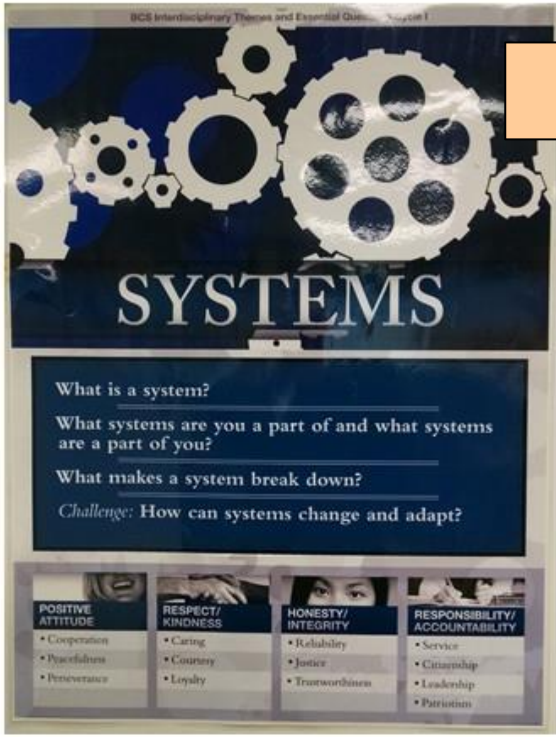
INCOSE 2016 international workshop
Los Angeles, CA, USA
January 30 - February 2, 2016

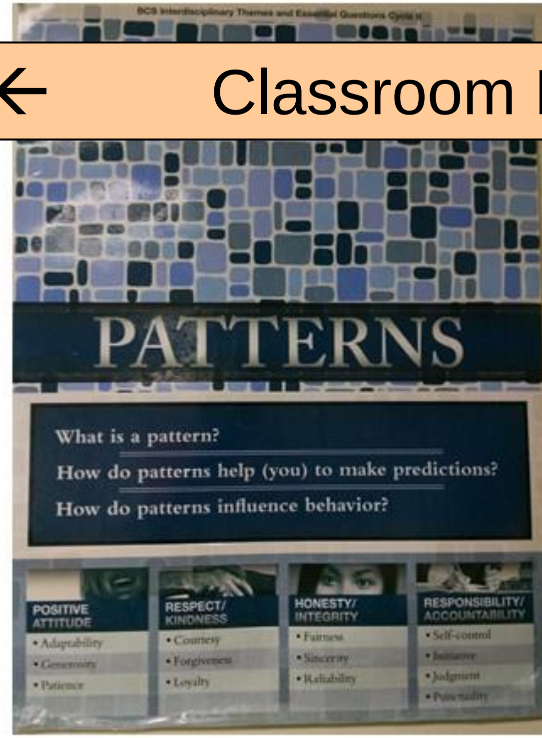
INCOSE Patterns Working Group

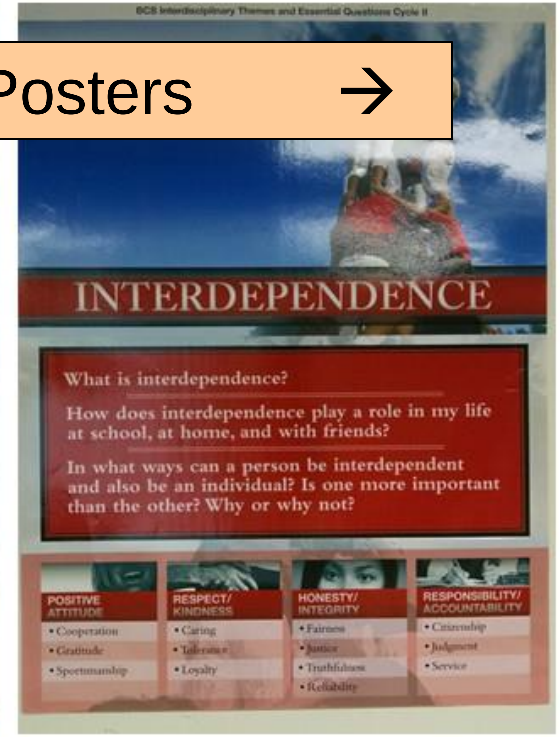
Meetings: Jan 30-31, 2016 (during IW2016)

Working group web site:
http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns
Meeting web site:
http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns_challenge_team_mtg_01.30-31.16 V1.2.1

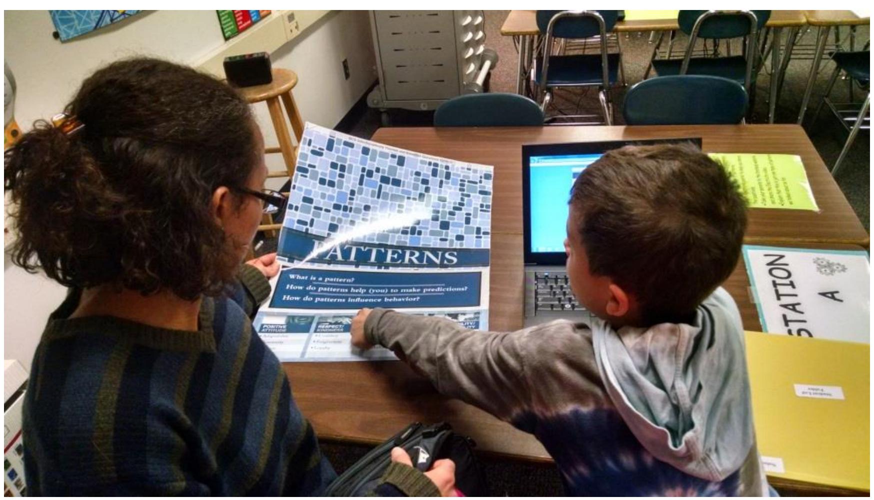
Be prepared for the next generation.







← Classroom Posters →



Troy Peterson's Son, Third Grade, 9 years old
Birmingham Covington School, Michigan

"Integrated learning is a critical element in all classrooms at BCS. The ability to understand big themes common to human experience, past and present, provides students with a framework for thinking about themselves and their world. The themes and essential questions for this year were carefully chosen to connect and link all curriculum areas."

http://www.bcsonline.info/Why_BCS/Curriculum.html

Summary of Patterns WG activities at IW2016:

- Half day joint workshop with SoS WG, on Patterns in Systems of Systems: Saturday, Jan 30, 1330-1730 Pacific Time US
- 90+ minute joint session with PLE WG, on Harvesting Product Line Patterns from Legacy Products: Sunday, Jan 31, 1300-1500 Pacific Time US
- 90 minute joint session with Agile WG, review of Agile Systems Engineering Life Cycle Pattern: Sunday, Jan 31, 1530-1700 PT
- Regular meeting of Patterns WG, including updates and planning: Sunday, Jan 31, 1300-1730 PT

Agenda . . .

Meeting(s) Agenda: INCOSE MBSE Patterns Working Group at INCOSE IW2016 (Web conferenced)

Saturday, Jan 30, 13:30-17:30 Pacific Time—Joint with SoS WG; **Sunday**, Jan 31, 13:00-17:30 Pacific Time—Joint with PLE and Agile WGs

Patterns WG General Background Reading: Team web site: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns
Minutes of IS2015 meeting of July 12-13, 2015: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns_challenge_team_mtg_01.12.15
PBSE Methodology Summary from the Patterns WG: http://www.omgwiki.org/MBSE/doku.php?id=mbse:pbse

Saturday, Jan 30, 2016 Joint workshop of Patterns WG and SoS WG: Patterns in Systems of Systems Workshop pre-reading, slides, and remote participant access information posted at: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns_challenge_team_mtg_01.30-31.16		**13:30-17:30 PT**
Introduction, review of workshop objectives and agenda	(Joint)	13:30 – 13:40
Introduction to Systems of Systems	(SoS WG)	13:40 – 14:20
S*Patterns and their Application to SoS	(Patterns WG)	14:20 – 15:00
IW-Wide Break		15:00 - 15:30
Preparatory discussion of interesting issues arising from patterns and SoSs Structured interactive small group breakout sessions—to brainstorm and discuss suggestions	(SoS Patterns breakout facilitators) (Workshop participants)	15:30 – 17:00
Plenary discussion, reflecting on previous work and plans for potential future activities	(Joint)	17:00 – 17:30
Adjourn		17:30

Sunday, Jan 31, 2016 Includes joint Patterns WG sub-sessions with PLE WG and Agile WG Pre-reading and slides for each session, and remote participant web access information posted at: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns_challenge_team_mtg_01.30-31.16	13:00-17:30 PT
Meeting start up: • Review of meeting objectives and agenda • Introduction of participants • Overview of existing Patterns WG Projects	13:00 – 13:15 (Starts *immediately* after IW lunch)
Joint sub-session with PLE WG: Patterns in Product Lines • Introduction to the PLE WG • Use of PBSE to harvest formal MBSE Patterns from legacy products • Discussion of interest in a joint project (by PLE WG and Patterns WG) • Additional discussion in the PLE WG meeting, on Monday, Feb 1	13:15 – 15:00
IW-Wide Break	15:00 - 15:30
Joint sub-session with Agile WG: Introduction to the Agile System Life Cycle Model Pattern (from INCOSE ASELCM Project) • Introduction to the INCOSE ASELCM Discovery Project • Technical introduction to the ASELCM Pattern content—a formal S*Pattern being constructed in this project • Additional project status discussion in the Agile WG meeting, on Monday, Feb 1	15:30 - 17:00

Identification / Discussion of 2016 Projects
• Existing projects already in progress
• Additional projects of interest to members
  ○ Specific domain patterns
  ○ Captured science patterns
  ○ Mapping to ISO 15288—processes versus data
  ○ Mapping to COTS tools
  ○ Implementation strategies
  ○ Outreach to other WGs
  ○ Other interests from team members
• Meetings calendar
	17:00 – 17:30

For more information, contact-- Patterns WG: Bill Schindel schindel@ictt.com Troy Peterson peterson_troy@bah.com
SoS WG: Judith Dahmann jdahmann@mitre.org Agile WG: Rick Dove dove@parshift.com
PLE WG: Hugo-Guillermo Chale-Gongora hugo-guillermo.chale-gongora@transport.alstom.com

The INCOSE Patterns Working Group: ***<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 two years, over 100 colleagues have participated in Patterns Working Group 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

- Participants:
  - Sunday, Jan 30, joint workshop with SoS Working Group:
    - We had 28 on-site participants, plus 5 remote participants
  - Monday, Jan 31, joint workshop with PLE Working Group:
    - Introduction of participants
  - Monday, Jan 31, joint ASELCM Pattern Review w/Agile WG:
    - Introduction of participants

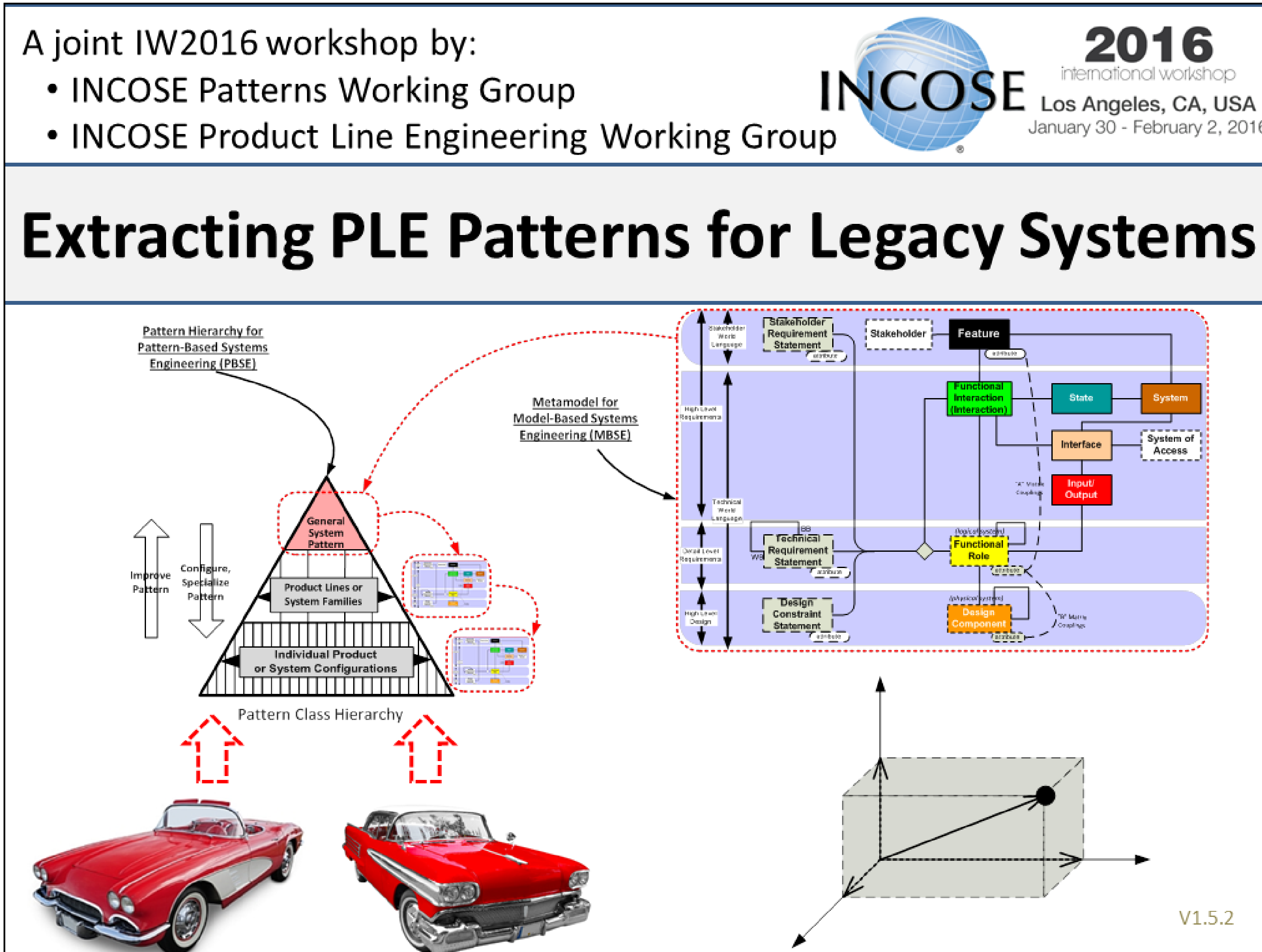
- **We began as the MBSE Initiative Patterns Challenge Team:**
  - **Those teams have now become INCOSE Working Groups**

- **This Working Group 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

WG Announcements and Updates

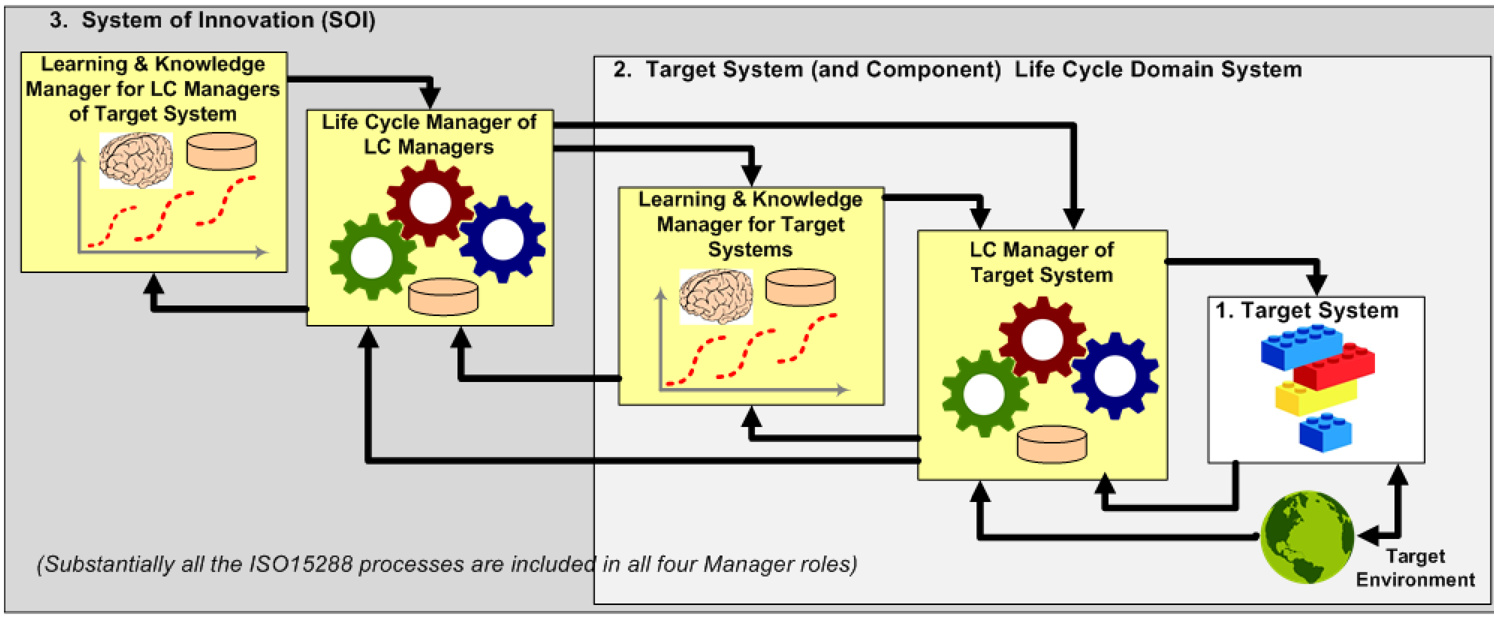
- MBSE Patterns Challenge Team has become the Patterns Working Group:
  - Patterns WG co-chair, Troy Peterson, now INCOSE Assistant Director for SE Transformation.
- INCOSE Great Lakes Regional Conference (GLRC9), Cleveland, October 23-25, 2015:
  - Included patterns-related sessions from four members of Patterns WG: Hoffman, Thukral, Peterson, Schindel.
- INCOSE IS2015, Seattle:
  - Included five of our WG members' papers with Pattern content: Pickard (best paper award); Cook; Peterson; Sanyal; Schindel.
- Agile SE Life Cycle Model (ASELCM) Project (joint w/Agile WG):
  - First four host site clinic/workshops were held in 2H2015.
  - Joint review later in today's meeting and Feb 1 Agile WG meeting.
- Our WG's PBSE Methodology Summary posted in INCOSE MBSE Methodology Survey: http://www.omgwiki.org/MBSE/doku.php?id=mbse:pbse
- Other announcements or updates?

Joint PLE WG – Patterns WG Interest Exploratory Workshop (1315-1500)



Joint Agile WG – Patterns WG Review of ASELCM Pattern (1530-1700)

The INCOSE ASELCM Pattern:
A Reference Model for Agility in Systems



2016 international workshop
Los Angeles, CA, USA
January 30 - February 2, 2016

REVIEW OF PATTERN CONTENT:
January 31, 2016, Patterns WG Meeting

V1.3.1

schindel@ictt.com 812-232-2062, attributed copies permitted

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 , Roadmaps and Migration Archetypes	• Interface Patterns
• Captured science patterns	• Specific domain patterns
• Example Product Line Engineering (PLE) Pattern (Joint w/PLE WG)	• Other interests from team members?

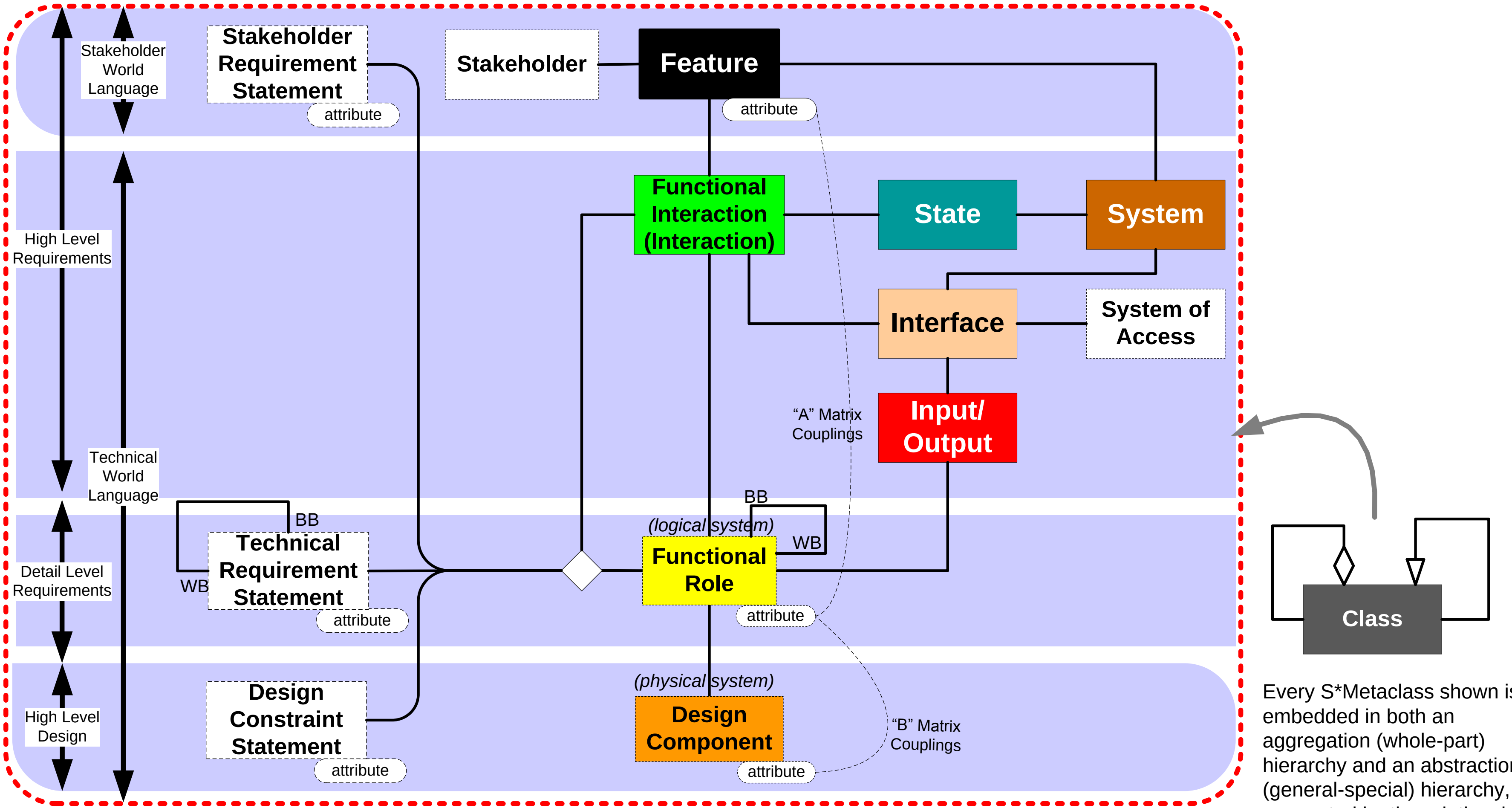
- Notice the “Joint with xxxxx Working Group” emphasis—why shouldn’t all the projects of this WG be with other INCOSE WGs?
- Future meetings schedule: Pace, rate, calendar
- Outreach: Who else should be involved? Example—other INCOSE WGs that are natural Patterns applications. Ideas?

Appendix

Background information on Pattern-Based Systems Engineering (PBSE)

Our WG's PBSE Methodology Summary posted in INCOSE MBSE Methodology Survey:
http://www.omgwiki.org/MBSE/doku.php?id=mbse:pbse

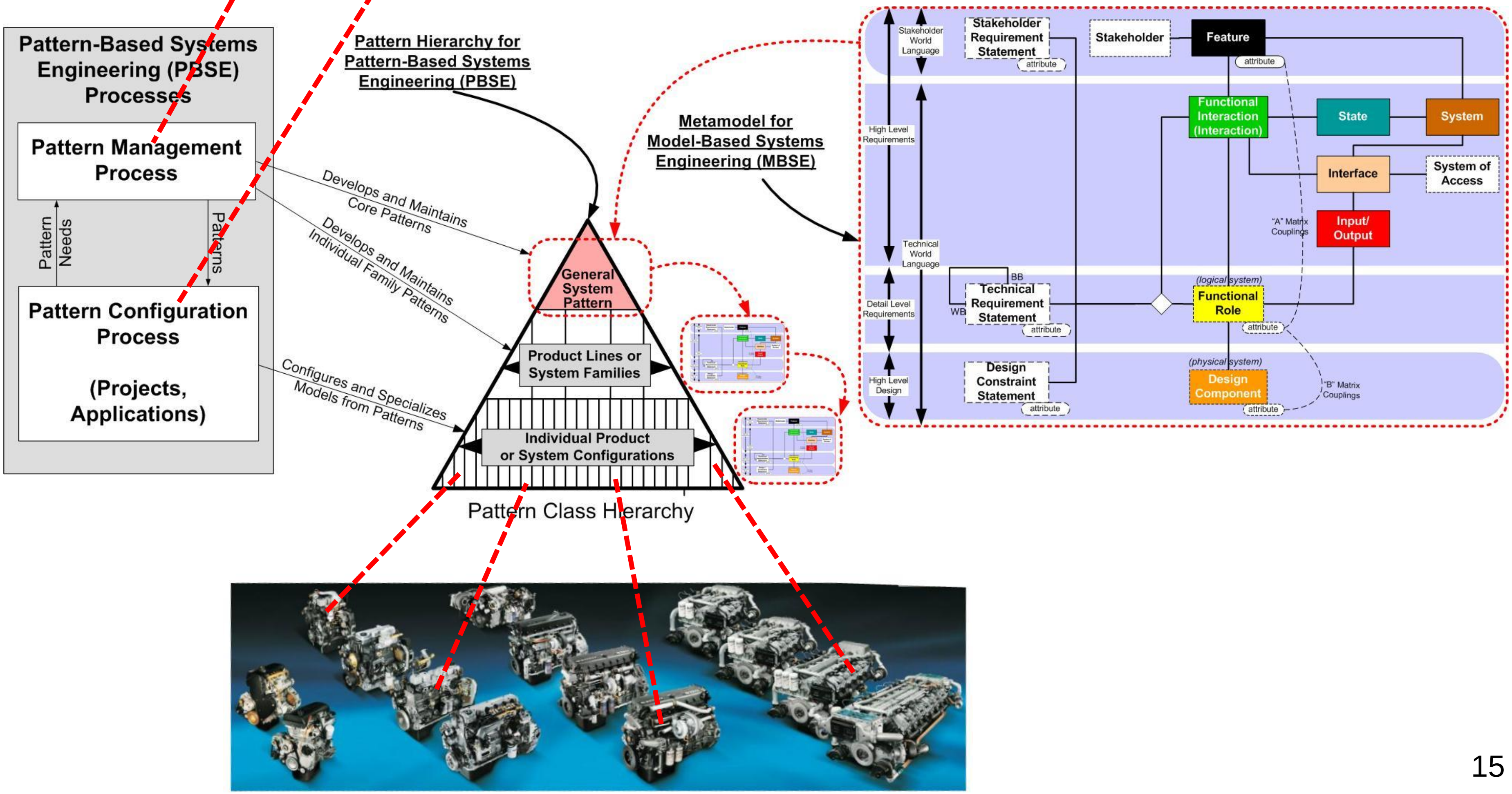
Summary of some major S*Metamodel classes and relationships—the underlying semantics of all S*Models (Refer to S*Glossary for definitions)



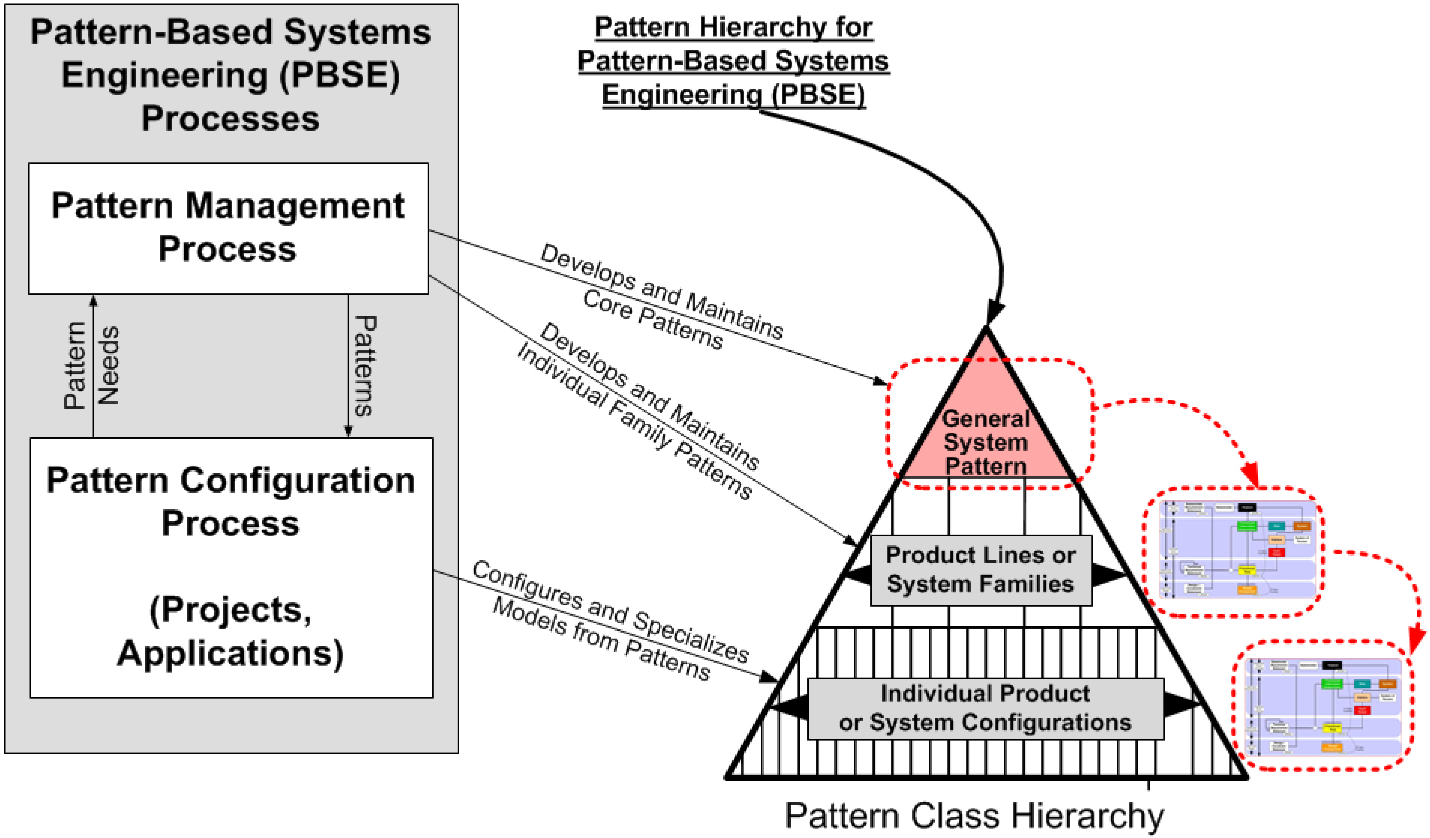
Every S*Metaclass shown is embedded in both an aggregation (whole-part) hierarchy and an abstraction (general-special) hierarchy, connected by the relationship types shown.

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.

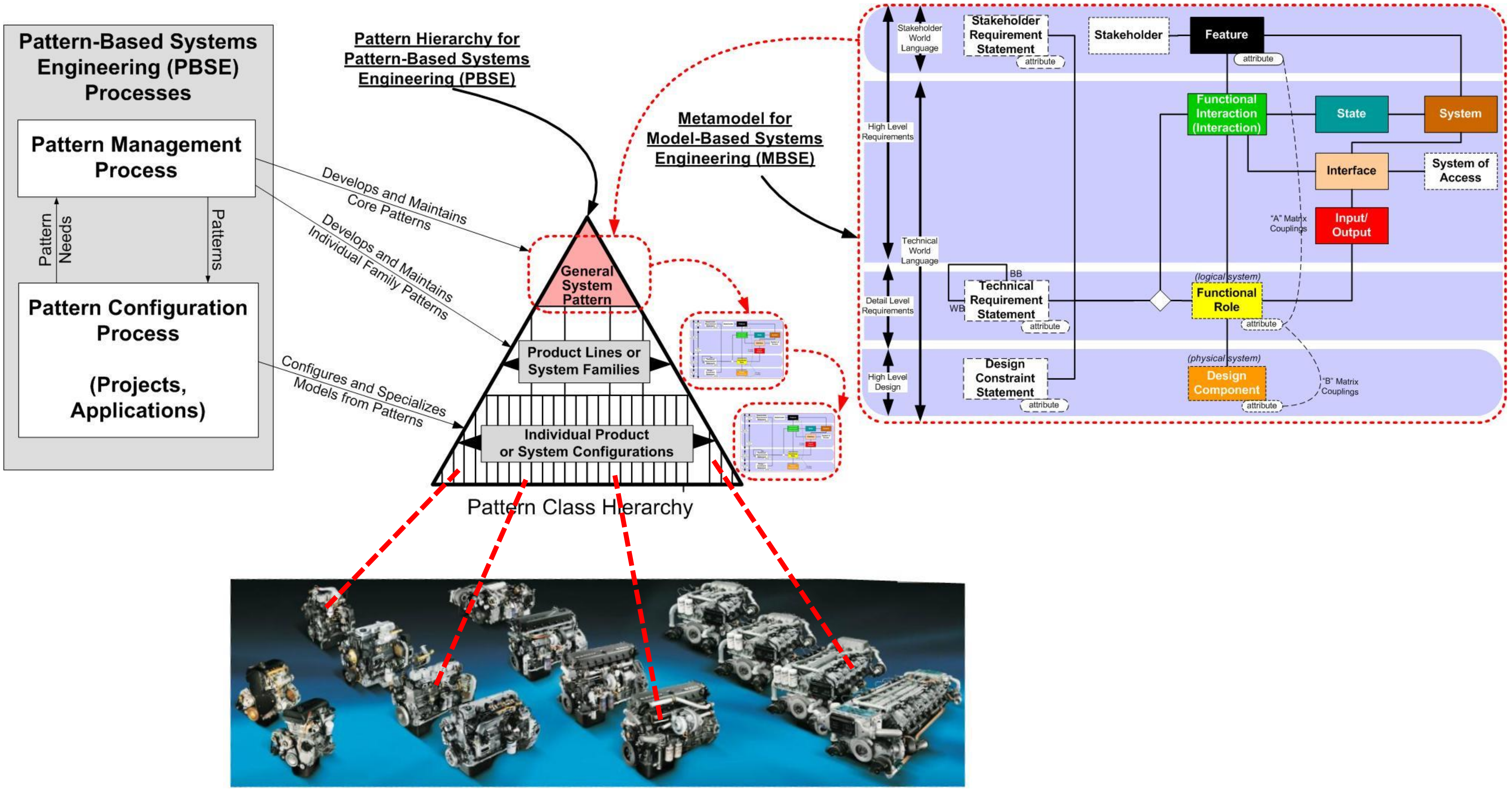


Business process optimized for PBSE fulfill a different vision:

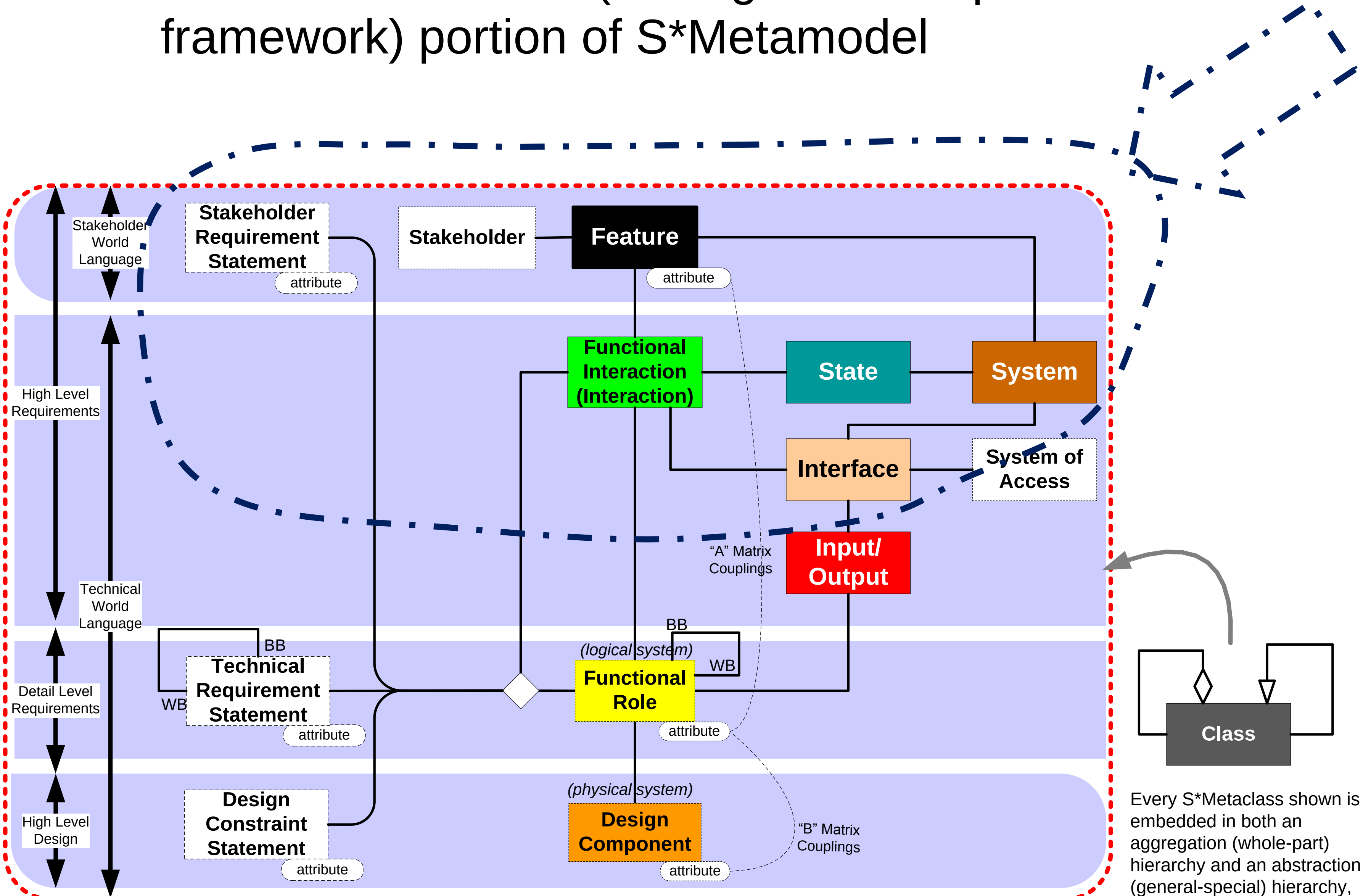


Why do most representations of the systems engineering process appear to assume starting from no formal knowledge about the system of interest & its domain?

What is in an S*Pattern, and why?



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

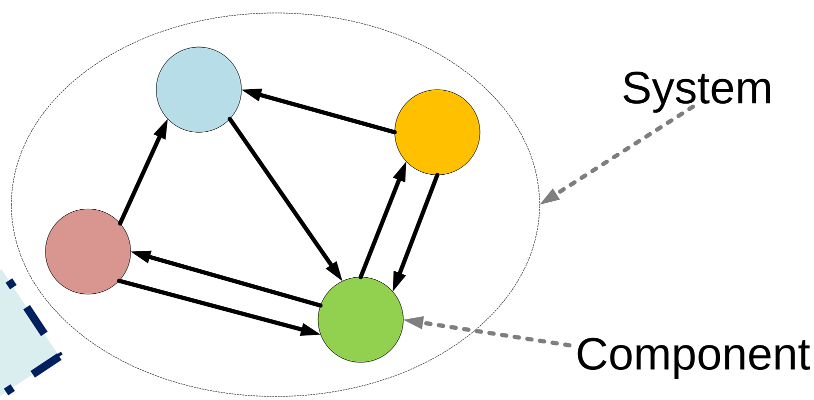


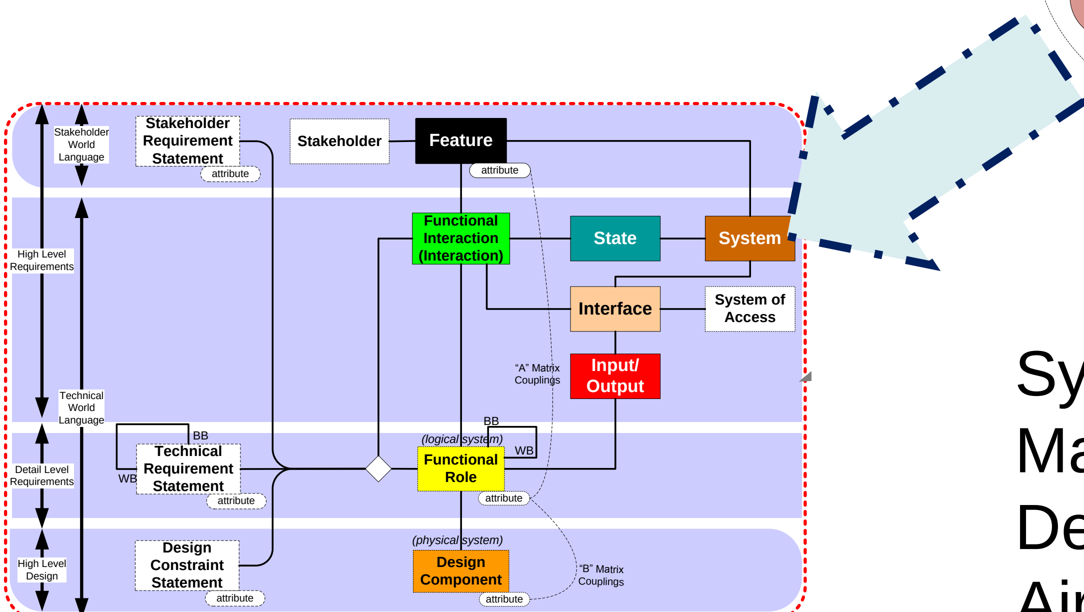
Every S*Metaclass shown is embedded in both an aggregation (whole-part) hierarchy and an abstraction (general-special) hierarchy, connected by the relationship types shown.

- 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.

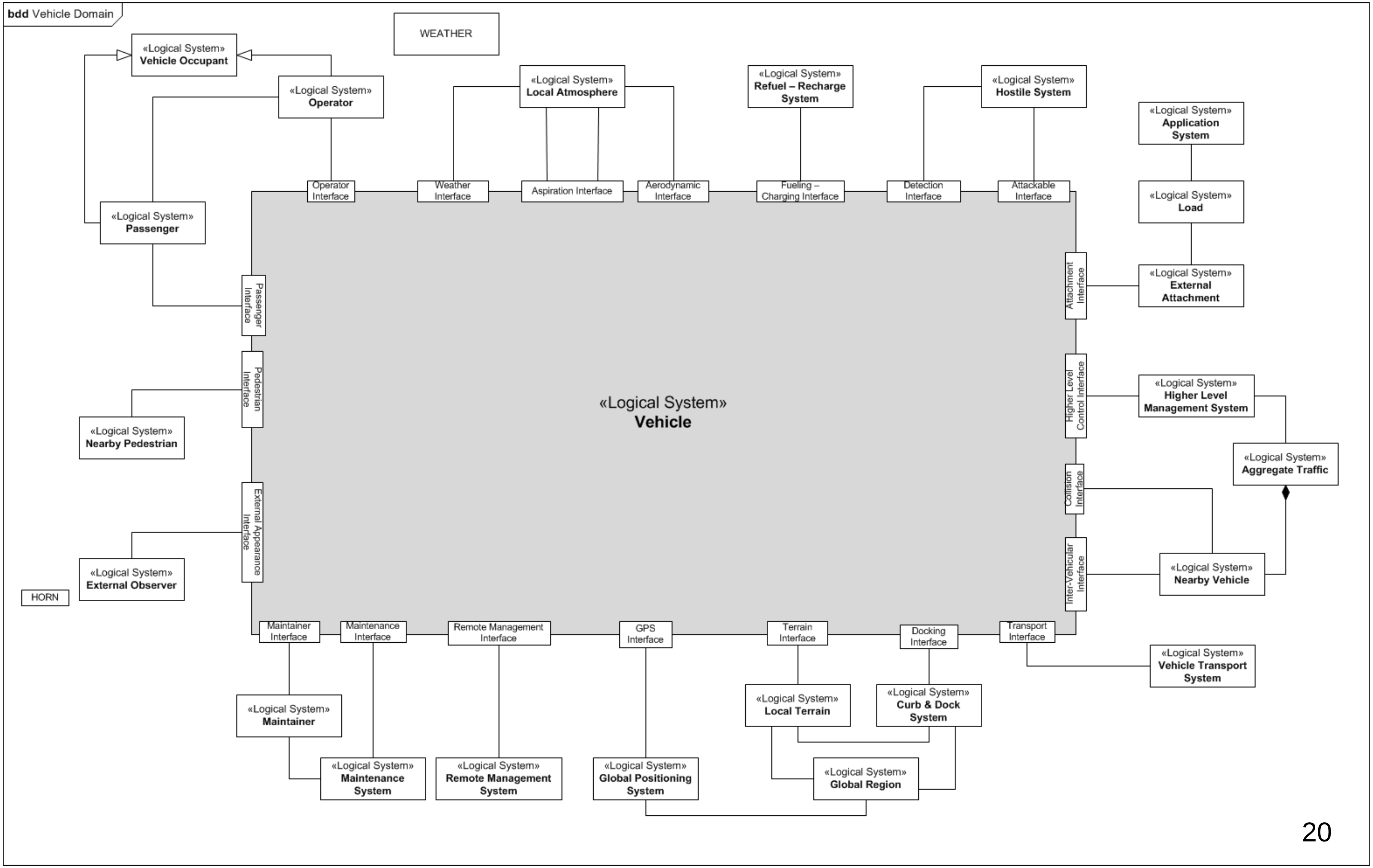
- A Component can be a System.



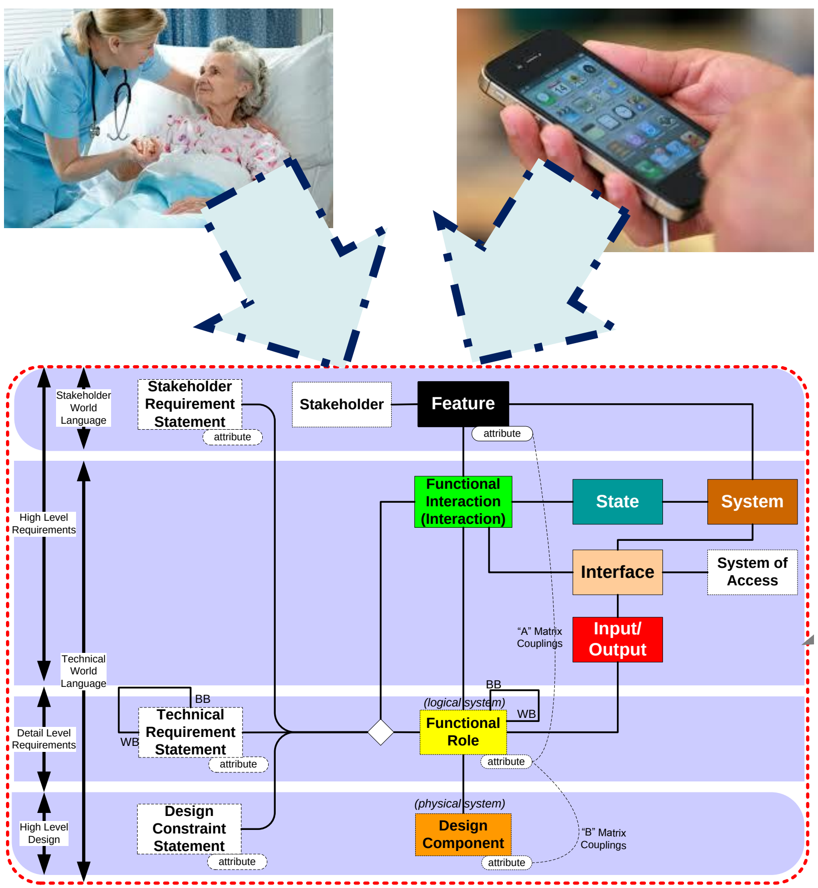


Systems: Engine, Vehicle, Manufacturing Line, Medical Device, Consumer Product, Aircraft, Engine, etc.

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



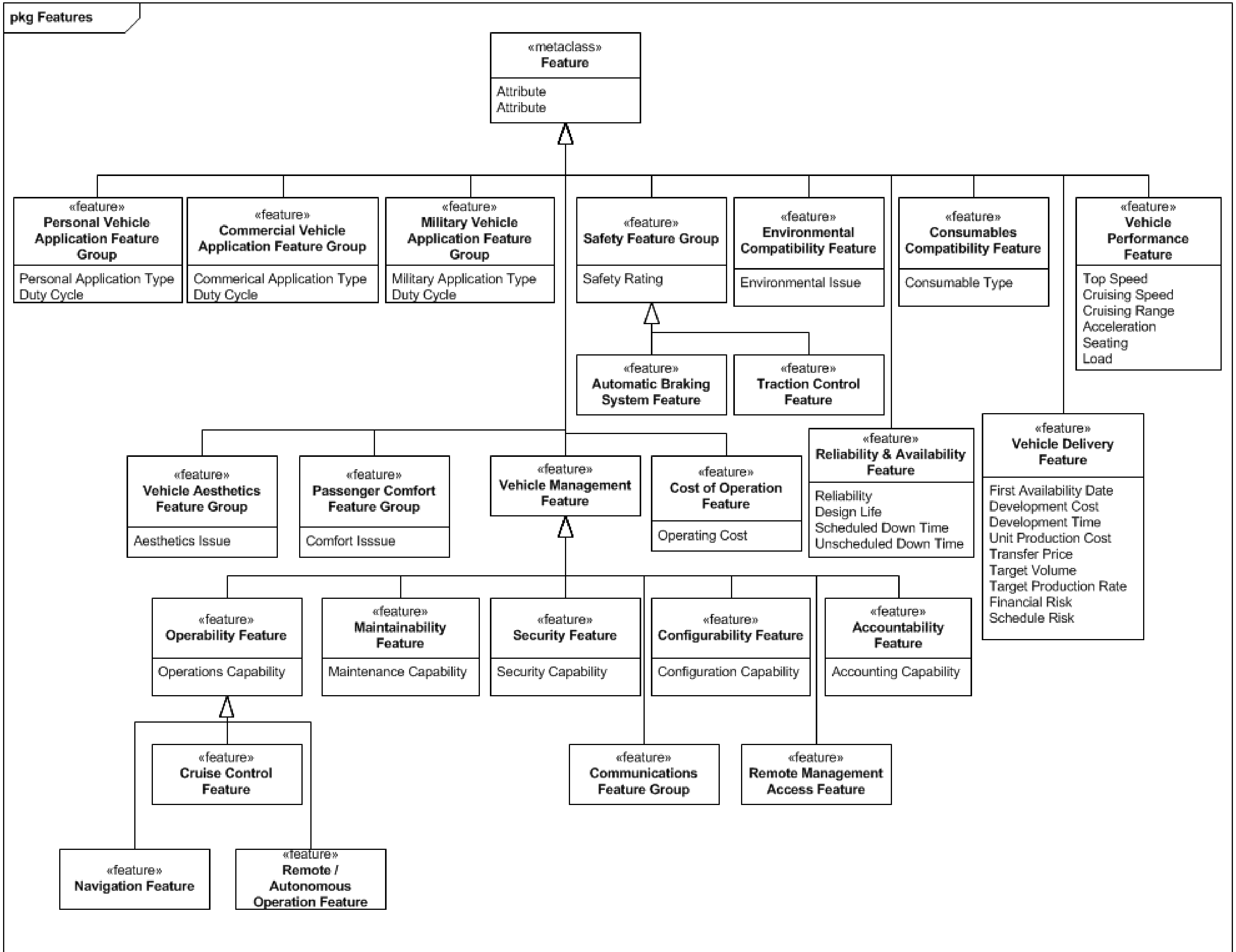
- A Stakeholder is a person, organization, community, or other entity with a stake in the behavior of a system.
- A Feature is a system behavior or capability having value to a Stakeholder, described in Stakeholder concepts & language.
- Features are the basis of Stakeholder selection of systems.



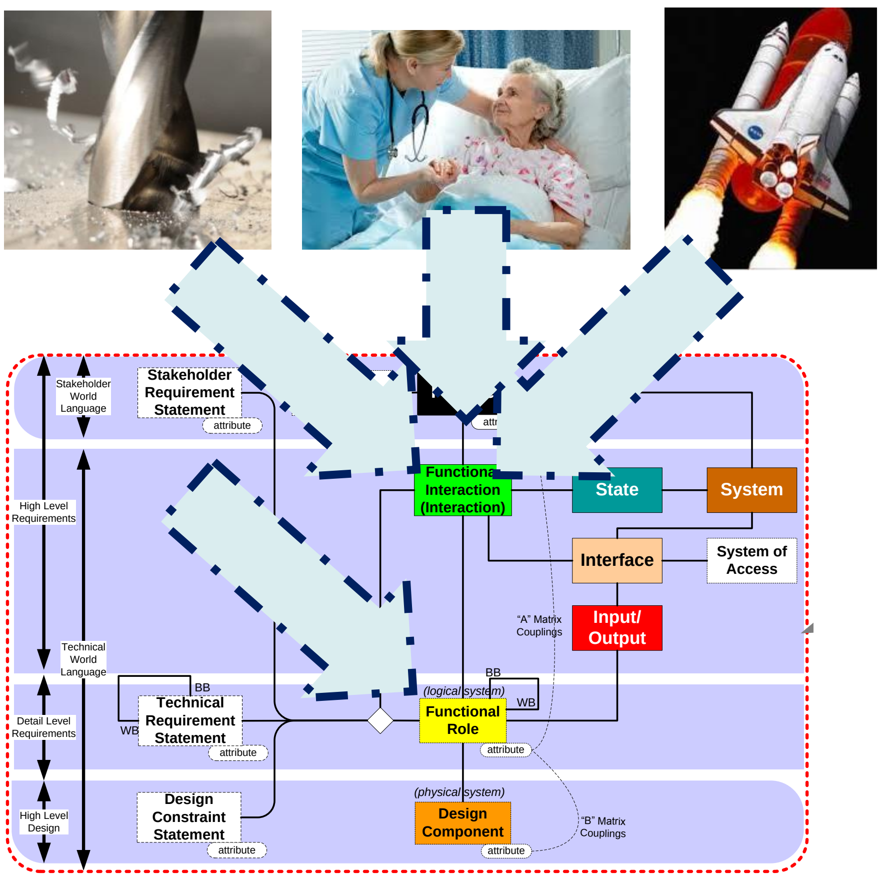
- Features are parameterized by Feature Attributes.
- These measures of effectiveness are in Stakeholder terms, so are frequently subjective and non-technical.

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)



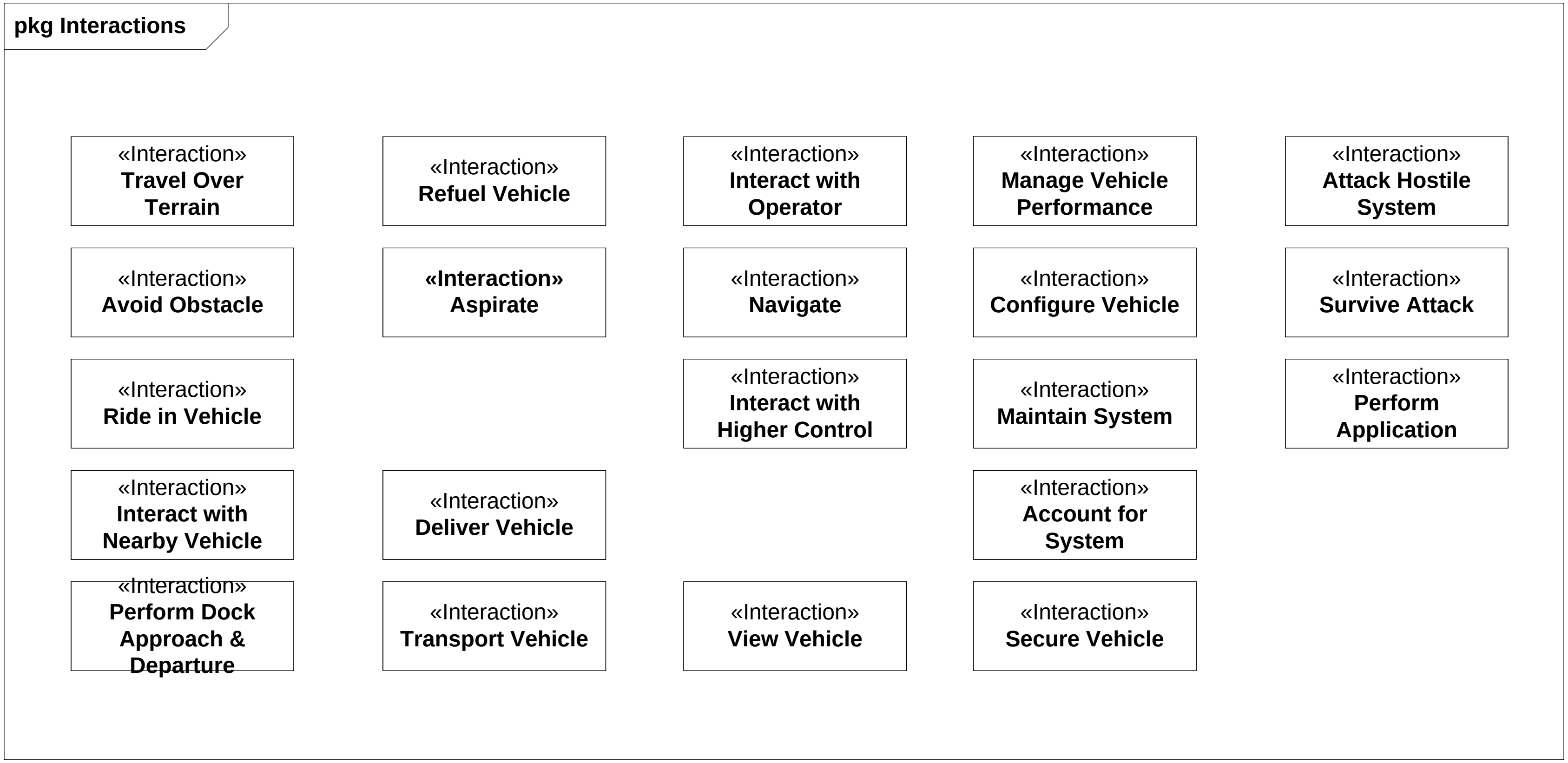
- A (Functional) Interaction is an exchange of energy, force, mass, or information, by two or more entities, said to play (Functional) Roles in the Interaction.
- All behavior occurs in the context of Interactions.



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

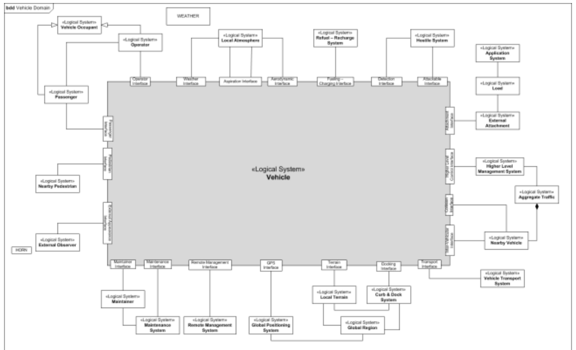
Interactions: Travel over Terrain, Ride in Vehicle, Maintain System, etc.

- 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.

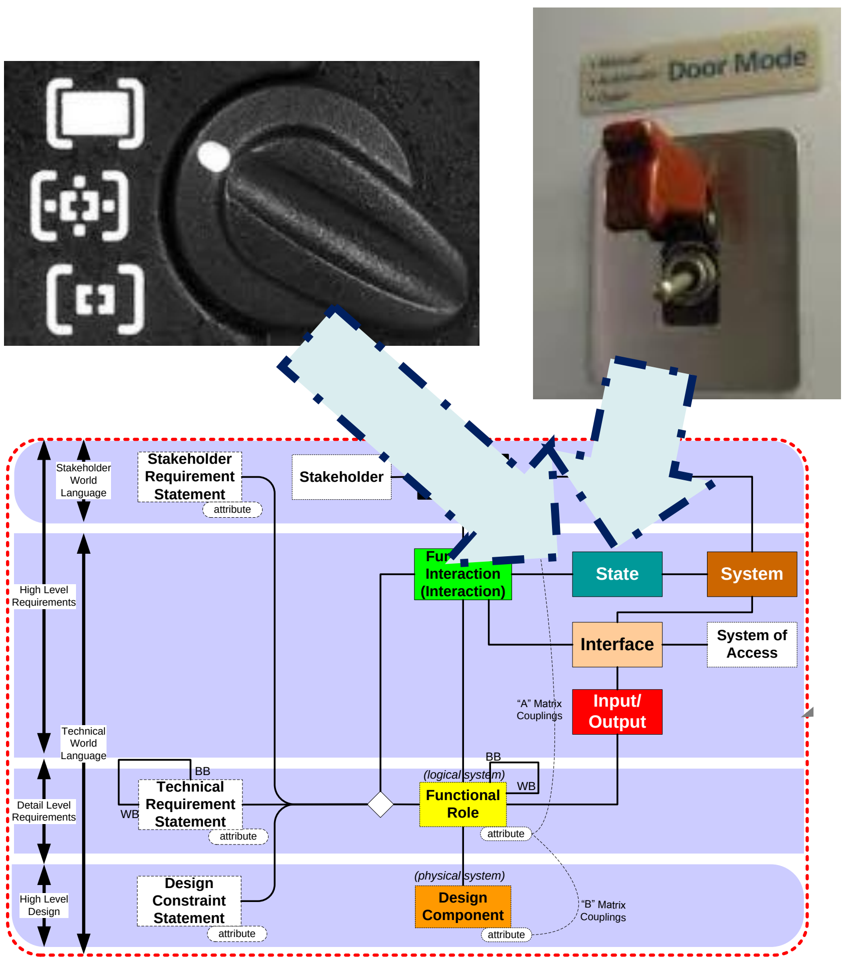


Vehicle Interactions:
Which Actors Participate in Interaction?

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																



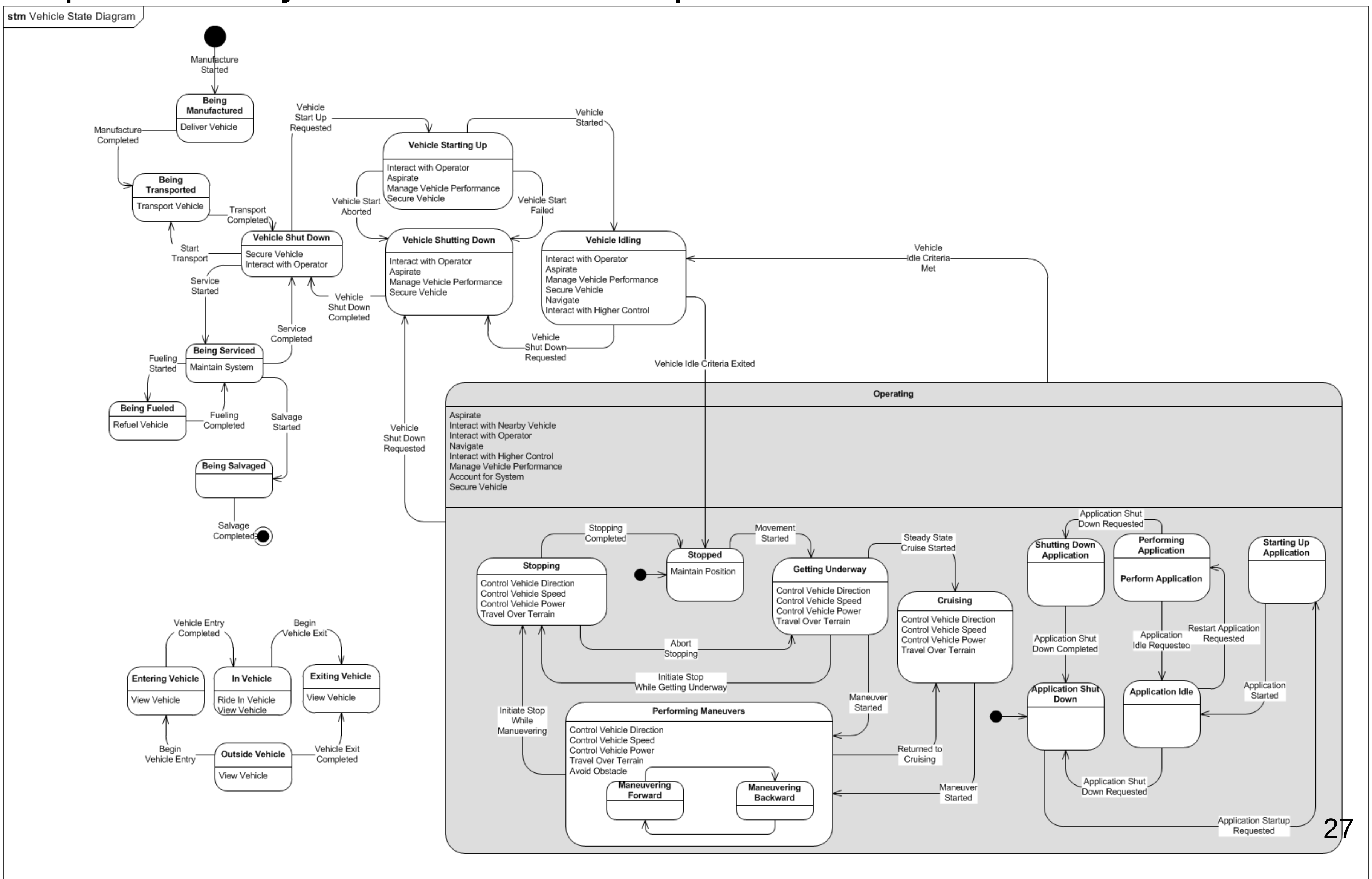
- A <u>State</u> is a condition of a system that determines its future behavior.
- Some state variables are continuous (e.g., position, velocity), and others are discrete (e.g., operational states).
- For the discrete case, Finite State Machine models are used.



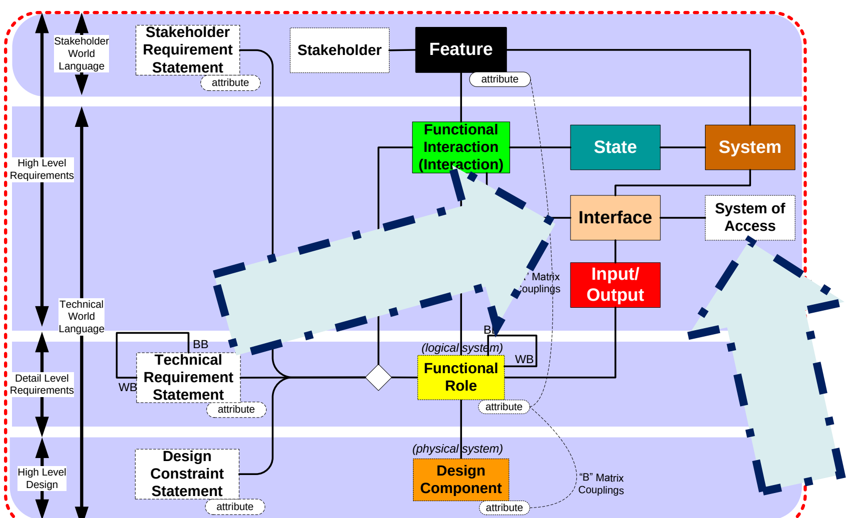
- The fact that different behavior is expected in the different (finite) states is represented by associating different Interactions with different (finite) States.

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

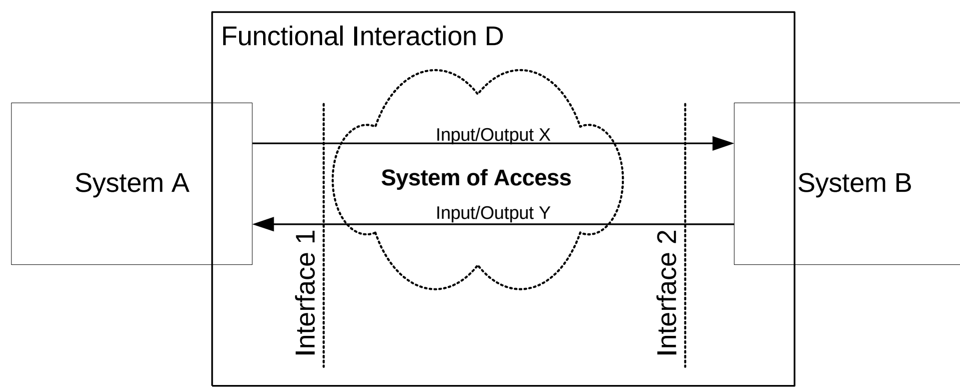
- In the High Level Requirements (HLR) subset of an S*Model, the State Model establishes a high level temporal (time) model of the system.
- The scope of such a State Model may be the entire System Life Cycle, an Operational Cycle, or other time scope.



An <u>Interface</u> is an association of (1) a system (which <u>has</u> 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).

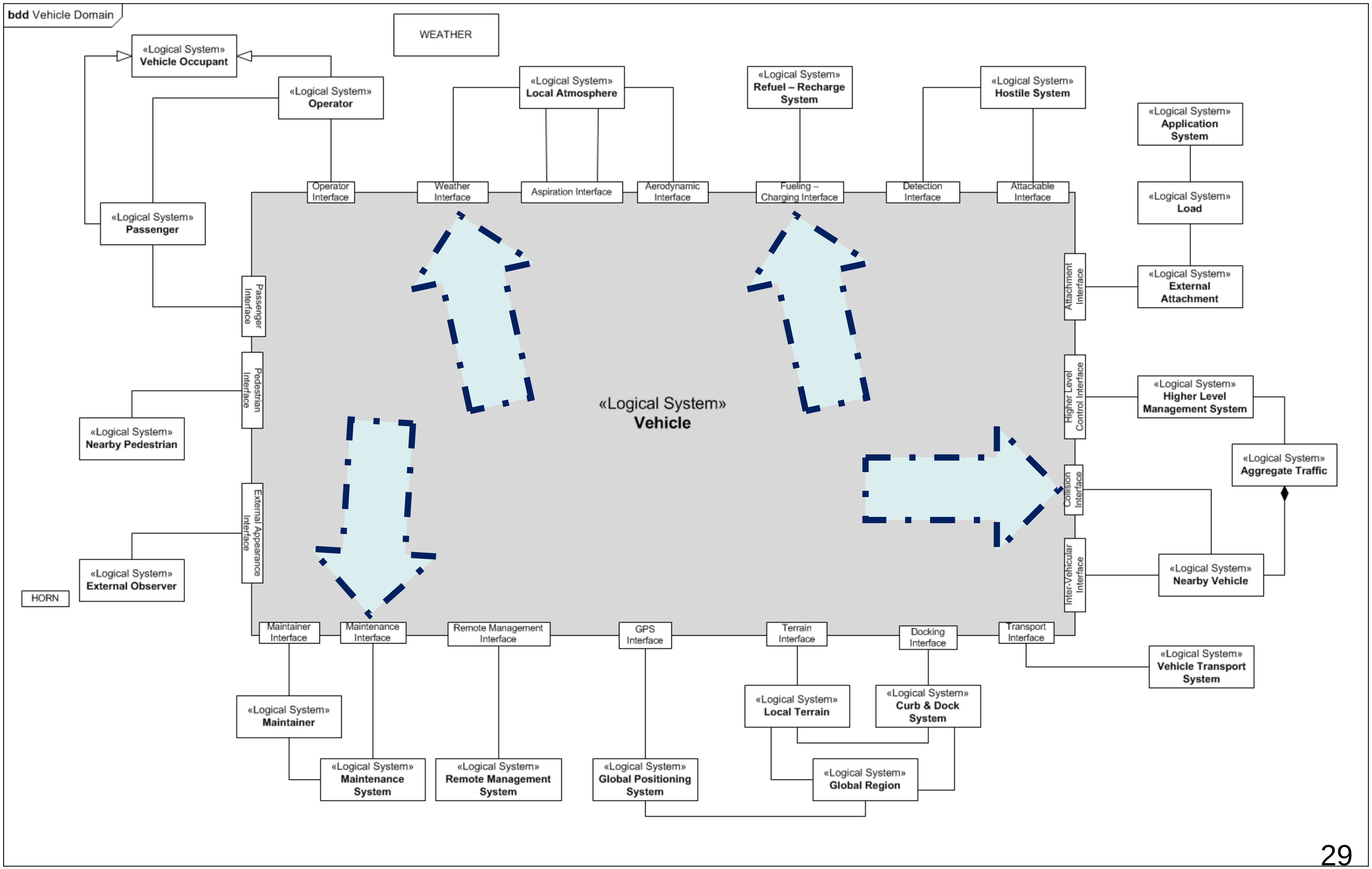


S*Models show that there are <u>multiple</u> interfaces between systems:

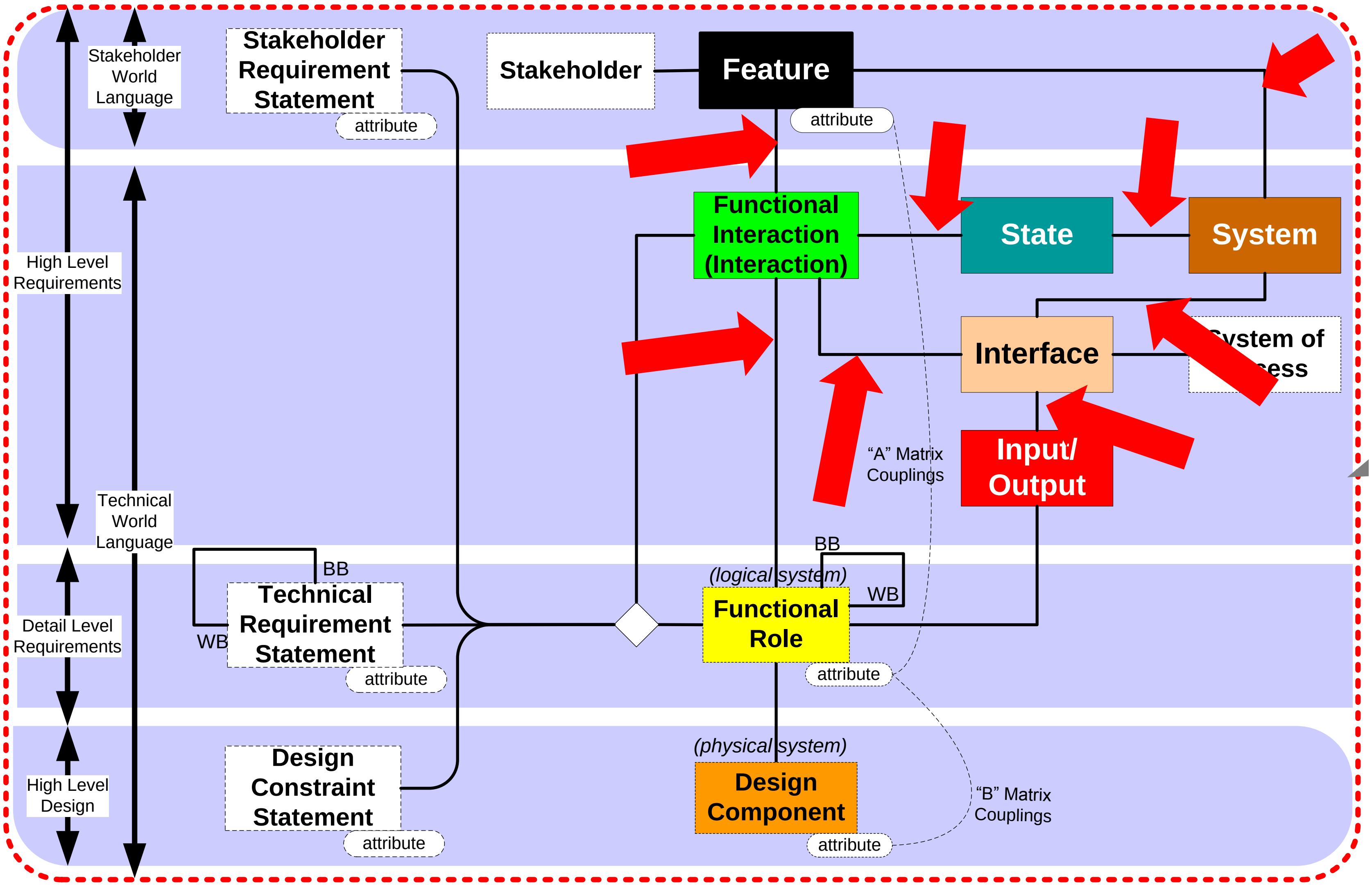


Systems of Access: Steering Wheel, Dashboard Indicator, Pedal, Headlights, etc.

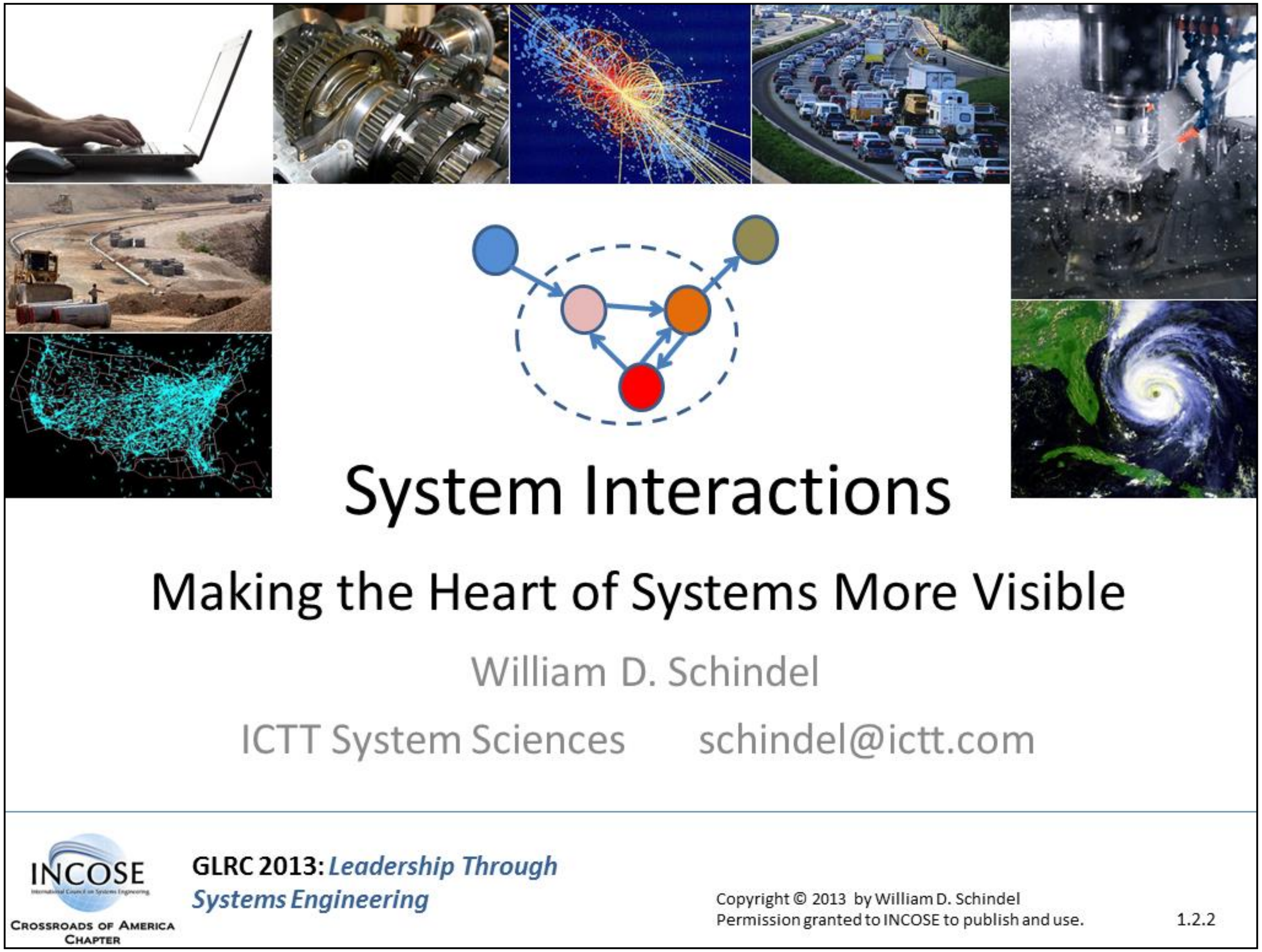
In S*Models, external Interfaces can appear at the edge of systems (i.e., in Domain Diagrams), and queries can be used to generate Interface Control Document (ICD) views.



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):

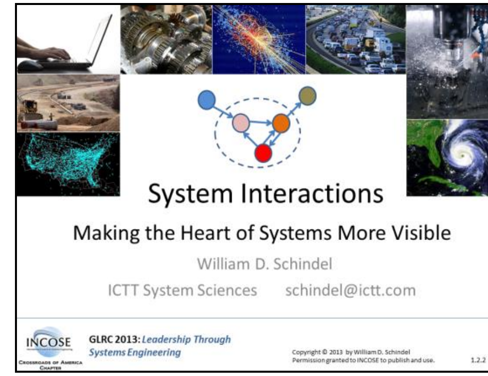


Criticality of modeled Interactions to the heart of MBSE, PBSE, science and engineering



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

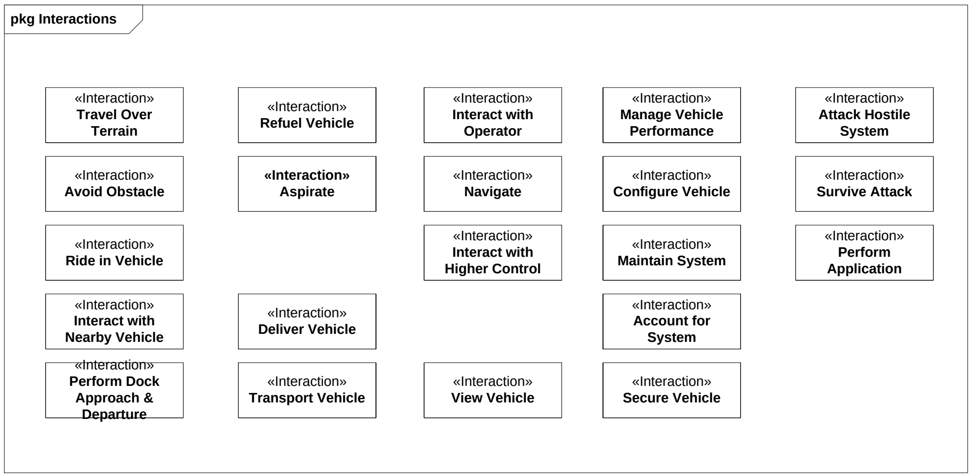
For more detail, see -->

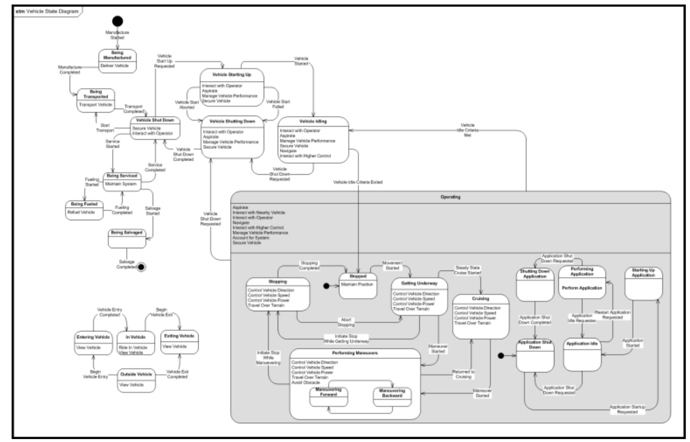


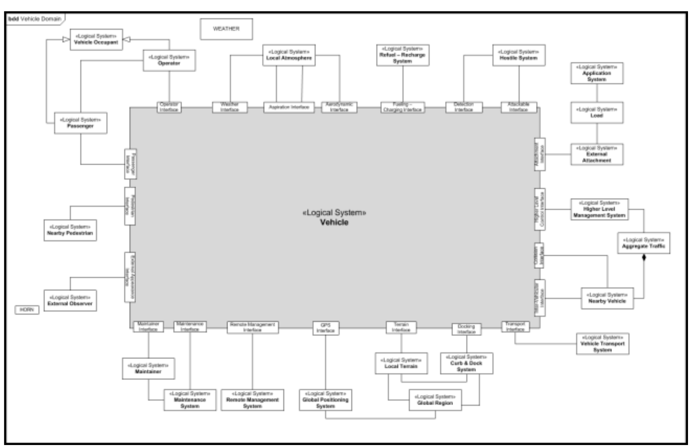
- In a nutshell:
  - Physical interaction models provide the context for all the laws of the hard sciences (Newton, Maxwell, Boltzmann, etc.).
  - Explicit models of physical interactions are perfectly legal in MBSE models (collaboration, activity, etc.), but are frequently under-emphasized in them.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



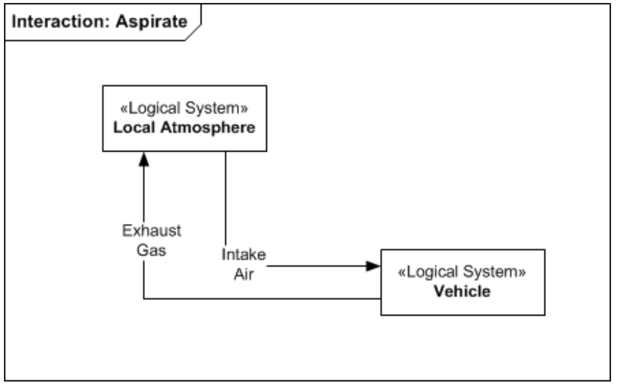


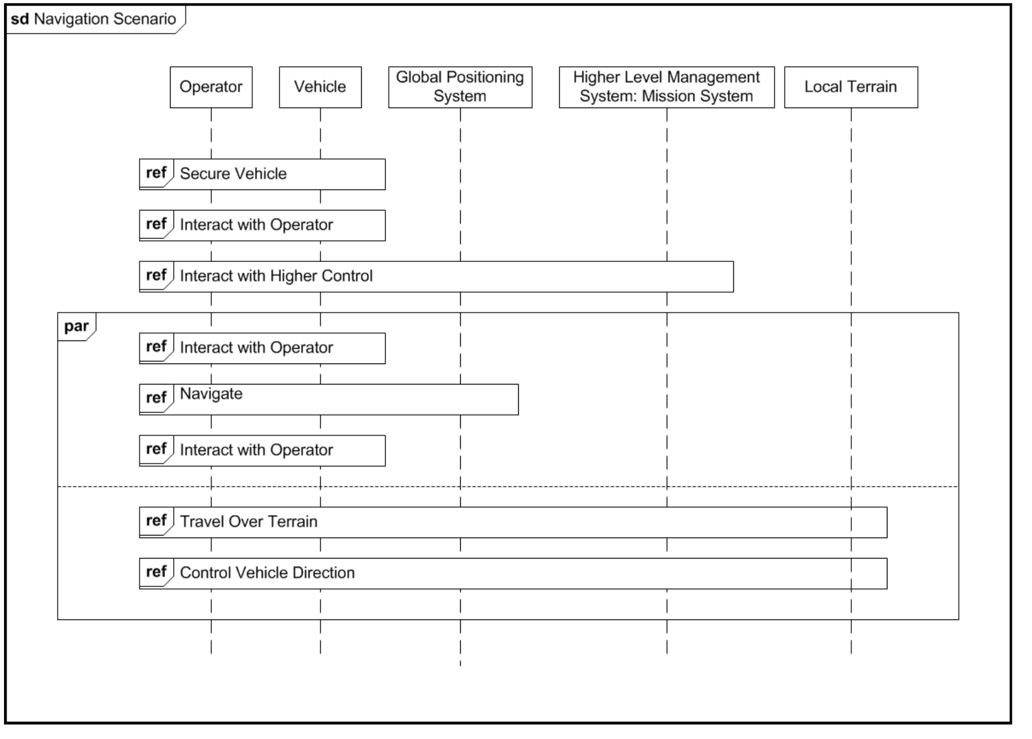


1			Actors																					
2	Interaction Name	Interaction Definition	Vehicle	Operator	Passenger	Vehicle Occupant	Nearby Pedestrian	External Observer	Maintainer	Maintenance System	Local Atmosphere	Refuel System	Hostile System	External Attachment	Load	Application System	Higher Level Management	Nearby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

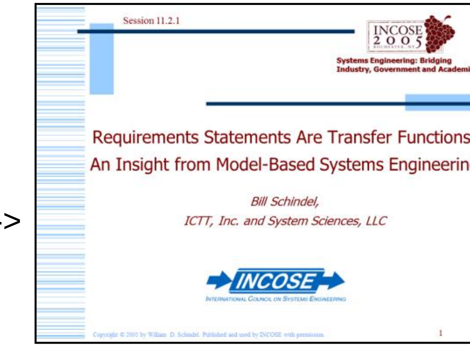
Abstract. Traditional systems engineering pays attention to careful composition of prose requirements statements. Even so, prose appears less than what is needed to advance the art of systems engineering into a theoretically-based engineering discipline comparable to Electrical, Mechanical, or Chemical 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 -->


Session 11.2.1

INCOSE 2005

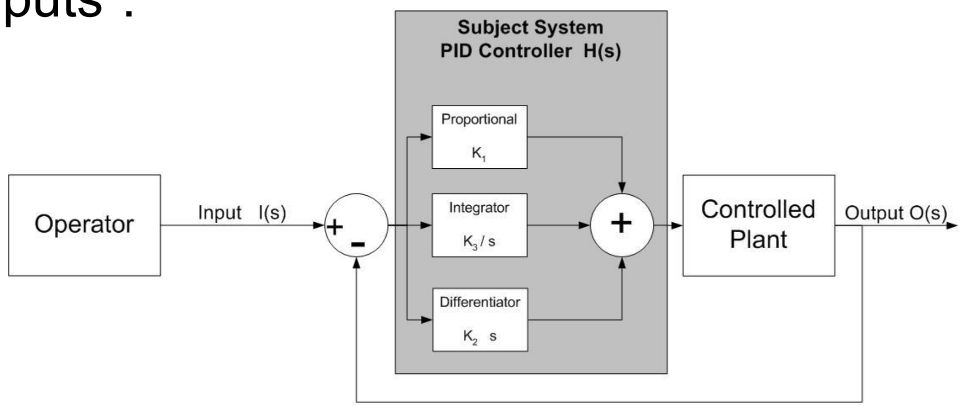
Systems Engineering: Bridging Industry, Government and Academia

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":



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

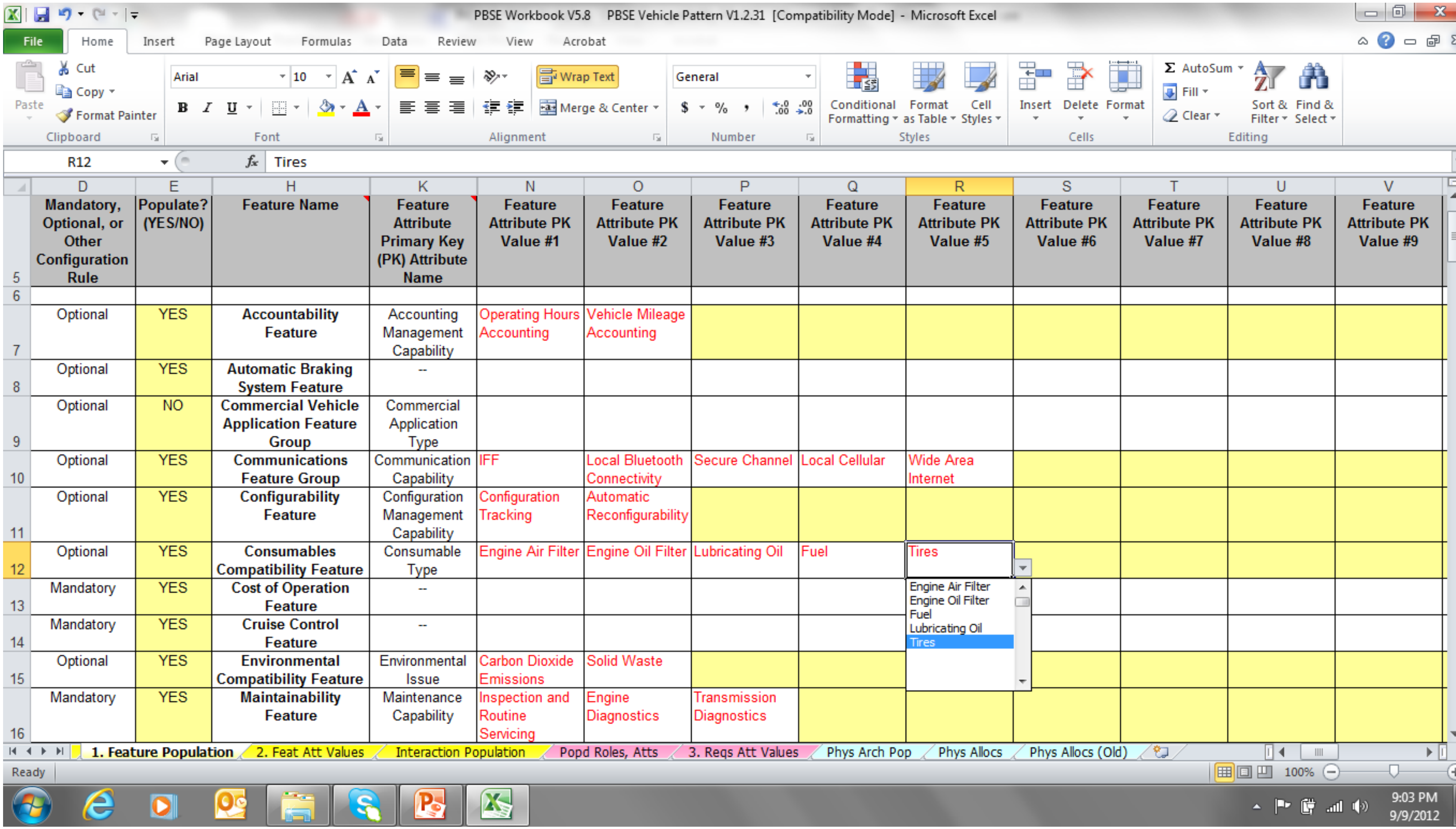
  - However, Transfer Functions are limited to linear systems, and describes their behavior in the frequency domain. Systems generally are not linear, and frequently not described by available mathematical equations!
  - However, for general systems we can extend the idea of (linear) Transfer Functions,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

Filling in the Feature Population Form—with Stakeholder Needs

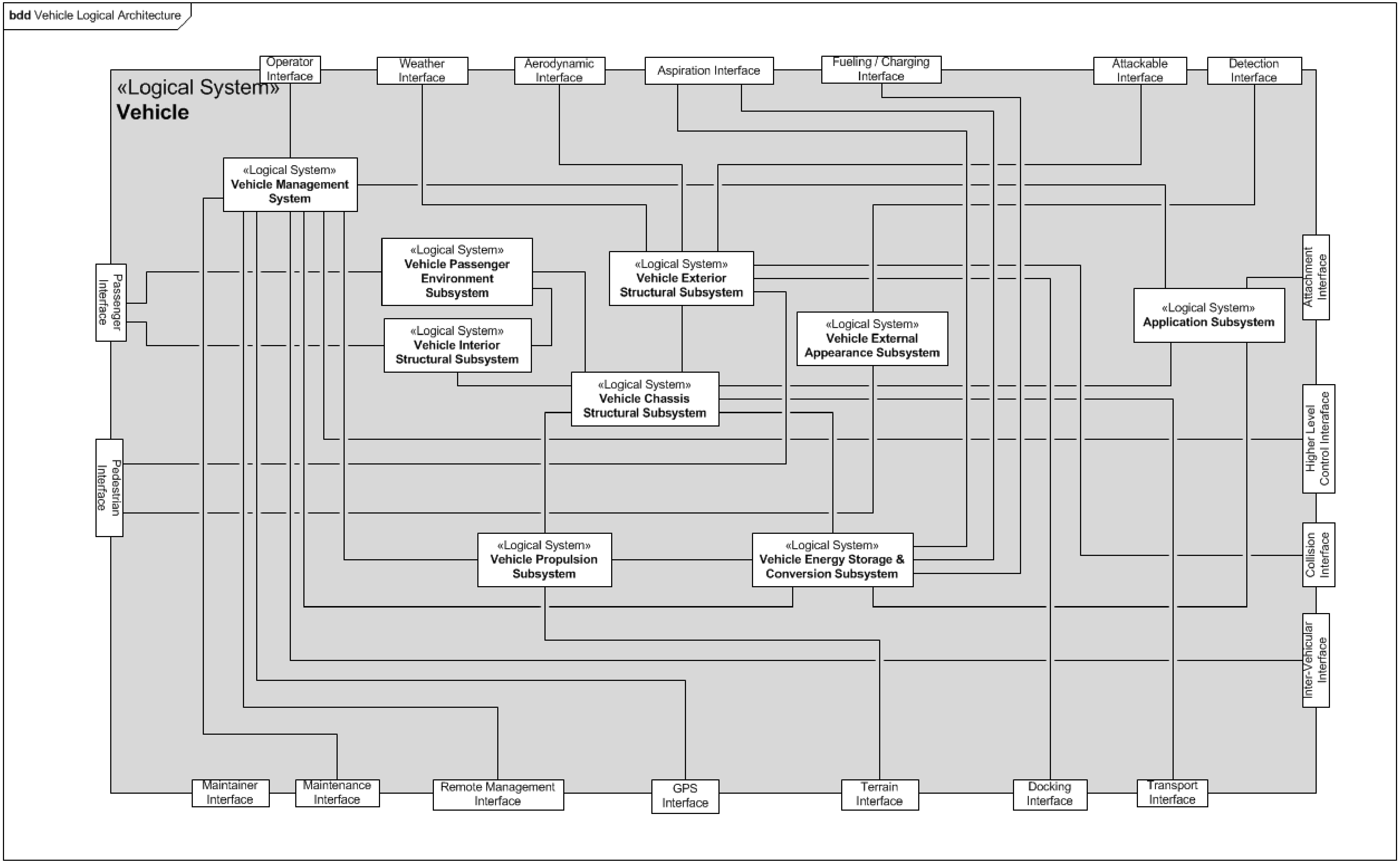


	D	E	H	K	N	O	P	Q	R	S	T	U	V
5	Mandatory, Optional, or Other Configuration Rule	Populate? (YES/NO)	Feature Name	Feature Attribute Primary Key (PK) Attribute Name	Feature Attribute PK Value #1	Feature Attribute PK Value #2	Feature Attribute PK Value #3	Feature Attribute PK Value #4	Feature Attribute PK Value #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						

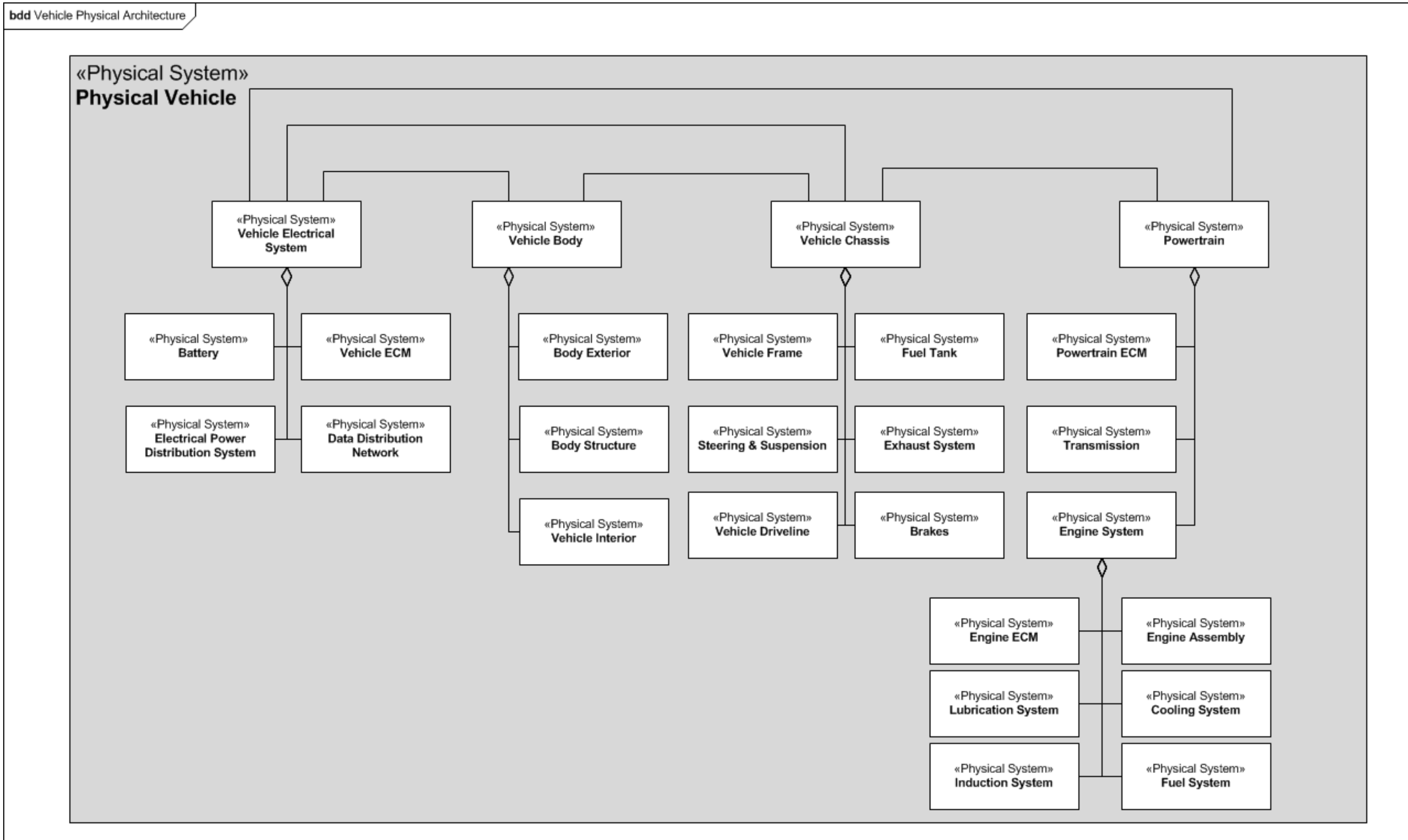
Resulting Auto-Populated Requirements

	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.

Logical Architecture Model

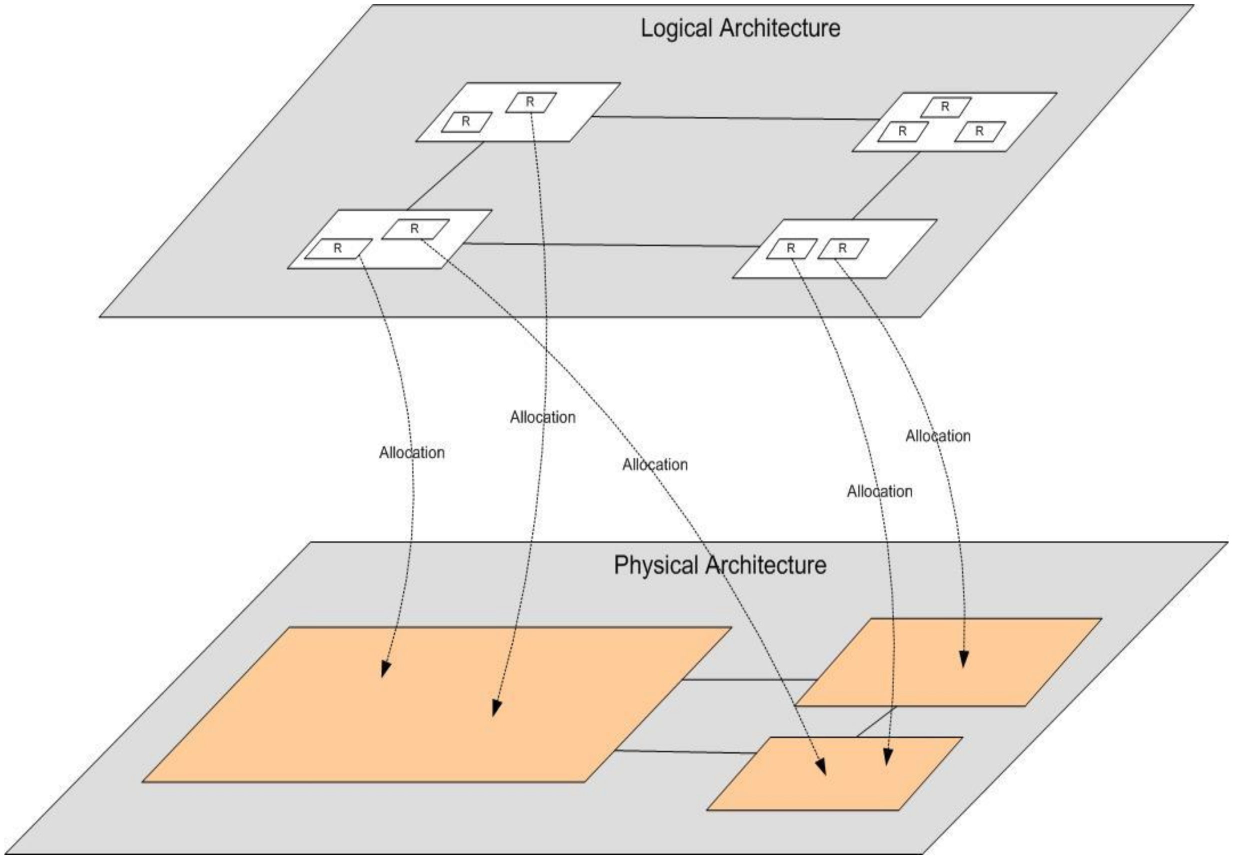


Physical Architecture Model



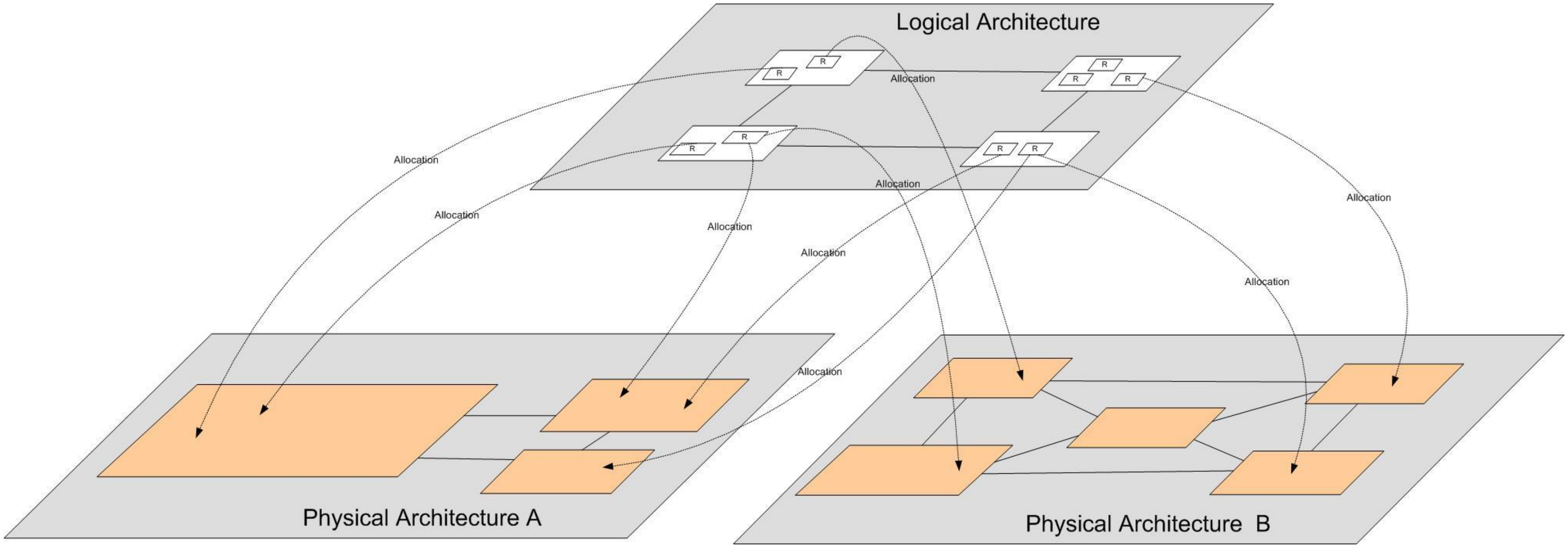
Acknowledgement: Influenced by related physical architecture work of John Thomas

Allocation of Logical Roles to Physical Architecture



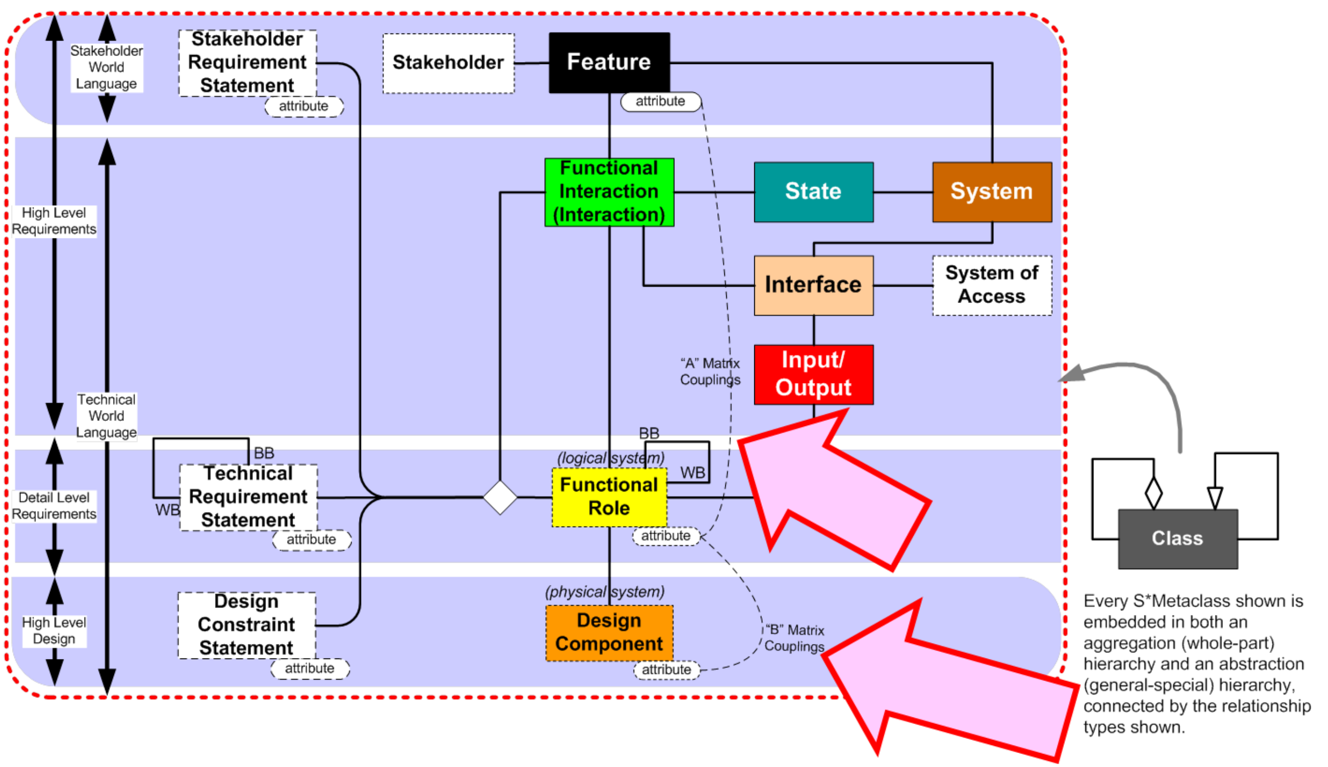
Allocation of Logical Roles to Physical Architecture

- Same Logical Architecture covers many Physical Architectures:

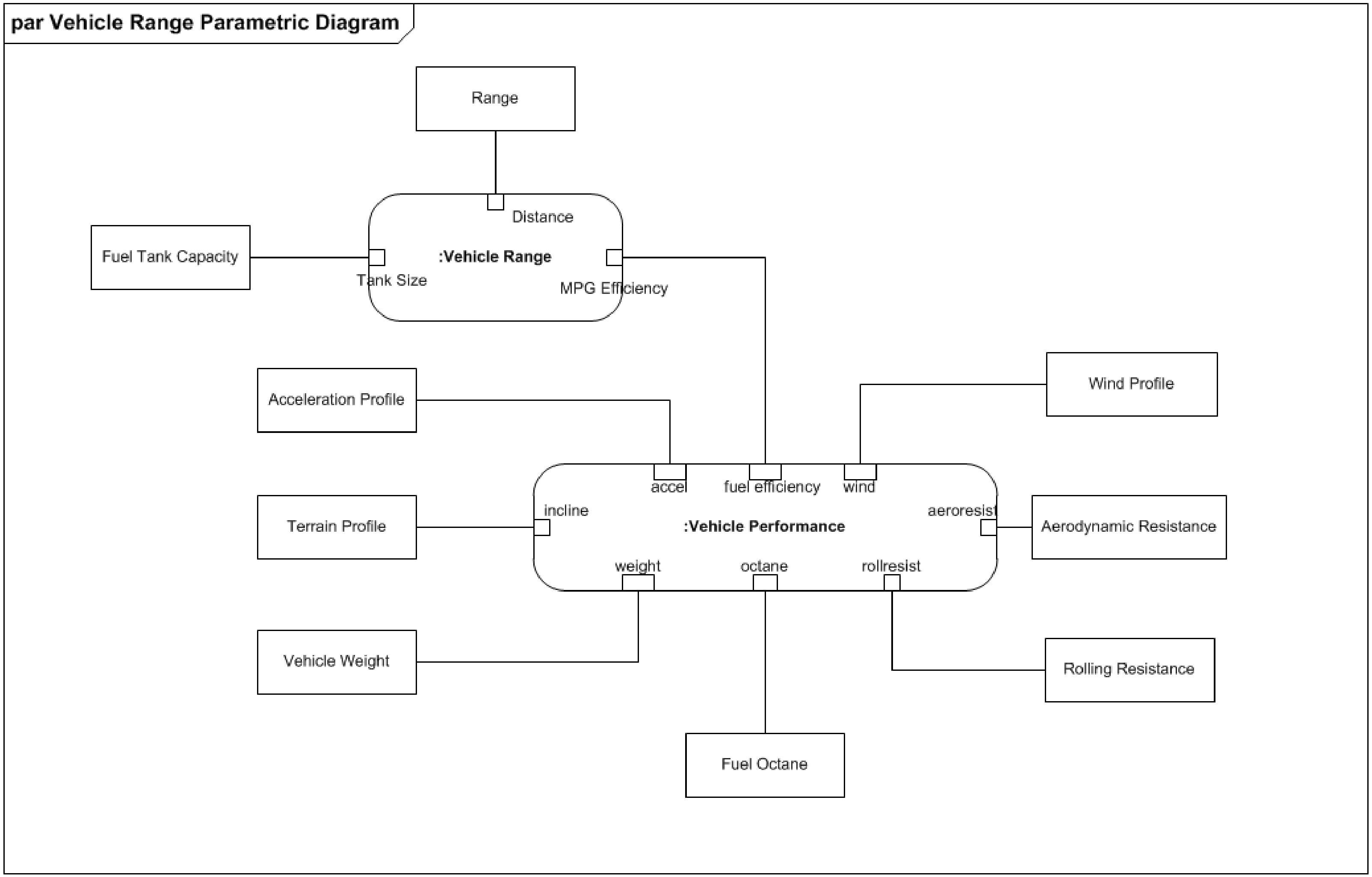


Attribute parametric couplings

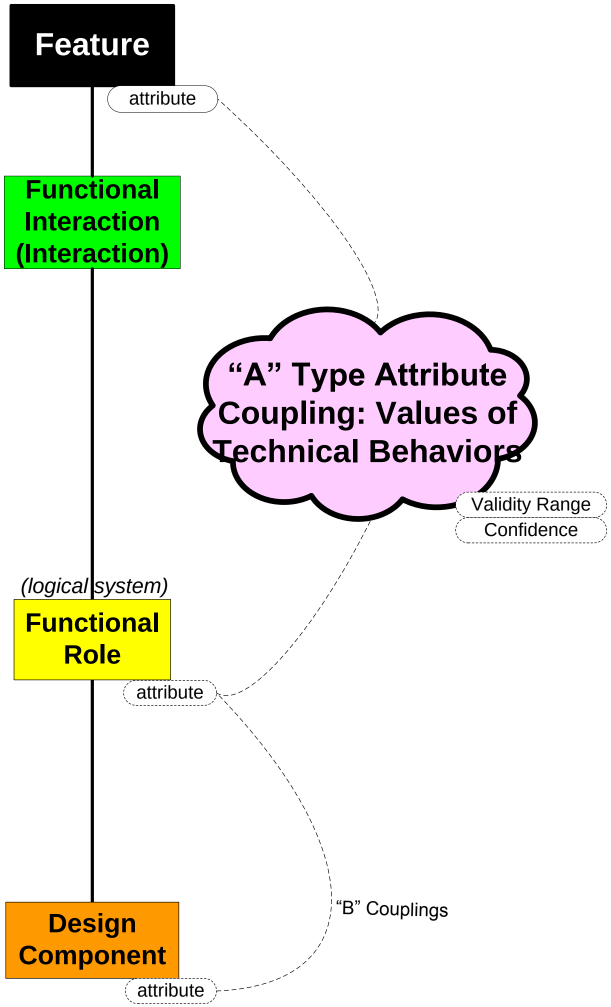
- Stakeholder Feature Attribute – to – Technical Roles & Requirements Attribute Couplings;
- Technical Roles & Requirements Attribute – to – Physical Component Attribute Couplings.



Attribute Coupling Model: SysML Notation



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

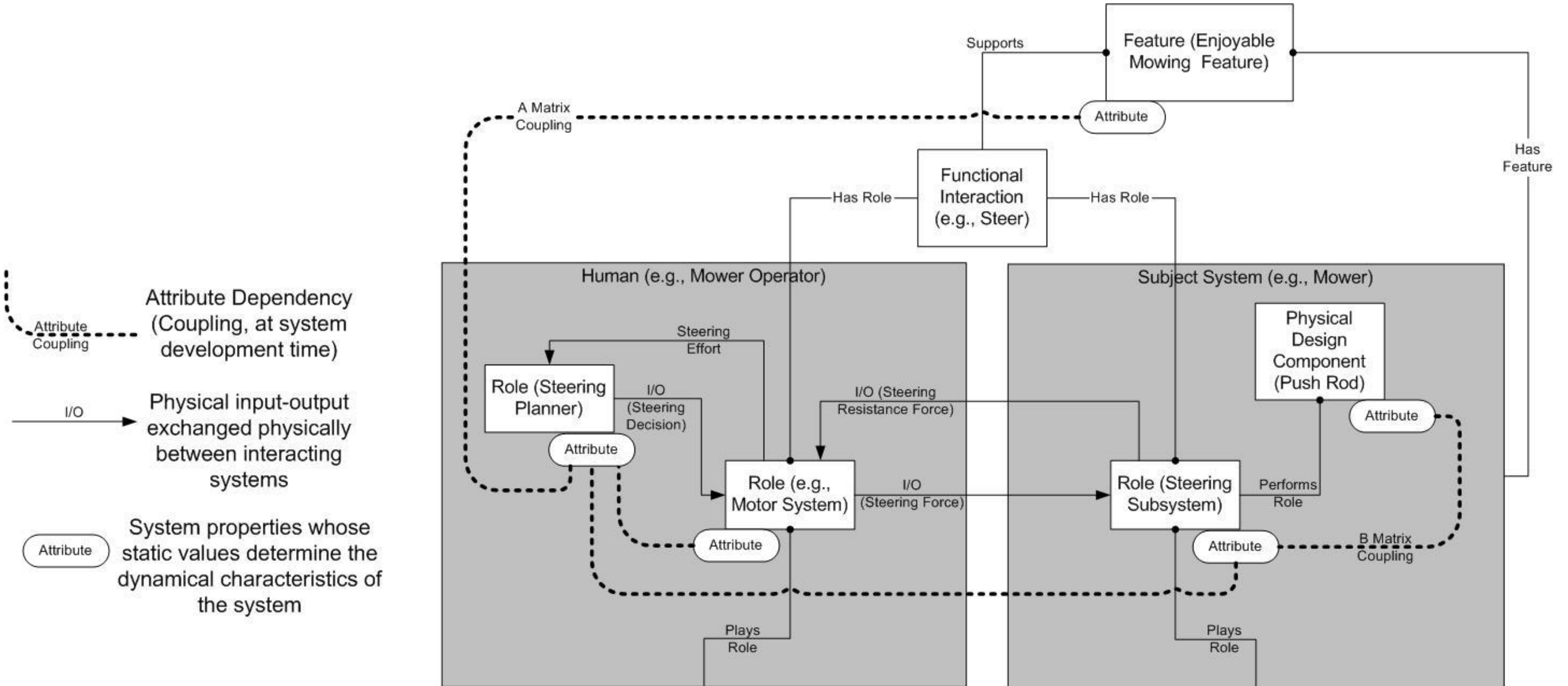


These "A" type parametric couplings describe how parameter value changes in technical behavior (the attributes of Roles / Requirements) bear on changes in Stakeholder-perceived value (the attributes of Stakeholder Features).

A simple example

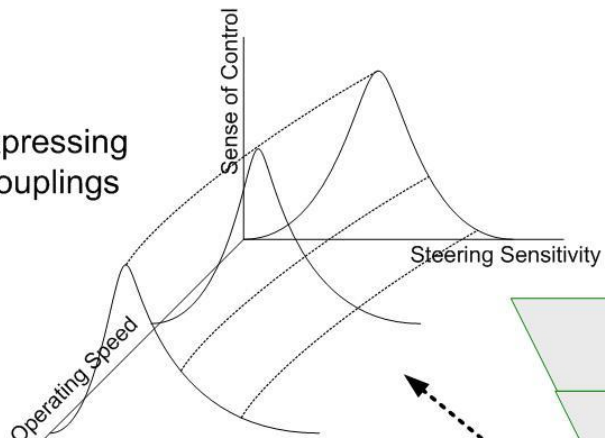
Operator fatigue, sense of control, coupled to technical steering gain, mower speed:

- "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.



Features

Feature Attributes

Functional Roles

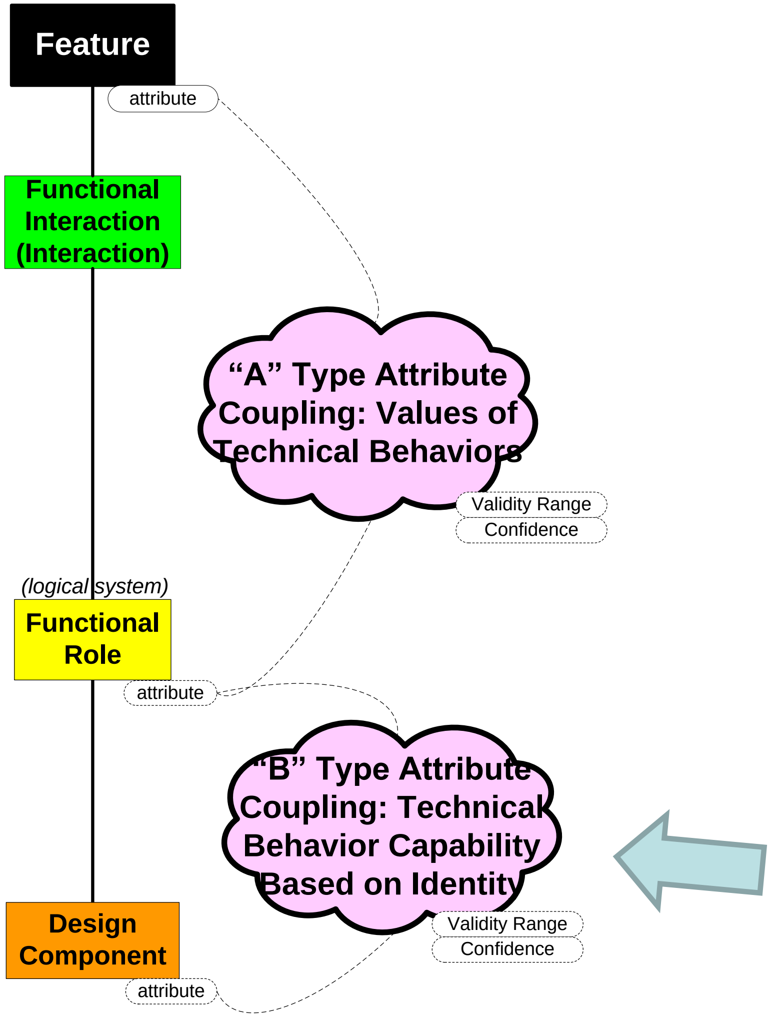
Role Attributes

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

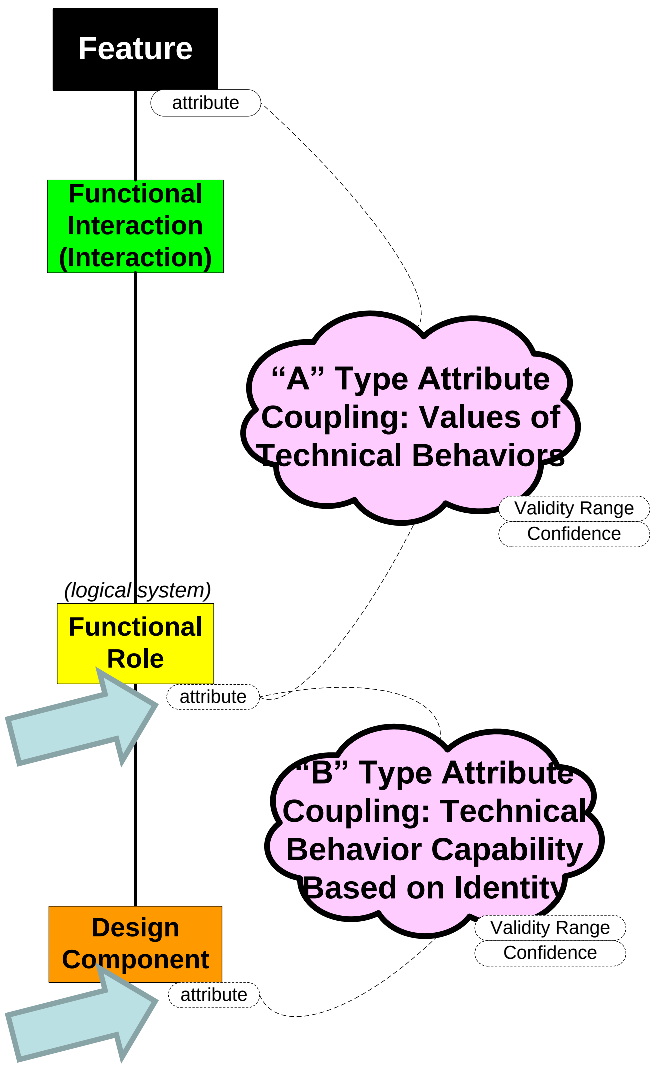
<u>"A" Matrix</u>: Expresses Feature-Role Attribute Couplings

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



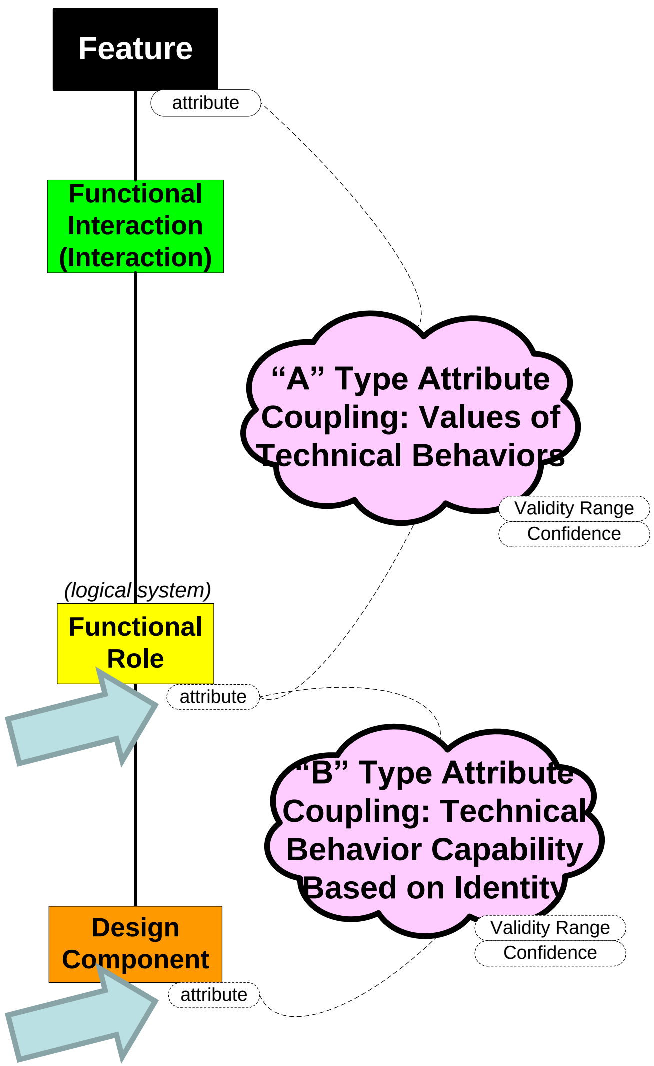
The "B" type parametric couplings describe how parameter value changes in <u>design components</u> (the attributes of Design Components) bear on changes in <u>technical behavior</u> (the attributes of Roles / Requirements)

Key methodology point:



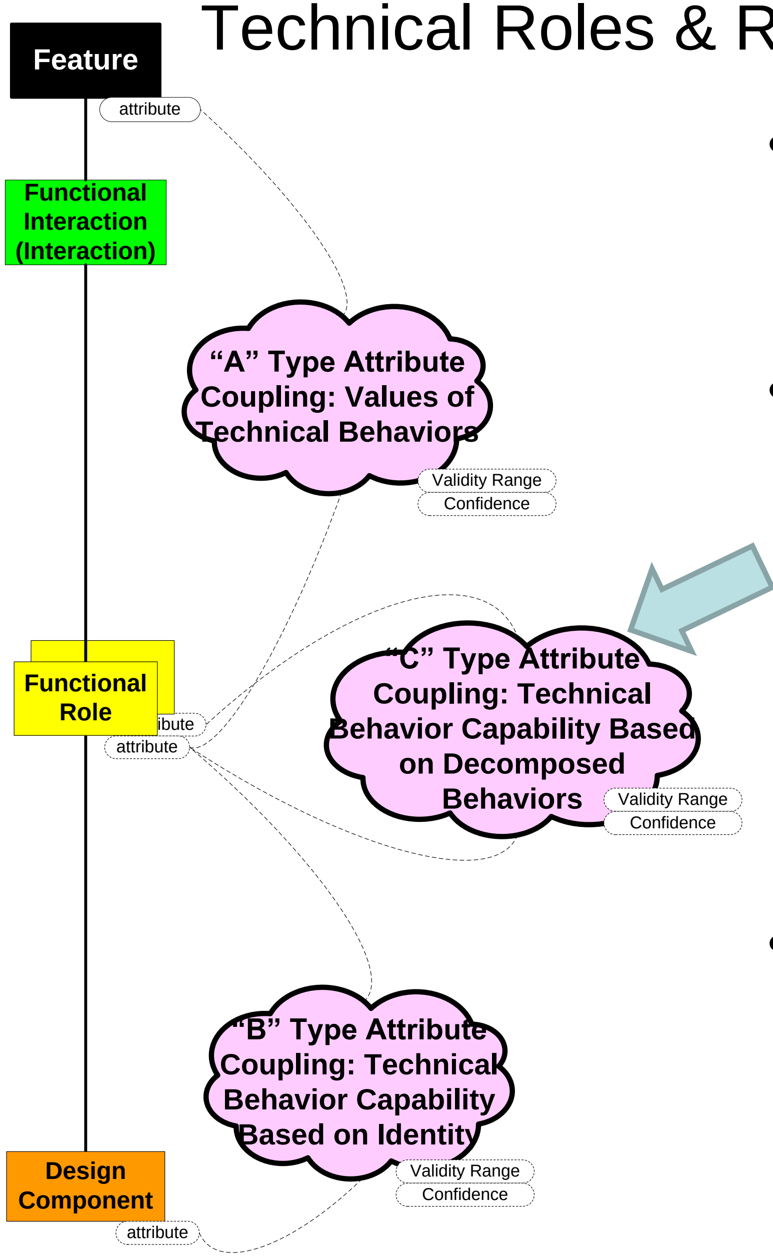
- Modeled technical behavior (including its parameterization) is focused in the Functional Roles (including their parameterization by Role Attributes, which are identical reappearances of the technical Requirements Attributes).
- So, the attributes of Design Components are not used to describe behavior! (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.

Our team's "Wave 1 projects" 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)

Our team's "Wave 2 projects" 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