INCOSE/OMG MBSE Initiative
PBSE Patterns Challenge Team

Meeting: May 19, 2015

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

Today’s meeting . . .

- In addition to announcements and updates, . . .
- Today’s meeting will focus on technical matters.
- As discussed by members in the April meeting . . .

(Schedule adjustable as needed)

Meeting Agenda: INCOSE PBSE Patterns Challenge Team of MBSE Initiative (Web conferenced)

Tuesday, May 19, 2015, 4:00 – 5:00 PM Eastern Time

Pre-reading and Background: Team web site: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns
Minutes of meeting of April 21, 2015: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns_challenge_team_mtg_04.21.15

	4:00 – 5:00 PM EST
Meeting start up: • Review of meeting objectives and agenda – *at request of members, today's meeting returns to more technical focus* • Introduction of participants • Why the Patterns Challenge Team exists: Goals and approach	4:00 – 4:05
Announcements and updates: • Our team's co-chair, Troy Peterson, named INCOSE Asst. Director for SE Transformation to MBSE • INCOSE Great Lakes Regional Conference (GLRC9) 2015: Cleveland, October 23-25, 2015; submissions, registration: 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 • Updated PBSE Methodology Summary, based on team feedback (thank you!), for INCOSE MBSE Methodologies submission • Agile SE Life Cycle Model (ASELCM) Project (joint w/Agile WG) host enterprise workshops to begin August; five orgs in pipeline • SE Social Network Pattern Project started (Hoffman), for presentation at GLRC2015 in October • Health Care Delivery Pattern Application (Joint with Health Care WG) started (Thukral), for presentation at GLRC2015 in Oct • INCOSE Chapter presentations on PBSE: Northern Ohio (May 19); Previous: Crossroads of America, Chicagoland, Enchantment, Michigan, Finger Lakes • Other announcements or updates?	4:05 – 4:15
PBSE technical subjects—discussion of four related subjects, in progression: • 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	4:15 – 4:50
Planning discussion: • Discussion of candidates for focus of Patterns Challenge Team meeting at IS2015 in July; meeting schedule for same • Future (Third Wave) Projects Pipeline Candidates: (see table below) • Future meetings schedule: Pace, rate, calendar • Outreach: Who else should be involved? Example—other INCOSE WGs that are natural Patterns applications. Ideas?	4:50 – 5:00

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)	

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>?***

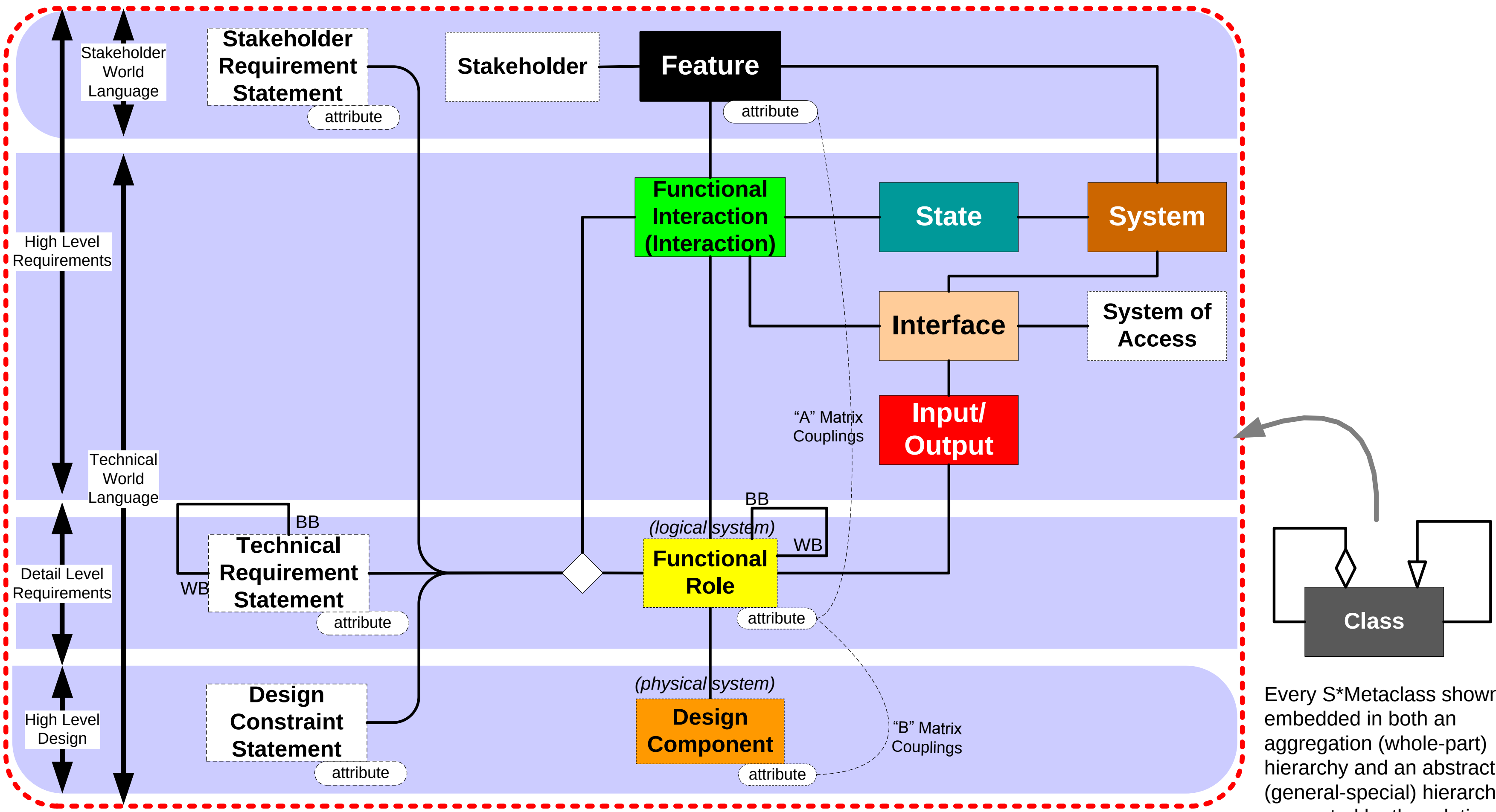
- Our most active members come from across diverse domains:
  - Automotive
  - Advanced Manufacturing
  - Aerospace
  - Consumer Products
  - Defense
  - Health Care, Medical Devices, Pharmaceuticals
  - Others
  - Today’s attendees?

- 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

What does the Patterns Challenge Team *<u>do</u>*?

- **This Challenge Team is concerned with *<u>configurable, re-usable system models</u>*, called “S*Patterns”:**
  1. Models containing a certain minimal set of elements are called <u>S*Models</u> (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 <u>S*Patterns</u>
  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

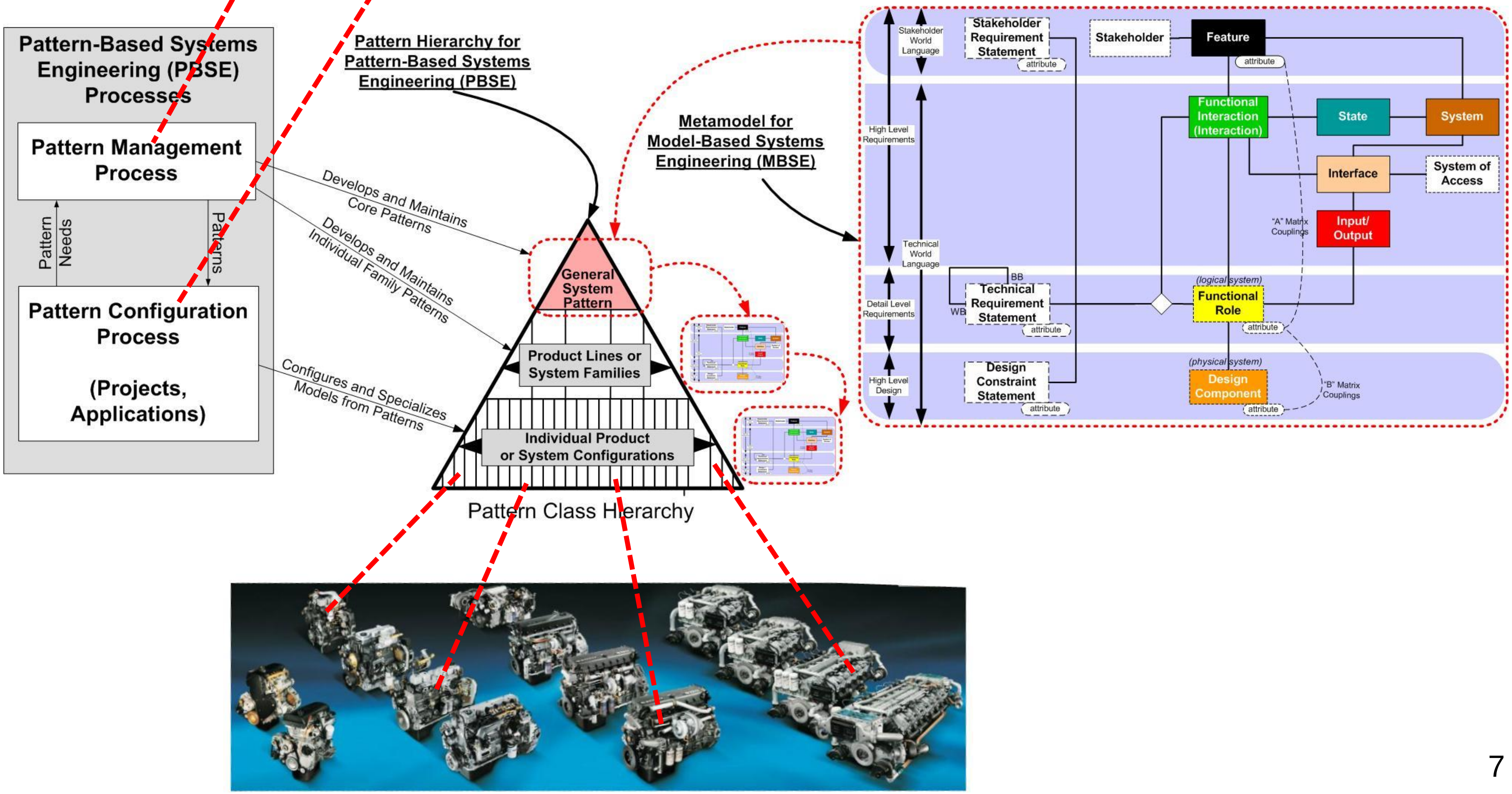
Summary of some major S*Metamodel classes and relationships—the underlying semantics of all S*Models (Refer to S*Glossary for definitions)



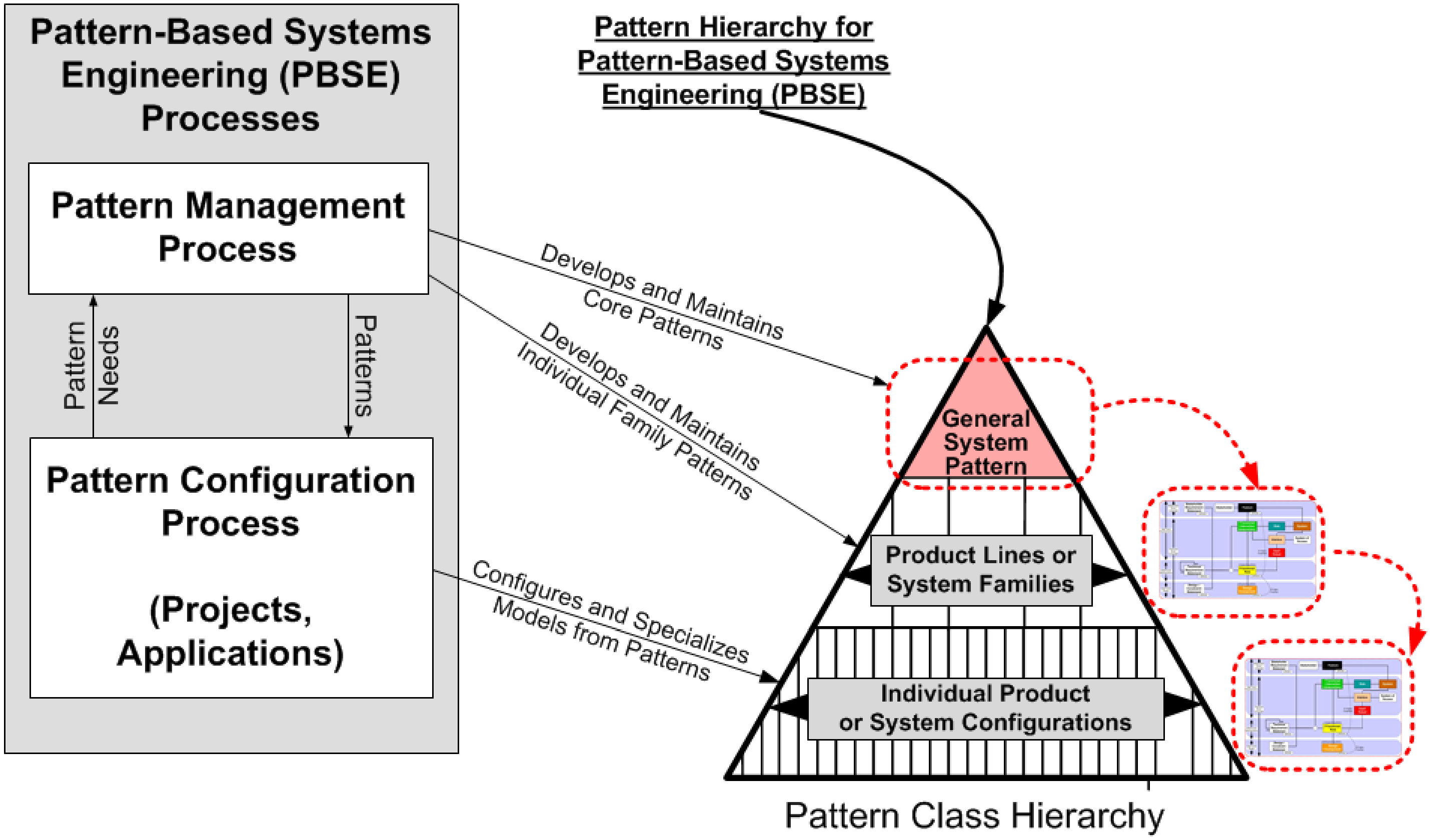
Every S*Metaclass shown is embedded in both an aggregation (whole-part) hierarchy and an abstraction (general-special) hierarchy, connected by the relationship types shown.

Summary of S*Metamodel

- 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?

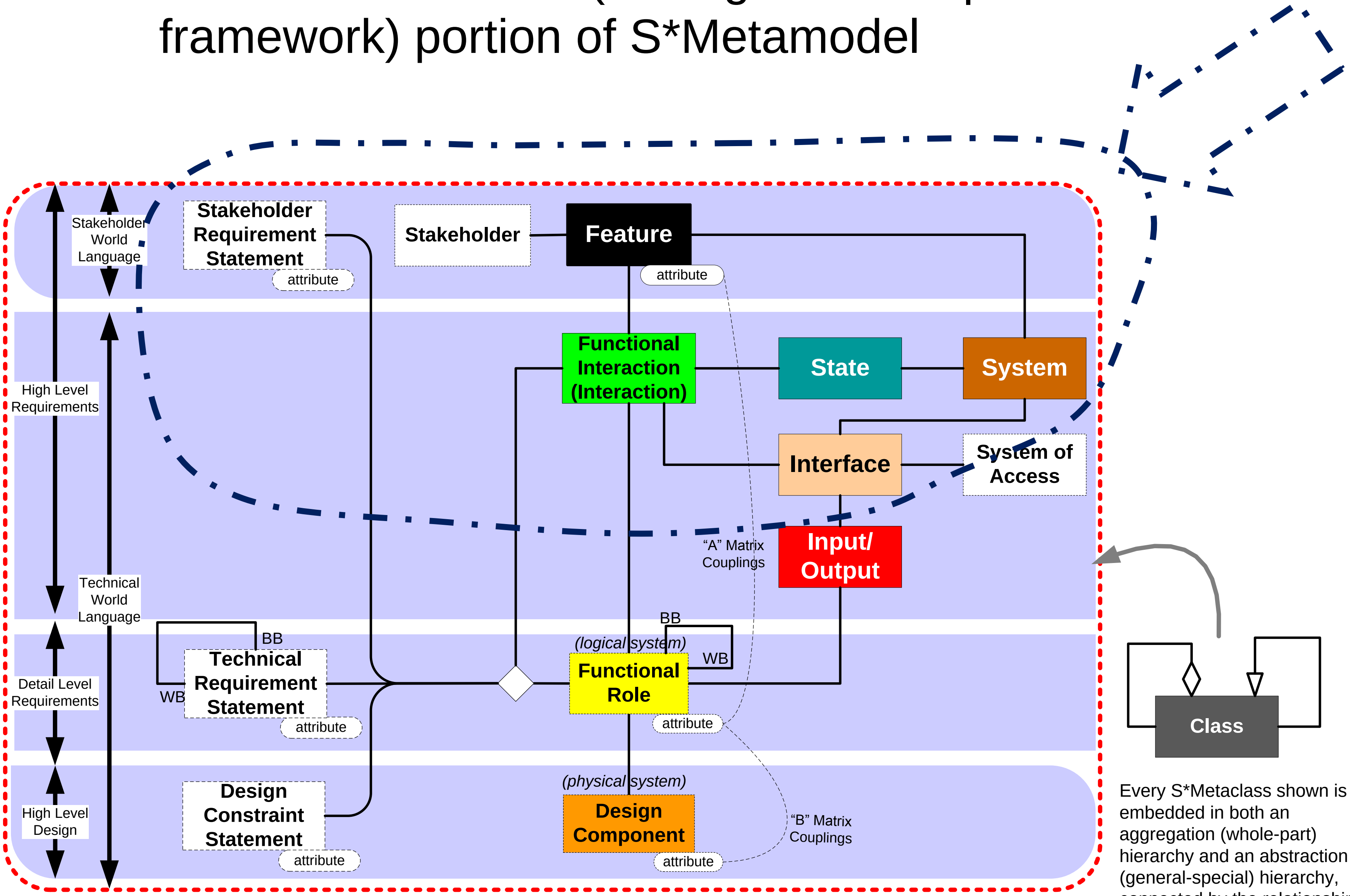
Team Announcement and Updates

- Our team's co-chair, Troy Peterson, named INCOSE Asst. Director for SE Transformation to MBSE
- INCOSE Great Lakes Regional Conference (GLRC9) 2015: Cleveland, October 23-25, 2015; submissions, registration: 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
- Updated PBSE Methodology Summary, based on team feedback (thank you!), for INCOSE MBSE Methodologies submission
- Agile SE Life Cycle Model (ASELCM) Project (joint w/Agile WG) host enterprise workshops to begin August; five orgs in pipeline
- SE Social Network Pattern Project started (Hoffman), for presentation at GLRC2015 in October
- Health Care Delivery Pattern Application (Joint with Health Care WG) started (Thukral), for presentation at GLRC2015 in Oct
- INCOSE Chapter presentations on PBSE: Northern Ohio (May 19); Previous: Crossroads of America, Chicagoland, Enchantment, Michigan, Finger Lakes
- Other announcements or updates?

Today's Session:
Four Related PBSE Technical Subjects

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

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

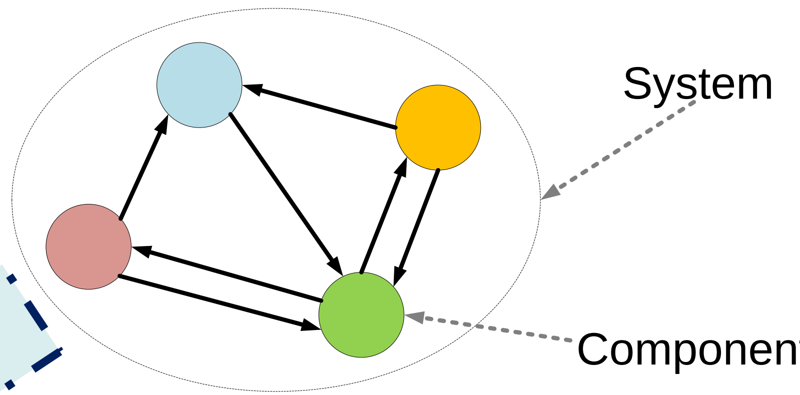


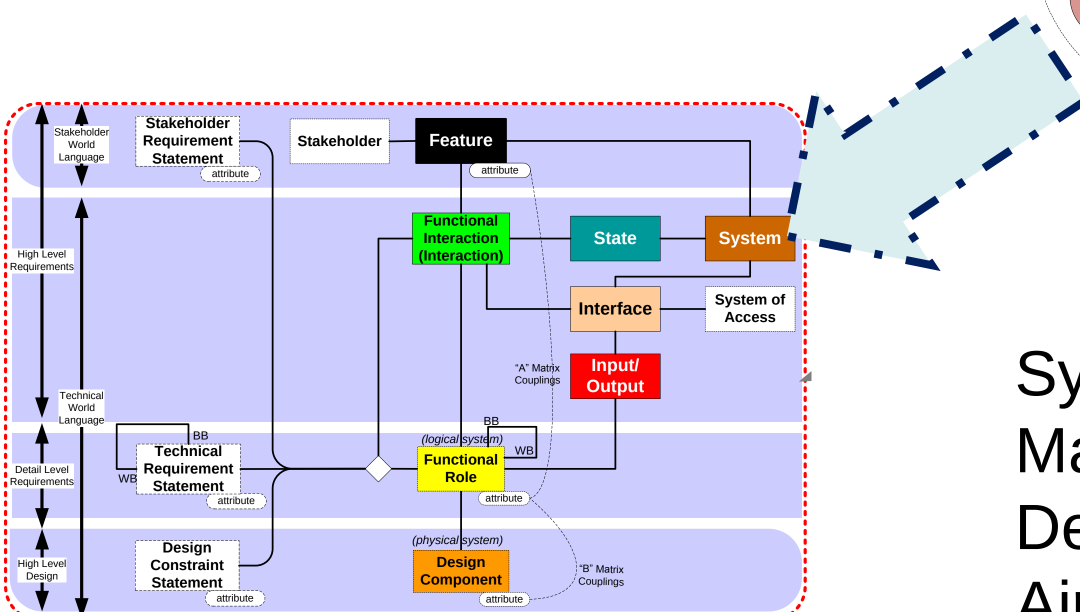
Every S*Metaclass shown is embedded in both an aggregation (whole-part) hierarchy and an abstraction (general-special) hierarchy, connected by the relationship types shown.

- 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.

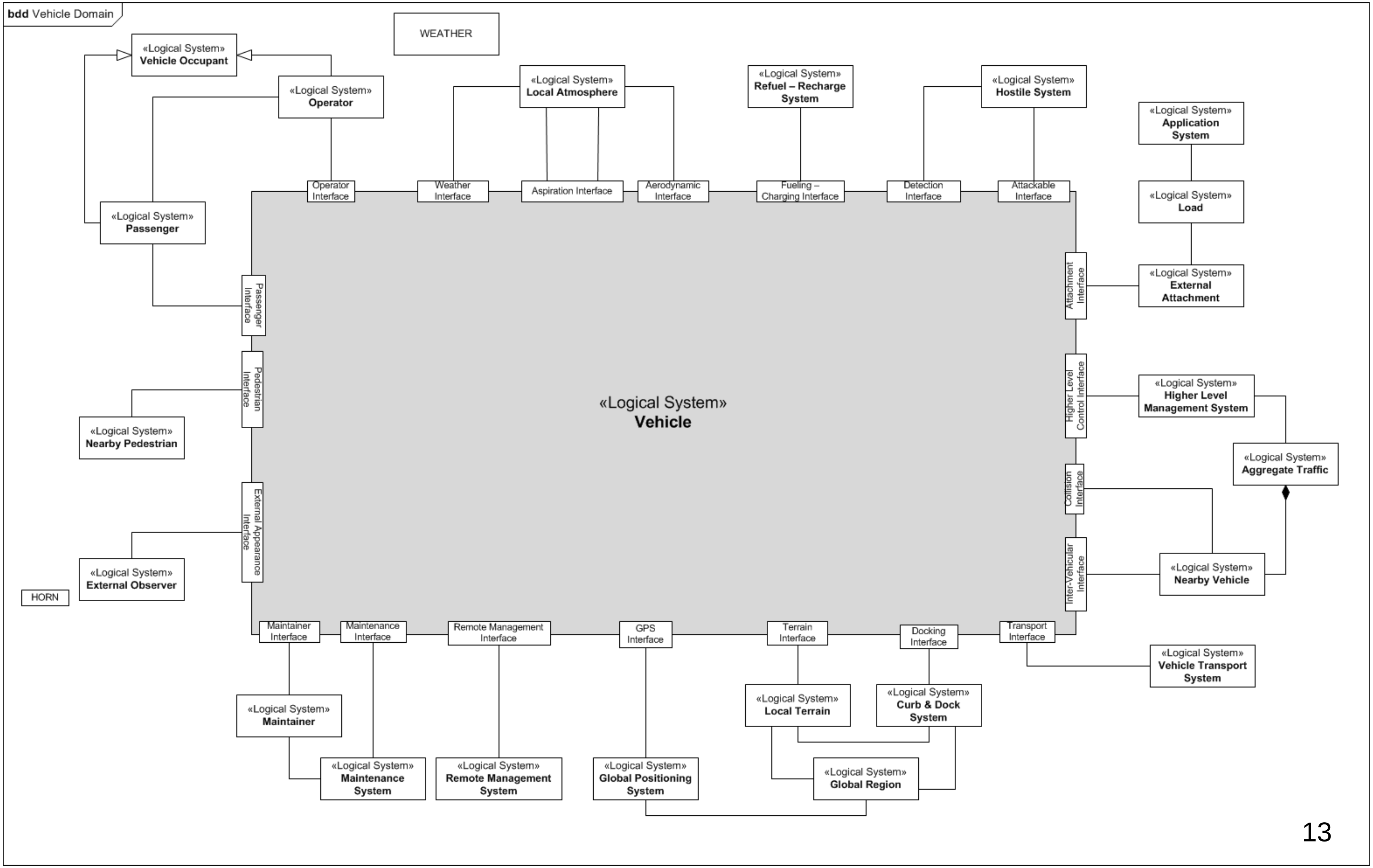
- A Component can be a System.



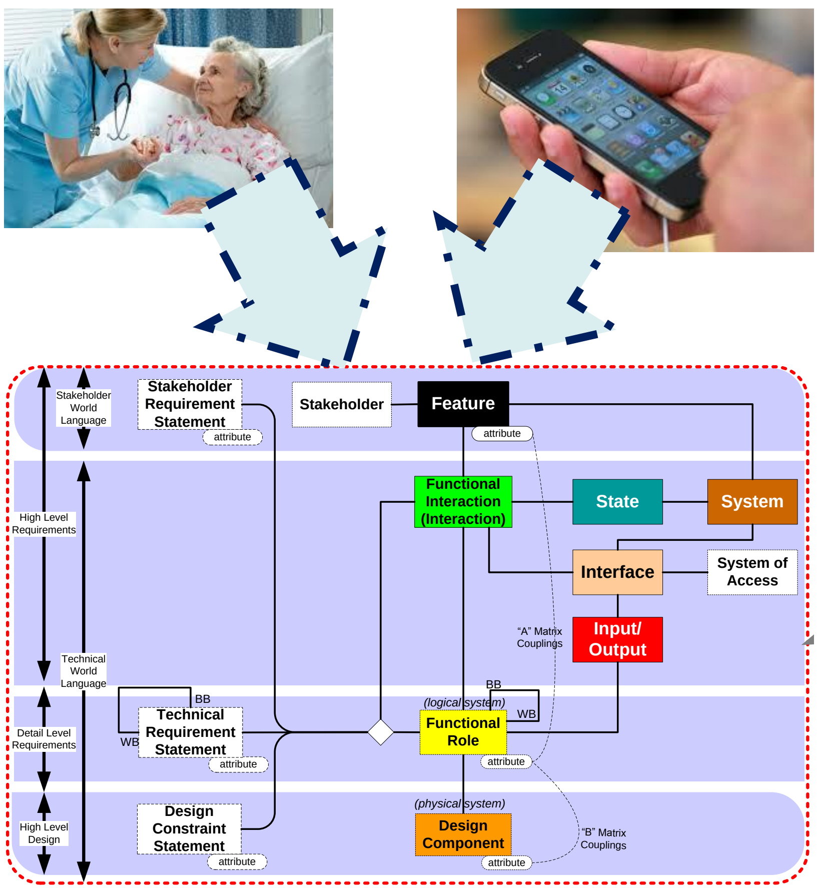


Systems: Engine, Vehicle, Manufacturing Line, Medical Device, Consumer Product, Aircraft, Engine, etc.

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



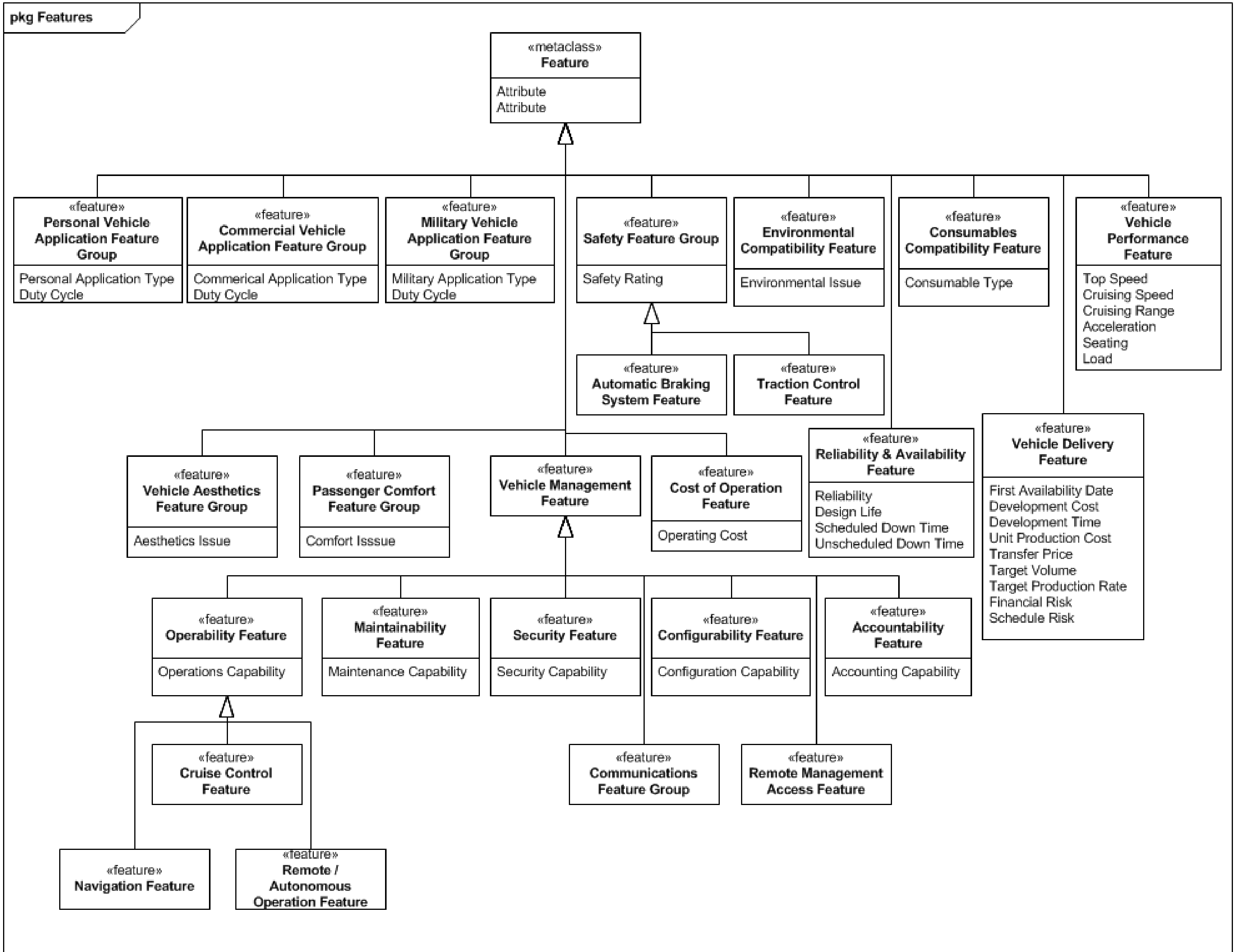
- A Stakeholder is a person, organization, community, or other entity with a stake in the behavior of a system.
- A Feature is a system behavior or capability having value to a Stakeholder, described in Stakeholder concepts & language.
- Features are the basis of Stakeholder selection of systems.



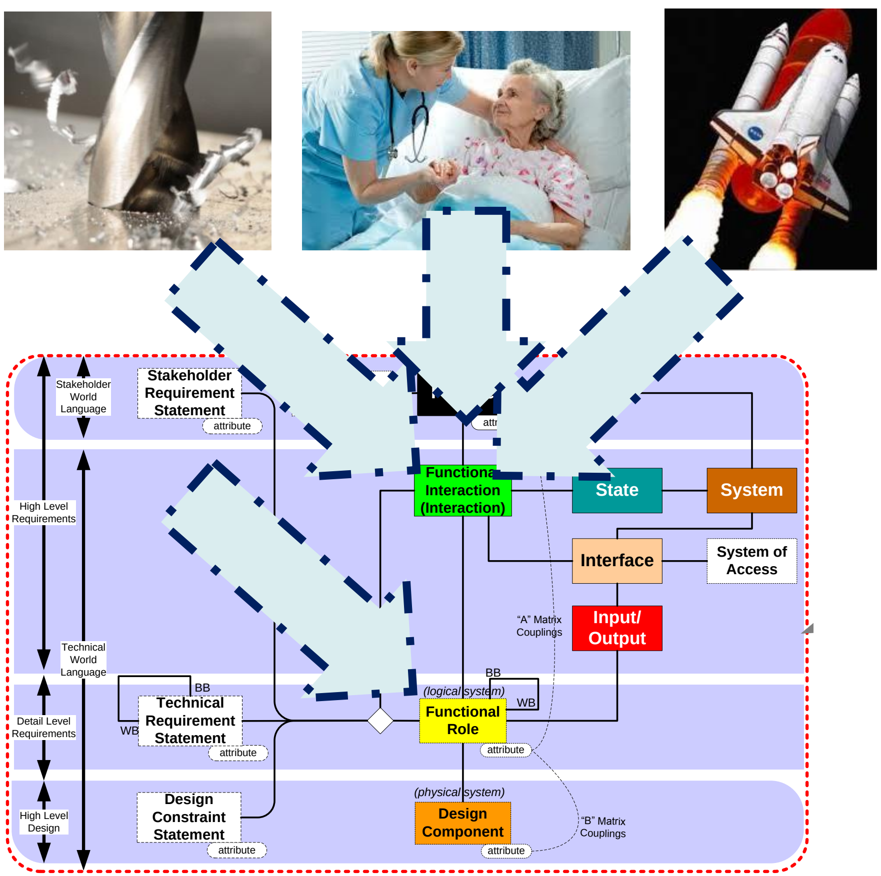
- Features are parameterized by Feature Attributes.
- These measures of effectiveness are in Stakeholder terms, so are frequently subjective and non-technical.

Features: Patient Monitoring, Threshold Detection, Production Capacity, Storage Capability, 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)



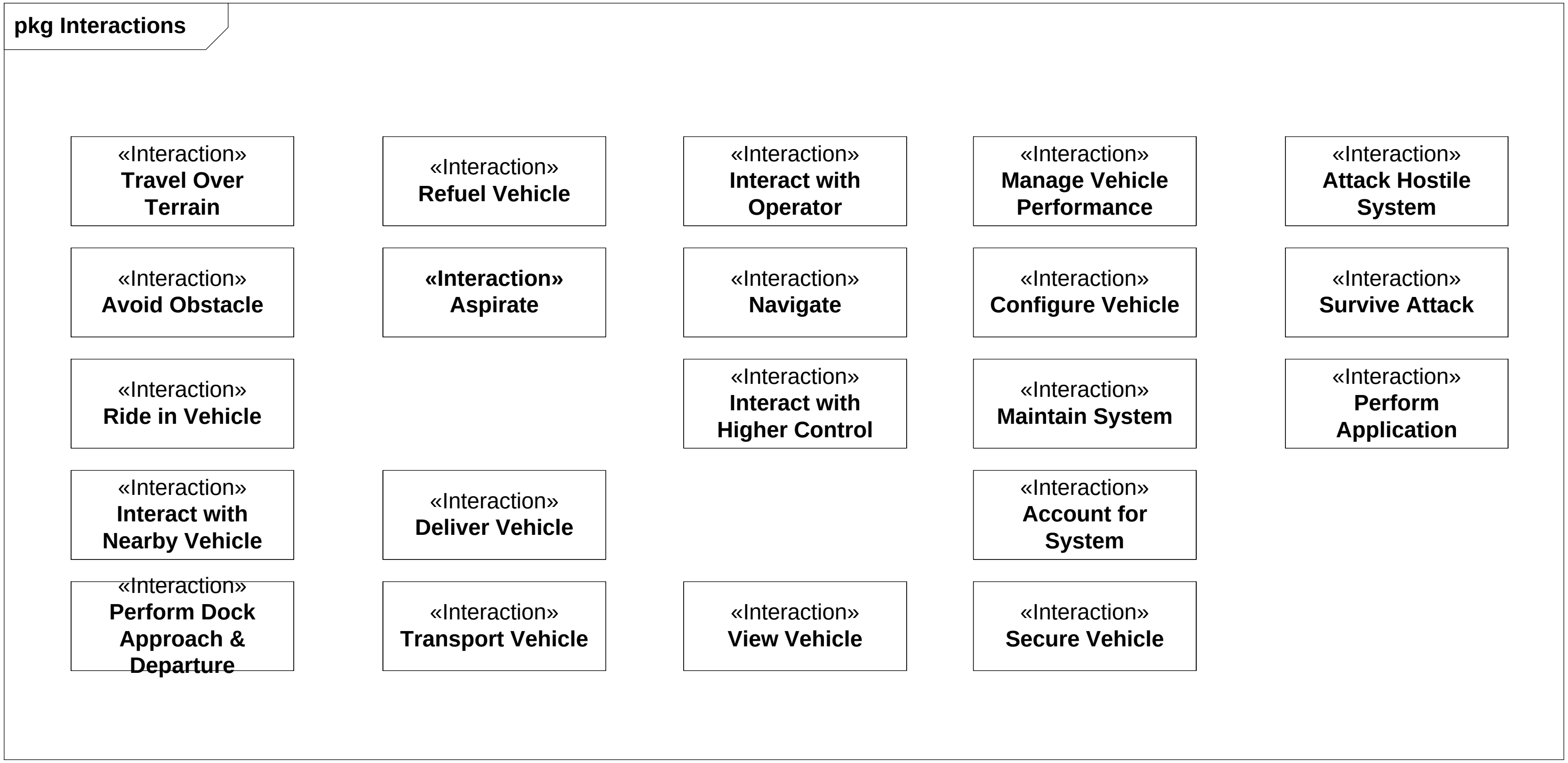
- A (Functional) <u>Interaction</u> is an exchange of energy, force, mass, or information, by two or more entities, said to play (Functional) <u>Roles</u> in the Interaction.
- All behavior occurs in the context of Interactions.



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

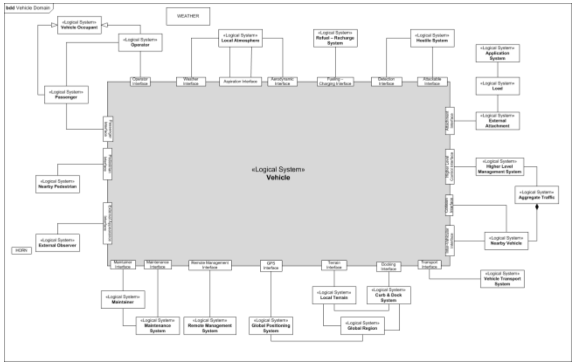
Interactions: Bore Hole, Examine Patient, Perform Ascent

- 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.

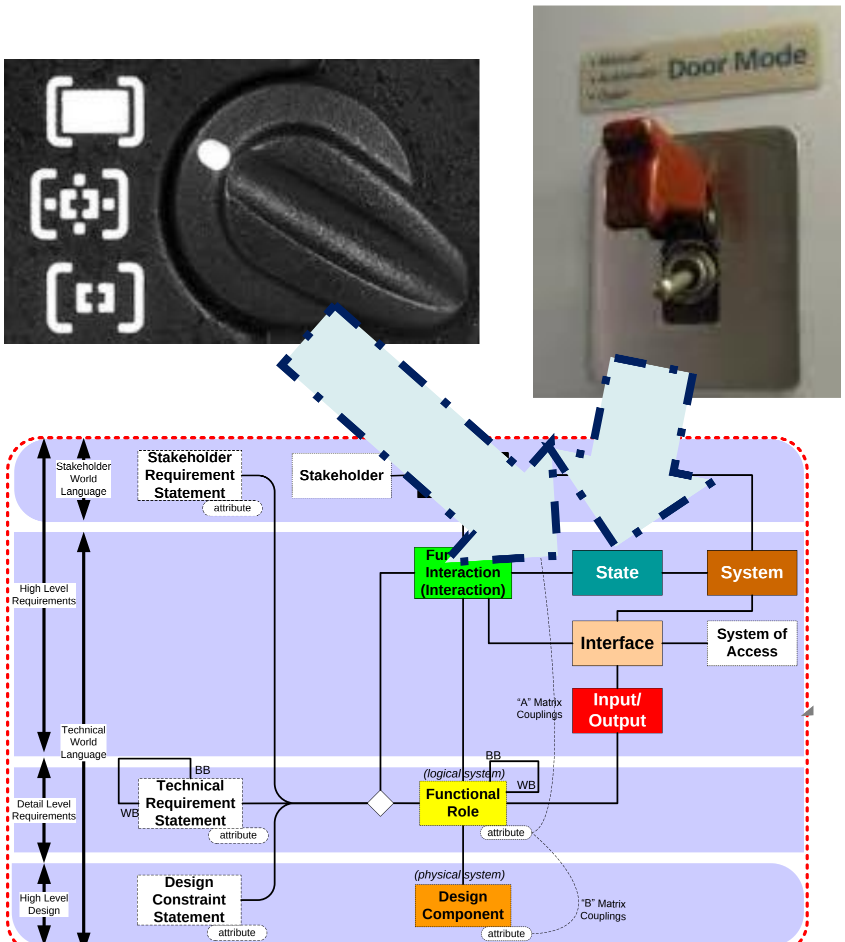


Vehicle Interactions:
Which Actors Participate in Interaction?

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																



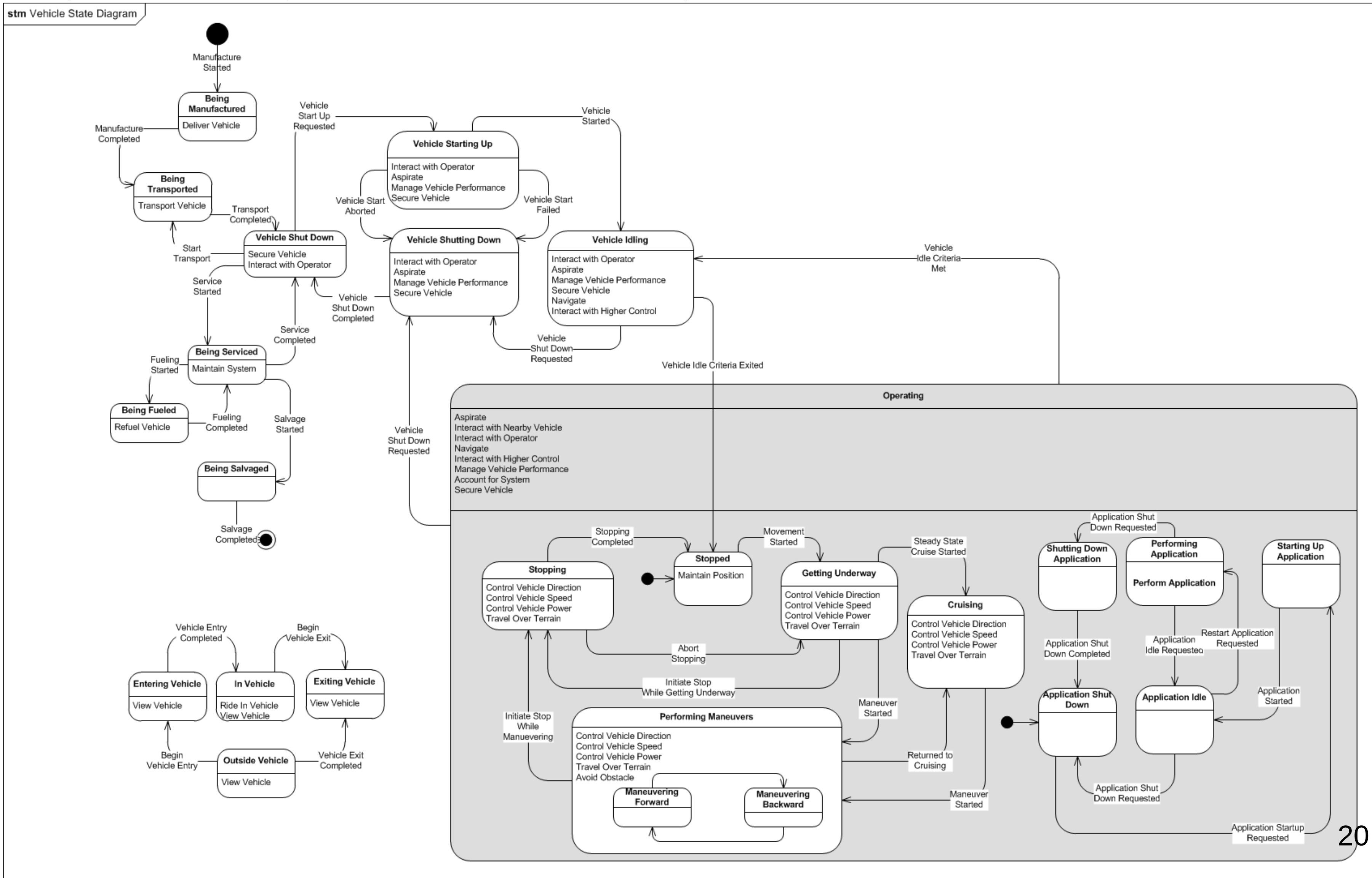
- A State is a condition of a system that determines its future behavior.
- Some state variables are continuous (e.g., position, velocity), and others are discrete (e.g., operational states).
- For the discrete case, Finite State Machine models are used.



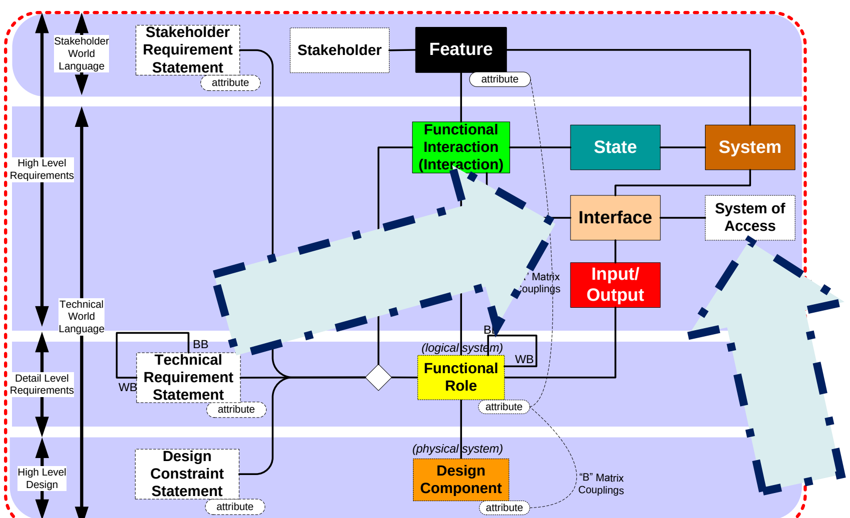
- The fact that different behavior is expected in the different (finite) states is represented by associating different Interactions with different (finite) States.

Finite States: Idle Mode, Off, Powering Up, Cruising, Recharging, Opening

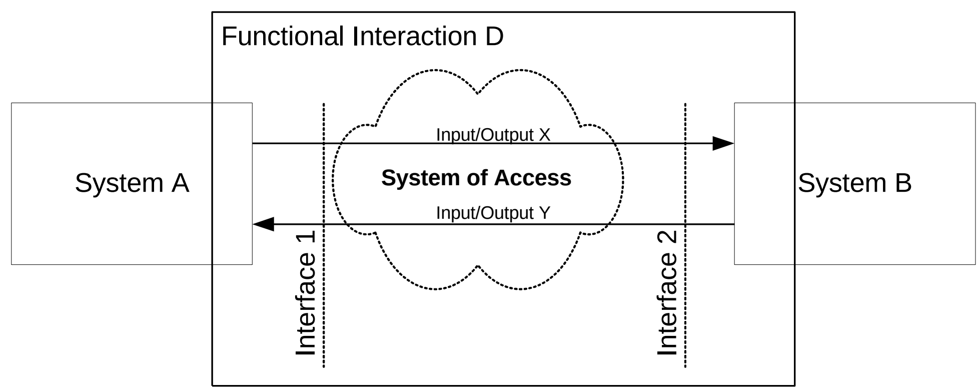
- In the High Level Requirements (HLR) subset of an S*Model, the State Model establishes a high level temporal (time) model of the system.
- The scope of such a State Model may be the entire System Life Cycle, an Operational Cycle, or other time scope.



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).

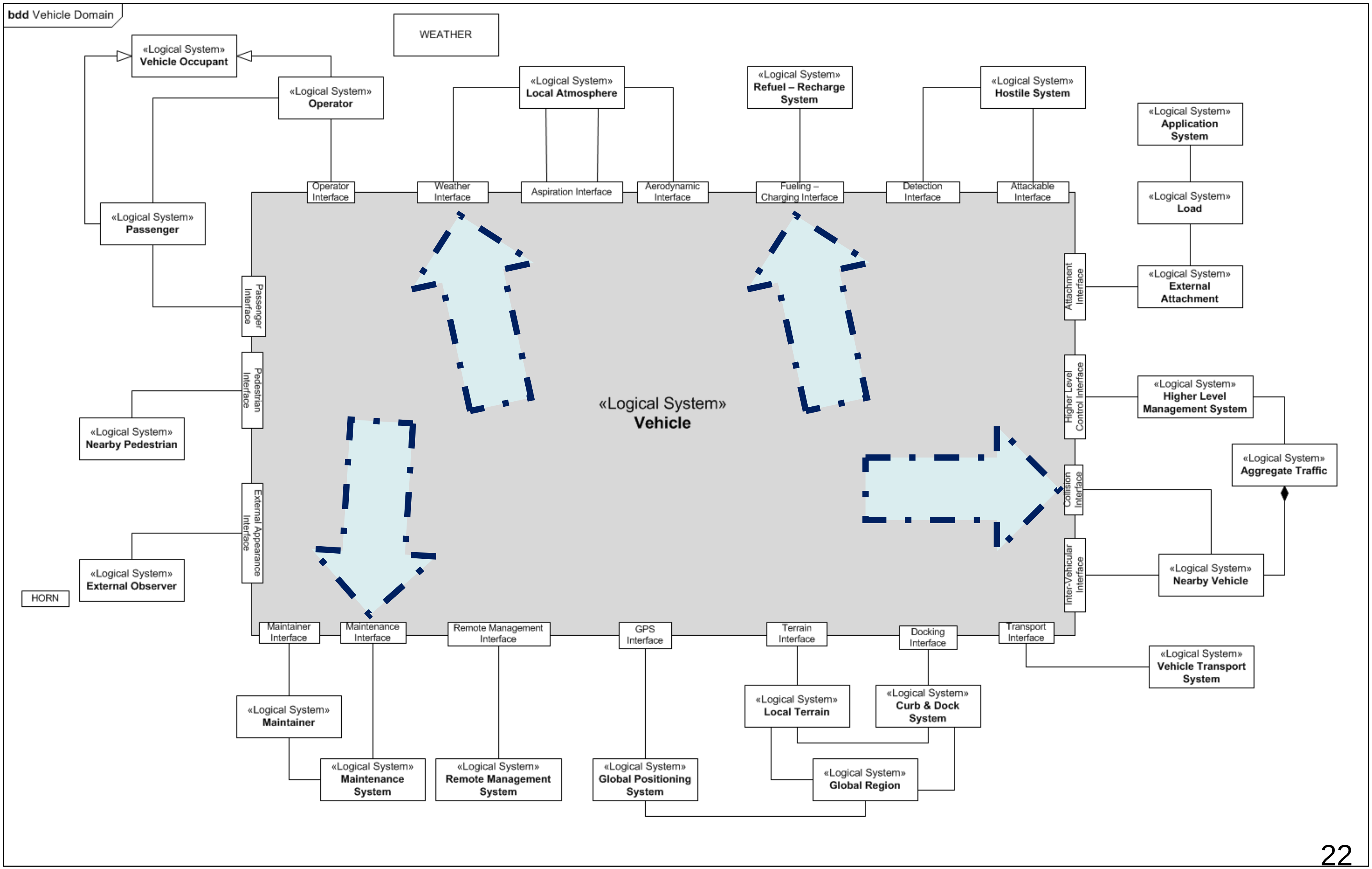


S*Models show that there are multiple interfaces between systems:

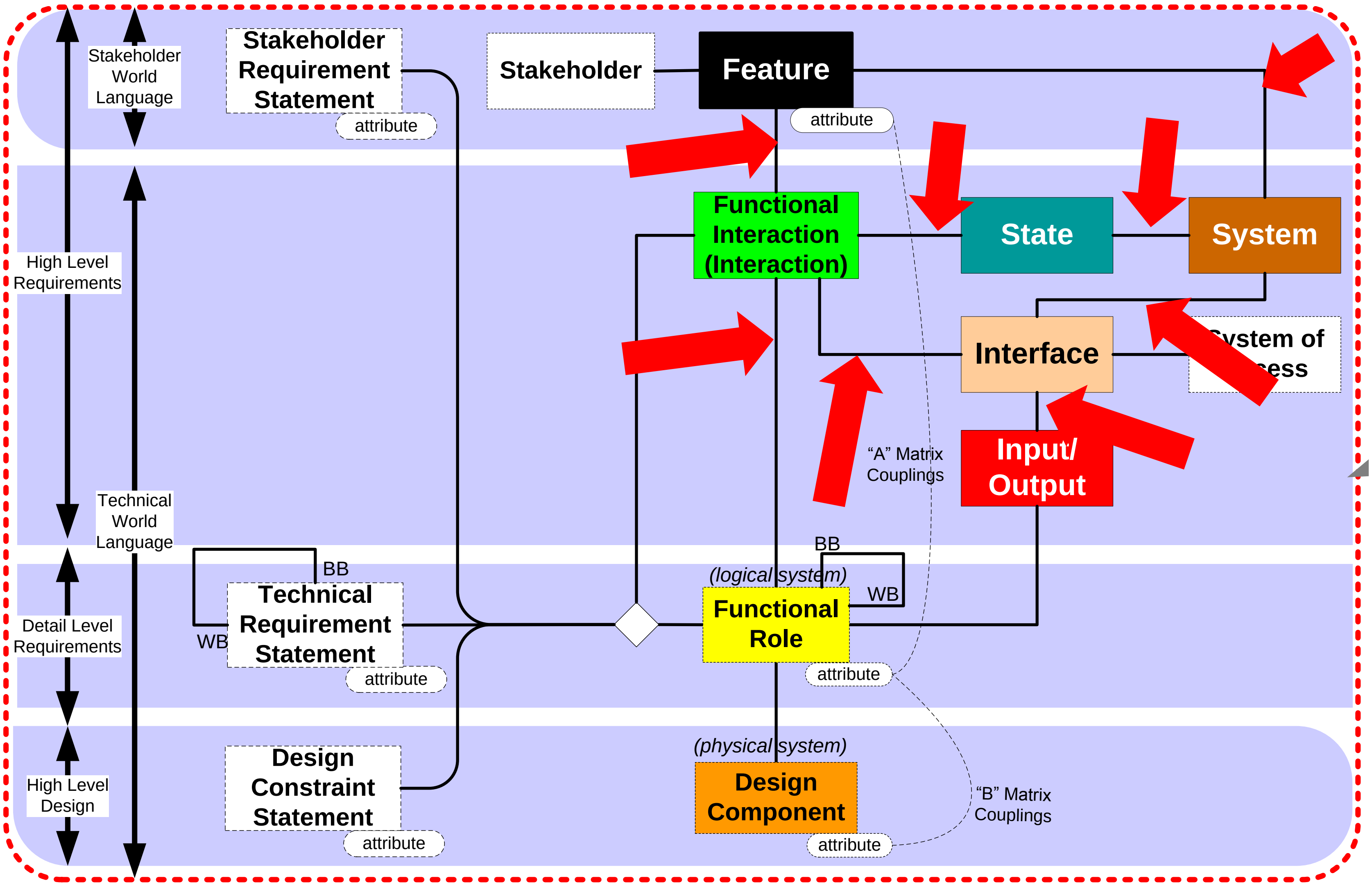


Systems of Access: Cabling, Hydraulic Lines, Internet, Mouse Driver, Pedals, Keyboard.

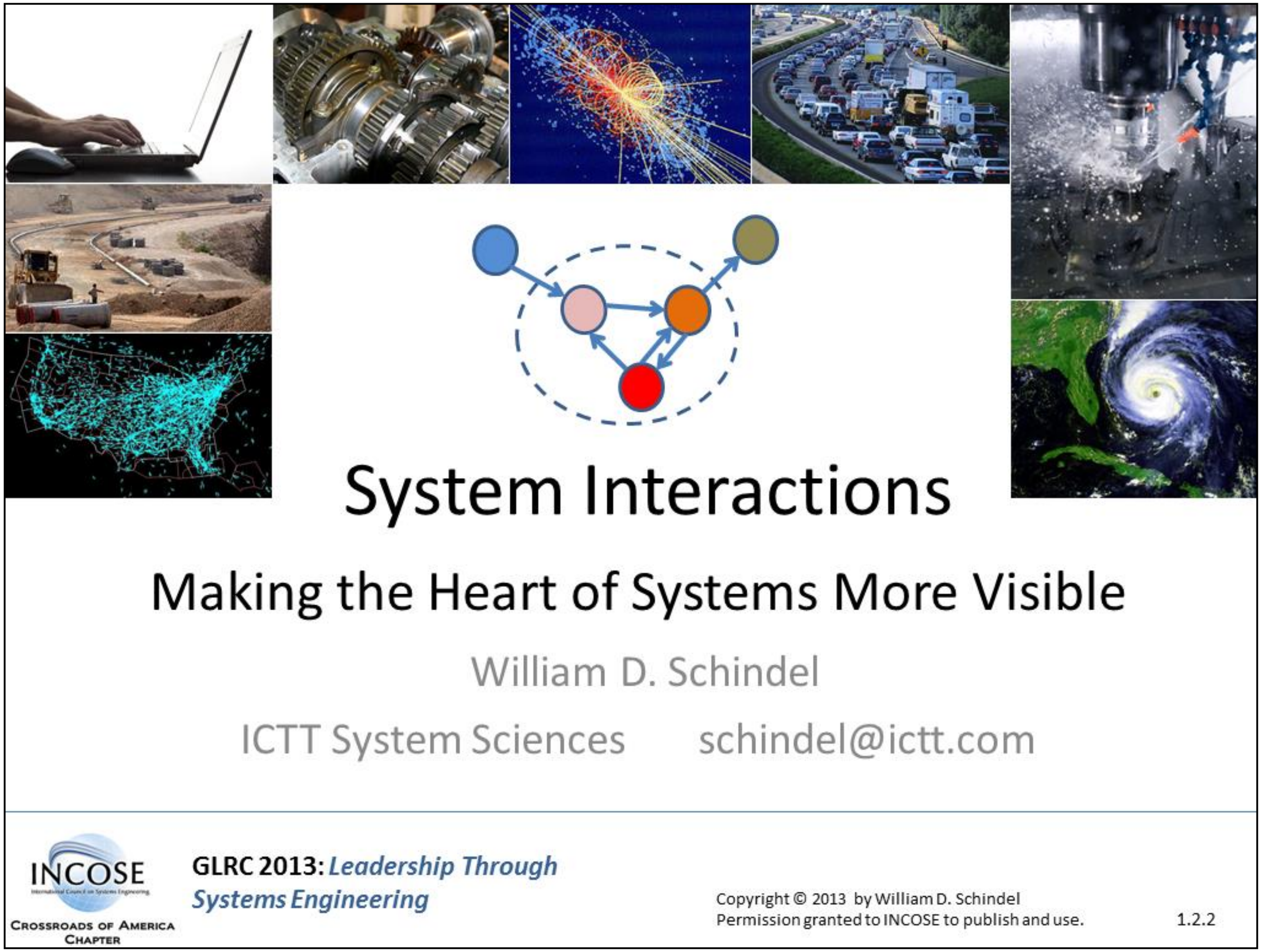
In S*Models, external Interfaces can appear at the edge of systems (i.e., in Domain Diagrams), and queries can be used to generate Interface Control Document (ICD) views.



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):

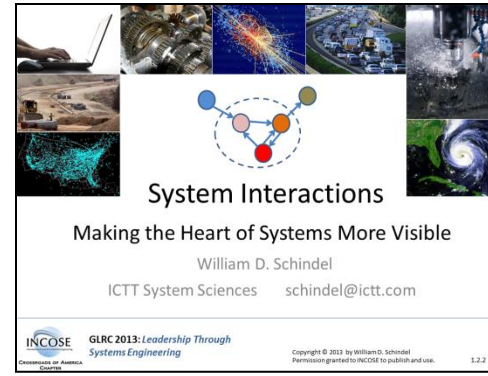


Criticality of modeled Interactions to the heart of MBSE, PBSE, science and engineering



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

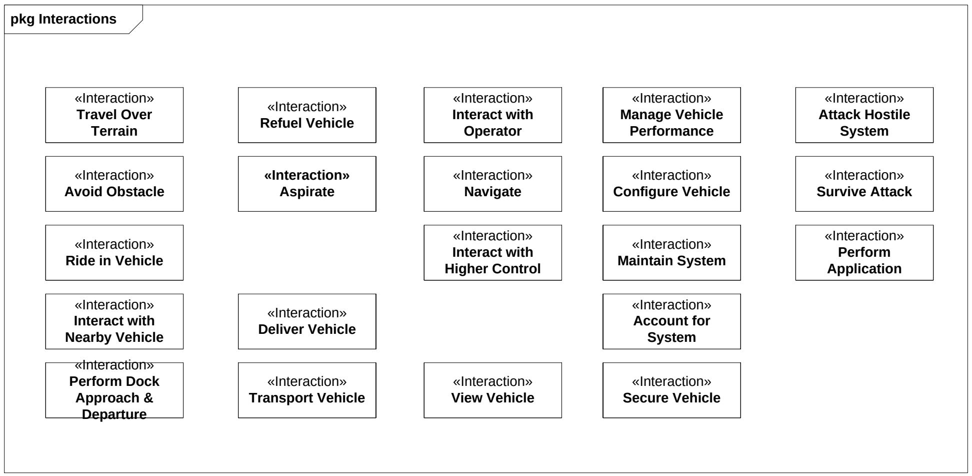
For more detail, see -->

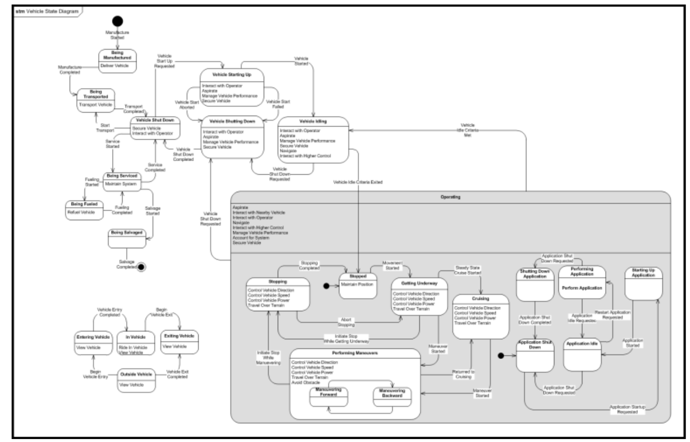


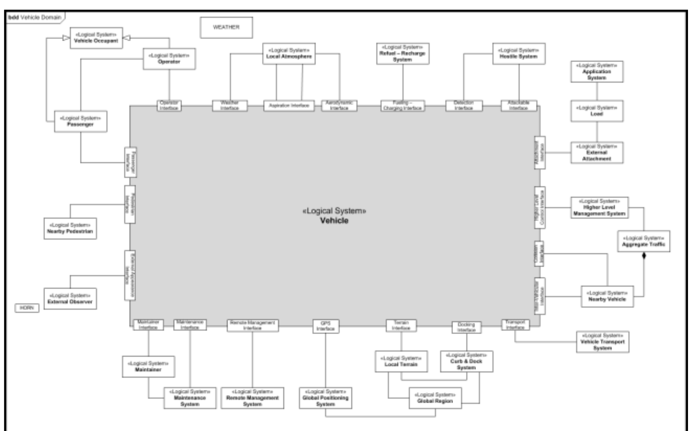
- In a nutshell:
  - Physical interaction models provide the context for all the laws of the hard sciences (Newton, Maxwell, Boltzmann, etc.).
  - Explicit models of physical interactions are perfectly legal in MBSE models (collaboration, activity, etc.), but are frequently under-emphasized in them.
  - 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):



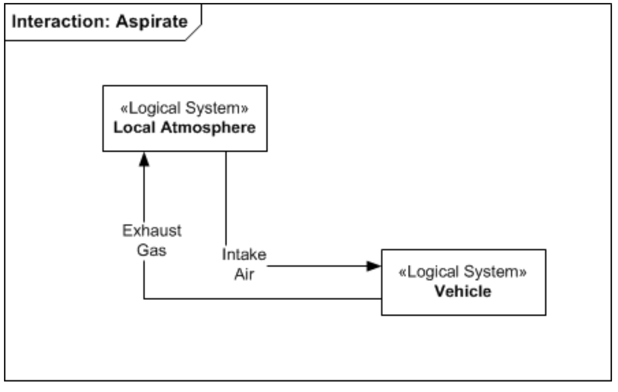


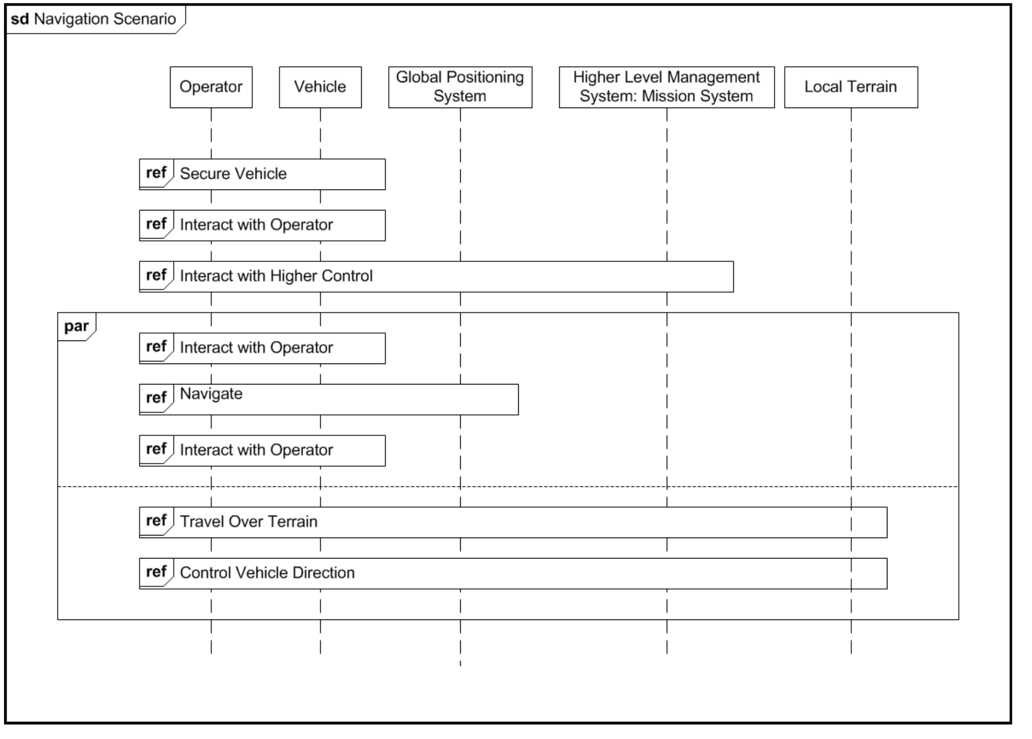


	Interaction Name	Interaction Definition	Vehicle	Operator	Passenger	Vehicle Occupant	Nearby Pedestrian	External Observer	Maintainer	Maintenance System	Local Atmosphere	Refuel System	Hostile System	External Attachment	Load	Application System	Higher Level Management	Nearby Vehicle	Vehicle Transport	Curb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

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.

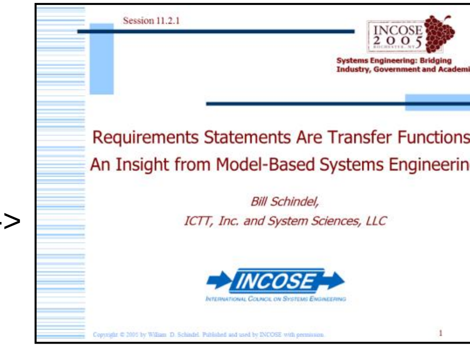
Systems Engineering Prose

Traditional Requirements Discipline. Composing good requirements statements prose has a

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

Understanding Requirements Statements as non-linear Transfer Functions

For more detail, see -->


Session 11.2.1

INCOSE 2005

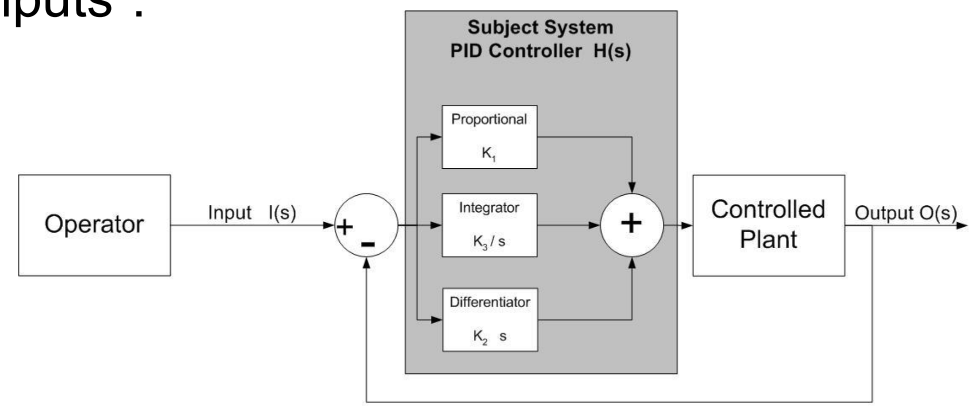
Systems Engineering: Bridging Industry, Government and Academia

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

Bill Schindel,
ICTT, Inc. and System Sciences, LLC

INCOSE

- 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 . . .

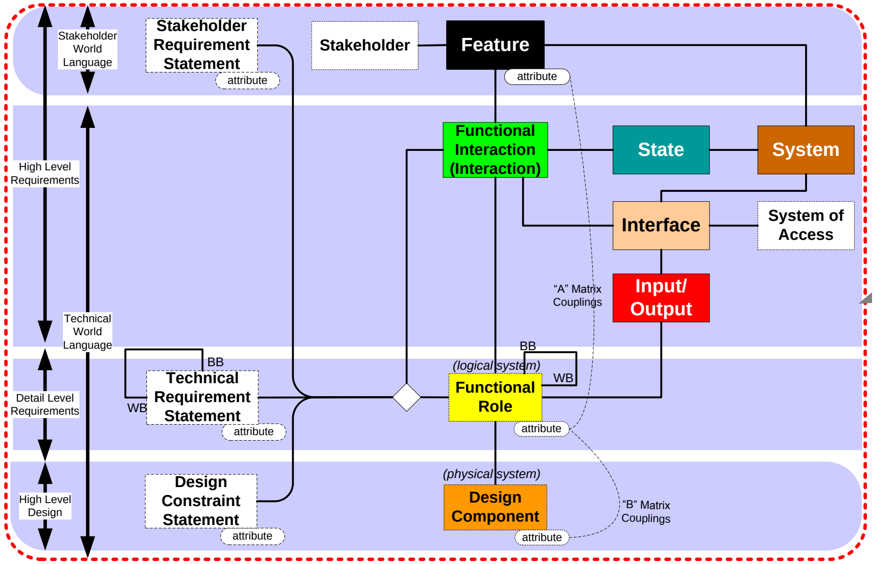
Understanding Requirements Statements as non-linear Transfer Functions

- 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.)

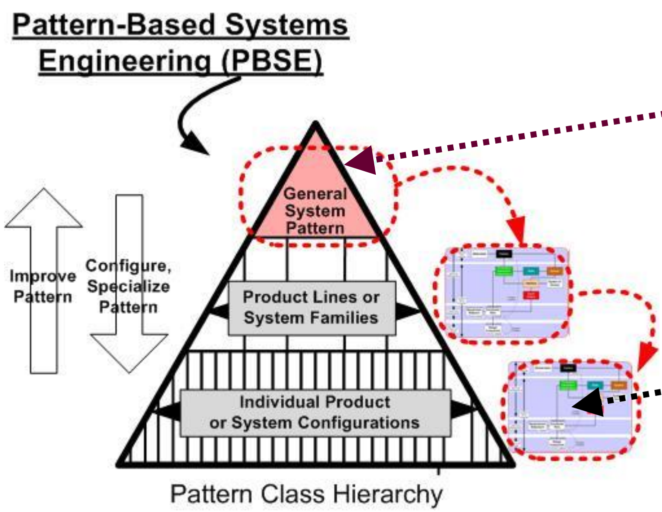
Gestalt Rules in PBSE – and their connections to the above and applications in understanding system patterns

- The above three discussions were about a web of systemic relationships in the descriptions (models) of systems:
  1. Relationship webs in the HLR models of systems
  2. Relationship webs in the DLR models of systems
  3. Including understanding Requirement Statement prose as a form of input-output Relationship (non-linear Transfer Function)
- What does this tell us about patterns (repeating regularities) that are <u>systemic</u> patterns—not just patterns of <u>parts</u>?



Gestalt Rules in PBSE – and their connections to the above and applications in understanding system patterns

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



Governing pattern

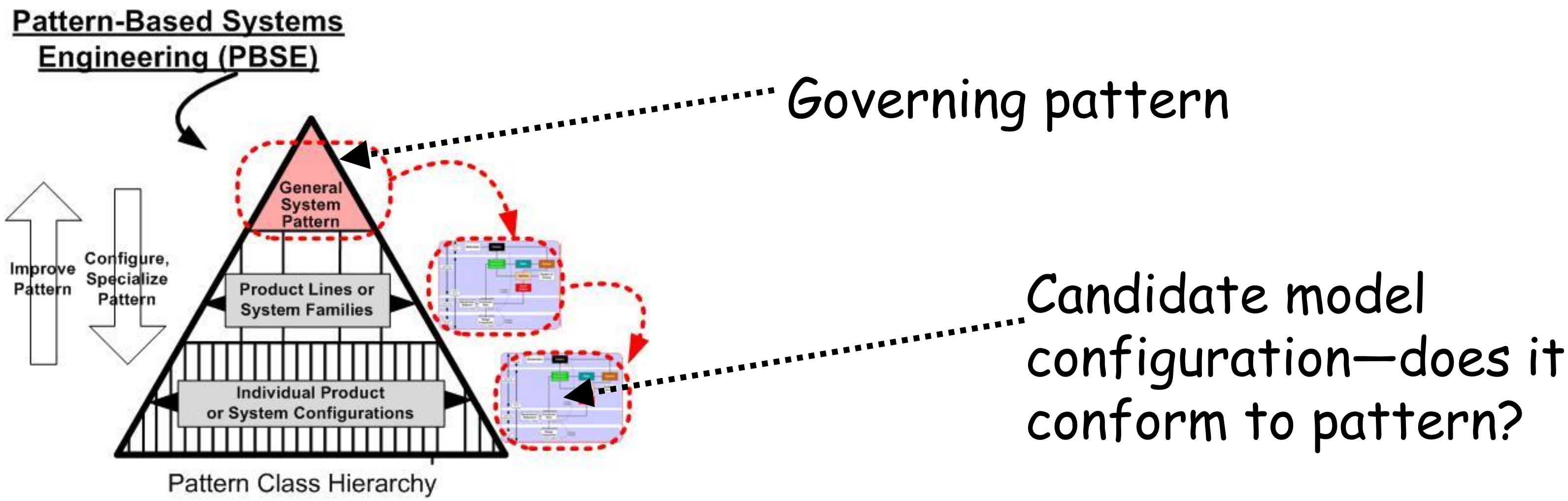
Candidate model configuration—does it conform to pattern?

Systematica™

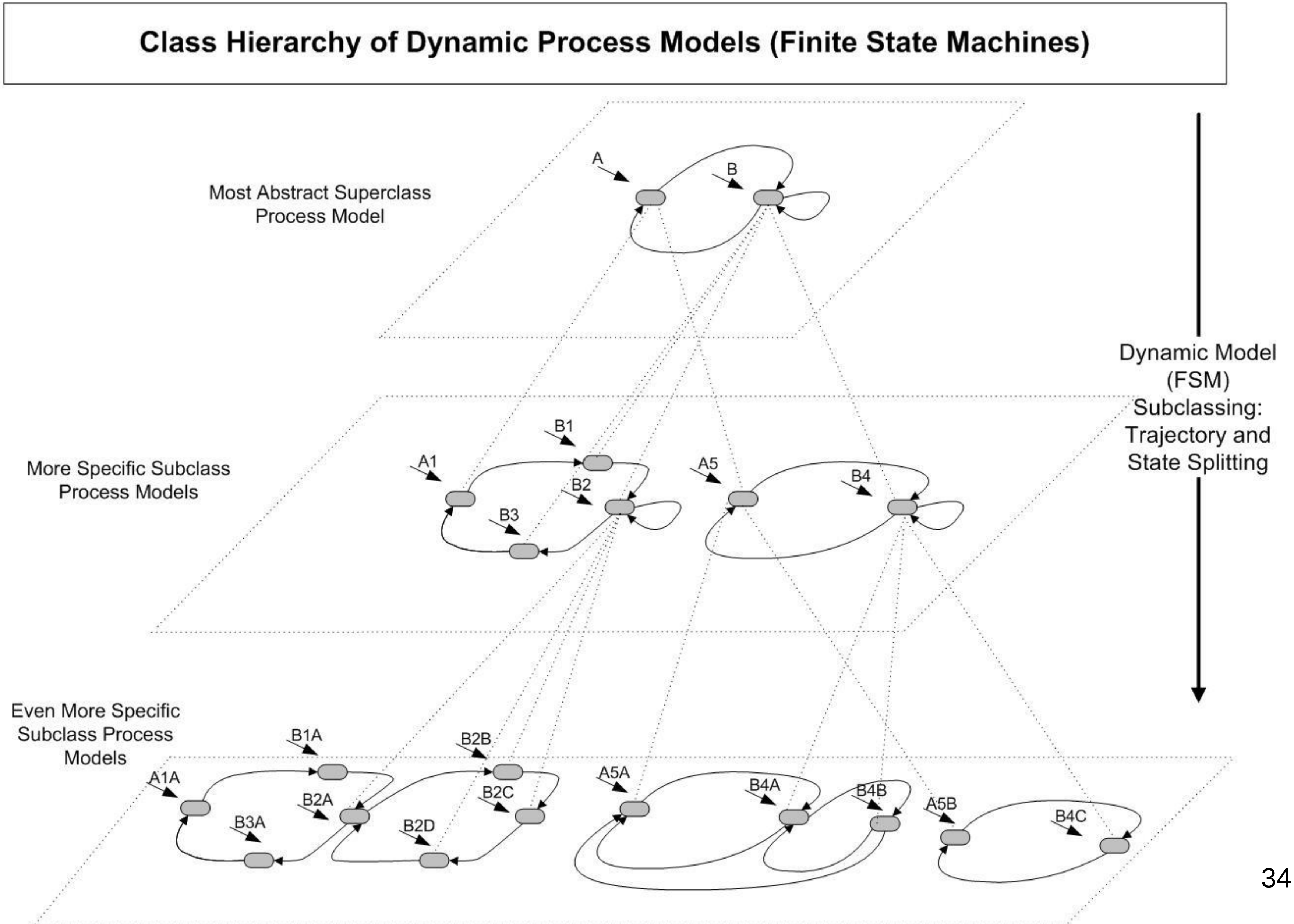
Do more with less

The Gestalt Rules

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



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



Planning Discussion: Next and Future Activities

- Discussion of candidates for focus of Patterns Challenge Team meeting at IS2015 in July; meeting schedule for same
- 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?
- Next Team Meeting: Tuesday, June 16, 4:00 PM EST