced by related physical architecture work of John Thomas

Allocation of Logical Roles to Physical Architecture



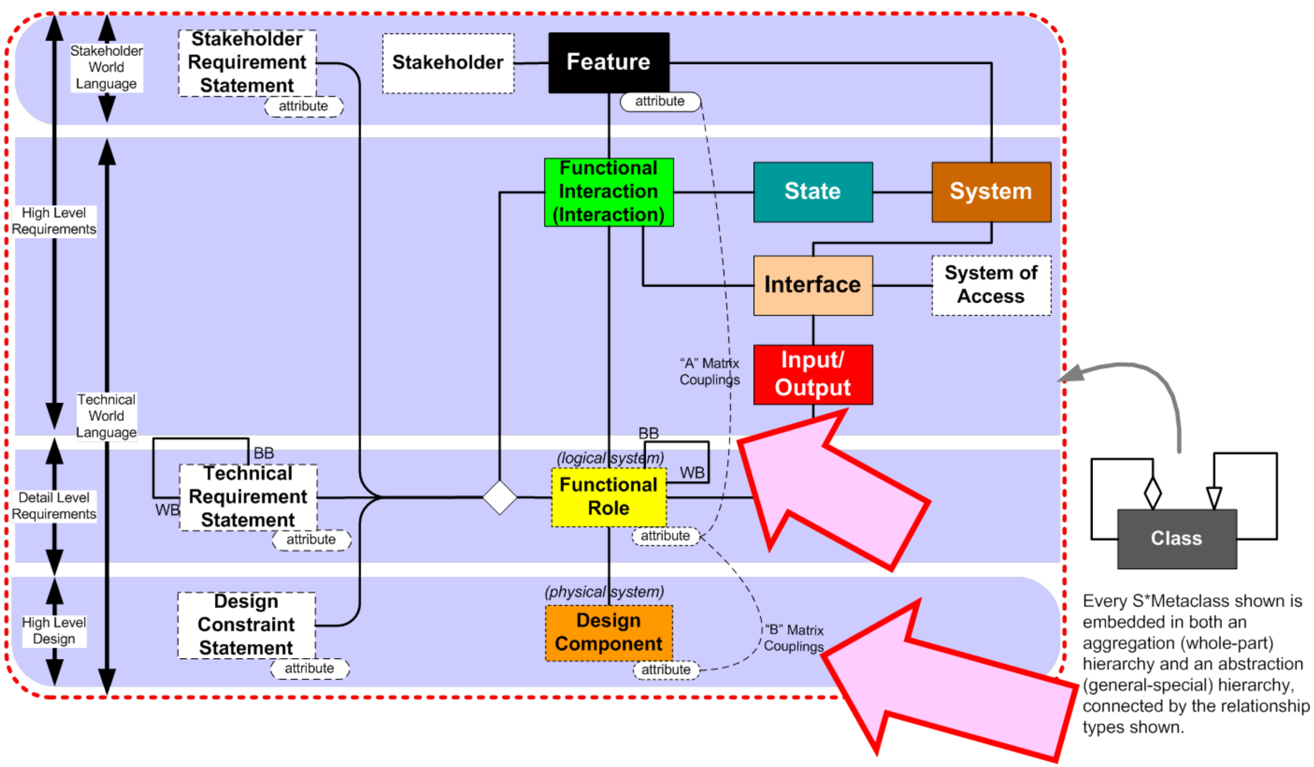
Allocation of Logical Roles to Physical Architecture

- Same Logical Architecture covers many Physical Architectures:



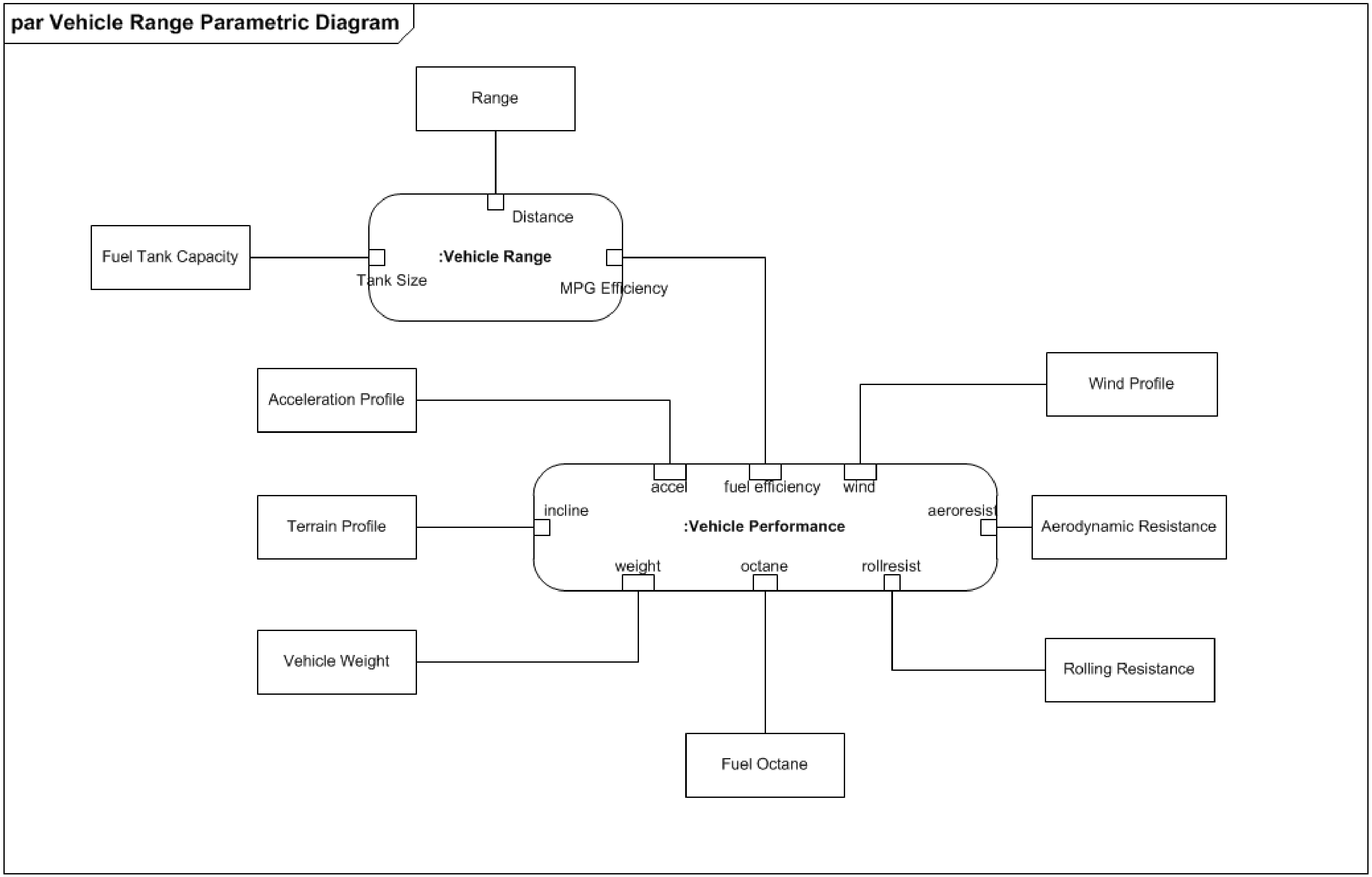
Attribute parametric couplings

- Stakeholder Feature Attribute – to – Technical Roles & Requirements Attribute Couplings;
- Technical Roles & Requirements Attribute – to – Physical Component Attribute Couplings.

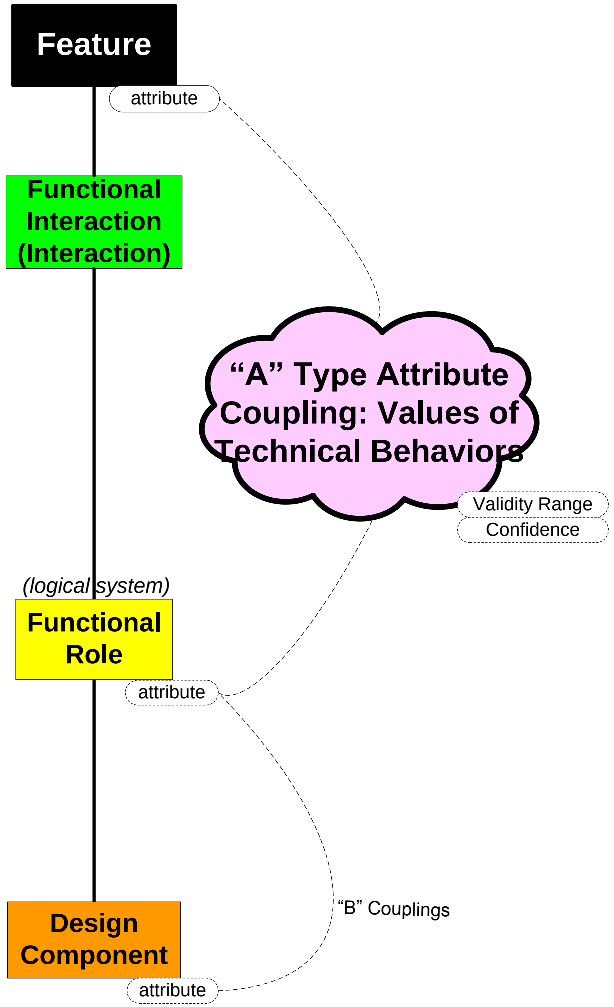


Every S*Metaclass shown is embedded in both an aggregation (whole-part) hierarchy and an abstraction (general-special) hierarchy, connected by the relationship types shown.

Attribute Coupling Model: SysML Notation



Stakeholder Feature Attribute – to – Technical Roles & Requirements Attribute Couplings



These "A" type parametric couplings describe how parameter value changes in technical behavior (the attributes of Roles / Requirements) bear on changes in Stakeholder-perceived value (the attributes of Stakeholder Features).

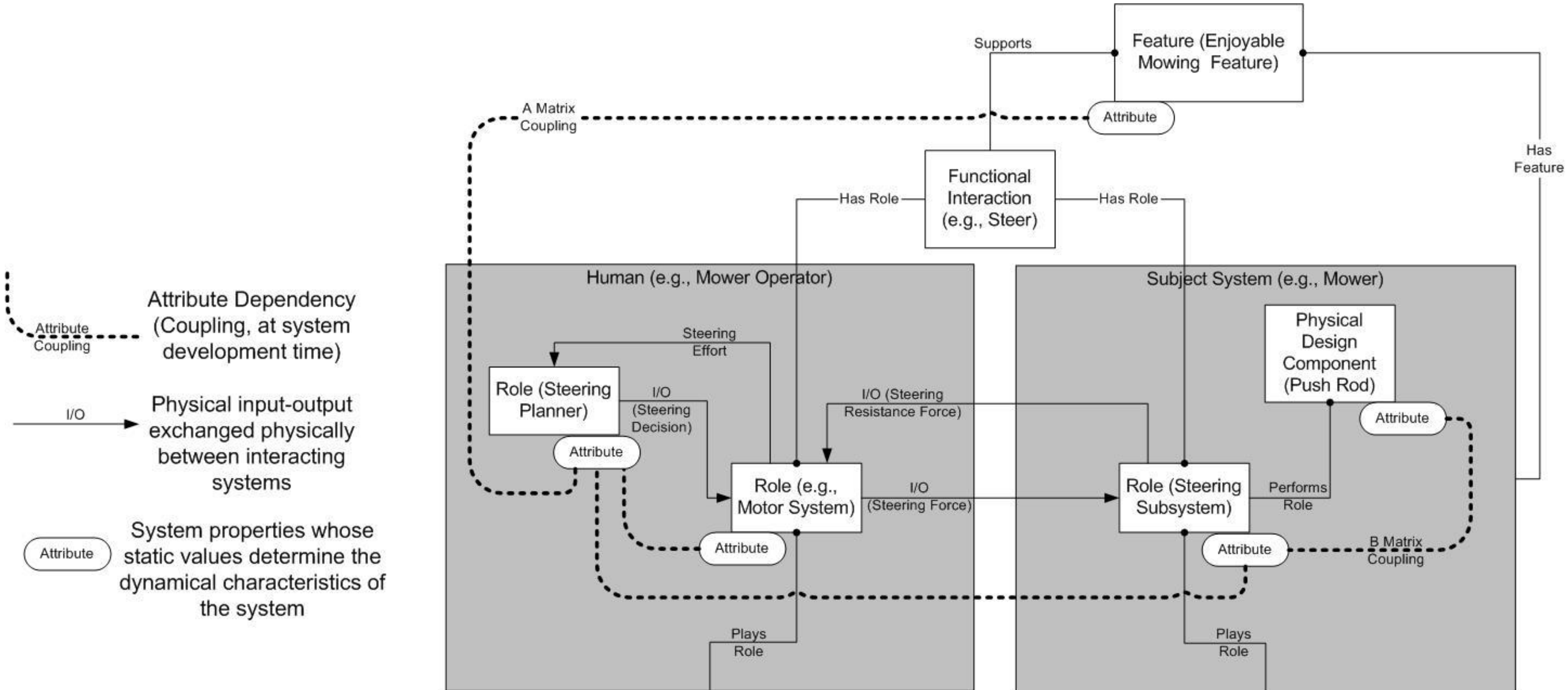
A simple example

Operator fatigue, sense of control, coupled to technical steering gain, mower speed:

- “Enjoyable Mowing Feature” attributes

  are coupled to attributes of . . .
- “Operator Steering Planner” role
- “Operator Motor System” role
- “Mower Steering Subsystem” role

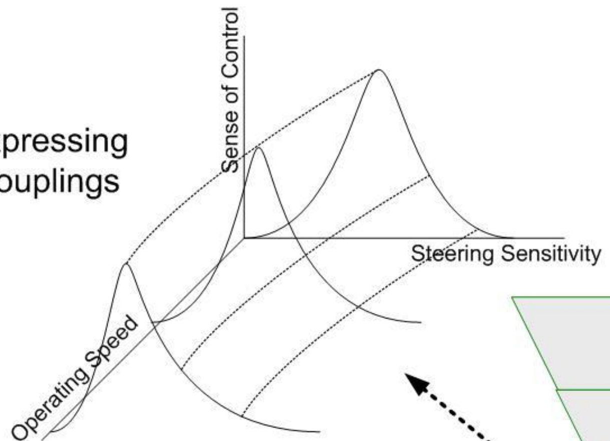
  which are coupled to attributes of . . .
- “Mower Steering Push Rod” component



A simple example

- Expressing these couplings as tables, charts, graphs, or otherwise captures our <u>best currently available knowledge</u> of human behavior as well as mechanics.
- Creates integrated view contributed to & shared across a team of specialists in humans versus machine design.

Expressing Couplings



Features

Feature Attributes

Functional Roles

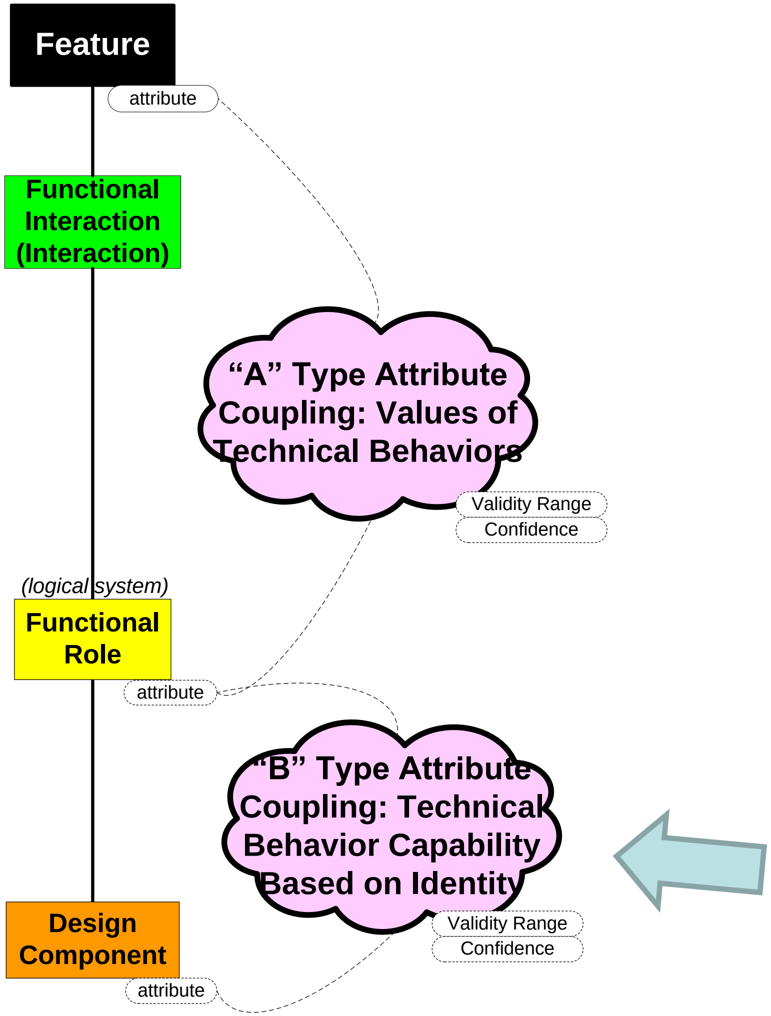
Role Attributes

Attribute Couplings

		Enjoyable Mowing Feature			Market Segment Fit Feature	
		Sense of Control	Operator Fatigue	Operator Comfort	Application Positioning	Purchase Price
Application Environment	Terrain Roughness			X	X	
Steering Subsystem	Steering Tactile	X				
	Steering Sensitivity	X	X		X	
	Cost					X
Seating System	Shock Absorbence		X	X	X	
	Seat Cushioning		X	X	X	
	Cost					X
Propulsion System	Max Acceleration	X	X		X	
	Max Mowing Speed	X	X		X	
	Braking	X			X	
	Cost					X

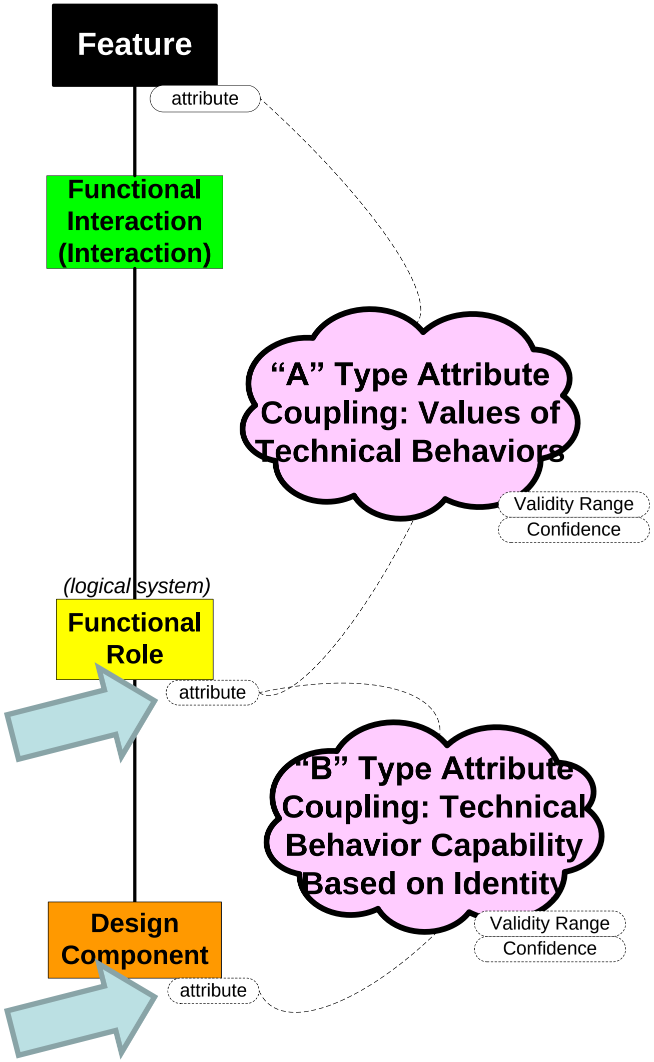
<u>"A" Matrix</u>: Expresses Feature-Role Attribute Couplings

Technical Roles & Requirements Attribute – to – Physical Component Attribute Couplings



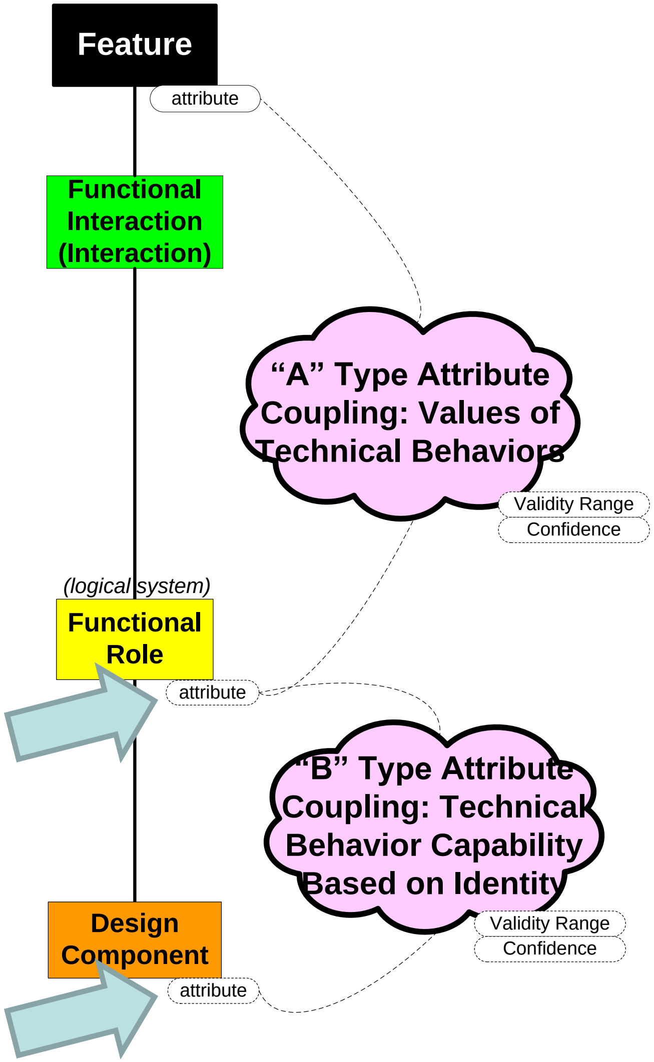
The "B" type parametric couplings describe how parameter value changes in <u>design components</u> (the attributes of Design Components) bear on changes in <u>technical behavior</u> (the attributes of Roles / Requirements)

Key methodology point:



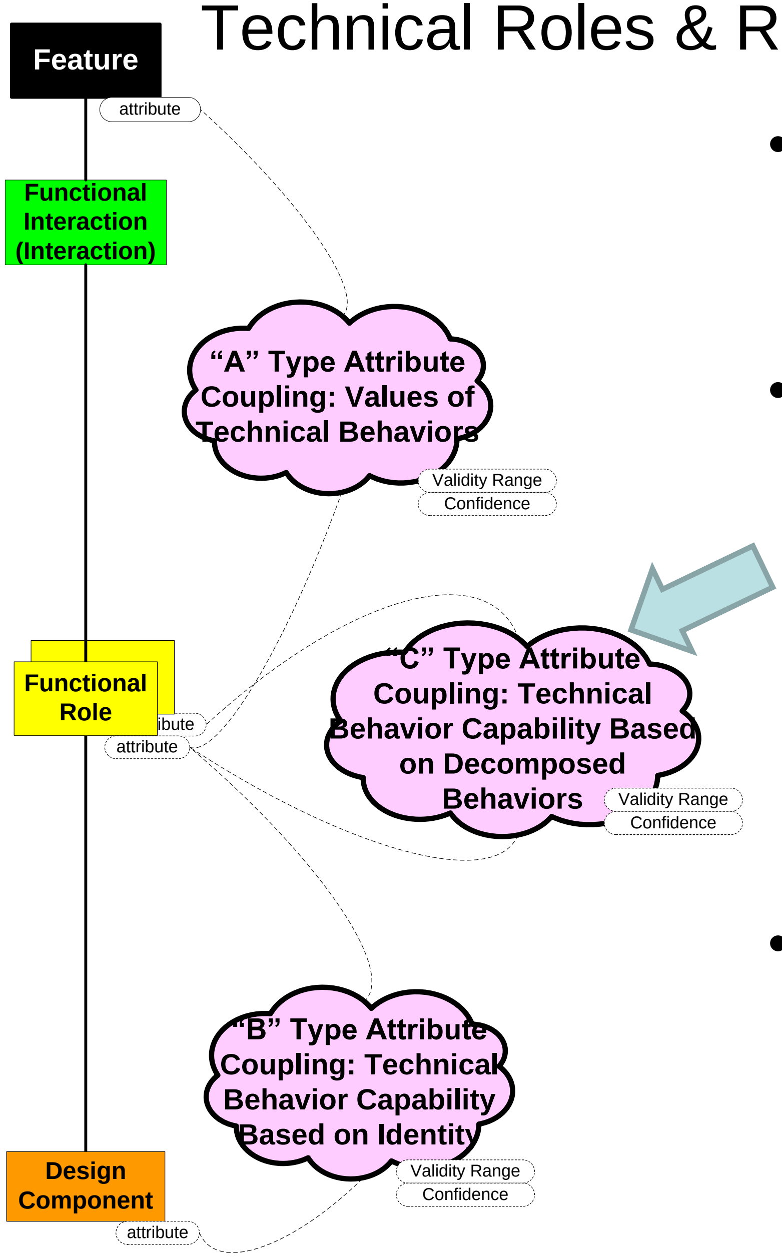
- Modeled technical behavior (including its parameterization) is focused in the Functional Roles (including their parameterization by Role Attributes, which are identical reappearances of the technical Requirements Attributes).
- So, the attributes of Design Components are not used to describe behavior! (After all, Design Components are characterized by their identity, not their behavior – their behavior comes entirely from allocations of Functional Roles to them.)
- The attributes of Design Components therefore describe identity or existence, not behavior.
- Examples include: Part Number, Department Name, Material of Construction, Chemical Element, Person, etc.

Key methodology point:



- In managing complex patterns and their multiple configurations, that aspect of the S*Model approach has tremendous utility.
- Among other things, it greatly simplifies parameter space complexity and proliferation of variables / namespace size.
- When systems are configured, all behavior parameter values (whether required, or achieved capability, or best in class, or competitor product), become "shadow values" of the same Functional Role attributes, for differently configured systems, including their Design Components.
- It also means that things like vendor data sheets, materials specifications, and similar information fit neatly into "B" coupling matrices or tables that show the values of Role Attributes for different Components, Materials, Compounds, etc.

Technical Roles & Requirements – to – Decomposed Technical Roles & Requirements Attribute Couplings



- When decomposing multi-level logical architectures, a third kind of attribute coupling appears.
- This "C" coupling describes how values of parameters of behavior (Functional Role attributes) are impacted by changes in values of parameters of subsystem behavior (Functional Role attributes).
- This is where mathematically expressed emergent phenomena of physics, chemistry, and larger scales are expressed.

<u>Our team's "Wave 1 projects</u>" IS2015 authors (if present)—brief summary of their papers, to be presented later this week:

- Improving Automated Test of Safety Critical Aerospace Systems -- Dave Cook, Moog Aircraft, et al (Session 10.1.1, Thursday)
- Reducing Error Escapes in the Development Process – Andy Pickard, Rolls Royce, et al (Session 7.4.2, Wednesday)
- Autonomous Vehicle Pattern – Troy Peterson, Booz Allen Hamilton, et al (Session 4.3.2, Tuesday)
- Improving Product Life Cycle Management – Saumya Sanyal, K2 Firm, et al (Session 10.3.2, Thursday)
- Representing and Improving System Life Cycle Trajectories – Bill Schindel, ICTT System Sciences (Session 6.4.2, Tuesday)

<u>Our team's "Wave 2 projects</u>" leaders—brief summary of their work now underway, and interest in participation by others:

- Systems Engineering Social Network Pattern – Chris Hoffman, Cummins, Inc.
- Health Care High Fidelity Transcription System Pattern – Vijay Thukral, Cientive Group
- Improving the Connection to Systems Value – Troy Peterson, Booz Allen Hamilton, et al
- Agile Systems Engineering Life Cycle Model Project & Pattern – Bill Schindel, ASELCM Team, and Agile Systems WG
- Strengthening Metamodel Support for MBSE: The Systems Phenomenon – Bill Schindel

INCOSE Agile Systems Engineering Life Cycle Model Project

The Agile
Systems Pattern
A Reference Model for
Agility in Systems

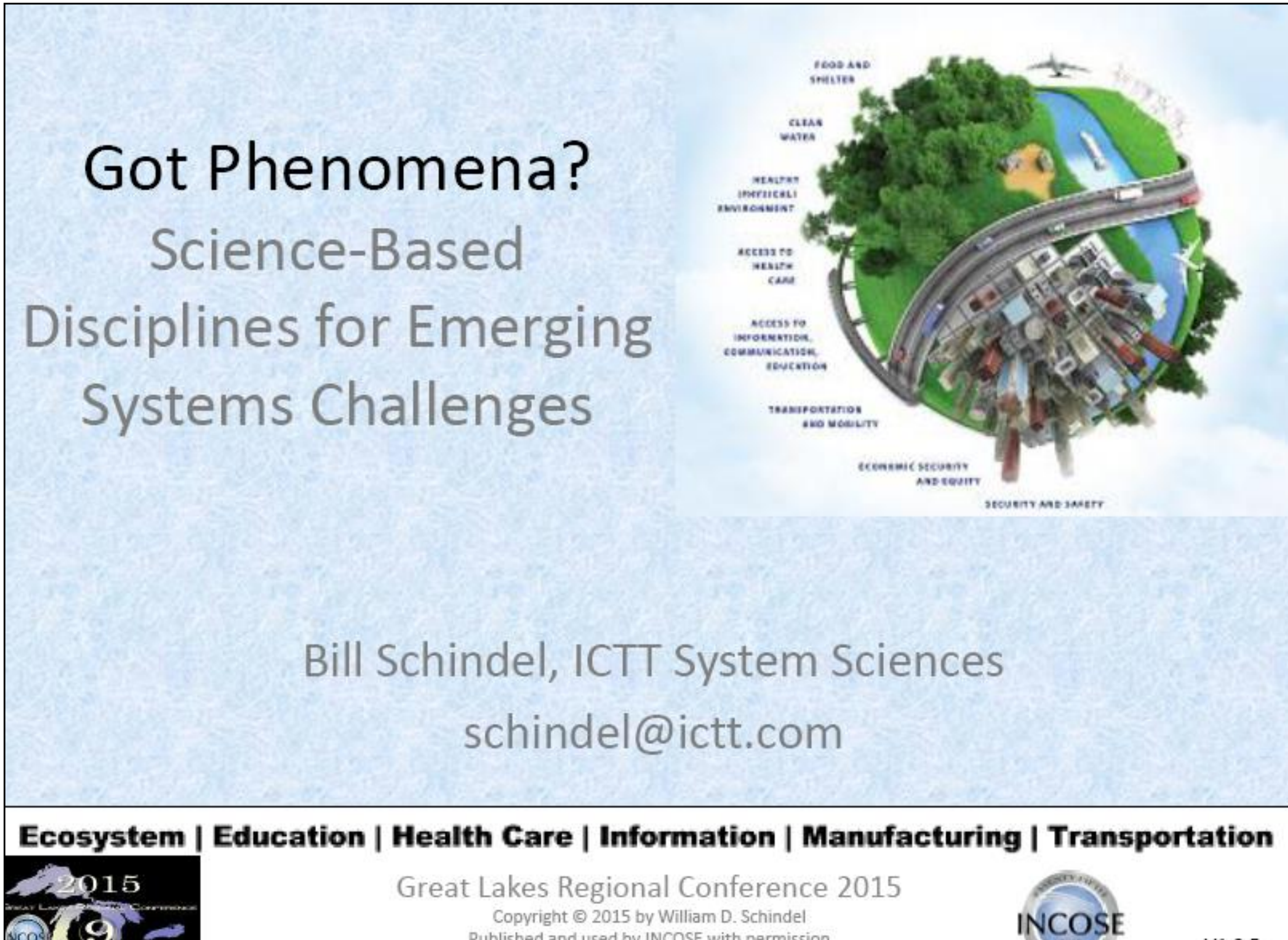
Bill Schindel, ICTT System Sciences
schindel@ictt.com

Ecosystem | Education | Health Care | Information | Manufacturing | Transportation

Great Lakes Regional Conference 2015

V1.2.1

Strengthening Metamodel Support for MBSE: The Systems Phenomenon



Planning Discussion: Next and Future Activities

- Future (Third Wave) Projects Pipeline Candidates:

Mapping PBSE to COTS Tools and Information Systems	Example SOS Pattern (Joint with SoS WG)
Mapping to ISO 15288; Processes vs. Data (Maps vs. Itineraries)	Supporting INCOSE objective for SE model-based; Case for Stronger Model Semantics
PBSE Implementation Strategies	Other interests from team members
Example Product Line Engineering (PLE) Pattern (Joint w/PLE WG)	

- Future meetings schedule: Pace, rate, calendar
- Outreach: Who else should be involved? Example—other INCOSE WGs that are natural Patterns applications. Ideas?