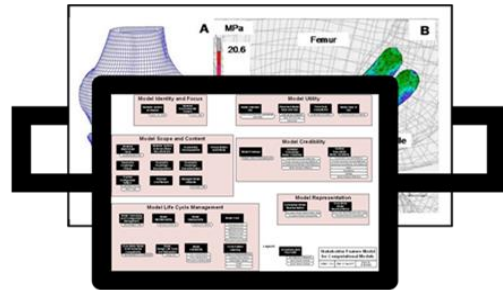
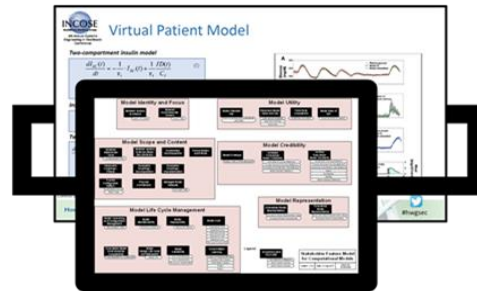


The Model Characterization Pattern (MCP)

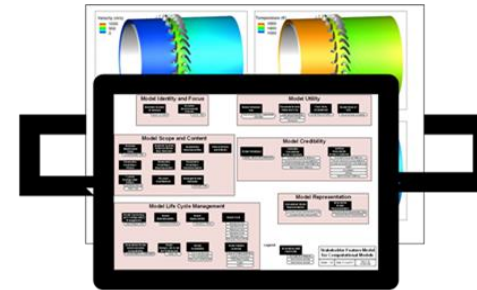
FEA Model



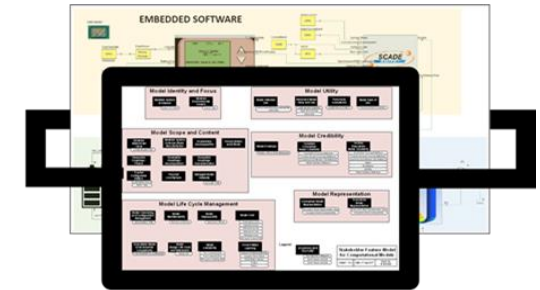
ODE Model



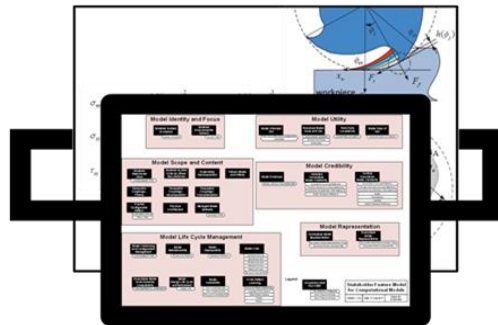
CFD Model



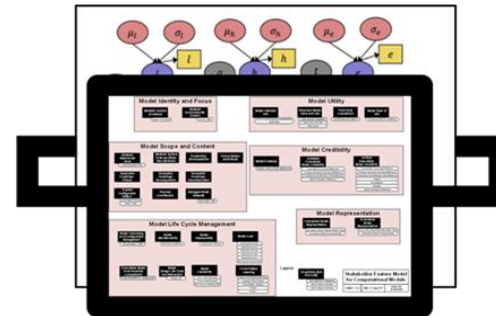
Multi-Domain
System Model



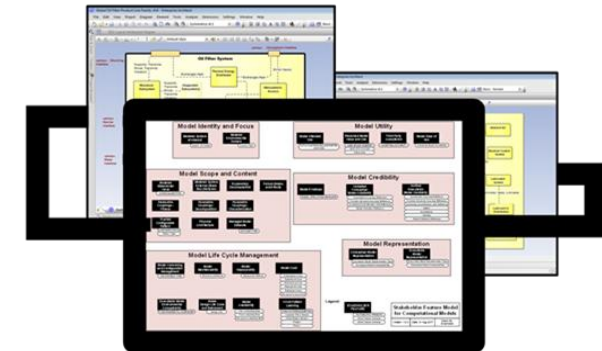
Physics-Based
PDE Model



Data-Driven Bayesian
Network Model



MBSE Model



A Universal Characterization & Labeling
S*Pattern for All Computational Models



In a Nutshell: What you can do with the MCP in Computational Model Connected Projects and Enterprises

1. Rapidly generate very systematic model requirements for new or existing models, for use in model development, verification, validation, and life cycle management.
2. More effectively plan new or improved computational models, and know when you need them, versus making use of existing model assets.
3. Lower the experience threshold needed to plan and manage computational models, including model VVUQ.
4. More effectively manage large collections of diverse computational models and related information.
5. Improve access to collections of models by exposing their characteristics to users more effectively.
6. More effectively share models across supply chains and regulatory domains.
7. Lower the cost and time necessary to obtain trusted/credible models in regulated or other domains.
8. Use or manage models that were generated by others; increase the range of others who can effectively use models that you generate; reduce the likelihood of model misuse.
9. Improve the accumulation and effective use of model-based enterprise knowledge.
10. Improve the integration of model-related work across specific engineering disciplines and overall systems engineering.
11. Increase ability to manage the integration of multiple computational models (e.g., using FMI), including their integrated VVUQ.

Contents

- Origins: A Community Effort
- An Increasingly Model-Based World
- Challenges for Model Stakeholders
- Phenomena, PIRT, Computational Models, System Models
- The Model Characterization Pattern (MCP)—an S*Pattern
- Configurable MCP Feature Groups for Models (Stakeholder Requirements)
- Configurable MCP Domain Pattern for Models
- Configurable MCP Technical Requirements for Models
- Use in Projects: Configuring the MCP for a Model or Project
- Use in Projects: What you can do with the MCP in a Project
- Infrastructure: Mapping MCP to specific company practices, tools, languages, model types, artifacts
- Patterns: Accumulating Trustable Model-Based Knowledge
- Want to Learn More? Want to Participate?
- References

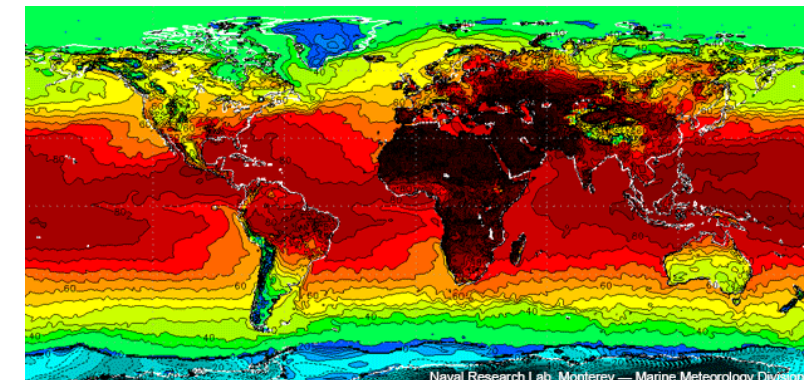
- Appendix I: MCP Features--Configurable Stakeholder Requirements for a Computational Model
- Appendix II: MCP Technical Requirements--Configurable Technical Requirements for a Computational Model
- Appendix III: Model Credibility Assessment Frameworks (CAFs)
- Appendix IV: S*Models, S*Patterns, S*Metamodel
- Appendix V: ASELCM S*Pattern, Trusted Models, Effective Group Learning
- Appendix VI: Example of a Configured MCP

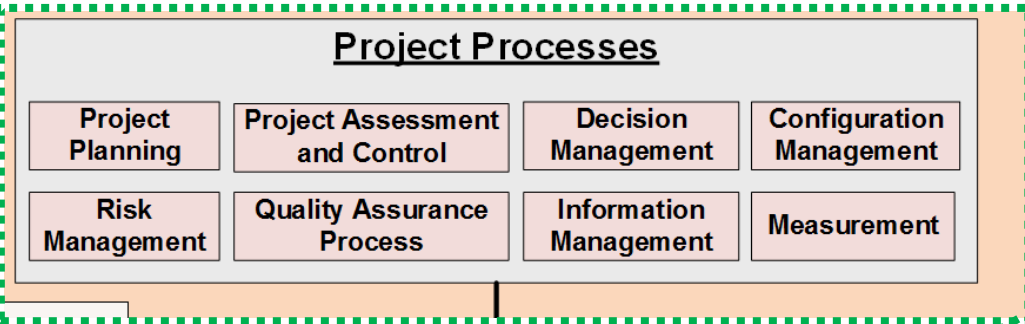
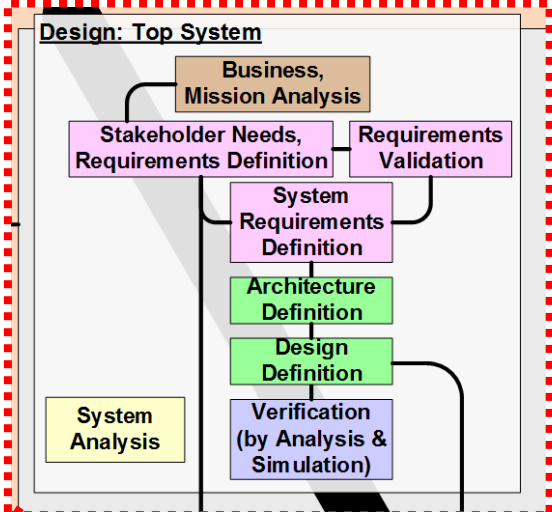
- International Council on Systems Engineering (INCOSE)--
Model-Based Patterns Working Group:
 - Model Planning & Characterization Pattern (MCP) formalized universal model wrapper, across diverse models from INCOSE and other model-oriented societies and communities;
- ASME Model V&V 50 Subcommittee--Model Life Cycle Working Group:
 - Model VVUQ guidelines and standards authoring for establishing and maintaining computational model credibility across life cycles;
- V4 Institute (V4I--an NCDMM Institute):
 - Growing related virtual model capabilities across industry communities of practice;
- ICTT System Sciences:
 - Mapping to object-oriented S*Pattern, for accessibility in all enabled OMG SysML[®] system modeling tools.



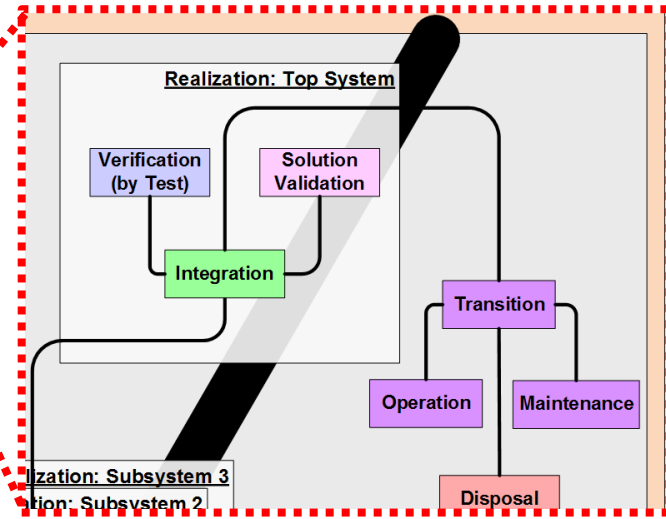
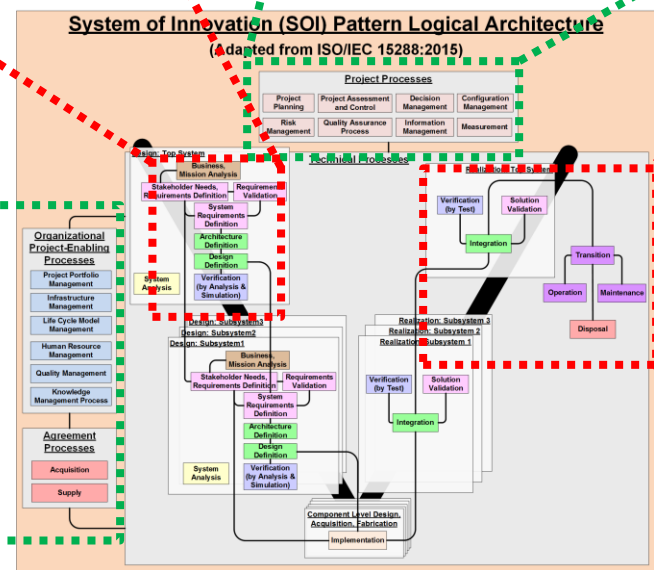
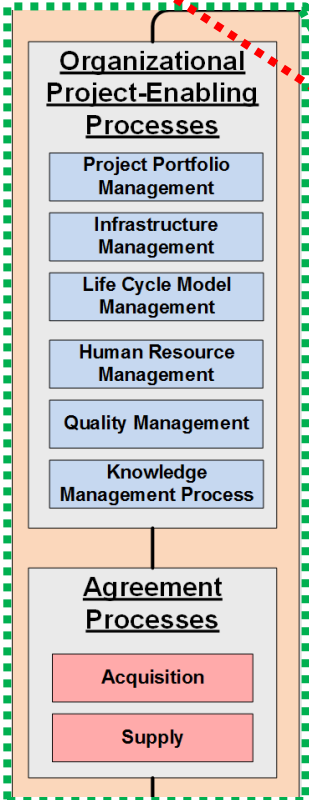
An Increasingly Model-Based World

- If we expect to use models to support critical decisions, then we are placing increased trust in models:
 - Critical financial, other business decisions
 - Human life safety
 - Societal impacts
 - Extending human capability
- Requires that we characterize the nature of that trust and manage its award, use:
 - The Validation, Verification, and Uncertainty Quantification (VVUQ) of the models themselves.





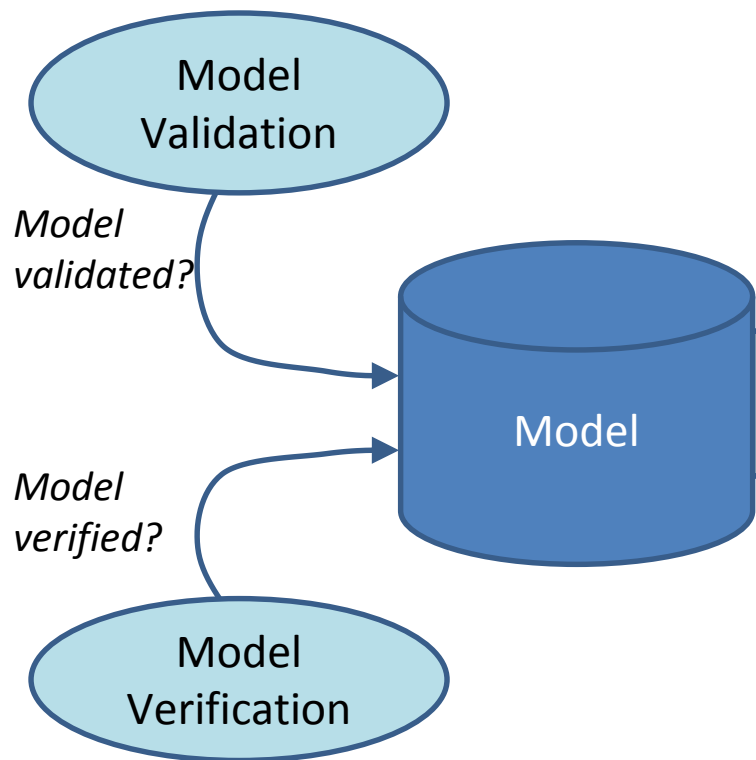
Many potential purposes for models (illustrated by ISO15288 life cycle management standard)



Life Cycle Process "Vee"

V&V of Models,
Per Emerging ASME Model V&V Standards

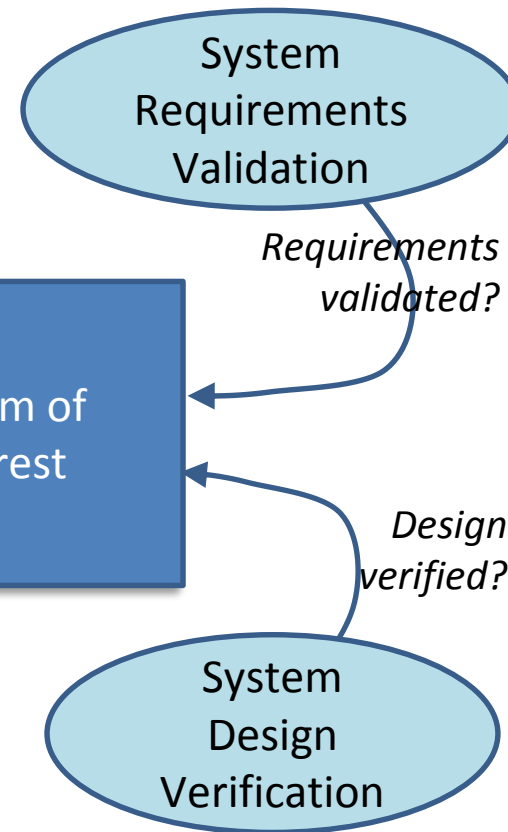
*Does the underlying Model adequately describe what it is intended to describe?
(Building the right model?)*



*Does the Model implementation adequately represent what the underlying Model described?
(Implementing the model right?)*

V&V of Systems,
Per ISO 15288 & INCOSE Handbook

*Do the System Requirements describe what stakeholders need?
(Planning the right system?)*



*Does the System Design define a solution meeting the System Requirements?
(Implementing the system right?)*

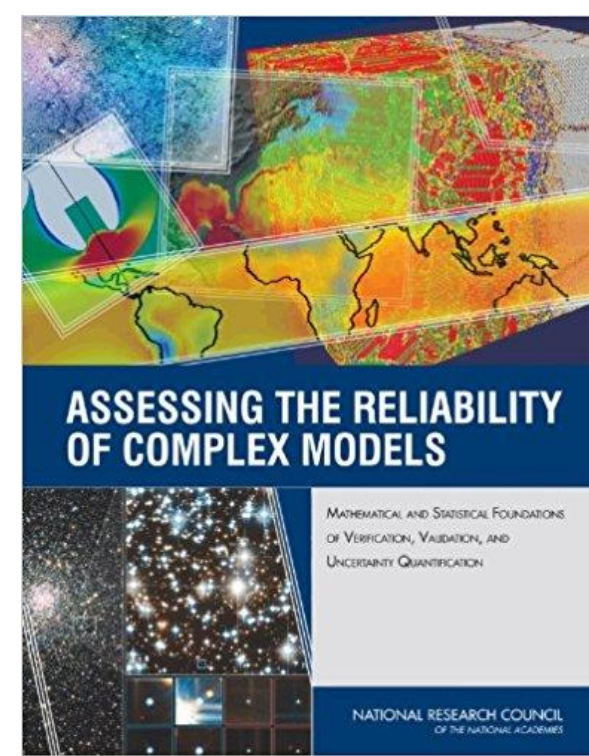
Don't forget: A model (on the left) may be used for system verification or validation (on the right!)

Quantitative Fidelity, including Uncertainty Quantification (UQ)

- There is a large body of literature on a mathematical subset of the UQ problem, at the heart of this subject.
- But, some additional systems work is needed, and has been in progress, toward the more general VVUQ framework, suitable for general systems VVUQ standards or guidelines.

General structure of uncertainty / confidence tracing:

- Do the modeled external Interactions qualitatively cover the modeled Stakeholder Features over the range of intended subject system situations of interest?
- Quantify confidence / uncertainty that the modeled Stakeholder Feature Attributes quantitatively represent the real system concerns of the subject system Stakeholders with sufficient accuracy over the range of intended situation envelopes, for intended model use.
- Quantify confidence / uncertainty that the modeled Technical Performance Attributes quantitatively represent the real system external behavior of the subject system with sufficient accuracy over the range of intended situation envelopes, for intended model use.





Related ASME activities and resources

- ASME has an active set of teams writing guidelines and standards on the Verification and Validation of Computational Models:
 - Inspired by the proliferation of computational models (FEA, CFD, Thermal, Stress/Strain, etc.)
 - It could fairly be said that this historical background means that effort was not focused on what most systems engineers would call “system-level models”
- Also conducts annual Symposium on Validation and Verification of Computational Models, in May.
- To participate in this work, in 2016 the chair of the INCOSE Patterns Working Group joined the ASME VV50 Committee on behalf of INCOSE:
 - With the idea that the framework ASME set as foundation could apply well to systems level models; and . . .
 - with a pre-existing belief that system level models are not as different from discipline-specific physics models as believed by systems community.
- Subsequently, the ASME V&V 50 Model Life Cycle WG Chair addressed the INCOSE IW2017 MBSE Workshop, on the related activity. (See References section)

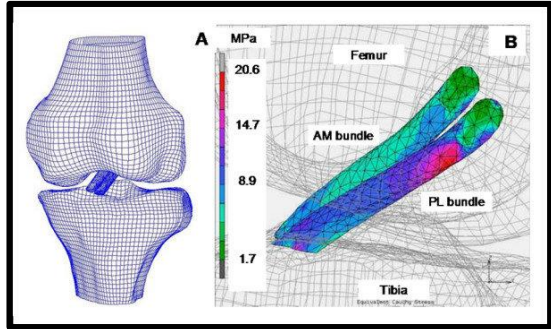
Challenges for Model Stakeholders

- The underlying basis of Model VVUQ is foundational competency in computational model practices already established within a relatively small community of experts and illustrated in related industry references, standards, texts, classes, and technical societies.
- But beyond this, model-intense enterprises are concerned with the further promulgation of virtual model practice into much larger internal communities of practice and their supply chains, domain regulators, and the extended ecosystem of the future Model-Based Economy.
- Accordingly, we must address the challenges to organizational, skill, and cultural issues that can limit future success unless addressed. These challenges include

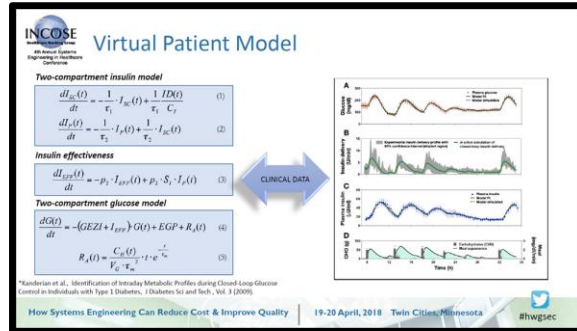
1. **Scaling up** to the population of people and volume of models and model transactions to be addressed in a world in which these will grow by orders of magnitude, overwhelming what might not otherwise be addressed by a more limited population of deeply expert model authors, model users, or model dependents--a world in which models are also being exchanged more extensively across supply chains beyond their originators.
2. **Managing models over their entire life cycle**, particularly for long-life models, including users and maintainers far from the model originator in both space (global supply chains) and time (decades).
3. **Increasing use of what has already been learned** (especially by others) about specific modeled product and system domains in past model cycles, so that what the same work and costly lesson discovery path is not repeatedly traveled at a cost in time, effort, and risk of model impact on human lives and other assets.
4. **Packaging general principles as actionable assets** moving from already described general advice, principles, and broad guidance of text books, classes, and standards, to wider and more accessible impact by packaging as structured actionable assets (data structures, tooling, actionable learning, etc.) delivering value without requiring as deep conscious expertise in detailed practice (e.g., packaging analysis of uncertainty propagation using configurable domain specific patterns, or enabling standards that are themselves models directly downloaded and immediately used in projects, shortening adoption cycles).
5. **Preparing for a more building-block world**, akin to the 1960's transformation from discrete electronics to integrated circuits, but in this case for model IP. Lifting all boats by enabling more contribution of multiple players to a world of integrated systems of models, without compromise to trust.
6. **Unifying external metadata "wrapper" (label) across all models** that will continue to be more and more diverse in their internal structure, theory, tooling, domain specifics, methodologies, styles, physics vs. data origins, and other aspects, to reduce the growth rate of challenge facing regulators and other judges of the credibility of these diverse models, appearing in a growing flood.

Diverse Virtual Models of All Types

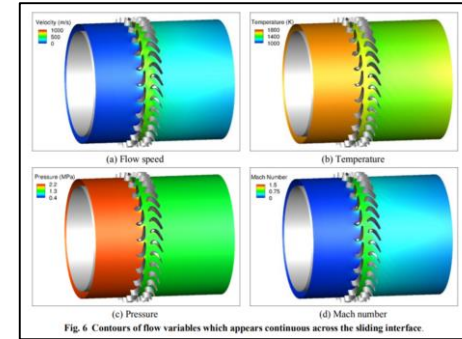
FEA Model



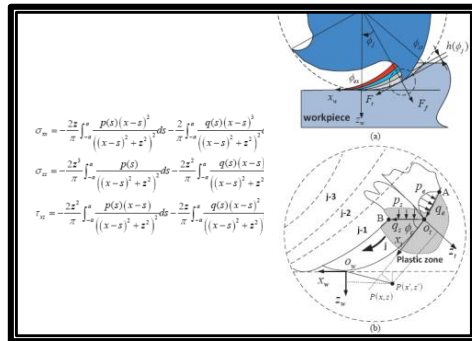
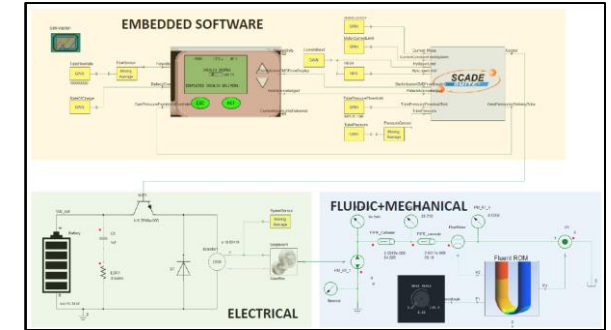
ODE Model



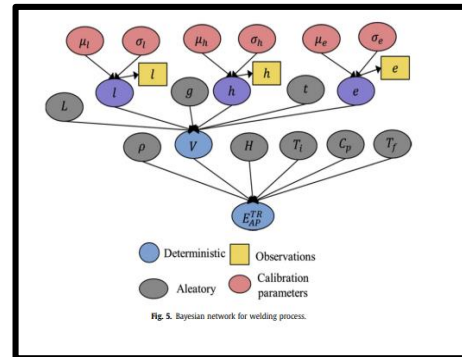
CFD Model



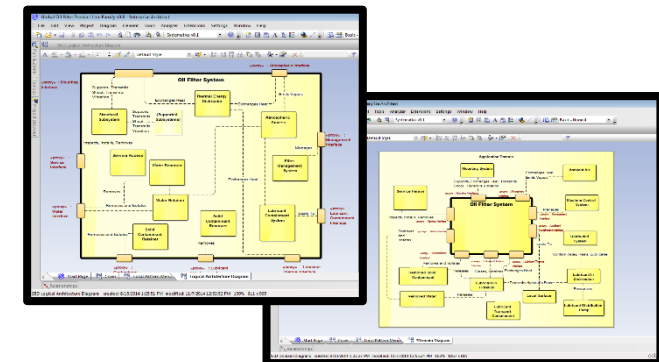
Multi-Domain System Model



Physics-Based PDE Model



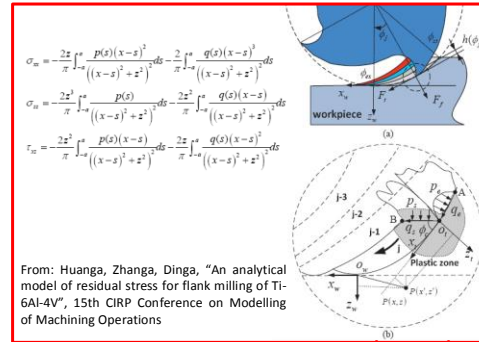
Data-Driven Bayesian Network Model



MBSE Model

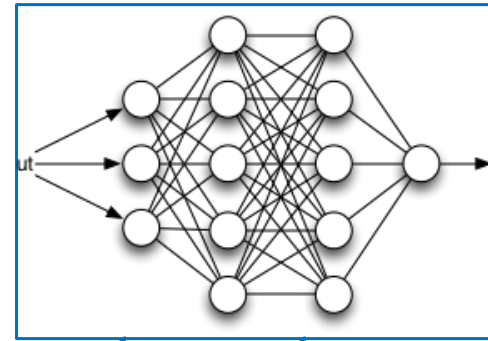
Physics-Based Model

- Predicts the external behavior of the System of Interest, visible externally to the external actors with which it interacts.
- Models internal physical interactions of the System of Interest, and how they combine to cause/explain externally visible behavior.
- Model has both external predictive value and phenomena-based internal-to-external explanatory value.
- Overall model may have high dimensionality.



Data Driven Model

- Predicts the external behavior of the System of Interest, visible to the external actors with which it interacts.
- Model intermediate quantities may not correspond to internal or external physical parameters, but combine to adequately predict external behavior, fitting it to compressed relationships.
- Model has external predictive value, but not internal explanatory value.
- Overall model may have reduced dimensionality.



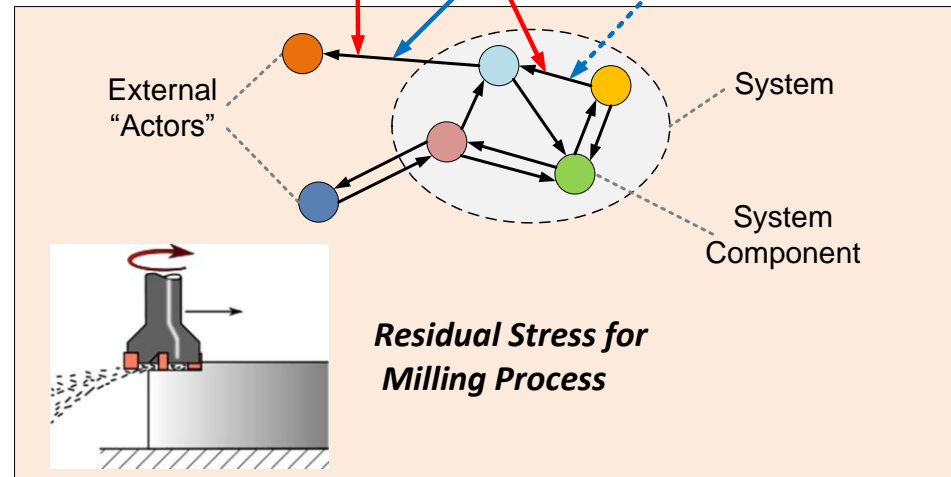
- Physical scientists and phenomena models from their disciplines can apply here.
- The hard sciences physical laws, and how they can be used to explain the externally visible behavior of the system of interest.

predicts,
explains

predicts

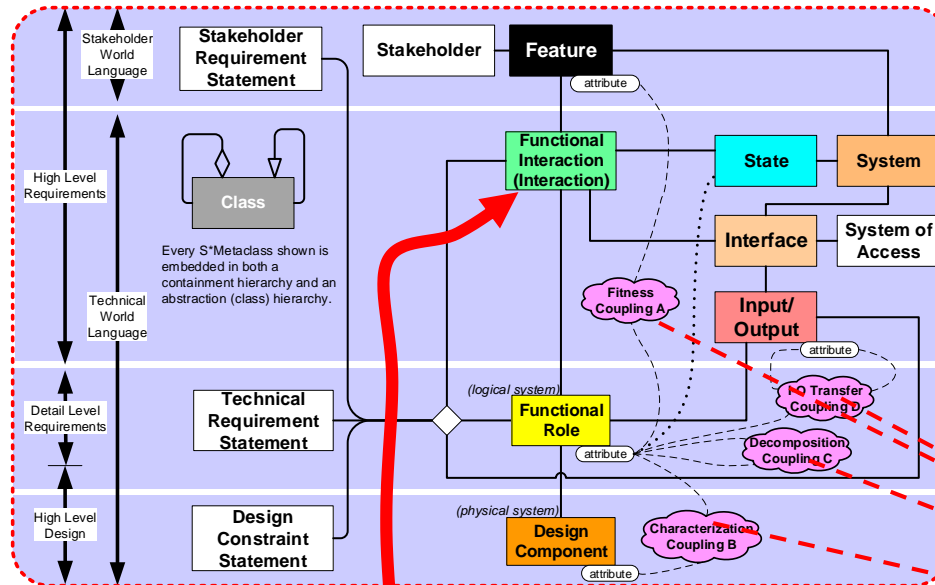
optional

- Data scientists and their math/IT tools can apply here (data mining, pattern extraction, cognitive AI tooling).
- Tools and methods for discovery / extraction of recurring patterns of external behavior.



Real Target System Being Modeled

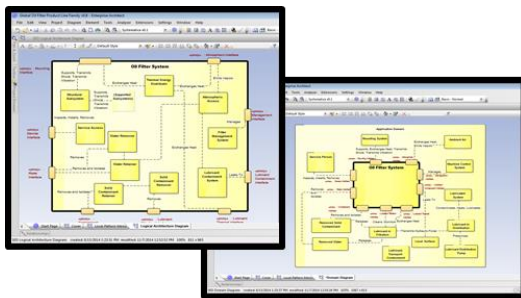
Phenomena Occur in Presence of Interactions



S*Metamodel informal summary pedagogical diagram
(formal S*Metamodel includes additional details.)
Appendix III

Attribute Coupling

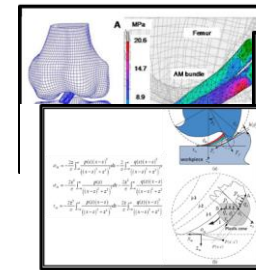
System Model



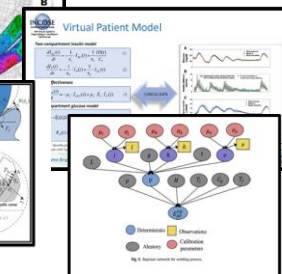
MBSE Model

Computational Model

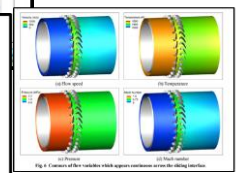
FEA Model



ODE Model

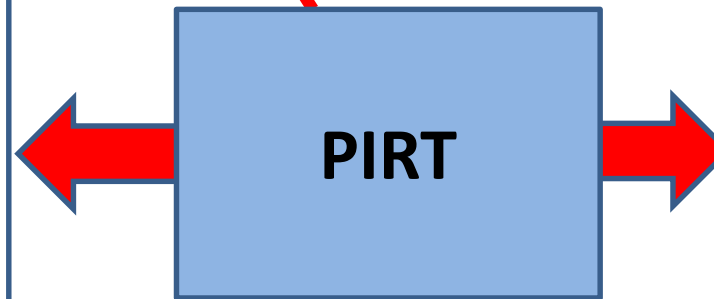


CFD Model



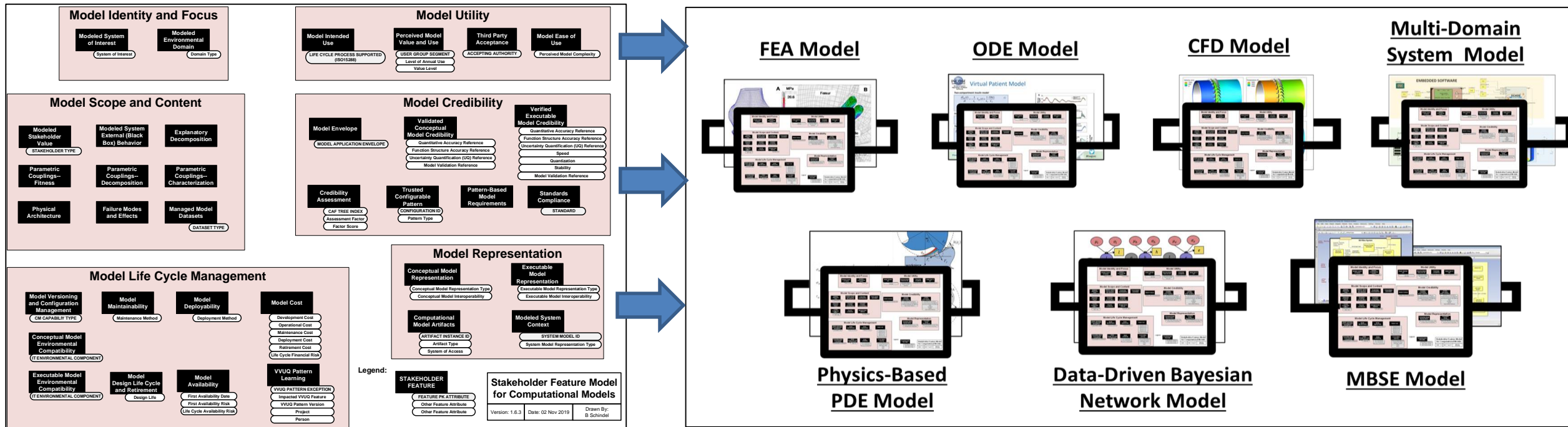
Physics-Based PDE Model

Data-Driven Bayesian Network Model



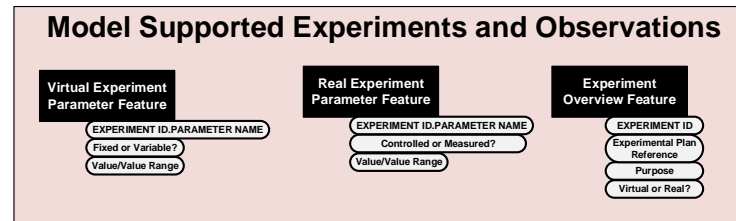
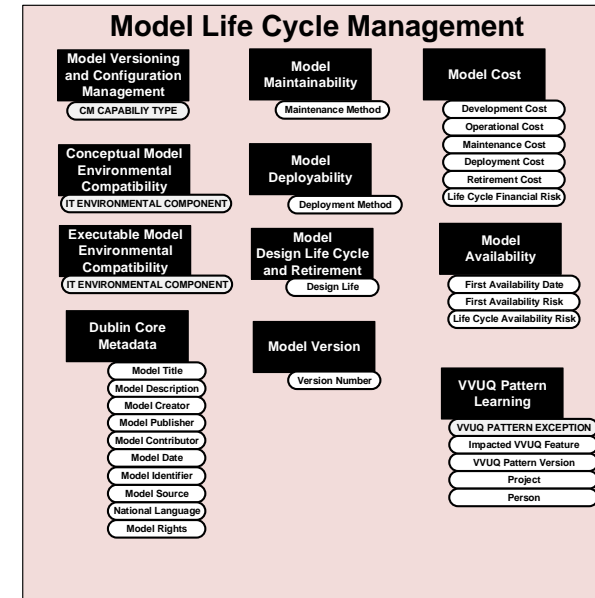
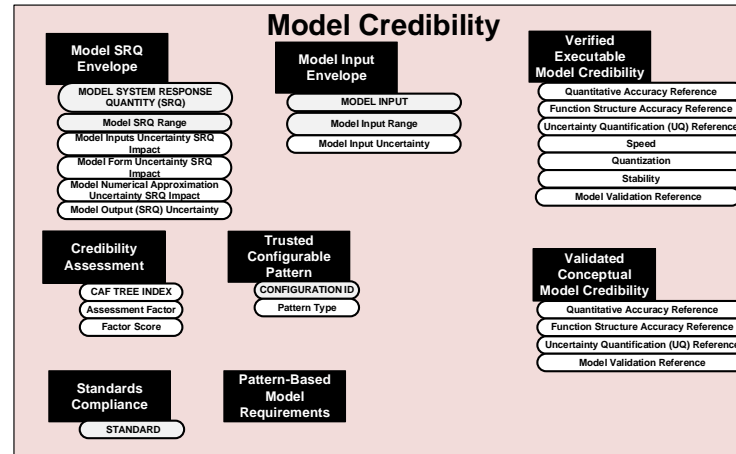
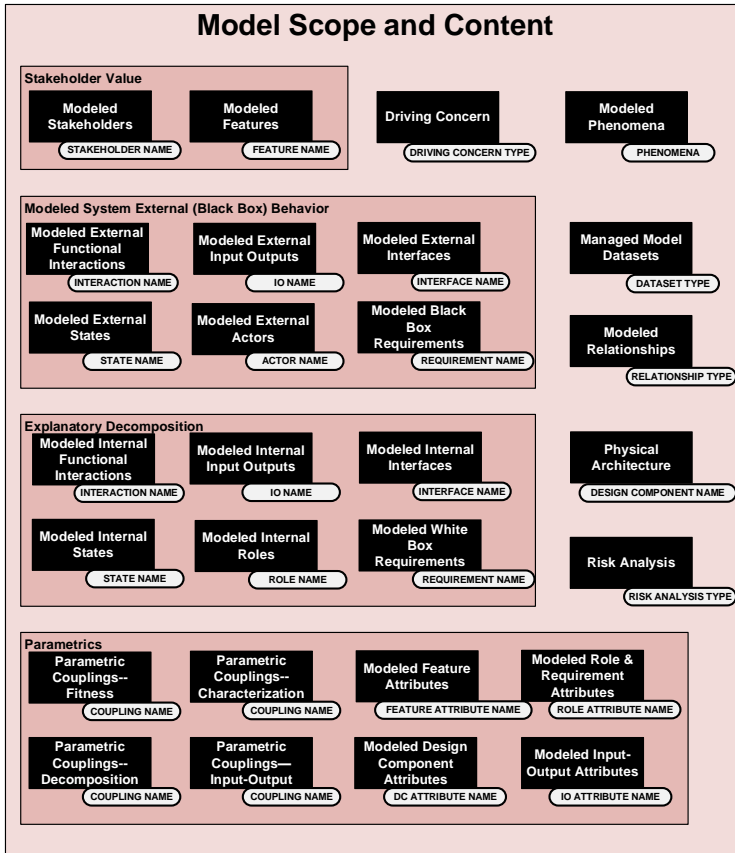
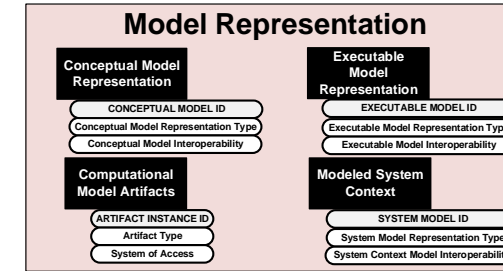
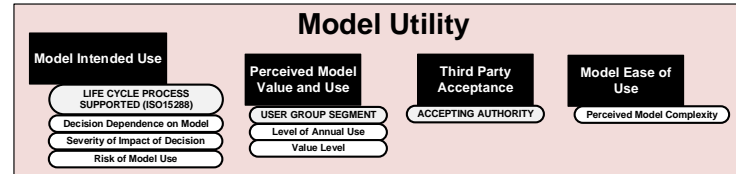
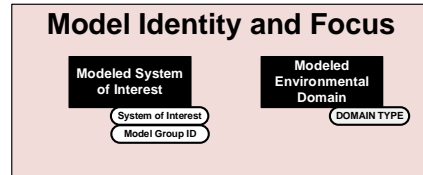
The Model Characterization Pattern (MCP)—an S* Pattern

- A universal “wrapper” across all computational model types.
- Provides a common characterization for all models.
- Key to managing the model’s entire life cycle, including but not limited to Model VVUQ.



Configurable MCP Feature Groups for Models (Computational Model's Stakeholder Requirements)

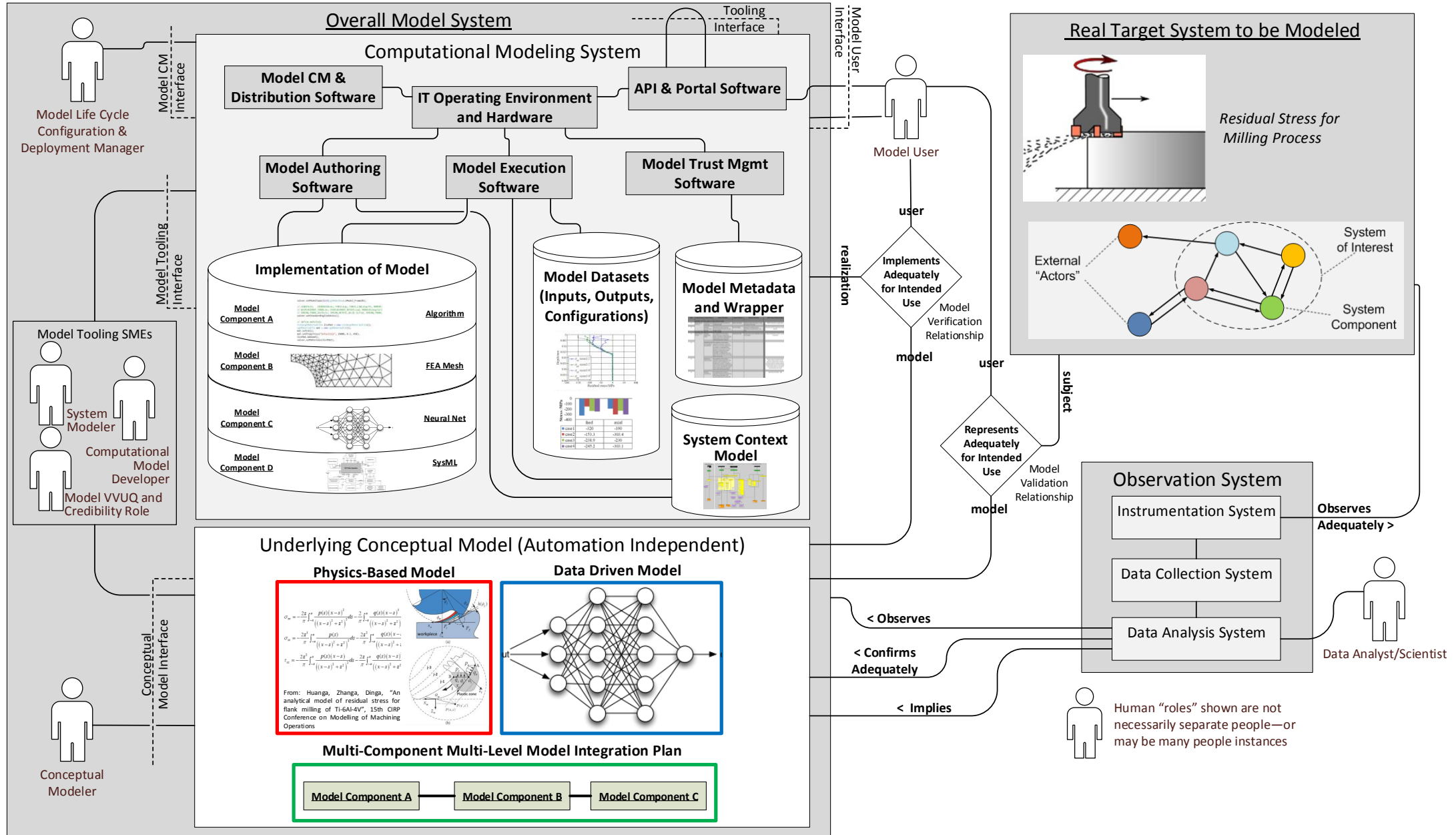
(See Appendix I for definitions of each Feature shown.)



Legend:

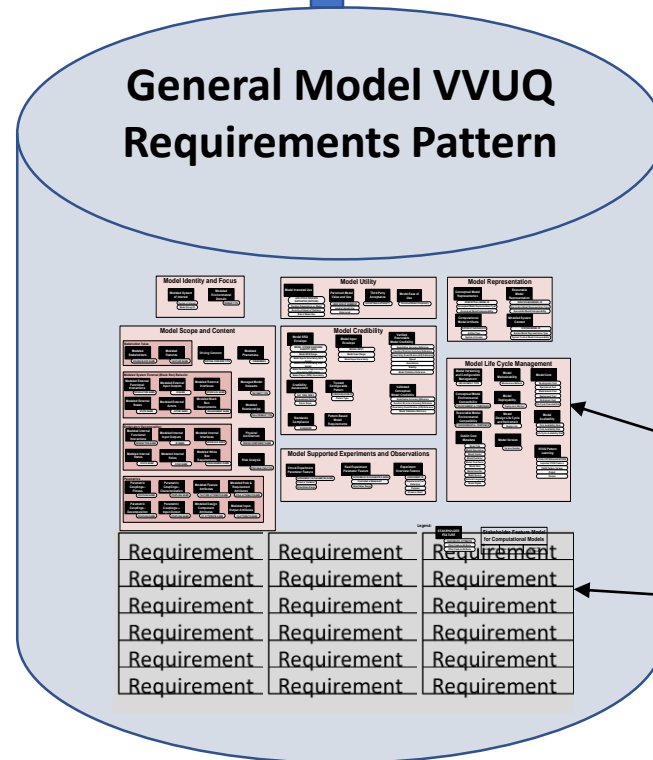


Configurable MCP Domain Pattern for Model Systems



General Pattern: Not all parts shown apply in all configuration instances

Configurable MCP Technical Requirements for Models

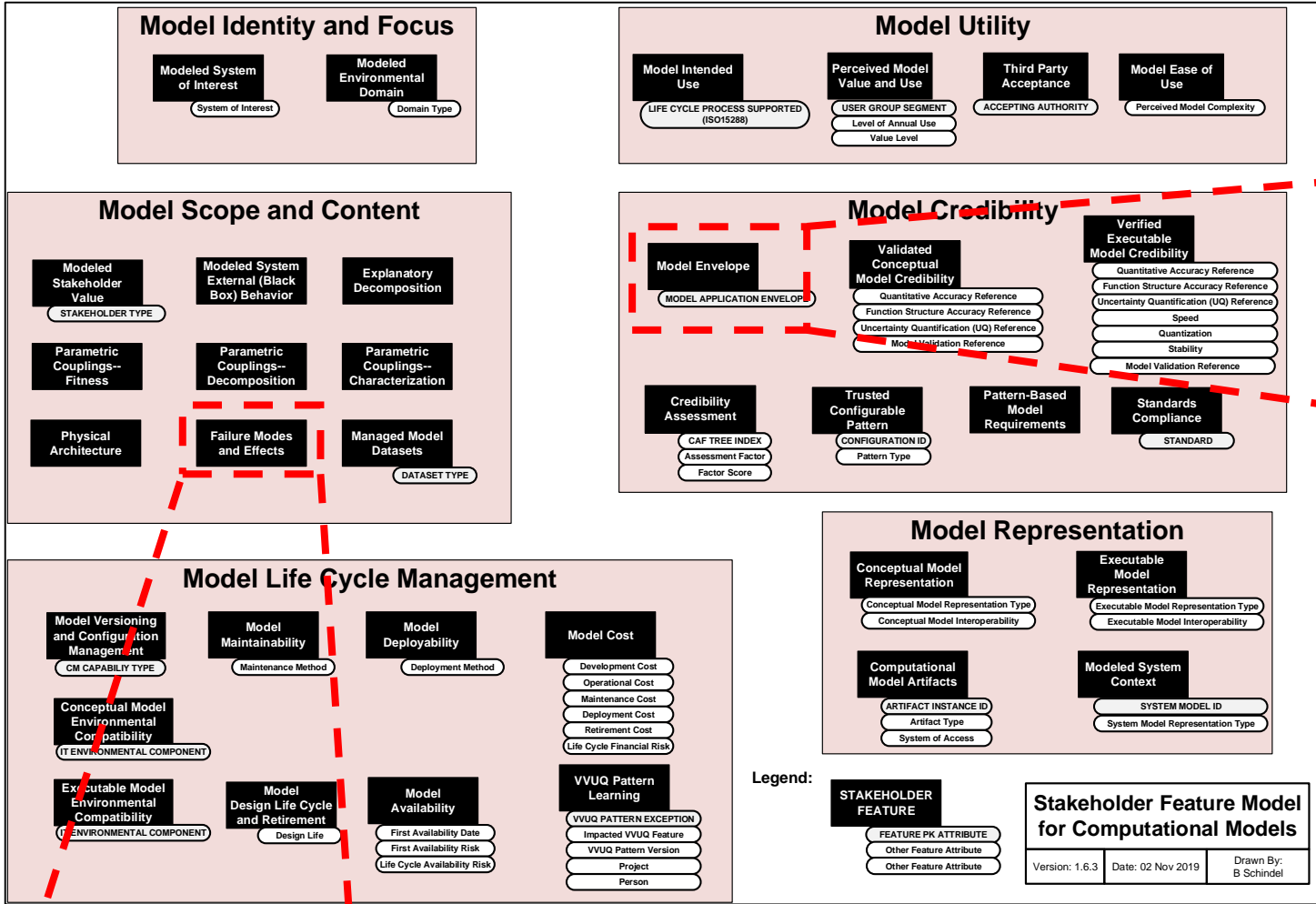


Any system modeling tool enabled with the S*Metamodel can be used for this process.

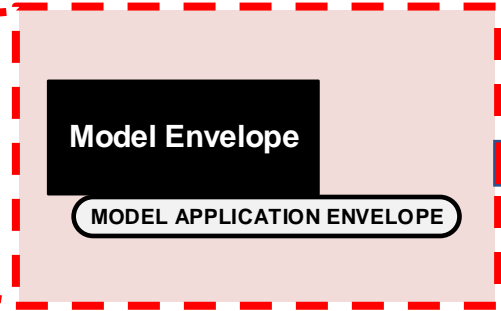
60 Model Stakeholder Requirements

~75 Model Technical Requirements

See Appendix II for each of the Technical Requirements)



Model Feature, from Configurable VVUQ Pattern



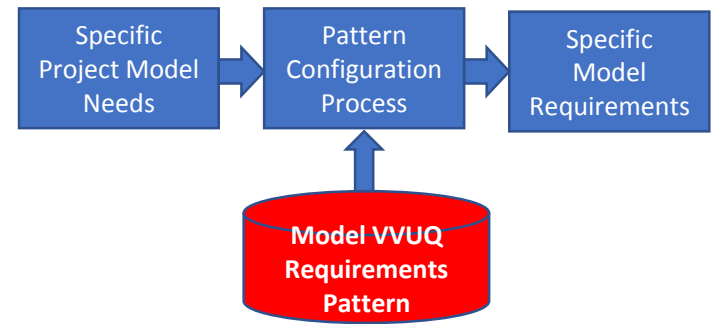
Model Requirement, from Configurable VVUQ Pattern

3.1.2 Modeled Envelope, External Technical:
 “The model shall represent the system of interest over a specified (discrete or continuous) range or envelope of technical external environment interaction configurations.”

Failure Modes and Effects
Model Feature, from Configurable VVUQ Pattern

Model Requirement, from Configurable VVUQ Pattern

2.6.1 Failure Mode: “The model shall include identification of component failure modes, as to underlying state leading to predicted failure.”



Use in Projects: Configuring the MCP for a Model or Project

Baseline Workbook 7.0.9 Model VVUQ Pattern 2.1.48 [Compatibility Mode] - Excel

File Home Insert Page Layout Formulas Data Review View Developer Help QuickBooks Tell me what you want to do

Clipboard Font Alignment Number Styles Cells Editing

H9 =OFFSET(ShortFeatures,ROW(B3)-1,0,1,1)

	D	E	H	K	N	O	P	Q	R	S	T	U	V
4			No. Populated Features: 33		BUTTON 1: Generate Feature Attribute Form and Clear Its Attribute Values		BUTTON 2: Refresh Feature Attribute Form and Retain Its Attribute Values						
	Enter information in YELLOW cells only.												
	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
5	0	YES	Model Envelope	--									
6	0	YES	Pattern-Based Model Requirements	--	In Service Application	Production							
7	0	YES	Standards Compliance	Standard	ASME VV20	ASME VV40							
8	0	YES	Trusted Configurable Pattern	--									
9	0	YES	Validated Conceptual Model Credibility	--									

Stkhdr Config 1. Feature Population 2. Feat Att Values Backup Feat Att Values Interact ...

4:51 PM 4/16/2019

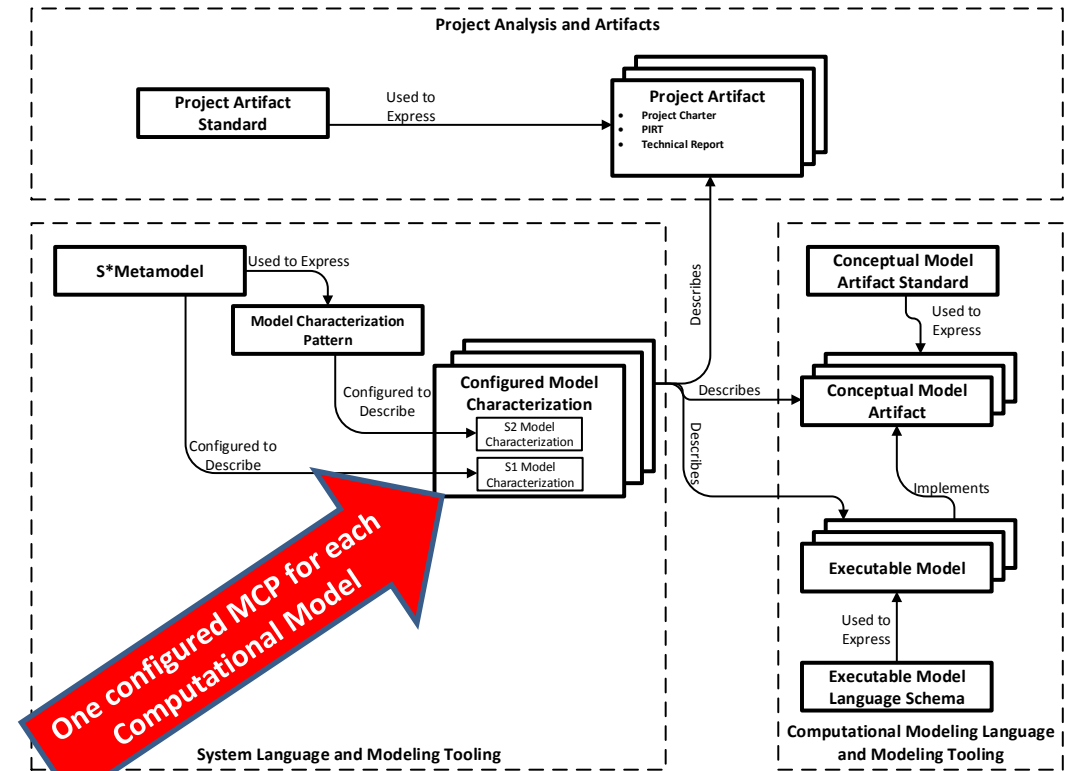
What you can do with the MCP in Computational Model Connected Projects and Enterprises

1. Rapidly generate very systematic model requirements for new or existing models, for use in model development, verification, validation, and life cycle management.
2. More effectively plan new or improved computational models, and know when you need them, versus making use of existing model assets.
3. Lower the experience threshold needed to plan and manage computational models, including model VVUQ.
4. More effectively manage large collections of diverse computational models and related information.
5. Improve access to collections of models by exposing their characteristics to users more effectively.
6. More effectively share models across supply chains and regulatory domains.
7. Lower the cost and time necessary to obtain trusted/credible models in regulated or other domains.
8. Use or manage models that were generated by others; increase the range of others who can effectively use models that you generate; reduce the likelihood of model misuse.
9. Improve the accumulation and effective use of model-based enterprise knowledge.
10. Improve the integration of model-related work across specific engineering disciplines and overall systems engineering.
11. Increase ability to manage the integration of multiple computational models (e.g., using FMI), including their integrated VVUQ.

Infrastructure: Mapping the Model Characterization S*Pattern to specific enterprise practices, tools, languages, model types

- A configured MCP for some computational model will typically refer to diverse information elements and artifacts of the model life cycle:

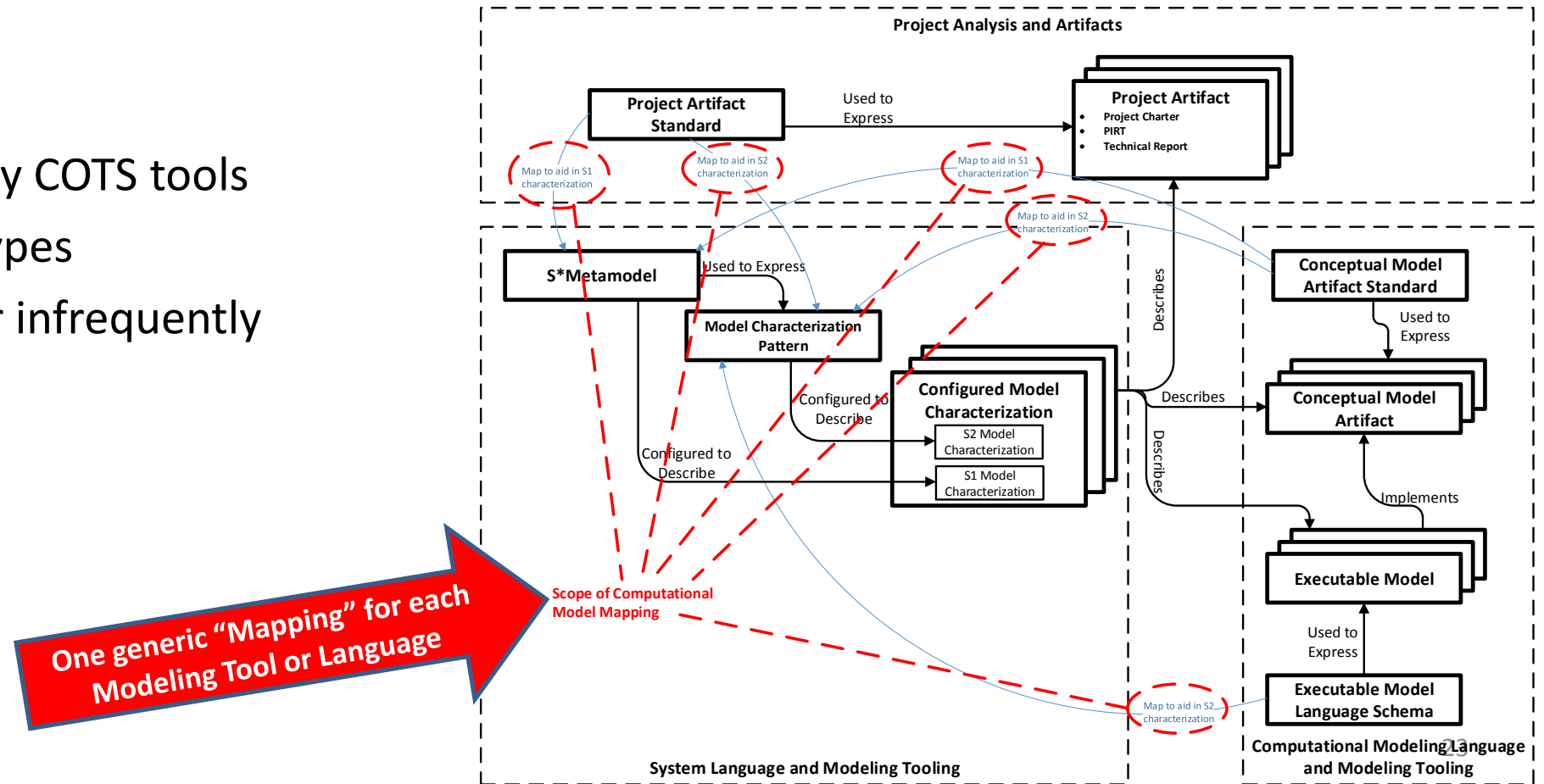
- Project charter
- Model Stakeholder and Technical Requirements
- The conceptual model and executable model
- Model authoring tools
- Model execution IT systems environment
- Modeler’s Notebook
- Phenomenon Identification and Ranking Table (PIRT)
- Reference documentation on the System of Interest
- Modeling Project Report
- VVUQ Report
- Procedures, standards, or other publications
- Other aspects of the Model System Domain Diagram earlier in these slides



- These are often physically separated and also may be individually diverse from one model to another—but they are all part of a complex interacting “System 2”:
- A configured Model Characterization Pattern helps “pull together” those elements.

Infrastructure: Mapping the Model Characterization S*Pattern to specific enterprise practices, tools, languages, model types

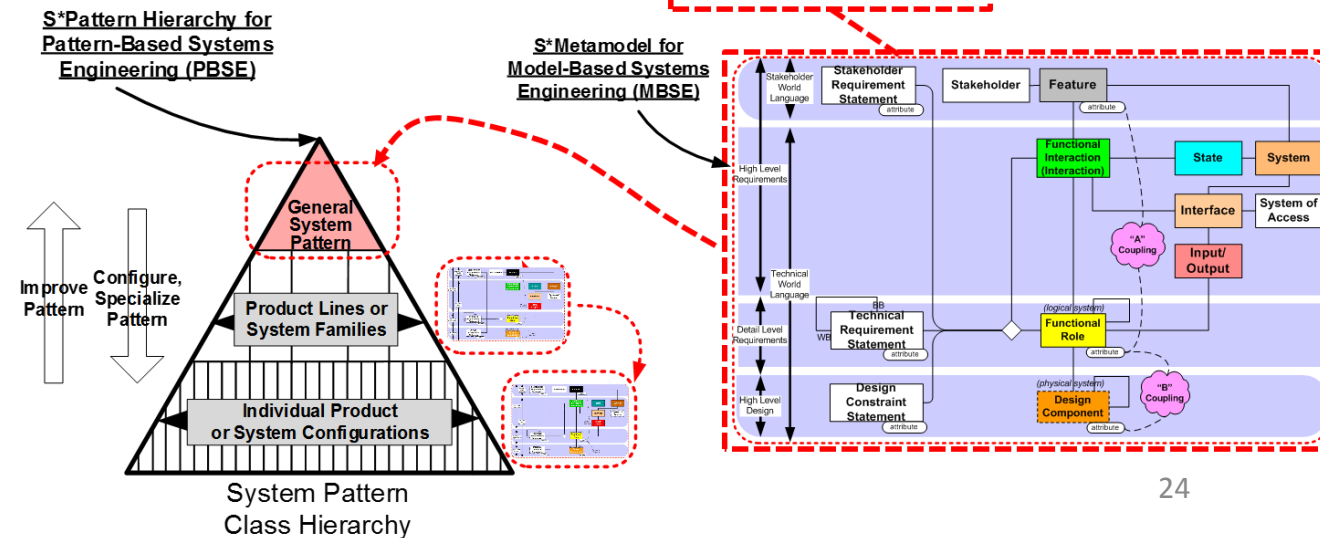
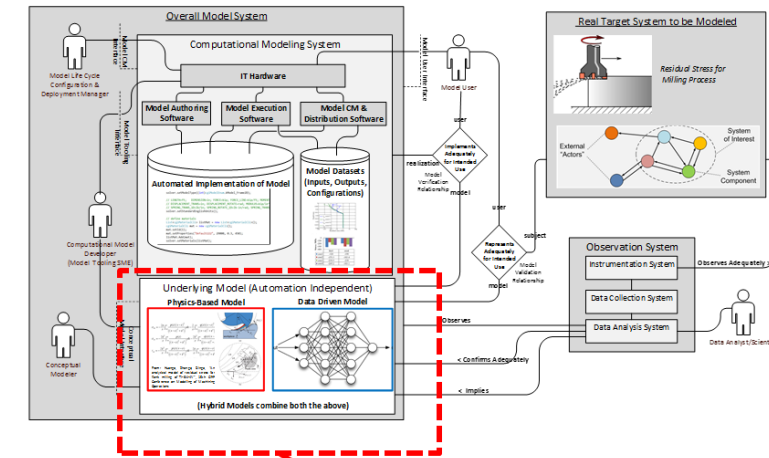
- The overall process or environment owner may also provide a standard “mapping” from the general Model Characterization Pattern to certain local targets, such as:
 - One time for each modeling tool or modeling language:
 - FEA
 - Neural Net
 - Specific third party COTS tools
 - Specific artifact types
 - Prepared one time or infrequently



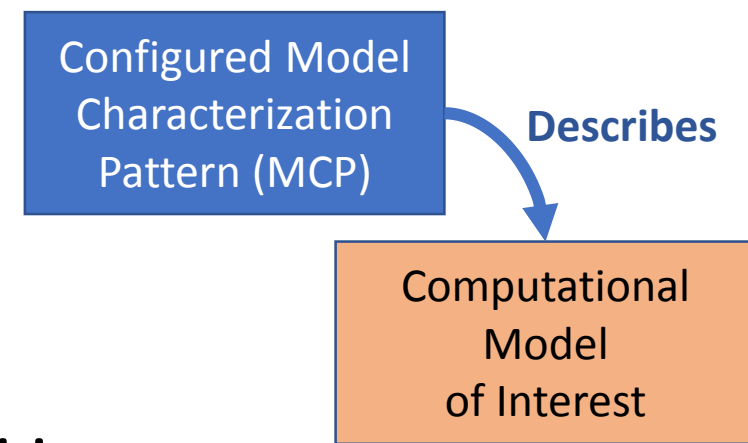
System of Interest Patterns:

Accumulating Trustable Model-Based Knowledge

- The imperative of managing model trust means Model VVUQ is not an option.
- Investment in trustable models and their VVUQ increases the need to make use of the leverage of model patterns.
- Same as the history of physical sciences.

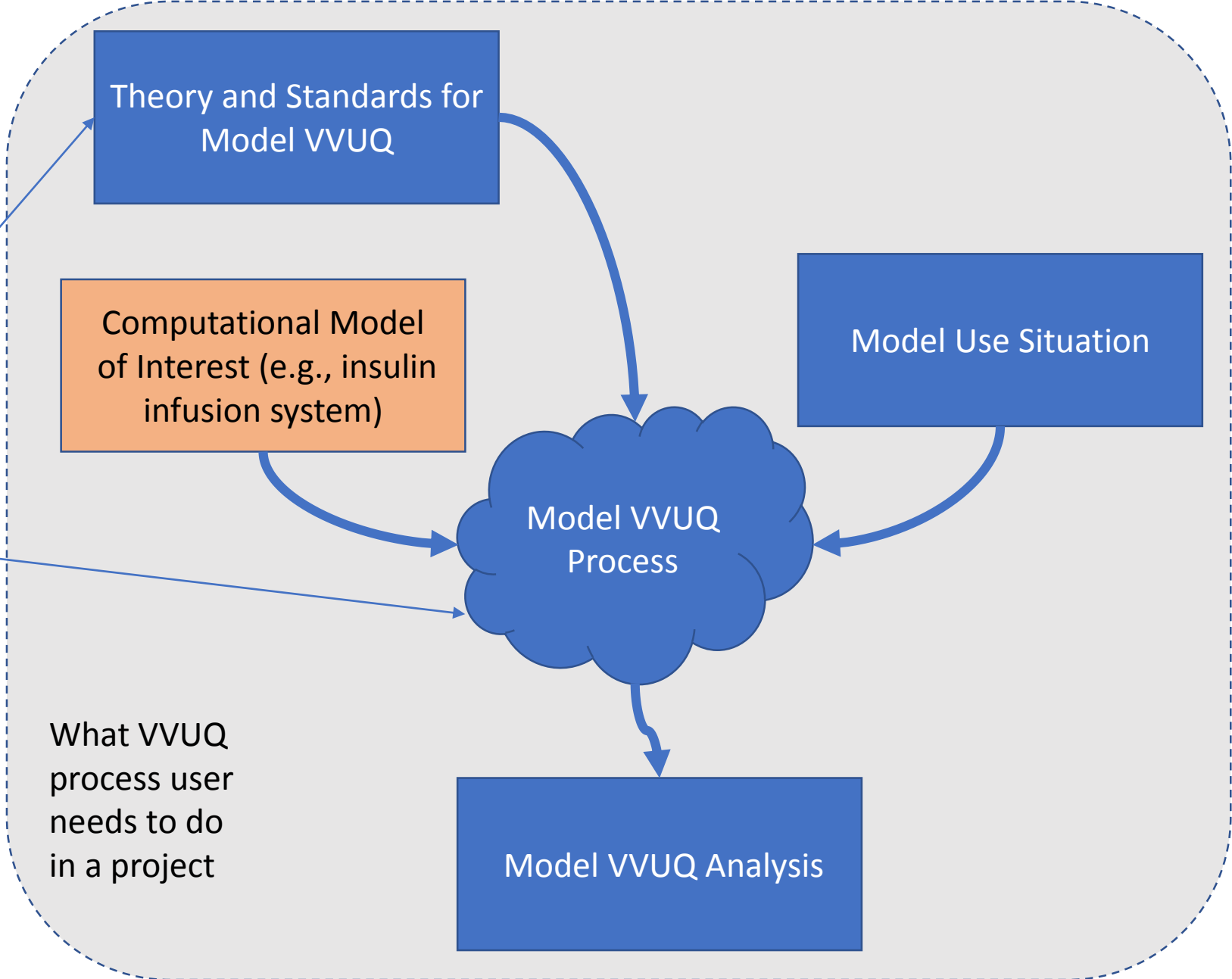


Leveraging Model VVUQ Theory to Leverage the Economics of Trusted Model-Based Patterns



- “Models of computational models” may sound odd, so . . .
- Why are we creating S^* Models of computational models of interest?
 1. To package decades of rich and valuable historical progress in theory of, and standards for, scientific model verification, validation, and uncertainty quantification
 - Into forms accessible by larger communities of less expert users;
 - Without diminishing, but instead gaining, VVUQ rigor, clarity, and standards alignment;
 2. Leveraging not only that theory but also hard-obtained learning about domain-specific models, into a form suitable for shared group learning as domain learning advances;
 3. Across otherwise diverse and rapidly changing virtual models, improve sharing ability of communities of enterprises, regulators, standards groups, supply chains, trade groups, lowering innovation friction while protecting critical IP;
 4. Improve ability to integrate families of diverse models across a single system or SoS;
 5. Enhance shared understanding of model planning, justification, documentation, migration, enhancement, and other model life cycle issues.

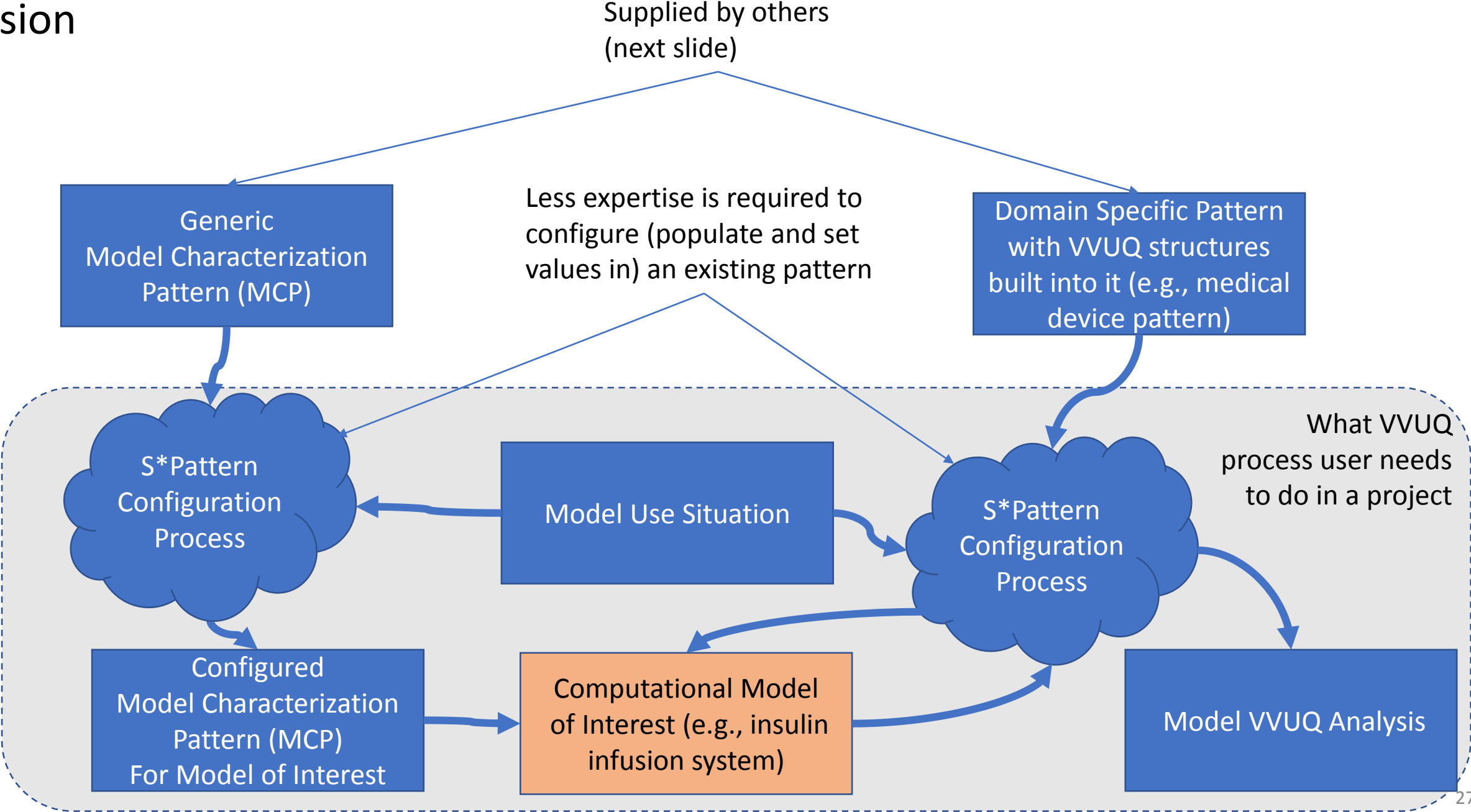
Current Practice

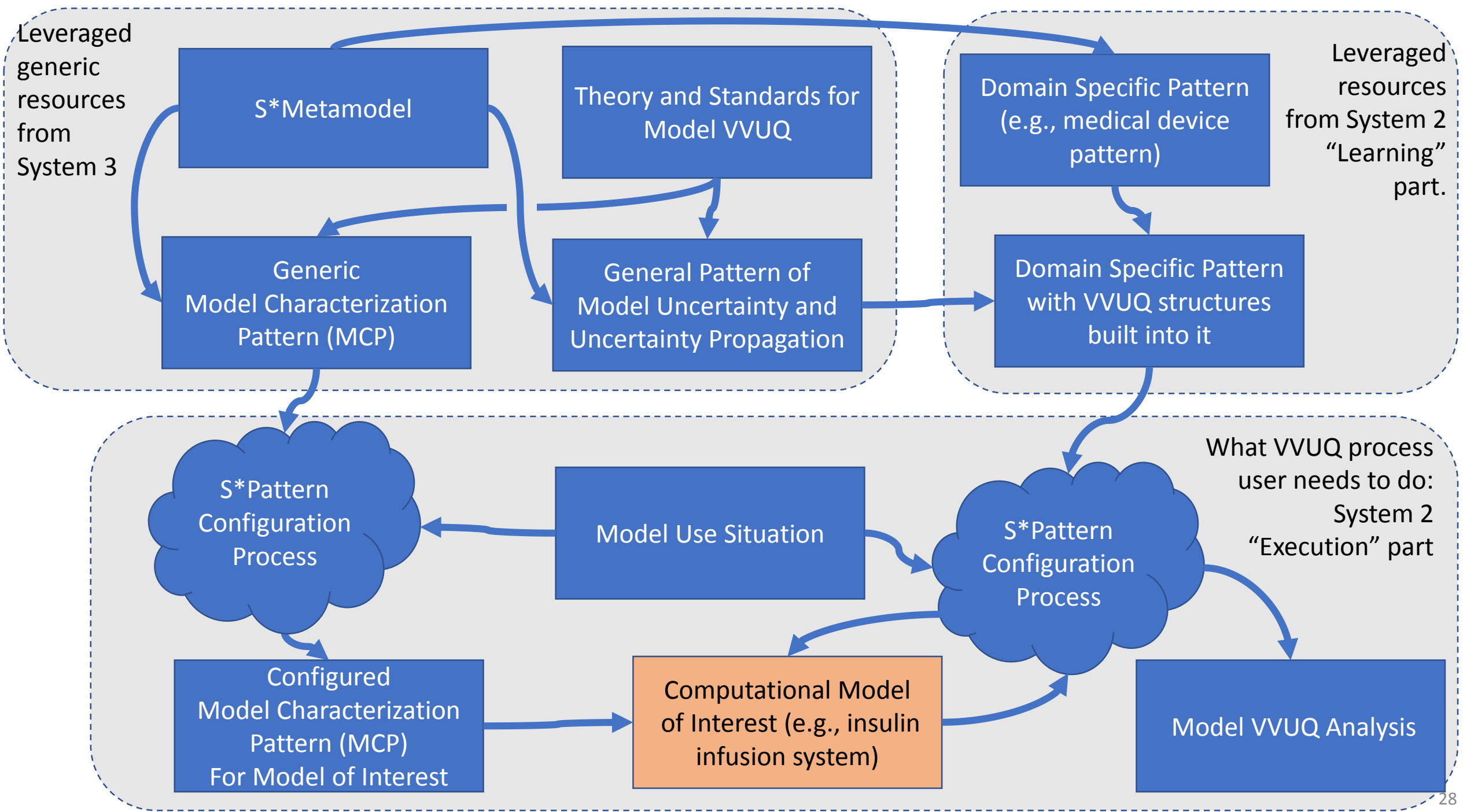


Expertise in these two areas may typically be limited. Practitioner knows more about Model Use Situation and Computational Model of Interest.

What VVUQ process user needs to do in a project

Vision





Want to Learn More? Participate?

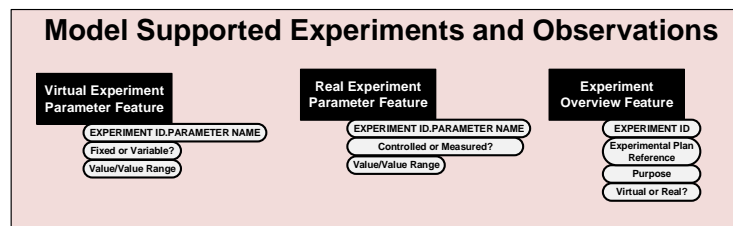
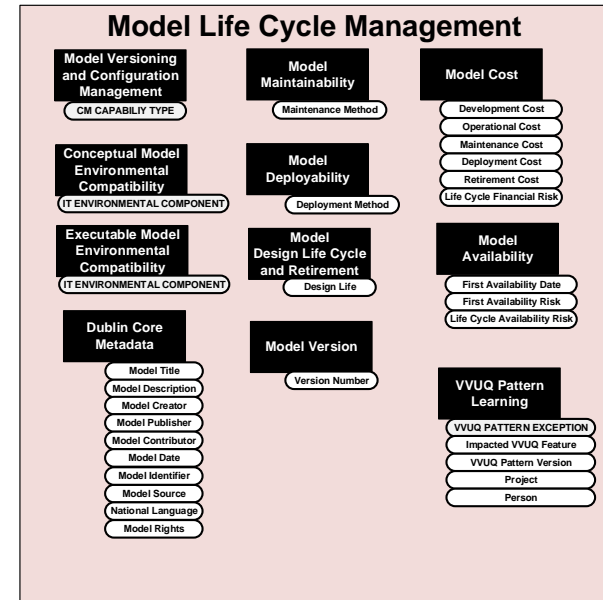
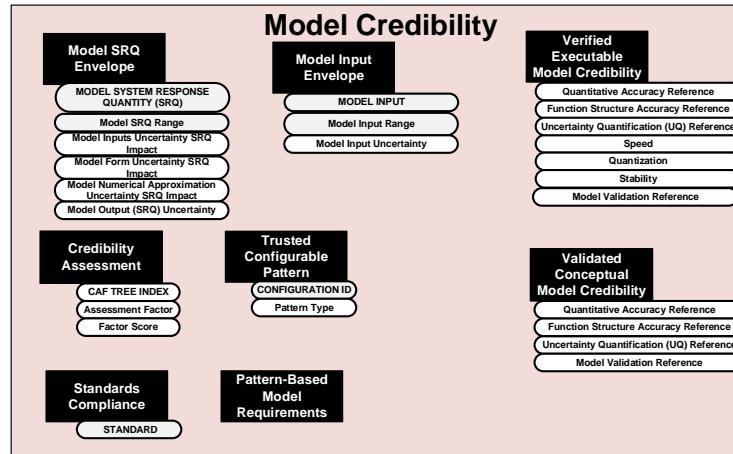
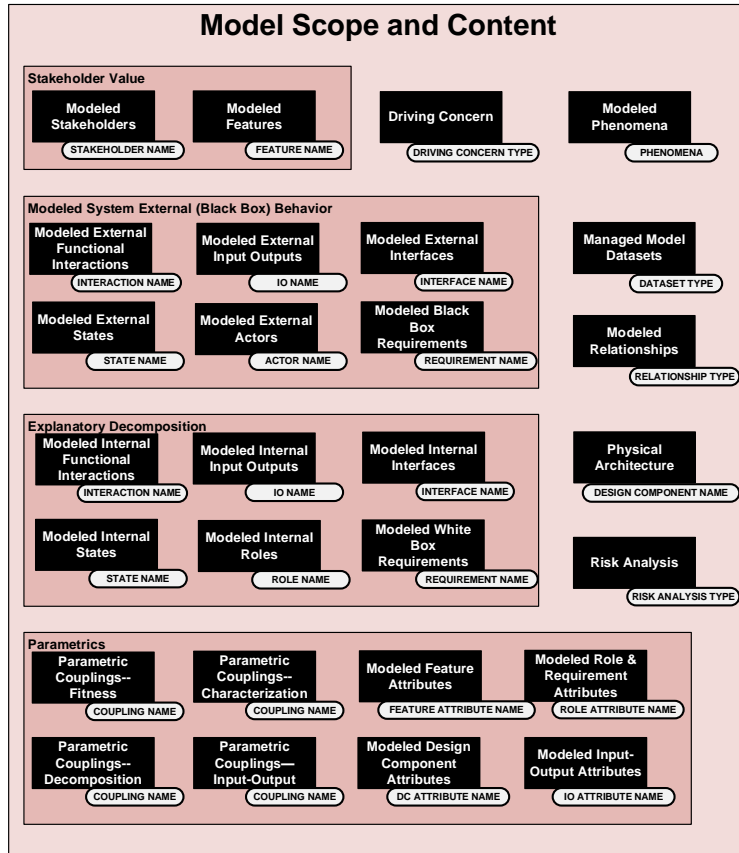
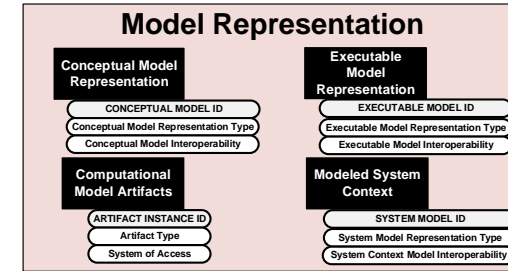
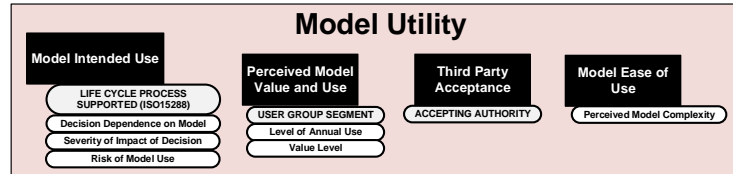
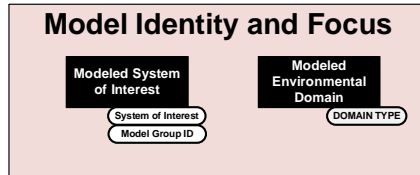
- For more information on:
 - ASME VV50 Subcommittee on Computational Model Life Cycle
 - INCOSE Model-Based Patterns Working Group
 - V4 Institute

Consult the References section

References

1. National Research Council, *Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification*, Washington, DC: The National Academies Press, 2012.
<https://doi.org/10.17226/13395> .
2. Hightower, Joseph, “Establishing Model Credibility Using Verification and Validation”, INCOSE MBSE Workshop, IW2017, Los Angeles, January, 2017.
[http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose mbse iw 2017:models and uncertainty in decision making rev a.pptx](http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose_mbse_iw_2017:models_and_uncertainty_in_decision_making_rev_a.pptx)
3. INCOSE MBSE Patterns Working Group, “MBSE Methodology Summary: Pattern-Based Systems Engineering (PBSE), Based On S*MBSE Models”, V1.5.5A, retrieve from: <http://www.omgwiki.org/MBSE/doku.php?id=mbse:pbse>
4. V4 Institute, “Trusted Models, Collaborative Learning, Accelerated Capability”, *Proc. of INCOSE 2018 Great Lakes Symposium on Systems Engineering: V4I Tutorial*, May 17, 2018, Indianapolis. Retrieve from:
http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:v4i_workshop_v1.5.2.pdf
5. Schindel, W., “What Is the Smallest Model of a System?”, *Proc. of the INCOSE 2011 International Symposium, International Council on Systems Engineering* (2011). Retrieve from:
https://www.researchgate.net/publication/283141249_What_Is_the_Smallest_Model_of_a_System
6. _____, “Got Phenomena? Science-Based Disciplines for Emerging Systems Challenges”, *Proc. of the INCOSE 2016 International Symposium*, Edinburgh, UK, 2016. Retrieve from:
http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:got_phenomena_v1.6.4.pdf

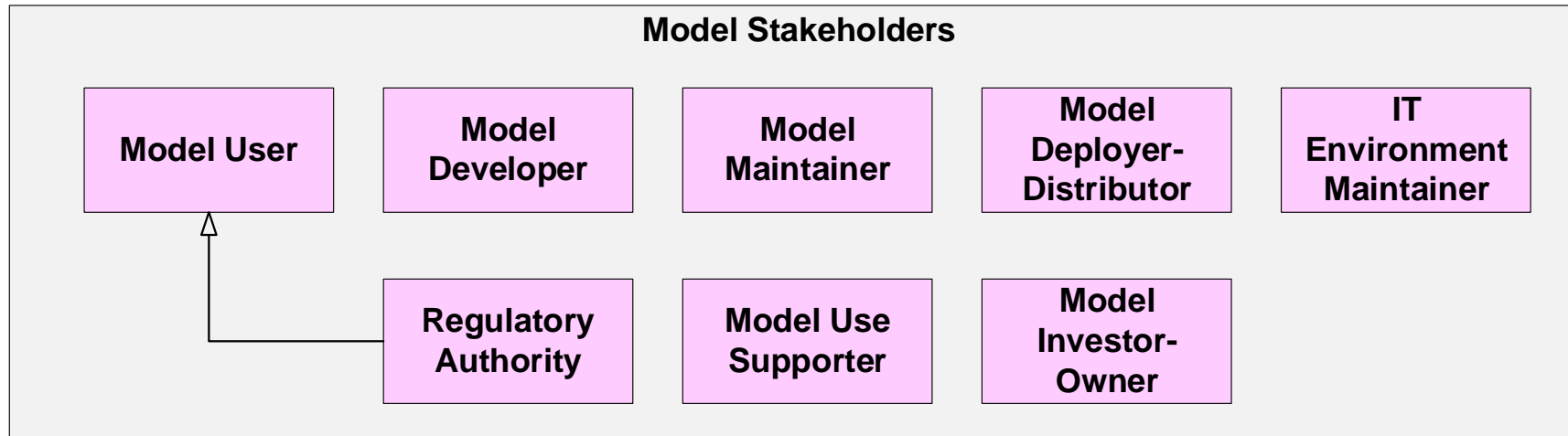
Appendix I: MCP Features--Configurable Stakeholder Requirements for a Computational Model



Legend:



Stakeholders for Models



Model Stakeholder Type	Definition
Model User	A person, group, or organization that directly uses a model for its agreed upon purpose. May include technical specialists, non-technical decision-makers, customers, supply chain members, regulatory authorities, or others.
Model Developer	A person who initially creates a model, from conceptualization through implementation, validation, and verification, including any related model documentation. Such a person may or may not be the same as one who subsequently maintains the model.
Model Maintainer	A person who maintains and updates a model after its initial development. In effect, the model maintainer is a model developer after the initial release of a model.
Model Deployer-Distributor	A person or organization that distributes and deploys a model into its intended usage environment, including transport and installation, through readiness for use.
Model Use Supporter	A person who supports or assists a Model User in applying a model for its intended use. This may include answering questions, providing advice, addressing problems, or other forms of support.
Regulatory Authority	An organization that is responsible for generating or enforcing regulations governing a domain.
Model Investor-Owner	A person or organization that invests in a model, whether through development, purchase, licenses, or otherwise, expecting a benefit from that investment.
IT Environment Maintainer	A person or organization that maintains the IT environment utilized by a computational model.

Computational Model Feature Groups: Configurable for Specific Models

Model Identity and Focus

Identifies the main subject or focus of the model.

Model Utility

Describes the intended use, user, utility, and value of the model.

Model Representation

Describes the representation used by the model, along with related artifacts.

Model Scope and Content

Describes the scope of content of the model.

Model Credibility

Describes the credibility of the model.

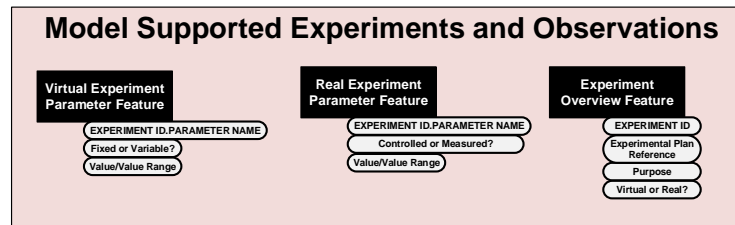
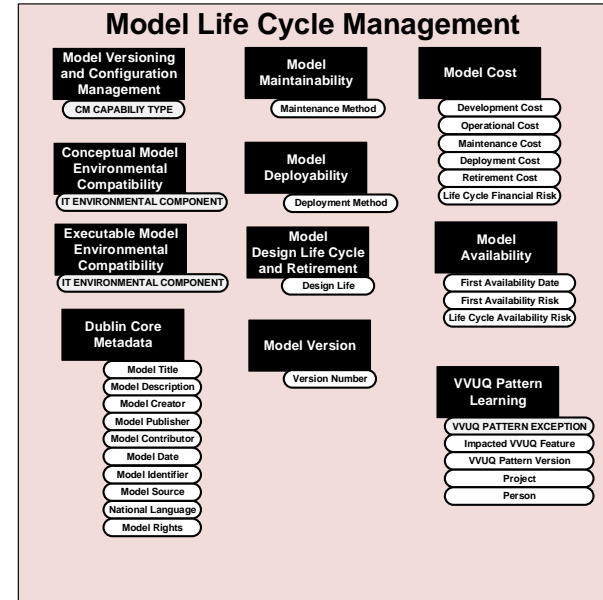
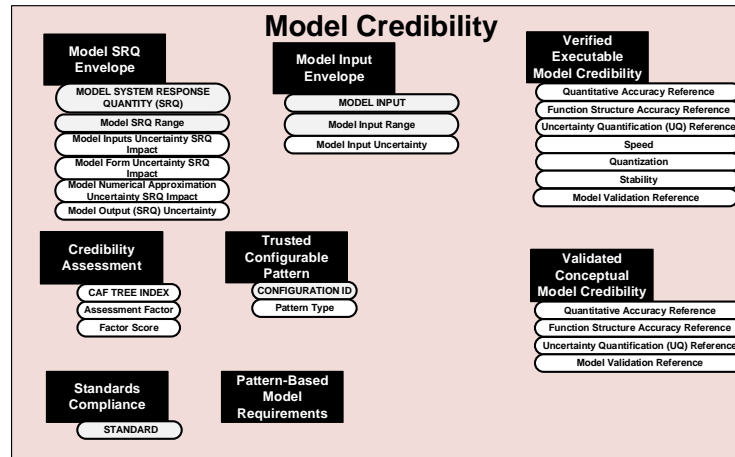
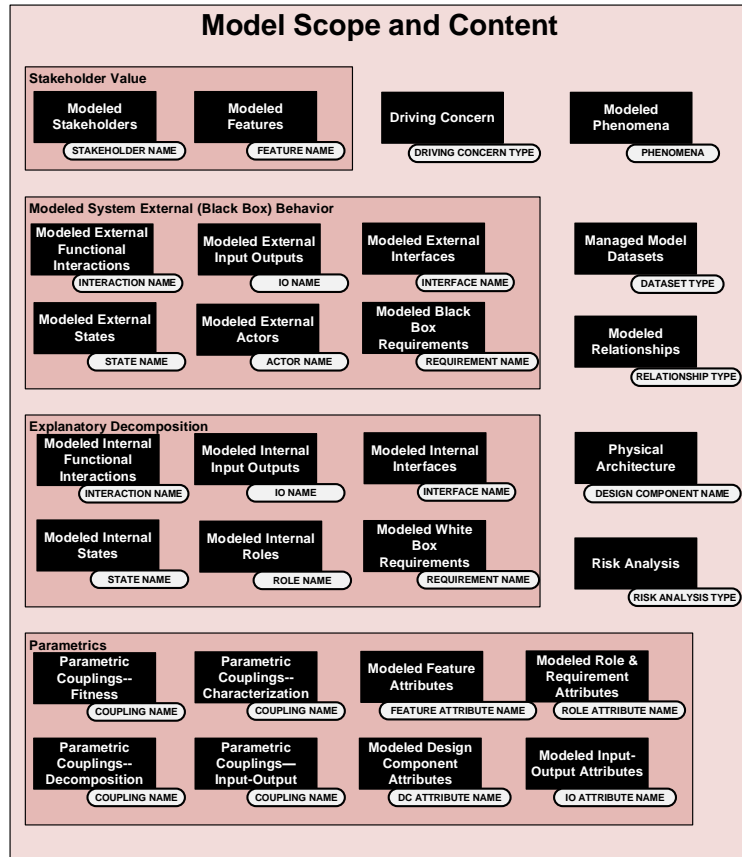
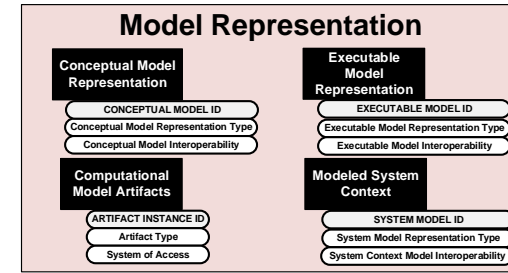
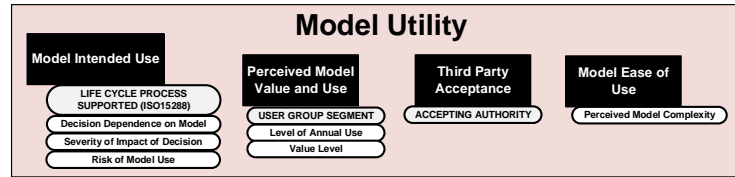
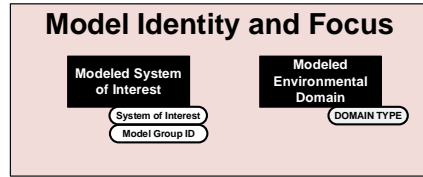
Model Supported Experiments and Observations

Describes the experimentation and observation related to creating and supported by the model, but does not cover the ability to perform those experiments

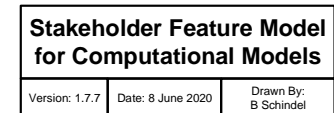
Model Life Cycle Management

Describes the related model life cycle management capabilities.

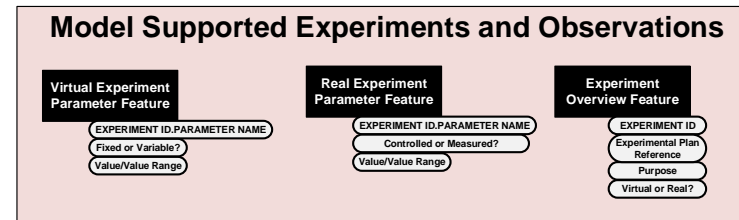
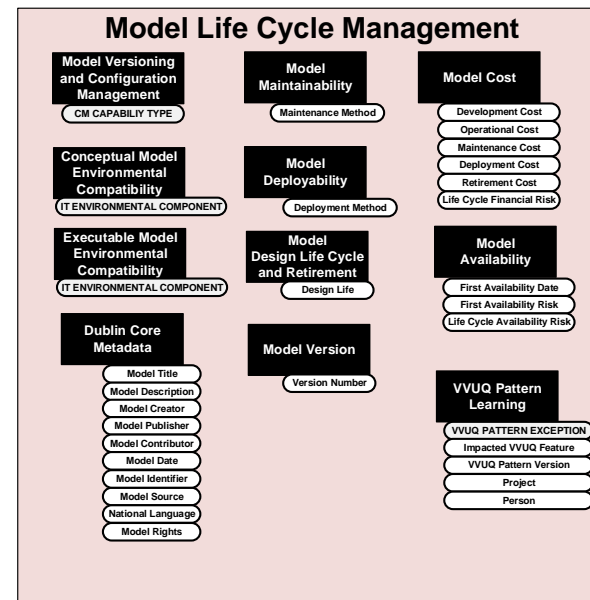
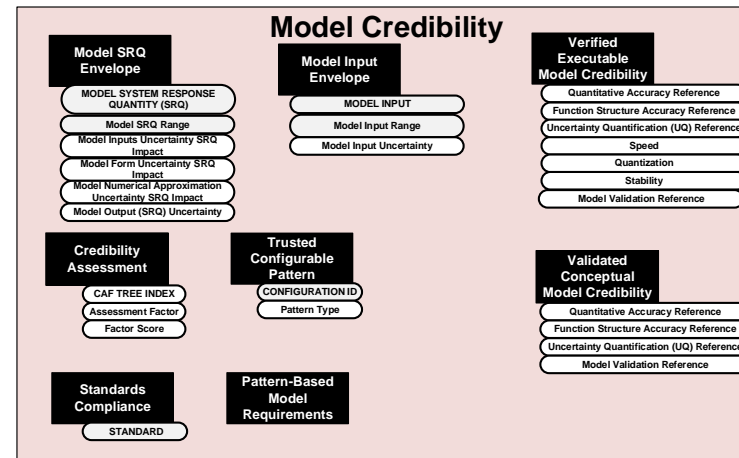
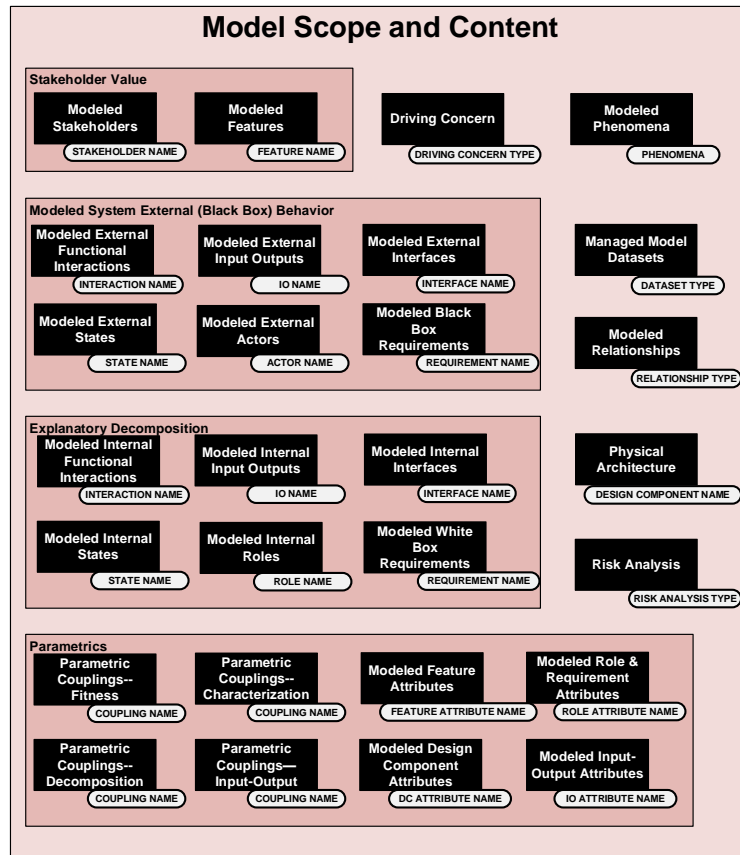
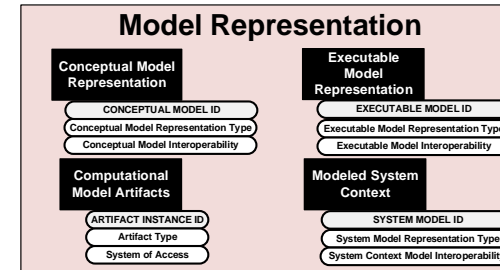
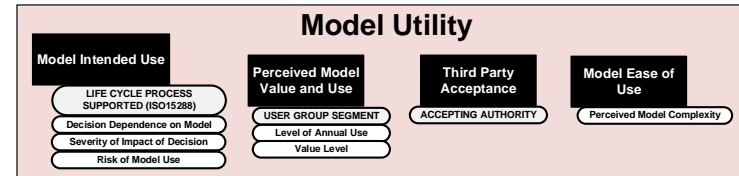
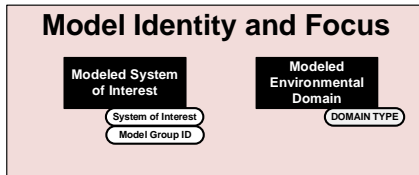
The Full Set of MCP Features



Legend:



Computational Model Feature Groups: 60 Features, in 7 Feature Groups, *Configurable for Specific Models*



Legend:

STAKEHOLDER FEATURE

FEATURE PK ATTRIBUTE
Other Feature Attribute
Other Feature Attribute

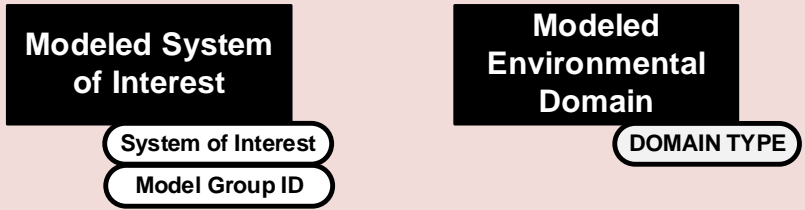
Stakeholder Feature Model for Computational Models

Version: 1.7.7

Date: 8 June 2020

Drawn By: B Schindel

Model Identity and Focus



In ASME V&V50 subcommittee work, the Modeled System of Interest above typically focuses on a manufacturing process (including material in process), usually relating it to some manufactured product.

Features	Feature Attribute	Attribute Definition	Possible Value(s)
Model Identity and Focus: Identifies the main subject or focus of the model			
Modeled System of Interest			
Identifies the type of system this model describes.	System of Interest	Name of system of interest, or class of systems of interest	
	Model Group ID	Identifies the Model Group, if any, to which the model belongs.	
Modeled Environmental Domain			
Identifies the type of external environmental domain(s) that this model includes.	Domain Type(s)	Name(s) of modeled domains (manufacturing, distribution, use, etc.)	

Model Utility

Model Intended Use

LIFE CYCLE PROCESS SUPPORTED (ISO15288)

Decision Dependence on Model

Severity of Impact of Decision

Risk of Model Use

Perceived Model Value and Use

USER GROUP SEGMENT

Level of Annual Use

Value Level

Third Party Acceptance

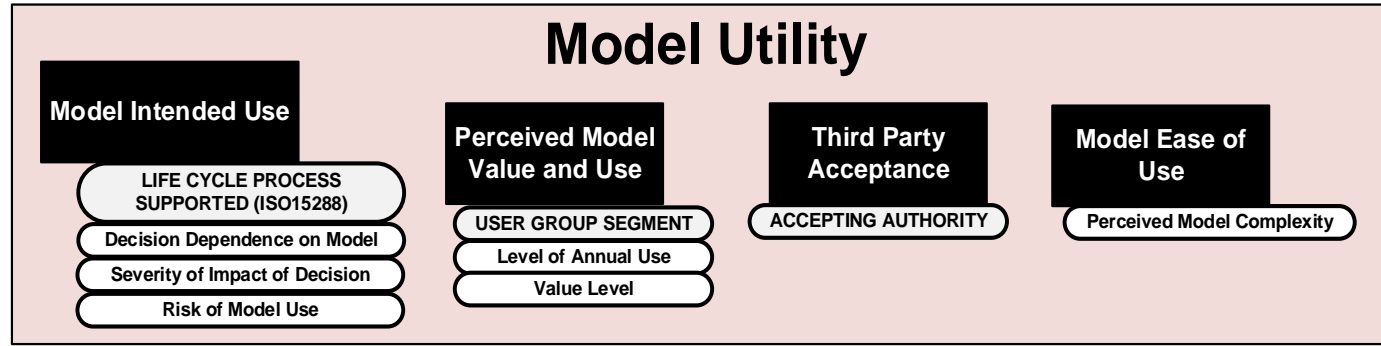
ACCEPTING AUTHORITY

Model Ease of Use

Perceived Model Complexity

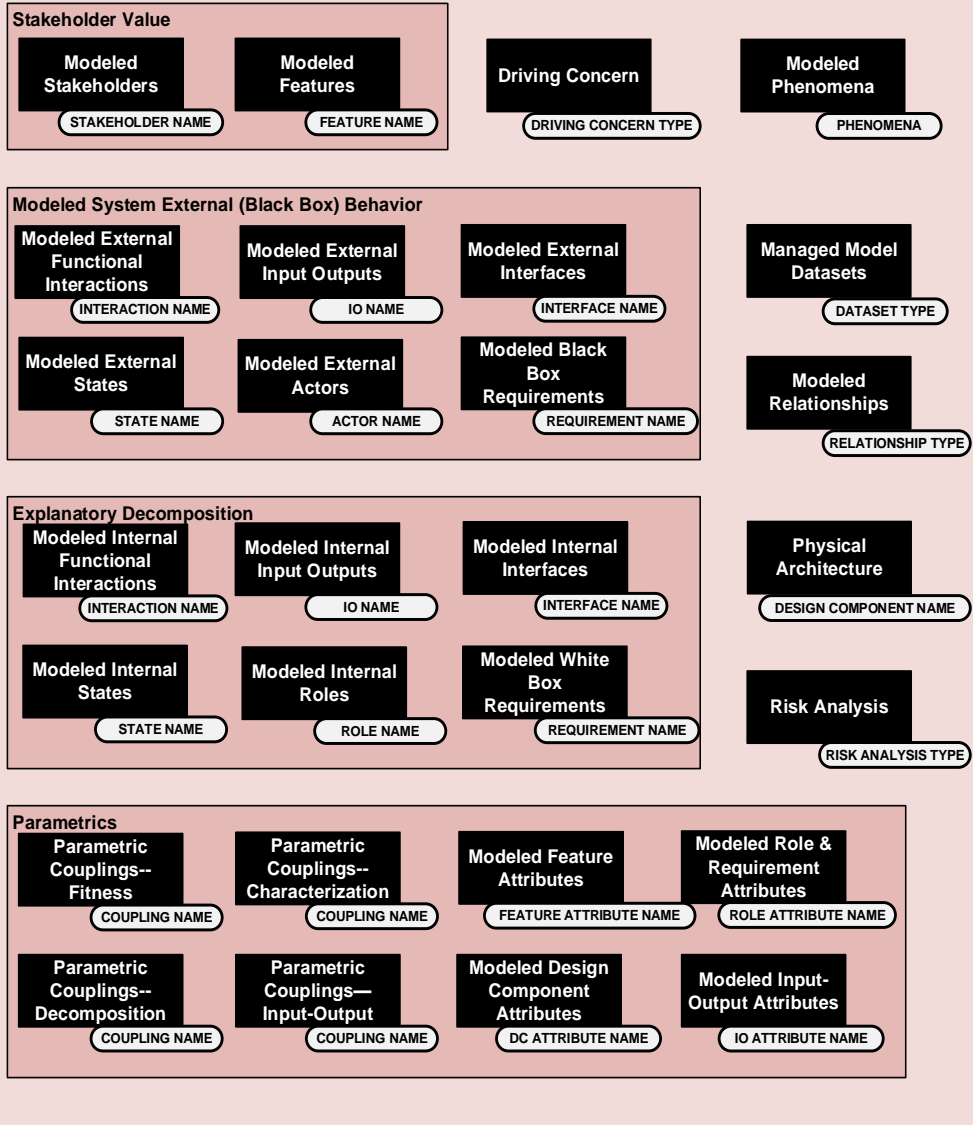
Features	Feature Attribute	Attribute Definition	Possible Value(s)
Model Utility: Describes the intended use, utility, and value of the model			
Model Intended Use			
The intended purpose(s) or use(s) of the model.	Life Cycle Process Supported	The intended life cycle management process to be supported by the model, from the ISO15288 process list. More than	
	Decision Dependence on Model	The extent to which a life cycle decision depends on information from the model.	
	Severity of Impact of Decision	The severity of the life cycle decision, independent of the model.	
	Risk of Model Use	The level of risk associated with the current version of the model.	
Perceived Model Value and Use			
The relative level of value ascribed to the model, by those who use it for its stated purpose.	User Group Segment	The identity of user group segment (multiple)	
	Level of Annual Use	The relative level of annual use by the segment	
	Value Level	The value class associated with the model by that segment	

Model Utility



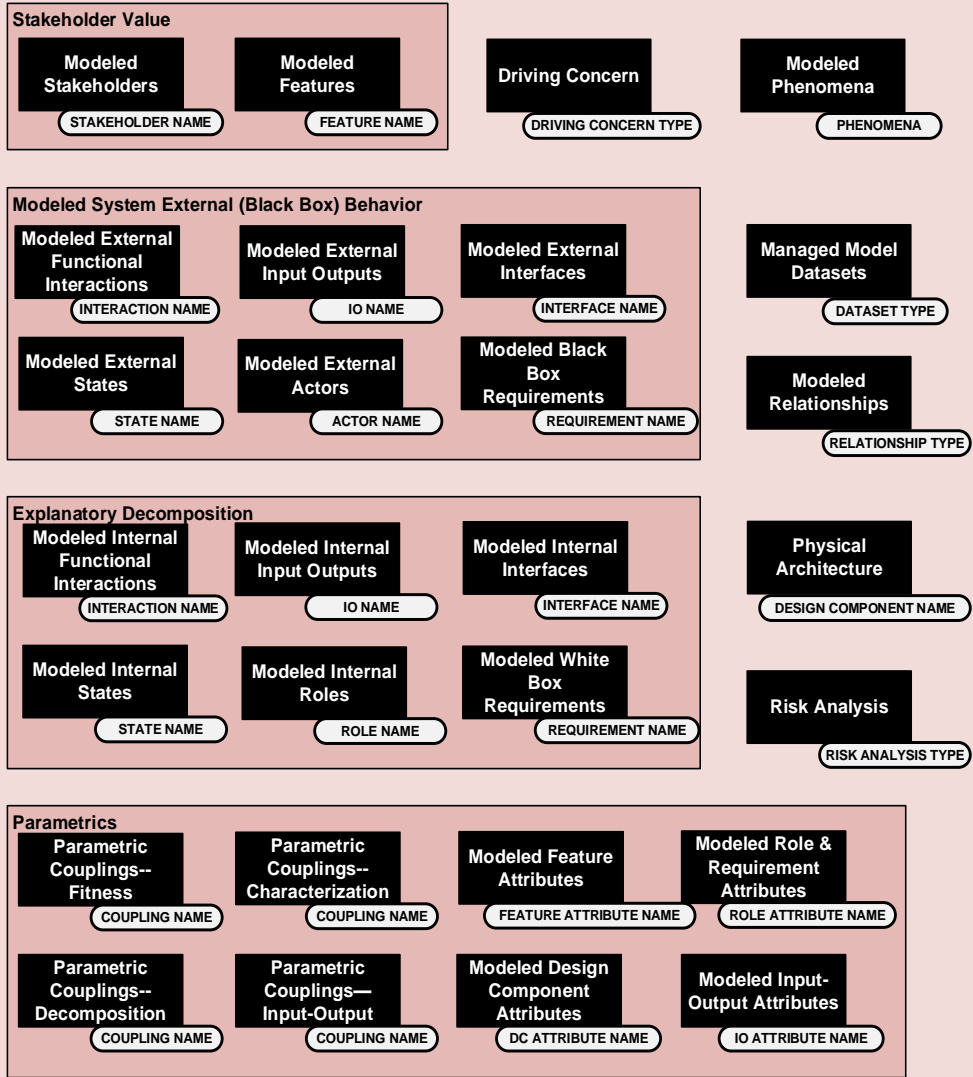
Features	Feature Attribute	Attribute Definition	Possible Value(s)
Third Party Acceptance			
The degree to which the model is accepted as authoritative, by third party regulators, customers, supply chains, and other entities, for its	Accepting Authority	The identity (may be multiple) of regulators, agencies, customers, supply chains, accepting the model	
Model Ease of Use			
The perceived ease with which the model can be used, as experienced by its intended users	Perceived Model Complexity	The qualitative value of the perceived complexity	

Model Scope and Content



Features	Feature Attribute	Attribute Definition	Possible Value(s)
Model Scope: Describes the scope of content of the model			
Stakeholder Value: The capability of the model to describe fitness or value of the System of Interest, by identifying its stakeholders and modeling the related Stakeholder Features.			
Modeled Stakeholders			
The capability of the model to describe fitness or value of the System of Interest, by identifying its stakeholders.	Stakeholder Name	The name(s) of covered stakeholders.	
Modeled Features			
The capability of the model to describe fitness or value of the System of Interest, by modeling the related Stakeholder Features.	Feature Name	The name(s) of covered features.	

Model Scope and Content



Features	Feature Attribute	Attribute Definition	Possible Value(s)
Modeled System External (Black Box) Behavior: The capability of the model to represent the objective external ("black box") technical behavior of the system, through significant interactions with its environment, based on modeled input-output exchanges through external interfaces, quantified by technical performance measures, and varying behavioral modes.			
Modeled External Functional Interactions			
The capability of the model to represent the objective external ("black box") technical behavior of the system, through significant interactions with its environment.	Interaction Name	The name(s) of functional interactions.	
Modeled External Input Outputs			
The capability of the model to represent the objective external ("black box") technical behavior of the system based on modeled input-output exchanges.	IO Name	The name(s) of Input-Outputs.	
Modeled External Interfaces			
The capability of the model to represent the objective external ("black box") technical behavior of the system through external interfaces.	Interface Name	The name(s) of interfaces.	

Model Scope and Content

Stakeholder Value

- Modeled Stakeholders (STAKEHOLDER NAME)
- Modeled Features (FEATURE NAME)

- Driving Concern (DRIVING CONCERN TYPE)

- Modeled Phenomena (PHENOMENA)

Modeled System External (Black Box) Behavior

- Modeled External Functional Interactions (INTERACTION NAME)
- Modeled External Input Outputs (IO NAME)
- Modeled External Interfaces (INTERFACE NAME)
- Modeled External States (STATE NAME)
- Modeled External Actors (ACTOR NAME)
- Modeled Black Box Requirements (REQUIREMENT NAME)

- Managed Model Datasets (DATASET TYPE)

- Modeled Relationships (RELATIONSHIP TYPE)

Explanatory Decomposition

- Modeled Internal Functional Interactions (INTERACTION NAME)
- Modeled Internal Input Outputs (IO NAME)
- Modeled Internal Interfaces (INTERFACE NAME)
- Modeled Internal States (STATE NAME)
- Modeled Internal Roles (ROLE NAME)
- Modeled White Box Requirements (REQUIREMENT NAME)

- Physical Architecture (DESIGN COMPONENT NAME)

- Risk Analysis (RISK ANALYSIS TYPE)

Parametrics

- Parametric Couplings--Fitness (COUPLING NAME)
- Parametric Couplings--Characterization (COUPLING NAME)
- Modeled Feature Attributes (FEATURE ATTRIBUTE NAME)
- Modeled Role & Requirement Attributes (ROLE ATTRIBUTE NAME)
- Parametric Couplings--Decomposition (COUPLING NAME)
- Parametric Couplings—Input-Output (COUPLING NAME)
- Modeled Design Component Attributes (DC ATTRIBUTE NAME)
- Modeled Input-Output Attributes (IO ATTRIBUTE NAME)

Features	Feature Attribute	Attribute Definition	Possible Value(s)
Modeled External States			
The capability of the model to represent the objective external ("black box") technical behavior of the system, through varying behavioral modes.	State Name	The name(s) of states.	
Modeled External Actors			
The capability of the model to represent the objective external ("black box") technical behavior of the system, through modeled external actors.	Actor Name	The name(s) of external actors.	
Modeled Black Box Requirements			
The capability of the model to represent the objective external ("black box") technical behavior of the system, quantified by technical performance measures in requirements statements.	Requirement Name	The name(s) of requirement names.	

Model Scope and Content

Stakeholder Value

Modeled Stakeholders

STAKEHOLDER NAME

Modeled Features

FEATURE NAME

Driving Concern

DRIVING CONCERN TYPE

Modeled Phenomena

PHENOMENA

Modeled System External (Black Box) Behavior

Modeled External Functional Interactions

INTERACTION NAME

Modeled External Input Outputs

IO NAME

Modeled External Interfaces

INTERFACE NAME

Modeled External States

STATE NAME

Modeled External Actors

ACTOR NAME

Modeled Black Box Requirements

REQUIREMENT NAME

Managed Model Datasets

DATASET TYPE

Modeled Relationships

RELATIONSHIP TYPE

Explanatory Decomposition

Modeled Internal Functional Interactions

INTERACTION NAME

Modeled Internal Input Outputs

IO NAME

Modeled Internal Interfaces

INTERFACE NAME

Modeled Internal States

STATE NAME

Modeled Internal Roles

ROLE NAME

Modeled White Box Requirements

REQUIREMENT NAME

Physical Architecture

DESIGN COMPONENT NAME

Risk Analysis

RISK ANALYSIS TYPE

Parametrics

Parametric Couplings--Fitness

COUPLING NAME

Parametric Couplings--Characterization

COUPLING NAME

Modeled Feature Attributes

FEATURE ATTRIBUTE NAME

Modeled Role & Requirement Attributes

ROLE ATTRIBUTE NAME

Parametric Couplings--Decomposition

COUPLING NAME

Parametric Couplings--Input-Output

COUPLING NAME

Modeled Design Component Attributes

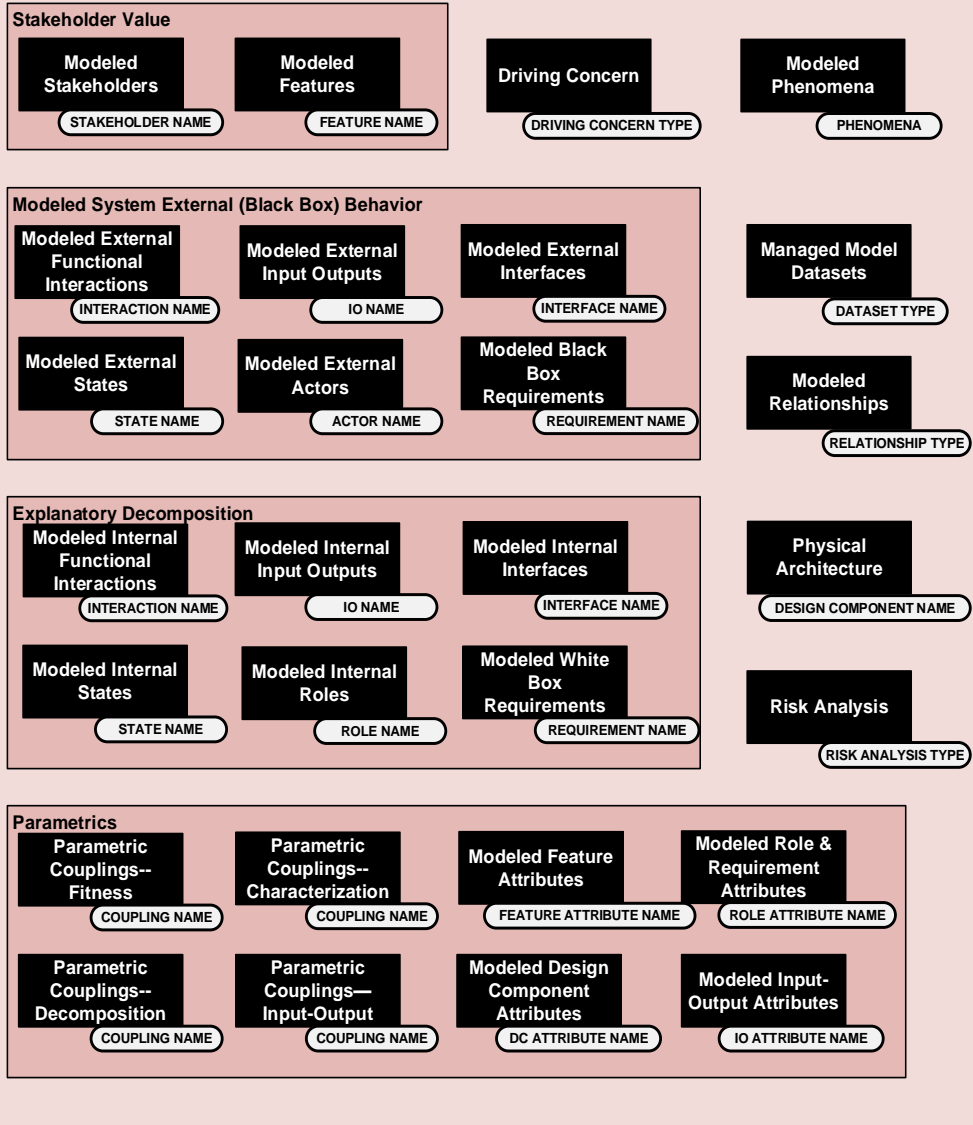
DC ATTRIBUTE NAME

Modeled Input-Output Attributes

IO ATTRIBUTE NAME

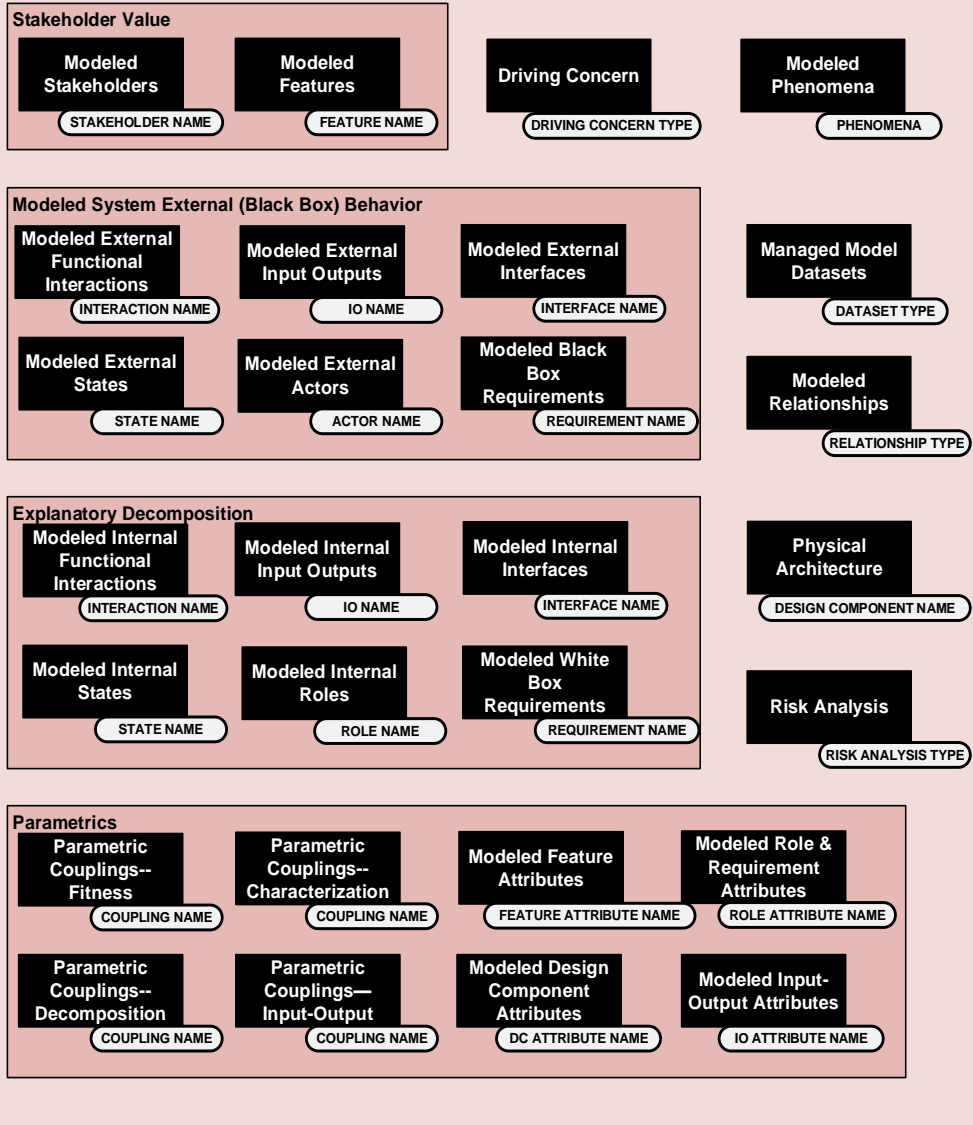
Features	Feature Attribute	Attribute Definition	Possible Value(s)
<p>Explanatory Decomposition: The capability of the model to represent the decomposition of its external technical behavior, as explanatory internal (“white box”) internal interactions of decomposed roles, further quantified by internal technical performance measures, and varying internal behavioral modes.</p>			
<p>Modeled Internal Functional Interactions</p>			
<p>The capability of the model to represent the decomposition of its external technical behavior, as explanatory internal (“white box”) interactions.</p>	Interaction Name	The name(s) of functional interactions.	
<p>Modeled Internal Input Outputs</p>			
<p>The capability of the model to represent the decomposition of its external technical behavior, as explanatory internal (“white box”) input-outputs among "white boxes."</p>	IO Name	The name(s) of Input-Outputs.	
<p>Modeled Internal Interfaces</p>			
<p>The capability of the model to represent the decomposition of its external technical behavior, as explanatory internal (“white box”) interfaces.</p>	Interface Name	The name(s) of interfaces.	

Model Scope and Content



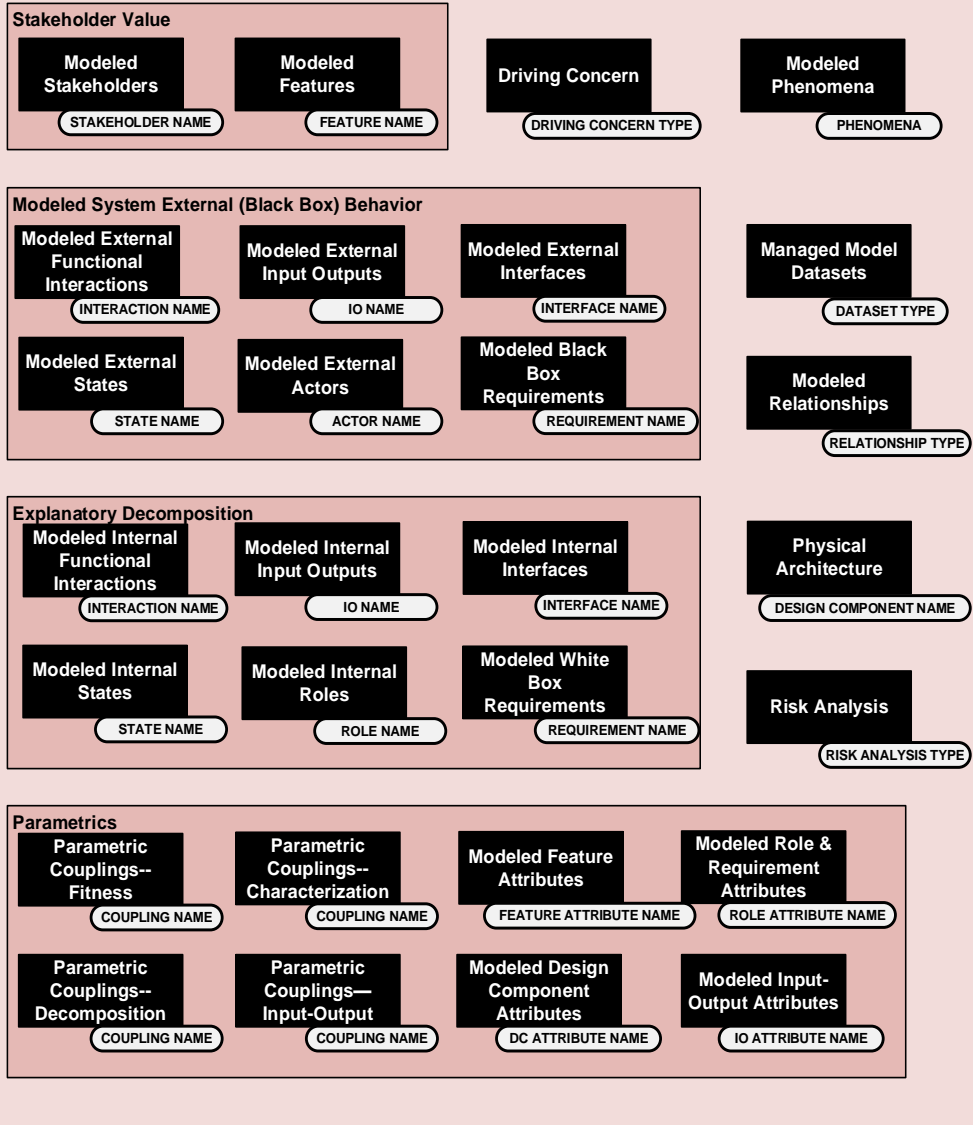
Features	Feature Attribute	Attribute Definition	Possible Value(s)
Modeled Internal States			
The capability of the model to represent the decomposition of its external technical behavior, as varying internal behavioral modes.	State Name	The name(s) of states.	
Modeled Internal Roles			
The capability of the model to represent the decomposition of its external technical behavior, as decomposed roles.	Role Name	The name(s) of external actors.	
Modeled White Box Requirements			
The capability of the model to represent the decomposition of its external technical behavior, as quantified by internal technical performance measures in requirement statements.	Requirement Name	The name(s) of requirement names.	

Model Scope and Content



Features	Feature Attribute	Attribute Definition	Possible Value(s)
Parametrics: The capability of the model to represent the parameters at stakeholder value, performance measure, and physical identity levels and the couplings between them.			
Parametric Couplings--Fitness The capability of the model to represent quantitative (parametric) couplings between stakeholder-valued measures of effectiveness and objective external black box behavior performance measures.	Fitness Coupling Name	The name(s) of Fitness Couplings.	
Parametric Couplings--Decomposition			
The capability of the model to represent quantitative (parametric) couplings between objective external black box behavior variables and objective internal white box behavior variables.	Decomposition Coupling Name	The name(s) of Decomposition Couplings.	

Model Scope and Content



Features	Feature Attribute	Attribute Definition	Possible Value(s)
Parametric Couplings--Characterization			
The capability of the model to represent quantitative (parametric) couplings between objective behavior variables and physical identity (material of construction, part or model number).	Characterization Coupling Name	The name(s) of Characterization Couplings.	
Parametric Couplings--Input-Output			
The capability of the model to represent quantitative (parametric) couplings between input-output variables.	IO Coupling Name	The name(s) of IO Couplings.	
Modeled Feature Attributes			
The capability of the model to represent Feature Attributes.	Feature Attribute Name	The name(s) of Feature Attributes.	
Modeled Role & Requirement Attributes			
The capability of the model to represent Role & Requirement Attributes.	Role Attribute Name	The name(s) of Role Attributes.	

Model Scope and Content

Stakeholder Value

Modeled Stakeholders

STAKEHOLDER NAME

Modeled Features

FEATURE NAME

Driving Concern

DRIVING CONCERN TYPE

Modeled Phenomena

PHENOMENA

Modeled System External (Black Box) Behavior

Modeled External Functional Interactions

INTERACTION NAME

Modeled External Input Outputs

IO NAME

Modeled External Interfaces

INTERFACE NAME

Modeled External States

STATE NAME

Modeled External Actors

ACTOR NAME

Modeled Black Box Requirements

REQUIREMENT NAME

Managed Model Datasets

DATASET TYPE

Modeled Relationships

RELATIONSHIP TYPE

Explanatory Decomposition

Modeled Internal Functional Interactions

INTERACTION NAME

Modeled Internal Input Outputs

IO NAME

Modeled Internal Interfaces

INTERFACE NAME

Modeled Internal States

STATE NAME

Modeled Internal Roles

ROLE NAME

Modeled White Box Requirements

REQUIREMENT NAME

Physical Architecture

DESIGN COMPONENT NAME

Risk Analysis

RISK ANALYSIS TYPE

Parametrics

Parametric Couplings--Fitness

COUPLING NAME

Parametric Couplings--Characterization

COUPLING NAME

Modeled Feature Attributes

FEATURE ATTRIBUTE NAME

Modeled Role & Requirement Attributes

ROLE ATTRIBUTE NAME

Parametric Couplings--Decomposition

COUPLING NAME

Parametric Couplings--Input-Output

COUPLING NAME

Modeled Design Component Attributes

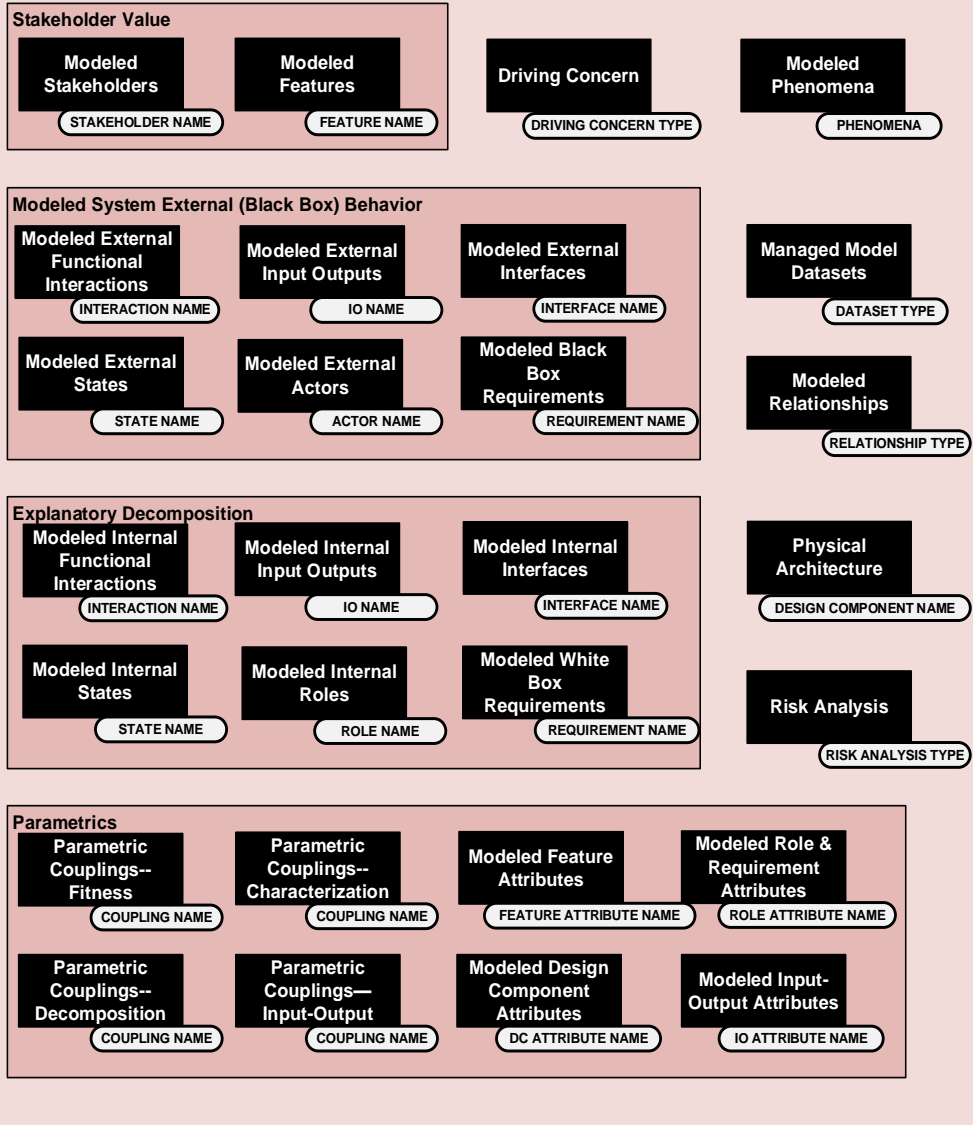
DC ATTRIBUTE NAME

Modeled Input-Output Attributes

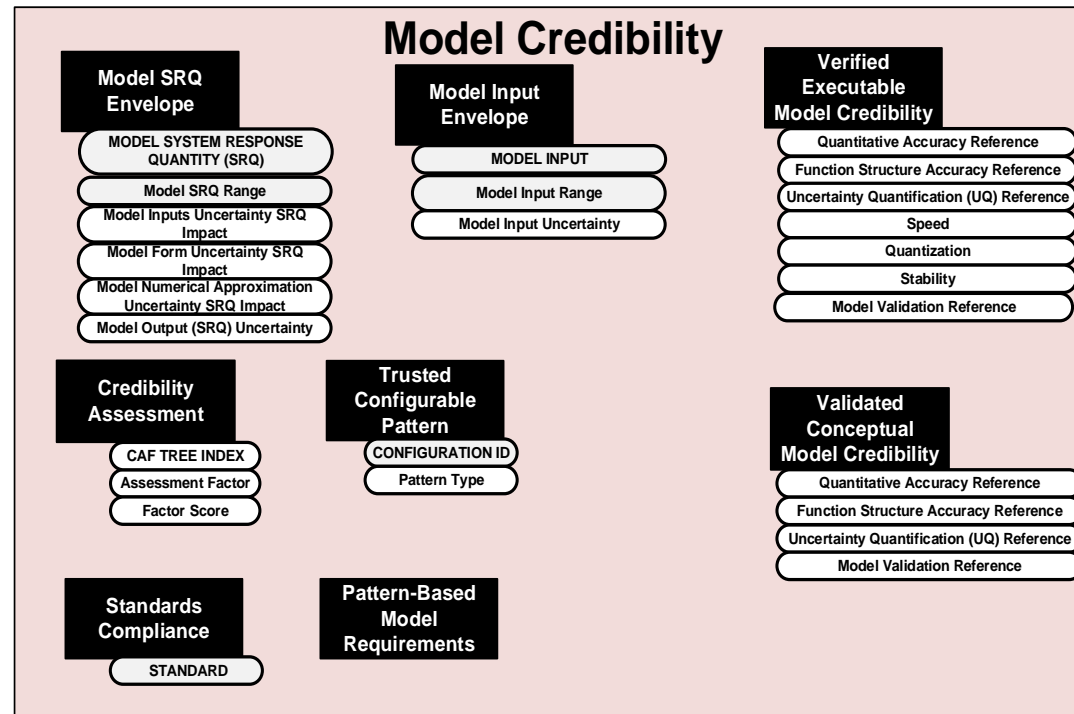
IO ATTRIBUTE NAME

Features	Feature Attribute	Attribute Definition	Possible Value(s)
Modeled Design Component Attributes			
The capability of the model to represent Design Component Attributes.	DC Attribute Name	The name(s) of Design Component Attributes.	
Modeled Input-Output Attributes			
The capability of the model to represent Input-Output Attributes.	IO Attribute Name	The name(s) of IO Attributes.	
Driving Concern			
Identifies the main focus driving the implementation of the model.	Driving Concern Type	The type(s) of driving concern.	
Modeled Phenomena			
Identifies the phenomena being represented in the model.	Phenomenon	The name(s) of individual phenomenon.	
Physical Architecture			
The capability of the model to represent the physical architecture of the system of interest. This includes identification of its major physical components and their architectural relationships.	Design Component Name	The name(s) of Design Components.	

Model Scope and Content

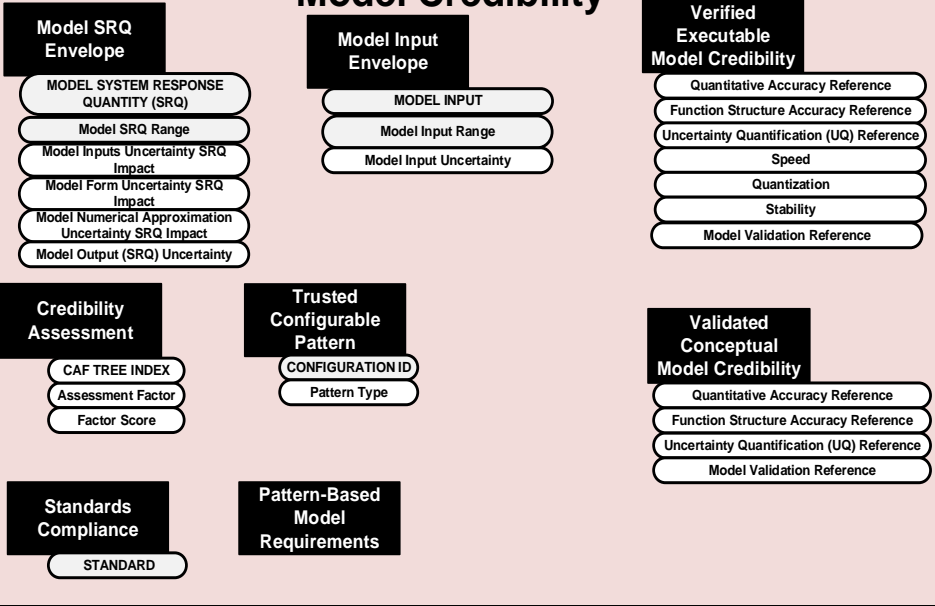


Features	Feature Attribute	Attribute Definition	Possible Value(s)
Managed Model Datasets			
The capability of the model to include managed datasets for use as inputs, parametric characterizations, or outputs	Dataset Type	The type(s) of data sets.	
Modeled Relationships			
The capability of the model to represent various types of modeled relationships.	Relationship Type	The name(s) of types of relationships.	
Risk Analysis			
The capability of the model to include Risk analysis in various forms.	Risk Analysis Type	The name(s) of types of risk analysis.	



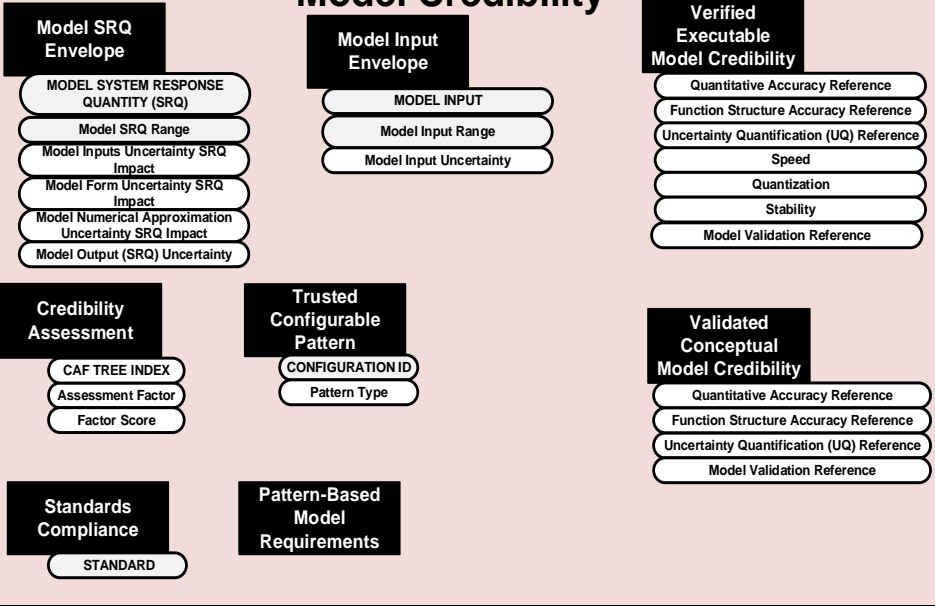
Features	Feature Attribute	Attribute Definition	Possible Value(s)
Model Credibility: Describes the credibility of the model			
Credibility Assessment			
The assessed credibility of a model, based upon a credibility assessment framework.	CAF Tree Index	An index indicating the position of a credibility assessment factor, in a tree of roll-up credibility assessment factors, in the form	
	Assessment Factor	The name of a credibility assessment factor in a credibility assessment framework.	
	Factor Score	The score value awarded to a credibility assessment factor for a particular model	

Model Credibility



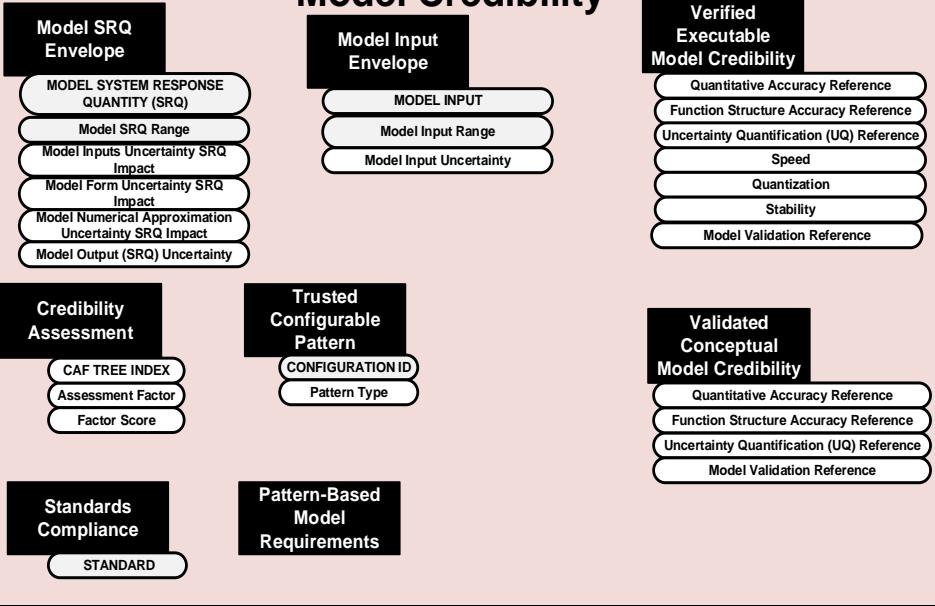
Features	Feature Attribute	Attribute Definition	Possible Value(s)
Model SRQ Envelope			
The capability of the model to meet its Model Credibility requirements over a stated range (envelope) of dynamical System Response Quantities (SRQs), which are outputs of the model.	Model SRQ	The System Response Quantity (SRQ) of the Model Application Envelope (not experimental envelope) being constrained. This is the same as the output of the model, and this Feature Attribute would be populated once for each output.	
	Model SRQ Range	The range over which the SRQ in the model is intended for use.	
	Model Inputs Uncertainty SRQ Impact	The SRQ-Associated impact of the aleatory, epistemic, and/or mixed uncertainty associated with inputs used in the model, including system parameters and environmental description.	
	Model Form Uncertainty SRQ Impact	The SRQ-Associated impact of the uncertainty in model form.	
	Model Numerical Approximation Uncertainty SRQ Impact	The SRQ-Associated impact of the numerical approximation uncertainty including discretization error, iterative convergence error, round-off error, and errors due to computer programming mistakes.	
	Model Output (SRQ) Uncertainty	An estimate of the SRQ uncertainty resulting from a combination of the model inputs, model form, and model numerical approximation uncertainties.	

Model Credibility



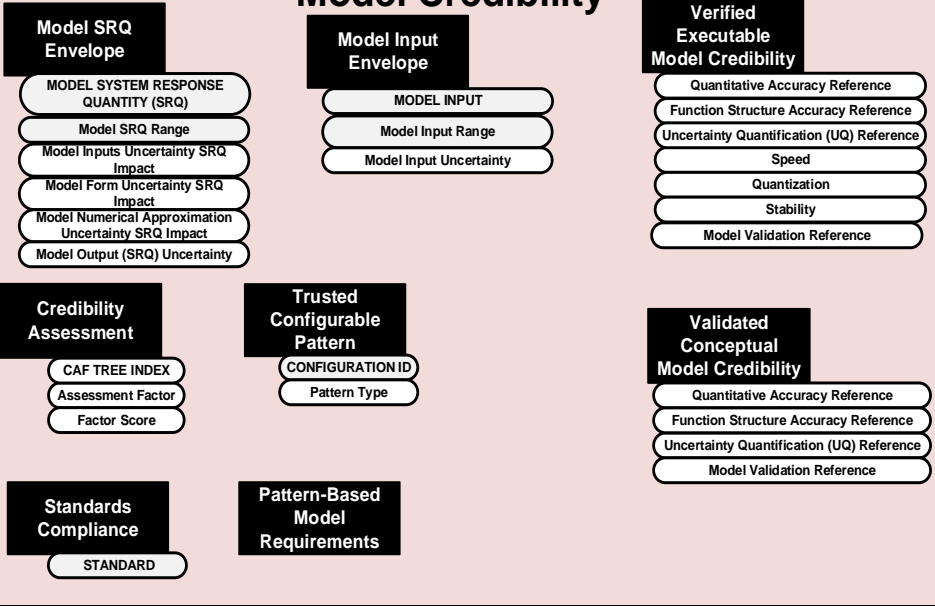
Features	Feature Attribute	Attribute Definition	Possible Value(s)
Model Input Envelope			
The capability of the model to meet its Model Credibility requirements over a stated range (envelope) of dynamical Inputs.	Model Input	The name(s) of the input of the Model Application Envelope (not experimental envelope) being constrained.	
	Model Input Range	The range over which the input in the model is intended for use.	
	Input Uncertainty	The aleatory, epistemic, and/or mixed uncertainty associated with the input.	
Standards Compliance			
Conforming to formal standards for models, modeling, model VVUQ, security, information technology, or other model supporting	Standard	The identification of a standard applicable to models, modeling, model VVUQ, security, information technology, or other model-supporting standards.	
Pattern-Based Model Requirements			
The requirements for this model were configured from the general model requirements pattern.			

Model Credibility



Features	Feature Attribute	Attribute Definition	Possible Value(s)
Trusted Configurable Pattern			
The capability of the model to be consistent with a trusted configurable pattern, representing different modeled system	Configuration ID	A specific system of interest configuration within the family that the pattern framework can represent.	
	Pattern ID	The identifier of the trusted configurable pattern.	
Validated Conceptual Model Credibility			
The validated capability of the conceptual portion of the model to represent the System of Interest, with acceptable Credibility.	Quantitative Accuracy Reference	The specification reference describing the quantitative accuracy of the conceptual model compared to the system of interest.	
	Function Structure Accuracy Reference	The specification reference describing the structural (presence or absence of behaviors) accuracy of the conceptual model compared to the system of interest.	
	Uncertainty Quantification (UQ) Reference	The specification reference describing the degree of uncertainty of the Credibility of the conceptual model to the system of interest.	
	Model Validation Reference	The reference documenting the validation of the conceptual model's Credibility to the system of interest.	

Model Credibility



Features	Feature Attribute	Attribute Definition	Possible Value(s)
Verified Executable Model Credibility			
The verified capability of the executable portion of the model to represent the System of Interest, with acceptable Credibility.	Quantitative Accuracy Reference	The specification reference describing the quantitative accuracy of the executable model to the conceptual model.	
	Structural Accuracy Reference	The specification reference describing the structural (presence or absence of elements) accuracy of the executable model to the conceptual model.	
	Uncertainty Quantification (UQ) Reference	The specification reference describing the degree of uncertainty of the Credibility of the executable model to the conceptual model.	
	Speed	The specification reference describing the execution run time (speed) for the executable model.	
	Quantization	The specification reference describing the quantization error of the executable e model.	
	Stability	The specification reference describing the level of stability of the accuracy and uncertainty of the executable model error characteristics.	
	Model Verification Reference	The reference documenting the verification of the executable model's Credibility to the conceptual model.	

Model Representation

Conceptual Model Representation

CONCEPTUAL MODEL ID

Conceptual Model Representation Type

Conceptual Model Interoperability

Computational Model Artifacts

ARTIFACT INSTANCE ID

Artifact Type

System of Access

Executable Model Representation

EXECUTABLE MODEL ID

Executable Model Representation Type

Executable Model Interoperability

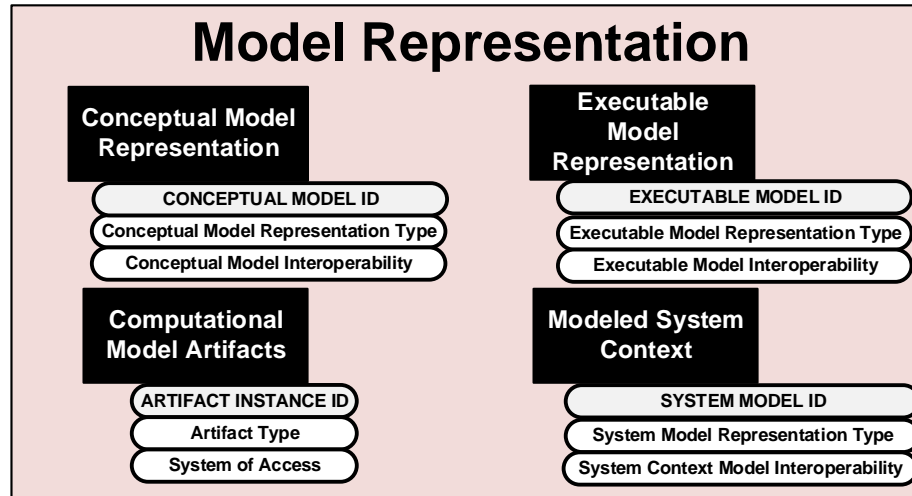
Modeled System Context

SYSTEM MODEL ID

System Model Representation Type

System Context Model Interoperability

Features	Feature Attribute	Attribute Definition	Possible Value(s)
Model Representation: Identifies the type of representation used by the model			
Modeled System Context			
The capability of the system context portion of the model to represent the system of interest, using a specific type of representation	System Model ID	The name of the system model.	
	System Model Representation Type	The type of system modeling language or metamodel used.	SysML (Cameo/MagicDraw) with Systematica Profile
	System Context Model Interoperability	The degree of interoperability of the system context model, for exchange with other environments	
Conceptual Model Representation			
The capability of the conceptual portion of the model to represent the system of interest, using a specific type of representation.	Conceptual Model ID	The name of the conceptual model.	
	Conceptual Model Representation Type	The type of conceptual modeling language or metamodel used.	
	Conceptual Model Interoperability	The degree of interoperability of the conceptual model, for exchange with other environments	
Executable Model Representation			
The capability of the executable portion of the model to represent the system of interest, using a specific type of representation	Executable Model ID	The name of the executable model.	
	Executable Model Representation Type	The type of executable modeling language or metamodel used.	
	Executable Model Interoperability	The degree of interoperability of the executable model, for exchange with other environments	



Features	Feature Attribute	Attribute Definition	Possible Value(s)
Computational Model Artifacts			
The capability of the model system (including its life cycle management system) to create, maintain, and access artifacts of the development and use of the	Artifact Instance ID	The unique identifier of an artifact.	
	Artifact Type	The type of an artifact	Model Validation Reference, Model Verification Reference
	System of Access	The method of accessing an artifact.	

Model Life Cycle Management

Model Versioning and Configuration Management

CM CAPABILITY TYPE

Model Maintainability

Maintenance Method

Model Cost

- Development Cost
- Operational Cost
- Maintenance Cost
- Deployment Cost
- Retirement Cost
- Life Cycle Financial Risk

Conceptual Model Environmental Compatibility

IT ENVIRONMENTAL COMPONENT

Model Deployability

Deployment Method

Executable Model Environmental Compatibility

IT ENVIRONMENTAL COMPONENT

Model Design Life Cycle and Retirement

Design Life

Model Availability

- First Availability Date
- First Availability Risk
- Life Cycle Availability Risk

Dublin Core Metadata

- Model Title
- Model Description
- Model Creator
- Model Publisher
- Model Contributor
- Model Date
- Model Identifier
- Model Source
- National Language
- Model Rights

Model Version

Version Number

VVUQ Pattern Learning

- VVUQ PATTERN EXCEPTION
- Impacted VVUQ Feature
- VVUQ Pattern Version
- Project
- Person

Features	Feature Attribute	Attribute Definition	Possible Value(s)
Model Life Cycle Management: Describes related model life cycle management capabilities			
Model Versioning and Configuration Management			
The capability of the model to provide for version and configuration management.	CM Capability Type	The type(s) of CM capabilities included (may be multiple)	
System Context Model Environmental Compatibility			
The capability of the system context model to be compatibly supported by specified information technology environment(s),	IT Environmental Component	The type(s) of IT environments or standards supported	
Conceptual Model Environmental Compatibility			
The capability of the conceptual model to be compatibly supported by specified information technology environment(s),	IT Environmental Component	The type(s) of IT environments or standards supported	
Executable Model Environmental Compatibility			
The capability of the model to be compatibly supported by specified information technology environment(s), indicating	IT Environmental Component	The type(s) of IT environments or standards supported	

Model Life Cycle Management

Model Versioning and Configuration Management

CM CAPABILITY TYPE

Model Maintainability

Maintenance Method

Model Cost

- Development Cost
- Operational Cost
- Maintenance Cost
- Deployment Cost
- Retirement Cost
- Life Cycle Financial Risk

Conceptual Model Environmental Compatibility

IT ENVIRONMENTAL COMPONENT

Model Deployability

Deployment Method

Executable Model Environmental Compatibility

IT ENVIRONMENTAL COMPONENT

Model Design Life Cycle and Retirement

Design Life

Model Availability

- First Availability Date
- First Availability Risk
- Life Cycle Availability Risk

Dublin Core Metadata

- Model Title
- Model Description
- Model Creator
- Model Publisher
- Model Contributor
- Model Date
- Model Identifier
- Model Source
- National Language
- Model Rights

Model Version

Version Number

VVUQ Pattern Learning

- VVUQ PATTERN EXCEPTION
- Impacted VVUQ Feature
- VVUQ Pattern Version
- Project
- Person

Features	Feature Attribute	Attribute Definition	Possible Value(s)
Model Design Life and Retirement			
The capability of the model to be sustained over an indicated design life, and retired on a planned basis.	Design Life	The planned retirement date	
Model Maintainability			
The relative ease with which the model can be maintained over its intended life cycle and use, based on capable maintainers, availability	Maintenance Method	The type of maintenance methodology used to maintain the model's capability and availability for the intended purposes over the intended life cycle.	
Model Deployability			
The capability of the model to support deployment into service on behalf of intended users, in its original or subsequent updated	Deployment Method	The type of method used to deploy (possibly in repeating cycles) the model into its intended use environment.	

Model Life Cycle Management

Model Versioning and Configuration Management

CM CAPABILITY TYPE

Model Maintainability

Maintenance Method

Model Cost

- Development Cost
- Operational Cost
- Maintenance Cost
- Deployment Cost
- Retirement Cost
- Life Cycle Financial Risk

Conceptual Model Environmental Compatibility

IT ENVIRONMENTAL COMPONENT

Model Deployability

Deployment Method

Executable Model Environmental Compatibility

IT ENVIRONMENTAL COMPONENT

Model Design Life Cycle and Retirement

Design Life

Model Availability

- First Availability Date
- First Availability Risk
- Life Cycle Availability Risk

Dublin Core Metadata

- Model Title
- Model Description
- Model Creator
- Model Publisher
- Model Contributor
- Model Date
- Model Identifier
- Model Source
- National Language
- Model Rights

Model Version

Version Number

VVUQ Pattern Learning

- VVUQ PATTERN EXCEPTION
- Impacted VVUQ Feature
- VVUQ Pattern Version
- Project
- Person

Features	Feature Attribute	Attribute Definition	Possible Value(s)
Model Cost			
The financial cost of the model, including development, operating, and maintenance cost	Development Cost	The cost to develop the model, including its validation and verification, to its first availability for service date	
	Operational Cost	The cost to execute and otherwise operate the model, in standardized execution load units	
	Maintenance Cost	The cost to maintain the model	
	Deployment Cost	The cost to deploy, and redeploy updates, per cycle	
	Retirement Cost	The cost to retire the model from service, in a planned fashion	
	Life Cycle Financial Risk	Risk to the overall life cycle cost of the model	

Model Life Cycle Management

Model Versioning and Configuration Management

CM CAPABILITY TYPE

Model Maintainability

Maintenance Method

Model Cost

- Development Cost
- Operational Cost
- Maintenance Cost
- Deployment Cost
- Retirement Cost
- Life Cycle Financial Risk

Conceptual Model Environmental Compatibility

IT ENVIRONMENTAL COMPONENT

Model Deployability

Deployment Method

Executable Model Environmental Compatibility

IT ENVIRONMENTAL COMPONENT

Model Design Life Cycle and Retirement

Design Life

Model Availability

- First Availability Date
- First Availability Risk
- Life Cycle Availability Risk

Dublin Core Metadata

- Model Title
- Model Description
- Model Creator
- Model Publisher
- Model Contributor
- Model Date
- Model Identifier
- Model Source
- National Language
- Model Rights

Model Version

Version Number

VVUQ Pattern Learning

- VVUQ PATTERN EXCEPTION
- Impacted VVUQ Feature
- VVUQ Pattern Version
- Project
- Person

Features	Feature Attribute	Attribute Definition	Possible Value(s)
Model Availability			
The degree and timing of availability of the model for its intended use, including date of its first availability and the degree of ongoing availability thereafter.	First Availability Date	Date when version will first be available	
	First Availability Risk	Risk to the scheduled date of first availability	
	Life Cycle Availability Risk	Risk to ongoing availability after introduction	
VVUQ Pattern Learning			
The ability to accumulate new discoveries about model-based methods into the VVUQ Pattern, as it is applied over model life cycles. These discoveries are exceptions to the existing VVUQ Pattern, and	VVUQ Pattern Exception	A summary of the exception noted to the current VVUQ Pattern (may be multiple exceptions)	
	Impacted VVUQ Feature	The impacted existing, modified, or additional feature of the VVUQ Pattern.	
	VVUQ Pattern Version	The version of the VVUQ Pattern in current use before change.	
	Project	Identifies the project in which the exception was noted	
	Person	Identifies the person describing the exception	

Model Life Cycle Management

Model Versioning and Configuration Management

CM CAPABILITY TYPE

Model Maintainability

Maintenance Method

Model Cost

- Development Cost
- Operational Cost
- Maintenance Cost
- Deployment Cost
- Retirement Cost
- Life Cycle Financial Risk

Conceptual Model Environmental Compatibility

IT ENVIRONMENTAL COMPONENT

Model Deployability

Deployment Method

Executable Model Environmental Compatibility

IT ENVIRONMENTAL COMPONENT

Model Design Life Cycle and Retirement

Design Life

Model Availability

- First Availability Date
- First Availability Risk
- Life Cycle Availability Risk

Dublin Core Metadata

- Model Title
- Model Description
- Model Creator
- Model Publisher
- Model Contributor
- Model Date
- Model Identifier
- Model Source
- National Language
- Model Rights

Model Version

Version Number

VVUQ Pattern Learning

- VVUQ PATTERN EXCEPTION
- Impacted VVUQ Feature
- VVUQ Pattern Version
- Project
- Person

Features	Feature Attribute	Attribute Definition	Possible Value(s)
Dublin Core Metadata			
Describes the model metadata using information from the Dublin Core Metadata specification.			
	Model Title	A name given to the resource.	
	Model Subject	The topic of the resource.	
	Model Description	An account of the resource.	
	Model Creator	An entity primarily responsible for making the resource.	
	Model Publisher	An entity responsible for making the resource available.	
	Model Contributor	An entity responsible for making contributions to the resource.	
	Model Date	A point or period of time associated with an event in the lifecycle of the resource.	
	Model Format	The file format, physical medium, or dimensions of the resource.	
	Model Identifier	An unambiguous reference to the resource within a given context.	
	Model Source	A related resource from which the described resource is derived.	
	National Language	A language of the resource.	
	Model Rights	Information about rights held in and over the resource.	

Model Life Cycle Management

Model Versioning and Configuration Management

CM CAPABILITY TYPE

Conceptual Model Environmental Compatibility

IT ENVIRONMENTAL COMPONENT

Executable Model Environmental Compatibility

IT ENVIRONMENTAL COMPONENT

Dublin Core Metadata

- Model Title
- Model Description
- Model Creator
- Model Publisher
- Model Contributor
- Model Date
- Model Identifier
- Model Source
- National Language
- Model Rights

Model Maintainability

Maintenance Method

Model Deployability

Deployment Method

Model Design Life Cycle and Retirement

Design Life

Model Version

Version Number

Model Cost

- Development Cost
- Operational Cost
- Maintenance Cost
- Deployment Cost
- Retirement Cost
- Life Cycle Financial Risk

Model Availability

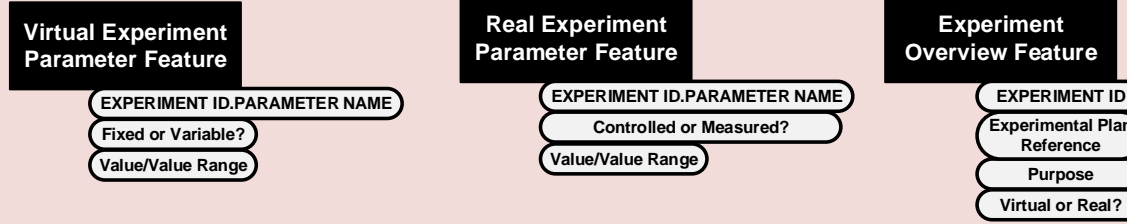
- First Availability Date
- First Availability Risk
- Life Cycle Availability Risk

VVUQ Pattern Learning

- VVUQ PATTERN EXCEPTION
- Impacted VVUQ Feature
- VVUQ Pattern Version
- Project
- Person

Features	Feature Attribute	Attribute Definition	Possible Value(s)
Model Version			
Identifies the version number assigned to the model.	Version Number	Version number associated with the model	

Model Supported Experiments and Observations



Features	Feature Attribute	Attribute Definition	Possible Value(s)
Model Supported Experiments and Observations: Describes the experimentation and observation related to creating and supported by the model.			
Experiment Overview Feature			
An overview of the experiments including identification, procedure reference, purpose and environment related to the	Experiment ID	The number or code identifying the individual experiment. (More than one value may be populated)	
	Experimental Plan Reference	A reference to the document that describes the procedure followed in the experiment.	
	Experiment Purpose	A description of the reason for doing the experiment.	Science-Oriented, Engineering Design-Oriented, DOE Screening, Model-Testing, Model-Training, In-Service Operation
	Virtual or Real?	A selection of whether the experiment is done using a virtual or real system.	

Model Supported Experiments and Observations

Virtual Experiment Parameter Feature

- EXPERIMENT ID.PARAMETER NAME
- Fixed or Variable?
- Value/Value Range

Real Experiment Parameter Feature

- EXPERIMENT ID.PARAMETER NAME
- Controlled or Measured?
- Value/Value Range

Experiment Overview Feature

- EXPERIMENT ID
- Experimental Plan Reference
- Purpose
- Virtual or Real?

Features	Feature Attribute	Attribute Definition	Possible Value(s)
Virtual Experiment Parameter Feature			
A description of the parameter set associated with each experiment in the virtual environment.	Experiment ID.Parameter Name	A concatenation of the Experiment ID from the Experiment Overview Feature and the name of the parameter in the experiment. (One instance for each parameter in the experiment.)	
	Fixed or Variable?	A selection of whether the parameter is fixed or variable during the experiment.	
	Value/Value Range	The associated fixed or variable value(s) for the parameters during the experiment.	
Real Experiment Parameter Feature			
A description of the parameter set associated with each experiment in the real environment.	Experiment ID.Parameter Name	A concatenation of the Experiment ID from the Experiment Overview Feature and the name of the parameter in the experiment. (One instance for each parameter in the experiment.)	
	Controlled or Measured?	A selection of whether the parameter is controlled or measured during the experiment.	
	Value/Value Range	The associated controlled or measured value(s) for the parameters during the experiment.	

Appendix II: MCP Technical Requirements--Configurable

Technical Requirements for a Computational Model

				AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	
1				Model Stakeholder Feature Group >>				Model Representation				Model Utility				Model Life Cycle							
2				Model Stakeholder Feature Name >>				Conceptual Model Representation	Executable Model Representation	Model Intended Use	Perceived Model Value and Use	Third Party Acceptance	Model Ease of Use	Model Versioning and Configuration	Executable Model Environmental	Model Design Life and Retirement	Model Maintainability	Model Deployability	Model Cost				
3	Model Stakeholder Feature Attribute >>			Conceptual Model Representation	Conceptual Model Interoperability	Executable Model Representation	Executable Model Interoperability	Life Cycle Process Supported	User Group Segment	Level of Annual Use	Value Level	Accepting Authority	Perceived Model Complexity	CM Capability Type	IT Environmental Component	Design Life	Maintenance Method	Deployment Method	Development Cost	Operational Cost	Maintenance Cost	Deployment Cost	
4	Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion																			
5	5. Model User Interface																						
5.1	Model User Interface	The model user interfaces, per the [Model UI Specification] shall facilitate the efficient and effective performance of the intended purpose of the model by a user of the designated persona type.	Includes reports, displays, views, and other outputs, as well as interactive user interface specifications.					X	X	X	X	X	X									X	
5.2	Model User Persona	The user of the model shall have the background and capabilities indicated by the [Model User Persona].							X		X		X									X	
6	6 Model Versioning and Configuration Management																						
6.1	Model Versioning	The model shall carry versioning information compatible with a required configuration management method.									X		X		X								
6.2	Managed Dataset Versioning	The model's managed datasets shall carry versioning information compatible with a required configuration management									X		X										
6.3	Model Documentation Versioning	The model's documentation shall carry versioning information compatible with a required configuration management									X		X										
7	7. Model Life Cycle Management																						
7.1	7.1 Operating Environment																						
7.1.1	IT Environment--Conceptual	The conceptual model shall be compatible with [Conceptual Modeling IT	This IT environment reference is for technologies used to support the conceptual model, such as modeling tools, content or configuration management systems.								X				X		X	X		X	X		
7.1.2	IT Environment--Implemented Model	The executable model shall be compatible with [Executable Model IT Environment].	This IT environment reference is for technologies used to support the development, deployment, maintenance, and run time use of the implemented model, such as development tools or configuration management systems. Whether the system reference								X				X		X	X		X	X		
Model Stakeholder Types				Model Stakeholder Features				Model Reqrmts, Traced to Ftrs															

Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
1	1. Model Focus and Domain		
1.1	System of Interest	The model shall identify the focal system of interest.	
1.2	External Domain	The model shall represent all the external Domain Actors with which the subject system significantly interacts	The Domain Environment is the context in which the modeled System of Interest interacts with the Actors that inhabit that domain. This part of the model requirements simply identifies (lists) those external actors, so that interactions with them may later be identified. Those interactions will be key parts of the model being specified. All external behavior is in the context of those interactions. "Interact" means exchange of energy, force, mass flow, or information, resulting in impact on state. "Significantly" means with respect to impact on the subject system stakeholder requirements (measures of effectiveness).
2	2. Scope of Model Content		
2.1	2. 1 Stakeholder Fitness Model		
2.1.1	Stakeholders	The model shall represent and define all the types and instances of Stakeholders with a significant stake in the System of Interest, across its life cycle.	Models of technical systems very frequently model physics aspects (as in PB Models) or at least the external manifestations of physical behavior (as in DD Models), and both of these cases are about objective physical facts, even if stochastic. However, engineered systems are also associated with human or business values, purpose, objectives, fitness for use, or similar (e.g., KPI) considerations. it is the "Stakeholder Feature Attributes" that express these. Although an engineered system is designed with these considerations in mind, "fitness for purpose" or "value" are not just about the behavior of the system of interest—they are also about the external world in which the system of interest will operate. With this in mind, system models frequently include both descriptions of (1) objective technical behavior as well as (2) a description of fitness space. Other terms are sometimes used for these two ideas, but the important point is that both representation of objective technical behavior and representation of stakeholder value are essential to engineered systems. This includes "trade studies"/ tradeoff analysis, other change impact analyses, failure modes and effects analyses, sensitivity analyses, and other engineering uses apply models of both technical behavior and its value.
2.1.2	Stakeholder Features	For modeled Stakeholder for the system of interest, the model shall represent and define all the Stakeholder Features of the System of Interest, representing packages of significant stakeholder value or fitness for intended use and life cycle of the System of Interest.	
2.1.3	Stakeholder Feature Attributes	For each identified Stakeholder Feature, the model shall represent and define all the Feature Attributes that parameterize or quantify the degree or type of stakeholder value or fitness.	
2.1.4	Parametric Couplings--Fitness	For each Measure of Effectiveness (Feature Attribute), the model shall represent the quantitative coupling that determines its values versus those of the Measures of Performance upon which its valuation or fitness depends.	
			The External Technical Performance Attributes, identified earlier above, and the Fitness or Value Attributes, also identified earlier above, are "coupled" in the sense that there are a quantitative relationships (couplings) between them. These "curves" or "surfaces" are how we express variation of utility with respect to technical performance. Examples include likelihood of purchase selection versus (coupled to) the technical features of a smart phone, or relative preferences for speed versus cost of an automobile.

Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
2.2	2.2 External Behavior Model		
2.2.1	External Interfaces	The Model shall represent the external Input-Outputs exchanged during interactions with Domain Actors, and the external Interfaces through which they are exchanged.	Input-Outputs are flows of energy, force, mass, or information, exchanged during the interactions noted above. These flow through Interfaces. Examples of Interfaces include radiating or absorbing surfaces, mechanical connections or fasteners, hydraulic connections, electrical connectors, liquid-liquid or liquid-solid boundaries, keyboards, displays, chemically active interfaces, sensors, actuators, biologically active interfaces, etc.
2.2.2	External Interactions	The model shall represent all the significant external interactions that the system of interest has with its listed environmental actors, listing which actors are involved in each interaction.	All behavior, and all the laws of the physical sciences, is in the context of Interactions, consisting of the exchange of energy, force, mass flow, or information, leading to state change in the interacting entities. Representing Interactions is accordingly central to Physics-Based Models. In addition, Data-Driven Models represent discovered and compressed description of the external appearance of those interactions, even though no underlying physics-based cause may be included. So, both types of models require that the models include identification of all the significant <u>external</u> interactions that the subject system has with its environmental actors. "Significant" in this requirement is always evaluated in terms of its impact on the modeled system stakeholder measures of effectiveness. Note that this requirement is not about interactions that are internal to the system of interest. Those are only of interest for certain types of models, and covered in another section later below.
2.2.3	Parasitics--External	The modeled external interactions shall include any parasitic aspects which arise from choice of internal design, materials, technologies, or solution approach but which were not otherwise required by the primary intended system purpose, where significant from a stakeholder perspective.	These are in principle a subset of the External Interactions referred to in the preceding section, but are noted here so that they are not overlooked. Some interactions that a system has with its environment may be "accidents" of its design, selected technology, or the environment itself. For example, a mechanical structural member (a part) may contribute parasitic or "stray" electrical capacitance that impacts the electronic behavior of the system. In engineered (human designed) systems, these interactions might be considered to fall in the category of "unintended" interactions, but they are just as real as those intended, and may have large technical and stakeholder impacts. Failure modes are a part of this behavior.

Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
2.2	2.2 External Behavior Model		
2.2.4	Dynamical Variables--External	For each identified Interaction, the model shall include the dynamically changing quantities significant to the interaction, for both the System of Interest and the External Actors in the Interaction.	The external behavior Interactions identified above are further parameterized by technical Measures of Performance, providing numerical or other measures that quantify the external behavior of the system objectively, without regard to stakeholder-judged “goodness”. Typical measures of this type include position, temperature, pressure, rates of change of those variables, mass flow rate, timing, or other technical measures. These parameters include the variables of physics and what technical instrumentation tries to measure. They are further divided into “fast changing dynamic variables” that describe system dynamics, and “slow changing static parameters” such as heat capacity, power ratings, mechanical dimensions or geometry, etc.
2.2.5	Static Parameters--External	For each identified Interaction, the model shall include the static or slow changing quantities characterizing the system’s performance of the interaction, for both the System of Interest and the External Actors in the Interaction.	
2.2.6	External Modes, States	The model shall represent the different behavioral modes (states) of the system of interest that are significant to the intended use of the model.	
2.2.7	External State Transitions	The model shall represent the possible (state) transitions between the modeled system behavioral modes.	States of a system of interest may be a finite set of “modes” (e.g., liquid, solid, gas, on, off, idling, cruising, stopped, shutting down, spinning up, steady state, landing, ascending, etc.) or a more continuous set of values of a state variable (temperature, pressure, position, velocity, etc.). In both cases, the state of the system of interest bears on (influences) its responses to inputs. This part of the model is concerned with the finite list of system modes. Both physics-based and data-driven models can be used to describe differing behavior of a system of interest in those different modes (from a finite list of states).
2.2.8	External Mode Characterization	For each of its modeled behavioral modes (states), the model shall represent which external interactions the system of interest can have with its environmental actors, from the list of possible interactions.	
2.2.9	Black Box Requirements	For each modeled interaction of the system of interest with its environment, the required external behavior of the system of interest shall be included in the model.	Requirements effectively describe transformations of system inputs into system outputs, parameterized in some cases by the system state or other parameters. "Black Box" refers to the idea that all such behaviors are visible external to the system of interest, behaving as an opaque element interacting with its environment, without visibility of its internals.

Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
2.3	2.3 Internal Behavior Model		
2.3.1	Internal Roles	For each modeled external Interaction, the model shall represent the decomposition of the behavior of the system of interest into internal interactions between internal roles.	Physics-based Models describe how internal interactions within the system of interest result in emergent characteristics of that system as a whole, as it interacts with its environmental actors. Accordingly, the behavior of the system of interest in its external interactions is decomposed into internal behavioral components. Examples include fluid dynamics models, continuum mechanics models of internal elasticity stress-strain interactions, models of thermal conduction through solids, models of mechanical part couplings leading to whole machine behaviors, etc. In each of these cases, behavior of the whole is decomposed into behavior of smaller elements and interactions between them.
2.3.2	Allocatable Roles	The model shall represent the internal decomposition of the system of interest functional roles until small enough to be allocated to single physical components of the modeled physical architecture.	
2.3.3	Dynamical Variables--Internal	For each modeled internal decomposed functional role, the model shall include the dynamically changing quantities significant to the related internal interactions.	The internal behavior Interactions identified above are further parameterized by technical Measures of Performance, providing numerical or other measures that quantify the behavior of the system objectively, without regard to stakeholder “goodness”. Typical measures of this type include position, temperature, pressure, rates of change of those variables, mass flow rate, timing, or other technical measures. These parameters include the variables of physics and what instrumentation tries to measure. They are further divided into “fast changing dynamic variables” that describe system dynamics, and “slow changing static parameters” such as heat capacity, power ratings, mechanical dimensions or geometry, etc.
2.3.4	Static Parameters--Internal	For each modeled internal Interaction, the model shall include the static or slow changing quantities characterizing the system’s performance of the related internal interactions.	
2.3.5	Parametric Couplings--Decomposition	For each behavioral role's Measure of Performance, the model shall represent the quantitative coupling that determines its values versus those of the internal (decomposed) Measures of Performance upon which it depends.	The External Technical Performance Attributes, identified earlier above, and the Internal Role Technical Performance Attributes, also identified above, are “coupled” in the sense that there are a quantitative relationships (couplings) between them. These curves, surfaces, tables, or other relationships express emergence of larger scale technical properties from the properties of decomposed roles. For Physics-Based Models, these couplings explain external behavior as emerging from real internal physical component interaction parameters. For Data-Driven Models, these couplings parameterize external behavior in terms of intermediate variables determined by pattern-extraction tools, but in this case those “internal” parameters may not necessarily have identified physical or explanatory significance.

Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
2.3	2.3 Internal Behavior Model		
2.3.6	Architectural Components	The model shall represent the set of physical components of the system of interest.	Physical architecture is the collection of material parts or segments (described by their identity, not behavior) and their organization (by physical relations between them). At least Physics-Based Models typically include representation of physical architecture.
2.3.7	Component Parameters	For each modeled physical component, the model shall include attributes describing the type or identity of the physical component, indicating material type or composition, manufacturer part number, of other non-behavioral identifier.	
2.3.8	Component Relationships	For each modeled physical component, the model shall represent its physical architectural relationships (connection, adjacency, geometry, containment hierarchy, etc.) with other physical components, defining the physical architecture of the system of interest.	
2.3.9	Parametric Couplings--Characterization	For each modeled physical component, the model shall represent the attribute value couplings between the identity attributes for that physical component and the behavior characterization attributes of any logical role allocated to that component by the model.	The inclusion of a specific physical material, manufactured component, or equipment item in a system of interest results in certain behavioral characteristics. This may be seen, for example, in material data sheets, component or equipment specifications. So, there is a modeled parametric coupling between behavioral attributes (e.g., melting point, hardness, pH, conductivity, elasticity, response time, transfer function, production rate, fuel economy) and the identity (type) attributes of a material, component, or equipment items (e.g., chemical identity, manufacturer part number, etc.). That attribute coupling associates identity attribute values with behavior attribute values.
2.3.10	Parasitics--Internal	The modeled internal behavioral roles and couplings shall include any parasitic aspects which arise from choice of internal design, materials, technologies, or solution approach but which were not otherwise required by the primary intended system purpose, where significant from a stakeholder perspective.	These are in principle a subset of the internal behavior roles and couplings already referred to in the above sections, but are noted here so that they are not overlooked. Some internal interactions of a system may be “accidents” of its design, selected technology, or external environment. For example, a rotating mechanical part may contribute parasitic or “stray” vibration that impacts the behavior of the system. In engineered (human designed) systems, these interactions might be considered to fall in the category of “unintended” interactions, but they are just as real as those intended, and may have large technical and stakeholder impacts. So the requirement in the those section are sufficient to include parasitic interactions, roles, and couplings, and with the same definition of “significant” 68 described. Failure modes are a part of this behavior.

Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
2.3	2.3 Internal Behavior Model		
2.3.11	Physical Allocation	For each modeled functional role (element of behavior), the model shall represent an allocation of that role to a physical component which performs or has that behavior.	For physically-based models, behavior (represented in the model by roles of functional interactions), from above sections, is ultimately associated with physical components, materials, or equipment items that have or perform that behavior. For those physical allocations to be unambiguous, each functional role must be decomposed to small enough behaviors that they can be allocated to a single physical component, leaving no ambiguity as to which physical component instance is responsible for a behavioral role. (A physical component can have more than one role allocated to it, but one role instance should only be allocated to one physical component.)
2.3.12	Allocation Uniqueness	The model shall represent allocation of each fully decomposed functional role to not more than one physical component.	
2.3.13	Allocation Completeness	For each modeled physical component, material, or equipment item, the model shall represent the allocation of all functional roles (elements of behavior) expected of that physical component, material, or equipment item.	
2.3.14	Internal Modes, States	The model shall represent the behavioral modes (states) of the internal system white box roles that are significant to the intended use of the model.	
2.3.15	Internal State Transitions	The model shall represent the possible (state) transitions between the modeled internal behavioral modes.	
2.3.16	Internal Mode Characterization	For each of its modeled internal modes (states), the model shall represent which interactions of internal roles may occur during such modes.	
2.3.17	White Box Requirements	For each Modeled Black box Requirement on the system of interest the model shall provide modeled White Box Requirements traceable to and decomposing that Black Box Requirement.	"White Box" Requirements describe the behavior of the decomposed White Box internal roles of the system, which, interacting with each other internal to the system of interest, result in the Black Box external behavior.

Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
2.4	2.4 Configurability		
2.4.1	Configurability	The model shall include configurability for different cases indicated.	This is about the ability to use the model as a configurable system pattern, re-using it across different system configurations in a common domain.
2.5	2.5 Model Run Datasets		
2.5.1	Managed Model Datasets	The model shall include documented example, validation, and verification data sets, including model inputs, model outputs, and model configuration.	
2.5.2	Queryable Model Datasets	The model shall include task-specific pre-run data sets, allowing their further use without additional model execution runs.	This option provides for post-execution dataset suitable for use by additional user tools, for subsequent analysis or other use.
2.5.3	Dataset Structure and Accuracy	The model run data sets shall satisfy [Data Set Structural] and [Data Set Accuracy] requirements.	
2.6	Failure Modes and Effects		
2.6.1	Failure Mode	The model shall include identification of component failure modes, as to underlying state leading to predicted failure.	Failures shall be judged as to their significance, based on their impact on modeled stakeholder features also in the model.
2.6.2	Failure Cause	For each identified failure mode, the model shall include identification of cause(s) of failure mode.	
2.6.3	Failure Probability	For each identified failure mode, the model shall include the probability of failure mode.	
2.6.4	Failure Effect	For each identified failure mode, the model shall include the effect(s) of the mode.	Failure effects should be significant impacts to modeled stakeholder features.
2.6.5	Effect Severity	For each identified failure effect, the model shall include the severity of impact of the failure.	Severity should be with respect to impact on modeled stakeholder features.

Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
3	3. Model Credibility		
3.1	3.1 Model Envelope		
3.1.1	Modeled Envelope-- Fitness	The model shall represent the system of interest over a specified (discrete or continuous) range or envelope of stakeholder feature configurations.	This is about range of validation and verification of the model. It is a different idea than the configurability of the model.
3.1.2	Modeled Envelope-- External Technical	The model shall represent the system of interest over a specified (discrete or continuous) range or envelope of technical external environment interaction configurations.	
3.1.3	Modeled Envelope-- Physical Design	The model shall represent the system of interest over a specified (discrete or continuous) range or envelope of physical design configurations.	
3.2	3.2 Conceptual Model Credibility		
3.2.1	Conceptual Model UQ--Function Structural	Compared to the modeled system of interest over a specified model envelope, the conceptual model shall satisfy function structural [Accuracy Requirements], within [Uncertainty Requirements], both as consistent with the model's intended use.	This is concerned with confidence in the structure of behavior, such as the presence or absence of individual functional interactions, and includes the conditional probability of their occurrence, timing, and relationships. The test of what behavior to include is with respect to its impact (through couplings) on stakeholder fitness impacting measures of effectiveness.
3.2.2	Conceptual Model UQ--Quantitative	Compared to the modeled system of interest over a specified model envelope, the conceptual model shall satisfy quantitative [Accuracy Requirements], within [Uncertainty Requirements], both as consistent with the model's intended use.	This is concerned with confidence in the quantitative aspects of behavior, indicated by the values of quantitative parameters and their couplings, and includes the conditional probability of their values. The test of what behavior to include is with respect to its impact (through couplings) on stakeholder fitness impacting measures of effectiveness.

Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
3	3. Model Credibility		
3.3	3.3 Implemented Model Credibility		
3.3.1	Implemented Model UQ-- Structural	The implemented computational model shall (compared to the real modeled system of interest over the specified model envelope) satisfy function structural [Accuracy Requirements], within [Uncertainty Requirements], both as consistent with the intended use of the model.	This is concerned with confidence in the structure of behavior, such as the presence or absence of individual functional interactions, and includes the conditional probability of their occurrence, timing, and relationships. The test of what behavior to include is with respect to its impact (through couplings) on stakeholder fitness impacting measures of effectiveness.
3.3.2	Implemented Model UQ-- Quantitative	The implemented computational model shall (compared with the conceptual model over a specified model envelope), satisfy quantitative [Accuracy Requirements], within [Uncertainty Requirements], both as consistent with the intended use of the model.	This is concerned with confidence in the quantitative aspects of behavior, indicated by the values of quantitative parameters and their couplings, and includes the conditional probability of their values. The test of what behavior to include is with respect to its impact (through couplings) on stakeholder fitness impacting measures of effectiveness.
3.3.3	Model Quantization Error	Compared with the conceptual model, the implemented computational model shall satisfy [Quantization Requirements] consistent with its intended use.	This is concerned with implemented model errors, with respect to the idealized conceptual model, caused by representation using finite-resolution (e.g., limited word length digital) representations, and their propagated growth effects.
3.3.4	Model Execution Time	The implemented computational model, in the [Required IT Environment] shall satisfy [Run Time Speed Requirements] consistent with its intended use.	This is concerned with rate of execution of the implemented computational model in the targeted IT environment, a function of both the model design and the IT environment. It may be compared to real time for the modeled system, or in terms of model run time length.
3.3.5	Model Error Stability	The rate of growth in inaccuracy and uncertainty over computational run dimensions for the implemented computational model shall not exceed specified levels.	This is concerned with propagated growth in implemented model error, over time or other forms of propagation through the model. It can be a function of the conceptual model inherent stability sensitivity, as well as implementation approach and quantization error.

Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
4	4. Model Representation		
4.1	4.1 Conceptual Model Representation		
4.1.1	Conceptual Model Representation Type	The conceptual model shall represent the system of interest using a designated model representation or modeling language type.	
4.1.2	Conceptual Model Portability and Interoperability	The conceptual model representation shall satisfy representation portability or interoperability requirements.	
4.1.3	Conceptual Model Documentation-- Use	The conceptual model documentation shall be sufficient for use of the model over its designated design life, by users having the capabilities indicated.	
4.1.4	Conceptual Model Documentation-- Maintenance and Support	"The conceptual model documentation shall be sufficient for maintenance of the model over its designated design life, by maintainers of capability indicated."	
4.1.5	Conceptual Model Documentation-- Model Requirements	The conceptual model documentation shall include the model requirements against which it has been validated, including intended use, content, envelope, accuracy and uncertainty specifications.	The Model Requirements Pattern specifies the categories of requirements that should be included in this documentation.

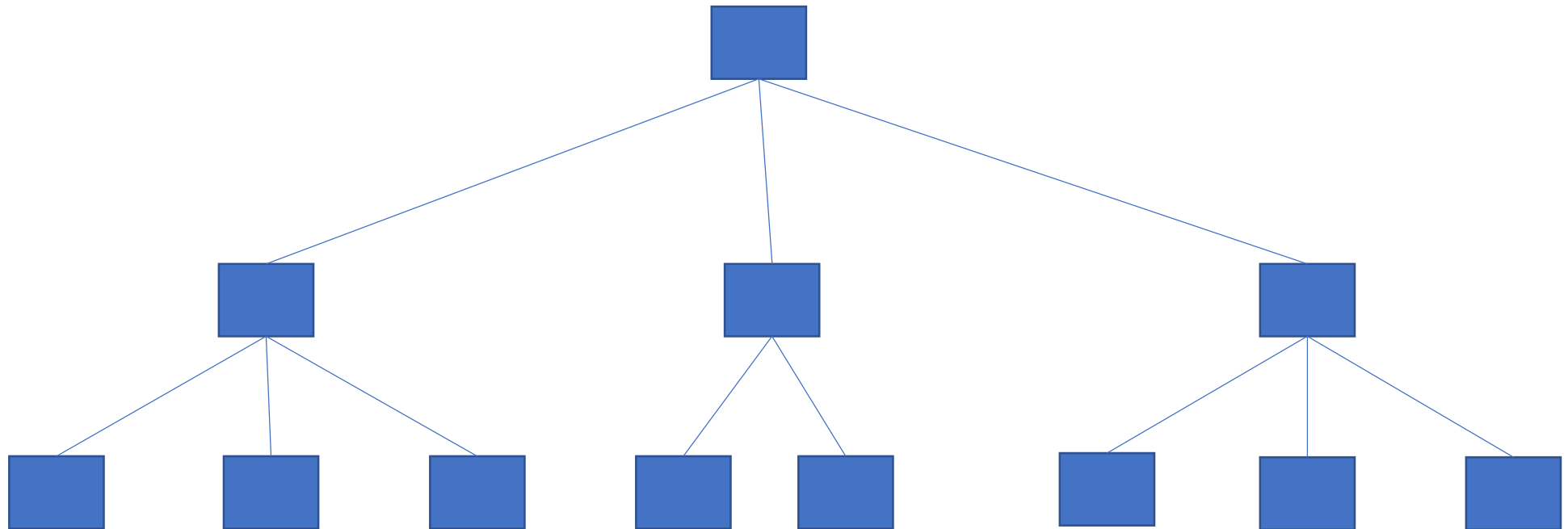
Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
4	4. Model Representation		
4.2	4.2 Implemented Executable Representation		
4.2.1	Implemented Model Representation Type	The implemented executable model shall represent the system of interest using a designated model language type.	
4.2.2	Implemented Model Portability and Interoperability	The implemented executable model representation shall satisfy representation portability or interoperability requirements.	
4.2.3	Implemented Model Documentation-- Use	The implemented model documentation shall be sufficient for use of the model over its designated design life, by users having the capabilities indicated.	
4.2.4	Implemented Model Documentation-- Maintenance and Support	The implemented model documentation shall be sufficient for maintenance and support of the model over its designated design life, by maintainers and supporters having the capabilities indicated.	
4.2.5	Implemented Model Documentation-- Deployment	The implemented model documentation shall be sufficient for deployment of the model over its designated design life, by deployers having the capabilities indicated.	
4.2.6	Implemented Model Documentation-- Model Requirements	The implemented model documentation shall include the model requirements against which it has been verified, including intended use, content, envelope, quantization, execution time, accuracy and uncertainty specifications.	The Model Requirements Pattern specifies the categories of requirements that should be included in this documentation.

Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
5	5. Model User Interface		
5.1	Model User Interface	The model user interfaces, per the [Model UI Specification] shall facilitate the efficient and effective performance of the intended purpose of the model by a user of the designated persona type.	Includes reports, displays, views, and other outputs, as well as interactive user interface specifications.
5.2	Model User Persona	The user of the model shall have the background and capabilities indicated by the [Model User Persona].	
6	6 Model Versioning and Configuration Management		
6.1	Model Versioning	The model shall carry versioning information compatible with a required configuration management method.	
6.2	Managed Dataset Versioning	The model's managed datasets shall carry versioning information compatible with a required configuration management method.	
6.3	Model Documentation Versioning	The model's documentation shall carry versioning information compatible with a required configuration management method.	

Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
7	7. Model Life Cycle Management		
7.1	7.1 Operating Environment		
7.1.1	IT Environment-- Conceptual Model	The conceptual model shall be compatible with [Conceptual Modeling IT Environment].	This IT environment reference is for technologies used to support the conceptual model, such as modeling tools, content or configuration management systems.
7.1.2	IT Environment-- Implemented Model	The executable model shall be compatible with [Executable Model IT Environment].	This IT environment reference is for technologies used to support the development, deployment, maintenance, and run time use of the implemented model, such as development tools or configuration management systems. Whether the system reference boundary of interest is the Overall Model System, which includes the IT Environment, or the smaller Model System, which does not include the IT Environment, some other requirements may be impacted. For example, run time requirements depend on the IT Environment as well as the implemented model.
7.2	7.2 Model Development		
7.2.1	Development Effort	The model development shall be completed within required time, effort, and development cost targets.	
7.2.2	Development Environment	The system shall provide a model development studio or development environment of type [Development Environment].	
7.2.3	Model Developer Persona	The developer of the executable model shall have the background and capabilities indicated by the [Model Developer Persona].	This is not a requirement on Model System, but on Developer.
7.3	7.3 Model Maintainability		
7.3.1	Maintenance Effort	The model maintenance shall be completed within required time, effort, and maintenance cost targets, by a maintainer of type [Maintainer Persona], over the designated design life of the model.	
7.3.2	Maintenance Environment	The system shall provide a model maintenance studio or maintenance environment of type [Maintenance Environment].	
7.3.3	Model Maintainer Persona	The Maintainer of the executable model shall have the background and capabilities indicated by the [Model Maintainer Persona].	

Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
7	7. Model Life Cycle Management		
7.4	7.4 Model Deployability		
7.4.1	Deployment Cycles	The model deployment cycles shall be completed within required time, effort, and deployment cost targets, by a deployer of type [Deployer Persona].	
7.4.2	Deployment Environment	The system shall provide a model deployment environment of type [Deployment Environment].	
7.4.3	Model Deployer Persona	The deployer of the executable model shall have the background and capabilities indicated by the [Model Deployer Persona].	
7.5	7.5 Model Retirement		
7.5.1	Retirement	The system shall provide retirement of the model, or model versions, from service, on an announced and scheduled basis, within the cost requirements listed.	
7.5.2	Retirement Impact	The system shall complete retirement of the model or model versions without compromise to surviving information security requirements.	
7.6	7.6 VVUQ Pattern Learning		
7.6.1	Other Model Requirement(s)	<Insert other Model Requirement(s) not covered by this VVUQ Model Requirements Pattern.>	Although the Model VVUQ Requirements Pattern is intended to describe all model requirements, other discoveries may be added here, as improvements to the VVUQ Pattern.
8	8. Applicable Standards		
8.1	Standards Compliance	The model shall satisfy the requirements of [Applicable Standards List].	
8.2	Security Compliance	The model system shall satisfy [Information Security Requirements].	77

Appendix III: Model Credibility Assessment Frameworks (CAFs)



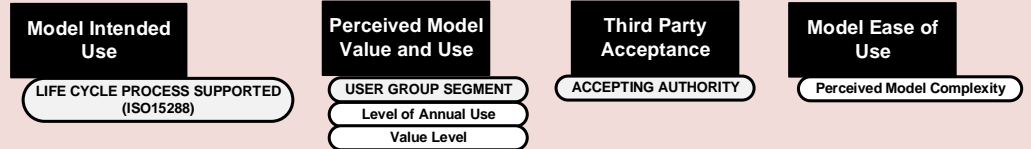
Model Credibility Assessment Feature

- Supports use of a Credibility Assessment Framework (CAF) to assess credibility of a model.
- Configurable for use with different types of CAF's.
- Preserves explicit record of basis of assessment.

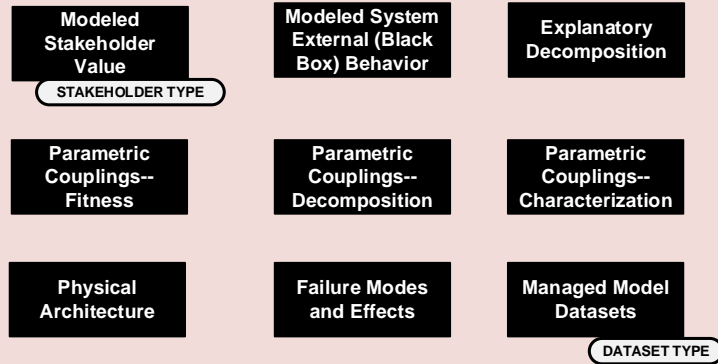
Model Identity and Focus



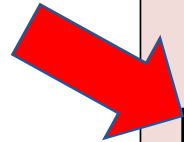
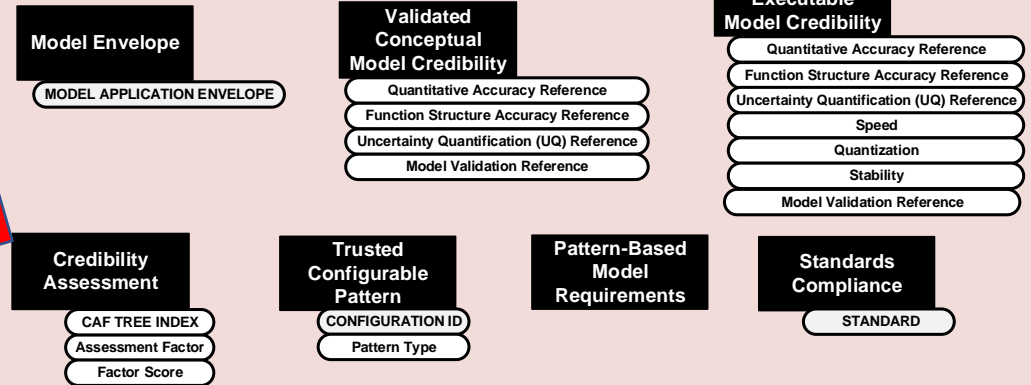
Model Utility



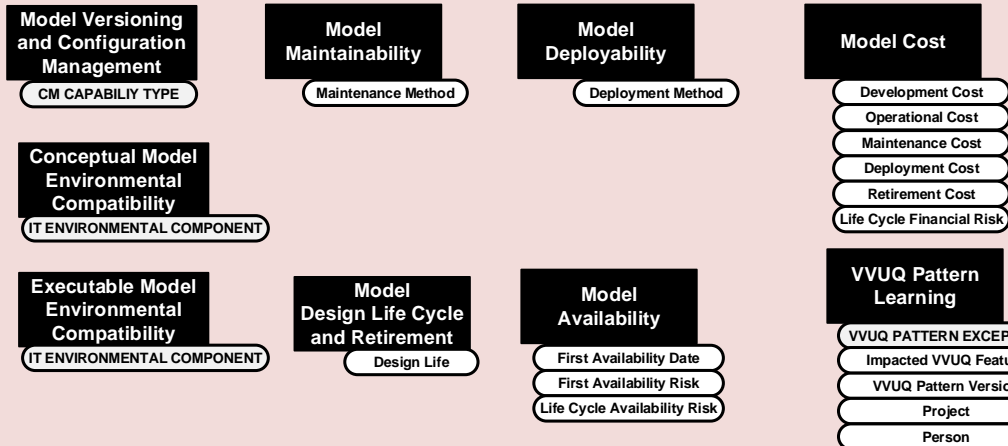
Model Scope and Content



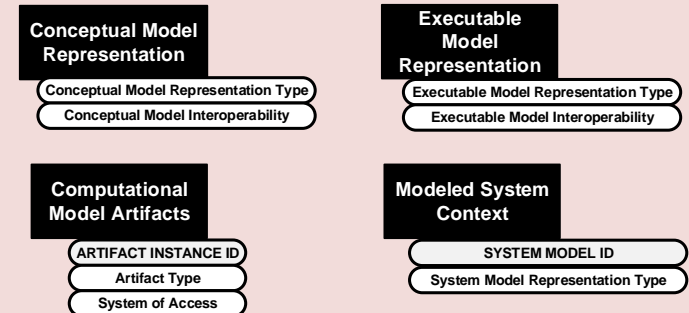
Model Credibility



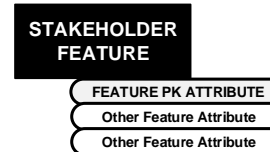
Model Life Cycle Management



Model Representation



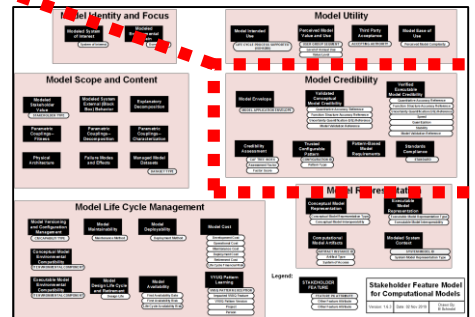
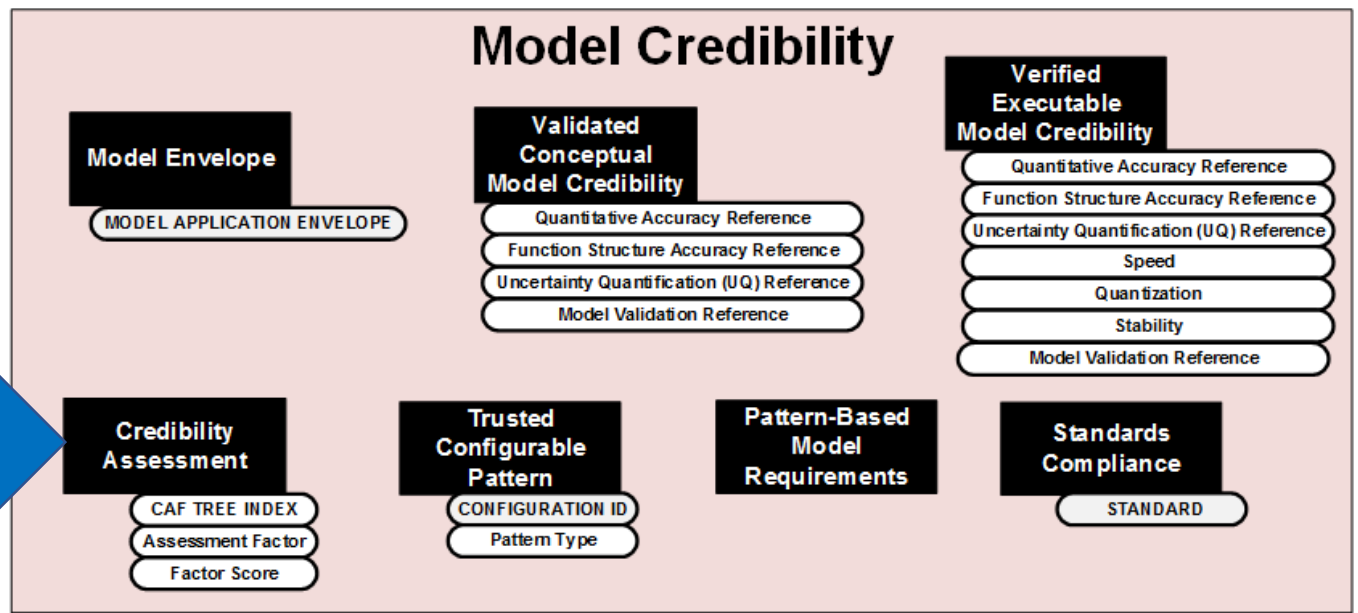
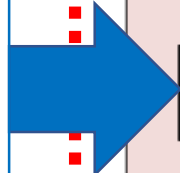
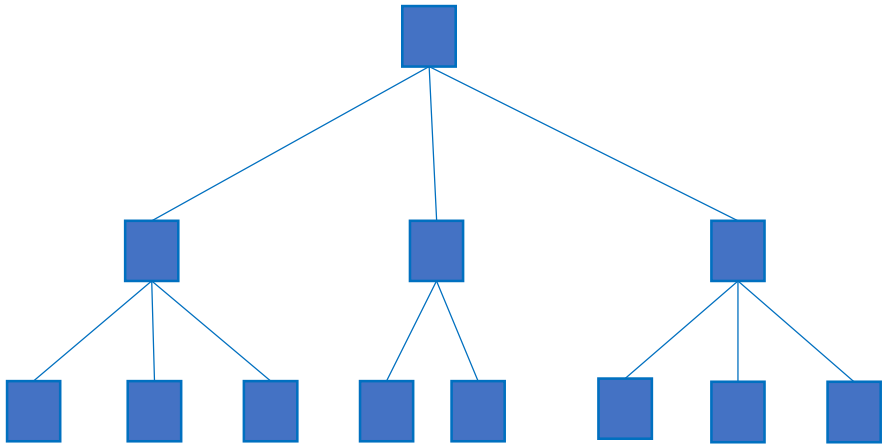
Legend:



Stakeholder Feature Model for Computational Models

Version: 1.6.3 | Date: 02 Nov 2019 | Drawn By: B Schindel

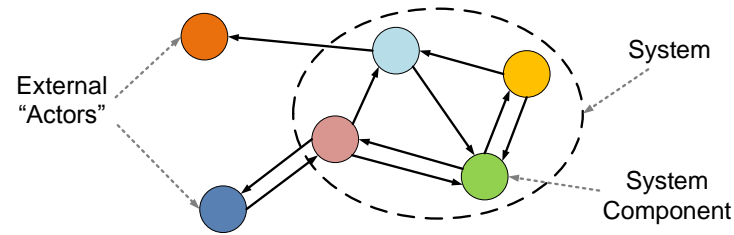
Credibility Assessment Framework



Definitions

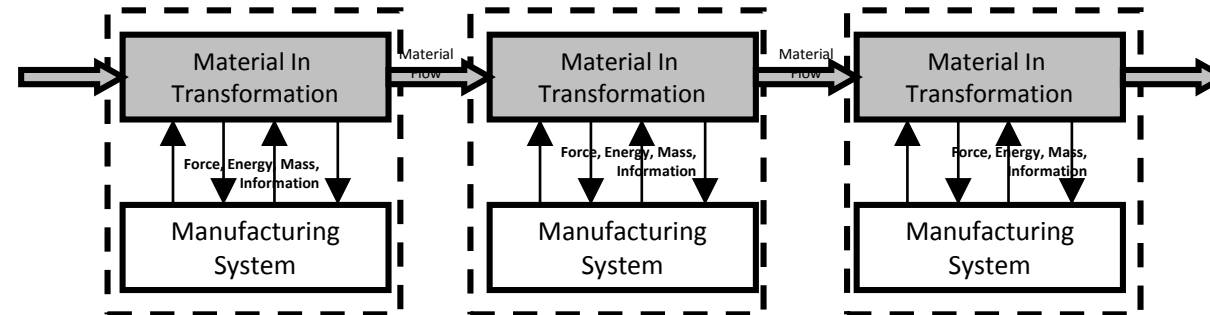
Feature Container	Feature Superclass	Feature Name	Config Rule Ref for Population	Feature Definition	Feature Attribute	PK	Attribute Definition
Model Credibility		Credibility Assessment		The assessed credibility of a model, based upon a credibility assessment framework.	CAF Tree Index	X	An index indicating the position of a credibility assessment factor, in a tree of roll-up credibility assessment factors, in the form a.b.c.d.
Model Credibility		Credibility Assessment		The assessed credibility of a model, based upon a credibility assessment framework.	Assessment Factor		The name of a credibility assessment factor in a credibility assessment framework.
Model Credibility		Credibility Assessment		The assessed credibility of a model, based upon a credibility assessment framework.	Factor Score		The score value awarded to a credibility assessment factor for a particular model

- A System is a set of interacting components:
 - By “interact”, we mean exchanging energy, forces, mass flows, or information, resulting in changes of state:

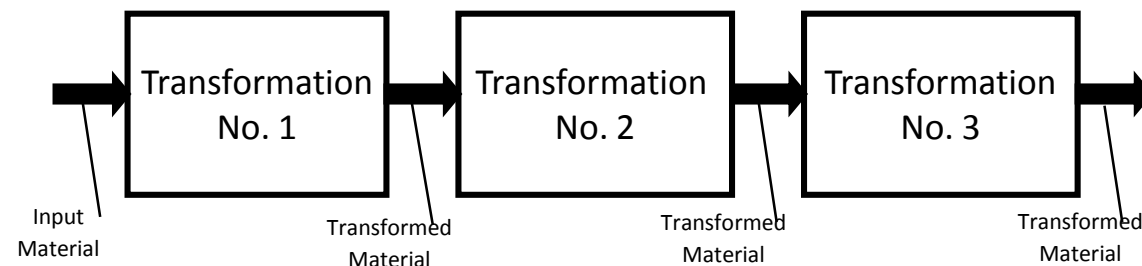


- So, a (Manufacturing or other) Process is a type of System.

- “White Box” view of a system sees its internal interactions.

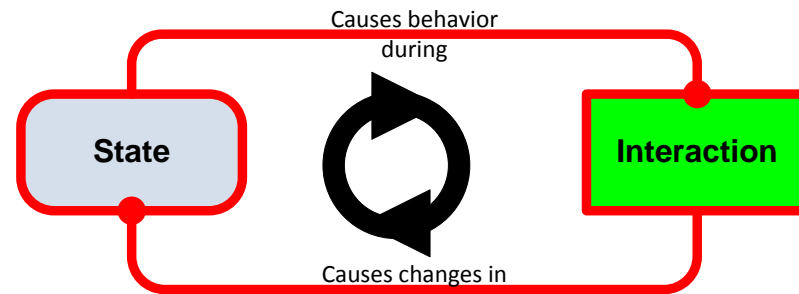
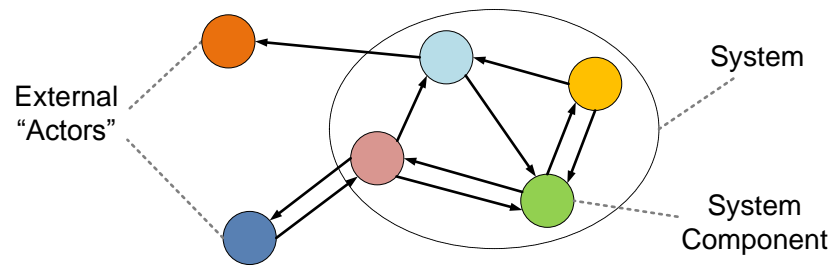


- “Black Box” view of a system sees only its external behavior.



The System Phenomenon

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

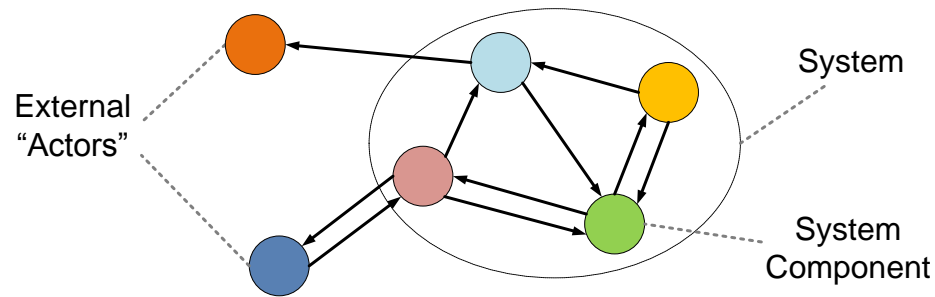


- Where interaction involves the exchange of energy, force, mass, or information, . . .
- Through which one component impacts the state of another component, . . .
- And in which the state of a component impacts its behavior in future interactions.

The System Phenomenon

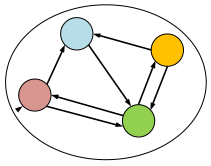
- Phenomena of the hard sciences are in each case instances of the following “System Phenomenon”:
 - *behavior emergent from the interaction of behaviors (phenomena themselves) a level of decomposition lower.*
- For each such phenomena¹, the emergent interaction-based behavior of the larger system is a stationary path of the action integral:

$$\mathcal{S} = \int_{t_1}^{t_2} L(x, \dot{x}, t) dt$$



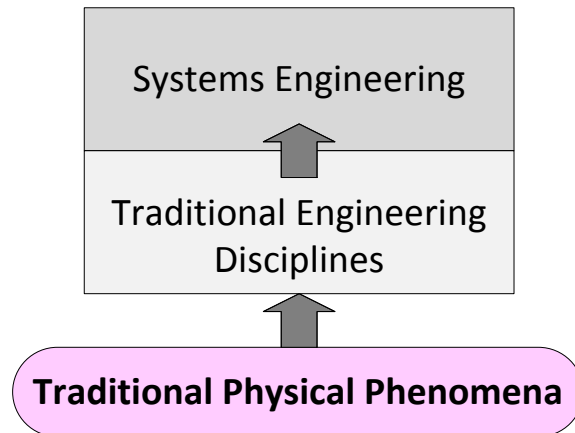
- Reduced to simplest forms, the resulting equations of motion (or if not solvable, empirically observed paths) provide “physical laws” subject to scientific verification—an amazing foundation across all phenomena.

(1) When stated with rigor, special cases for non-holonomic constraints, irreversible dynamics, discrete systems, data systems, etc., led to alternatives to the variational Hamilton's Principle—but the interaction-based structure of the System Phenomenon remained, and the underlying related Action and Symmetry principles became the basis of modern theoretical physics.

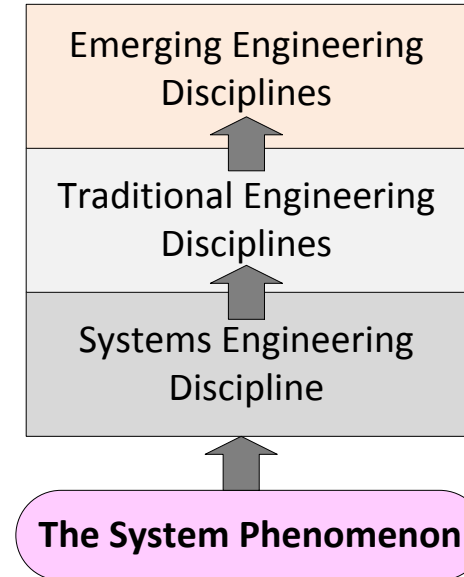


The System Phenomenon

A traditional view:



Our view:



- It is not Systems Engineering that lacks its own phenomenological foundation—instead, the System Phenomenon has been providing the foundation for all the other disciplines all alone!

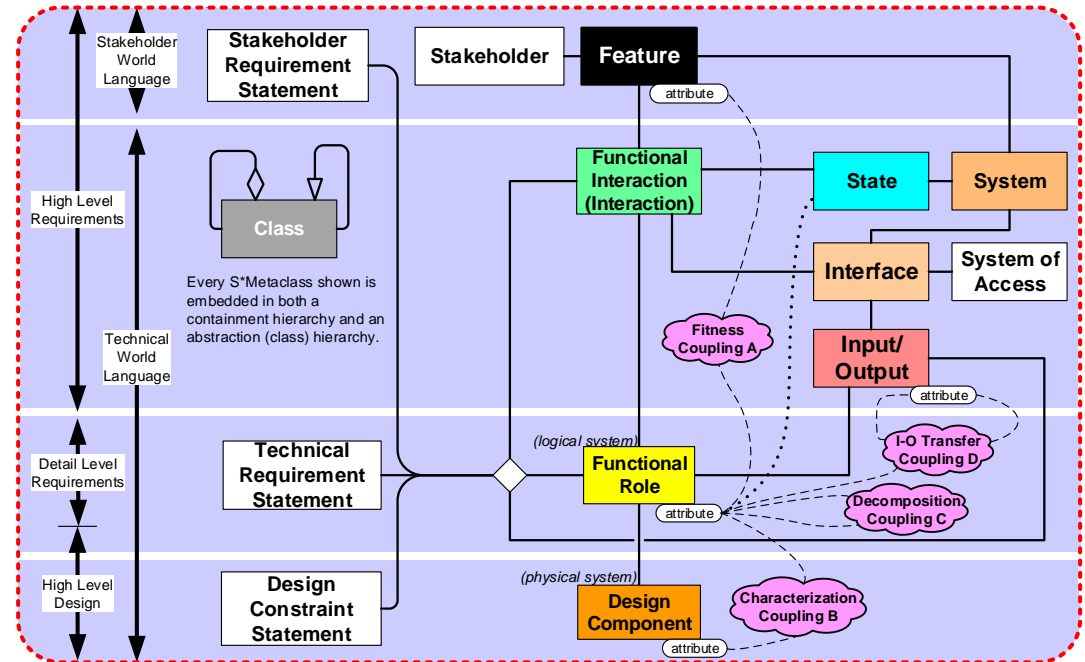
Representing System Patterns: The S* Metamodel Framework

- What is the smallest amount of information we need to represent pattern regularities?
 - Some people have used prose to describe system regularities.
 - This is better than nothing, but usually not enough to deal with the spectrum of issues in complex systems.
- We use S* Models, which are the minimum model-based information necessary:
 - This is not a matter of modeling language—your current favorite language and tools can readily be used for S* Models.
 - The minimum underlying information classes are summarized in the S* Metamodel, for use in any modeling language.
- The resulting system model is made configurable and reusable, thereby becoming an S* Pattern.

Representing System Patterns: The S* Metamodel Framework

- A metamodel is a model of other models;
 - Sets forth how we will represent Requirements, Designs, Verification, Failure Analysis, Trade-offs, etc.;
 - We utilize the (language independent) S* Metamodel from Systematica[®] Methodology:

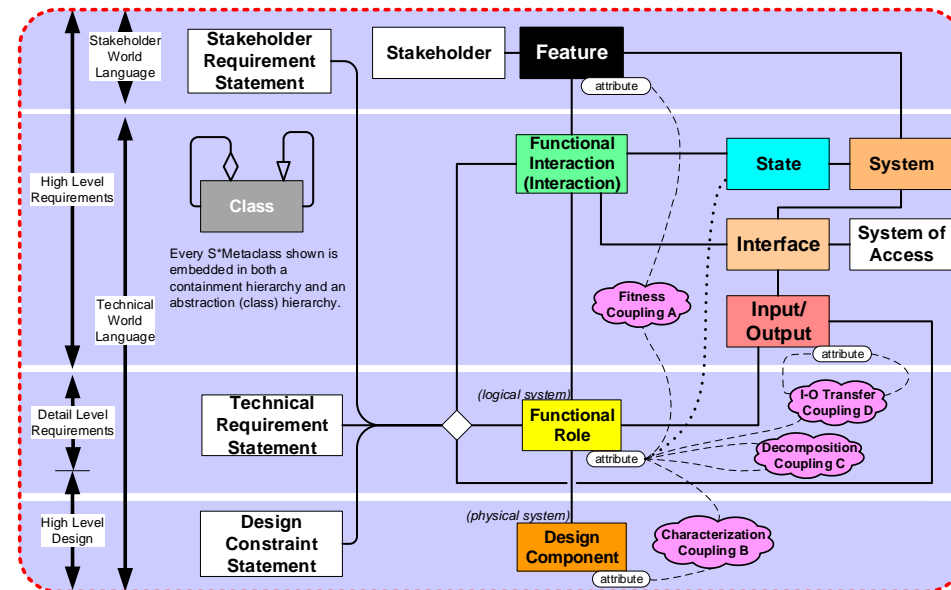
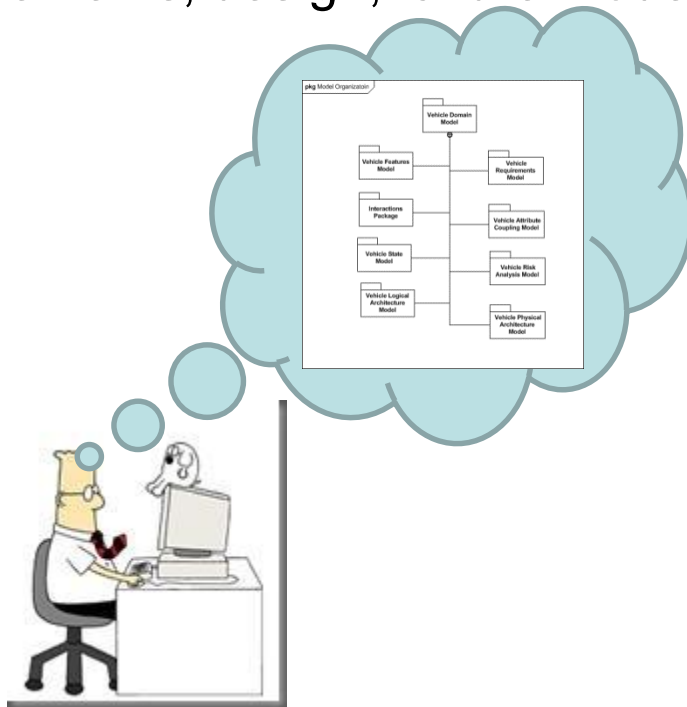
- The resulting system models may be expressed in a wide variety of third party COTS and enterprise information systems, based on S*Metamodel mappings to those environments.
- Has been applied to systems engineering in aerospace, transportation, medical, advanced manufacturing, communication, construction, other domains.



S*Metamodel informal summary pedagogical diagram
(formal S*Metamodel includes additional details.)

Taking advantage of Model-Based Systems Engineering (MBSE)

- **An S* Model** is any model conforming to the S*Metamodel.
- Typically expressed in the “views” of some modeling language or modeling conventions (e.g., mathematical ODE/PDEs, SysML™, free body diagram, etc.)—can be mapped into any third party COTS tool
- The S* Metamodel: The smallest set of model information sufficient to describe a system for purposes of engineering or science, over the system’s life cycle.
- Includes not only the physical Platform information, but all the extended system information (e.g., requirements, design, failure modes & risk analysis, design trade-offs & alternatives, decisions, etc.):



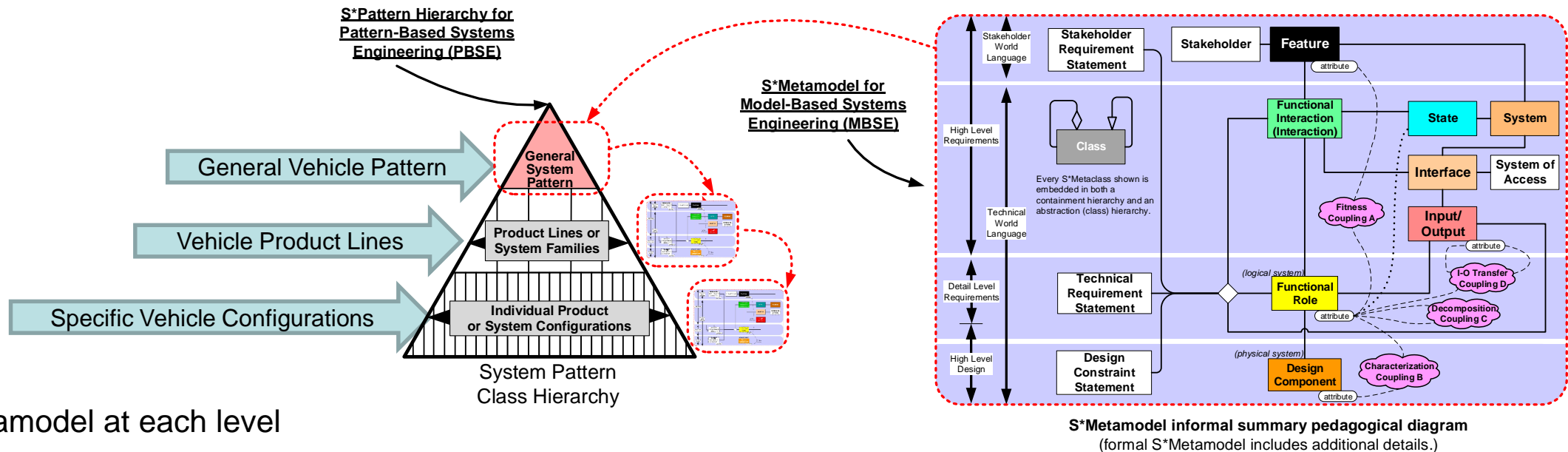
S*Metamodel informal summary pedagogical diagram
(formal S*Metamodel includes additional details.)

Over two decades of S*Model and S*Patterns practice, experience using S*Metamodel

Medical Devices Patterns	Construction Equipment Patterns	Commercial Vehicle Patterns	Space Tourism Pattern
Manufacturing Process Patterns	Vision System Patterns	Packaging Systems Patterns	Lawnmower Product Line Pattern
Embedded Intelligence Patterns	Systems of Innovation (SOI) Pattern	Consumer Packaged Goods Patterns (Multiple)	Orbital Satellite Pattern
Product Service System Patterns	Product Distribution System Patterns	Plant Operations & Maintenance System Patterns	Oil Filter Pattern
Life Cycle Management System Patterns	Production Material Handling Patterns	Engine Controls Patterns	Military Radio Systems Pattern
Agile Systems Engineering Life Cycle Pattern	Transmission Systems Pattern	Precision Parts Production, Sales, and Engineering Pattern	Higher Education Experiential Pattern

Extending the Concept to Patterns, and Pattern-Based Systems Engineering (PBSE)

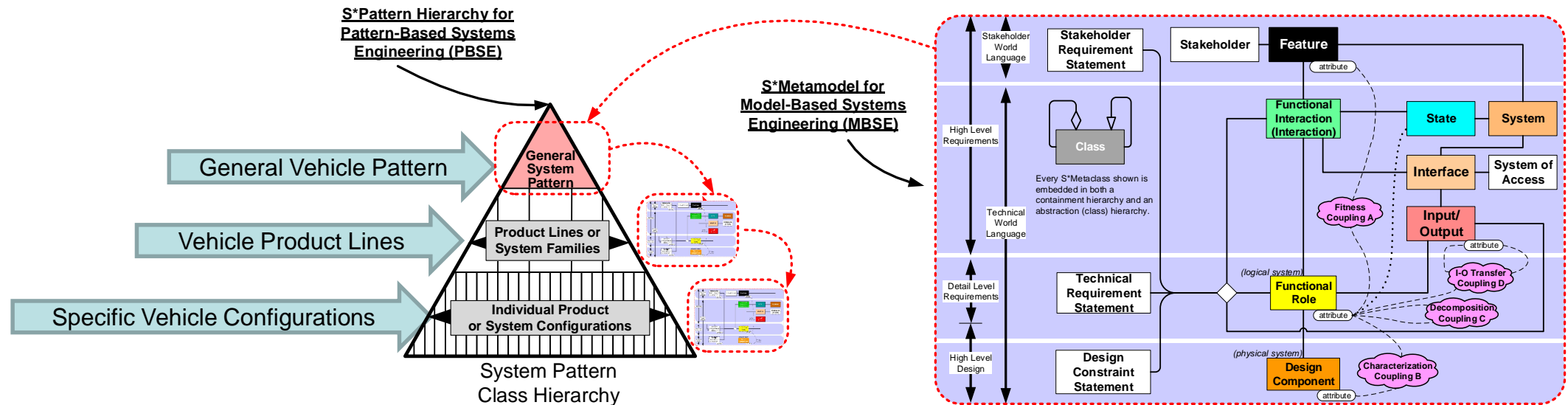
- An **S* Pattern** is a configurable, re-usable S* Model. It is an extension of the idea of a Platform (which is a configurable, re-usable design) or Enterprise / Industry Framework.
- The Pattern includes not only the physical Platform information, but all the extended system information (e.g., requirements, design, failure modes & risk analysis, design trade-offs & alternatives, decisions, etc.):



Same S*Metamodel at each level

Concept Summary: Pattern-Based Systems Engineering (PBSE)

- By including the appropriate S* Metamodel concepts, these can readily be managed in preferred modeling languages and tools—the ideas involved here are not specific to a modeling language or specific tool.
- The order-of-magnitude changes have been realized because projects that use PBSE rapidly start from an existing Pattern, gaining the advantages of its content, and feed the pattern with what they learn, for future users.
- The “game changer” here is the shift from “learning to model” to “learning the model”, freeing many people to rapidly configure, specialize, and apply patterns to deliver value in their model-based projects.

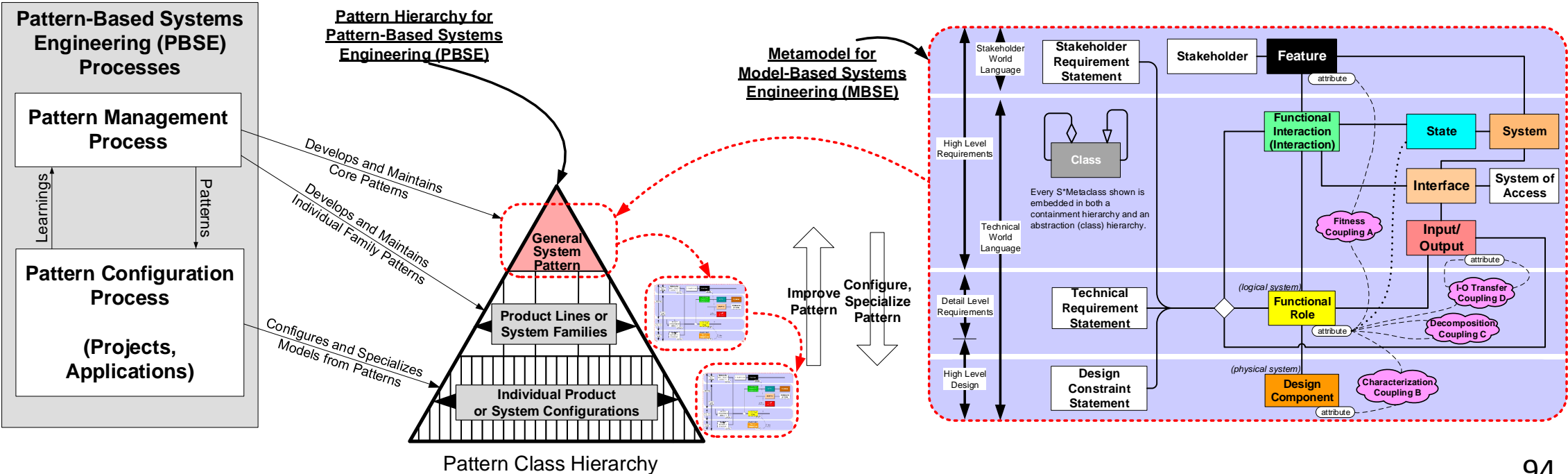


Same S*Metamodel at each level

S*Metamodel informal summary pedagogical diagram
(formal S*Metamodel includes additional details.)

Concept Summary: Pattern-Based Systems Engineering (PBSE)

- PBSE provides a specific technical method for implementing:
 - Platform Management and Product Line Engineering (PLE)
 - Enterprise or Industry Frameworks
 - System Standards
 - Trusted Experience Accumulation for Systems of Innovation
 - Lean Product Development & IP Asset Re-use



Definitions of Some S* Metamodel Classes

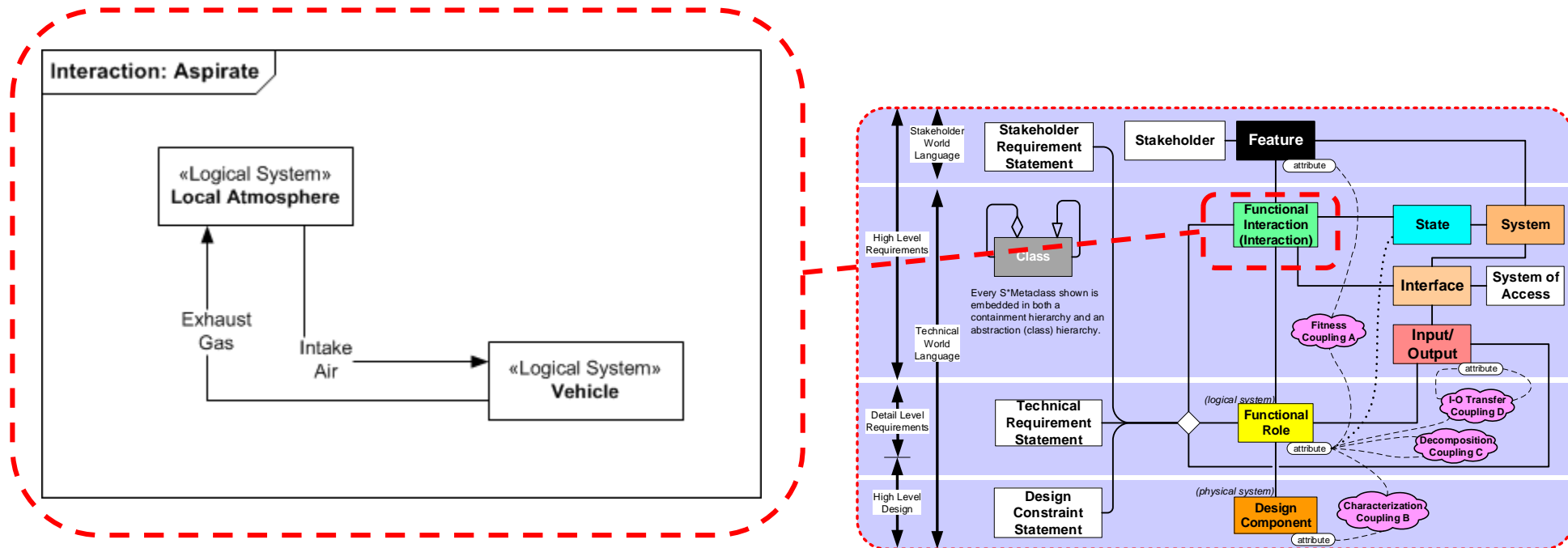
- **System**: A collection of interacting components. Example: Medical Device; Hospital Domain, Health Care Delivery System Domain.
- **Stakeholder**: A person or other entity with something at stake in the life cycle of a system. Example: Patient; Health Care Provider; Enterprise Shareholder
- **Feature**: A behavior of a system that carries stakeholder value. Example: Automatic Infusion Feature; Patient Safety Features; Device Connectivity Features
- **Functional Interaction (Interaction)**: An exchange of energy, force, mass, or information by two entities, in which one changes the state of the other. Example: Deliver Infusion; Transmit Shock and Vibration
- **Functional Role (Role)**: The behavior performed by one of the interacting entities during an Interaction; identified only by its externally visible behavior during interaction. Example: Patient; Device Operator; Injectable Storage Subsystem
- **Input-Output**: That which is exchanged during an interaction (generally associated with energy, force, material, or information). Example: Injected Material, Pressure, Status Signal

Definitions of some S* Metamodel Classes

- **System of Access**: A system which provides the means for physical interaction between two interacting entities. Examples: Control Button; Status Indicator; Temperature Sensor; Drive Actuator; Catheter; Tube Fitting; Beeper
- **Interface**: The association of a System (which “has” the interface), one or more Interactions (which describe behavior at the interface), the Input-Outputs (which pass through the interface), and a System of Access (which provides the means of the interaction). Examples: Injection Interface; Device Control Interface
- **State**: A mode, situation, or condition that describes a System’s condition at some moment or period of time. Example: Device Off; Starting Up; Loading; Performing Injection; Diagnosing Failure; Shutting Down
- **Design Component**: A physical entity that has identity, whose behavior is described by Functional Role(s) allocated to it. Examples: 316 L Stainless Steel; Sodium Chloride; Model 300 Infusion Pump; Department 516 Laboratory
- **Requirement Statement**: A (usually prose) description of the behavior expected of (at least part of) a Functional Role. Example: “The System shall complete any injection cycle within 2 seconds.”

Physical Interactions: At the heart of S* models

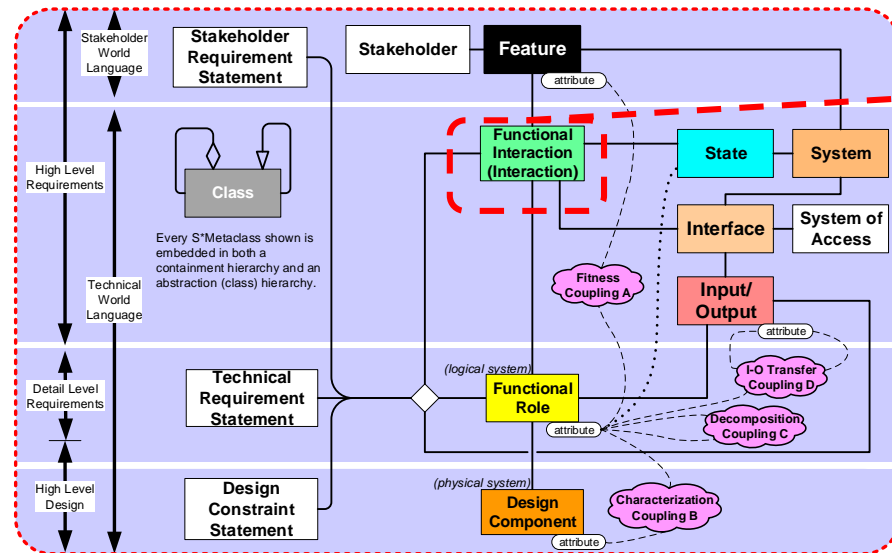
- S* models represent Interactions as explicit objects:
 - Goes to the heart of 300 years of natural science of systems as a foundation for engineering, including emergence.
 - All physical laws of science are about interactions in some way.
 - All functional requirements are revealed as external interactions (!)



S*Metamodel informal summary pedagogical diagram
(formal S*Metamodel includes additional details.)

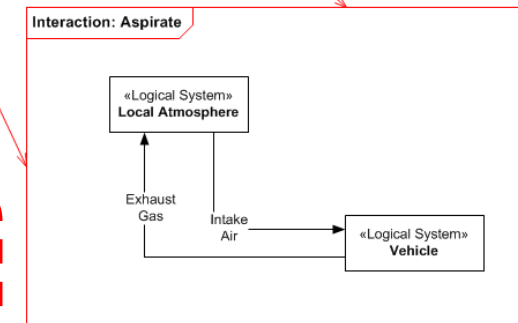
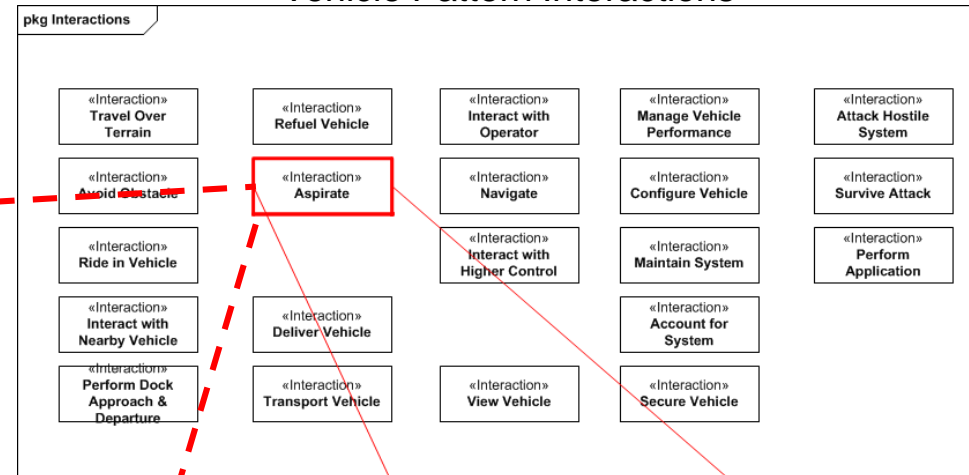
Physical Interactions: At the heart of S* models

- S* models represent Physical Interactions as explicit objects:



S*Metamodel informal summary pedagogical diagram
(formal S*Metamodel includes additional details.)

Vehicle Pattern Interactions



Interaction Diagram

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.

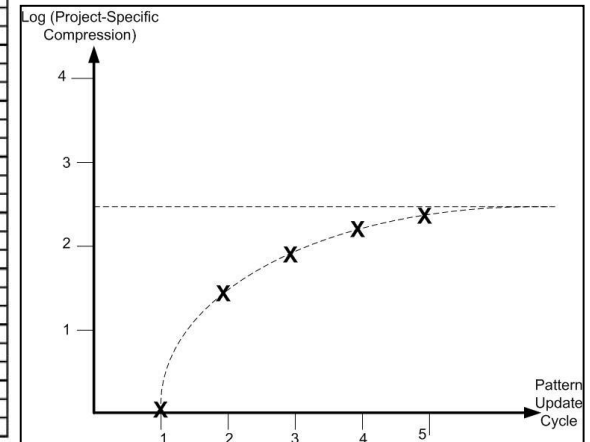
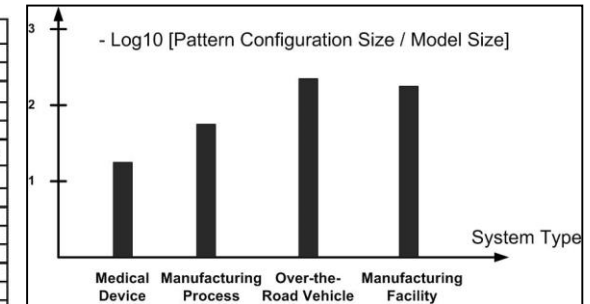
Pattern-based systems engineering (PBSE)

- Model-based Patterns:
 - In this approach, Patterns are reusable, configurable S* models of families (product lines, sets, ensembles) of systems.
 - A Pattern is not just the physical product family—it includes its behavior, decomposition structure, failure modes, and other aspects of its model.
- These Patterns are ready to be configured to serve as Models of individual systems in projects.
- Configured here is specifically limited to mean that:
 - Pattern model components are populated / de-populated, and
 - Pattern model attribute (parameter) values are set
 - both based on Configuration Rules that are part of the Pattern.
- S*Patterns based on the same S*Metamodel as S*Models.

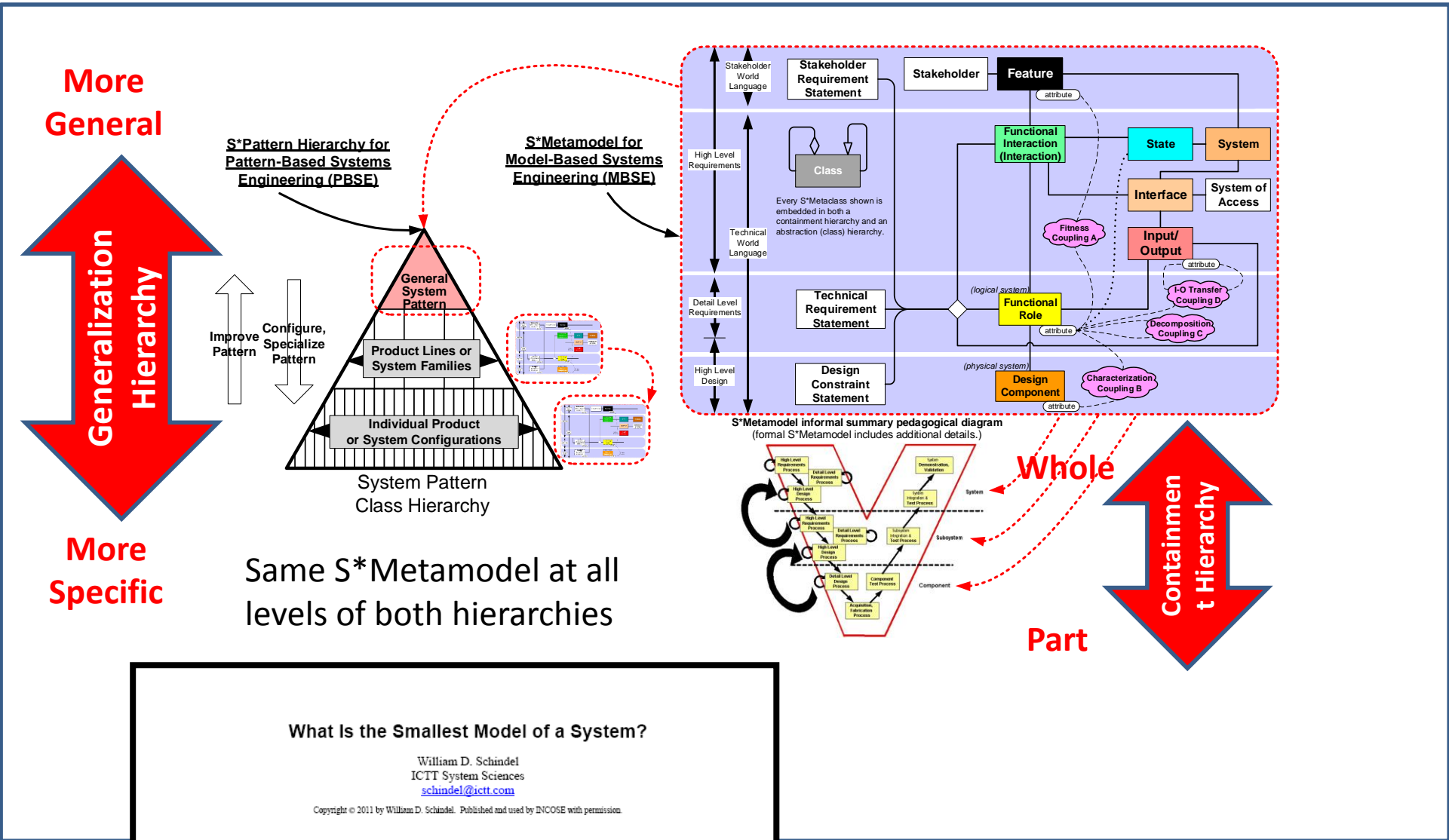
Pattern configurations

- A table of configurations illustrates how patterns facilitate compression;
- Each column in the table is a compressed system representation with respect to (“modulo”) the pattern;
- The compression is typically very large;
- The compression ratio tells us how much of the pattern is variable and how much fixed, across the family of potential configurations.

Lawnmower Product Line: Configurations Table									
		Units	Walk-Behind	Walk-Behind	Walk-Behind	Riding	Riding	Riding Mower	Autonomous
			Push Mower	Mower	Self-Propelled	Rider	Tractor	Tractor	Autonomous
			Push Mower	Self-Propelled	Wide Cut	Rider	Lawn	Garden	Auto Mower
	Model Number		M3	M5	M11	M17	M19	M23	M100
	Market Segment		Sm Resident	Med Resident	Med Resident	Lg Resident	Lg Resident	Home Garden	High End Suburban
Power	Engine Manufacturer		B&S	B&S	Tecumseh	Tecumseh	Kohler	Kohler	Elektroset
	Horsepower	HP	5	6.5	13	16	18.5	22	0.5
Production	Cutting Width	Inches	17	19	36	36	42	48	16
	Maximum Mowing Speed	MPH	3	3	4	8	10	12	2.5
	Maximum Mowing Productivity	Acres/Hr			1.6				
	Turning Radius	Inches	0	0	0	0	126	165	0
	Fuel Tank Capacity	Hours	1.5	1.7	2.5	2.8	3.2	3.5	2
	Towing Feature						x	x	
	Electric Starter Feature				x	x	x	x	
	Basic Mowing Feature Group		x	x	x	x	x	x	x
Mower	No. of Anti-Scalping Rollers		0	0	1	2	4	6	0
	Cutting Height Minimum	Inches	1	1.5	1.5	1.5	1	1.5	1.2
	Cutting Height Maximum	Inches	4	5	5	6	8	10	3.8
	Operator Riding Feature					x	x	x	
	Grass Bagging Feature		Optional	Optional	Optional	Optional	Optional	Optional	
	Mulching Feature		Standard	Factory Installed	Dealer Installed				
	Aerator Feature					Optional	Optional	Optional	
	Autonomous Mowing Feature								x
	Dethatching Feature					Optional	Optional	Optional	
Physical	Wheel Base	Inches	18	20	22	40	48	52	16
	Overall Length	Inches	18	20	23	58	56	68	28.3
	Overall Height	Inches	40	42	42	30	32	36	10.3
	Width	Inches	18	20	22	40	48	52	23.6
	Weight	Pounds	120	160	300	680	705	1020	15.6
	Self-Propelled Mowing Feature			x	x	x	x	x	x
	Automatic TransmFeature							x	
Financials	Retail Price	Dollars	360	460	1800	3300	6100	9990	1799
	Manufacturer Cost	Dollars	120	140	550	950	1800	3500	310
Maintenance	Warranty	Months	12	12	18	24	24	24	12
	Product Service Life	Hours	500	500	600	1100	1350	1500	300
	Time Between Service	Hours	100	100	150	200	200	250	100
Safety	Spark Arrest Feature		x	x	x	x	x	x	



Two entirely different hierarchies are involved:



What Is the Smallest Model of a System?

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

Copyright © 2011 by William D. Schindel. Published and used by INCOSE with permission.

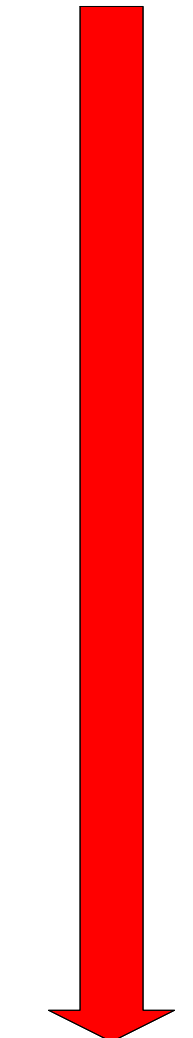
Abstract. How we represent systems is fundamental to the history of mathematics, science, and engineering. Model-based engineering methods shift the nature of representation of systems from historical prose forms to explicit data structures more directly comparable to those of science and mathematics. However, using models does not guarantee simpler representation—indeed a typical fear voiced about models is that they may be too complex.

Minimality of system representations is of both theoretical and practical interest. The mathematical and scientific interest is that the size of a system's "minimal representation" is one definition of its complexity. The practical engineering interest is that the size and redundancy of engineering specifications challenge the effectiveness of systems engineering processes. INCOSE thought leaders have asked how systems work can be made 10:1 simpler

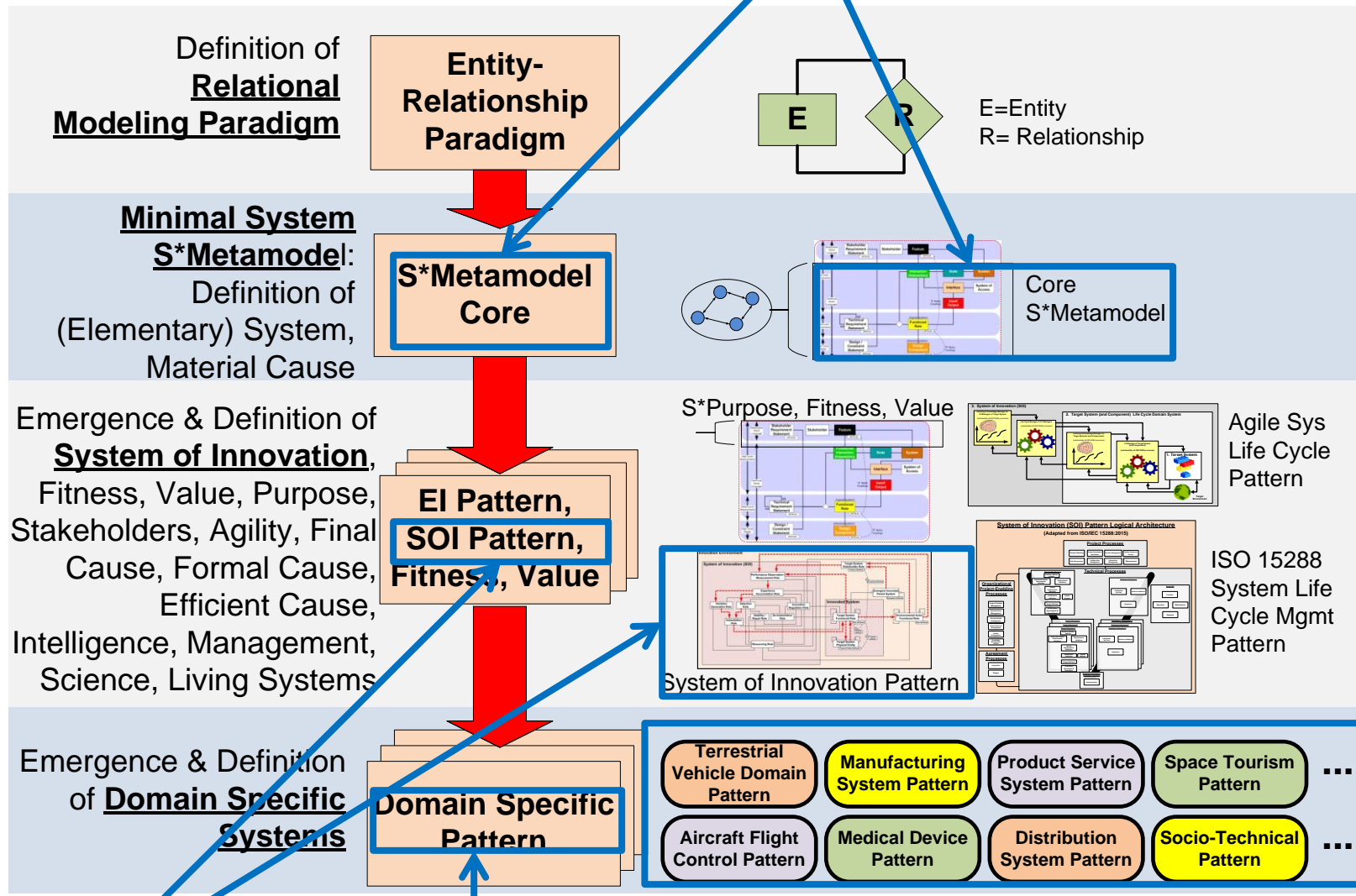
Universal systems nomenclature, domain-independent.

Emergence of Patterns from Patterns: S*Pattern Class Hierarchy

More General



More Specific

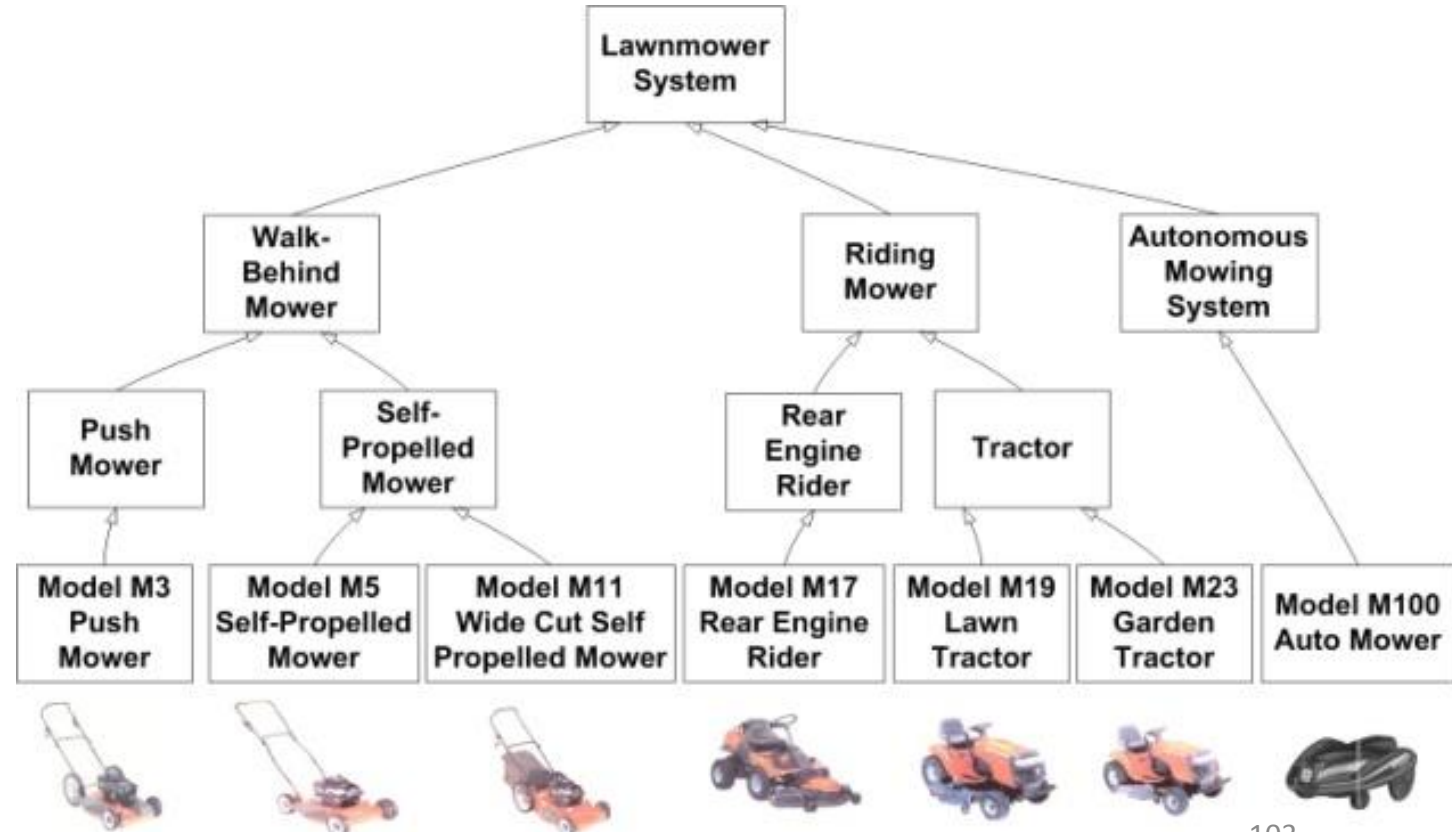
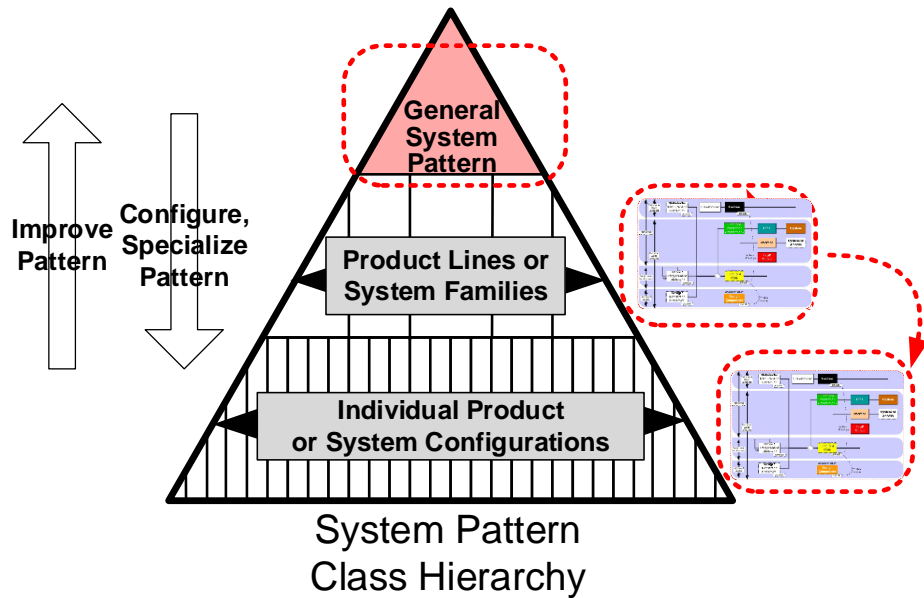


Domain-specific languages, frameworks, ontologies.

Generator of “new systems”; also maintainer, destroyer

S*Models as Configurations of S*Patterns

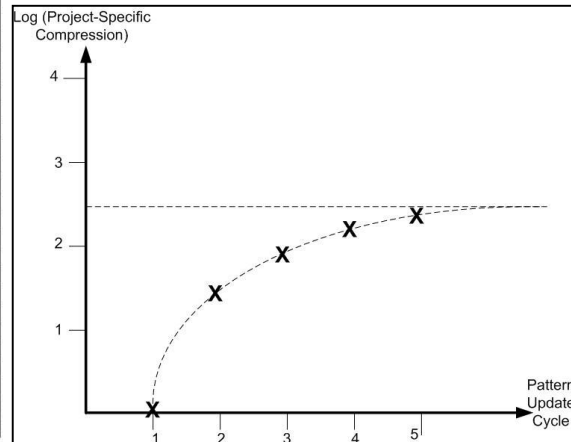
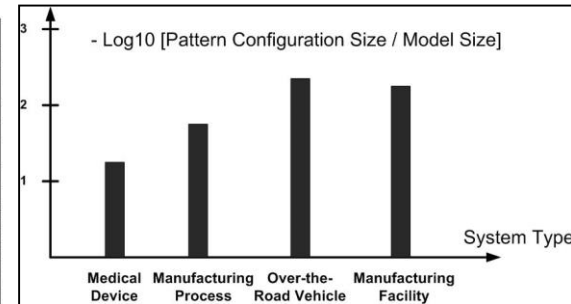
- Patterns as Compression: Lawnmowers; IEEE 802.11



Pattern configurations

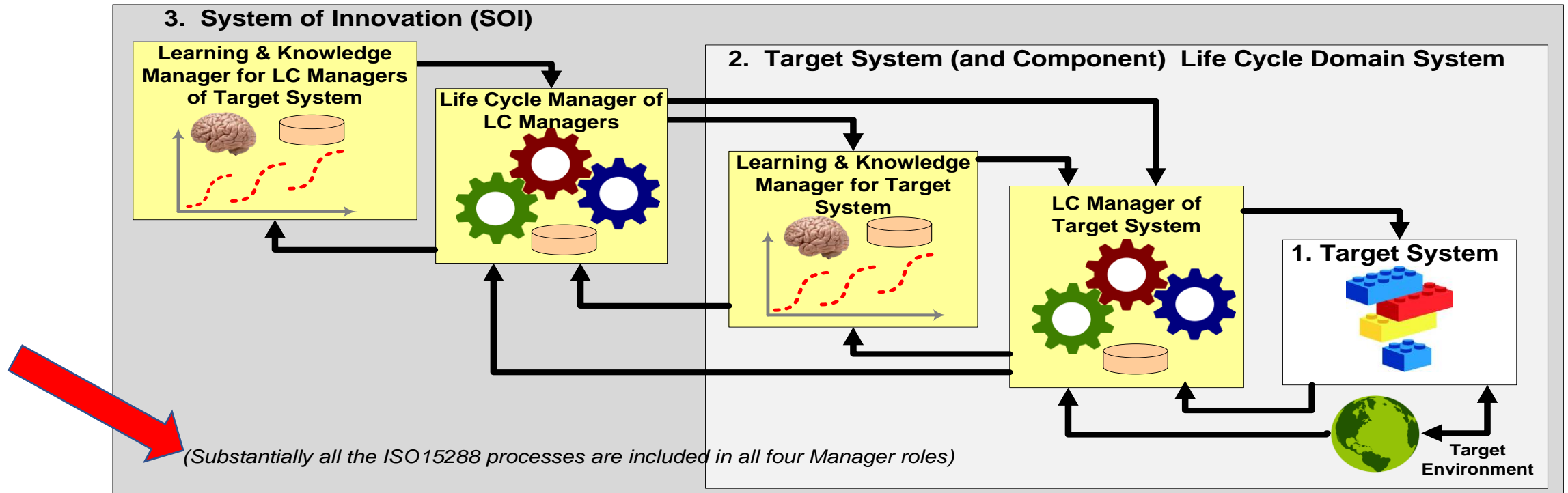
- A table of configurations illustrates how patterns facilitate compression;
- Each column in the table is a compressed system representation with respect to (“modulo”) the pattern;
- The compression is typically very large;
- The compression ratio tells us how much of the pattern is variable and how much fixed, across the family of potential configurations.

Lawnmower Product Line: Configurations Table									
		Units	Walk-Behind	Walk-Behind	Walk-Behind	Riding	Riding	Riding Mower	Autonomous
			Push Mower	Mower	Self-Propelled	Rider	Tractor	Tractor	Autonomous
			Push Mower	Self-Propelled	Wide Cut	Rider	Lawn	Garden	Auto Mower
	Model Number		M3	M5	M11	M17	M19	M23	M100
	Market Segment		Sm Resident	Med Resident	Med Resident	Lg Resident	Lg Resident	Home Garden	High End Suburban
Power	Engine Manufacturer		B&S	B&S	Tecumseh	Tecumseh	Kohler	Kohler	Elektroset
	Horsepower	HP	5	6.5	13	16	18.5	22	0.5
Production	Cutting Width	Inches	17	19	36	36	42	48	16
	Maximum Mowing Speed	MPH	3	3	4	8	10	12	2.5
	Maximum Mowing Productivity	Acres/Hr			1.6				
	Turning Radius	Inches	0	0	0	0	126	165	0
	Fuel Tank Capacity	Hours	1.5	1.7	2.5	2.8	3.2	3.5	2
	Towing Feature						x	x	
	Electric Starter Feature				x	x	x	x	
	Basic Mowing Feature Group		x	x	x	x	x	x	x
Mower	No. of Anti-Scalping Rollers		0	0	1	2	4	6	0
	Cutting Height Minimum	Inches	1	1.5	1.5	1.5	1	1.5	1.2
	Cutting Height Maximum	Inches	4	5	5	6	8	10	3.8
	Operator Riding Feature					x	x	x	
	Grass Bagging Feature		Optional	Optional	Optional	Optional	Optional	Optional	
	Mulching Feature		Standard	Factory Installed	Dealer Installed				
	Aerator Feature					Optional	Optional	Optional	
	Autonomous Mowing Feature								x
	Dethatching Feature					Optional	Optional	Optional	
Physical	Wheel Base	Inches	18	20	22	40	48	52	16
	Overall Length	Inches	18	20	23	58	56	68	28.3
	Overall Height	Inches	40	42	42	30	32	36	10.3
	Width	Inches	18	20	22	40	48	52	23.6
	Weight	Pounds	120	160	300	680	705	1020	15.6
	Self-Propelled Mowing Feature			x	x	x	x	x	x
	Automatic TransmFeature							x	
Financials	Retail Price	Dollars	360	460	1800	3300	6100	9990	1799
	Manufacturer Cost	Dollars	120	140	550	950	1800	3500	310
Maintenance	Warranty	Months	12	12	18	24	24	24	12
	Product Service Life	Hours	500	500	600	1100	1350	1500	300
	Time Between Service	Hours	100	100	150	200	200	250	100
Safety	Spark Arrest Feature		x	x	x	x	x	x	



Appendix V: ASELCM Pattern, Trusted Models, Effective Group Learning

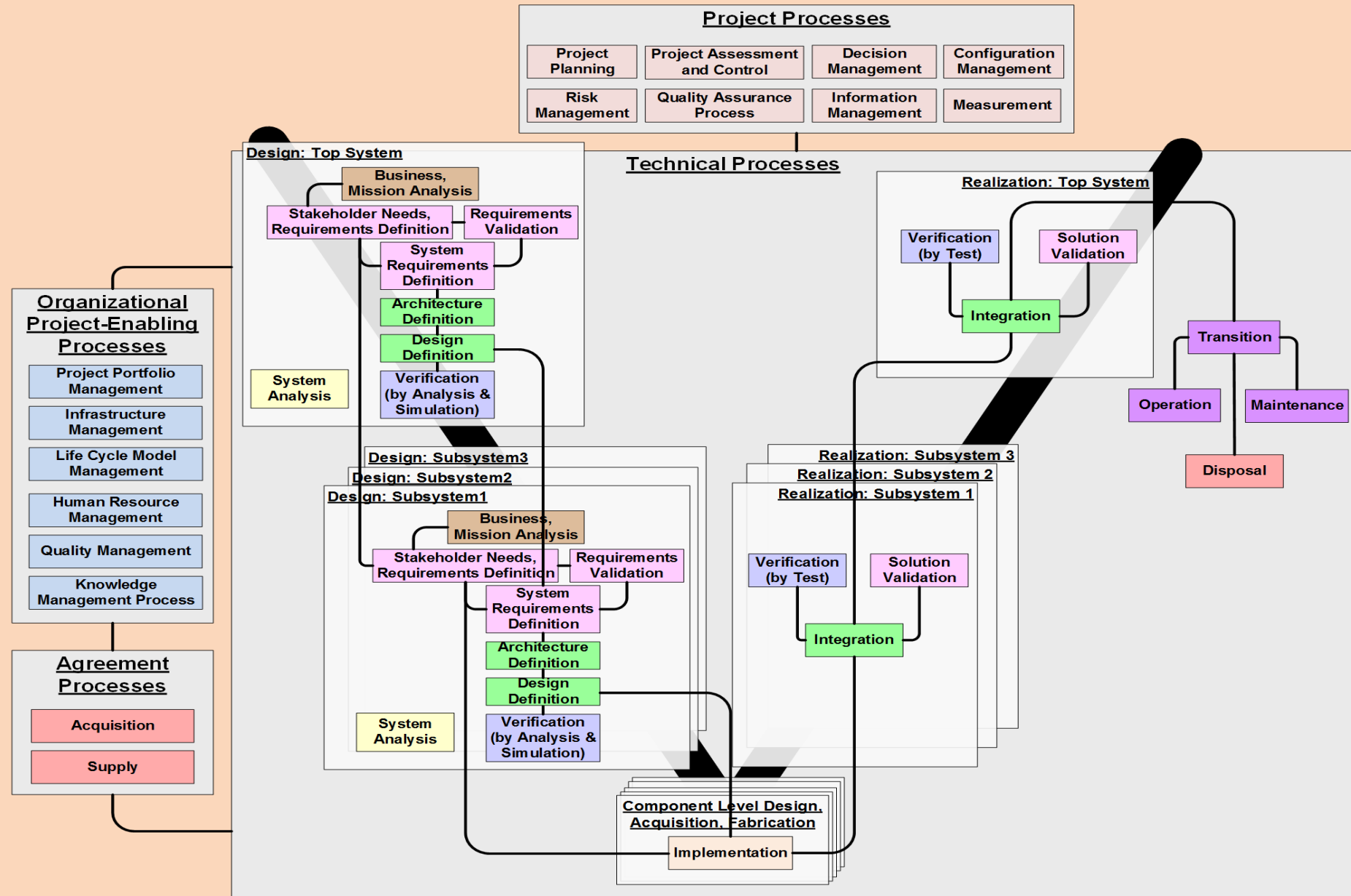
(Used for INCOSE Agile SE Project, INCOSE CIPR WG, etc. Generic innovation reference model: Descriptive, not prescriptive.)



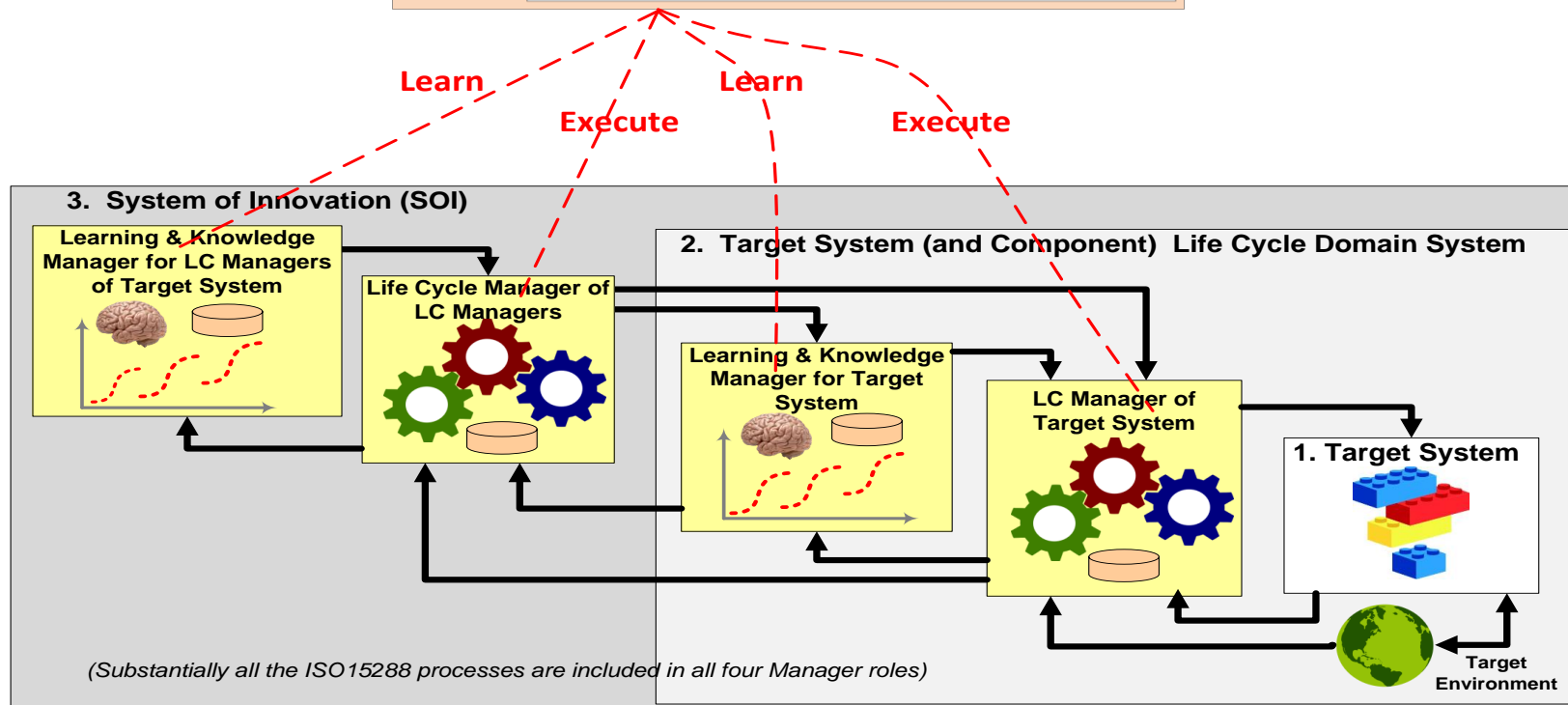
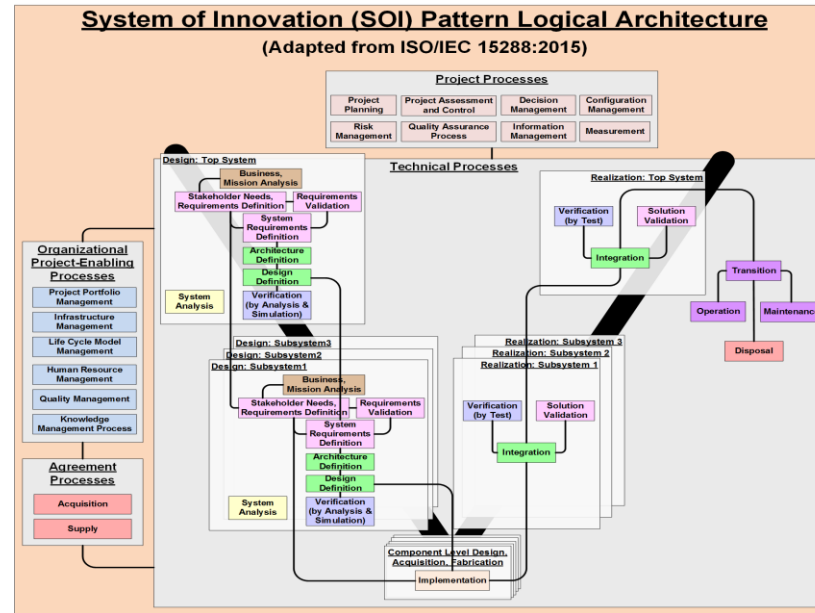
- System 1: Target system of interest, to be engineered or improved.
- System 2: The environment of (interacting with) S1, including all the life cycle management systems of S1, including learning about S1.
- System 3: The life cycle management systems for S2, including learning about S2.

System of Innovation (SOI) Pattern Logical Architecture

(Adapted from ISO/IEC 15288:2015)



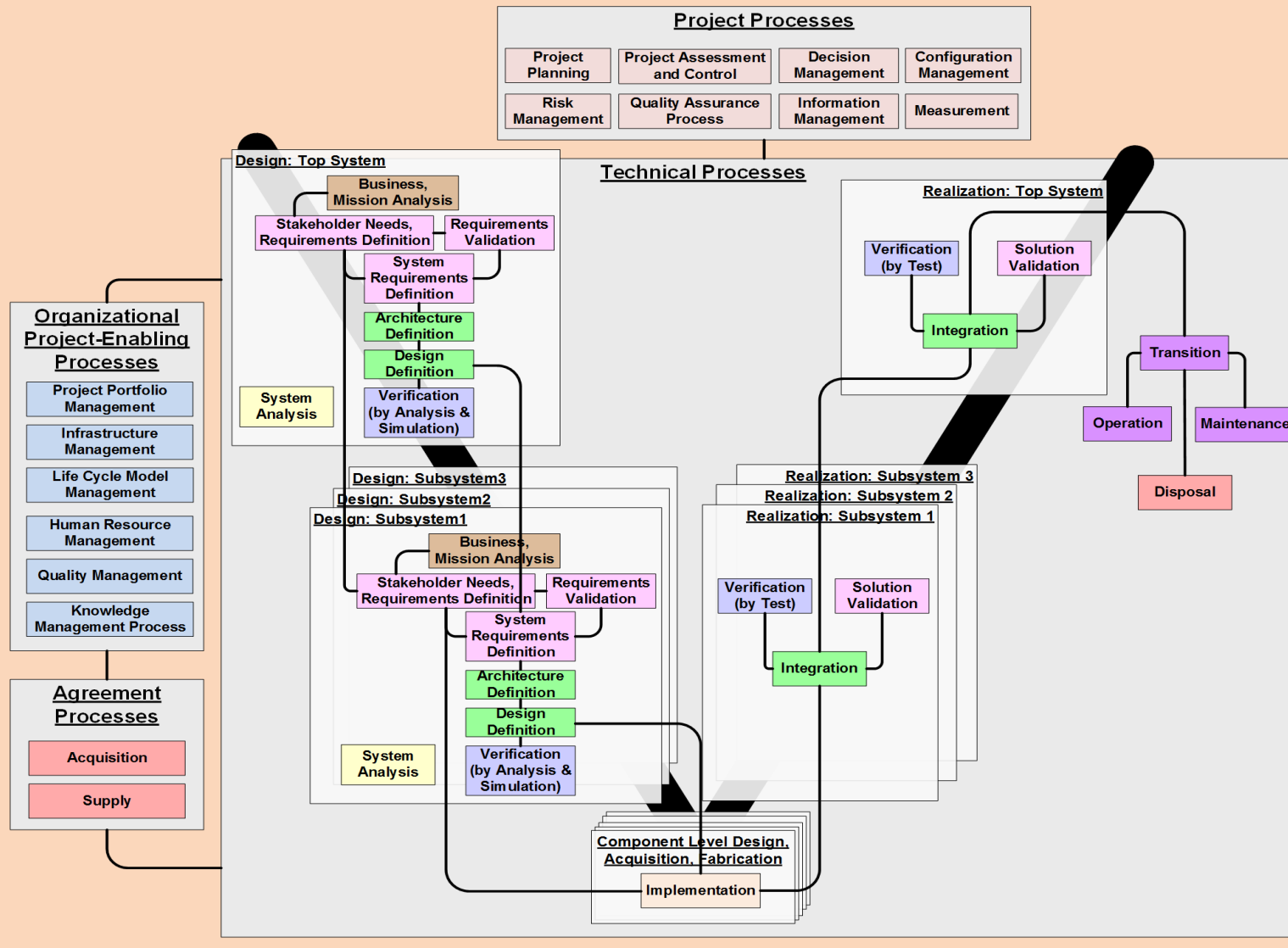
ISO 15288 processes appear 4 times, whether we recognize or not.



Models for what purposes? Possible ISO15288 answers

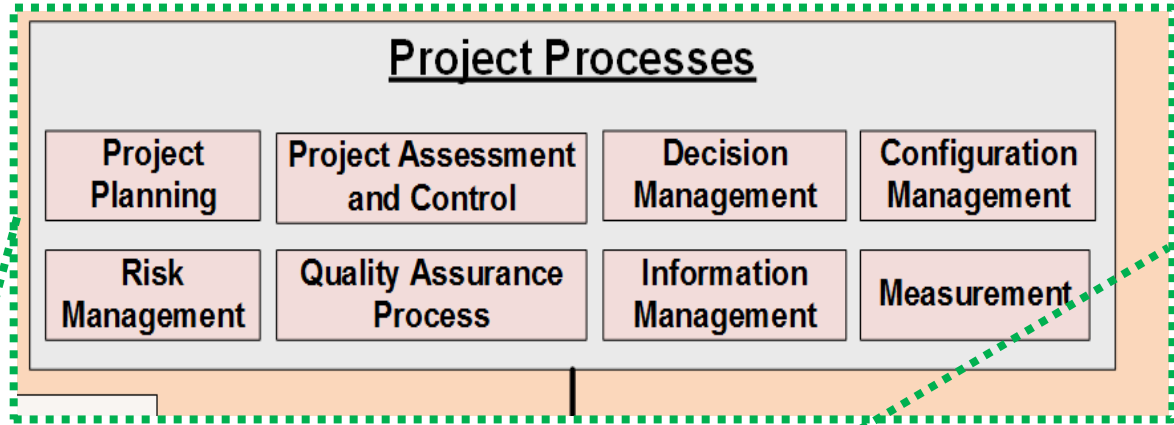
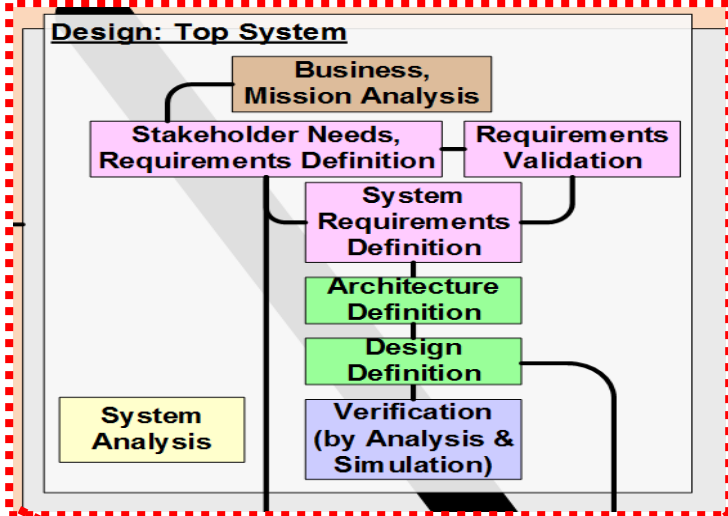
System of Innovation (SOI) Pattern Logical Architecture

(Adapted from ISO/IEC 15288:2015)

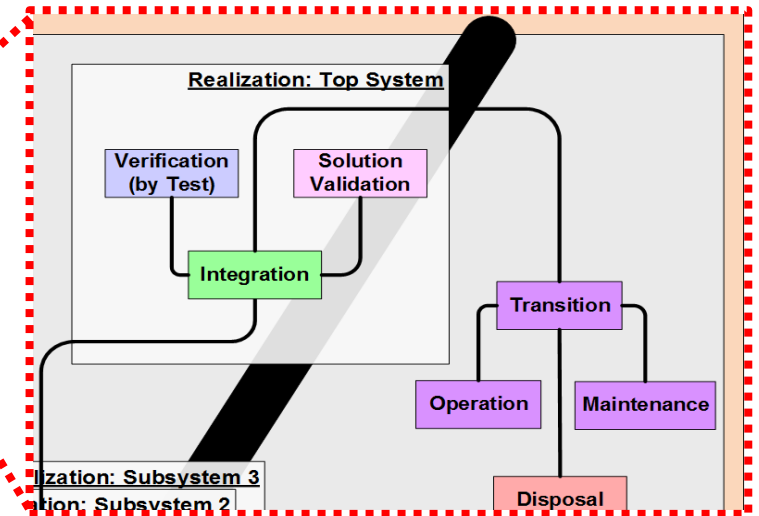
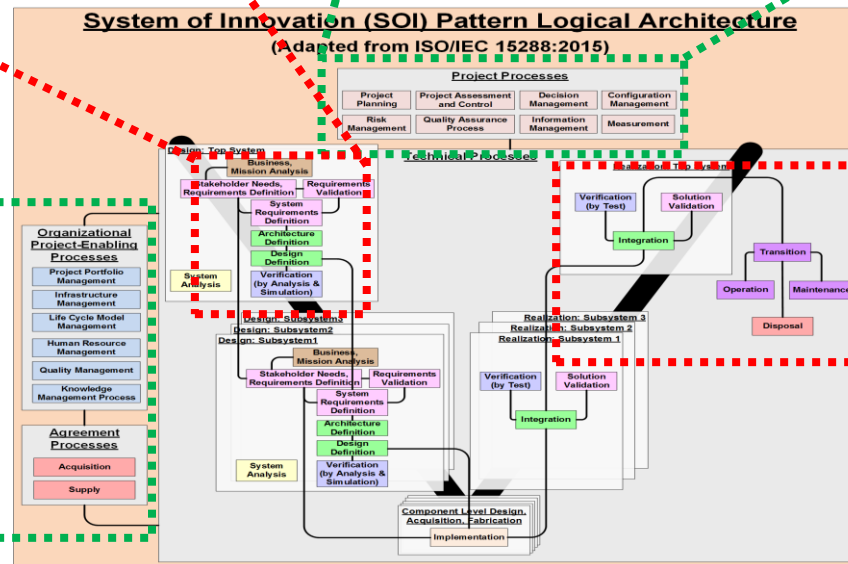
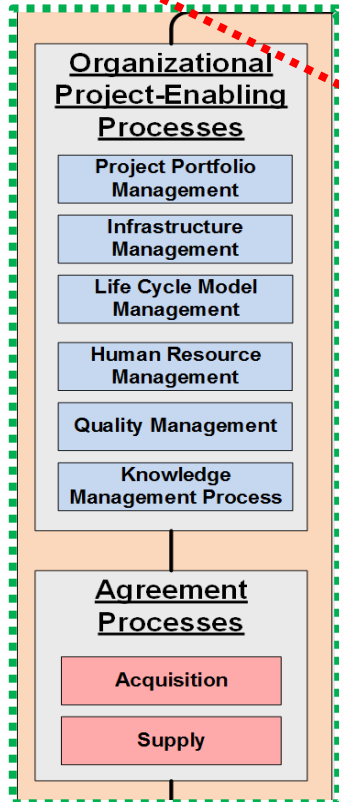


Potentially for any ISO 15288 processes:

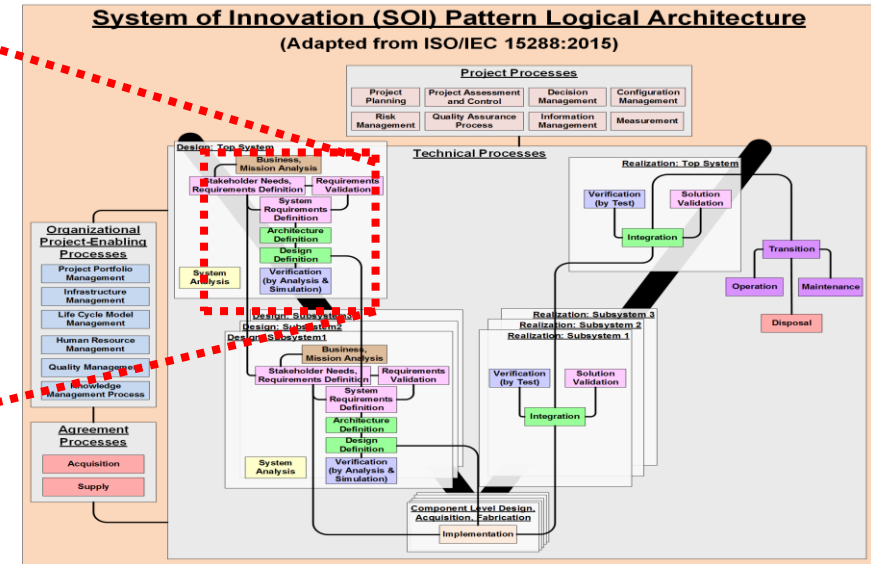
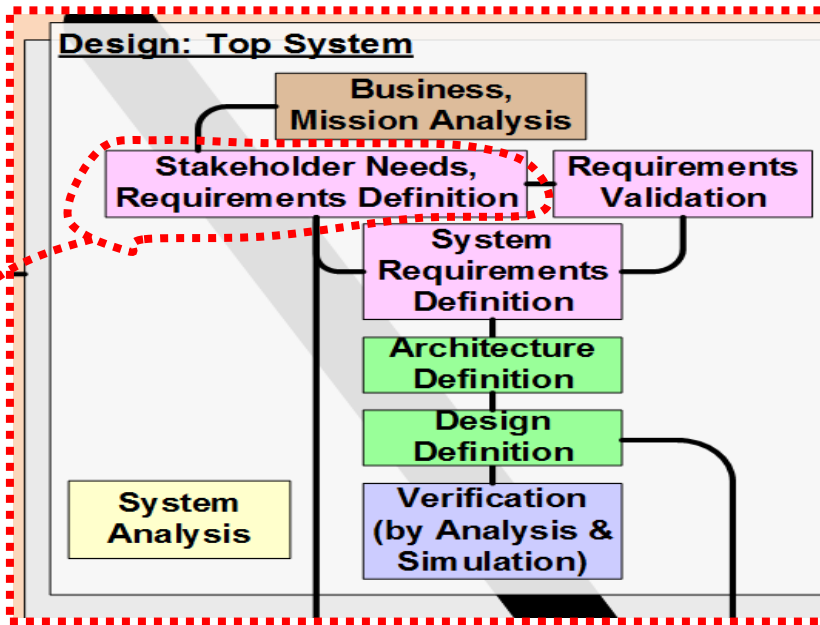
- If there is a net benefit . . .
- Some more obvious than others.
- The INCOSE MBE Transformation is using ISO 15288 framework as an aid to migration planning and assessment.



Many potential purposes for models



Each 15288 process definition suggests potentially assessable model impacts



- "Stakeholders of the system are identified.
- Required characteristics and context of use of capabilities and concepts in the life cycle stages, including operational concepts, are defined.
- Constraints on a system are identified.
- Stakeholder needs are defined.
- Stakeholder needs are prioritized and transformed into clearly defined stakeholder requirements.
- Critical performance measures are defined.
- Stakeholder agreement that their needs and expectations are reflected adequately in the requirements is achieved.
- Any enabling systems or services needed for stakeholder needs and requirements are available.
- Traceability of stakeholder requirements to stakeholders and their needs is established."

INCOSE MB Transformation; planning and assessment

- One way to stay focused pragmatically is to be very clear about explicit purposes for models.
- Because ISO 15288 offers a (relatively) well-known and accessible reference model for the life cycle management of systems, it provides a convenient “menu” listing of potential high level purposes of models in the life cycle of systems.
- The INCOSE Model-Based Transformation team is using this as the basis of an MBSE migration and maturation planning and assessment instrument . . .

INCOSE MB Transformation; Planning and Assessment Instrument

The INCOSE MBSE Transformation products are based on identification of --

Stakeholders in the MBSE Transformation:

1. Model Consumers (Model Users);
2. Model Creators (including Model Improvers);
3. Complex Idea Communicators (Model "Distributors");
4. Model Infrastructure Providers, Including Tooling, Language and Other Standards, Methods;
5. INCOSE and other Engineering Professional Societies.

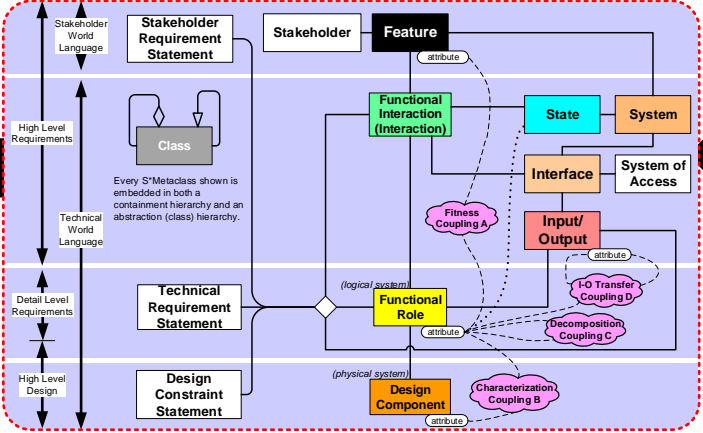
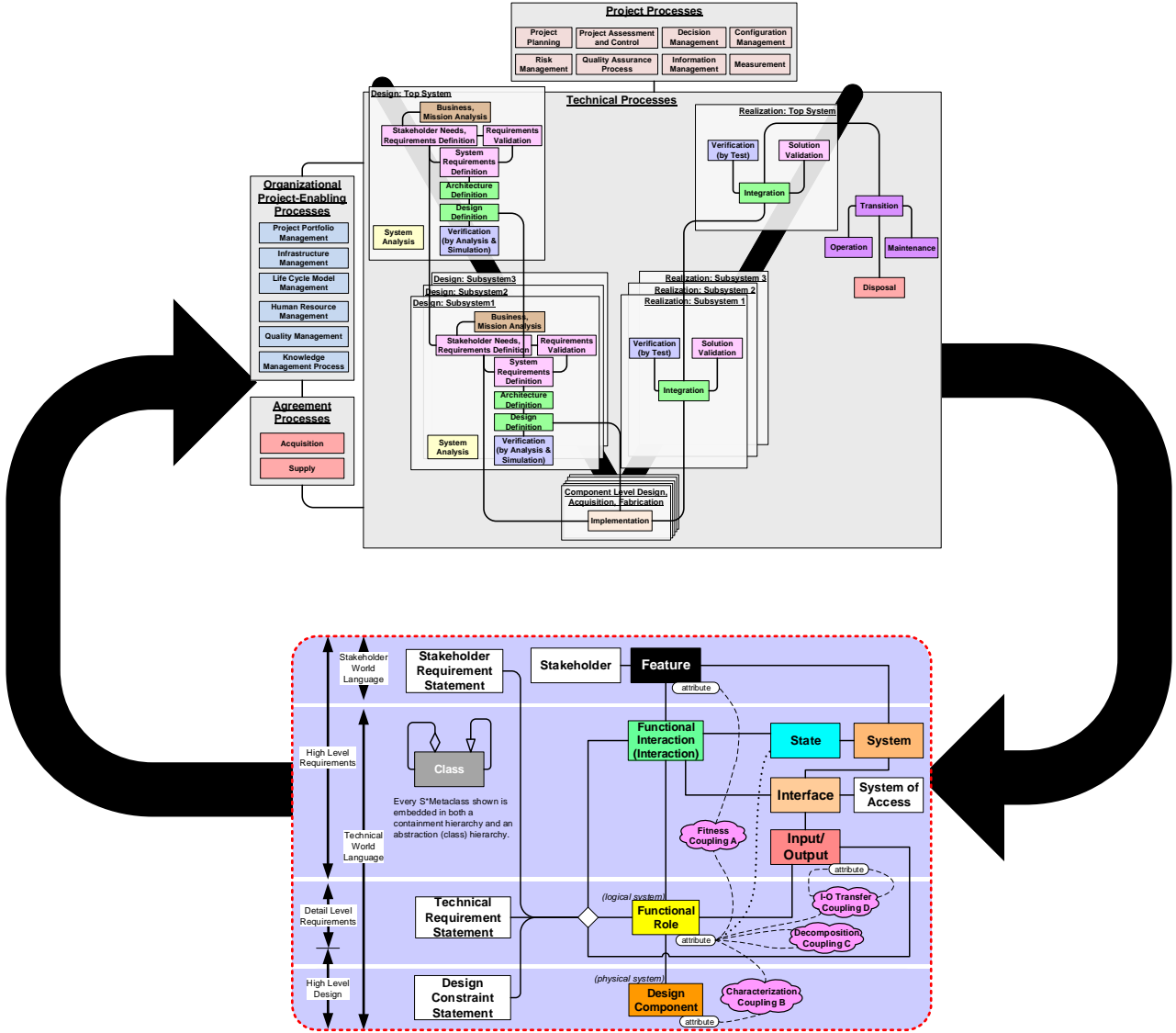
Notice that group (1) is by far the largest population of stakeholders, for future MBSE impact potential.

Innovation Process

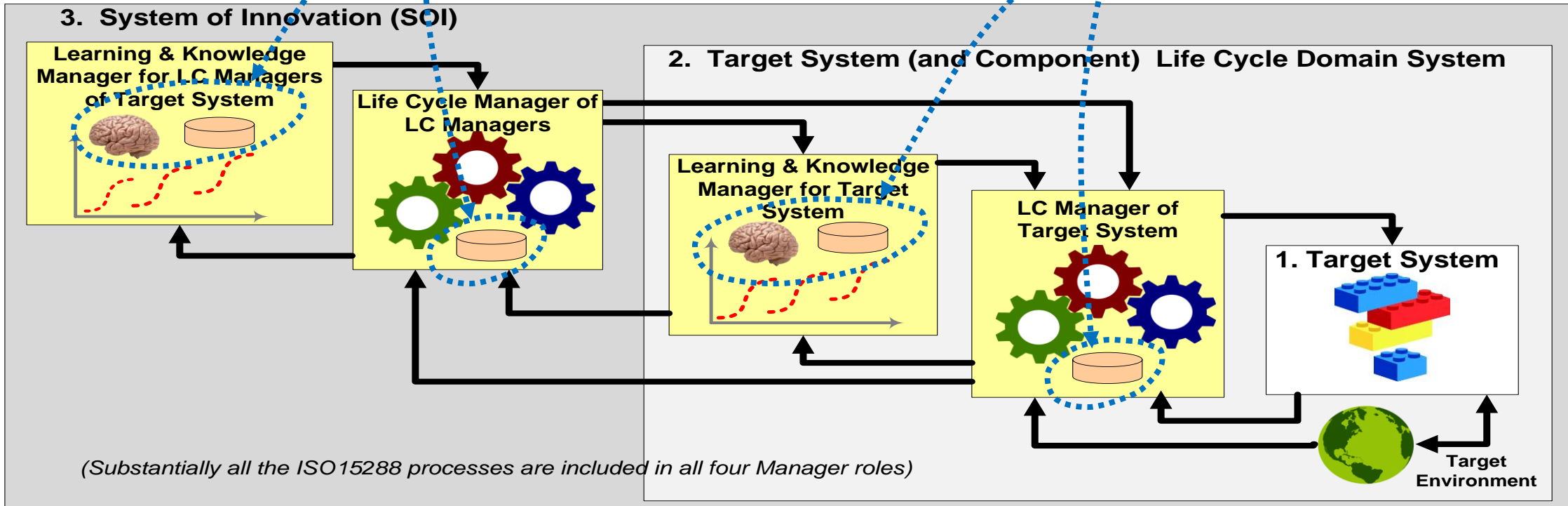
Models help make this real:

Shifting the emphasis from traditional focus on process and procedure, to greater emphasis on the state of the web of information passing through that process and procedure.

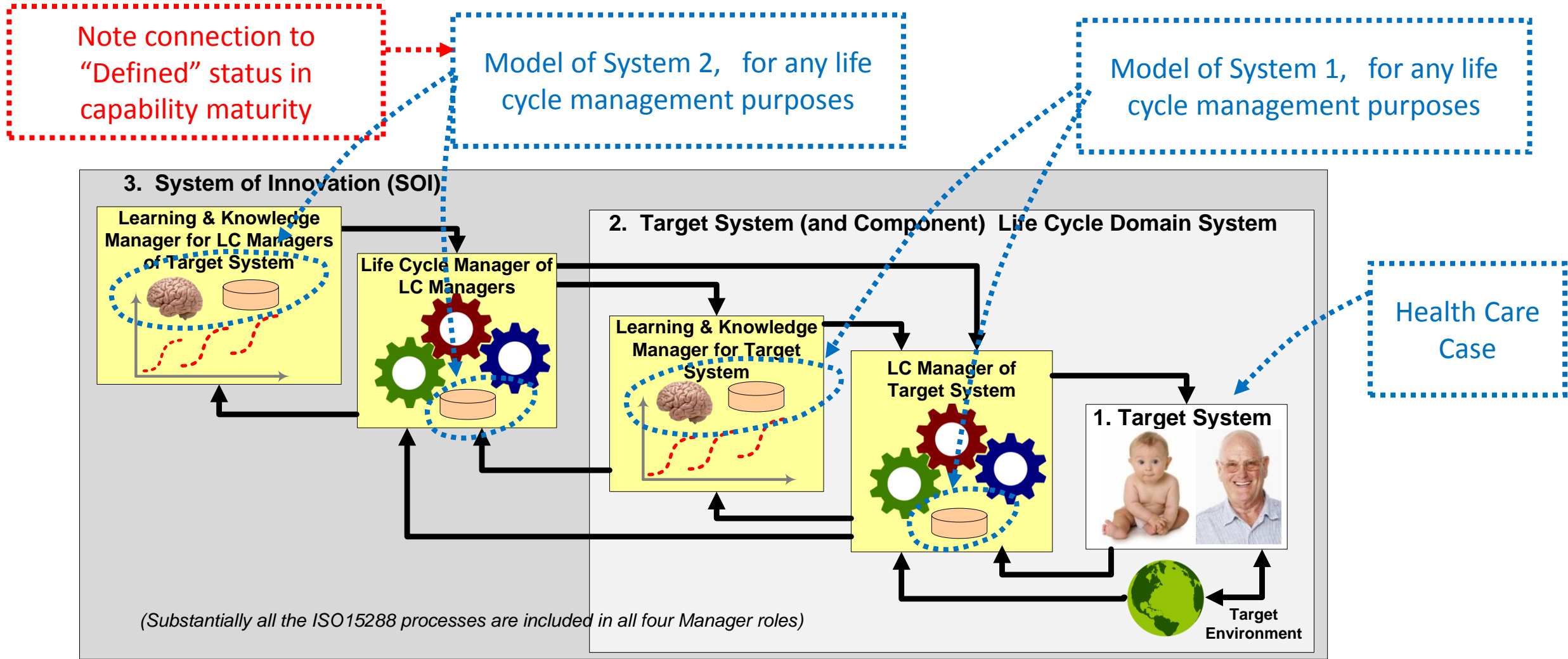
Compare to the traditional engineering disciplines.



Information Passing Through Innovation Process

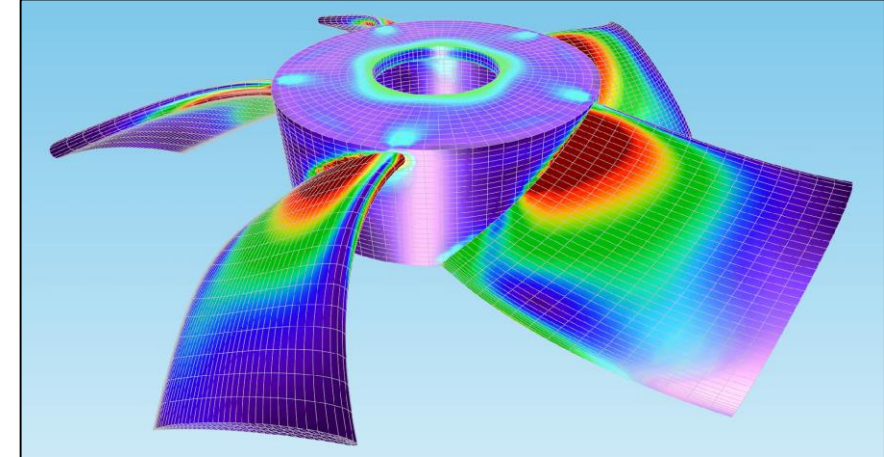
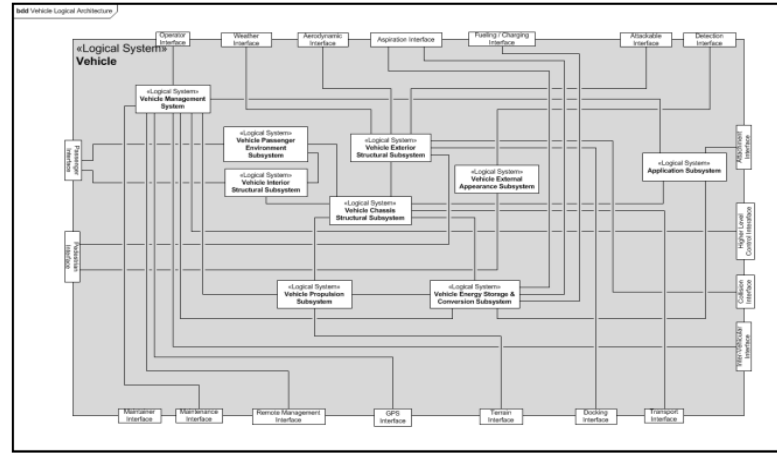


- System 1: Target system of interest, to be engineered or improved.
- System 2: The environment of (interacting with) S1, including all the life cycle management systems of S1, including learning about S1.
- System 3: The life cycle management systems for S2, including learning about S2.



- System 1: Target system of interest, to be engineered or improved.
- System 2: The environment of (interacting with) S1, including all the life cycle management systems of S1, including learning about S1.
- System 3: The life cycle management systems for S2, including learning about S2.

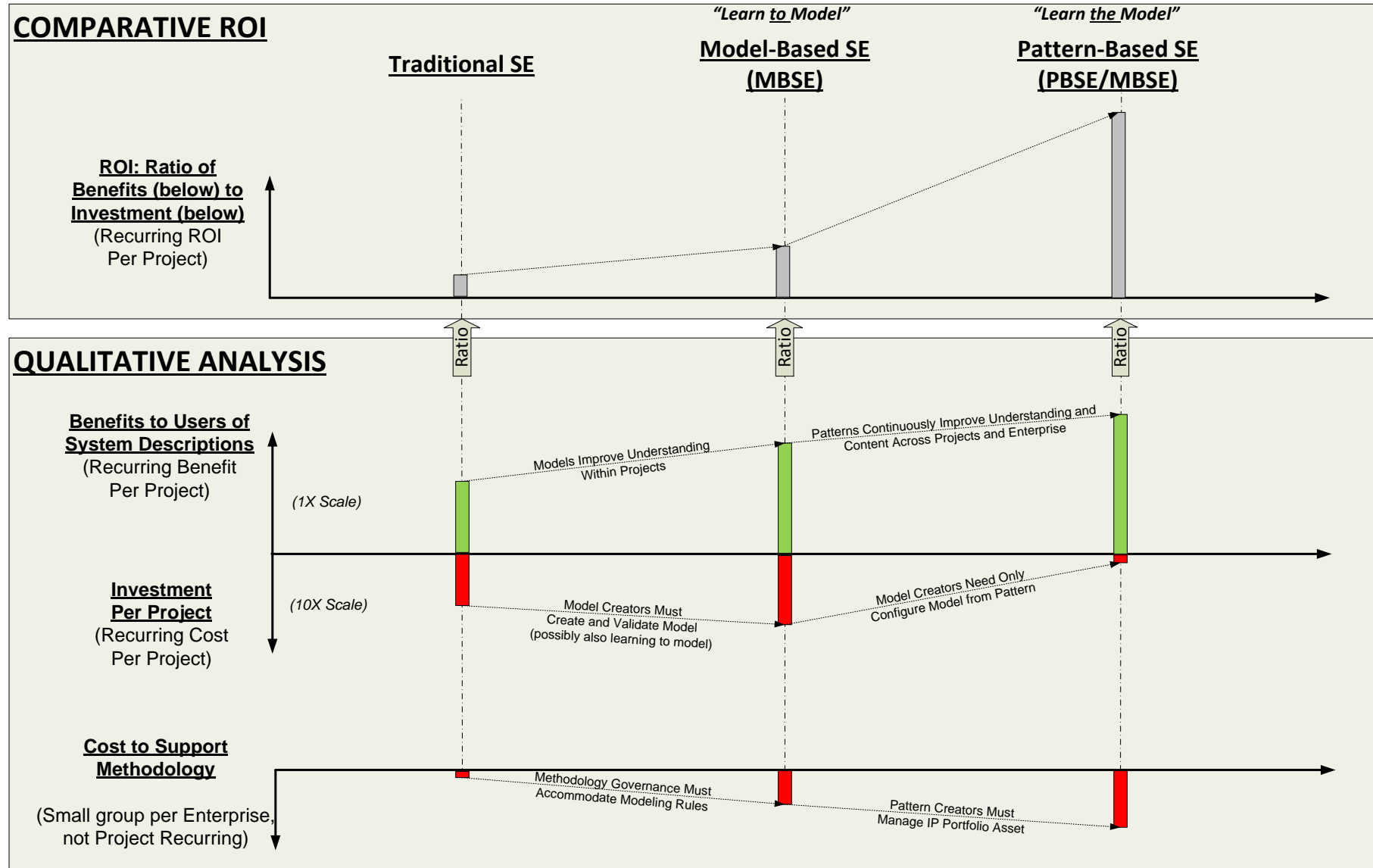
Enthusiasm for Models



The INCOSE systems community has shown growing enthusiasm for “engineering with models” of all sorts:

- Historical tradition of math-physics engineering models
- A World in Motion: INCOSE Vision 2025
- Growth of the INCOSE IW MBSE Workshop
- Growth in systems engineers in modeling classes
- INCOSE Board of Directors’ objective to accelerate transformation of SE to a model-based discipline
- Joint INCOSE activities with NAFEMS

Comparative Benefits and Costs Summary: Qualitative Relationships



Further analysis of the INCOSE MBE Transformation Stakeholders

Population ←-- Size (Log)	Stakeholders in A Successful MBSE Transformation (showing their related roles and parent organizations)						
		Industry & Govt. Initiatives	Organizations Internalizing MBSE, Including Govt. Contractors & Commercial	Vendors of MBSE Tooling and Services	Academia and Researchers	Technical Societies, Other Non- Technical Organizations	
Model Consumers (Model Users):							
****	Non-technical stakeholders in various Systems of Interest, who acquire / make decisions about / make use of those systems, and are informed by models of them. This includes mass market consumers, policy makers, business and other leaders, investors, product users, voters in public or private elections or selection decisions, etc.	X	X			X	
**	Technical model users, including designers, project leads, production engineers, system installers, maintainers, and users/operators.	X	X			X	
*	Leaders responsible to building their organization's MBSE capabilities and enabling MBSE on their projects	X	X			X	
Model Creators (including Model Improvers):							
*	Product visionaries, marketers, and other non-technical leaders of thought and organizations	X	X		X	X	
*	System technical specifiers, designers, testers, theoreticians, analysts, scientists	X	X		X	X	
*	Students (in school and otherwise) learning to describe and understand systems				X	X	
*	Educators, teaching the next generation how to create with models	X	X		X		
*	Researchers who advance the practice		X	X	X		
*	Those who translate information originated by others into models	X	X		X	X	
*	Those who manage the life cycle of models	X	X		X	X	
Complex Idea Communicators (Model "Distributors"):							
**	Marketing professionals	X	X	X		X	
**	Educators, especially in complex systems areas of engineering and science, public policy, other domains, and including curriculum developers as well as teachers	X	X	X	X		
**	Leaders of all kinds	X	X	X	X	X	
Model Infrastructure Providers, Including Tooling, Language and Other Standards, Methods:							
*	Suppliers of modeling tools and other information systems and technologies that house or make use of model-based information			X			
*	Methodologists, consultants, others who assist individuals and organizations in being more successful through model-based methods	X	X	X	X		
*	Standards bodies (including those who establish modeling standards as well as others who apply them within other standards)	X				X	
INCOSE and other Engineering Professional Societies							
*	As a deliverer of value to its membership					X	
*	As seen by other technical societies and by potential members					X	
*	As a great organization to be a part of					X	
*	As promoter of advance and practice of systems engineering and MBSE					X	

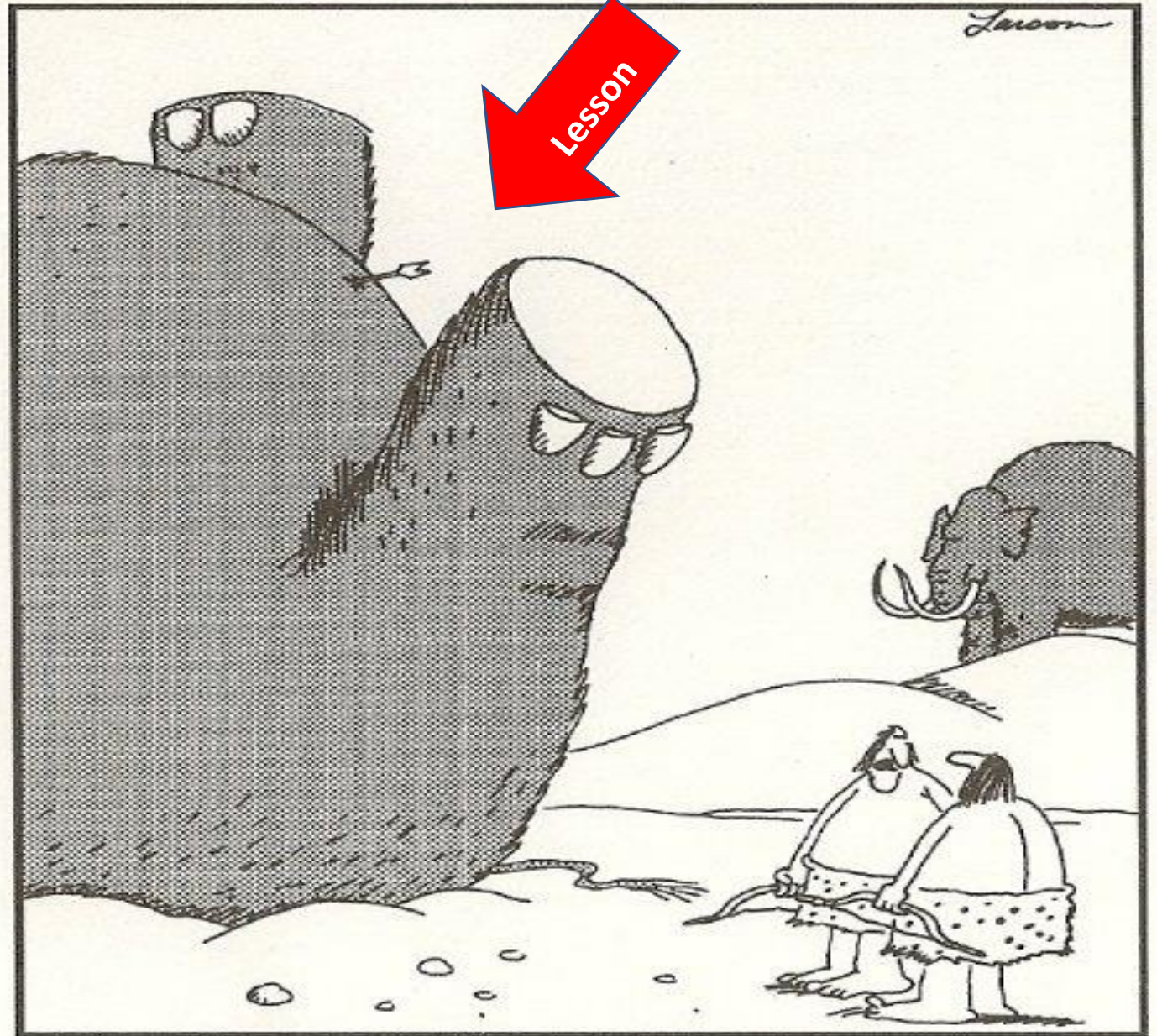
Lessons Learned: Effective Learning?

- In many enterprises, recording “lessons learned” is institutionalized as good practice:
 - At least, at the end of a project;
 - Often, in the form of a report or memorandum to file.
- Likewise, “Knowledge Management” efforts are noted, focusing on encoding what is deemed important for future work of others.
- Measuring effectiveness of such practices:
 - Instead of how often the data is referred to, how about . . .
 - how frequently related future work that could be impacted is effectively impacted, versus repeating similar work or problem consequences.

Lessons Learned?

Lessons Learned Report

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed aliquam odio eget massa feugiat, at tincidunt quam ullamcorper. Nullam ac purus tortor. Duis a ullamcorper augue. Pellentesque eu eros hendrerit, tempor tellus vitae, suscipit.

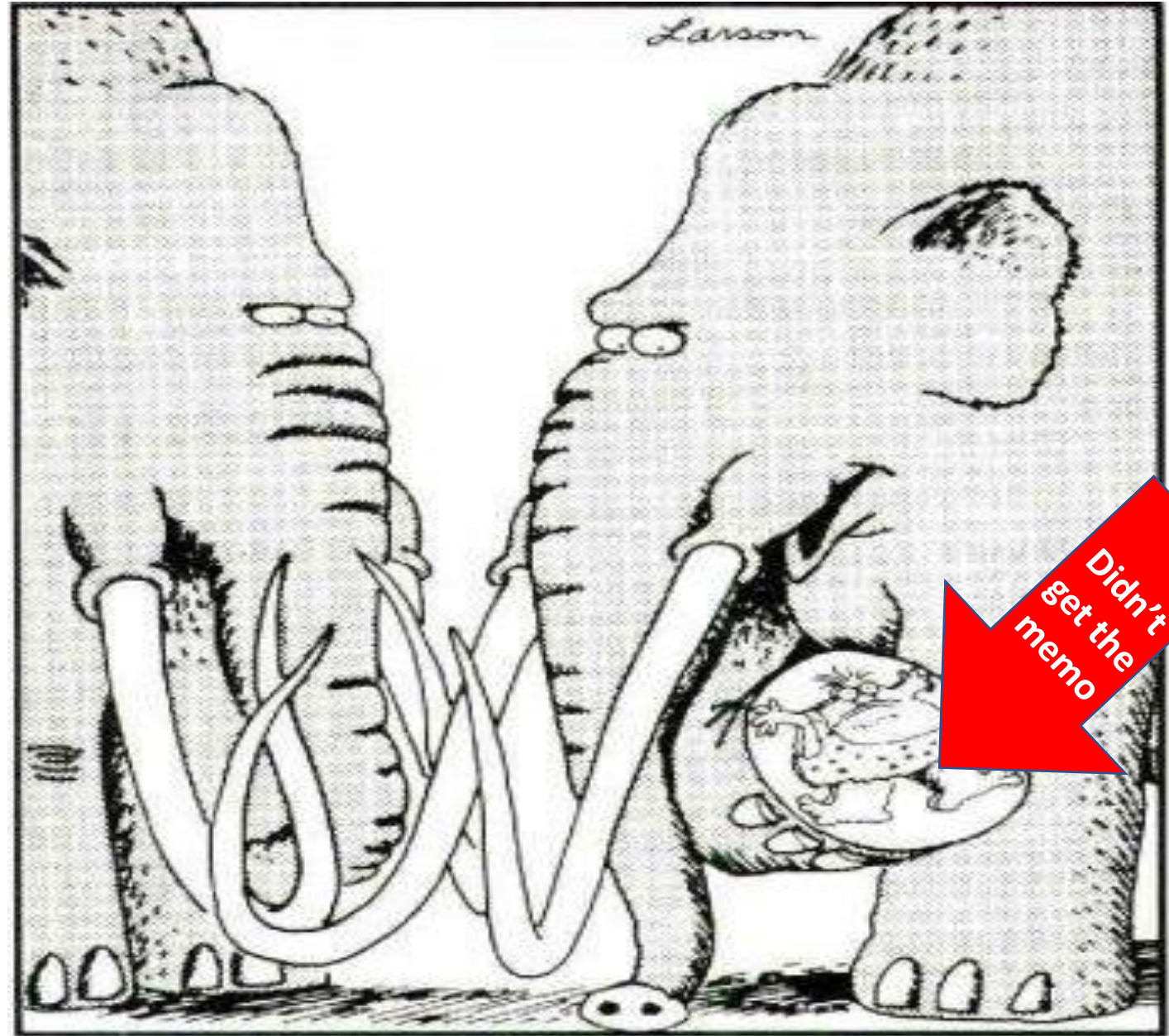


“We should write that spot down.”

Lessons Effectively Learned?

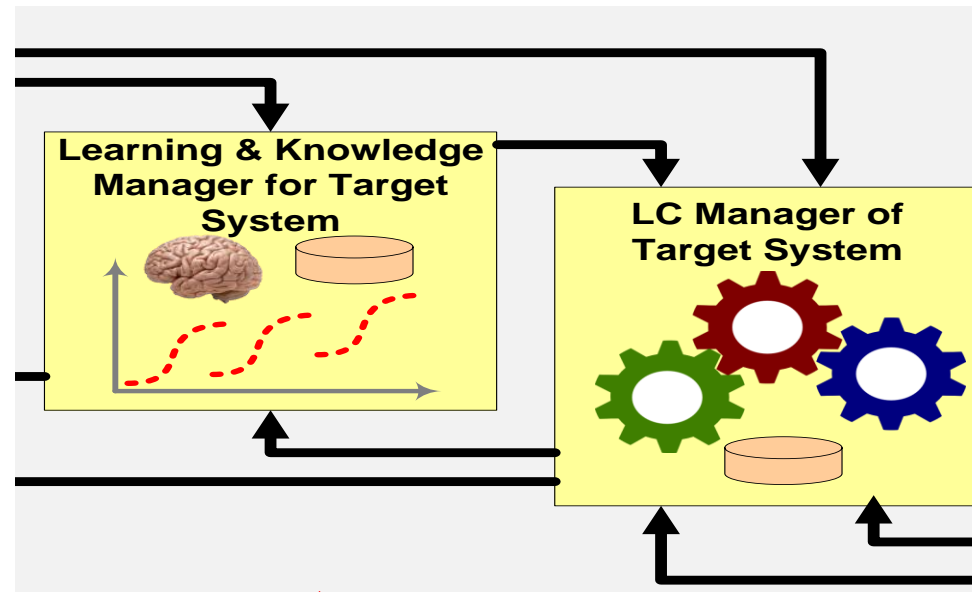
Lessons Learned Report

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed aliquam odio eget massa feugiat, at tincidunt quam ullamcorper. Nullam ac purus tortor. Duis a ullamcorper augue. Pellentesque eu eros hendrerit, tempor tellus vitae, suscipit.



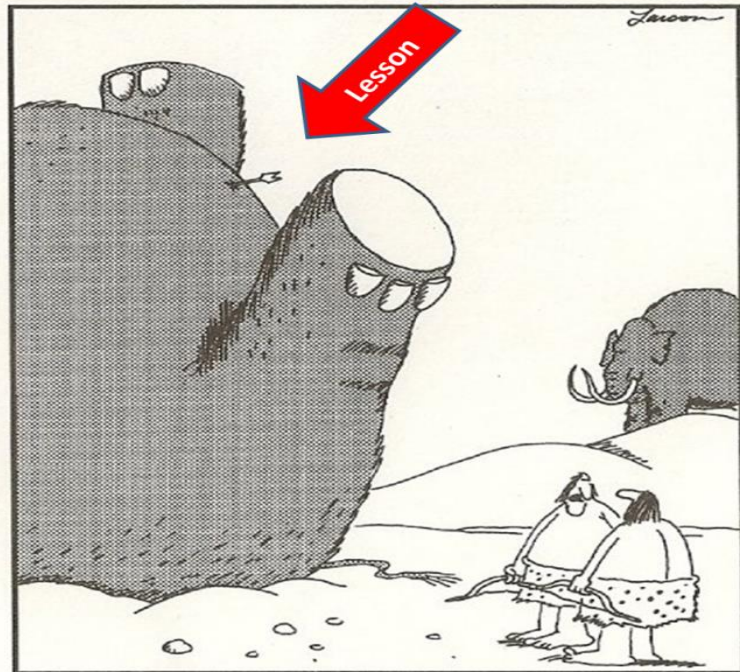
"Well, what the? ... I thought I smelled something."

Learning

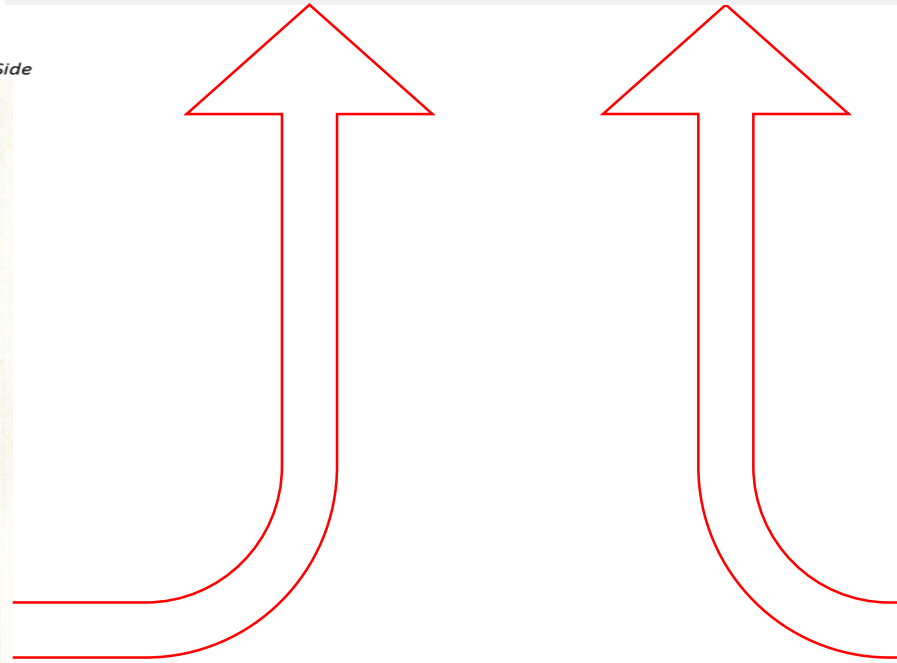
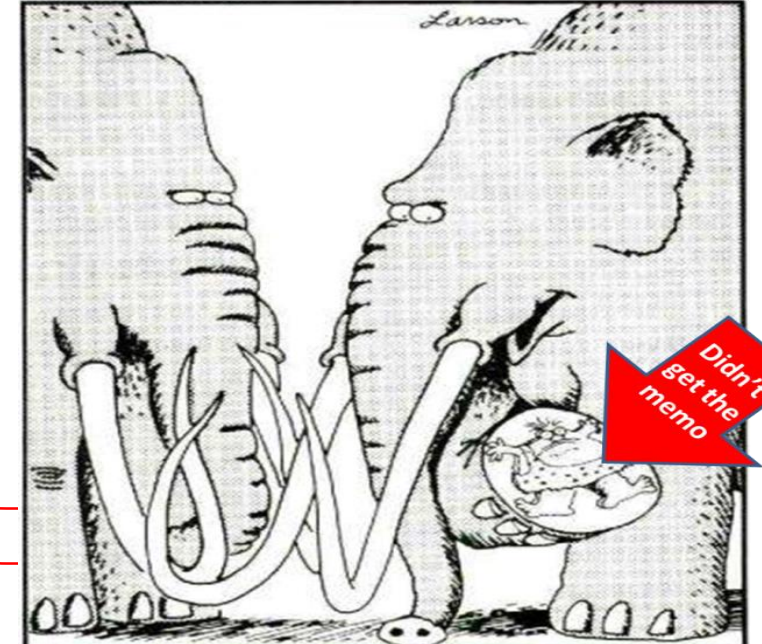


Executing

Copyright Gary Larson, *The Far Side*

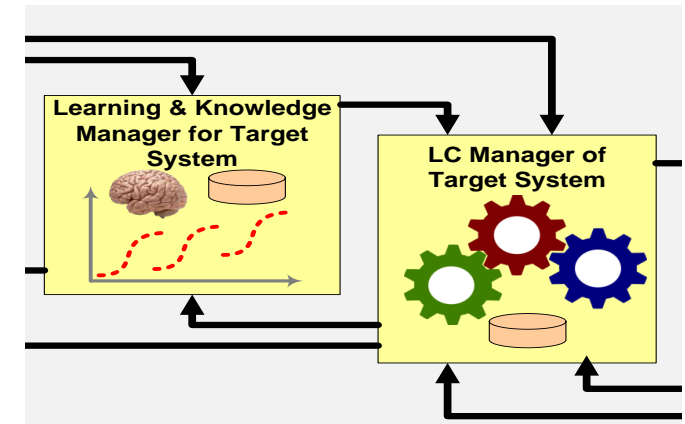


Copyright Gary Larson, *The Far Side*



Lessons Learned: Effective Learning?

- Where are the “lessons learned” encoded? What would cause them to be accessed?
- Compare to biology:
 - “Muscle Memory” builds “motor” learning directly into a future situation, for future unconscious use, vs. syllogistic reasoning that may not be remembered fast enough, or at all
 - This is about “effective learning” for future agile use
 - Just having a growing file of “lessons learned”, even if text searchable, is not the same as building what we learn directly in line with the path of future related work that will have to access it in order to be executed.
- Just because we label a report “lessons learned” does not mean that those who will need this information in the future will have access to it.



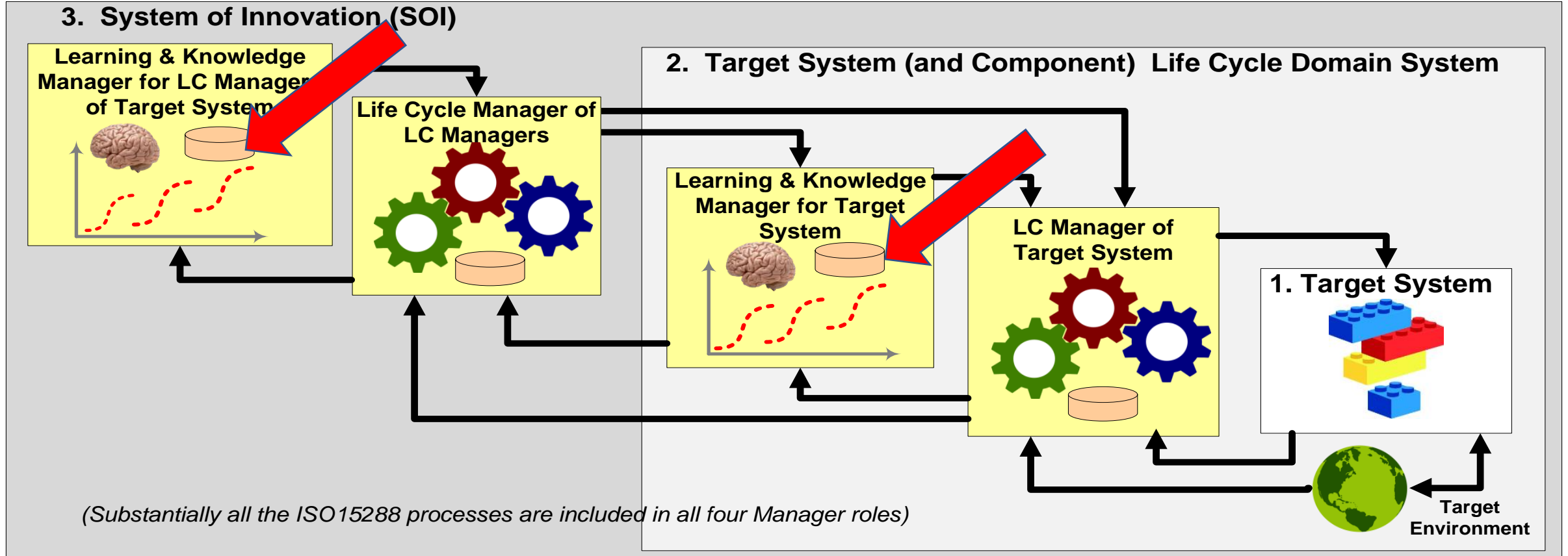
Learned models from STEM (~300 years) offer the most dramatic example of positive collaborative impact of effectively shared and validated models

- Effective Model Sharing:
 - We cannot view MBSE as mature if we perform modeling “from scratch”, instead of building on what we (*including others*) already know.
 - This is the basis of MBSE Patterns, Pattern-Based Systems Engineering (PBSE), and the work of the INCOSE MBSE Patterns Working Group.
 - S1 Patterns are built directly into future S2 project work of other people—effective sharing only occurs to extent it impacts future tasks performed by others.
 - This sharing may occur across individuals, departments, enterprises, domains, markets, society.
 - It applies not only to models of S1 (by S2), but also models of S2 (by S3).
- Effective Model Validation:
 - Especially when shared, models demand that we trust them.
 - This is the motivation for Model Validation, Verification, and Uncertainty Quantification (Model VVUQ) being pursued with ASME standards committees.
 - Effectiveness of Model VVUQ is essential to MBSE Maturity.
 - Because Model VVUQ adds significantly to the cost of a trusted model, MBSE Patterns are all the more important—they IP of enterprises, industries.

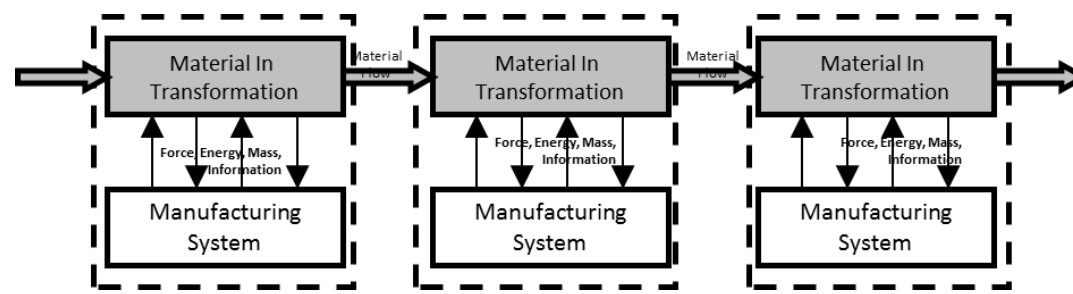
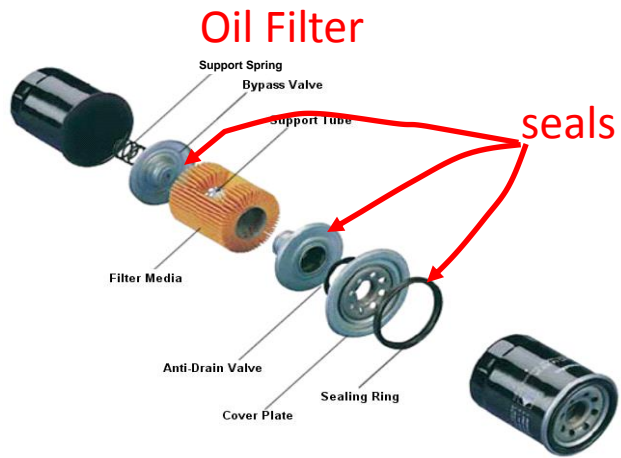
An emerging special case: Regulated markets

- Increasing use of computational models in safety-critical, other regulated markets is driving development of methodology for Model VVUQ:
 - See, for example, ASME V&V 10, 20, 30, 40, 50, 60.
- Models have economic advantages, but the above can add new costs to development of models for regulatory submission of credible evidence:
 - Cost of evidentiary submissions to FDA, FAA, NRC, NTSB, EPA, OSHA, when supported by models—includes VVUQ of those models.
- This suggests a vision of collaborative roles for engineering professional societies, along with regulators, and enterprises:
 - Trusted shared MBSE Patterns for classes of systems
 - Configurable for vendor-specific products
 - With Model VVUQ frameworks lowering the cost of model trust for regulatory submissions
- Further emphasizes the issue of trust in models . . .

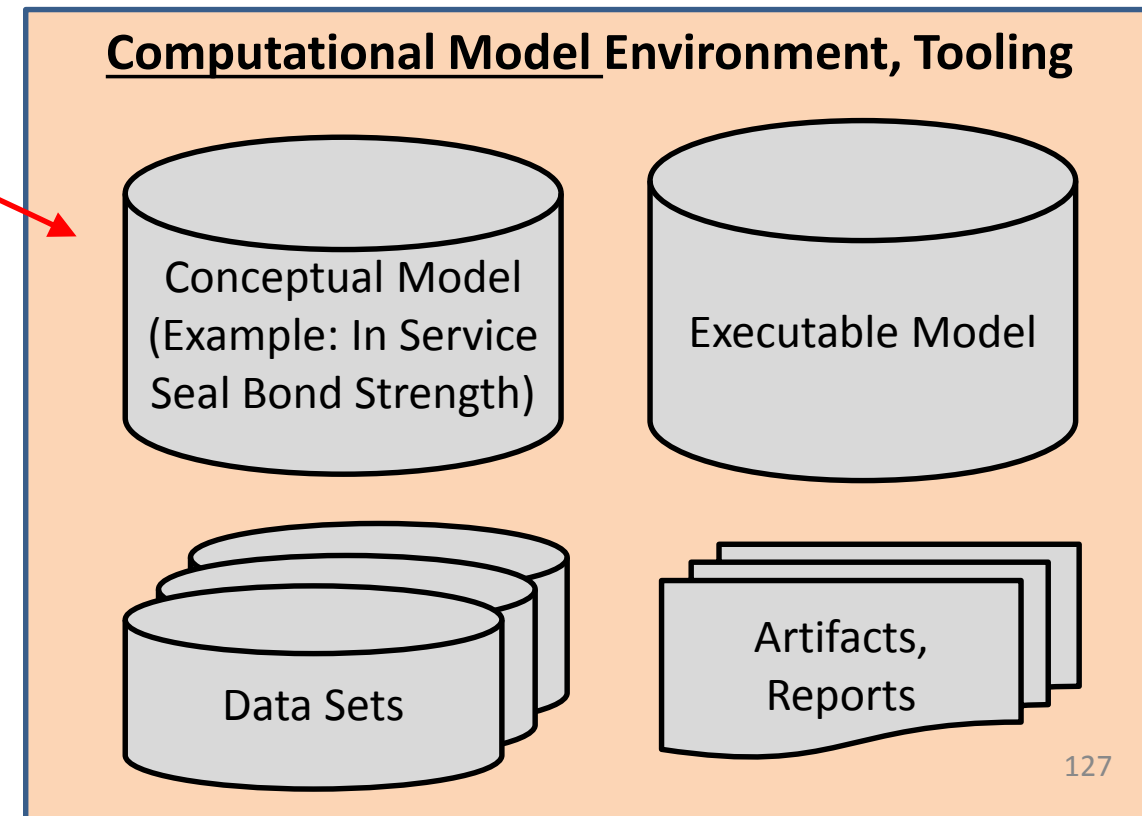
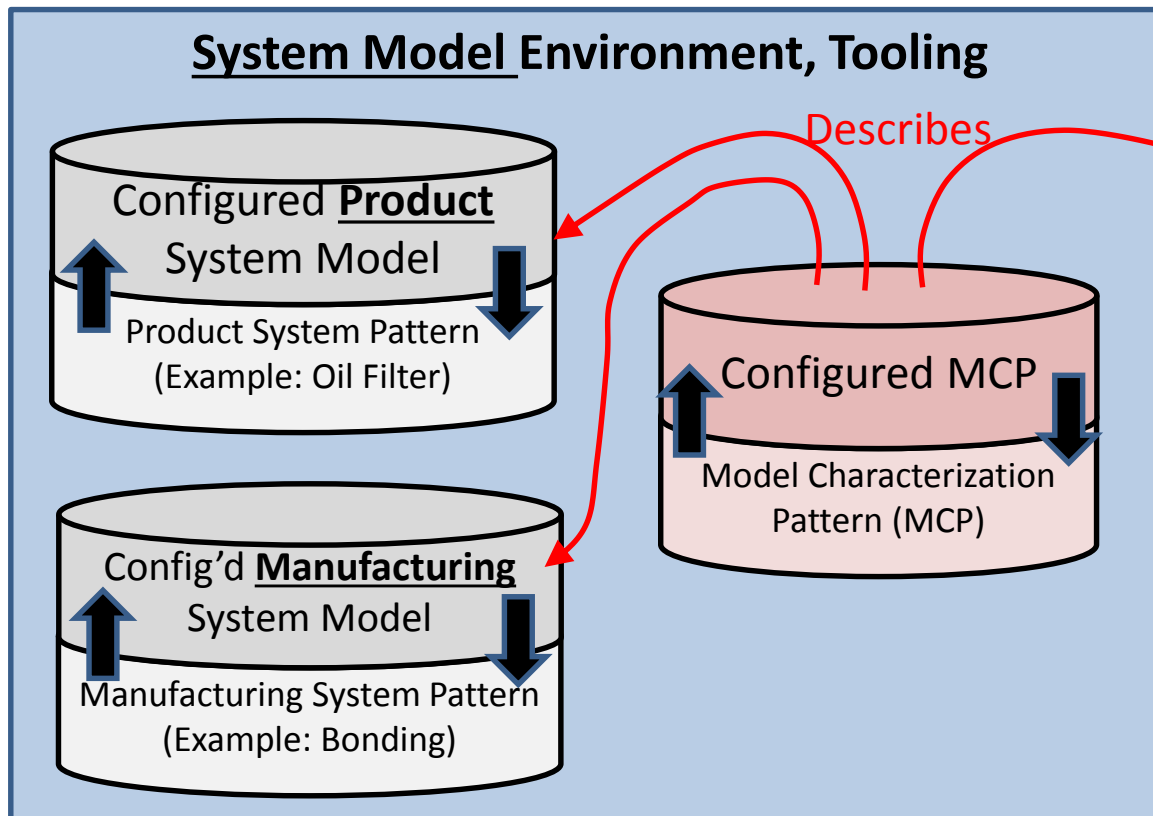
An emerging special case: Regulated markets



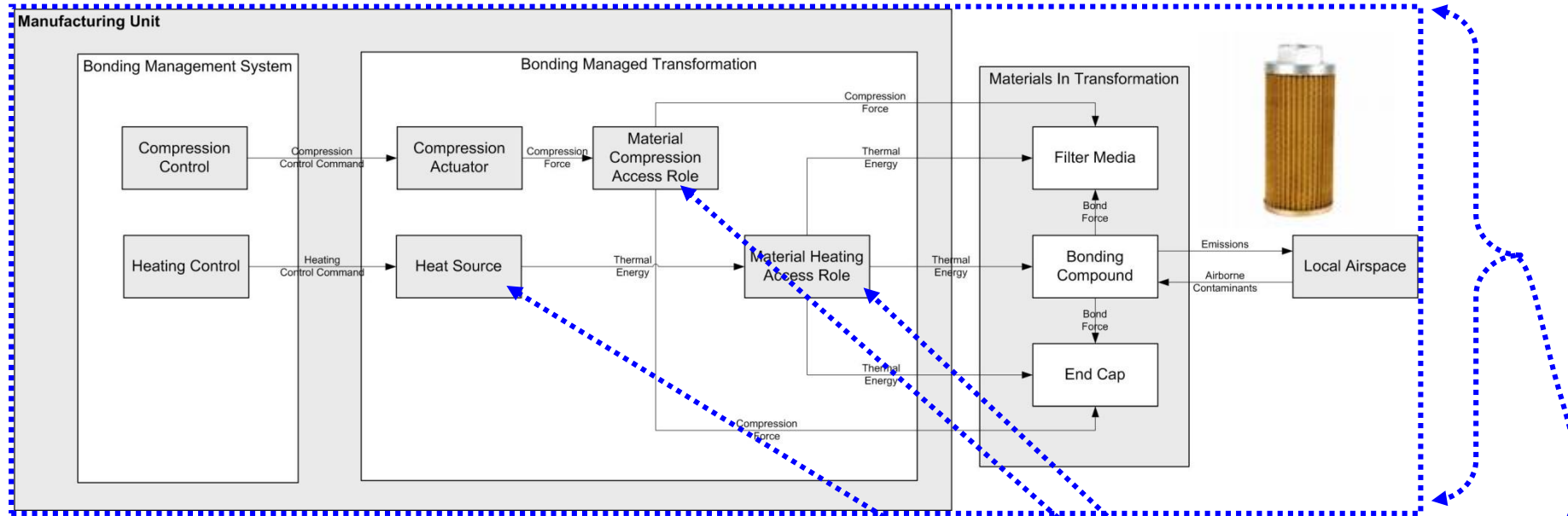
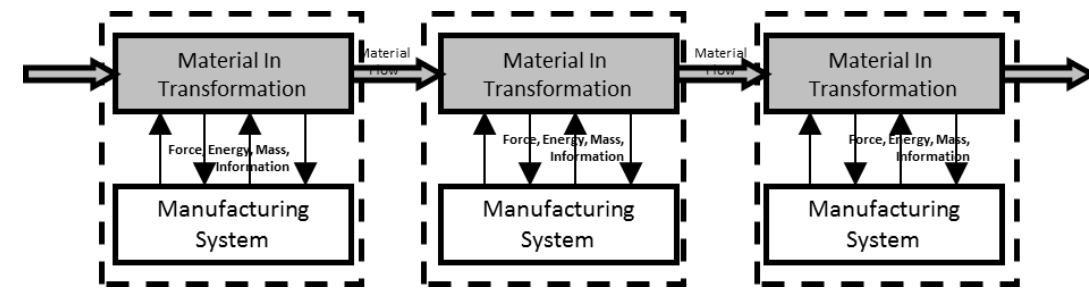
- Trusted shared MBSE Patterns for classes of systems
- Configurable for vendor-specific products
- With Model VVUQ frameworks lowering the cost of model trust for regulatory submissions



Appendix VI: Example Configured MCP--Heavy Duty Oil Filter Manufacturing and Product Performance



Manufacturing Interaction: Form Thermo Compression Bond

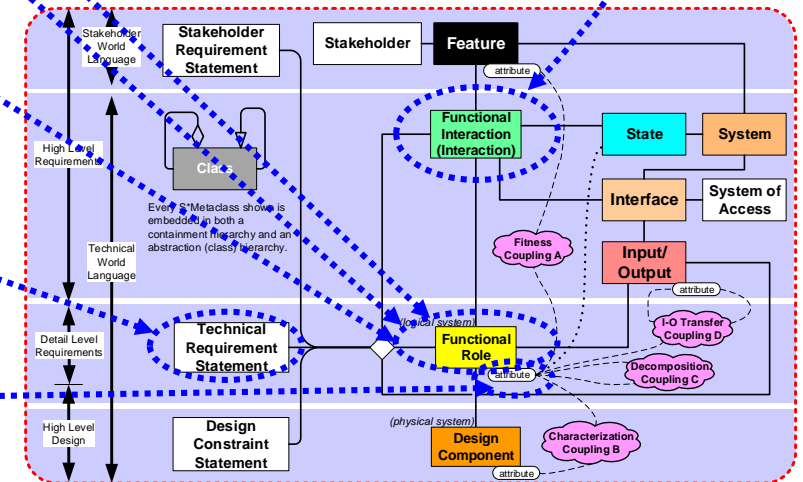


Requirements Statements may be viewed as Transfer Functions across Functional Roles, greatly improving ability to audit regular detail requirements by embedding them in the Model:

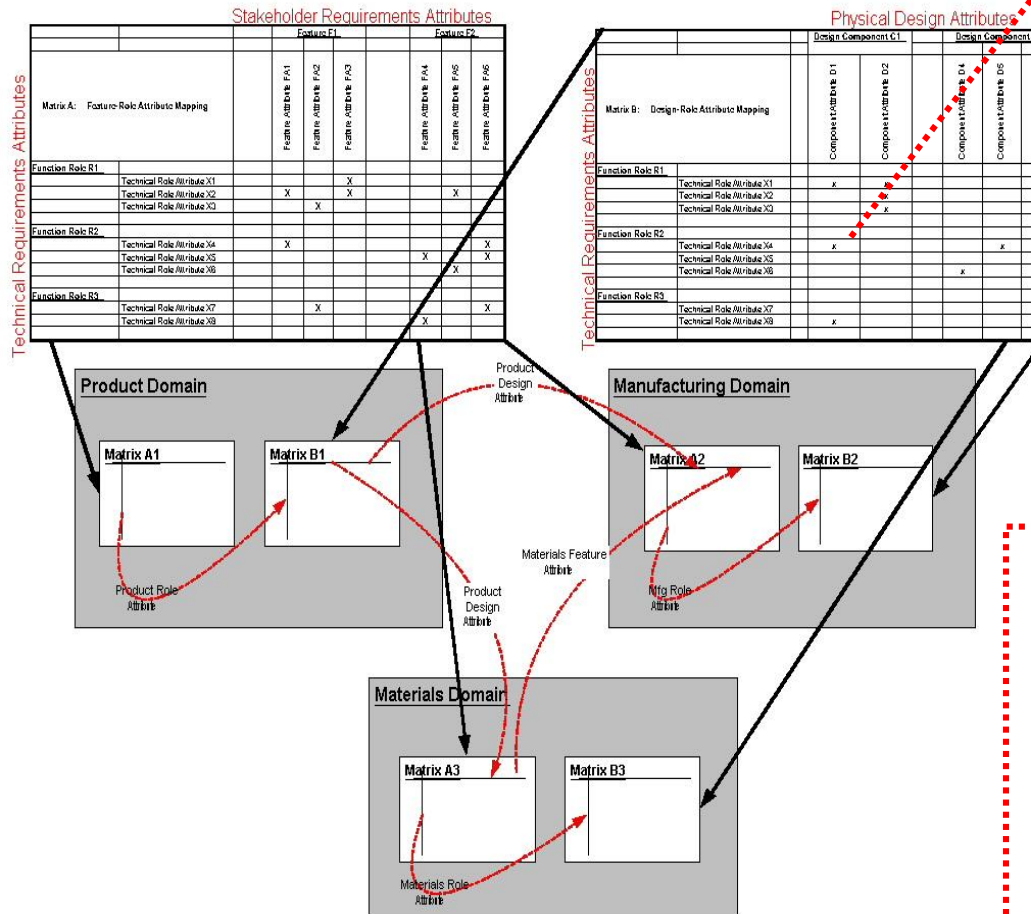
Requirement OFM-32: "The Manufacturing System shall deliver a Compression Force of [Min Bond Force] for a period of [Min Bond Time]."

Requirement OFM-33: "The Manufacturing System shall deliver Thermal Energy sufficient to maintain a bond temperature of [Min Bond Temperature] for a period of [Min Bond Time]."

Requirement OF-51: "The Oil Filter shall operate at lubricant pressure of [Max Lubricant Pressure] with structural failure rates less than [Max Structural Failure Rate] over an in-service life of [Min Service Life]."



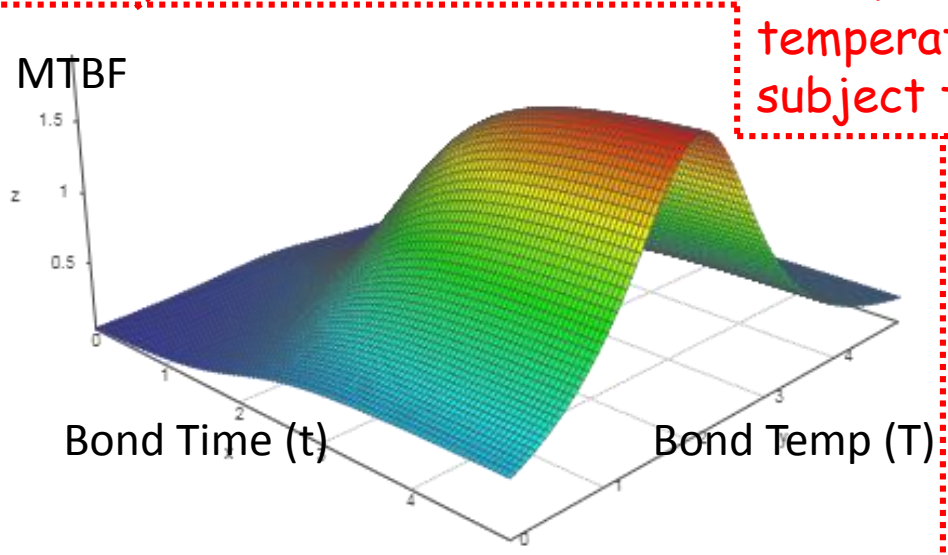
Parametric attribute couplings cross manufacturing process, materials, and product performance domains



The Coupling Model is a unifying framework integrating all forms of coupling:

- First principles equations
- Empirical datasets
- Graphical relations
- Data tables
- Prose statements
- Fuzzy relationships
- Other

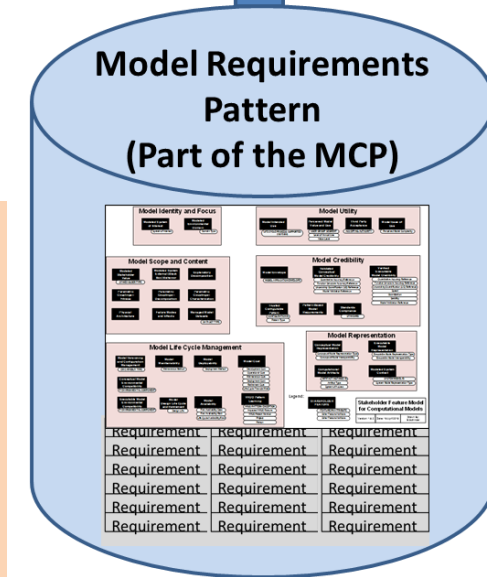
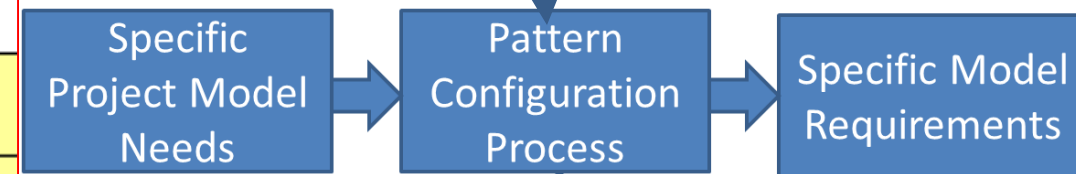
Computational model expresses oil filter product's in service MTBF under pressure, as a function of manufacturing bonding temperature and time—subject to Model VVUQ.

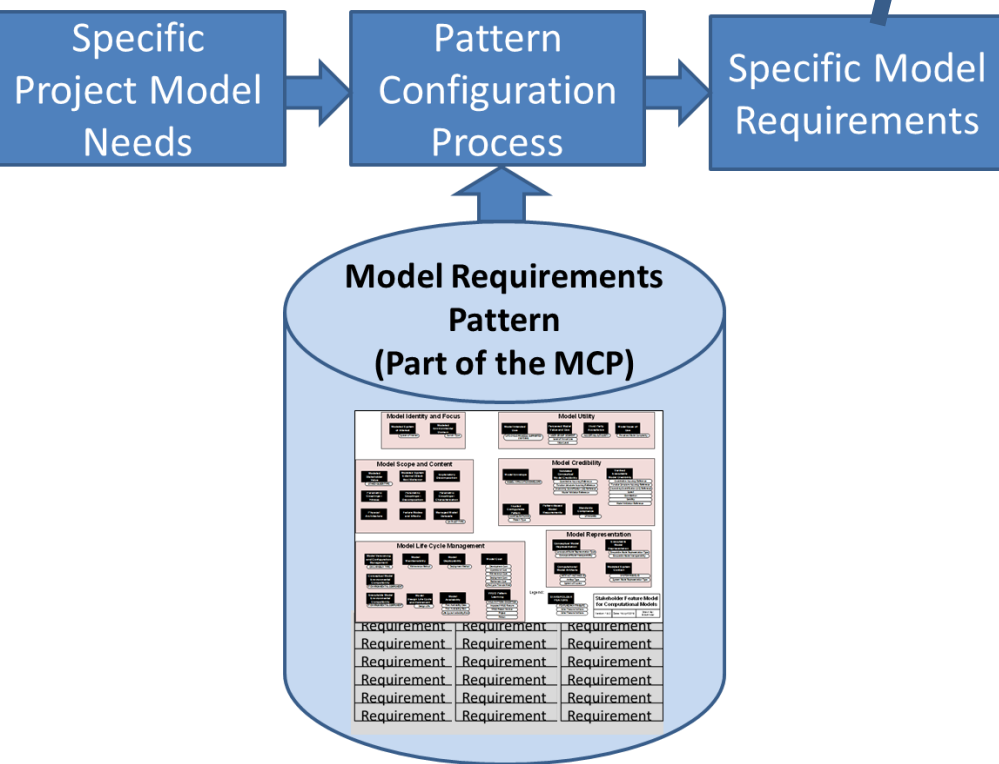


Configuring the MCP for a Model or Project

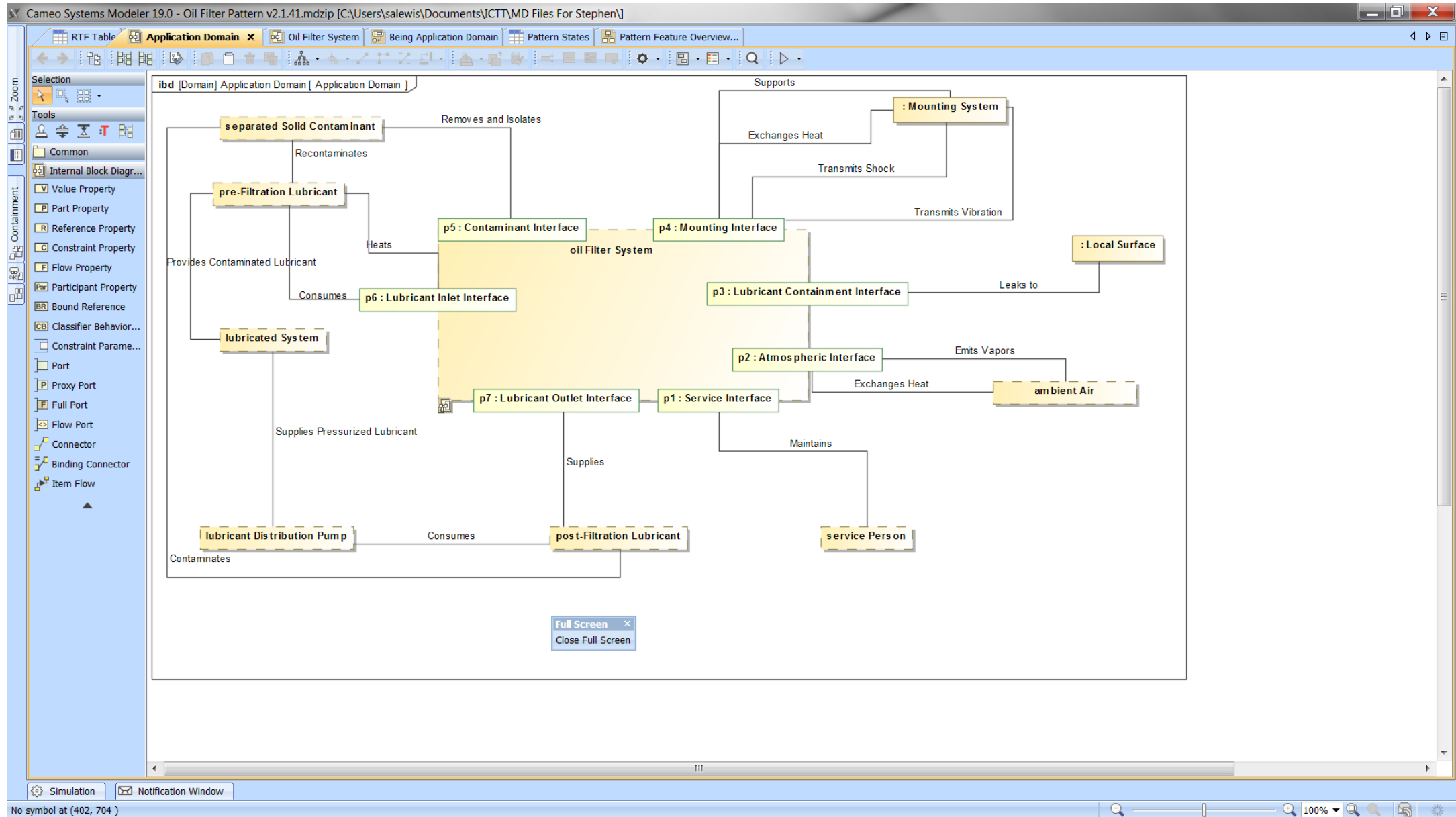
Populate? (YES/NO)	Feature Name	Feature Attribute Primary Key (PK) Attribute	Feature Attribute PK Value #1	Feature Attribute PK Value #2	Feature Attribute PK Value #3	Feature Attribute PK Value #4	Feature Attribute PK Value #5	Feature Attribute Value
YES	Model Ease of Use	--						
YES	Model Intended Use	Life Cycle Process Supported	Verification Process					
YES	Perceived Model Value and Use	User Group Segment	Product Designer	Acquisition Process Supply Process Project Planning Process Project Assessment and Decision Management Risk Management Process Configuration Management Information Management				
YES	Third Party Acceptance	Accepting Authority	FTC					
YES	Model Credibility	--						
YES	Model Identity and Focus							
YES	Model Life Cycle Management							
YES	Model Representation							
YES	Model Scope and Content							
YES	Model Utility	--						

Model planner selecting which MCP Features to populate, along with their attribute values (example shows entering ISO 15288 process areas from pull-down list)



Features	Functional Role	Req ID	Requirement
Model Availability [], Model Ease of Use[], Model Intended Use[0], Model Intended Use[Verification Process], Model Intended Use[Risk Management Process]	Virtual Model System	5.1	The model user interfaces, per the [Model UI Specification] shall facilitate the efficient and effective performance of the intended purpose of the model by a user of the designated persona type.
 <p>The diagram illustrates the flow from 'Specific Project Model Needs' through a 'Pattern Configuration Process' to 'Specific Model Requirements'. A 'Model Requirements Pattern (Part of the MCP)' is shown as a central database-like structure containing various model-related diagrams and a grid of requirements.</p>			<p>1 The Model shall represent the external Input-Outputs exchanged during interactions with Domain Actors, and the external Interfaces through which they are exchanged.</p> <p>3 The modeled external interactions shall include any parasitic aspects which arise from choice of internal design, materials, technologies, or solution approach but which were not otherwise required by the primary intended system purpose, where significant from a stakeholder perspective.</p> <p>1 The model shall include identification of component failure modes, as to underlying state leading to predicted failure.</p> <p>2 For each identified failure mode, the model shall include identification of cause(s) of failure mode.</p> <p>3 For each identified failure mode, the model shall include the probability of failure mode.</p> <p>4 For each identified failure mode, the model shall include the effect(s) of the mode.</p>
Failure Modes and Effects[]	Virtual Model System	2.6.5	For each identified failure effect, the model shall include the severity of impact of the failure.
<p>Sampling of resulting populated Model Technical Requirements</p>			<p>model shall represent the system of interest over a specified (discrete or continuous) range or envelope of technical external environment interaction configurations.</p>

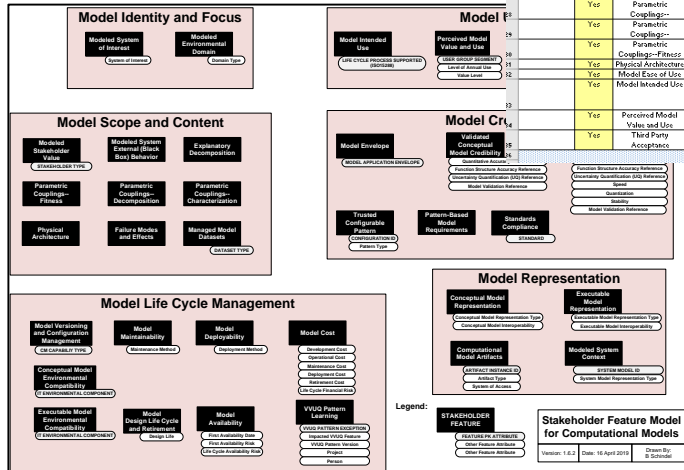
From SysML Model of Oil Filter Product Family



Configuring MCP Model Stakeholder Features

A	B	D	E	F	G	H	I	J
Mandatory, Optional, or Other Configuration Rule	Populated? (Yes/No)	Feature Name	Feature Attribute Primary Key (PK) Attribute	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
Yes	Yes	Model Envelope	--					
Yes	Yes	Pattern-Based Model Requirements	--					
Yes	Yes	Standards Compliance	Standard	ASME VV10	ASME VV40			
Yes	Yes	Trusted Configurable Pattern	--					
Yes	Yes	Validated Conceptual Model	--					
Yes	Yes	Verified Executable Model Credibility	--					
Yes	Yes	Modeled Environmental Domain	Domain Type	In Service Application	Production			
Yes	Yes	Modeled System of Interest	--					
Yes	Yes	Conceptual Model Environmental Compatibility	IT Environmental Component	MATLAB				
Yes	Yes	Executable Model Environmental Compatibility	IT Environmental Component	MATLAB				
Yes	Yes	Model Availability	--					
Yes	Yes	Model Cost	--					
Yes	Yes	Model Deployability	--					
Yes	Yes	Model Design Life and Retirement	--					
Yes	Yes	Model Maintainability	--					
Yes	Yes	Model Versioning and Configuration Management	CM Capability Type	Basic Version Management				
No	No	VVUQ Pattern Learning	VVUQ Pattern Exception					
Yes	Yes	Computational Model Artifacts	Artifact Instance ID	Verification Simulation Report	Validation Simulation Report			
Yes	Yes	Conceptual Model Representation	--					
Yes	Yes	Executable Model Representation	--					
Yes	Yes	Modeled System Content	System Model ID	Oil Filter System				
Yes	Yes	Explanatory Decomposition	--					
Yes	Yes	Failure Modes and Effects	Dataset Type	Model Validation Dataset	Simulation Dataset			
No	No	Modeled Stakeholder Value	Stakeholder Type					
Yes	Yes	Modeled System External (Black Box) Behavior	--					
Yes	Yes	Parametric Configurability	--					
Yes	Yes	Parametric Configurability-Fitness	--					
Yes	Yes	Physical Architectures	--					
Yes	Yes	Model Ears of Use	--					
Yes	Yes	Model Intended Use	Link Cycle Process Supported	Verification Process	Decision Management Process			
Yes	Yes	Perceived Model Value and Use	Product Support	User Group Support	Product Designer	FTC		
Yes	Yes	Third Party Acceptance	Accepting Authority					

	A	B	D	E	F	G	H	I
	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
1								
2		Yes	Model Envelope	--				
3		Yes	Pattern-Based Model Requirements	--				
4		Yes	Standards Compliance	Standard	ASME VV10	ASME VV40		
5		Yes	Trusted Configurable Pattern	--		ASME VV10 ASME VV20 ASME VV30 ASME VV40 ASME VV50		
6		Yes	Validated Conceptual Model Credibility	--				
7		Yes	Verified Executable Model Credibility	--				
8		Yes	Modeled Environmental Domain	Domain Type	In Service Application	Production		
9		Yes	Modeled System of Interest	--				
10		Yes	Conceptual Model Environmental Compatibility	IT Environmental Component	MATLAB			
11		Yes	Executable Model Environmental Compatibility	IT Environmental Component	MATLAB			
12		Yes	Model Availability	--				
13		Yes	Model Cost	--				
14		Yes	Model Deployability	--				
15		Yes	Model Design Life and Retirement	--				
16		Yes	Model Maintainability	--				
17		Yes	Model Versioning and Configuration Management	CM Capability Type	Basic Version Management			
18		No	VVUQ Pattern Learning	VVUQ Pattern Exception				
19		Yes	Computational Model Artifacts	Artifact Instance ID	Verification Simulation Report	Validation Simulation Report		



Resulting Configured Computational Model Reqs

Cameo Systems Modeler 19.0 - Model Characterization Pattern 2.1.48 [MD File 1.0.4].mdzip [C:\Users\JShery\Documents\V4\ASME conference 201805]

Pattern Features and Fea... Configured Requirements x Pattern Functional Inter... Pattern Logical Systems Pattern Feature Interact... Pattern Interaction Role... Pattern Requirements Pattern Feature Overview...

Add New Add Existing... Delete Remove From Table Columns Export

#	Functional Interaction	Functional Role	△ Name	Text	Functional Failure	Status	Rationale
1	View Conceptual Model[SOI]	Virtual Model System	55 1.1	The model shall identify the focal system of interest.			
2	View Conceptual Model[DOM]	Virtual Model System	55 1.2	The model shall represent all the external Domain Actors with which the subject system significantly interacts.			
3	View Conceptual Model[CPLNGS--FITNES]	Virtual Model System	55 2.1.4	For each Measure of Effectiveness (Feature Attribute), the model shall represent the quantitative coupling that determines its values versus those of the Measures of Performance upon which its valuation or fitness			
4	View Conceptual Model[EXTRN]	Virtual Model System	55 2.2.1	The Model shall represent the external Input-Outputs exchanged during interactions with Domain Actors, and the external Interfaces through which they are exchanged.			
5	View Conceptual Model[EXTRN]	Virtual Model System	55 2.2.2	The model shall represent all the significant external interactions that the system of interest has with its listed environmental actors, listing which actors are involved in each interaction.			
6	View Conceptual Model[EXTRN]	Virtual Model System	55 2.2.3	The modeled external interactions shall include any parasitic aspects which arise from choice of internal design, materials, technologies, or solution approach but which were not otherwise required by the primary intended system purpose, where significant from a stakeholder perspective.			
7	View Conceptual Model[EXTRN]	Virtual Model System	55 2.2.4	For each identified Interaction, the model shall include the dynamically changing quantities significant to the interaction, for both the System of Interest and the External Actors in the Interaction.			
8	View Conceptual Model[EXTRN]	Virtual Model System	55 2.2.5	For each identified Interaction, the model shall include the static or slow changing quantities characterizing the systems performance of the interaction, for both the System of Interest and the External Actors in the			
9	View Conceptual Model[EXTRN]	Virtual Model System	55 2.2.6	The model shall represent the different behavioral modes (states) of the system of interest that are significant to the intended use of the model.			
10	View Conceptual Model[EXTRN]	Virtual Model System	55 2.2.7	The model shall represent the possible (state) transitions between the modeled system behavioral modes.			
11	View Conceptual Model[EXTRN]	Virtual Model System	55 2.2.8	For each of its modeled behavioral modes (states), the model shall represent which external interactions the system of interest can have with its environmental actors, from the list of possible interactions.			
12	View Conceptual Model[EXTRN]	Virtual Model System	55 2.2.9	For each modeled interaction of the system of interest with its environment, the required external behavior of the system of interest shall be included in the model.			
13	View Conceptual Model[DECMP]	Virtual Model System	55 2.3.1	For each modeled external Interaction, the model shall represent the decomposition of the behavior of the system of interest into internal interactions between internal roles.			
14	View Conceptual Model[DECMP]	Virtual Model System	55 2.3.2	The model shall represent the internal decomposition of the system of interest functional roles until small enough to be allocated to single physical components of the modeled physical architecture.			
15	View Conceptual Model[DECMP]	Virtual Model System	55 2.3.3	For each modeled internal decomposed functional role, the model shall include the dynamically changing quantities significant to the related internal interactions.			
16	View Conceptual Model[DECMP]	Virtual Model System	55 2.3.4	For each modeled internal Interaction, the model shall include the static or slow changing quantities characterizing the system's performance of the related internal interactions.			
17	View Conceptual Model[CPLNGS--DECMP]	Virtual Model System	55 2.3.5	For each behavioral roles Measure of Performance, the model shall represent the quantitative coupling that determines its values versus those of the internal (decomposed) Measures of Performance upon which it depends.			
18	View Conceptual Model[PHYSARCH]	Virtual Model System	55 2.3.6	The model shall represent the set of physical components of the system of interest.			
19	View Conceptual Model[PHYSARCH]	Virtual Model System	55 2.3.7	For each modeled physical component, the model shall include attributes describing the type or identity of the physical component, indicating material type or composition, manufacturer part number, or other non-behavioral			
20	View Conceptual Model[PHYSARCH]	Virtual Model System	55 2.3.8	For each modeled physical component, the model shall represent its physical architectural relationships (connection, adjacency, geometry, containment hierarchy, etc.) with other physical components, defining the physical architecture of the system of interest.			
21	View Conceptual Model[CPLNGS--CHAR]	Virtual Model System	55 2.3.9	For each modeled physical component, the model shall represent the attribute value couplings between the identity attributes for that physical component and the behavior characterization attributes of any logical role allocated to that component by the model.			
22	View Conceptual Model[CPLNGS--CHAR]	Virtual Model System	55 2.3.10	The modeled internal behavioral roles and couplings shall include any parasitic aspects which arise from choice of internal design, materials, technologies, or solution approach but which were not otherwise required by the primary intended system purpose, where significant from a stakeholder perspective.			
23	View Conceptual Model[PHYSARCH]	Virtual Model System	55 2.3.11	For each modeled functional role (element of behavior), the model shall represent an allocation of that role to a physical component which performs or has that behavior.			
24	View Conceptual Model[PHYSARCH]	Virtual Model System	55 2.3.12	The model shall represent allocation of each fully decomposed functional role to not more than one physical component.			
25	View Conceptual Model[PHYSARCH]	Virtual Model System	55 2.3.13	For each modeled physical component, material, or equipment item, the model shall represent the allocation of all functional roles (elements of behavior) expected of that physical component, material, or equipment item.			
26	View Conceptual Model[DECMP]	Virtual Model System	55 2.3.14	The model shall represent the behavioral modes (states) of the internal system white box roles that are significant to the intended use of the model.			
27	View Conceptual Model[DECMP]	Virtual Model System	55 2.3.15	The model shall represent the possible (state) transitions between the modeled internal behavioral modes.			
28	View Conceptual Model[DECMP]	Virtual Model System	55 2.3.16	For each of its modeled internal modes (states), the model shall represent which interactions of internal roles may occur during such modes.			
29	View Conceptual Model[DECMP]	Virtual Model System	55 2.3.17	For each Modeled Black box Requirement on the system of interest the model shall provide modeled White Box Requirements traceable to and decomposing that Black Box Requirement.			
30	Configure Model from Pattern	Virtual Model System	55 2.4.1	The model shall include configurability for different cases indicated.			
31	View Datasets[Model Validation Dataset]	Virtual Model System	55 2.5.1	The model shall include documented example, validation, and verification data sets, including model inputs, model outputs, and model configuration.			
32	View Datasets[Model Validation Dataset]	Virtual Model System	55 2.5.2	The model shall include task-specific datasets from previous model executions, allowing their further use without additional model execution runs.			
33	View Datasets[Model Validation Dataset]	Virtual Model System	55 2.5.3	The model run data sets shall satisfy [Data Set Structural] and [Data Set Accuracy] requirements.			
34	View Conceptual Model[FMEA]	Virtual Model System	55 2.6.1	The model shall include identification of component failure modes, as to underlying state leading to predicted failure.			
35	View Conceptual Model[FMEA]	Virtual Model System	55 2.6.2	For each identified failure mode, the model shall include identification of cause(s) of failure mode.			
36	View Conceptual Model[FMEA]	Virtual Model System	55 2.6.3	For each identified failure mode, the model shall include the probability of failure mode.			
37	View Conceptual Model[FMEA]	Virtual Model System	55 2.6.4	For each identified failure mode, the model shall include the effect(s) of the mode.			
38	View Conceptual Model[FMEA]	Virtual Model System	55 2.6.5	For each identified failure effect, the model shall include the severity of impact of the failure.			
39	View Conceptual Model[EXTRN]	Virtual Model System	55 3.1.2	The model shall represent the system of interest over a specified (discrete or continuous) range or envelope of technical external environment interaction configurations.			
40	View Conceptual Model[DECMP]	Virtual Model System	55 3.1.3	The model shall represent the system of interest over a specified (discrete or continuous) range or envelope of physical design configurations.			

Filter is not applied. 77 rows are displayed in the table. [Show Description](#)

Notification Window

Ready