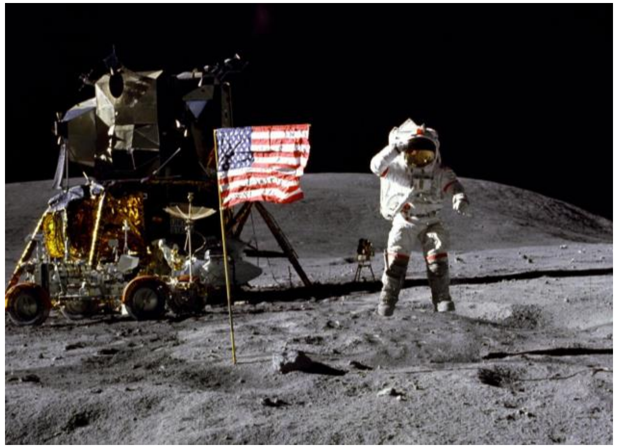
Feature Space: Integrating Value, Purpose, Mission, Risk, and Variation



INCOSE North Texas Chapter Meeting

June 14, 2022 V1.2.2

Bill Schindel
INCOSE Patterns Working Group
schindel@ictt.com

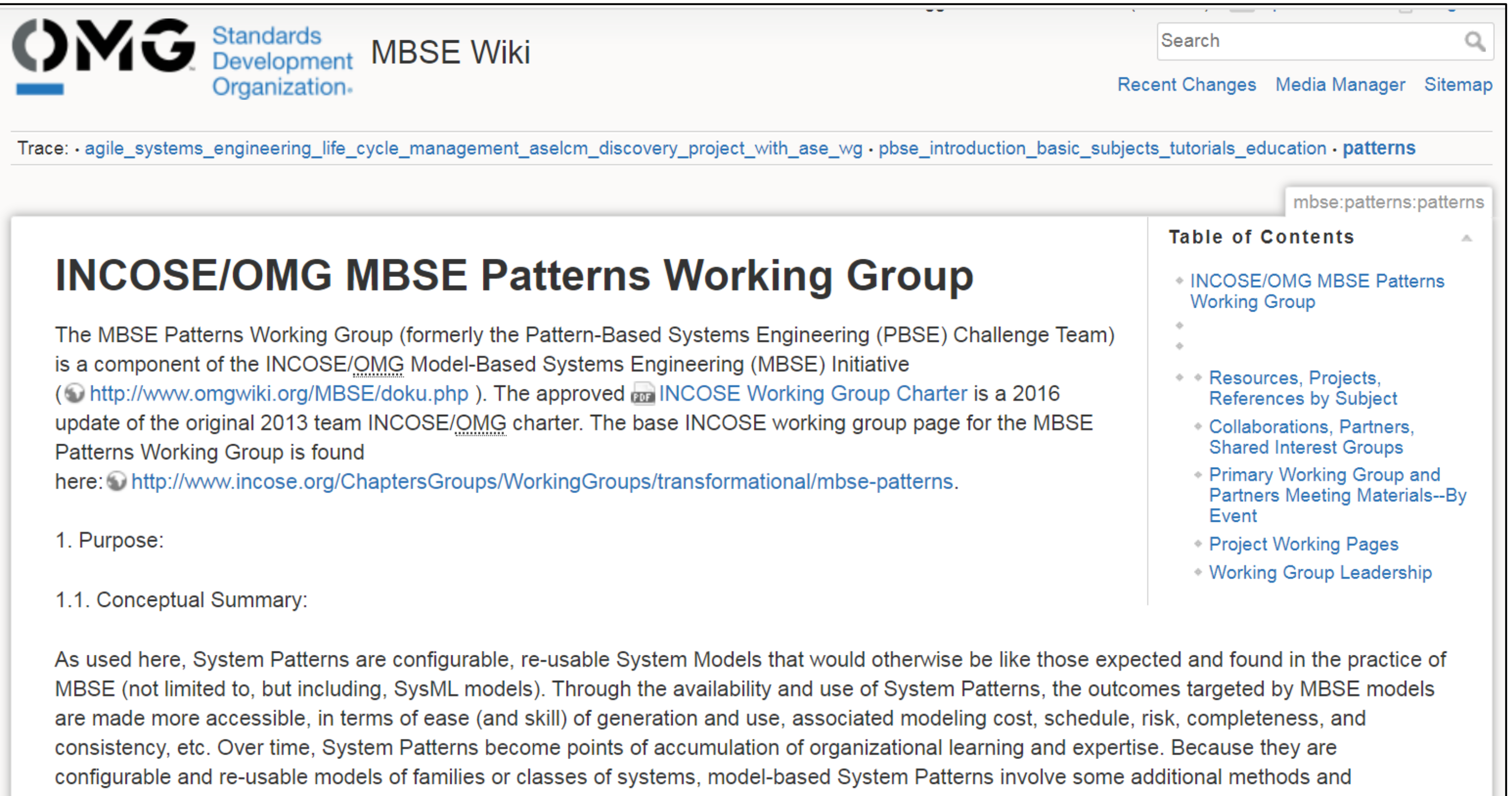
Abstract

- Model-based systems engineering offers the possibility of clarity of models that powered the scientific revolution. Among the surprising results of this is realization that, for appropriately structured models, some seemingly separate aspects of engineering can be combined into a simpler integrated representation.
- Engineers are accustomed to thinking of mission engineering, stakeholder needs analysis, requirements engineering, optimization of design, performing risk analysis, and engineering of product line variants as a series of related but different subjects that collectively add up to a complex problem. However . . .
- In this talk, we will summarize some implications of the question "What is the smallest model of a system?", for purposes of engineering and science across life cycles. We will take a look at Feature Space, how it reduces degrees of freedom to give a clearer integrated view of system value, purpose risk, and varied configuration, along with SysML realization of this approach.

Contents

The INCOSE MBSE Patterns Working Group

- Originated in 2013 as one of the INCOSE-OMG MBSE Initiative challenge teams, advancing in 2016 to INCOSE Working Group.
- Focused on model-based representation of recurring, configurable system-level patterns.
- History of projects emphasizing collaboration with other technical societies & INCOSE Working Groups.
- Numerous publications and resources available for download from Patterns Working Group web site-- https://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns (*Note this is on OMG Wiki!*)
- You are invited to participate!


OMG Standards Development Organization. MBSE Wiki

Search

Recent Changes Media Manager Sitemap

Trace: · agile_systems_engineering_life_cycle_management_aselcm_discovery_project_with_ase_wg · pbse_introduction_basic_subjects_tutorials_education · **patterns**

mbse:patterns:patterns

INCOSE/OMG MBSE Patterns Working Group

The MBSE Patterns Working Group (formerly the Pattern-Based Systems Engineering (PBSE) Challenge Team) is a component of the INCOSE/OMG Model-Based Systems Engineering (MBSE) Initiative (http://www.omgwiki.org/MBSE/doku.php). The approved INCOSE Working Group Charter is a 2016 update of the original 2013 team INCOSE/OMG charter. The base INCOSE working group page for the MBSE Patterns Working Group is found here: http://www.incose.org/ChaptersGroups/WorkingGroups/transformational/mbse-patterns.

1. Purpose:

1.1. Conceptual Summary:

As used here, System Patterns are configurable, re-usable System Models that would otherwise be like those expected and found in the practice of MBSE (not limited to, but including, SysML models). Through the availability and use of System Patterns, the outcomes targeted by MBSE models are made more accessible, in terms of ease (and skill) of generation and use, associated modeling cost, schedule, risk, completeness, and consistency, etc. Over time, System Patterns become points of accumulation of organizational learning and expertise. Because they are configurable and re-usable models of families or classes of systems, model-based System Patterns involve some additional methods and

Table of Contents

32nd Annual INCOSE international symposium

hybrid event

Detroit, MI, USA
June 25 - 30, 2022

MBSE Patterns Working Group will meet at IS2022 on Sunday, June 26, at 1:30 ET—if you are not on site, you can join virtually: https://www.incose.org/symp2022/symposium/event-schedule

What is the smallest model of a system?

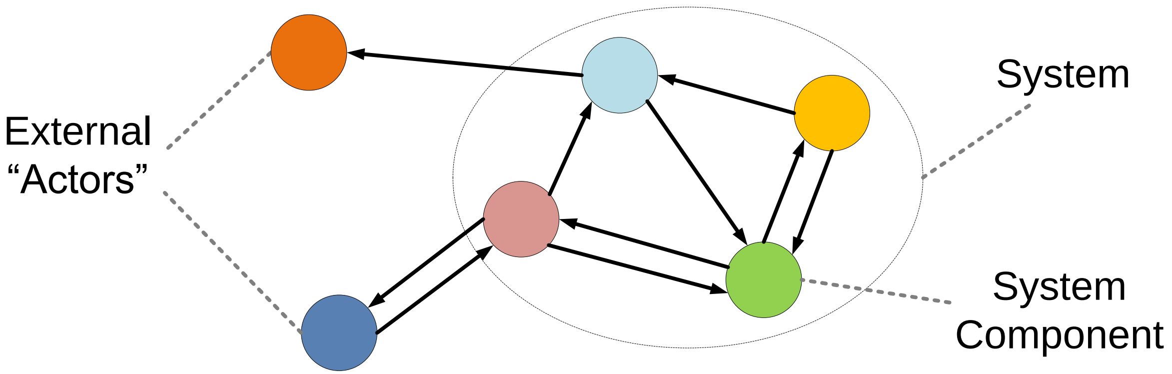
- The use of model-based recurring patterns is at the center of the explosive success of science, engineering, and mathematics in transforming the human-experienced world over the last 300 years.
- In pursuing the use of model-based patterns for systems engineering, we soon realized that the underlying theory supporting MBSE would need to be strengthened to include the model-centric lessons those 300 years--if MBSE is to have the kind of transformative impact *in practice* that STEM has had.
- The beginning of that process, twenty years ago, was to ask the question:

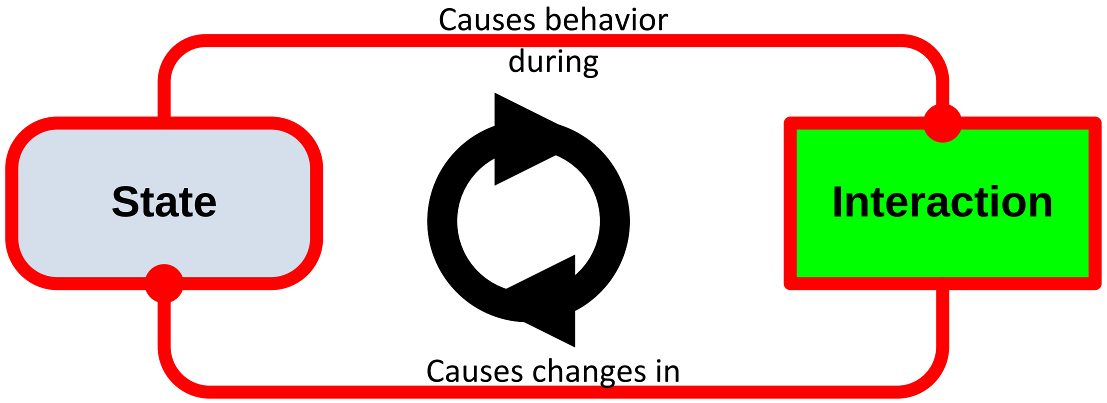
> *"What is the smallest model of a system, for purposes of science and engineering, over the life cycles of systems?"*

- Our program this evening is about a few aspects of where that led.

Formalizing a Few Representational Concepts

- Definition*: In the perspective described here*, by "System" we mean a collection of interacting system components:



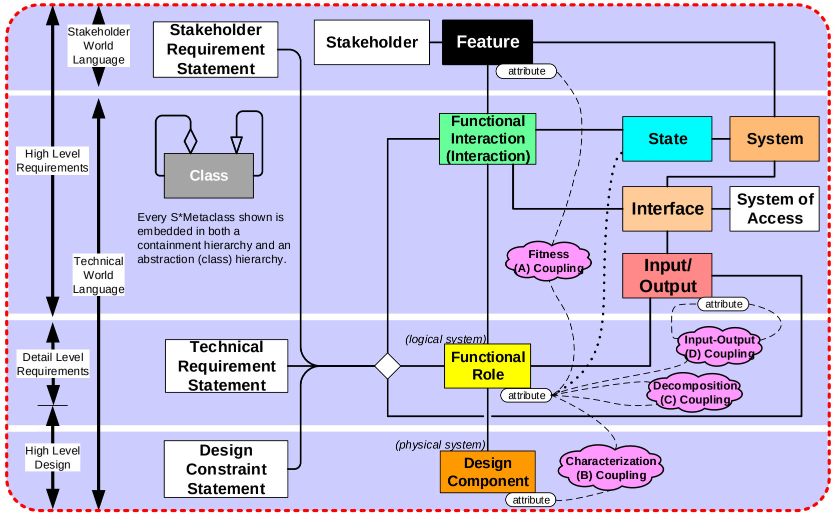


- By "interacting" we mean the exchange of energy, force, material, or information (all of these are "input-outputs") between system components, . . .
- . . . through which one component impacts the state of another component.
- By "state" we mean a property of a component that impacts its input-output behavior during interactions. (Note the circular cause-effect definition chain here.)
- So, a component's "behavior model" describes input-output-state relationships during interaction—*there is no "naked behavior" in the absence of interaction.**
- The behavior of a system involves emergent *states of the system as a whole*, exhibited in its behavior during its own external interactions, resulting in observable holistic aspects.

(* This means that if you are trying to model system behavior outside of Interactions, you may be fooling yourself.)

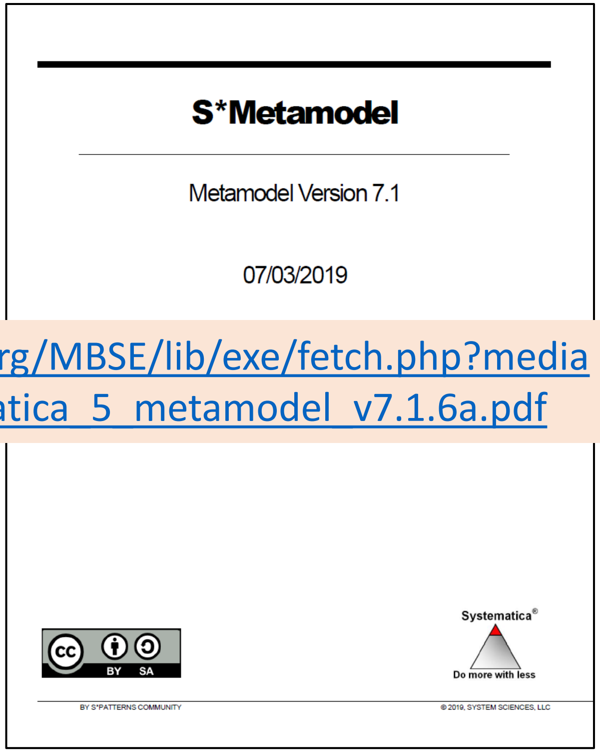
What have we learned in 300 years?

- The history of science and engineering offers plenty of insight about this subject, which is much older and deeper than what we today call “MBSE”.
- What is the practical significance of this to SE practice?
- Important because contemporary MBSE models often:
  - Are missing key aspects (are too small)
  - Contain redundant conflicting aspects (are too big)
  - At the same time!
  - We will be discussing prominent examples of both.
- The S*Metamodel is a formalization of that minimal content—much of which is familiar, but some of which is less visible in current practices we observe.
- This is not about an alternative modeling language or tooling—the agnostic S*Metamodel has been mapped to contemporary languages and tooling (including OMG SysML) for a number of years, and works just fine in current COTS modeling tool environments.



Informal pedagogical S*Metamodel subset

Formal S*Metamodel
(~100 pages UML & prose)


S*Metamodel

Metamodel Version 7.1

07/03/2019

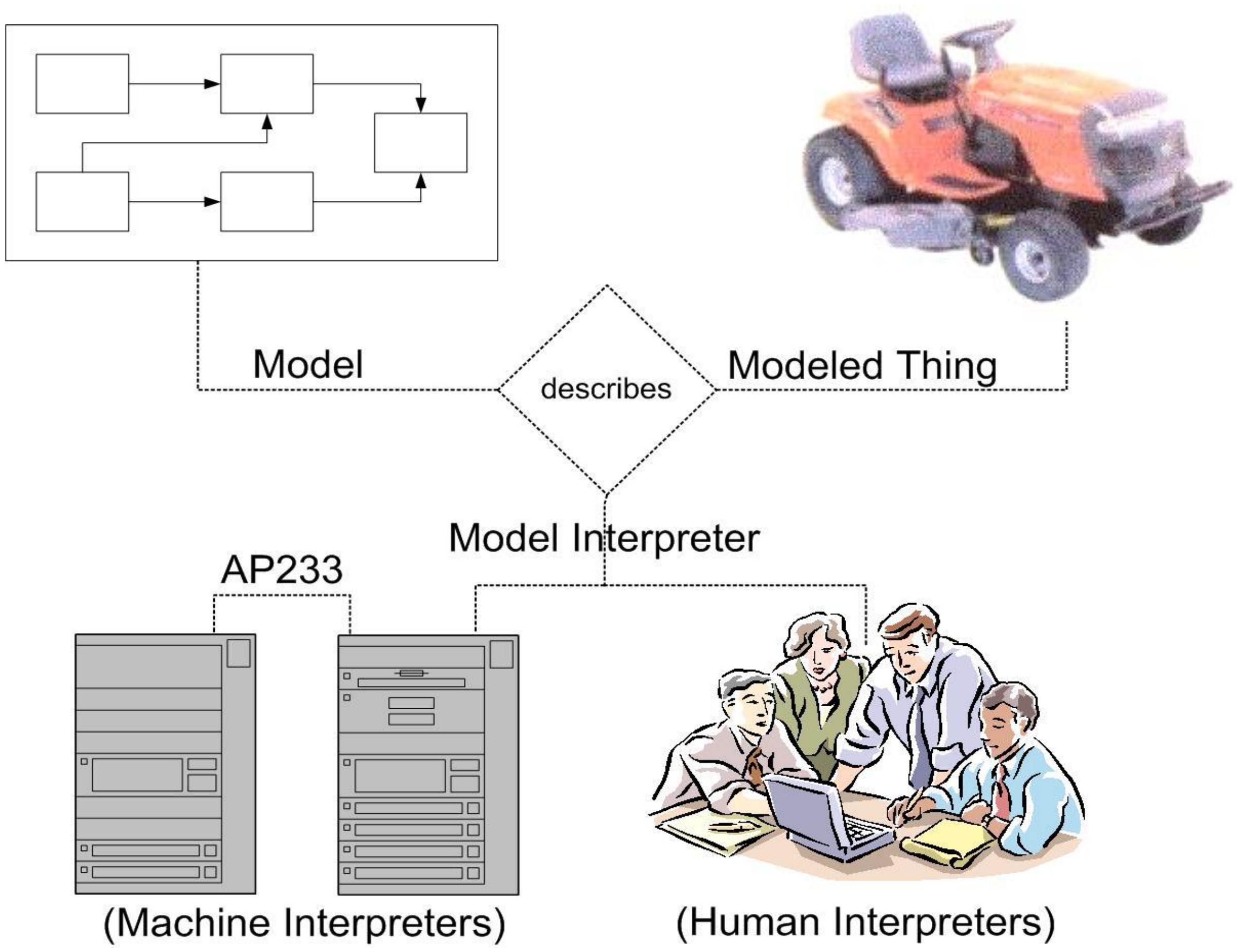
Systematica®
Do more with less

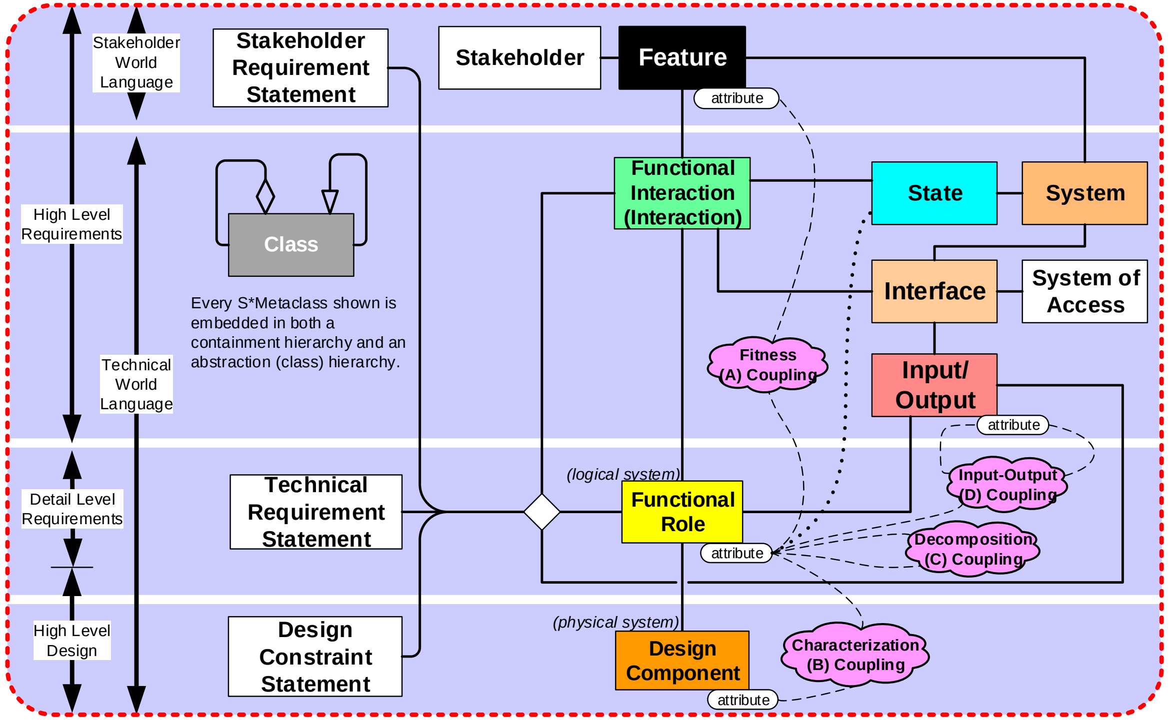
BY S*PATTERNS COMMUNITY

© 2019, SYSTEM SCIENCES, LLC

https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:systematica_5_metamodel_v7.1.6a.pdf

An S*Model is any model (descriptive information construct) of a system (in any modeling language, views, or tooling) which conforms to (maps to) the S*Metamodel:

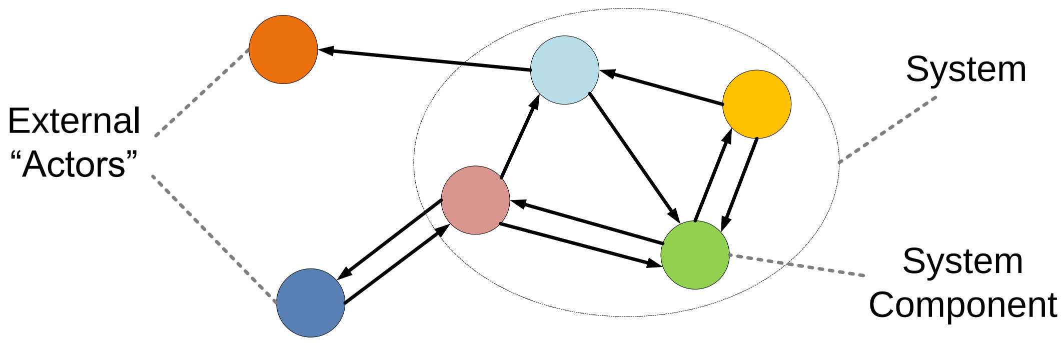




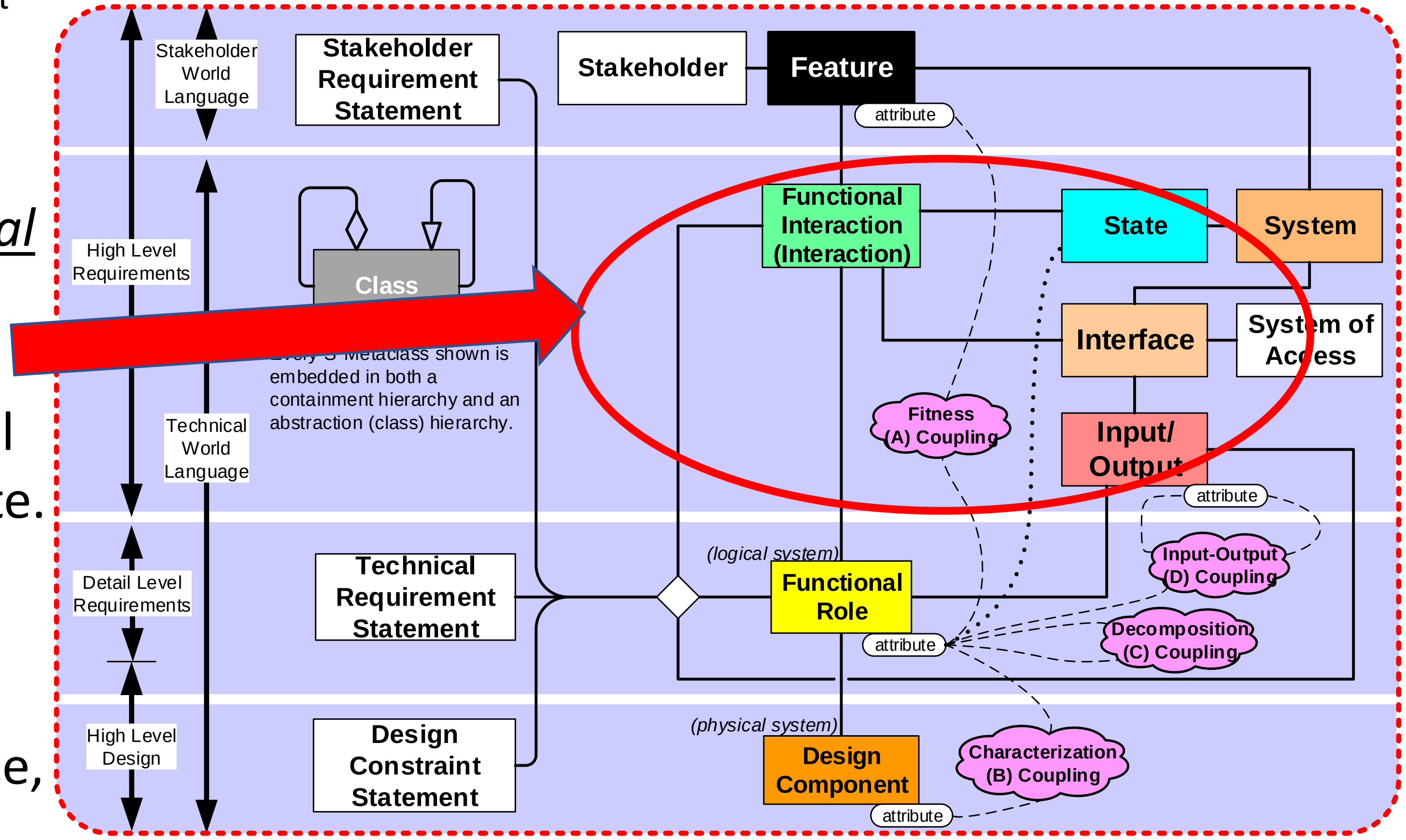
Some of the types of information found in S*Models (independent of language or tooling).

S* is short for Systematica

What lessons of 300 years? One Aspect is External Behavior as Interaction



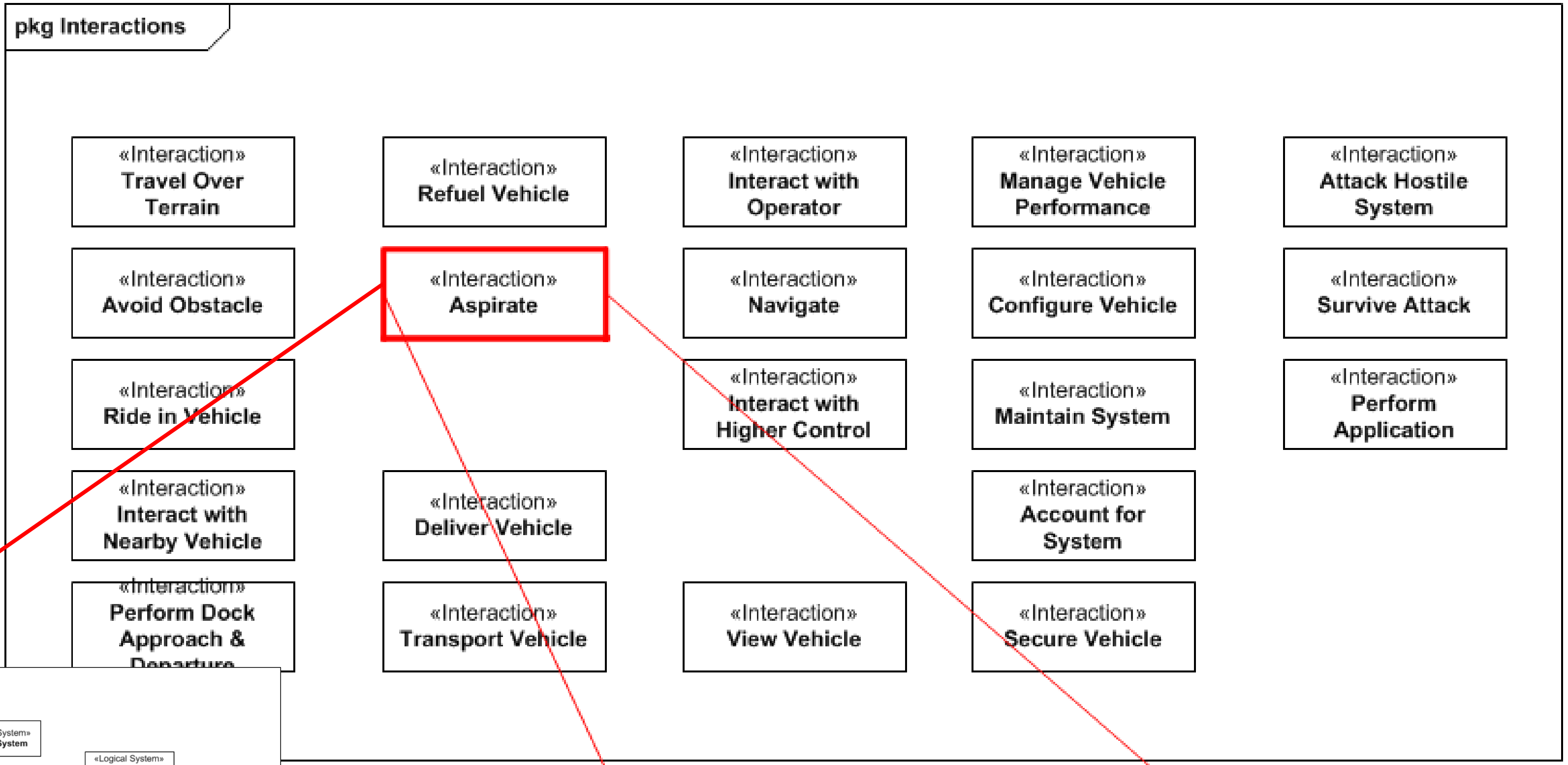
- **Interaction/Behavior Space**: Describes Behavior as *Interactions with Environmental Actors*. An objective description more familiar to scientist and engineers—exchanges of forces, energy flows, material flows, information, causing changes of state.
- **Example**: Travel Over Terrain
- The world of the physical sciences—all the known laws of mechanics, electrical science, etc. are in the context of Interactions.

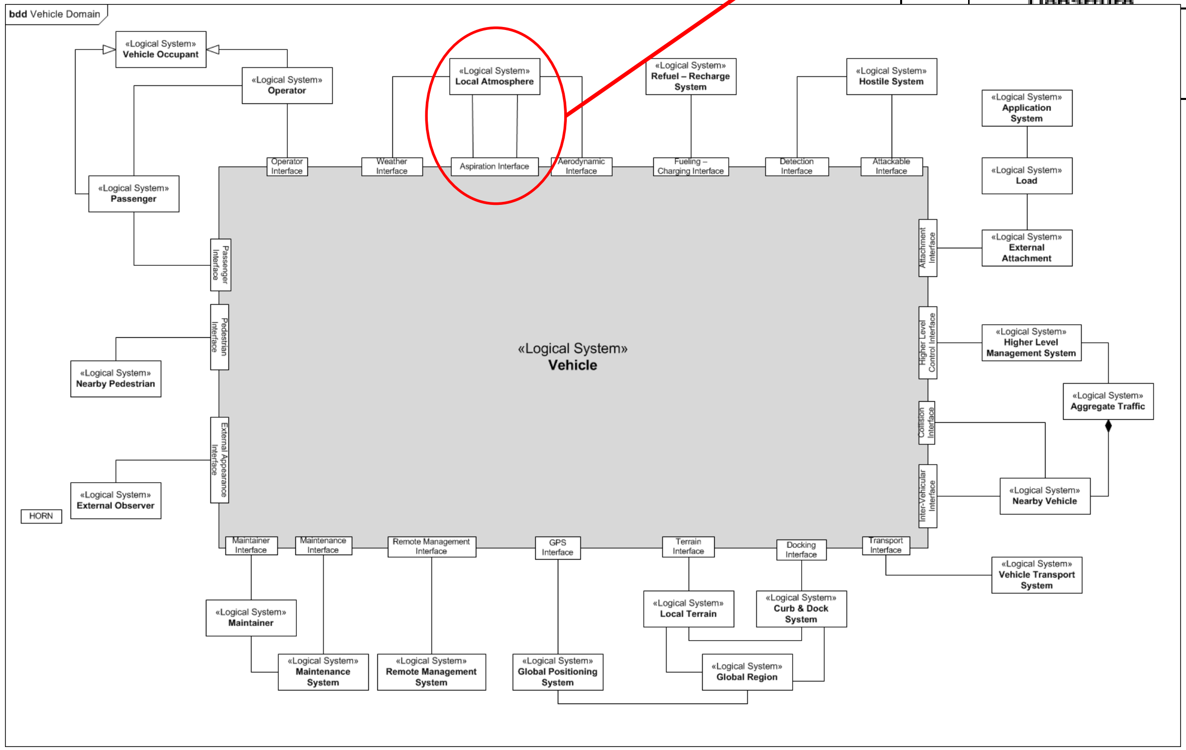


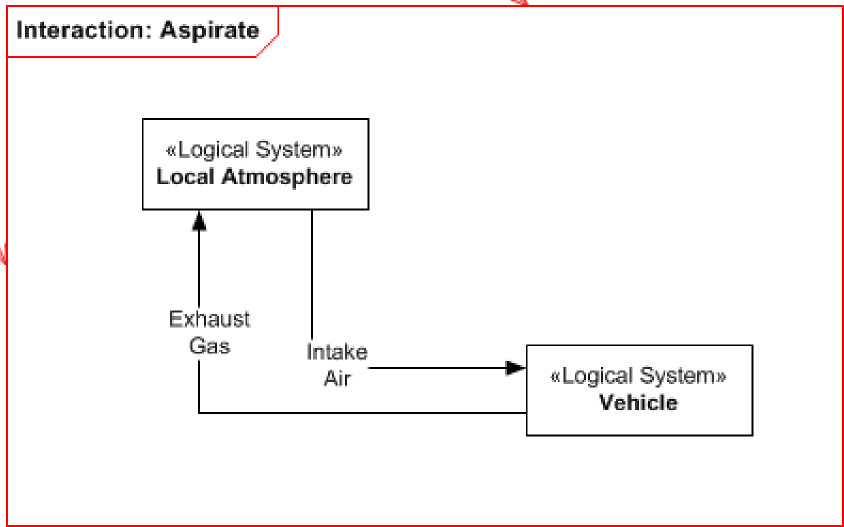
Informal pedagogical S*Metamodel subset

Example: Vehicle Interactions with External Actors

https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:pbse_tutorial_glrc_2016_v1.7.4.pdf



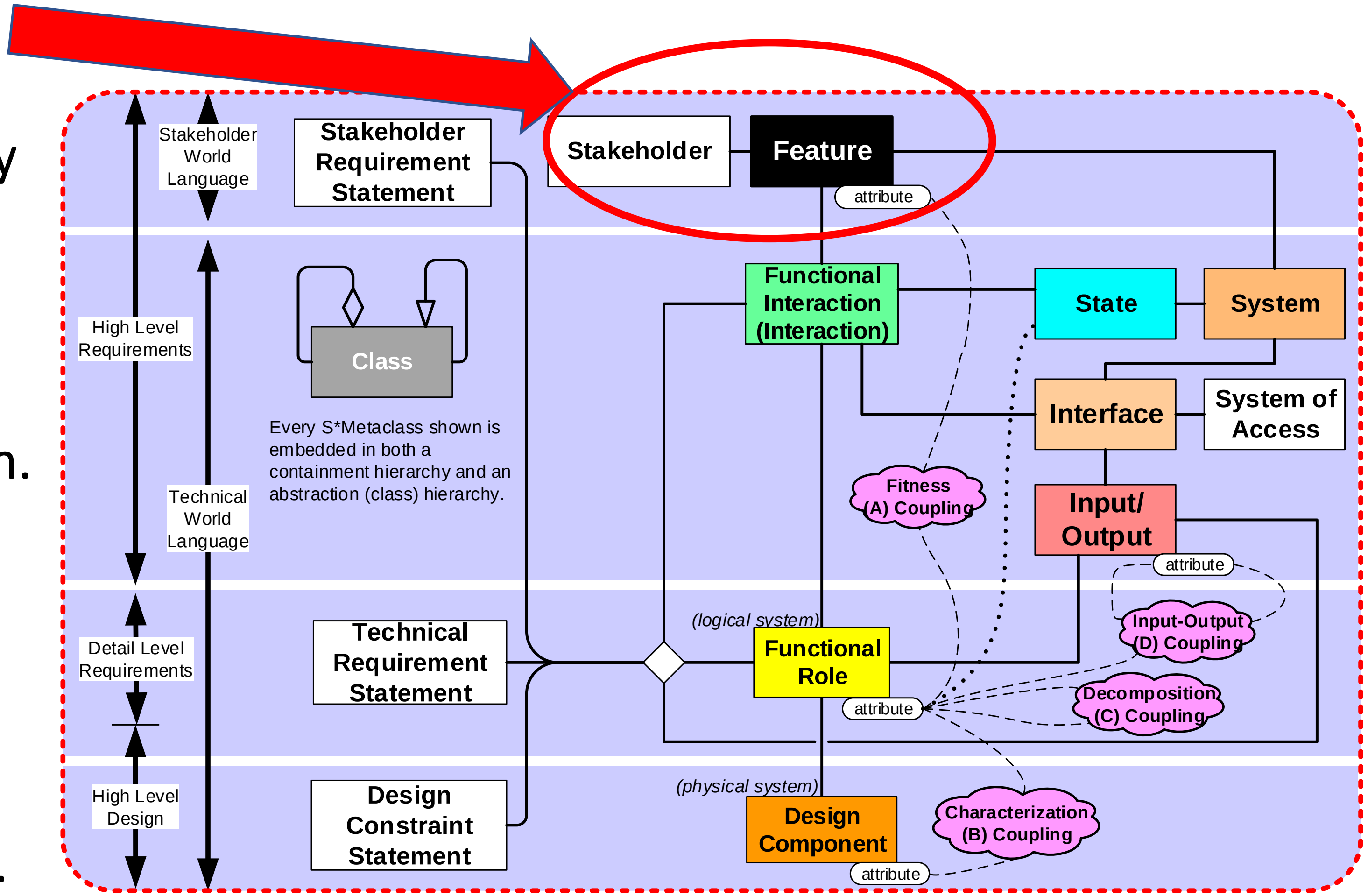




What have we learned in 300 years?

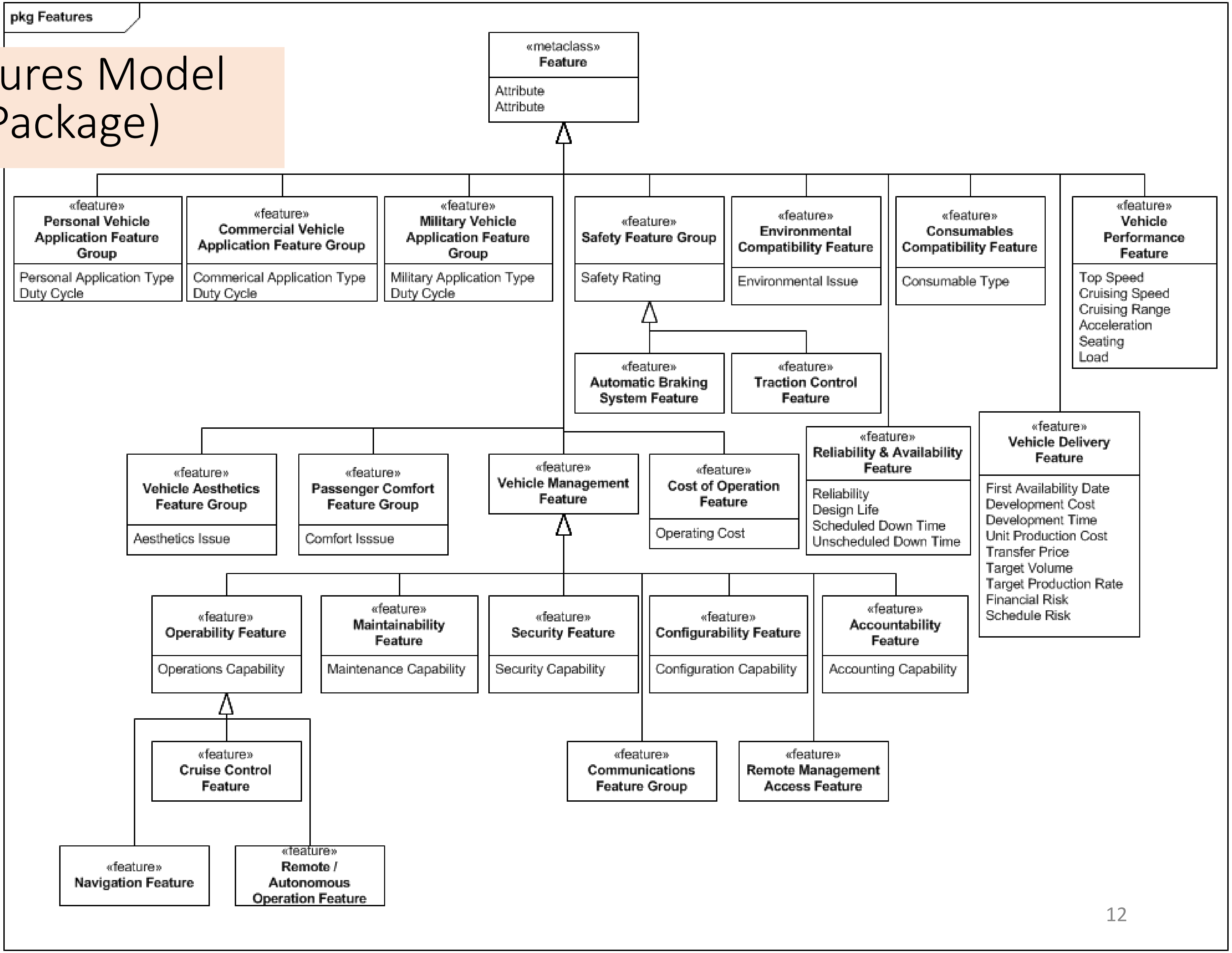
Another Aspect, and a subject of this talk, is Stakeholder Value: Introduction to Feature Space

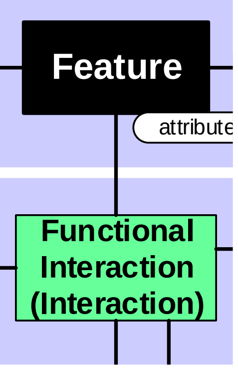
- **Feature Space**: Describes Stakeholder Value Space. This description frequently includes subjective aspects, and is *describing stakeholders as much as it is describing the system of interest*.
- Scoreboard/tradespace for optimization.
- The structure & semantics (not just quantitative) of value.
- The world of Stakeholder Needs, Customer Requirements, Capabilities, Preferences & Priorities, Release Trains.
- Connecting models to the C-Suite.



Informal pedagogical S*Metamodel subset

Example: Vehicle Features Model (in a SysML Features Package)





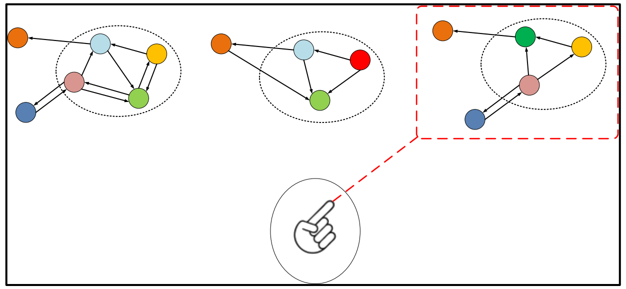
System Behavior is Modeled Twice (Features and Interactions)

Objective-Subjective Link & The Value Selection Phenomenon

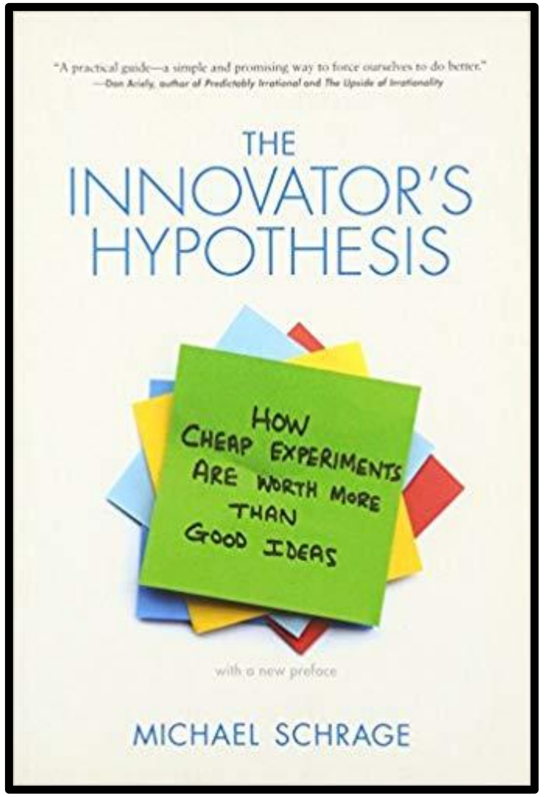
- Engineers know that value is essential to their practice, but its "soft" or subjective nature seems challenging to connect to hard science and engineering phenomena.
- System engineers currently learn to seek out and represent (may model in detail) stakeholder needs, measures of effectiveness (MOEs), objective functions connected to derived requirements and technical performance, etc. to answer: *what value does your system contribute*?
- This nearly always includes "conflicting" dimensions of value, when "trade space" value dimensions appear to trade against each other—as in performance vs. cost. The resulting balancing act led to notions of Pareto Frontiers and other multi-variate forms, Arrow's Impossibility Theorem, and other formulations and insights about the dimensions of value.
- For many systems, lack of good knowledge (*by even the customer*) about value has changed engineering into a *discovery* project, as in Agile Methods, Minimum Viable Products, Pivoting, Hypothesis Experiments, and similar approaches.
- Meanwhile, what are the phenomena associated with value, what is the bridge between subjective value and objective science, where are the related mathematics and recurring patterns, and what are the impacts on future SE practice?
- What follows is not the same as simply "modeling idealized value", which might seem natural but which has some challenges for direct observation.

What is the distinction we are making here?

“Modeling Value” in the traditional sense (e.g., MOEs/Measures of Effectiveness, etc.) sounds a lot like “Modeling Value Selection”—***<u>so what distinction we are making?</u>***



- This is where the “objective science” comes in!
- We are interested in models that can be tested in actual experiments with real selection agents.
- Systems engineering needs to catch up with what business has discovered and put into practice in recent years—driving discovery with real experiments that test the validity of hypothesized value, in a dynamic, pivoting enterprise.
- We are interested in what actual selection behavior tells us about value—not just what isolated offerings of opinion about value or statements of preference. *<u>What really gets selected?</u>*
- That is the distinction of the <u>Value Selection Phenomenon</u>.
- It is a real phenomenon that always occurs and can be observed.
- It also can be *<u>influenced</u>* by advertising, culture, context, bias.
- It can also help us engage the “multi-variate” value challenge.



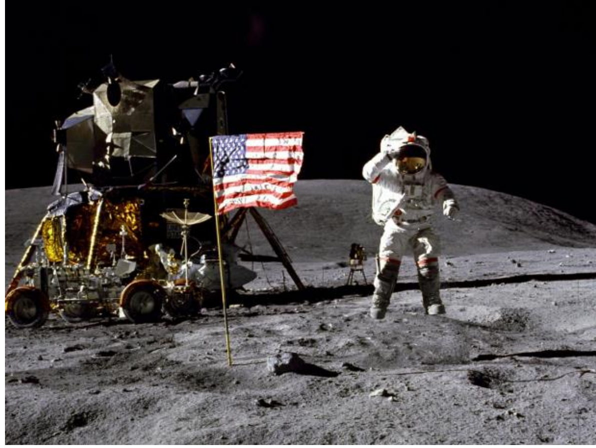
Even if value (both human-based and otherwise) seems elusive or subjective, the expression of value in the real world is always via selection, and selection itself is an *observable interaction-based phenomenon: The Value Selection Phenomenon*:

Settings	Types of Selection	Selection Agents	
Consumer Market	Retail purchase selection	Individual Consumer; Overall Market	
Operational Use	Decision to use product A or use product B	User	
Military Conflict	Direct conflict outcome; threat assessment	Military Engagement	X
Product design	Design trades	Designer	
Commercial Market	Performance, cost, support	Buyer	
Biological Evolution	Natural selection	Environmental Competition	X
Product Planning	Opportunity selection	Product Manager	
Market Launch	Optimize choice across alternatives	Review Board	
Securities Investing	What to buy, what to sell, acceptable price	Individual Investor; Overall Market	
College-Student "Matching Market"	Selection of individuals, selection of class profile, selection of school	Admissions Committee; Student & Family	
Life choices	Ethical, moral, religious, curiosities, interests	Individual	
Democratic election	Voting	Voters; Voting Blocks	
Business	Risk Management, Decision Theory	Risk Manager, Decision Maker	

Not all selection is by human agents

Mission; Mission Features; Mission Engineering

https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2016/systems/18950_RobertGold.pdf



- The Defense community in particular has increased emphasis (especially the last ~5 years) on "Mission Engineering"-- this includes explicitly representing Mission in system models.
- This healthy trend could be viewed as catching up with some commercial systems engineering—improving the representation of what holistic engineered domain systems are intended to accomplish.
- Mission Space is a subset of Feature Space, concerned with primary purpose of either an engineering Systems of Interest (e.g., an Aircraft) or the larger domain systems of which they are a part (e.g., a Task Force):
  - Example: A10 Aircraft Close Air Support Mission.
  - Some models show multiple levels of systems, so we can think about their respective Missions.
- Not all important stakeholder Features or capabilities are Mission or Primary Purpose level Features:
  - Remember the "-ilities"? (e.g., the Maintainability Feature)
  - Remember system management functional areas (SMFAs) "FCAPS"? (e.g., the Security Management Feature)
- Configurable patterns (Mission Packages) of Mission Features are encouraged for common understanding, sharing across subsystems, programs, suppliers, rapid response, for common language, ontology.

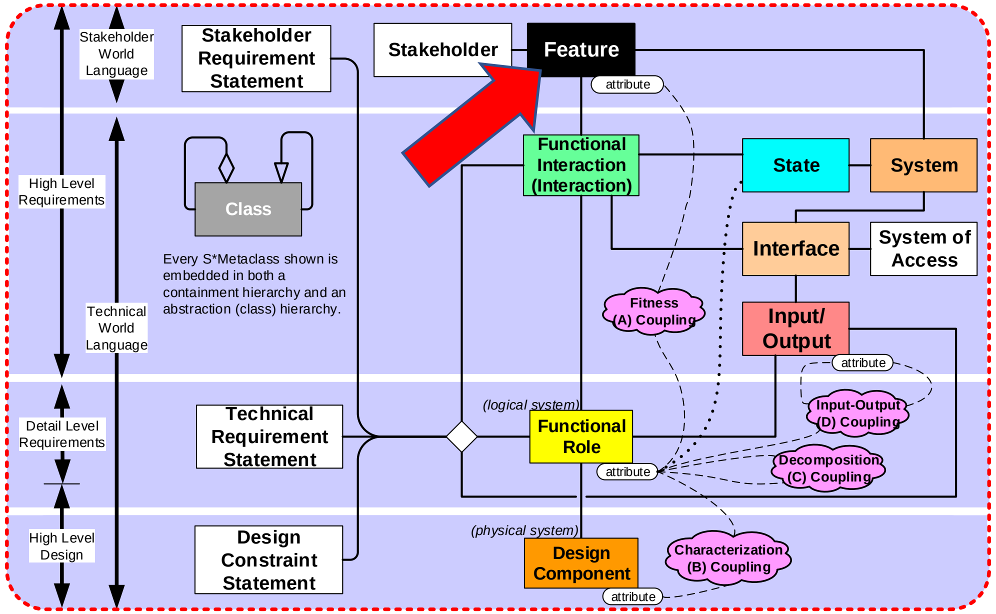
Models of risk: What do we need to represent?

- Traditional systems engineering example risk analysis representations are well-established, and can be found in:
  - Failure Modes and Effects Analysis (FMEA) or Failure Modes, Effects, and Criticality Analysis (FEMCA).
  - Special cases for risks of designs, risks of production and other processes, risks introduced by human operators (D-FMEA, P-FMEA, A-FMEA).
  - Fault Tree Analysis (FTA).
  - Preliminary Hazard Analysis (PHA).
  - Reliability Centered Maintenance (RCM) analysis.
  - Hazards and Operability Analysis (HAZOP).
  - Safety and Cybersecurity Analysis variations on the above.
- Some are used as part of review of a candidate system design, others to analyze an existing system, as in the case of RCM planning of preventive maintenance, etc.

Models of risk: What do we need to represent?

- Most systems engineering relevant definitions of risk include these ideas:
  1. Potential, threat, or hazard of some future harm, injury, death, setback, economic or market loss, impacted assets, loss of advantage, etc.
     - (Sometimes we alternatively model "risk" of a more positive "opportunity" for gains of various kinds--not our current focus in this session.)
  2. One or more causal "modes" which can lead to the negative outcome.
  3. A degree of uncertainty as to that occurrence, whether expressed as likelihood, probability distribution, or otherwise.
  4. Some idea of the severity of impact of such a negative outcome, were it to occur, and what parties would be impacted.
  5. Ability (or inability) to detect whether the negative situation has occurred or is occurring.
  6. Ability (or inability) to mitigate or work around or otherwise tolerate the negative situation, were it to occur.
  7. In some cases, a time evolutionary aspect representing when in time these ideas might apply.
  8. Means of assembling a relatively complete (and potentially large) collection of the above possibilities, and weighing them relative to each other so that attention and resources can be assembled, focused, and allocated to their prevention, mitigation, planning, or other disposition. ("Risk Priority Number" (RPN) scores are common: RPN = Probability x Severity x Detectability.)

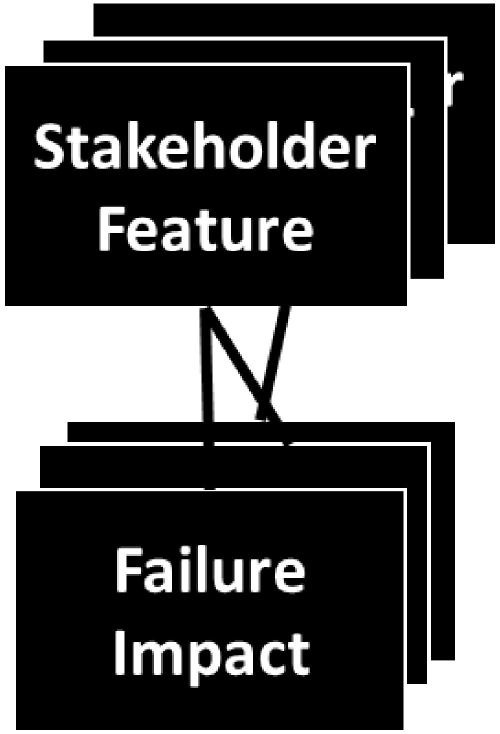
Risk: Failure Impacts



- A pleasant surprise is the discovery that a solid MBSE model of a system's Stakeholder S*Features brings us very close to having a model of all the *potential* negative Effects (the "E" part of FMEA) that a system may present (even if some are not *realizable*):
  - Even if we don't yet know the system's design (!), a fully-modeled Feature Space is a direct path to the Failure Effect Space, because . . .
- ***All risk is risk to Stakeholder Features***:
  - Why? If you ever discover a risk that is not represented in the modeled Feature space of a system, then you have just found a missing part of the positive Feature model. (Add it.)
  - So, it turns out that the negative FMEA model part improves the positive part of the model, and the positive part of the model improves the negative FMEA part of the model . . .

All Risk is Risk to Stakeholder Features

- Therefore we can view each **Stakeholder Feature** in the model as having one or more associated **Failure Impacts** that are about risks (at least in principle) to stakeholders (should they ever occur).
- Example:
  - Feature = Telephone Voice Communication Service
  - Failure Impact = Loss of Voice Communication Service
- What we mean by these are the "E" part of an FMECA: The "Effect" or consequence of the risk realization.
- Note that one Feature may be associated with multiple Failure Impacts.



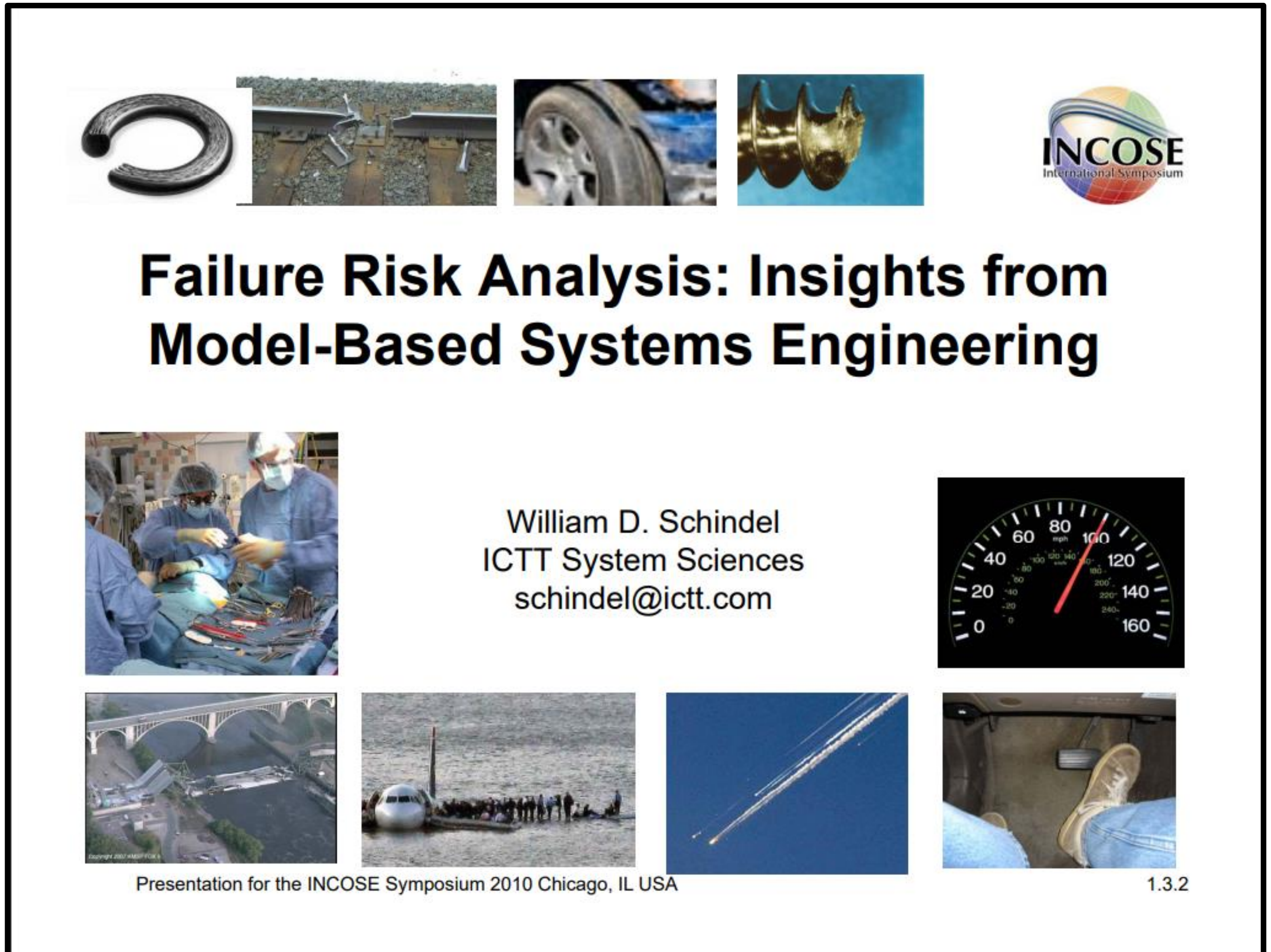
Simple FMEA, Generated from Configurable System Pattern

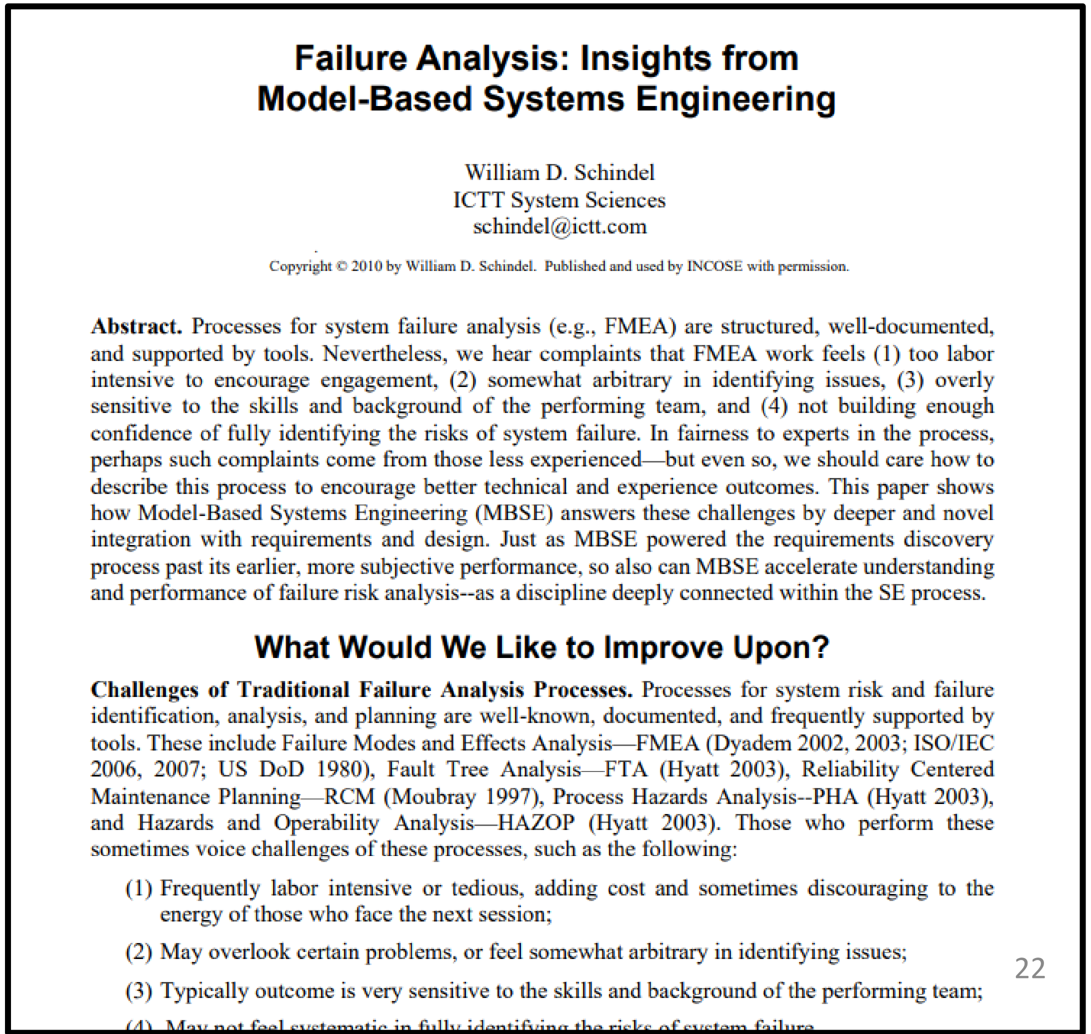
Feature	Effect (Failure Impact)	Severity	Functional Failure (Counter Requirement)	Component	Failure Mode	Probability	Mitigation (Control)
Navigation Feature [GPS-based Location Sensing]	No Confidence in Displayed Position	Serious (4)	The system displays a location that is not accurate to 10 feet.	Vehicle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

https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:pbse_tutorial_glrc_2016_v1.7.4.pdf

To find out more about integrated FMECA

https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:improving_failure_analysis_using_mbse_v1.3.2.pdf





Failure Analysis: Insights from Model-Based Systems Engineering

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

Abstract. Processes for system failure analysis (e.g., FMEA) are structured, well-documented, and supported by tools. Nevertheless, we hear complaints that FMEA work feels (1) too labor intensive to encourage engagement, (2) somewhat arbitrary in identifying issues, (3) overly sensitive to the skills and background of the performing team, and (4) not building enough confidence of fully identifying the risks of system failure. In fairness to experts in the process, perhaps such complaints come from those less experienced—but even so, we should care how to describe this process to encourage better technical and experience outcomes. This paper shows how Model-Based Systems Engineering (MBSE) answers these challenges by deeper and novel integration with requirements and design. Just as MBSE powered the requirements discovery process past its earlier, more subjective performance, so also can MBSE accelerate understanding and performance of failure risk analysis--as a discipline deeply connected within the SE process.

What Would We Like to Improve Upon?

Challenges of Traditional Failure Analysis Processes. Processes for system risk and failure identification, analysis, and planning are well-known, documented, and frequently supported by tools. These include Failure Modes and Effects Analysis—FMEA (Dyadem 2002, 2003; ISO/IEC 2006, 2007; US DoD 1980), Fault Tree Analysis—FTA (Hyatt 2003), Reliability Centered Maintenance Planning—RCM (Moubray 1997), Process Hazards Analysis--PHA (Hyatt 2003), and Hazards and Operability Analysis—HAZOP (Hyatt 2003). Those who perform these sometimes voice challenges of these processes, such as the following:

(1) Frequently labor intensive or tedious, adding cost and sometimes discouraging to the energy of those who face the next session;

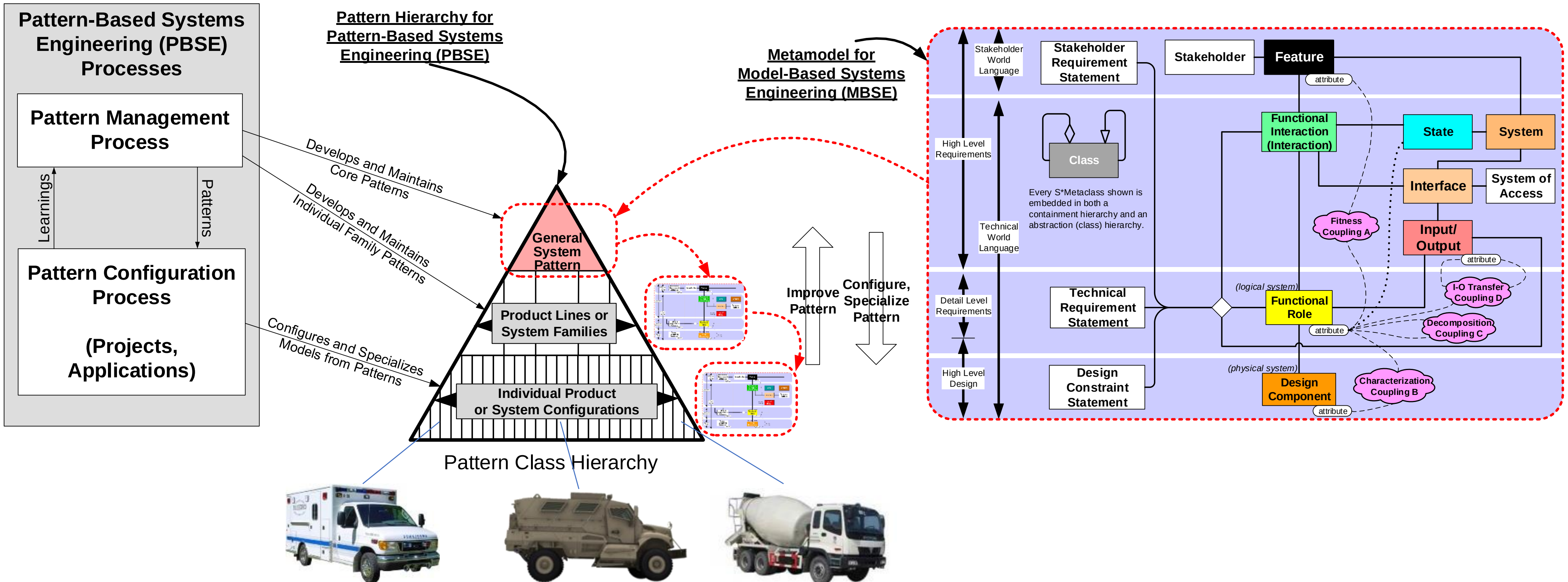
(2) May overlook certain problems, or feel somewhat arbitrary in identifying issues;

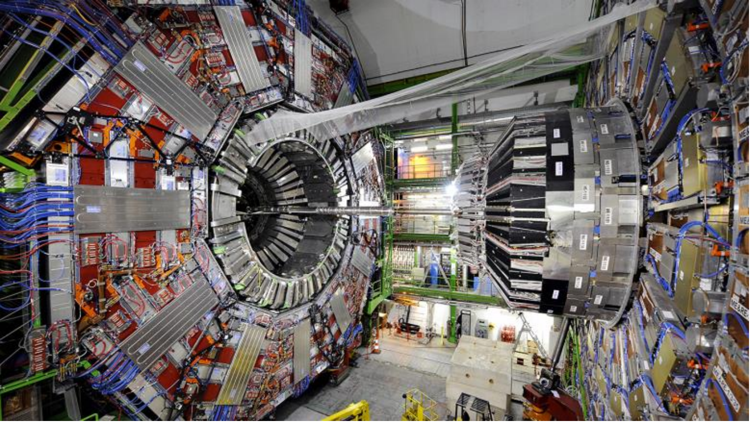
(3) Typically outcome is very sensitive to the skills and background of the performing team;

(4) May not feel systematic in fully identifying the risks of system failure.

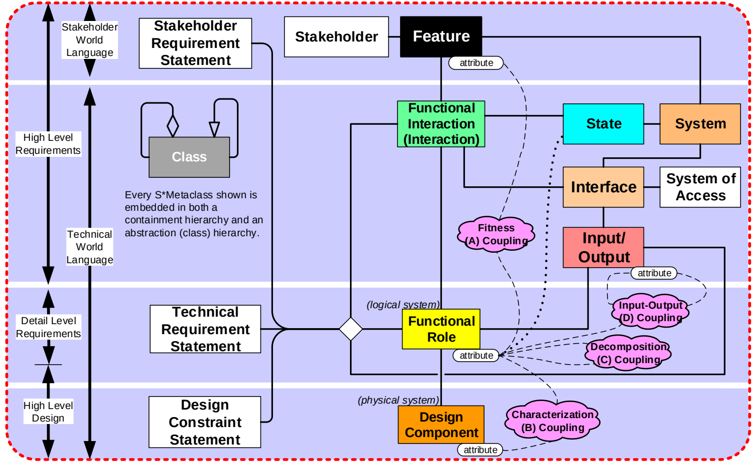
S*Patterns are Configurable MBSE Models of Families

- S*Patterns are configurable S*Models of generalized systems or system families.
- They are intended to be re-configurable, re-usable, and accumulate learning.
- They are often patterns of "whole systems", as opposed to components.
- They are model-based patterns (there is a long history of other patterns).
- As S*Models, they are based on the S*Metamodel.
- Closely related to: Models of Product Platforms, Architectural Frameworks, Ontologies, Configurable Product Line Engineering Models, Domain Specific Modeling Languages

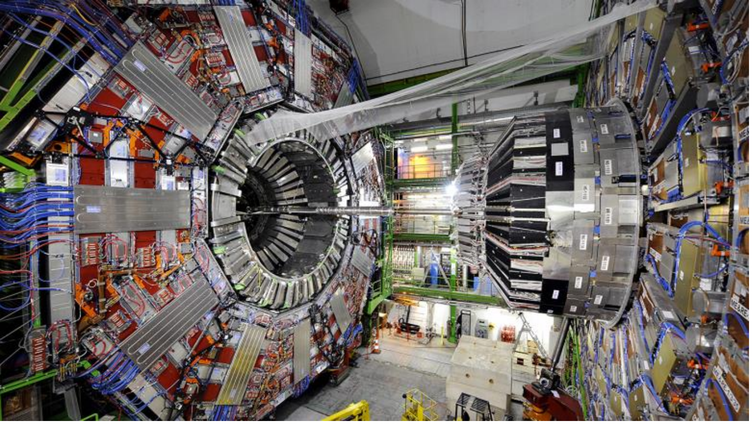




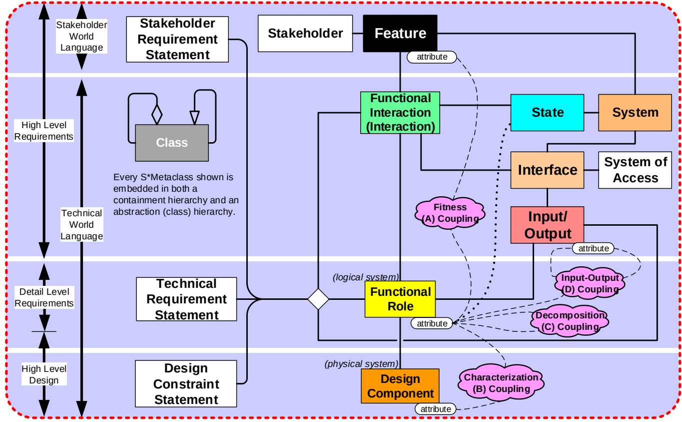
S*Pattern Configuration: Driven from Feature Space



- Engineers and other subject matter experts are familiar with extensive lists of detail configuration variables at many levels:
  - holistic system capability options
  - large subsystem assemblies options
  - small hardware components choices
  - software component variations
  - configurable datasets, option switches, and other managed options
- Most of these points of variation tend to be in the more detailed level lists of numerous smaller elements.
- Collectively, these choices roll up to overall capabilities visible to system stakeholders—those also vary accordingly, but are fewer in number.



S*Pattern Configuration: *Driven from Feature Space*

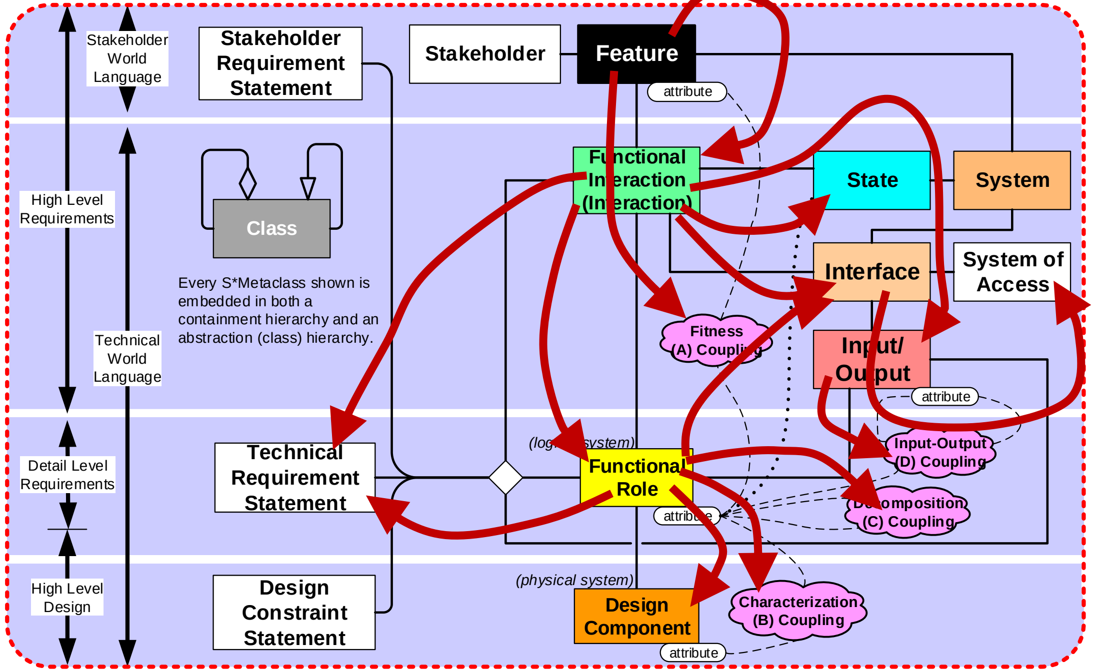


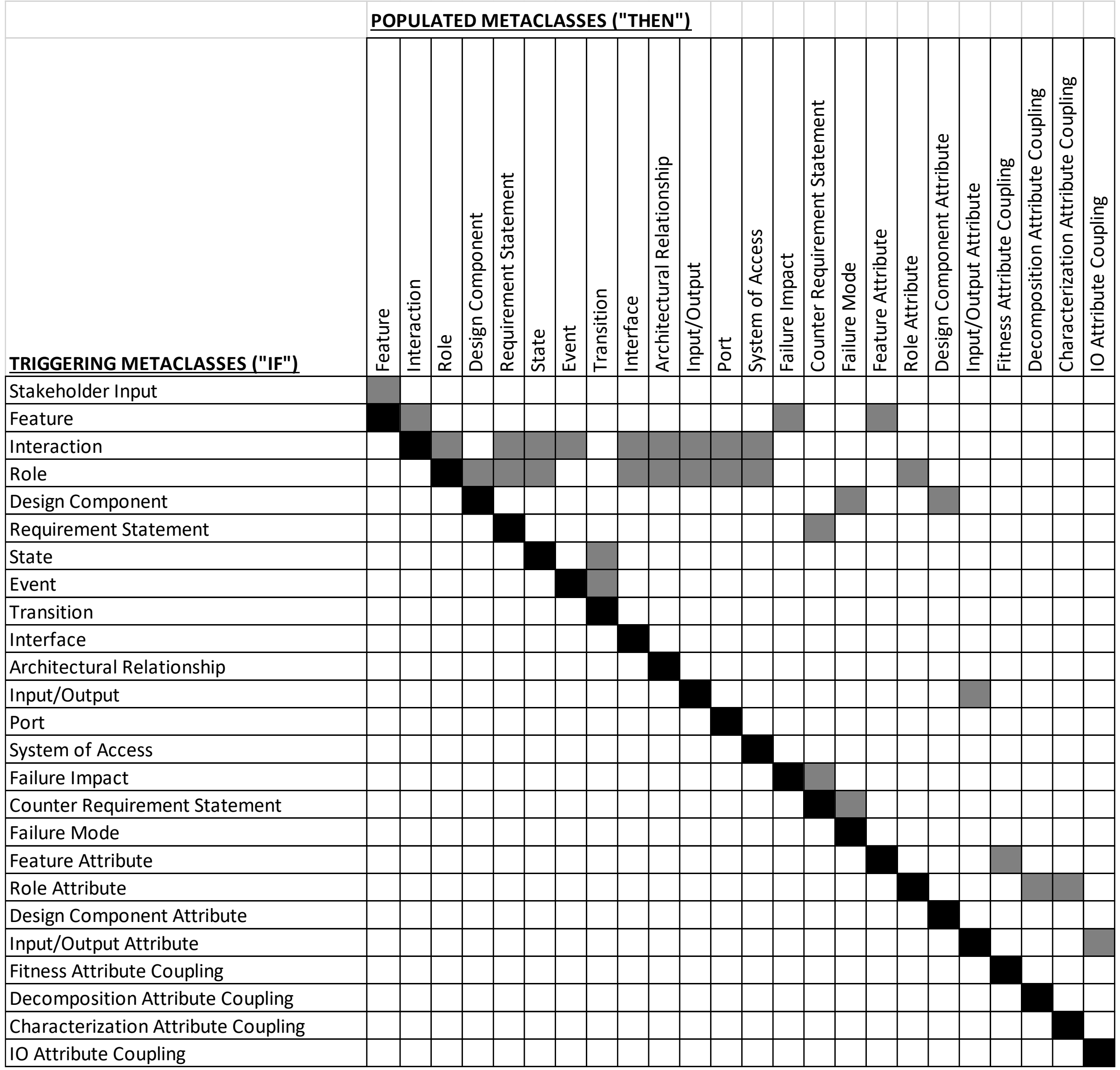
- With that in mind, it can be somewhat surprising to realize that <u>all configurability at any "level" is for Stakeholder Feature level reasons.</u>
- **<u>Obvious question</u>**: What about a small component level variant that you say has no effect significant to any stakeholder?
  - Answer: We can eliminate that component option choice, and no one will care!
- **<u>Obvious objection</u>**: But, what about option choices made to optimize supply chain robustness, or minor behaviors like minor parasitics (e.g., noise).
  - Answer: But you said that no stakeholder cared!
- **<u>Our point here</u>**: Stakeholder Feature Space is supposed to be large enough to cover the life cycle stakeholders who "count", including production, support, shareholders, etc.
- **<u>Implied warning</u>**: So, the term "non functional requirement" can be misleading.

Propagation of configuration population is inherent to the nature of all engineered systems

- S*Feature Space drives configuration from a smaller set of (stakeholder based) degrees of freedom / points of variation.
- Simplifies Product Line Engineering (PLE) model configuration rule-making and integrates PLE.

Inherent PLE degrees of freedom configuration propagation:



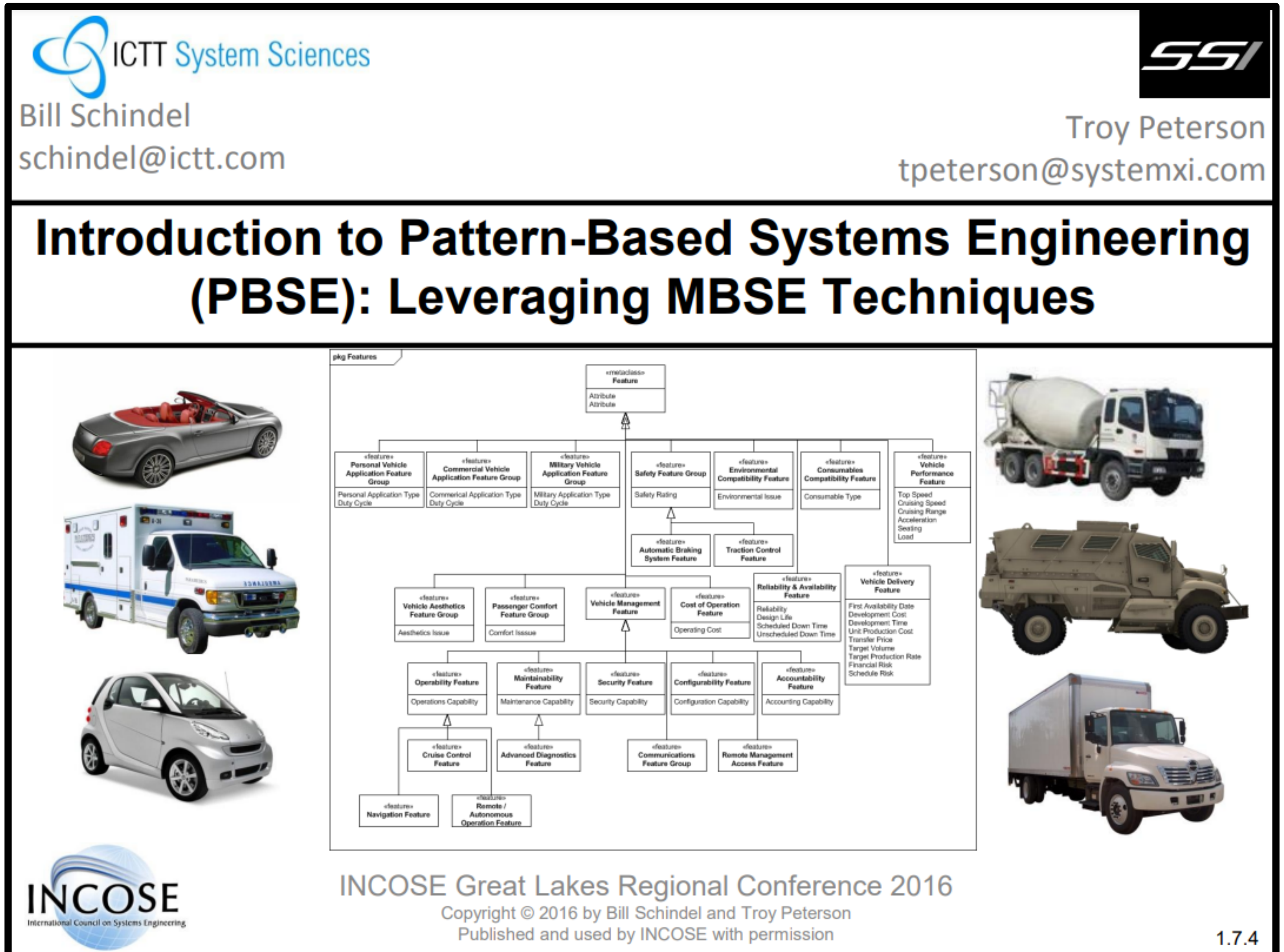


Relationship to Feature-Based PLE ala' ISO 26580

Very similar in the PLE aspects, with a few differences:

- ISO26580 PLE specifies modeling what changes, but specifies omitting what does not change; S*Feature models include baseline capabilities.
- ISO26580 refers to all the points of variation as "Features", with rules to be established between them; S*Patterns begins with a smaller set of "Stakeholder Features" degrees of freedom in stakeholder value space, then recognizes all the other points of variation throughout the model but connects them with each other up to the Stakeholder Features points of variation.
- This shows that the number of real degrees of freedom, after considering constraints, is smaller.
- Effectively complies with ISO26580 while making its use simpler and more integrated.

How to find out more about configurable model-based patterns

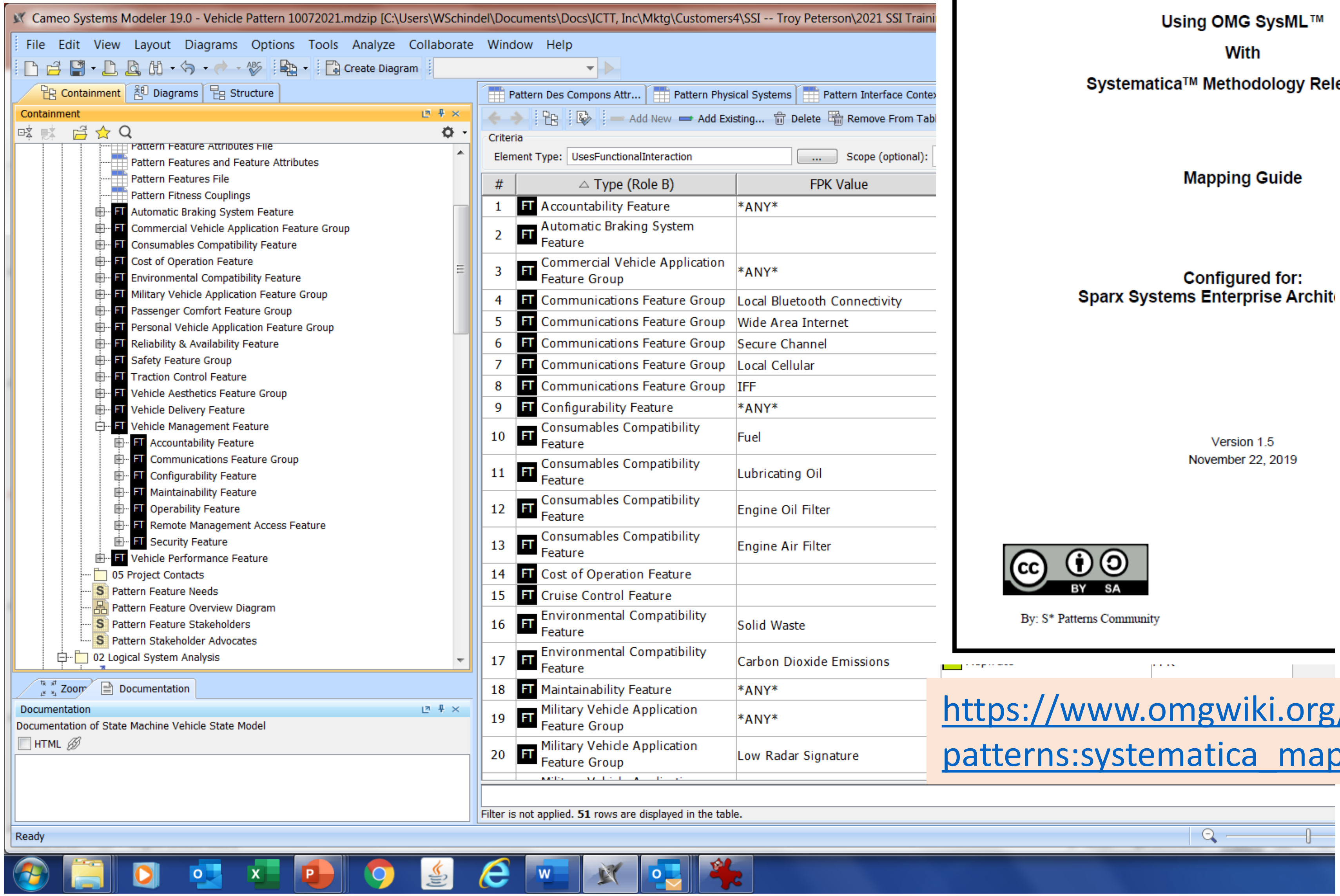


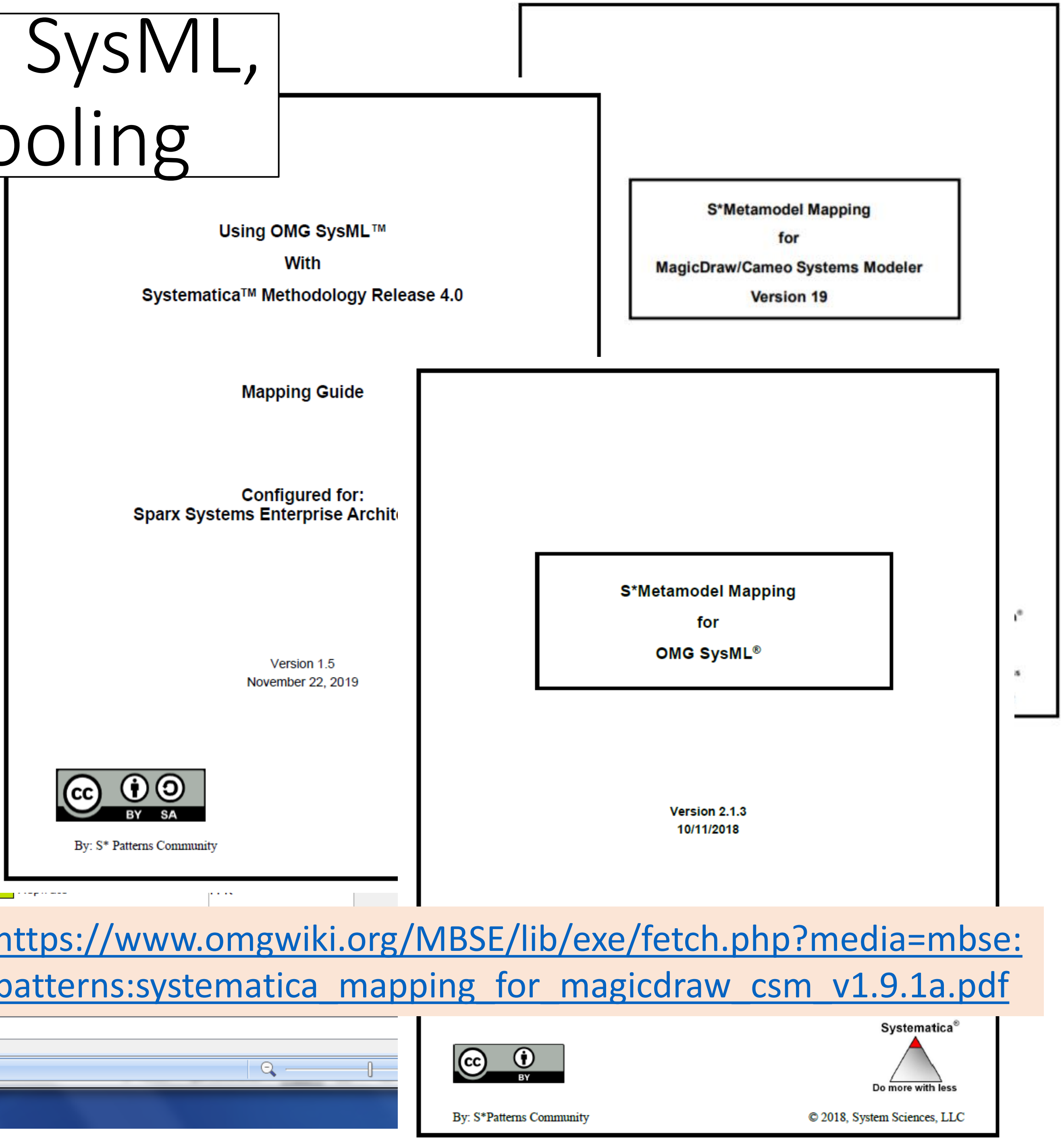
https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:pbse_tutorial_glrc_2016_v1.7.4.pdf

https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:pbse_extension_of_mbse--methodology_summary_v1.6.1.pdf

https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:glrc_2018_tutorial--mbse_emerging_issues_v1.4.2.pdf

Existing mappings into OMG SysML, other languages, and your tooling





https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:systematica_mapping_for_magicdraw_csm_v1.9.1a.pdf

Interested? How to get involved

- INCOSE Patterns Working Group will meet (on site and virtually) at IS2022 on June 26, 1:30-4:30 PM ET: https://www.incose.org/symp2022/symposium/event-schedule

- Or, just contact Bill Schindel schindel@ictt.com

- Current working group projects: https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:iw22_mbse_workshop_round_robin--mbse_patterns_wg_schindel_v1.2.2.pdf

Conclusion, questions, and discussion

<u>Conclusion</u>: Modeling and management of System Value, Mission, Purpose, Risk, and Configurability are deeply connected by Feature Space.

<u>Examples include</u>: Configurability of Mission; Risk to Mission.

<u>Discussion</u>:

-
-
-
-
-
-

Thank you!

References

Integrated Examples:

- Oil Filter "S*MBSE Patterns: A Small Scale Example", https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:oil_filter_example_v1.6.2.pdf
- Feature / Capability Space of the System of Innovation: "The Innovation Ecosystem: Introduction to the INCOSE ASELCM Pattern", INCOSE North Texas Chapter, December, 2021. https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:incose_north_texas_pgm_12.14.2021_v1.2.2.pdf
- Terrestrial Vehicle: https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:pbse_tutorial_glrc_2016_v1.7.4.pdf

Mission Engineering:

- R. Gold, "Mission Engineering", MDIA SE Conference Oct 2015. https://ndiastorage.blob.core.usgovcloudapi.net/ndia/2016/systems/18950_RobertGold.pdf
- R. Giachetti, A. Hernandez, "Mission Engineering", SEBoK. https://www.sebokwiki.org/wiki/Mission_Engineering
- Example defense mission: CV-22 Variant Osprey Aircraft: https://www.af.mil/About-Us/Fact-Sheets/Display/Article/104531/cv-22-osprey/#:~:text=The%20mission%20of%20the%20CV,missions%20for%20special%20operations%20forces
- Example defense mission: Special Operations Wing: https://www.hurlburt.af.mil/About-Us/Fact-Sheets/Fact-Sheets/Article/204524/1st-special-operations-wing/

Integrating Risk Analysis into Feature Space:

- FMECA Slides and paper "Failure Risk Analysis: Insights from Model-Based Systems Engineering" Proc. of INCOSE 2010 International Symposium". https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:improving_failure_analysis_using_mbse_v1.3.2.pdf and https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:impact_of_mbse_on_failure_analysis_v1.2.1.pdf

Integrating Configurability into Feature Space:

- Tutorial: "Introduction to Pattern-Based Systems Engineering(PBSE): Leveraging MBSE Techniques", INCOSE Great Lakes Symposium, 2016. Retrieve from-- https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:pbse_tutorial_glrc_2016_v1.7.4.pdf
- ISO26580 (2021) "Software and systems engineering — Methods and tools for the feature-based approach to software and systems product line engineering", ISO, 2021. https://www.iso.org/obp/ui/#iso:std:iso-iec:26580:ed-1:v1:en

References, continued

Value:

- Schrage, M., *The Innovator's Hypothesis: How Cheap Experiments Are Worth More than Good Ideas* (MIT Press) Hardcover – September, 2014
- The Value Selection Phenomenon: "Discussion Inputs to INCOSE Vision 2035 Theoretical Foundations Section". https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:science_math_foundations_for_systems_and_systems_engineering--1_hr_awareness_v2.3.2a.pdf
- "Value Engineering and Body of Knowledge", SAVE International, 2007. Retrieve from http://www.value-eng.org/pdf_docs/monographs/vmstd.pdf

Pattern-Based MBSE:

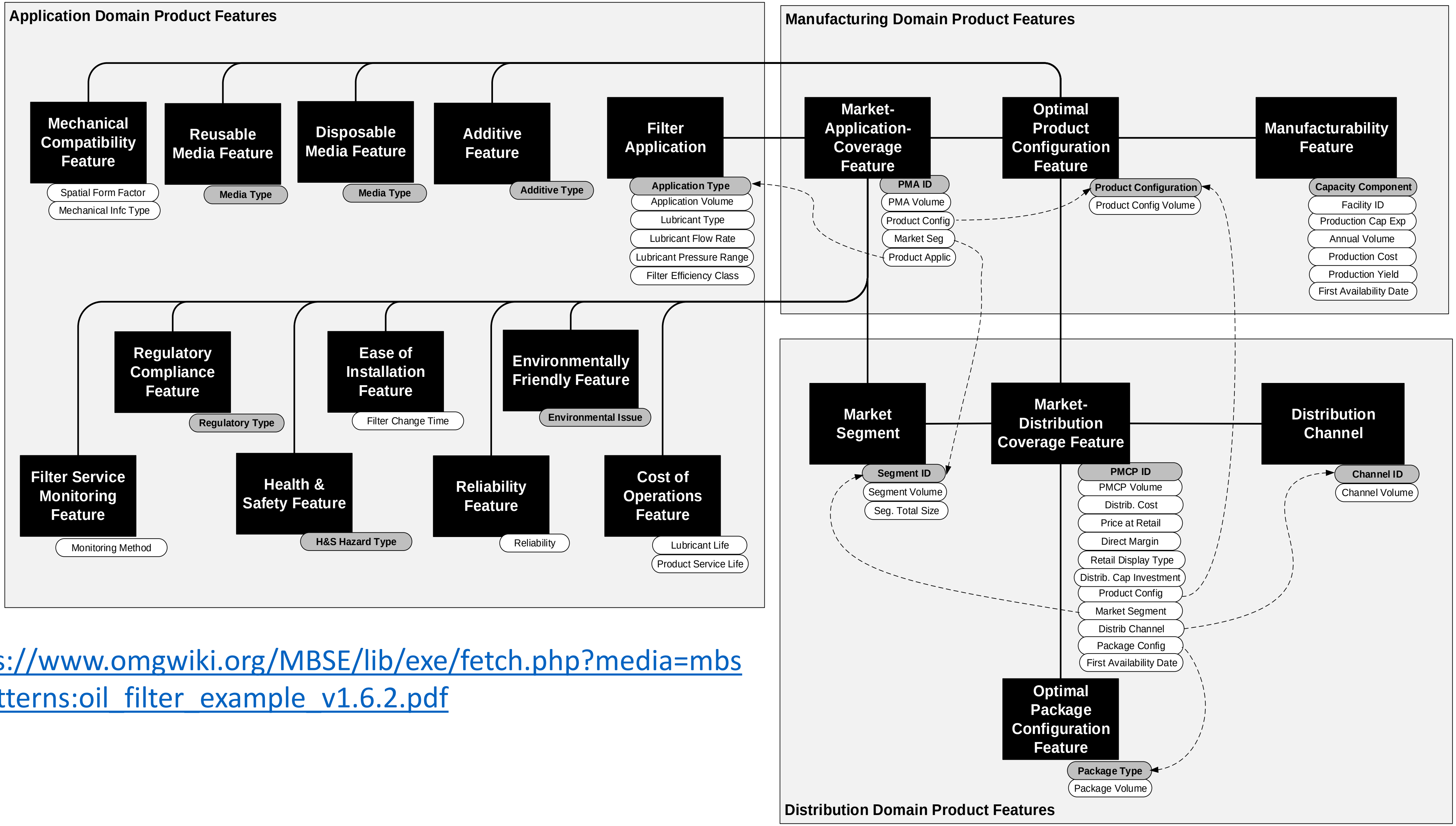
- INCOSE MBSE Patterns Working Group Web Site: https://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns
- PBSE Methodology Reference: "Methodology Summary: Pattern-Based Systems Engineering (PBSE), Based On S*MBSE Models" https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:pbse_extension_of_mbse--methodology_summary_v1.6.1.pdf
- S*Metamodel: https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:systematica_5_metamodel_v7.1.6a.pdf
- S*Metamodel Mapping to OMG SysML Third Party COTS Tooling: https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:systematica_mapping_for_magicdraw_csm_v1.9.1a.pdf

Speaker background

- Bill Schindel is president of ICTT System Sciences. His engineering career began in mil/aero systems with IBM Federal Systems, included faculty service at Rose-Hulman Institute of Technology, and founding of three systems enterprises.
- He chairs the INCOSE MBSE Patterns Working Group, and served on the lead team of the INCOSE Agile Systems Engineering Life Cycle Discovery Project. He is an active member of the ASME VV50 working group on model credibility in advance manufacturing, and the AIAA digital thread and digital twin case study teams.
- Schindel is an INCOSE Fellow and CSEP, and is a director and past president of the INCOSE Crossroads of America Chapter.
- schindel@ictt.com

Supplemental backup

Oil Filter Family: Feature Space Across Domains



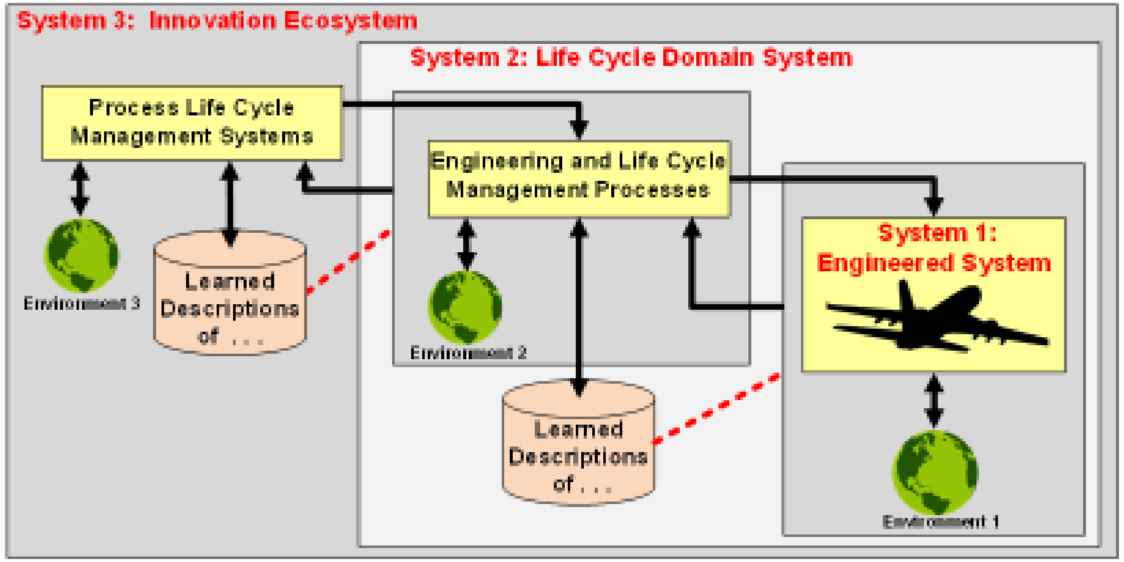
https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:oil_filter_example_v1.6.2.pdf

Three integrated model classes

Failure Mode | **Counter Requirement** | **Failure Impact**

- In the "smallest model" sense, the positive side of a system model is directly integrated with, and testable against, the negative side of the system model.
- Three classes of information are implied, which are commonly seen in FMECAs and other risk analysis (under various names) but not always integrated with their positive counterparts:
- **Failure Mode**: An abnormal state of a subsystem or component that results in abnormal (out of spec) behavior of that subsystem or component. (Part of State Space)
  - Example: Bearing Fracture (Note that you can also model causes, mitigations, detection, etc.)
- **Counter-Requirement**: A requirements-like statement that describes the abnormal (out of spec) behavior of a subsystem or component. (The Functional Failure part of a FMECA; Counter Requirements are a part of Requirements Space)
  - Example: The system fails to transmit rotary torque.
- **Failure Impact**: The effect or consequence impacting a stakeholder, associated with the Feature that is impacted. (The "E" part of a FMECA; part of Feature Space.)
  - Example: Vehicle Transportation Services Not Available

The Innovation Ecosystem: Introduction to the INCOSE ASELCM Pattern



Level 0 View, INCOSE ASELCM Pattern

INCOSE North Texas Chapter Meeting
December 14, 2021
V1.2.2

Bill Schindel
INCOSE Patterns Working Group
schindel@ictt.com

https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:incose_north_texas_pgm_12.14.2021_v1.2.2.pdf