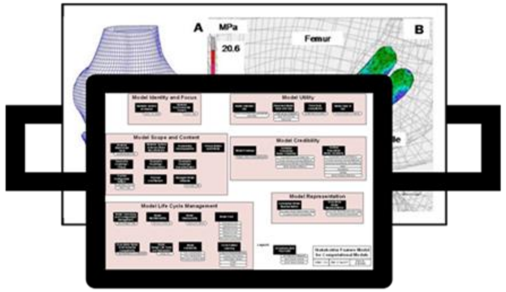
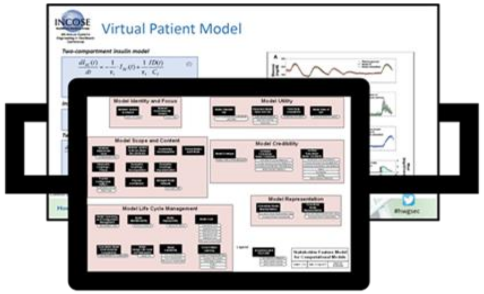
The Model Characterization Pattern (MCP)

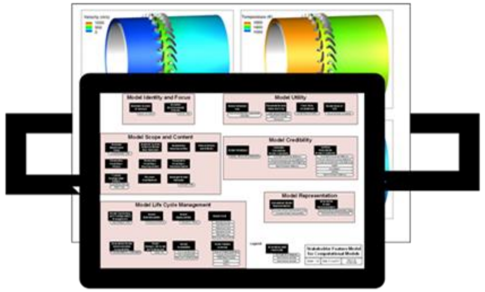
FEA Model



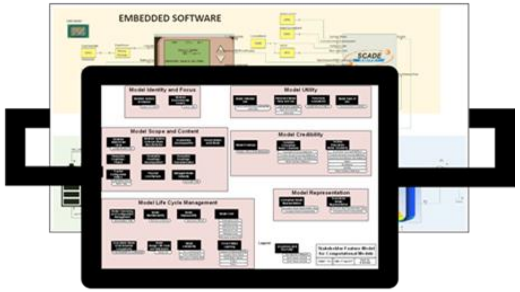
ODE Model

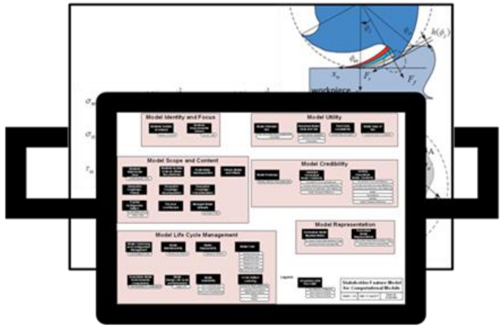


CFD Model

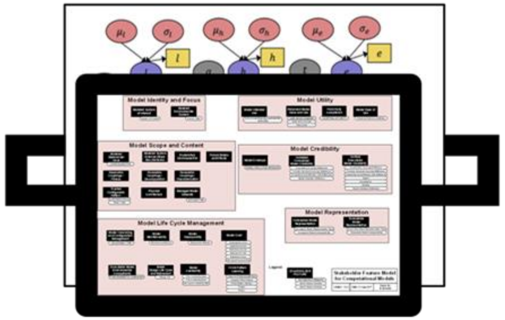


Multi-Domain System Model

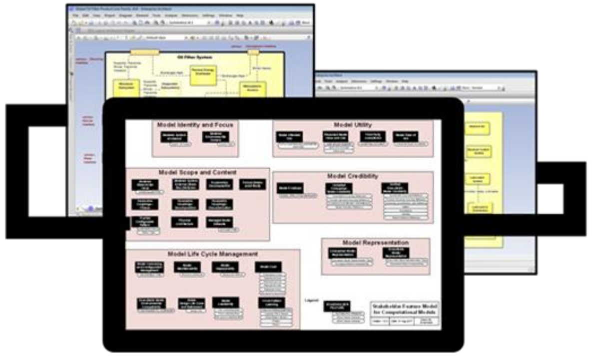




Physics-Based PDE Model



Data-Driven Bayesian Network Model



MBSE Model

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

V 1.8.1

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: S*Models, S*Patterns, S*Metamodel
- Appendix IV: ASELCM S*Pattern, Trusted Models, Effective Group Learning

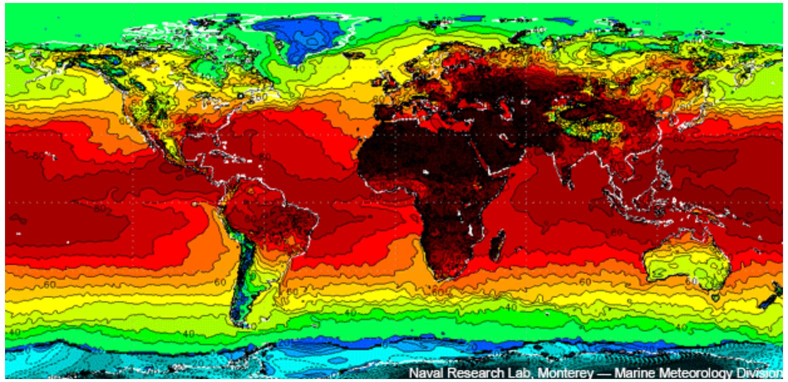
Origins: A Community Effort

- 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.

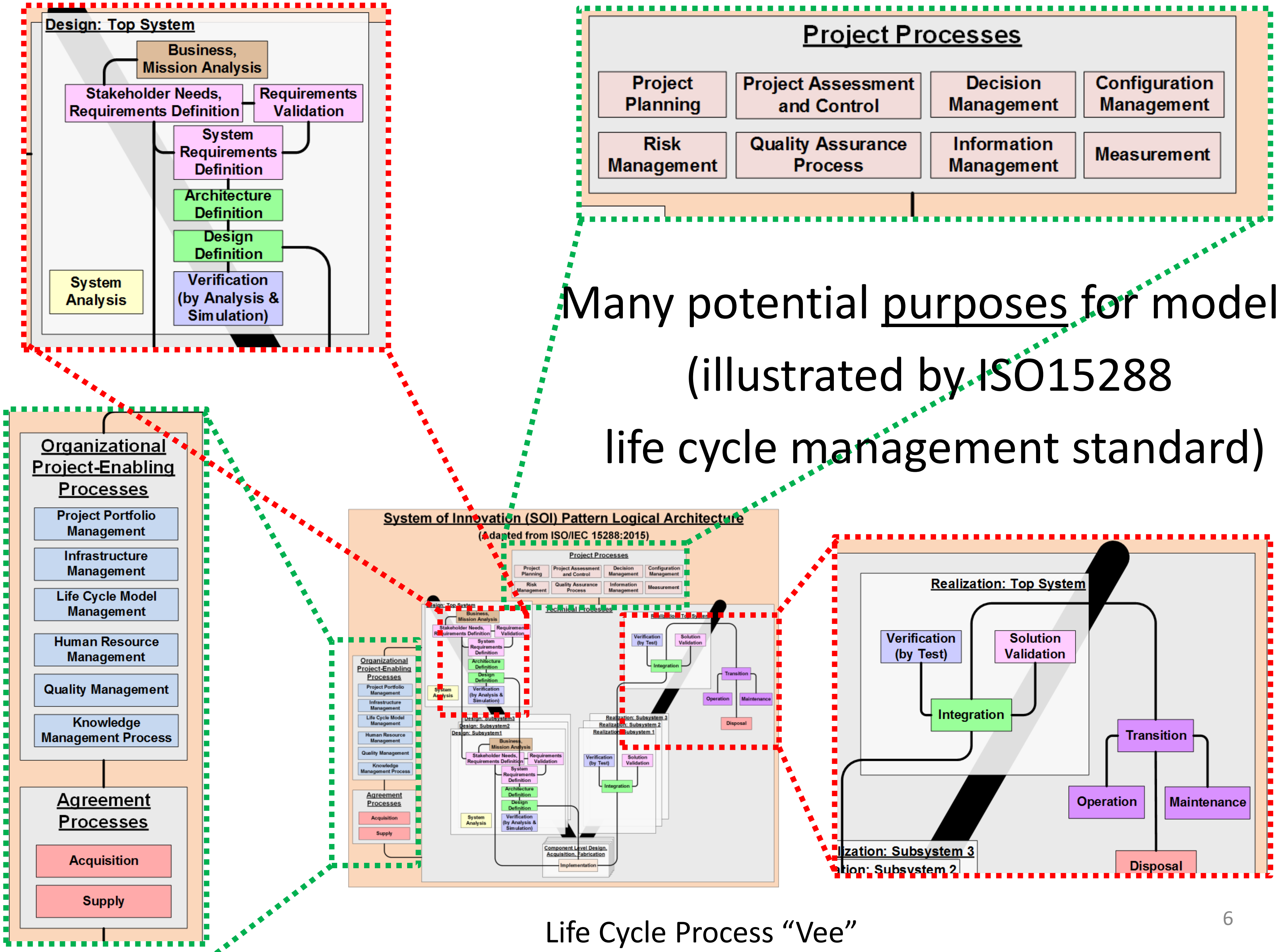
An Increasingly Model-Based World

- If we expect to use models to support critical decisions, then we are placing <u>*increased trust in models*</u>:
  - Critical financial, other business decisions
  - Human life safety
  - Societal impacts
  - Extending human capability

- Requires that we <u>characterize the nature of that trust</u> and manage its award, use:
  - The Validation, Verification, and Uncertainty Quantification (VVUQ) <u>*of the models themselves*.</u>



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

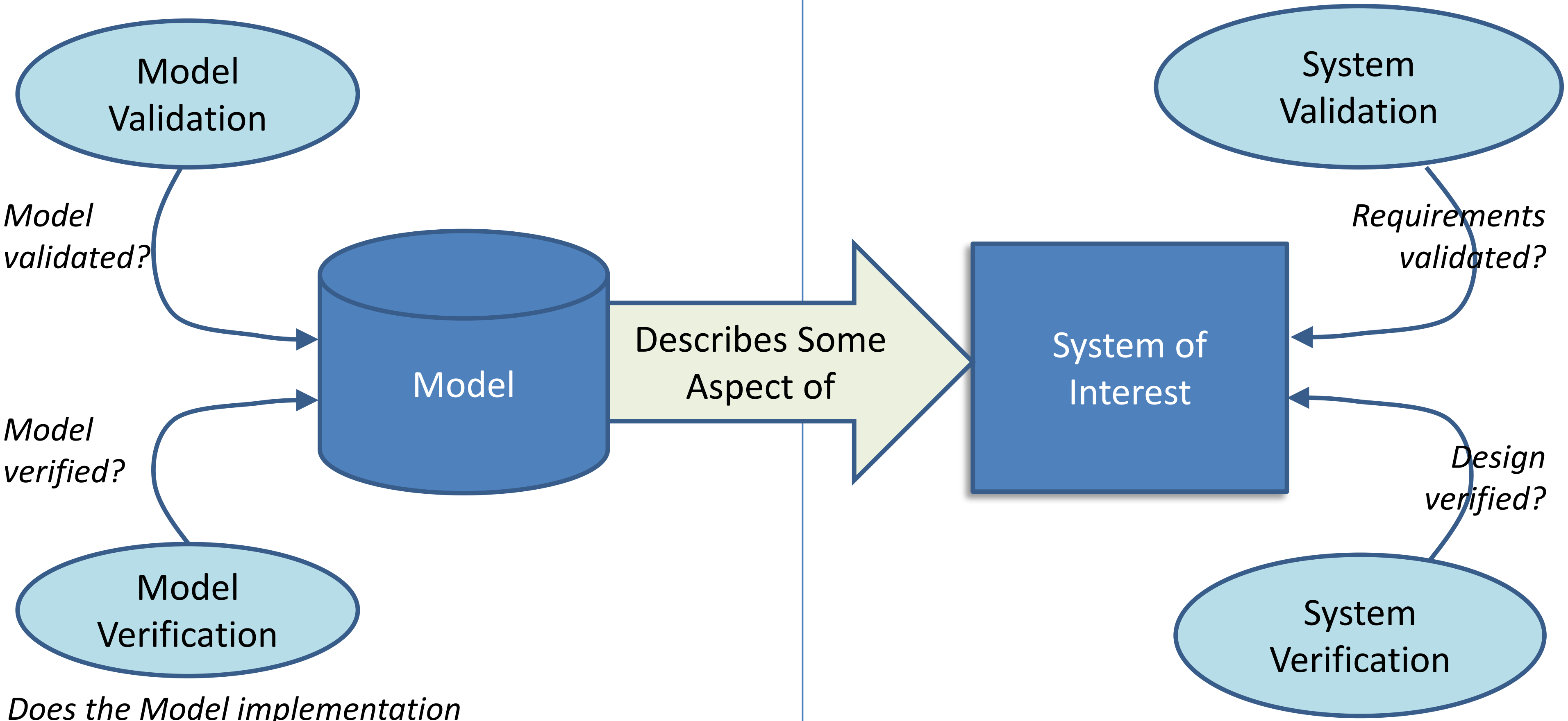
Does the Model adequately describe what it is intended to describe?

Does the Model implementation adequately represent what the Model says?

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

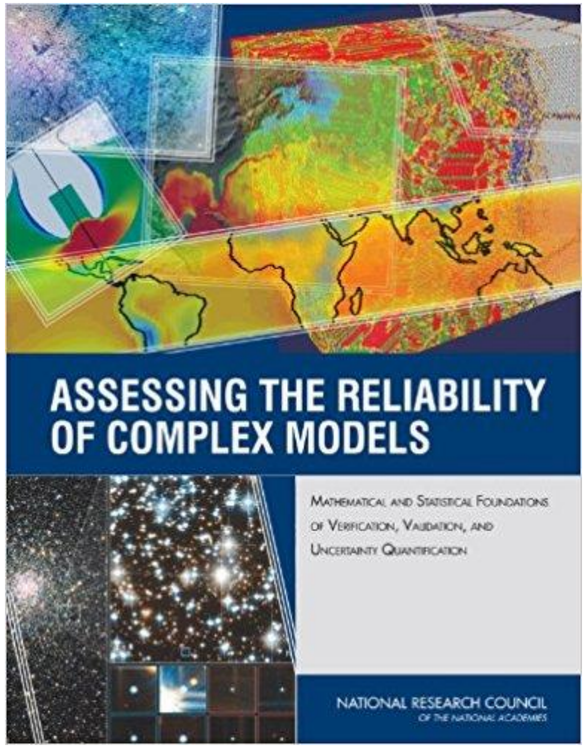
Do the System Requirements describe what stakeholders need?

Does the System Design define a solution meeting the System Requirements?



Don't forget: A model (on the left) <u>may</u> 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 <u>systems</u> 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 <u>*qualitatively*</u> cover the modeled Stakeholder Features over the range of intended subject system situations of interest?
- Quantify confidence / uncertainty that the modeled <u>*Stakeholder Feature Attributes*</u> 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 <u>*Technical Performance Attributes*</u> 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)

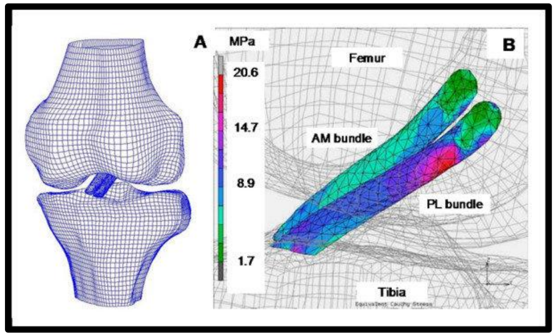
Challenges for Model Stakeholders

- 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 . . . .

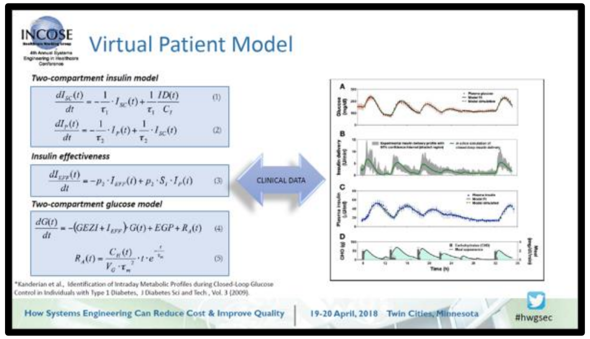
1. ***Scaling up*** *to the population of people and volume of models and model transactions to be addressed in a world in which these will grow by orders of magnitude, overwhelming what might not otherwise be addressed by a more limited population of deeply expert model authors, model users, or model 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

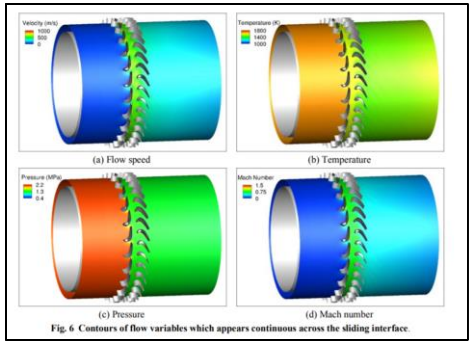
FEA Model



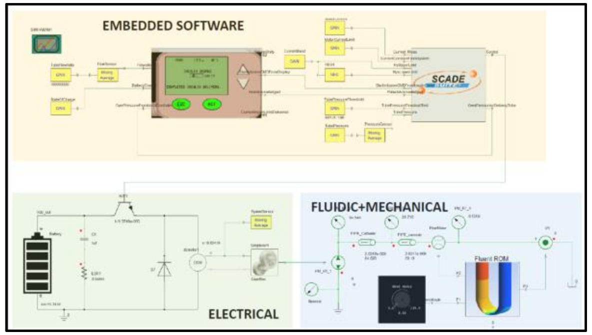
ODE Model

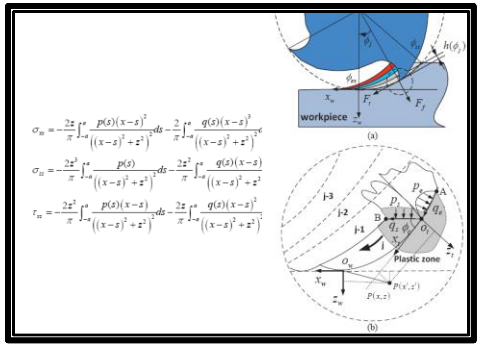


CFD Model

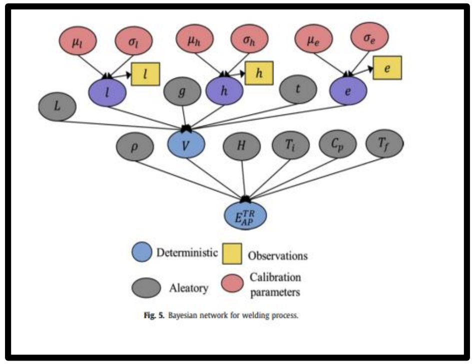


Multi-Domain System Model

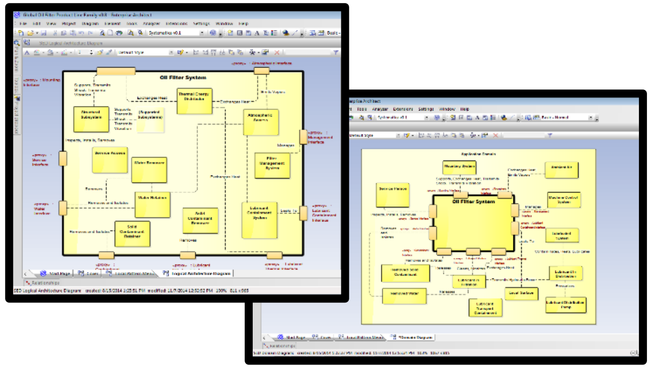




Physics-Based PDE Model



Data-Driven Bayesian Network Model



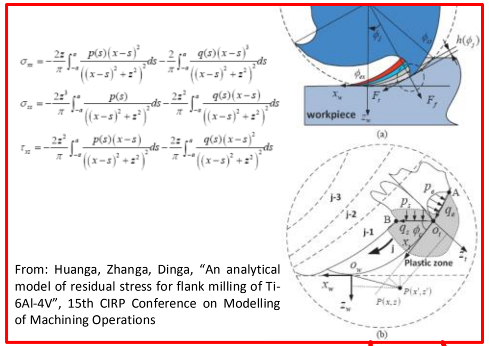
MBSE Model

Physics-Based Model

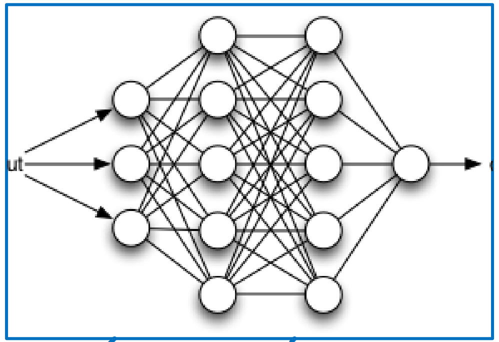
- Predicts the external behavior of the System of Interest, visible externally to the external actors with which it interacts.
- Models internal physical interactions of the System of Interest, and how they combine to cause/explain externally visible behavior.
- Model has both external predictive value 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.



From: Huanga, Zhanga, Dinga, "An analytical model of residual stress for flank milling of Ti-6Al-4V", 15th CIRP Conference on Modelling of Machining Operations



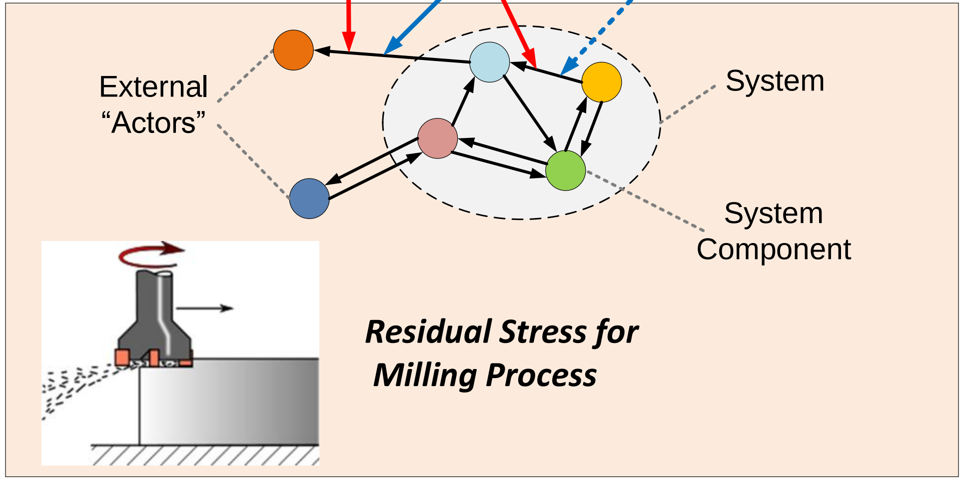
- Physical scientists and phenomena models from their disciplines can apply here.
- The hard sciences physical laws, and how they can be used to explain the externally visible behavior of the system of interest.

predicts, explains

predicts

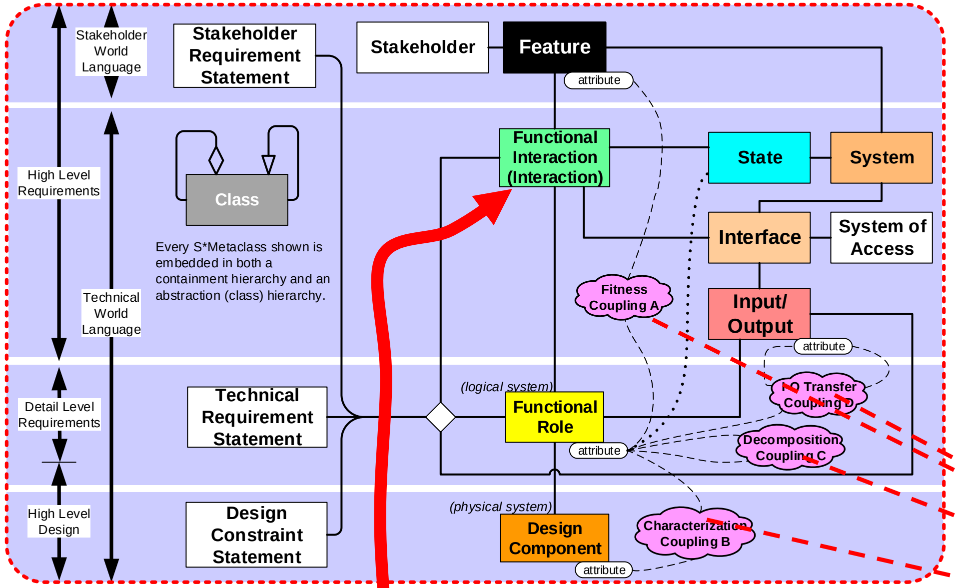
optional

- 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



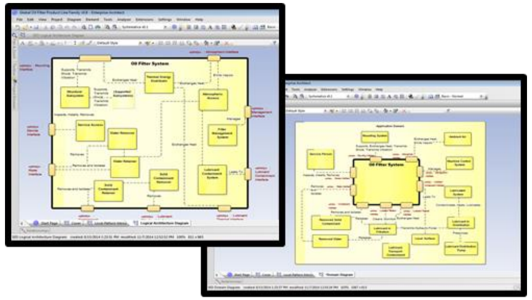
S*Metamodel informal summary pedagogical diagram

(formal S*Metamodel includes additional details.)

Appendix III

Attribute Coupling

System Model



MBSE Model

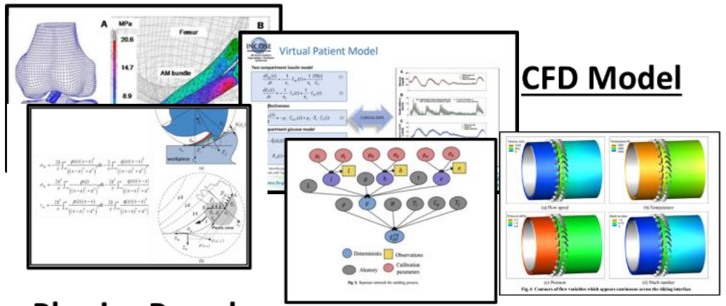
PIRT

Computational Model

FEA Model

ODE Model

CFD Model

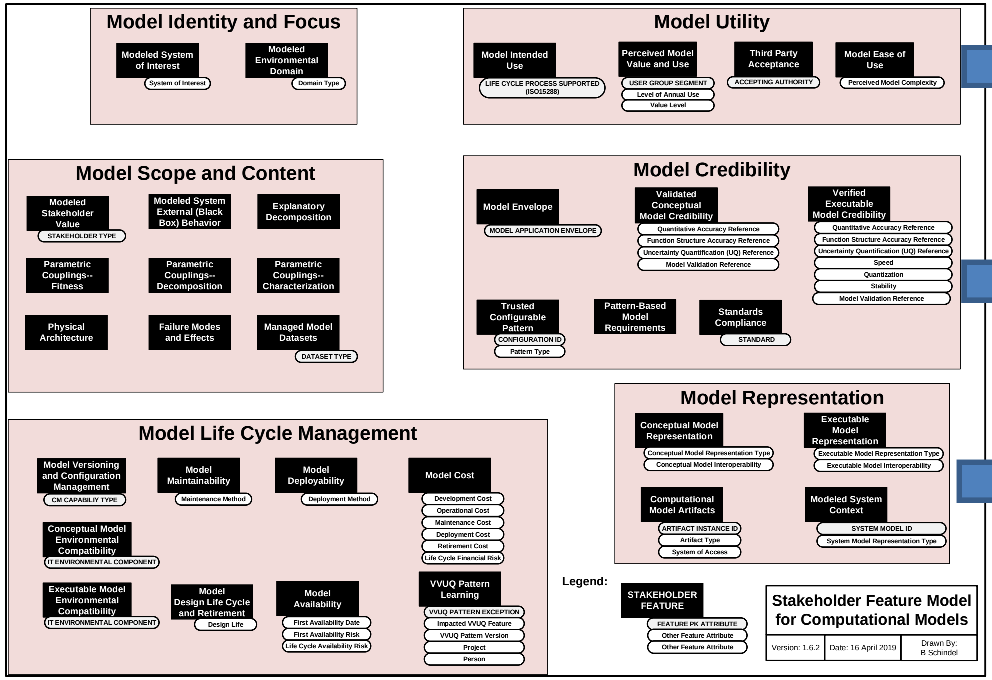


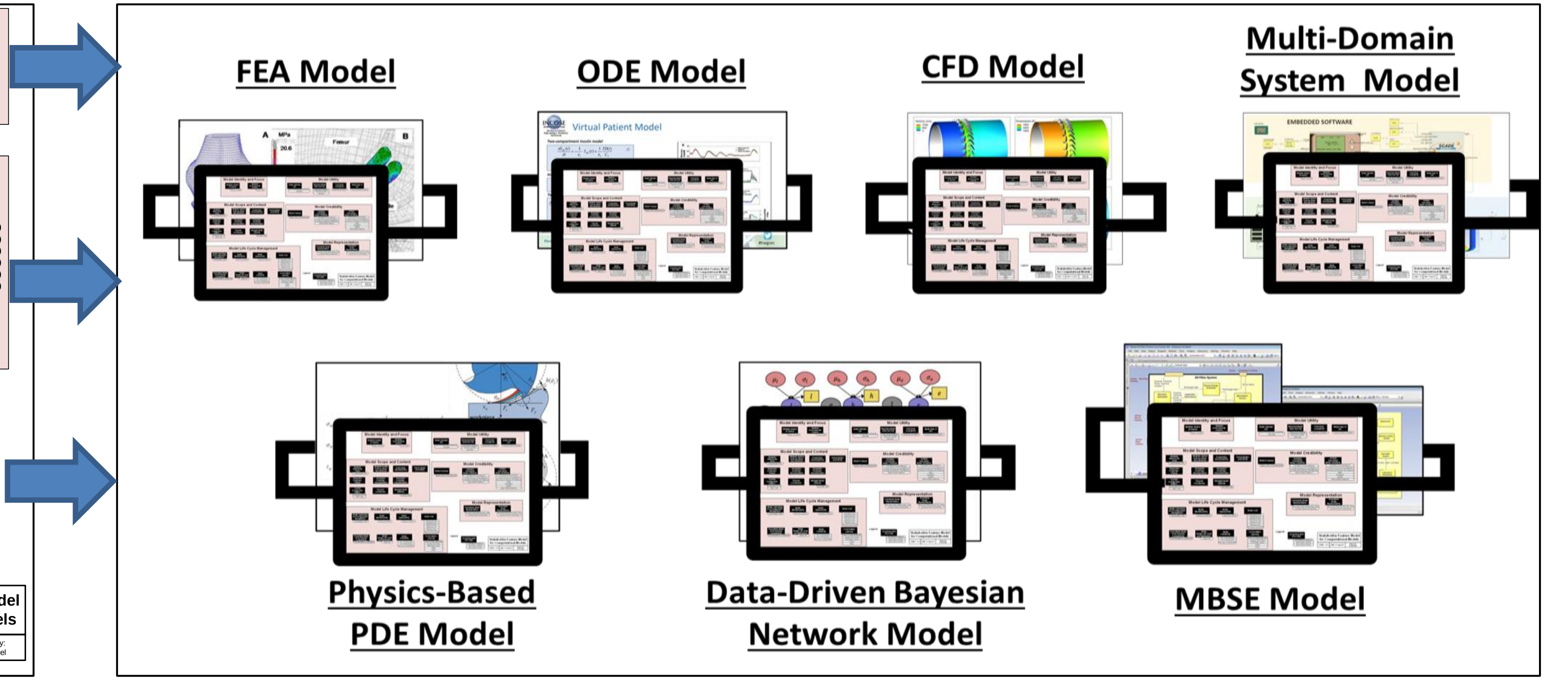
Physics-Based PDE Model

Data-Driven Bayesian Network Model

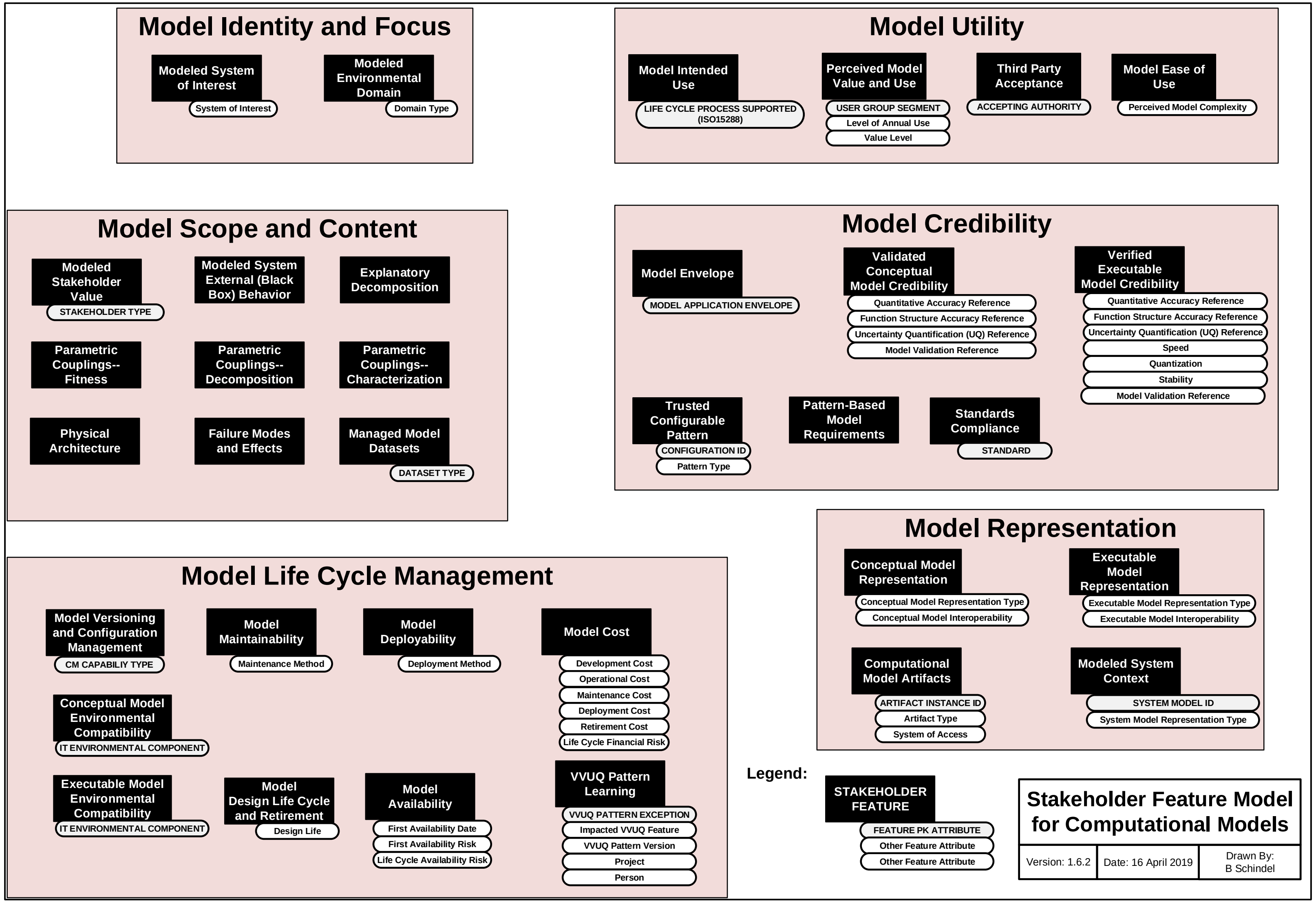
The Model Characterization Pattern (MCP)—an S*Pattern

- 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.



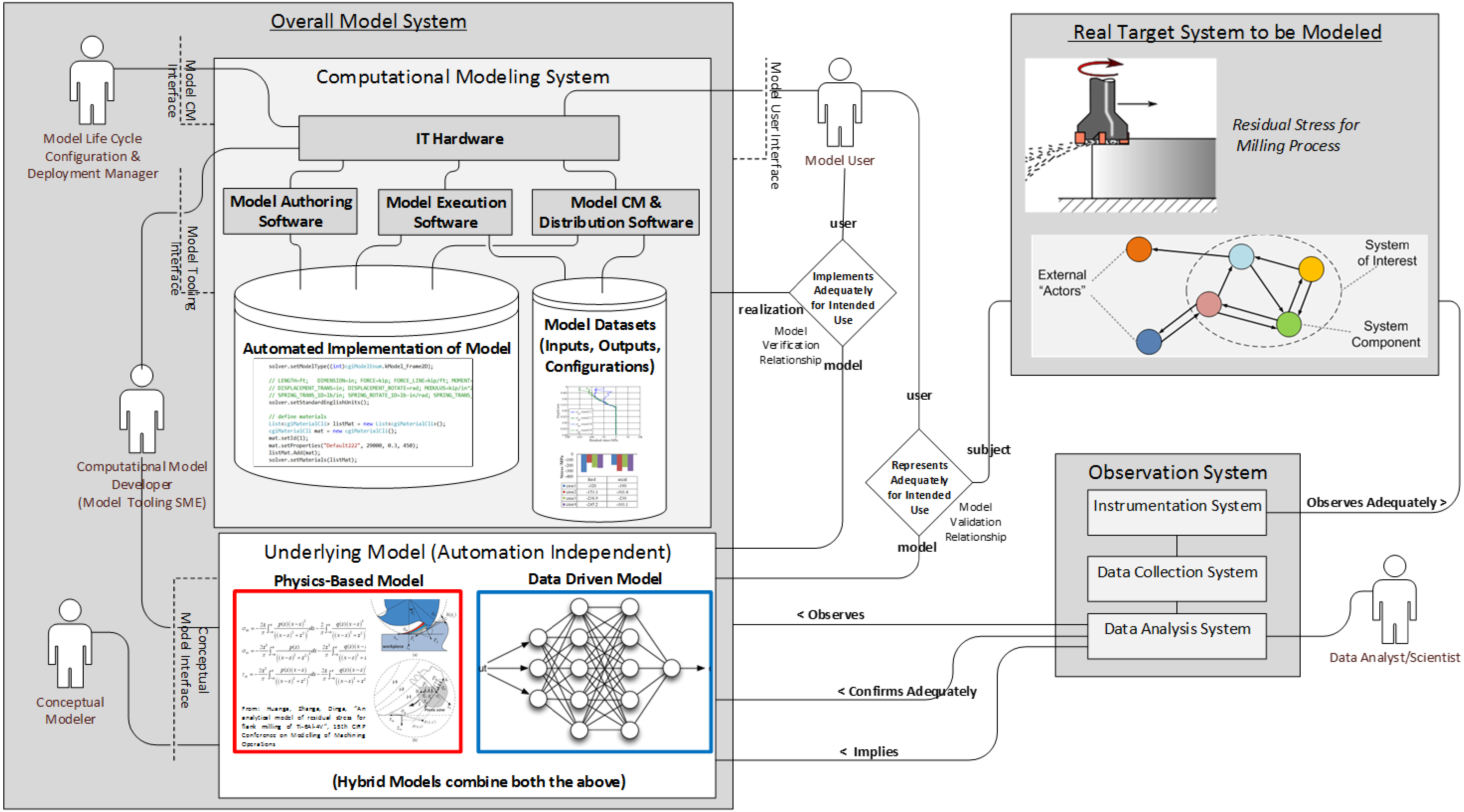


Configurable MCP Feature Groups for Models (Computational Model's Stakeholder Requirements)

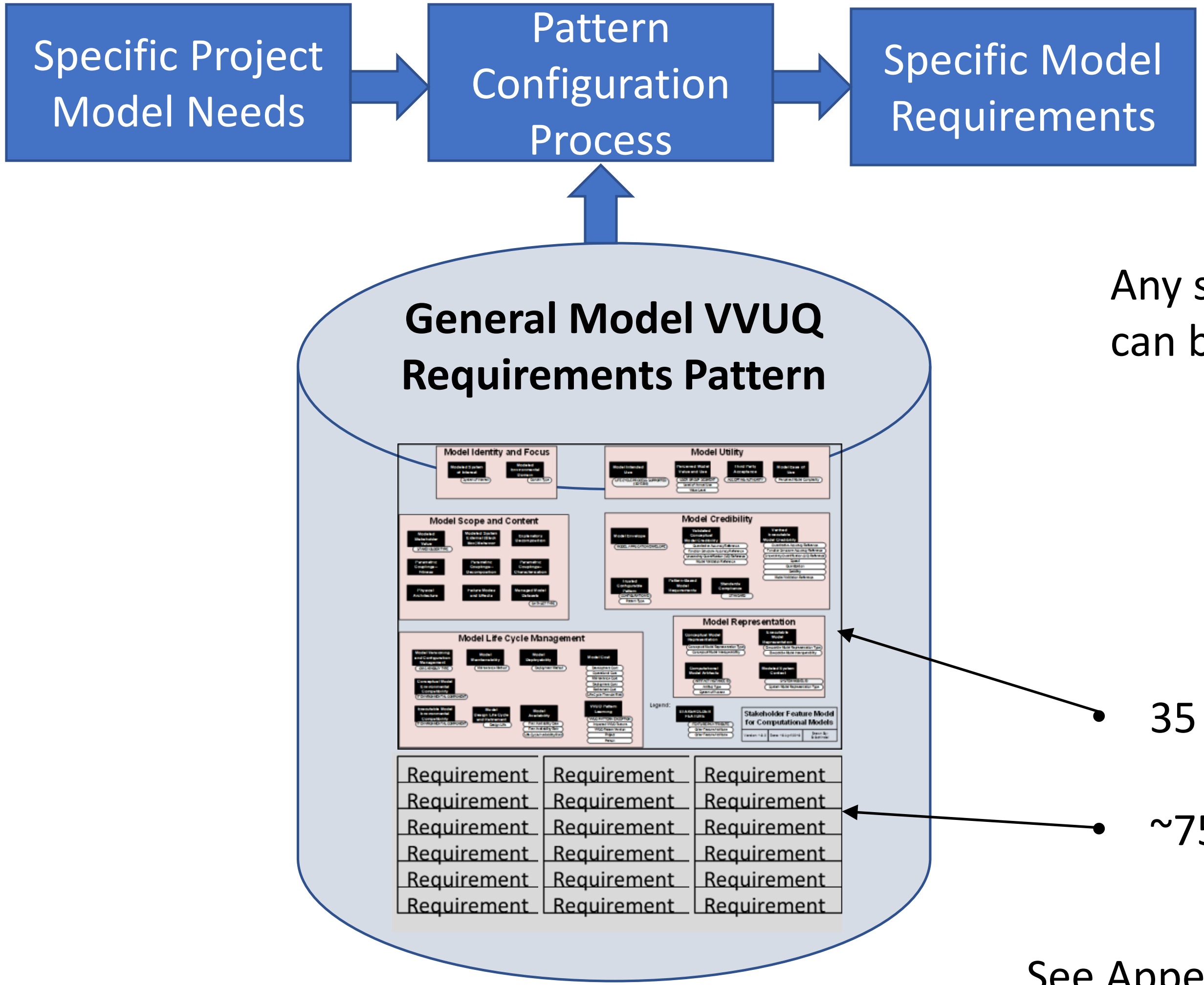


(See Appendix I for definitions of each Feature shown.)

Configurable MCP Domain Pattern for Model Systems



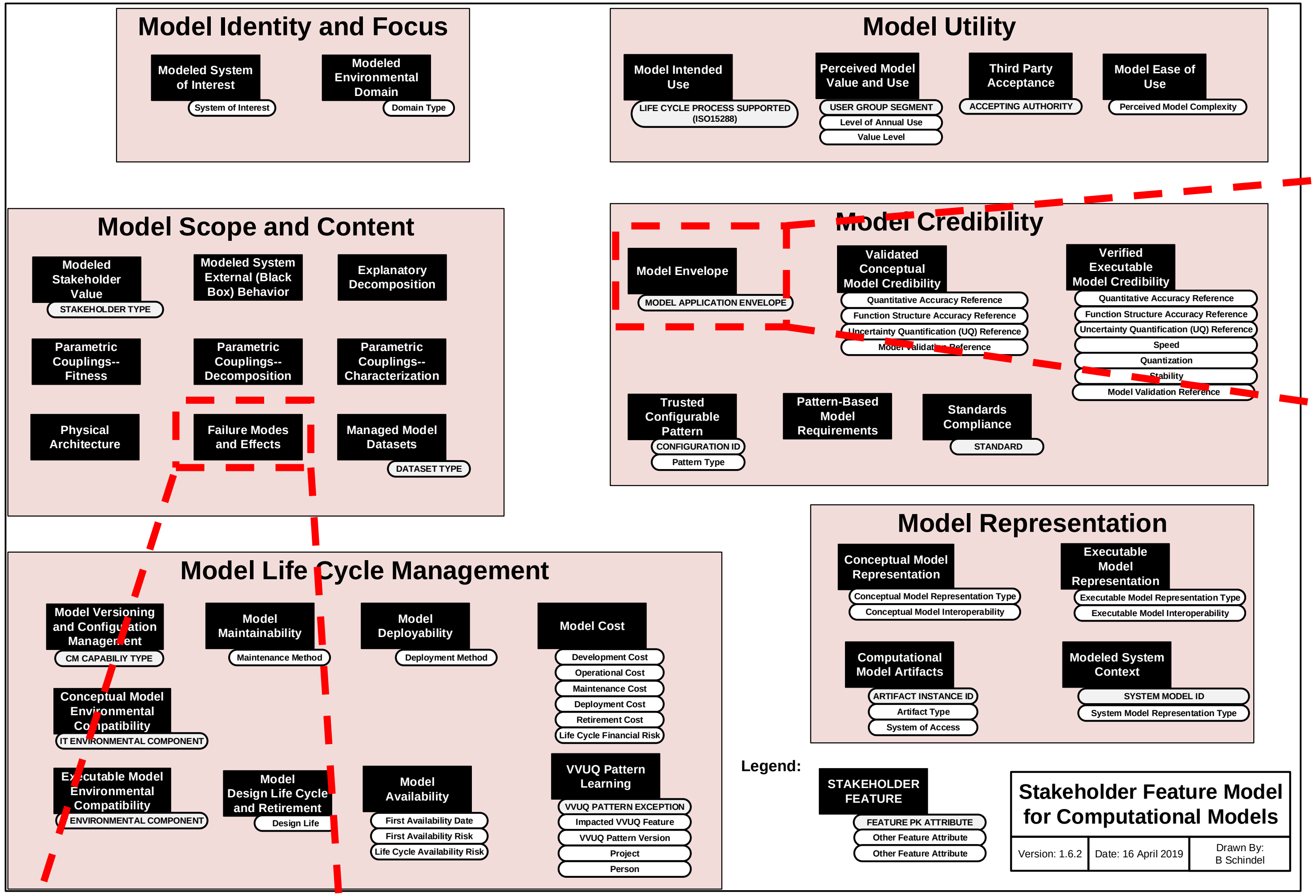
Configurable MCP Technical Requirements for Models



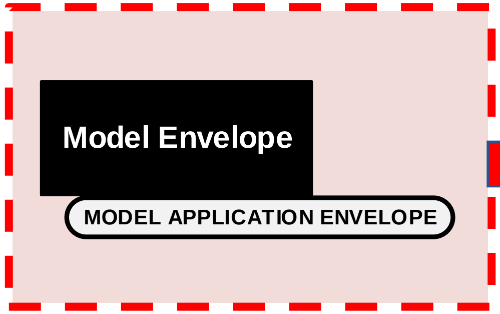
Any system modeling tool enabled with the S*Metamodel can be used for this process.

- 35 Model Stakeholder Requirements
- ~75 Model Technical Requirements

See Appendix II for each of the Technical Requirements)



Model Feature, from Configurable VVUQ Pattern



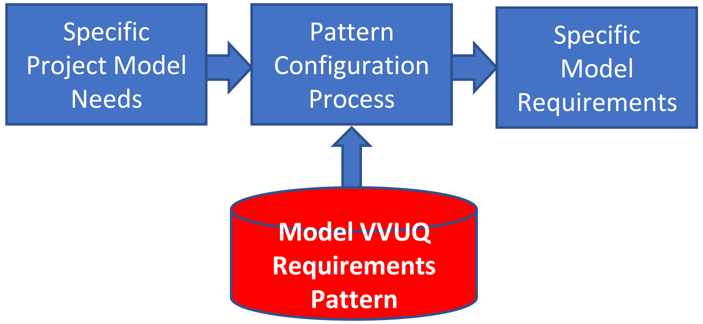
Model Requirement, from Configurable VVUQ Pattern

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."

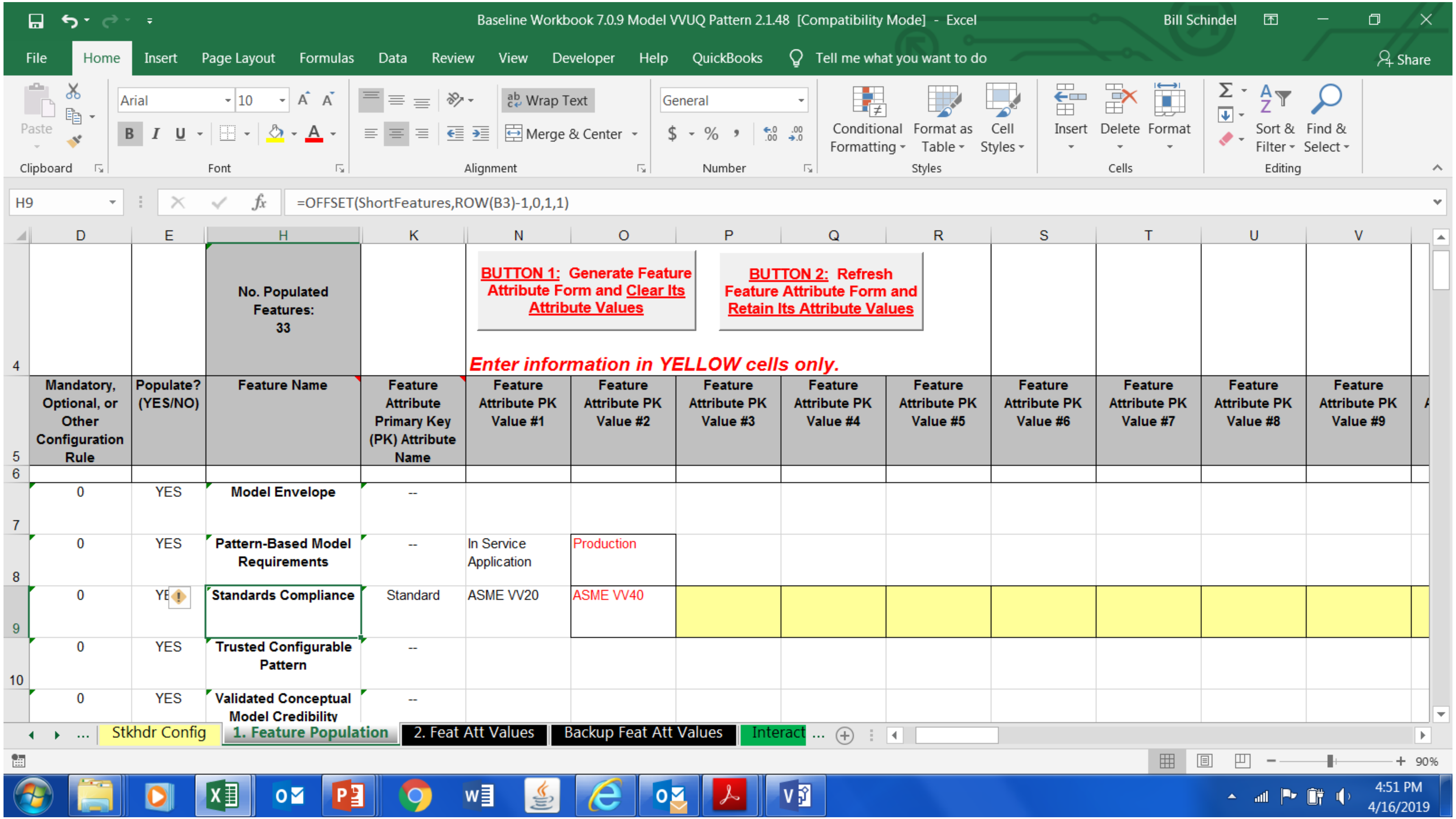
Model Requirement, from Configurable VVUQ Pattern

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

Model Feature, from Configurable VVUQ Pattern



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

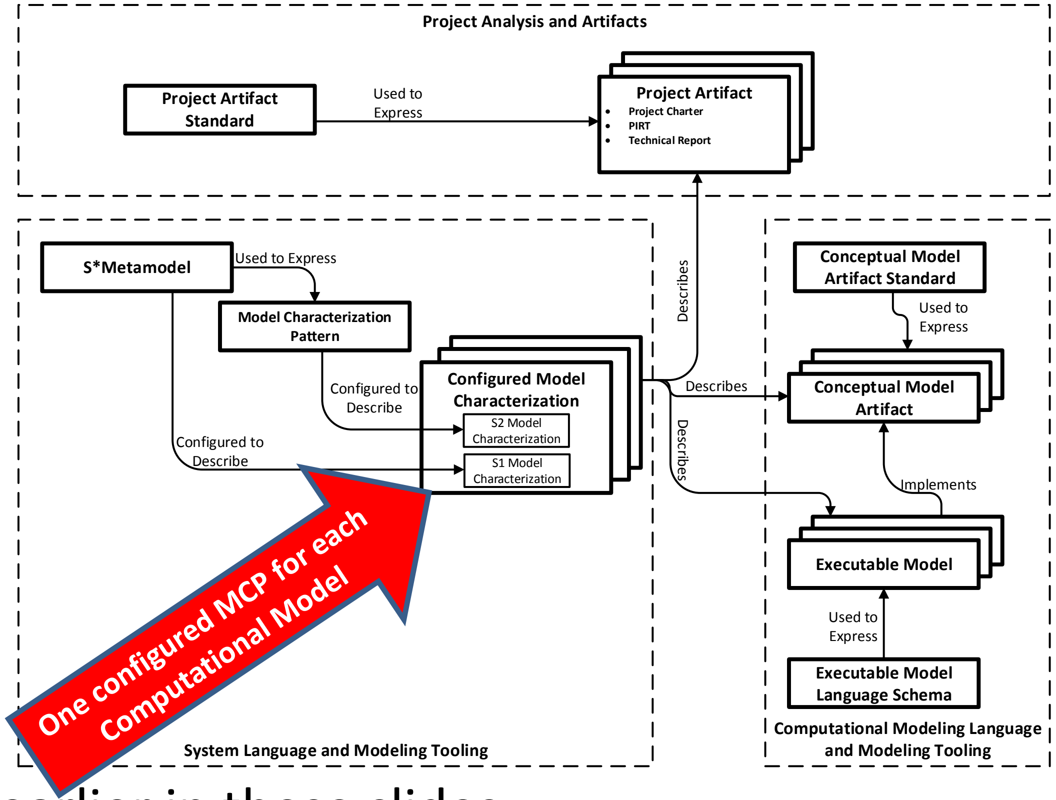


What you can do with the MCP in Computational Model Connected Projects and Enterprises

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

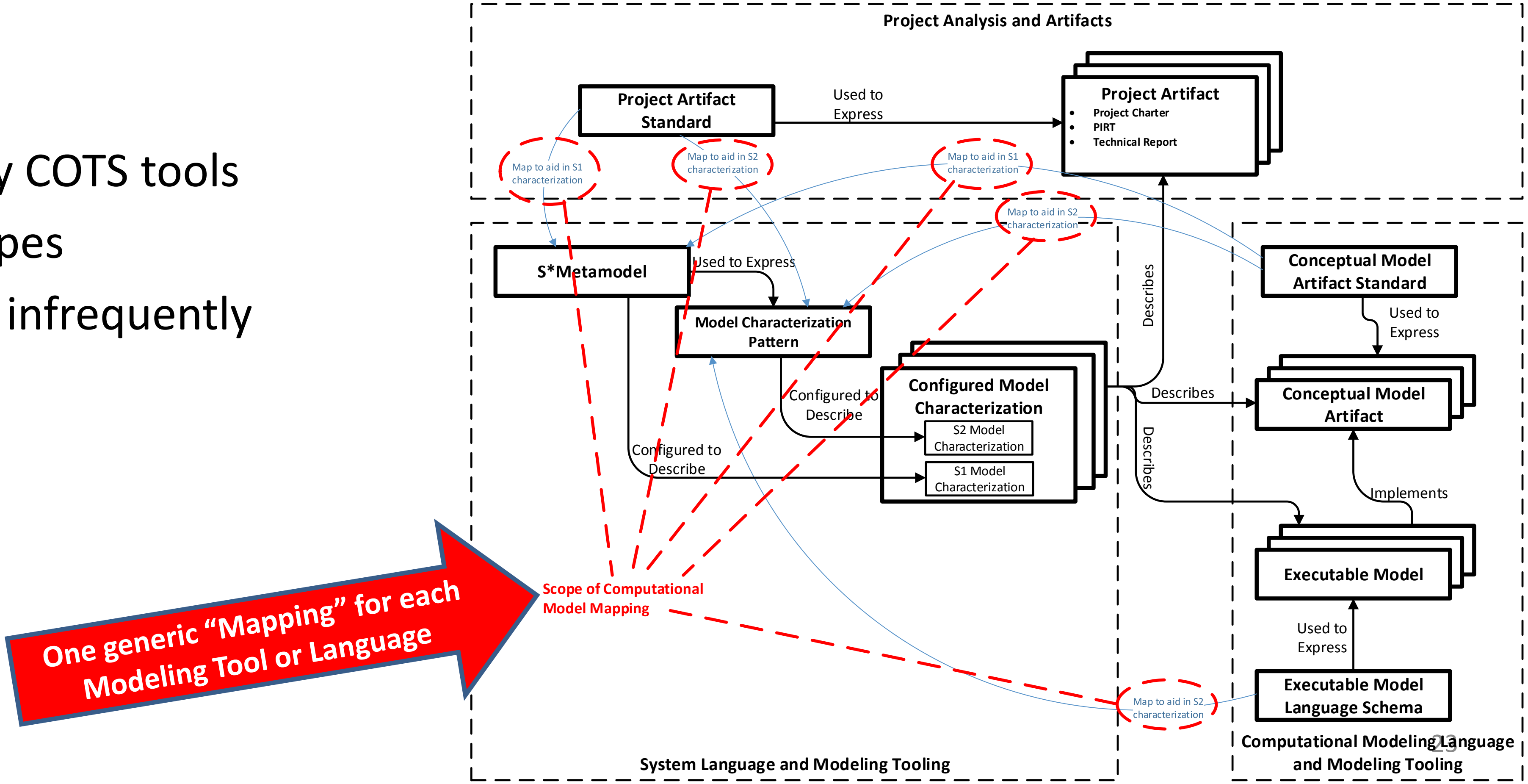
- A configured MCP for some computational model will typically refer to diverse information elements and artifacts of the model life cycle:
  - Project charter
  - Model Stakeholder and Technical Requirements
  - The conceptual model and executable model
  - Model authoring tools
  - Model execution 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.

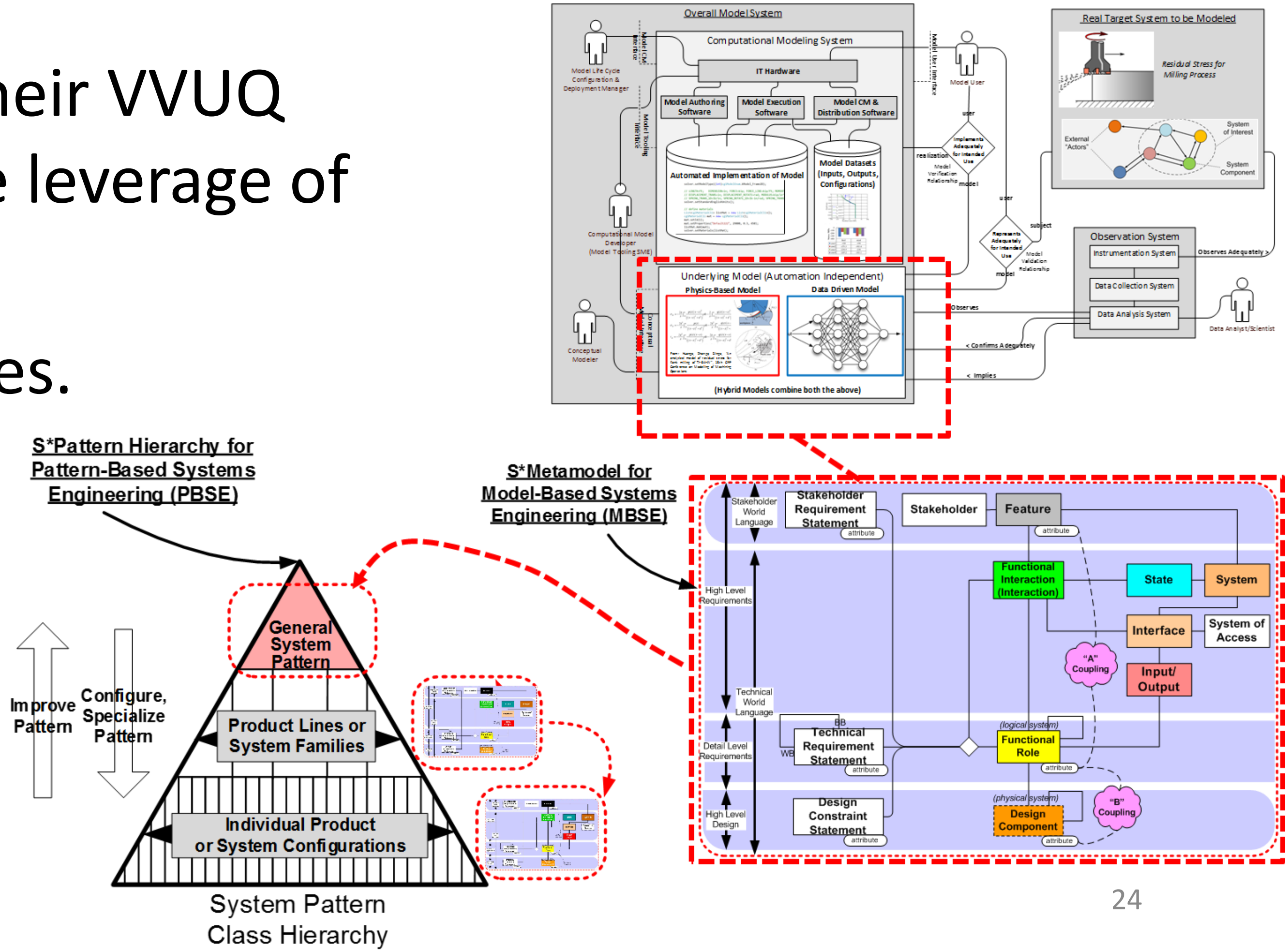
Infrastructure: Mapping the Model Characterization S*Pattern to specific enterprise practices, tools, languages, model types

- The overall process or environment owner may also provide a standard "mapping" from the <u>general</u> 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

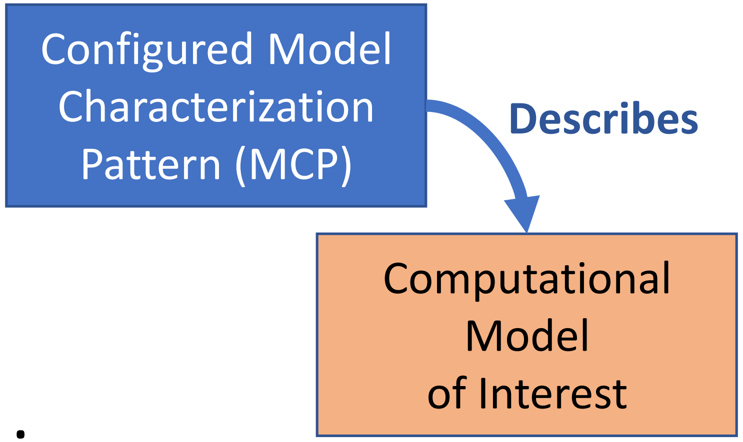


System of Interest Patterns: Accumulating Trustable Model-Based Knowledge

- 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.

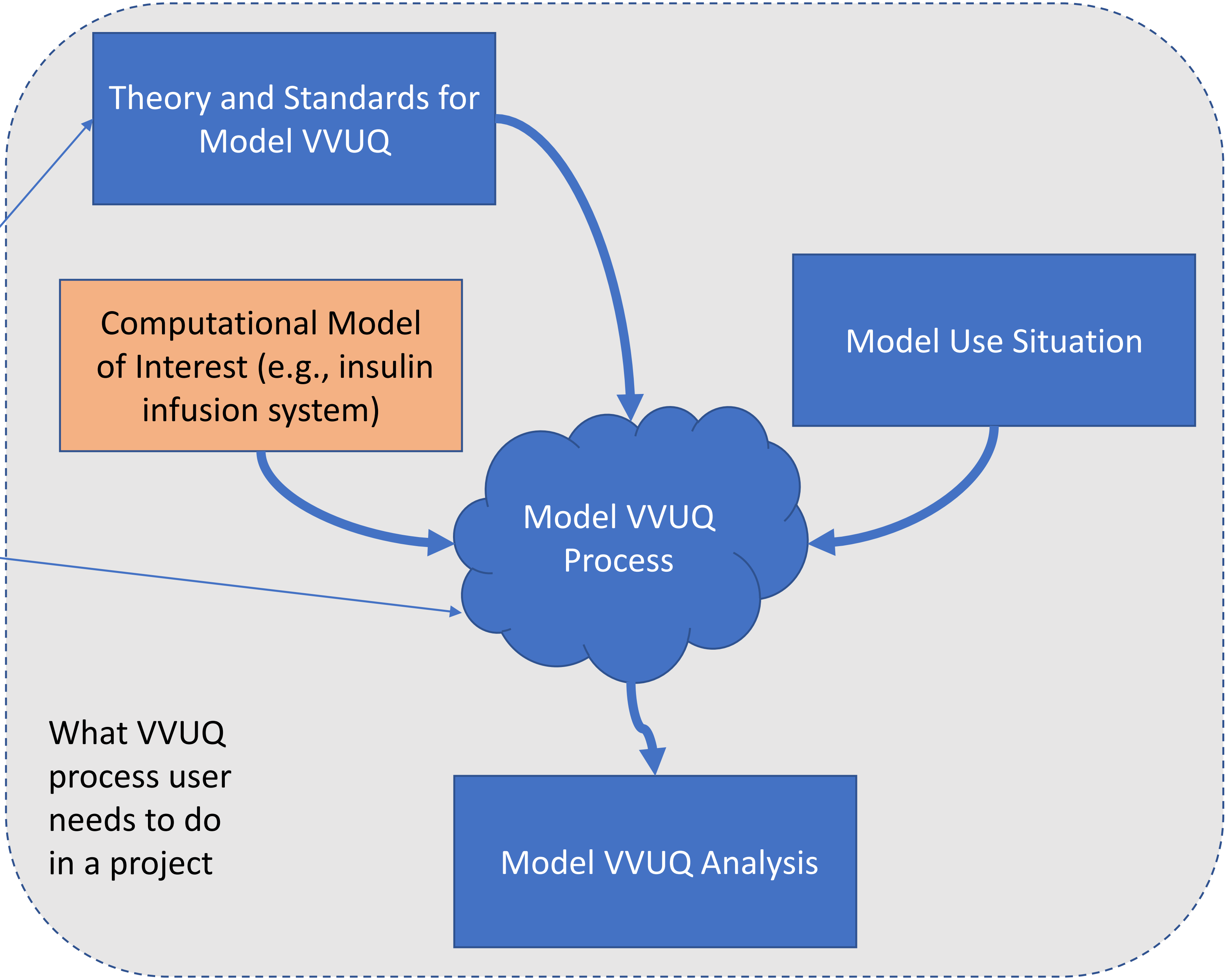


Leveraging Model VVUQ Theory to Leverage the Economics of Trusted Model-Based Patterns



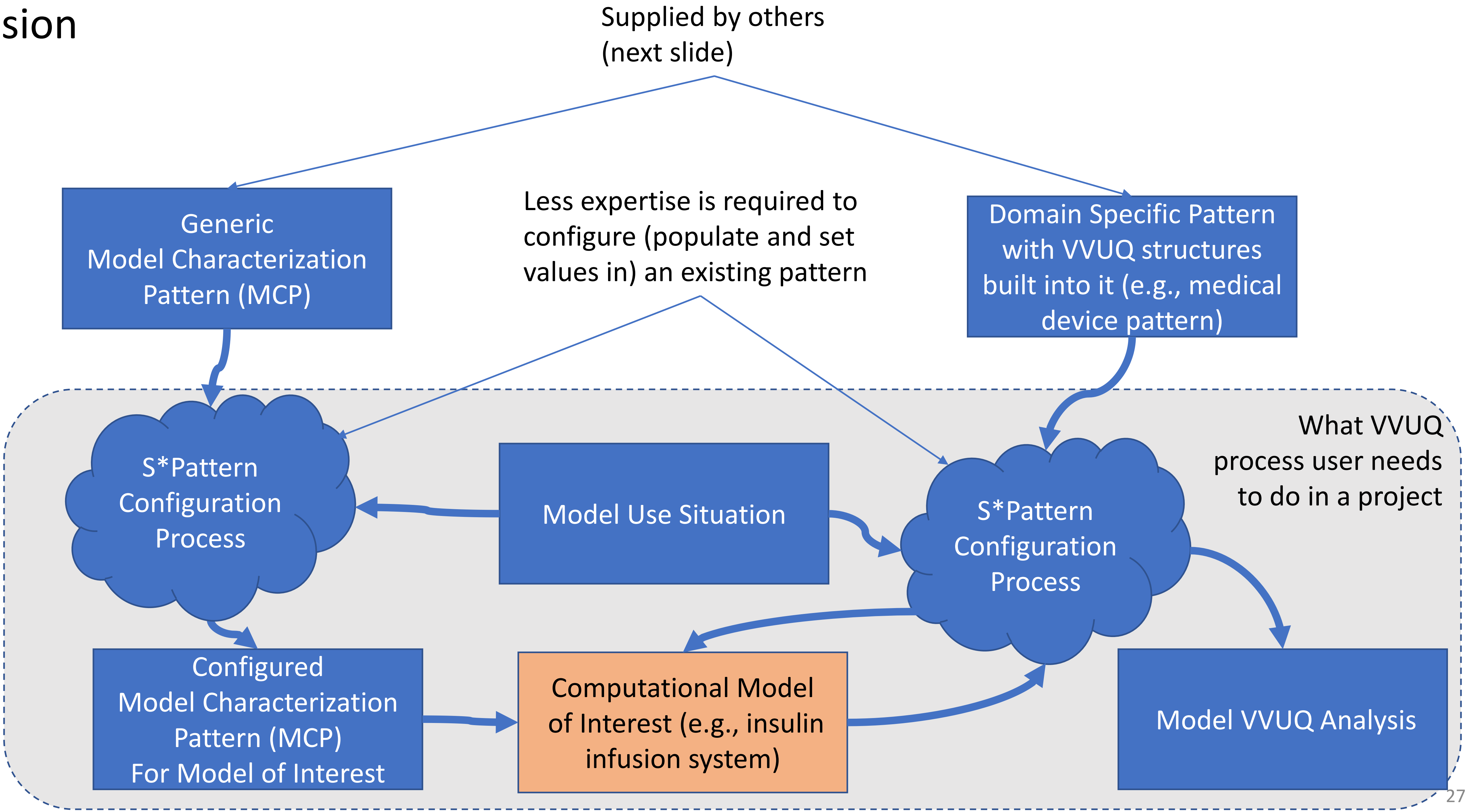
- "Models of computational models" may sound odd, so . . .
- <u>*Why are we creating S*Models of computational models of interest?*</u>
  1. To package decades of rich and valuable historical progress in theory of, and standards for, scientific model verification, validation, and uncertainty quantification . . . .
     - 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

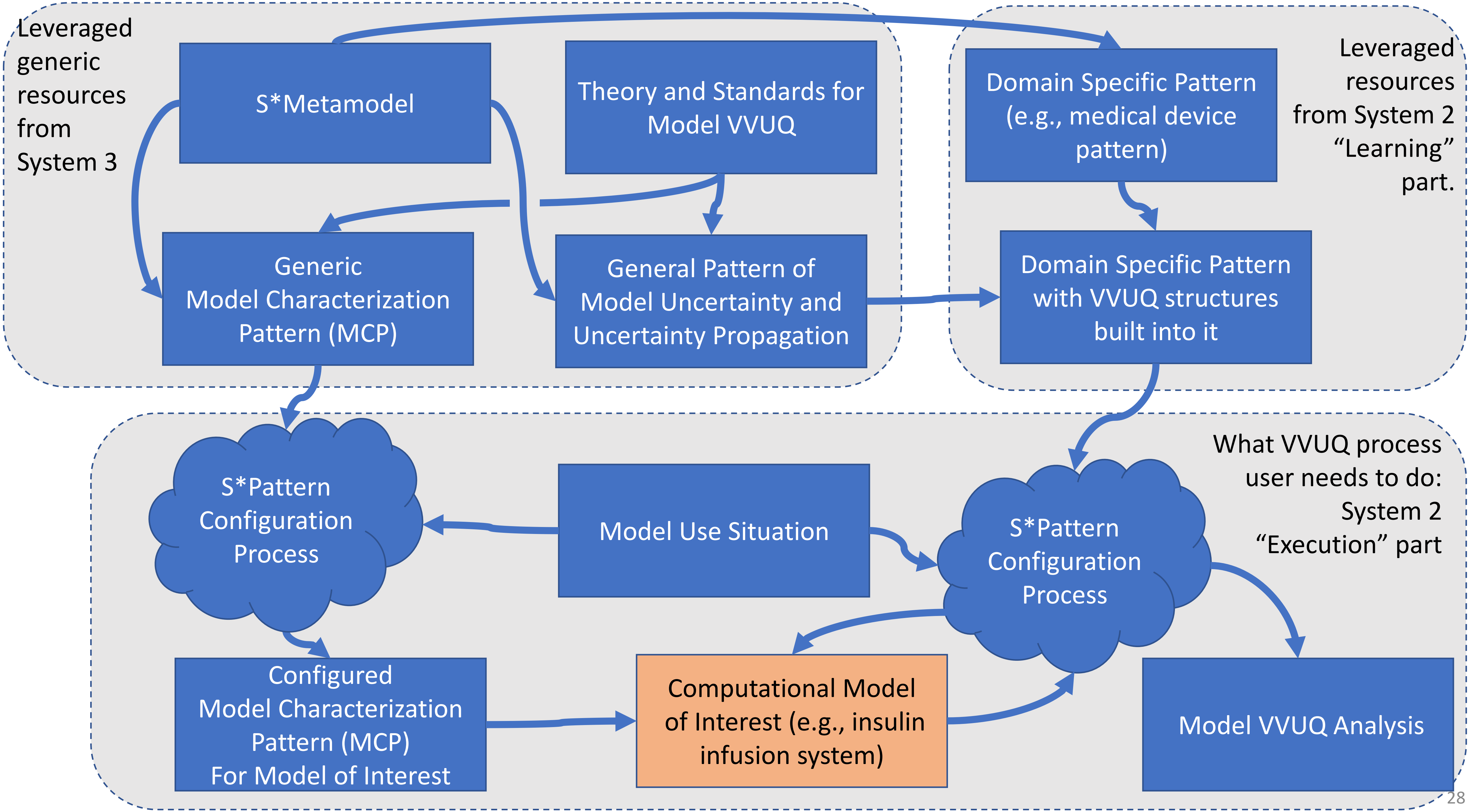


Expertise in these two areas may typically be limited. Practitioner knows more about Model Use Situation and Computational Model of Interest.

Vision



Supplied by others
(next slide)
Generic
Model Characterization
Pattern (MCP)
Less expertise is required to
configure (populate and set
values in) an existing pattern
Domain Specific Pattern
with VVUQ structures
built into it (e.g., medical
device pattern)
What VVUQ
process user needs
to do in a project
S*Pattern
Configuration
Process
Model Use Situation
S*Pattern
Configuration
Process
Configured
Model Characterization
Pattern (MCP)
For Model of Interest
Computational Model
of Interest (e.g., insulin
infusion system)
Model VVUQ Analysis


Leveraged generic resources from System 3
S*Metamodel
Theory and Standards for Model VVUQ
Domain Specific Pattern (e.g., medical device pattern)
Leveraged resources from System 2 “Learning” part.
Generic Model Characterization Pattern (MCP)
General Pattern of Model Uncertainty and Uncertainty Propagation
Domain Specific Pattern with VVUQ structures built into it
S*Pattern Configuration Process
Model Use Situation
S*Pattern Configuration Process
What VVUQ process user needs to do: System 2 “Execution” part
Configured Model Characterization Pattern (MCP) For Model of Interest
Computational Model of Interest (e.g., insulin infusion system)
Model VVUQ Analysis

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

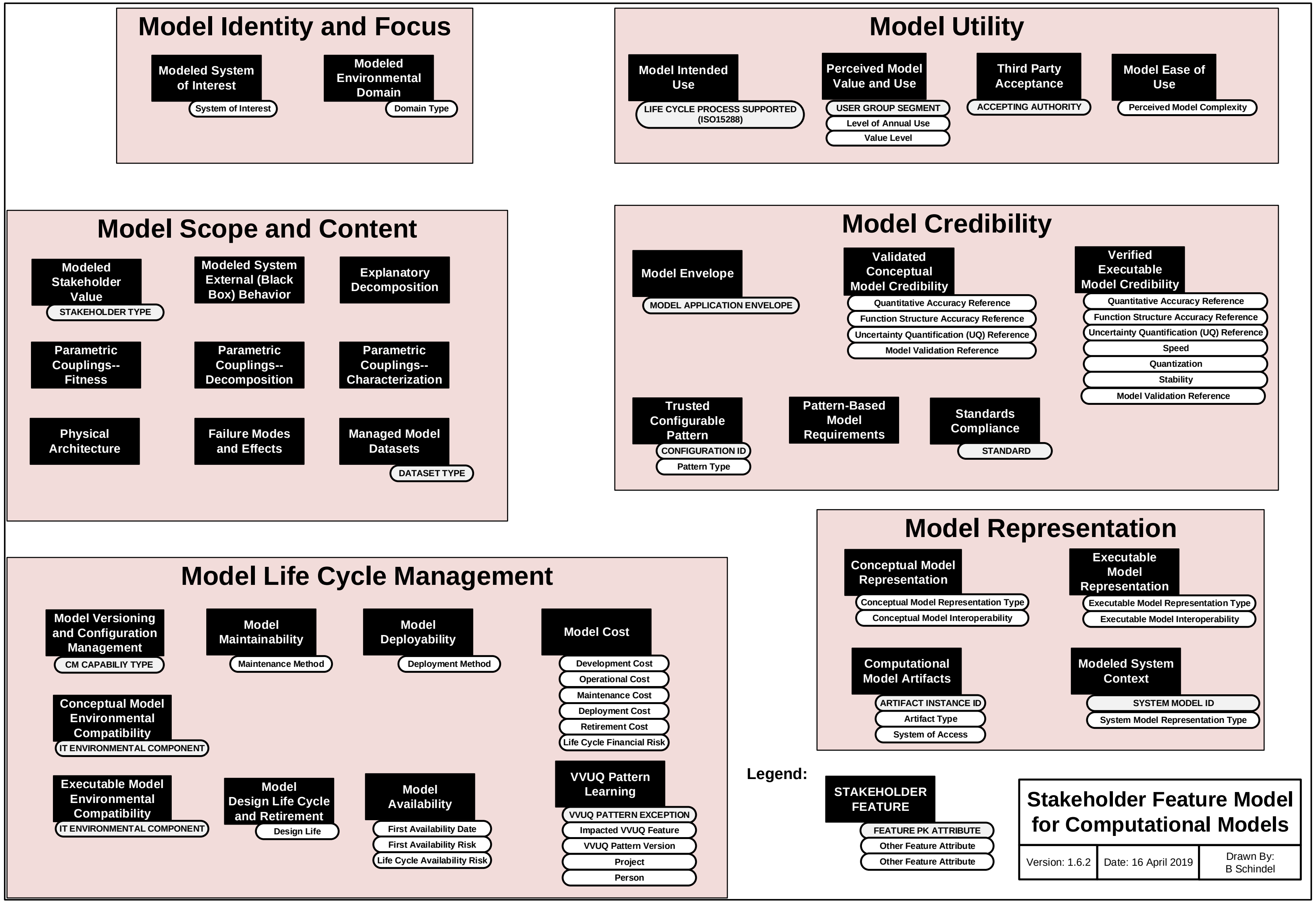
3. INCOSE MBSE Patterns Working Group, "MBSE Methodology Summary: Pattern-Based Systems Engineering (PBSE), Based On S*MBSE Models", V1.5.5A, retrieve from: http://www.omgwiki.org/MBSE/doku.php?id=mbse:pbse

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

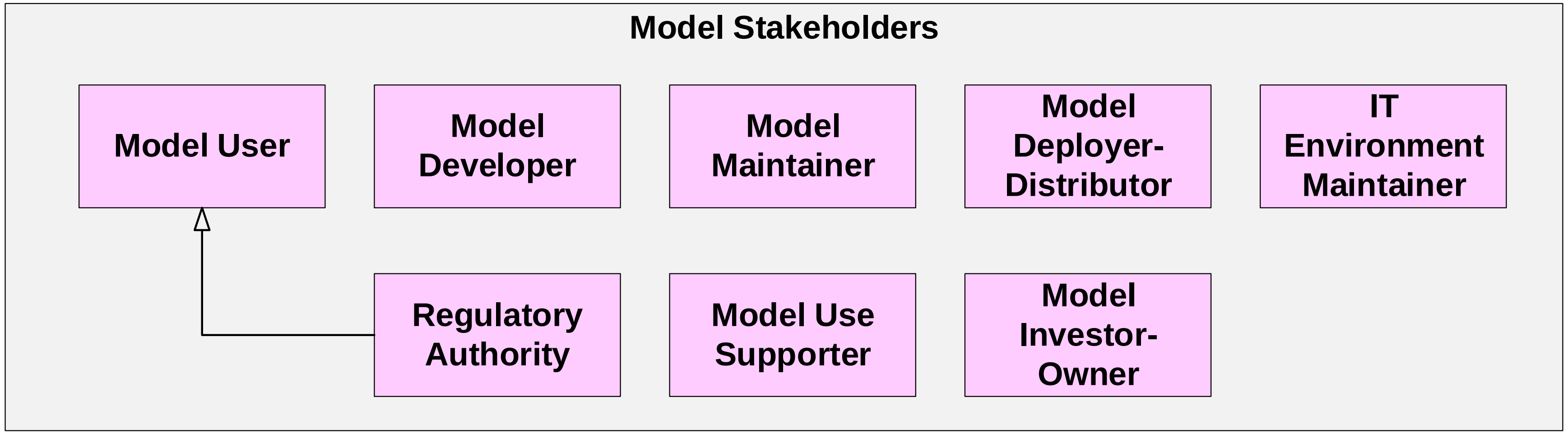
5. Schindel, W., "What Is the Smallest Model of a System?", *Proc. of the INCOSE 2011 International Symposium, International Council on Systems Engineering* (2011). Retrieve from: https://www.researchgate.net/publication/283141249_What_Is_the_Smallest_Model_of_a_System

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



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 Scope and Content

Describes the scope of content of the model.

Model Credibility

Describes the credibility of the model.

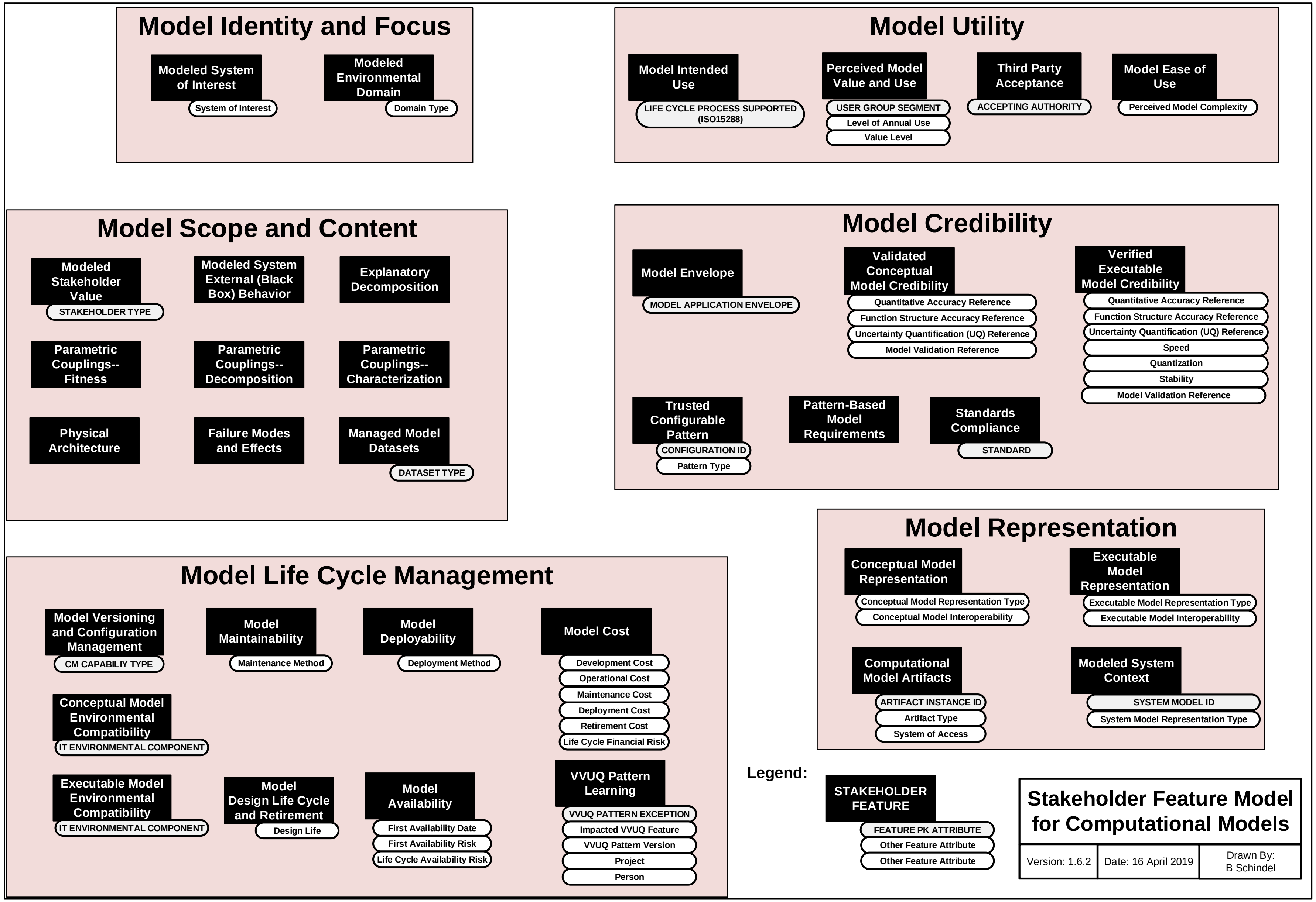
Model Life Cycle Management

Describes the related model life cycle management capabilities.

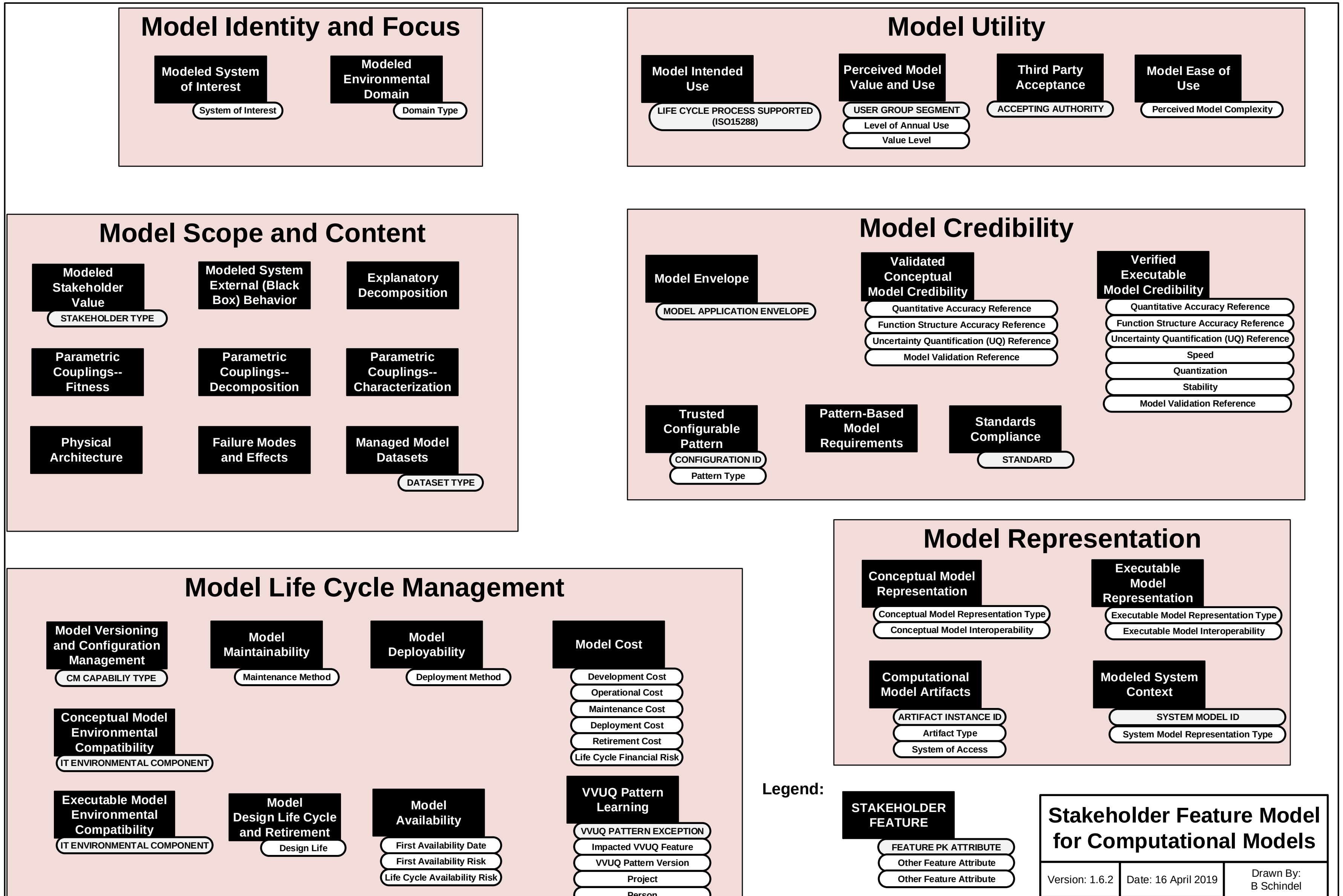
Model Representation

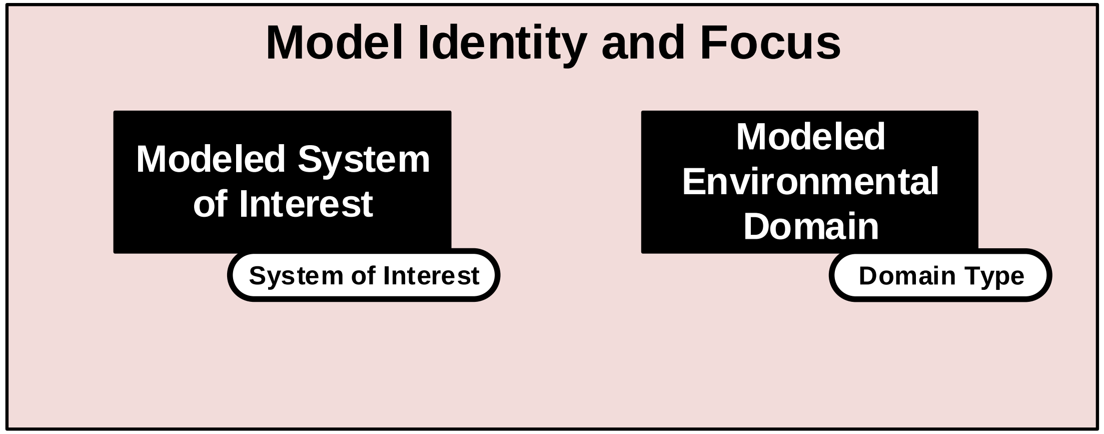
Describes the representation used by the model, along with related artifacts.

The Full Set of MCP Features



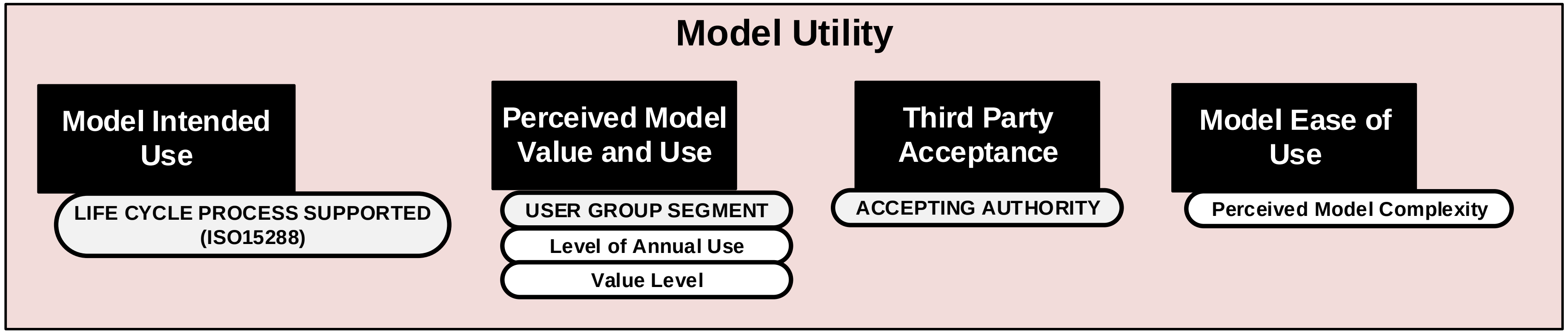
Computational Model Feature Groups: 29 Features, in 6 Feature Groups, *Configurable for Specific Models*





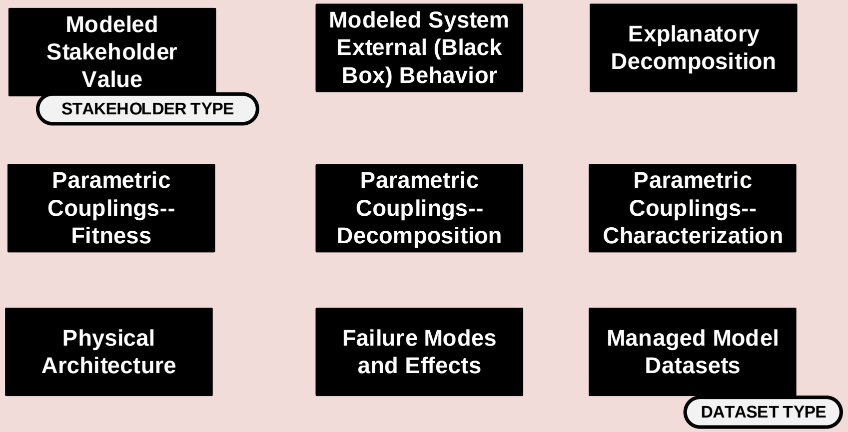
Feature Container	Feature Superclass	Feature Name	Config Rule Ref for Population	Feature Definition	Feature Attribute	PK	Attribute Definition
Model Identity and Focus		Modeled Environmental Domain		Identifies the type of external environmental domain(s) that this model includes.	Domain Type(s)	X	Name(s) of modeled domains. More than one instance may be populated.
Model Identity and Focus		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

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.

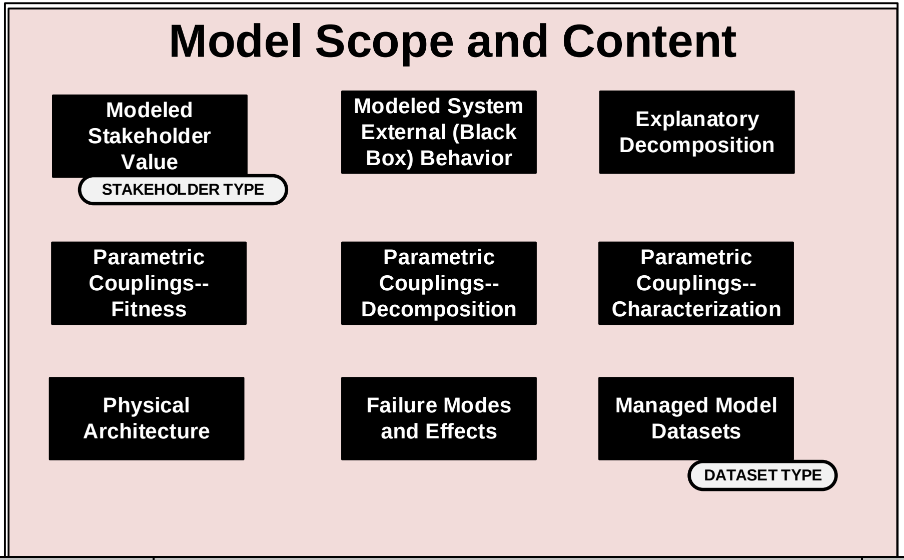


Feature Container	Feature Superclass	Feature Name	Config Rule Ref for Population	Feature Definition	Feature Attribute	PK	Attribute Definition
Model Utility		Model Ease of Use		The perceived ease with which the model can be used, as experienced by its intended users	Perceived Model Complexity		High, Medium Low
Model Utility		Model Intended Use		The intended purpose(s) or use(s) of the model.	Life Cycle Process Supported	X	The intended life cycle management process to be supported by the model, from the ISO15288 process list. More than one value may be listed.
Model Utility		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	X	The identify of using group segment (multiple)
Model Utility		Perceived Model Value and Use		The relative level of value ascribed to the model, by those who use it for its stated purpose.	Level of Annual Use		The relative level of annual use by the segment
Model Utility		Perceived Model Value and Use		The relative level of value ascribed to the model, by those who use it for its stated purpose.	Value Level		The value class associated with the model by that segment
Model Utility		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 stated purpose.	Accepting Authority	X	The identity (may be multiple) of regulators, agencies, customers, supply chains, accepting the model

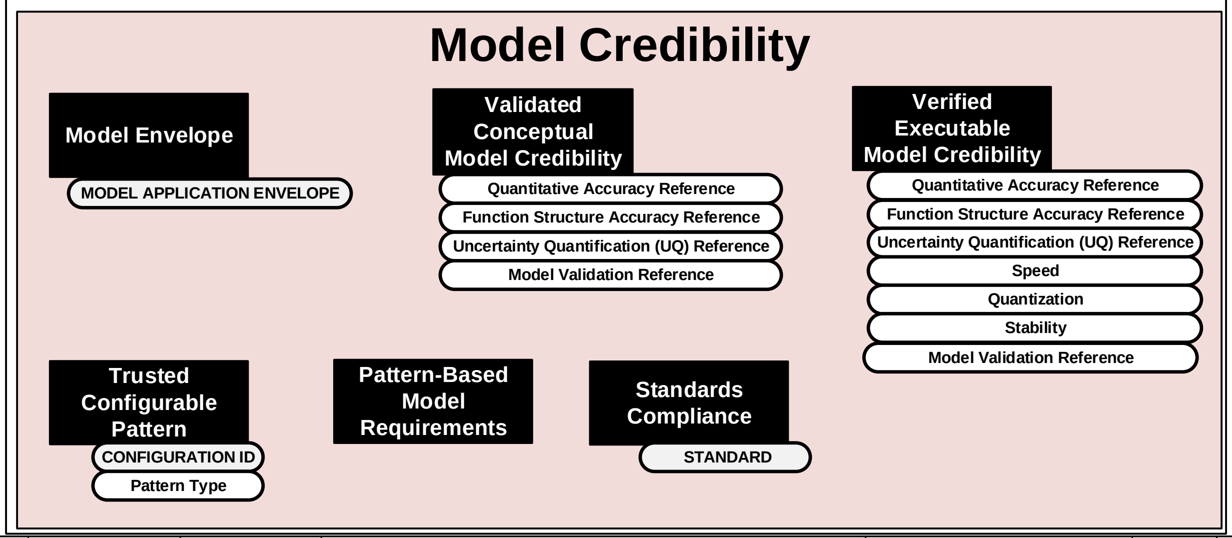
Model Scope and Content



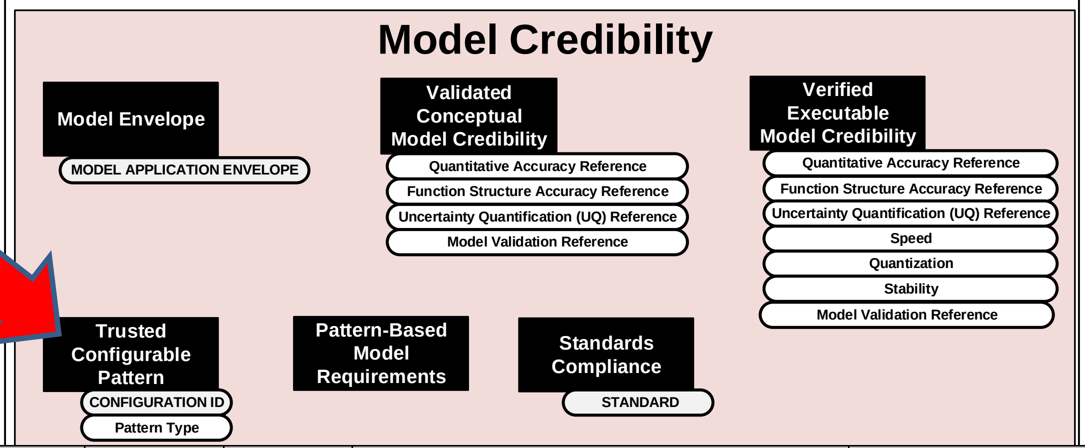
Feature Container	Feature Superclass	Feature Name	Config Rule Ref for Population	Feature Definition	Feature Attribute	PK	Attribute Definition
Model Scope and Content		Explanatory Decomposition		The capability of the model to represent the decomposition of its external technical behavior, as explanatory internal ("white box") internal interactions of decomposed roles, further quantified by internal technical performance measures, and varying internal behavioral modes.			
Model Scope and Content		Failure Modes and Effects		The capability of the model to include identification and analysis of system failure modes, their impact effects, causes, and liklihoods of occurrence.			
Model Scope and Content		Managed Model Datasets		The capability of the model to include managed datasets for use as inputs, parametric characterizations, or outputs	Dataset Type	X	The type(s) of data sets (may be multiple)
Model Scope and Content		Modeled 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.	Stakeholder Type	X	Classes of covered stakeholders. More than one instance may be populated.
Model Scope and Content		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.			



Feature Container	Feature Superclass	Feature Name	Config Rule Ref for Population	Feature Definition	Feature Attribute	PK	Attribute Definition
Model Scope and Content		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).			
Model Scope and Content		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.			
Model Scope and Content		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.			
Model Scope and Content		Physical Architecture		The capabiliy 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.			

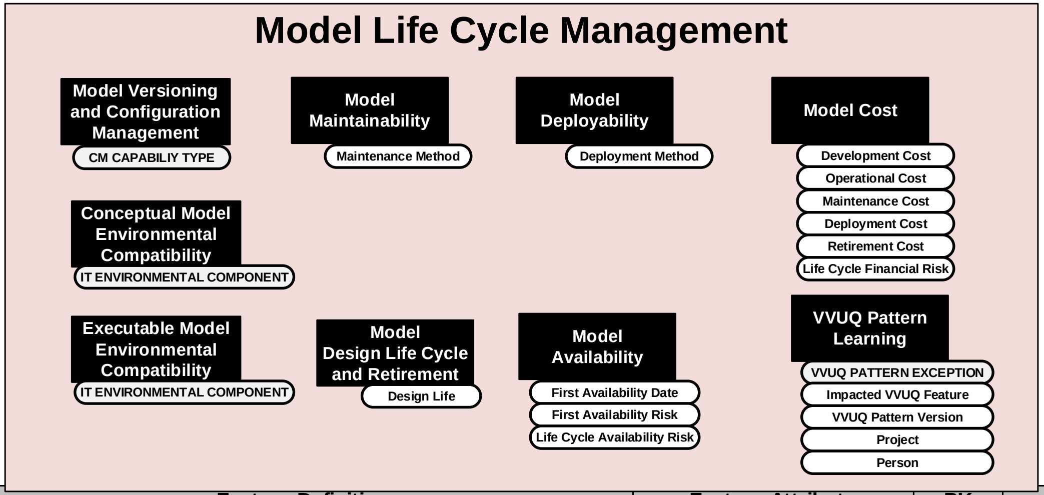


Feature Container	Feature Superclass	Feature Name	Config Rule Ref for Population	Feature Definition	Feature Attribute	PK	Attribute Definition
Model Credibility		Verified Executable Model Credibility		The verified capability of the executable portion of the model to represent the System of Interest, with acceptable Credibility.	Quantitative Accuracy Reference		The specification reference describing the quantitative accuracy of the conceptual model compared to the system of interest.
Model Credibility		Verified Executable Model Credibility		The verified capability of the executable portion of the model to represent the System of Interest, with acceptable Credibility.	Structural Accuracy Reference		The specification reference describing the structural (presence or absence of elements) accuracy of the executable model to the conceptual model.
Model Credibility		Verified Executable Model Credibility		The verified capability of the executable portion of the model to represent the System of Interest, with acceptable Credibility.	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 Credibility		Verified Executable Model Credibility		The verified capability of the executable portion of the model to represent the System of Interest, with acceptable Credibility.	Speed		The specification reference describing the execution run time (speed) for the executable model.
Model Credibility		Verified Executable Model Credibility		The verified capability of the executable portion of the model to represent the System of Interest, with acceptable Credibility.	Quantization		The specification reference describing the quantization error of the executabl e model.
Model Credibility		Verified Executable Model Credibility		The verified capability of the executable portion of the model to represent the System of Interest, with acceptable Credibility.	Stability		The specification reference describing the level of stability of the accuracy and uncertainty of the executable model error characteristics.
Model Credibility		Verified Executable Model Credibility		The verified capability of the executable portion of the model to represent the System of Interest, with acceptable Credibility.	Model Validation Reference		The reference documenting the validation of the conceptual model's Credibility to the system of interest.

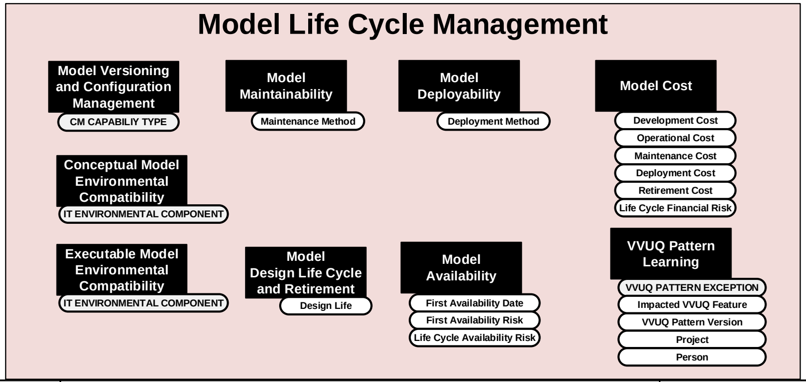


Of special importance to the economics of trust and VVUQ

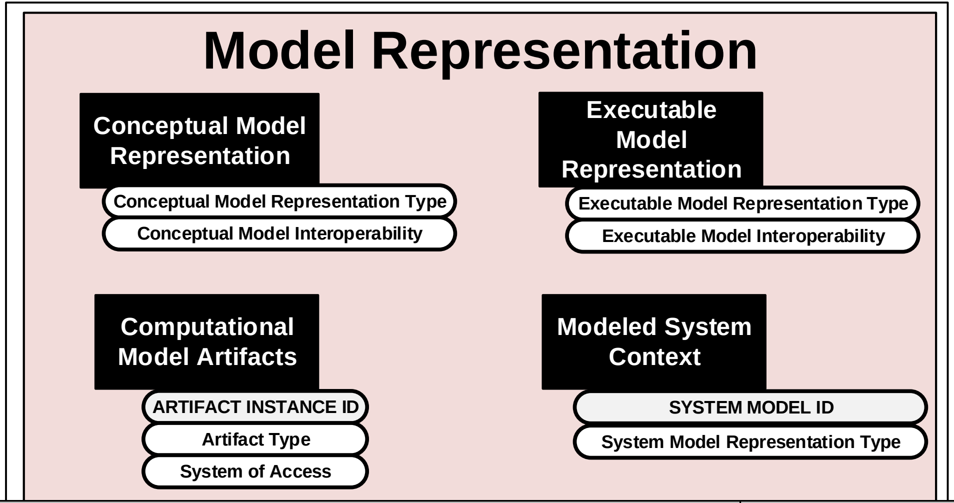
Feature Container	Feature Superclass	Feature Name	Config Rule Ref for Population	Feature Definition	Feature Attribute	PK	Attribute Definition
Model Credibility		Model Envelope		The capability of the model to meet its Model Credibility requirements over a stated range (envelope) of dynamical inputs, outputs, and parameter values.	Model Application Envelope		The range over which the model is intended for use.
Model Credibility		Pattern-Based Model Requirements		The requirements for this model were configured from the general model requirements pattern.			
Model Credibility		Standards Compliance		Conforming to formal standards for models, modeling, model VVUQ, security, information technology, or other model-supporting standards.	Standard	X	The identification of a standard applicable to models, modeling, model VVUQ, security, information technology, or other model-supporting standards.
Model Credibility		Trusted Configurable Pattern		The capability of the model to serve as a configurable pattern, representing different modeled system configurations across a common domain, spreading the cost of establishing trusted model frameworks across a community of applications and configurations.	Configuration ID		A specific system of interest configuration within the family that the pattern framework can represent.
Model C		usted Configurable Pattern		The capability of the model to serve as a configurable pattern, representing different modeled system configurations across a common domain, spreading the cost of establishing trusted model frameworks across a community of applications and configurations.	Pattern ID		The identifier of the trusted configurable pattern.
bility		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.
Model Credibility		Validated Conceptual Model Credibility		The validated capability of the conceptual portion of the model to represent the System of Interest, with acceptable Credibility.	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.
Model Credibility		Validated Conceptual Model Credibility		The validated capability of the conceptual portion of the model to represent the System of Interest, with acceptable Credibility.	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 Credibility		Validated Conceptual Model Credibility		The validated capability of the conceptual portion of the model to represent the System of Interest, with acceptable Credibility.	Model Validation Reference		The reference documenting the validation of the conceptual model's Credibility to the system of interest.



Feature Container	Feature Superclass	Feature Name	Config Rule Ref for Population	Feature Definition	Feature Attribute	PK	Attribute Definition
Model Life Cycle Management		Conceptual Model Environmental Compatibility		The capability of the conceptual model to be compatibly supported by specified information technology environment(s), indicating compatibility, portability, and interoperability.	IT Environmental Component	X	The type(s) of IT environments or standards supported
Model Life Cycle Management		Executable Model Environmental Compatibility		The capability of the model to be compatibly supported by specified information technology environment(s), indicating compatibility, portability, and interoperability.	IT Environmental Component	X	The type(s) of IT environments or standards supported
Model Life Cycle Management		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
Model Life Cycle Management		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 Risk		Risk to the scheduled date of first availability
Model Life Cycle Management		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.	Life Cycle Availability Risk		Risk to ongoing availability after introduction
Model Life Cycle Management		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
Model Life Cycle Management		Model Cost		The financial cost of the model, including development, operating, and maintenance cost	Operational Cost		The cost to execute and otherwise operate the model, in standardized execution load units
Model Life Cycle Management		Model Cost		The financial cost of the model, including development, operating, and maintenance cost	Maintenance Cost		The cost to maintain the model
Model Life Cycle Management		Model Cost		The financial cost of the model, including development, operating, and maintenance cost	Deployment Cost		The cost to deploy, and redeploy updates, per cycle
Model Life Cycle Management		Model Cost		The financial cost of the model, including development, operating, and maintenance cost	Retirement Cost		The cost to retire the model from service, in a planned fashion
Model Life Cycle Management		Model Cost		The financial cost of the model, including development, operating, and maintenance cost	Life Cycle Financial Risk		Risk to the overall life cycle cost of the model

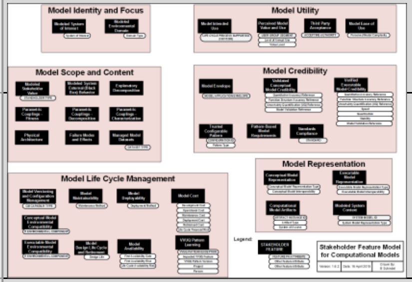


Feature Container	Feature Superclass	Feature Name	Config Rule Ref for Population	Feature Definition	Feature Attribute	PK	Attribute Definition
Model Life Cycle Management		Model Deployability		The capability of the model to support deployment into service on behalf of intended users, in its original or subsequent updated versions	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 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 Life Cycle Management		Model Maintainability		The relative ease with which the model can be maintained over its intended life cycle and use, based on capable maintainers, availability of effective model documentation, and degree of complexity of the model	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 Life Cycle Management		Model Versioning and Configuration Management		The capability of the model to provide for version and configuration management.	CM Capability Type	X	The type(s) of CM capabilities included (may be multiple)
Model Life Cycle Management		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 candidates for inclusion into future versions of that pattern.	VVUQ Pattern Exception	X	A summary of the exception noted to the current VVUQ Pattern (may be multiple exceptions)
Model Life Cycle Management		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 candidates for inclusion into future versions of that pattern.	Impacted VVUQ Feature		The impacted existing, modified, or additional feature of the VVUQ Pattern.
Model Life Cycle Management		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 candidates for inclusion into future versions of that pattern.	VVUQ Pattern Version		The version of the VVUQ Pattern in current use before change.
Model Life Cycle Management		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 candidates for inclusion into future versions of that pattern.	Project		Identifies the project in which the exception was noted
Model Life Cycle Management		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 candidates for inclusion into future versions of that pattern.	Person		Identifies the person describing the exception



Feature Container	Feature Superclass	Feature Name	Config Rule Ref for Population	Feature Definition	Feature Attribute	PK	Attribute Definition
Model Representation		Computational Model Artifacts		The capability of the model system (including its life cycle management system) to create, mainain, and access artifacts of the development and use of the model.	Artifact Instance ID	X	The unique identifier of an artifact.
Model Representation		Computational Model Artifacts		The capability of the model system (including its life cycle management system) to create, mainain, and access artifacts of the development and use of the model.	Artifact Type		The type of an artifact.
Model Representation		Computational Model Artifacts		The capability of the model system (including its life cycle management system) to create, mainain, and access artifacts of the development and use of the model.	System of Access		The method of accessing an artifact.
Model Representation		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 Representation Type		The type of conceptual modeling language or metamodel used.
Model Representation		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 Interoperability		The degree of interoperability of the conceptual model, for exchange with other environments
Model Representation		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 Representation Type		The type of executable modeling language or metamodel used.
Model Representation		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 Interoperability		The degree of interoperability of the executable model, for exchange with other environments
Model Representation		Modeled System Context		The capability to provide system level modeling of the larger context of a computational model.	System Model ID	X	The unique identifier of a system level model.
Model Representation		Modeled System Context		The capability to provide system level modeling of the larger context of a computational model.	System Model Representation Type		The type of representation used for a system model

Appendix II: MCP Technical Requirements--Configurable Technical Requirements for a Computational Model



Row	A	B	C	D	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 Cy								
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																			
82	5	5. Model User Interface																					
83	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		
84	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		
85	6	6 Model Versioning and Configuration Management																					
86	6.1	Model Versioning	The model shall carry versioning information compatible with a required configuration management method.									X			X		X						
87	6.2	Managed Dataset Versioning	The model's managed datasets shall carry versioning information compatible with a required configuration management									X			X								
88	6.3	Model Documentation Versioning	The model's documentation shall carry versioning information compatible with a required configuration management									X			X								
89	7	7. Model Life Cycle Management																					
90	7.1	7.1 Operating Environment																					
91	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 Reqrmnts, 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 external 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.	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).
2.2.7	External State Transitions	The model shall represent the possible (state) transitions between the modeled system behavioral modes.	
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" 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 signficant 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 familure.	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 quantiative 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 quantiative 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 propgated 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.

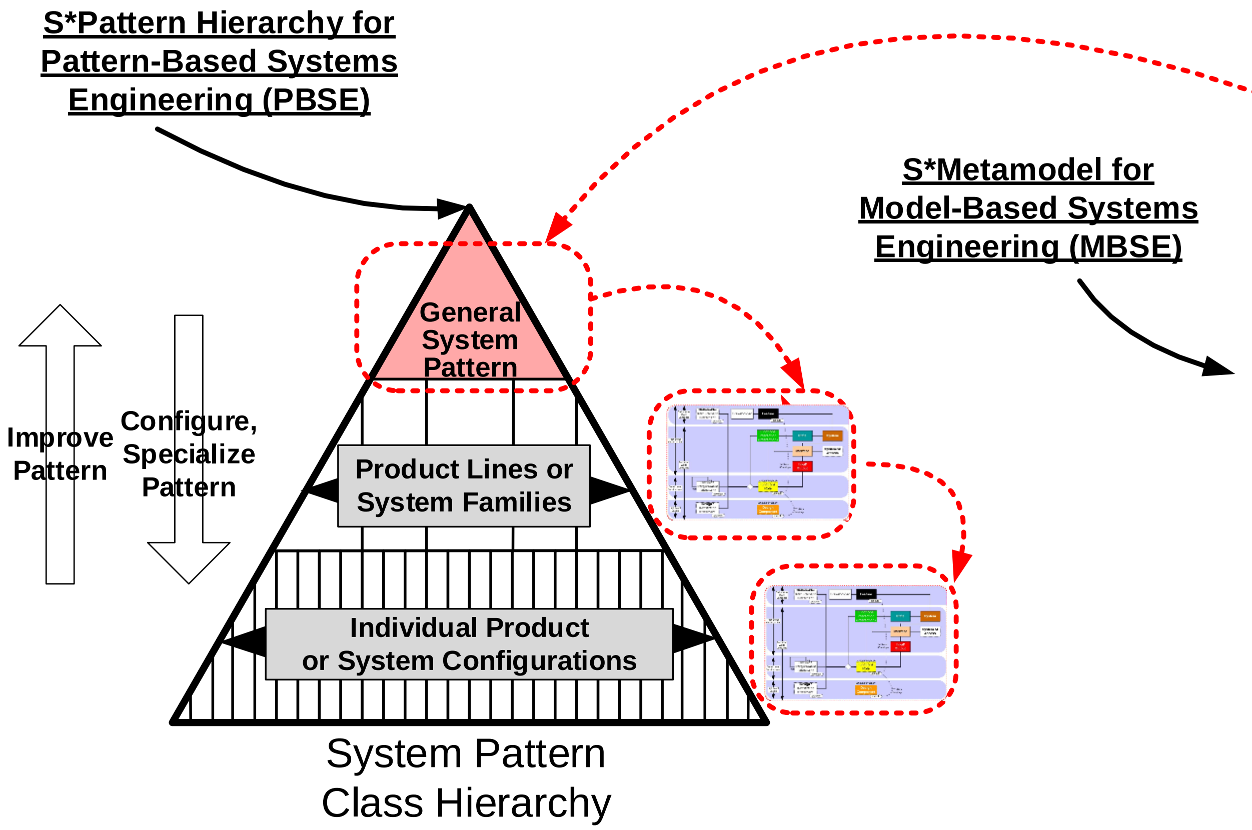
Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
4	**4. Model Representation**		
4.2	**4.2 Implemented Executable Representation**		
4.2.1	Implemented Model Representation Type	The implemented executable model shall represent the system of interest 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, esecution time, accuracy and uncertainty specifications.	The Model Requirements Pattern specifies the categories of requirements that should be included in this documentation.

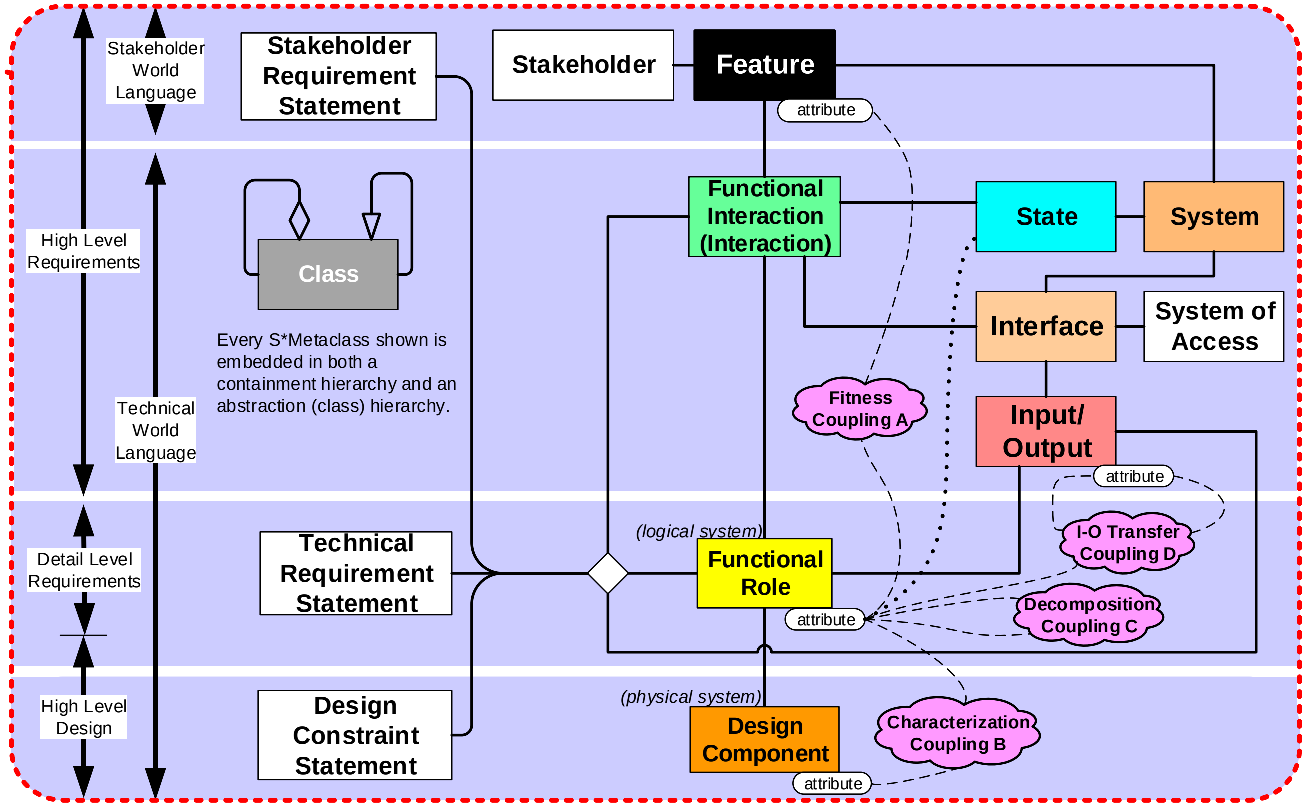
Req ID	Model Requirement Name	Model Requirement (configure further as needed)	Explanation, discussion
5	**5. Model User Interface**		
5.1	Model User Interface	The model user interfaces, per the [Model UI Specification] shall facilitate the efficient and effective performance of the intended purpose of 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 compete 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 Requriements 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].	

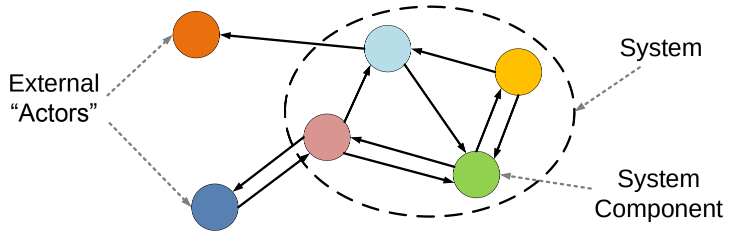
Appendix III:
S*Metamodel, S*Models and S*Patterns



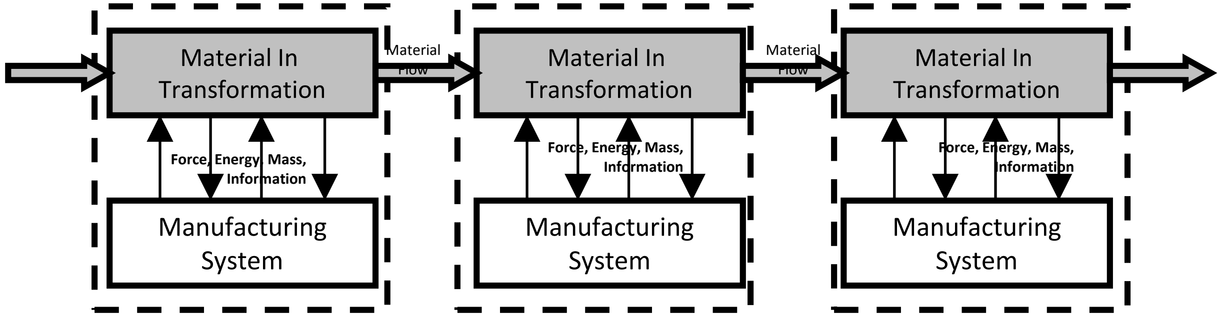


S*Metamodel informal summary pedagogical diagram
(formal S*Metamodel includes additional details.)

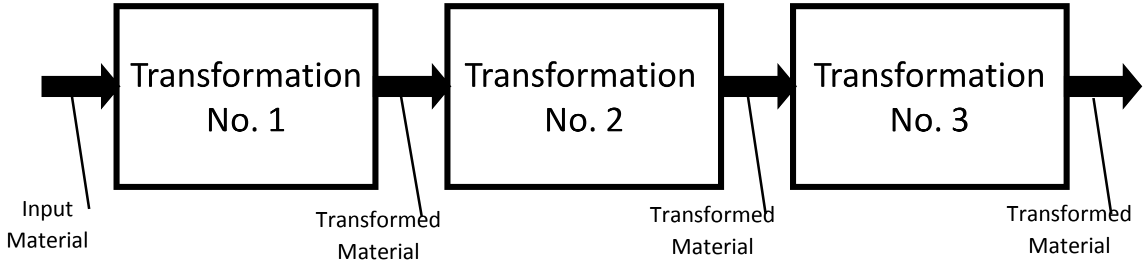
- A System is a set of interacting components:
  - By “interact”, we mean exchanging energy, forces, mass flows, or information, resulting in changes of state:



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



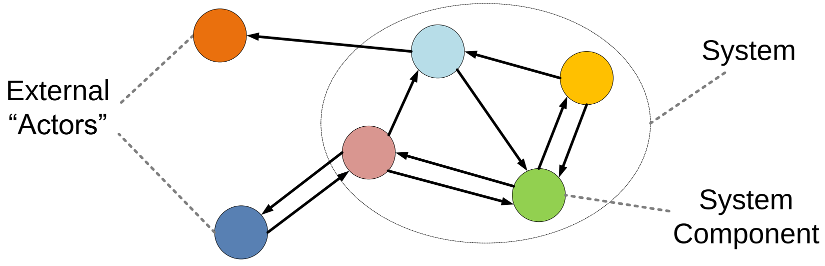
- “White Box” view of a system sees its <u>internal</u> interactions.

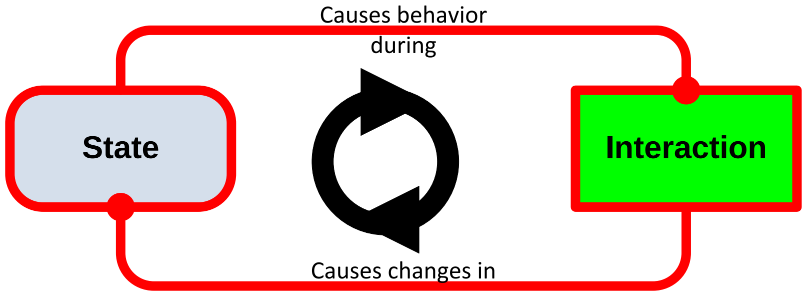


- “Black Box” view of a system sees only its <u>external</u> behavior.

The System Phenomenon

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



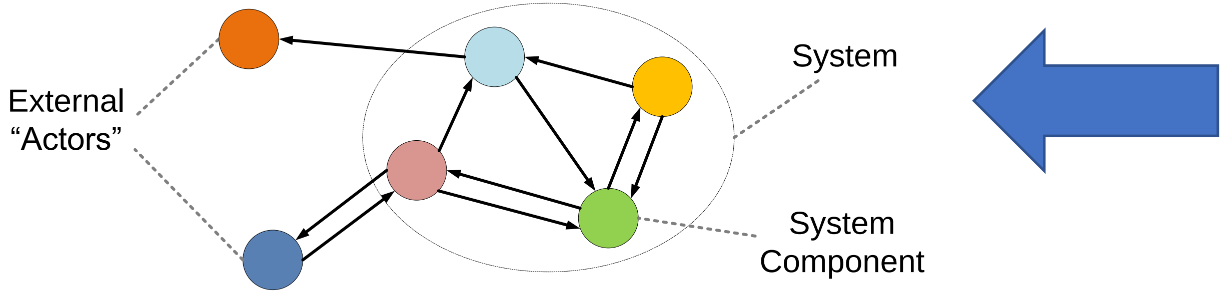


- Where interaction involves the exchange of energy, force, mass, or information, . . .
- Through which one component impacts the state of another component, . . .
- And in which the state of a component impacts its behavior in future interactions.

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 phenomena[1], the emergent interaction-based behavior of the larger system is a stationary path of the action integral:

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



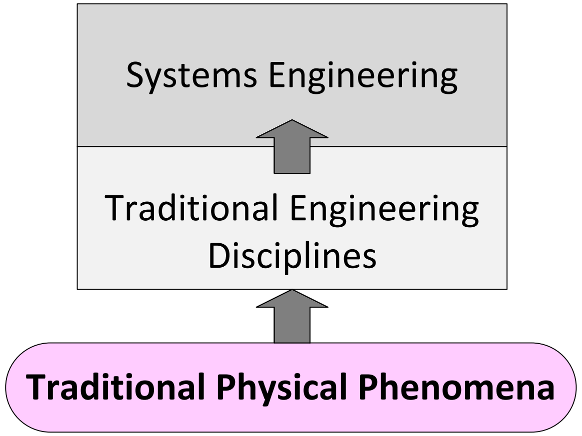
(Hamilton's Principle[1])

- 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.

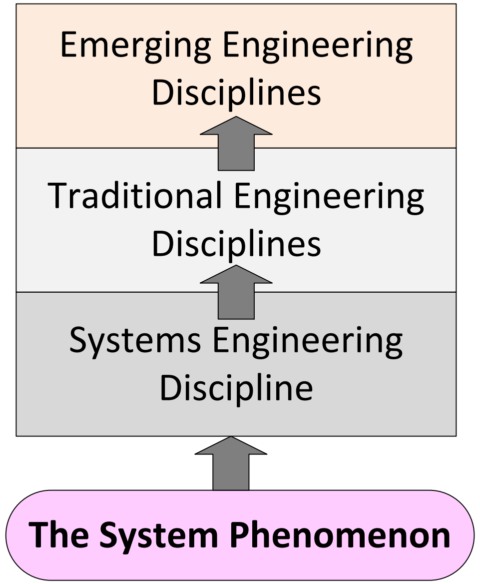
(1) When stated with rigor, special cases for non-holonomic constraints, irreversible dynamics, discrete systems, data systems, etc., led to alternatives to the variational Hamilton's Principle—but the ***interaction-based structure*** of the System Phenomenon remained, and the underlying related Action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

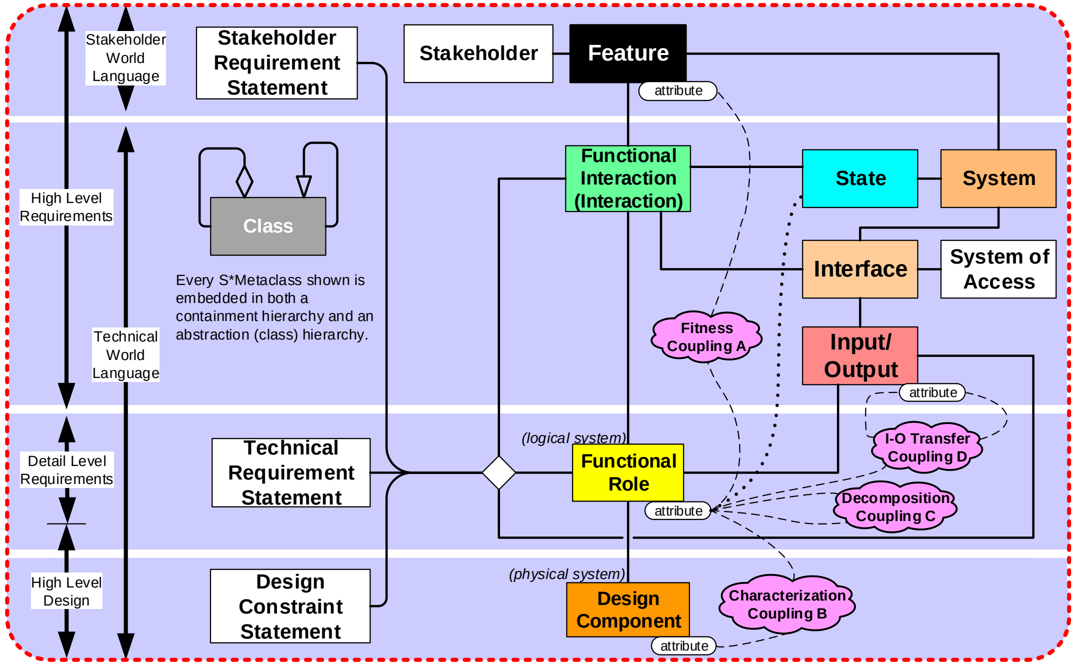
Representing System Patterns: The S* Metamodel Framework

- 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 <u>metamodel</u> 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:

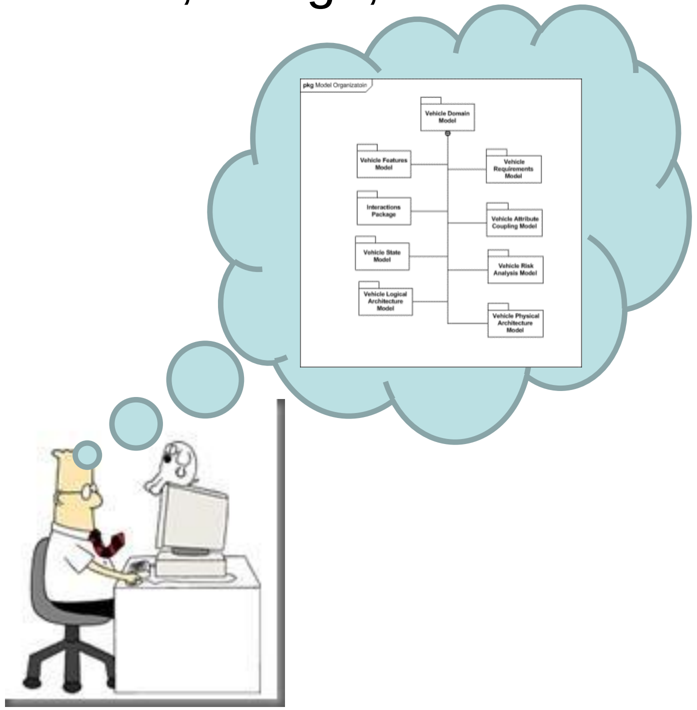
- The resulting system models may be expressed in a wide variety of third party COTS and enterprise information systems, based on S*Metamodel mappings to those environments.
- Has been applied to systems engineering in aerospace, transportation, medical, advanced manufacturing, communication, construction,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

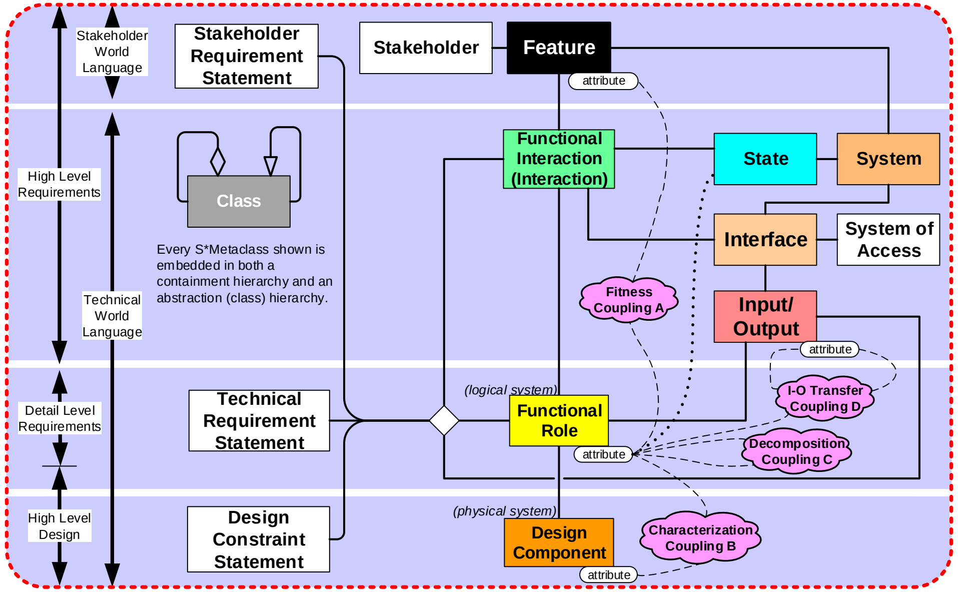


S*Metamodel informal summary pedagogical diagram
(formal S*Metamodel includes additional details.)

Taking advantage of Model-Based Systems Engineering (MBSE)

- **<u>An S* Model</u>** 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.):





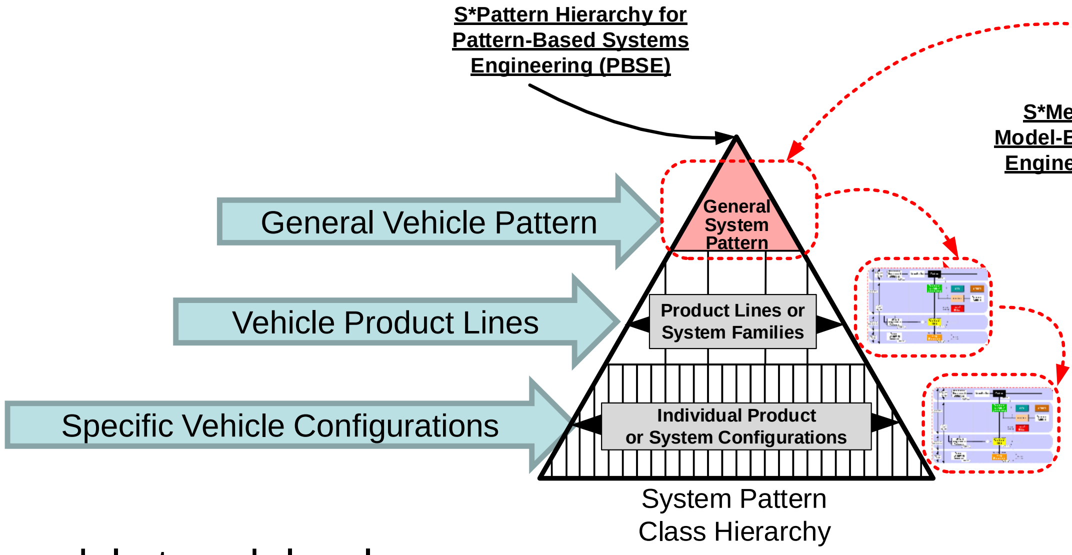
S*Metamodel informal summary pedagogical diagram
(formal S*Metamodel includes additional details.)

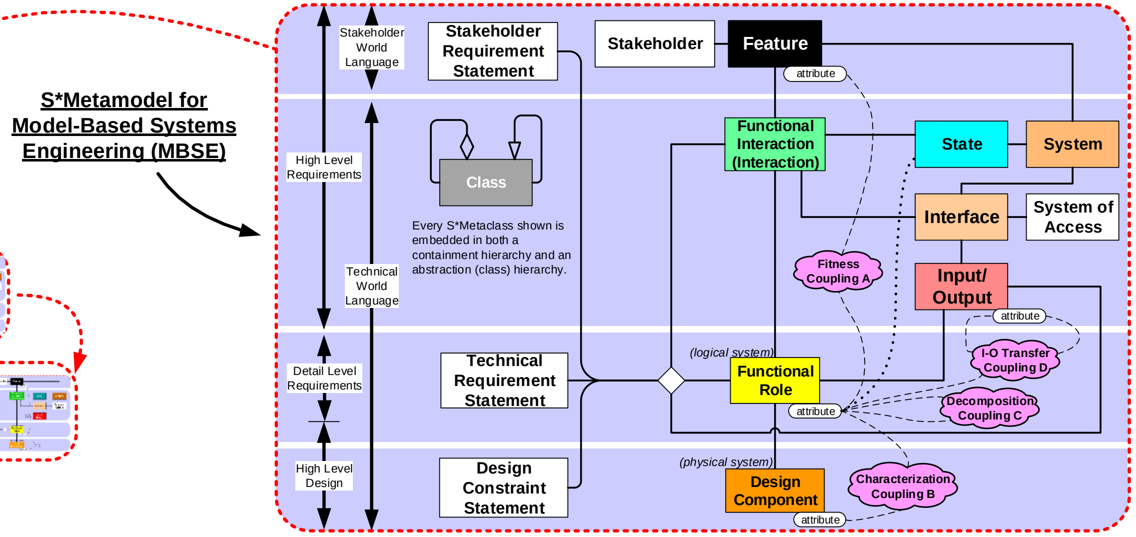
Over two decades of S*Model and S*Patterns practice, experience using S*Metamodel

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.):



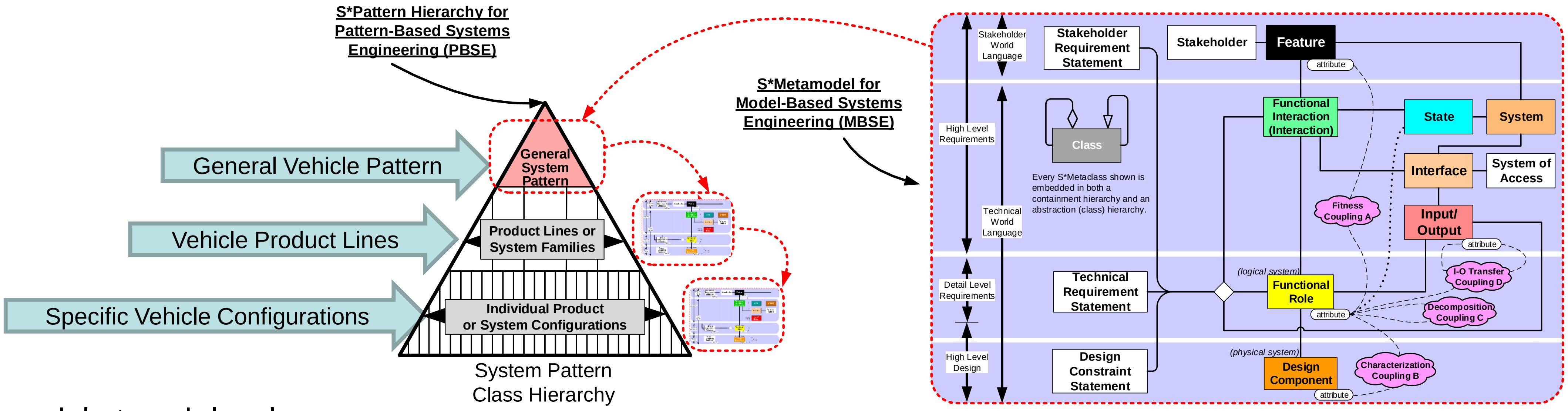


Same S*Metamodel at each level

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

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.

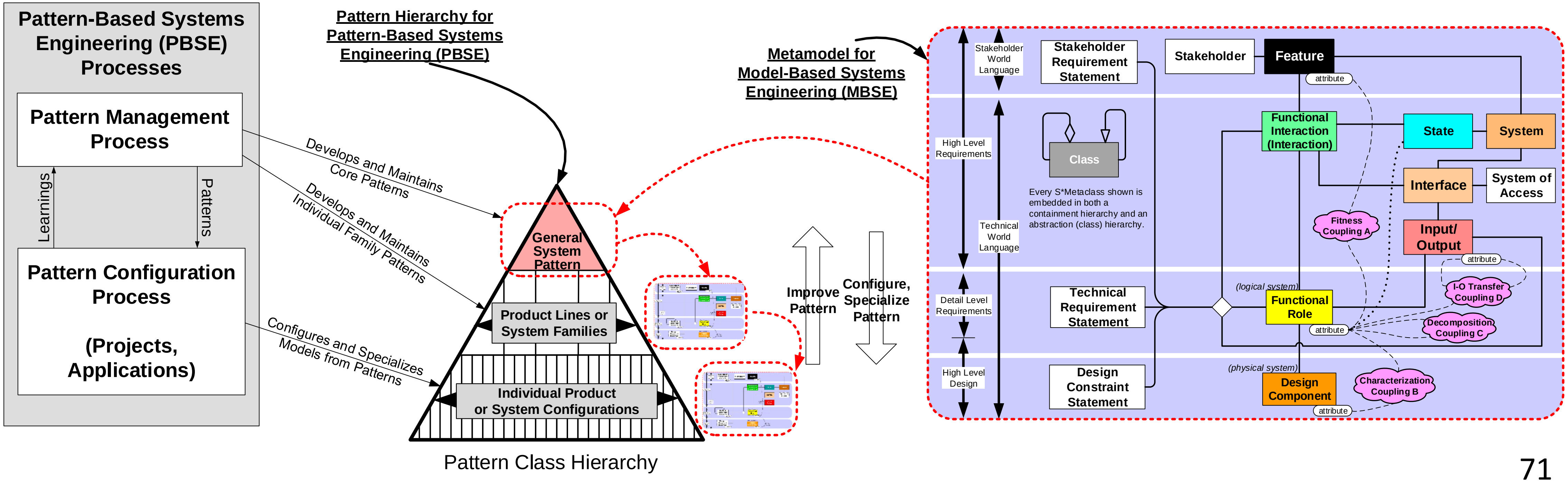


Same S*Metamodel at each level

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

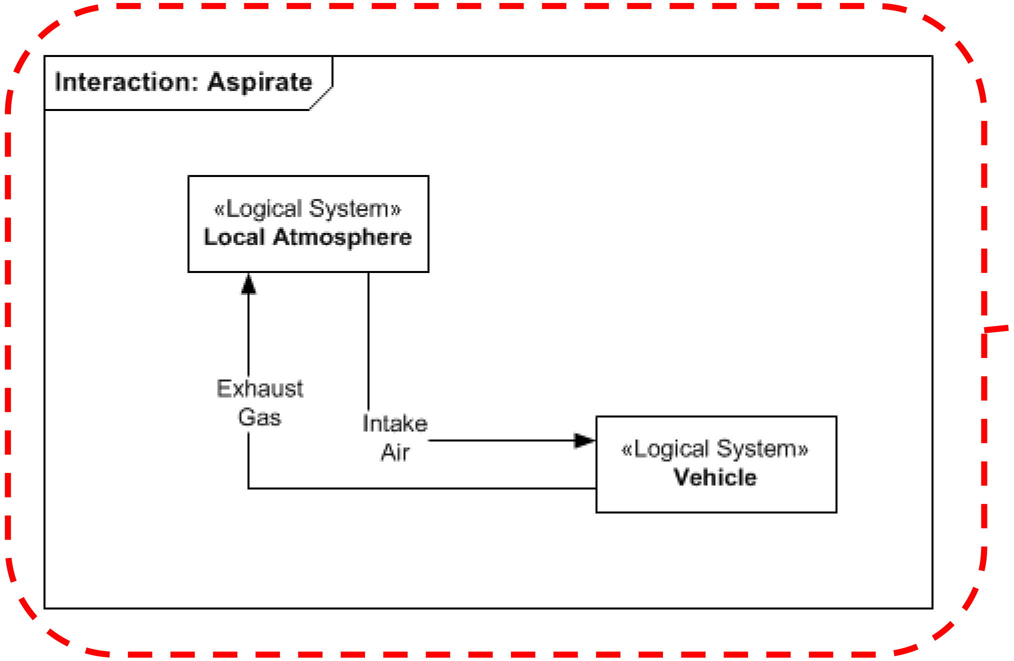
- **System**: A collection of interacting components. Example: Medical Device; Hospital Domain, Health Care Delivery System Domain.
- **Stakeholder**: A person or other entity with something at stake in the life cycle of a system. Example: Patient; Health Care Provider; Enterprise Shareholder
- **Feature**: 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

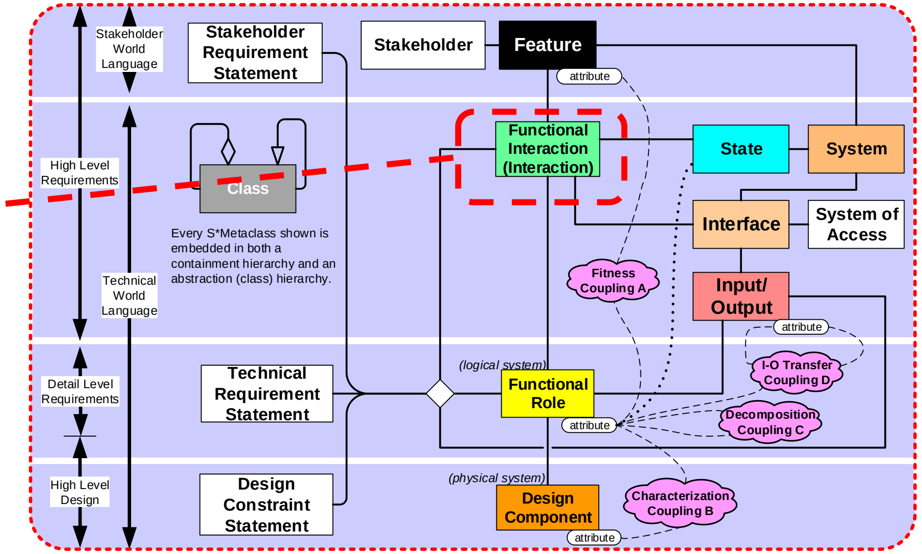
Definitions of some S* Metamodel Classes

- **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 (!)

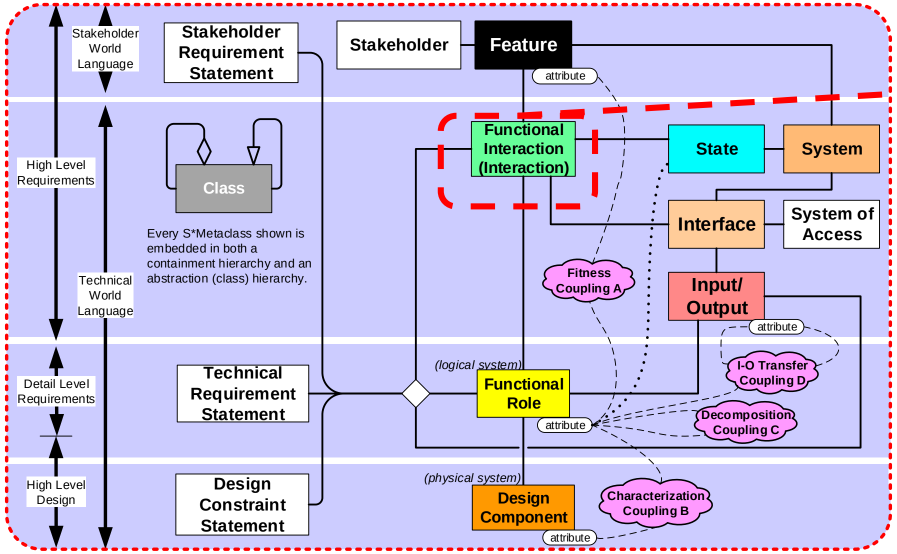




S*Metamodel informal summary pedagogical diagram
(formal S*Metamodel includes additional details.)

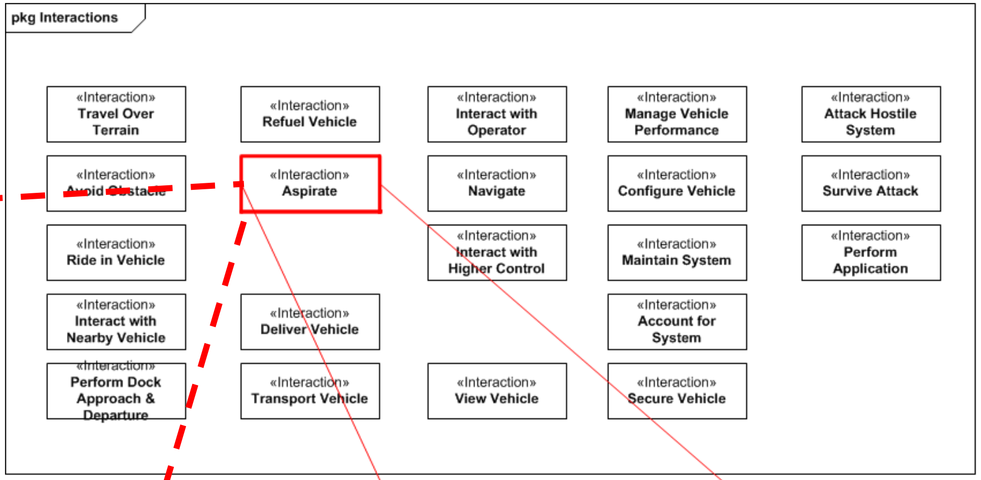
Physical Interactions: At the heart of S* models

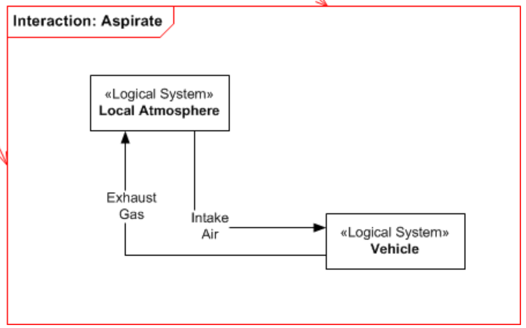
- S* models represent <u>Physical Interactions</u> as explicit objects:



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

Vehicle Pattern Interactions





Interaction Diagram

<u>Aspirate</u>: 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)

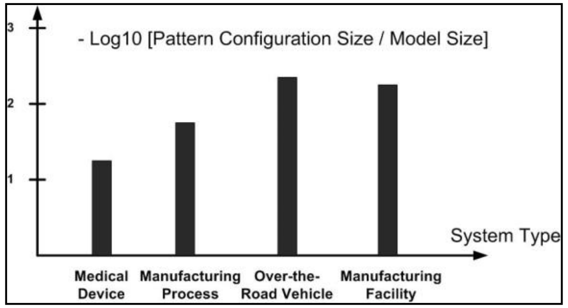
- Model-based Patterns:
  - In this approach, Patterns are reusable, configurable S* models of families (product lines, sets, ensembles) of systems.
  - A Pattern is not just the physical product family—it includes its behavior, decomposition structure, failure modes, and other aspects of its model.
- 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.

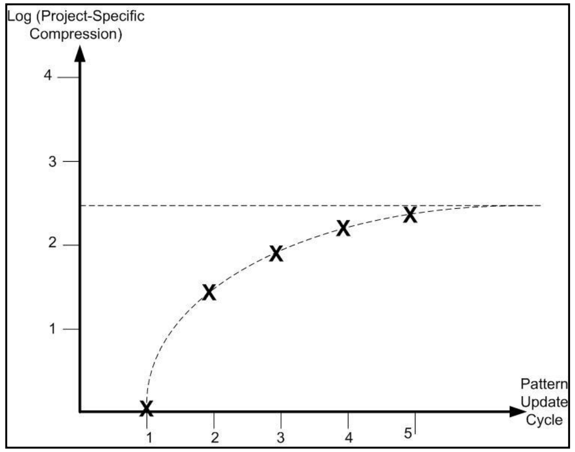
Pattern configurations

- 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.

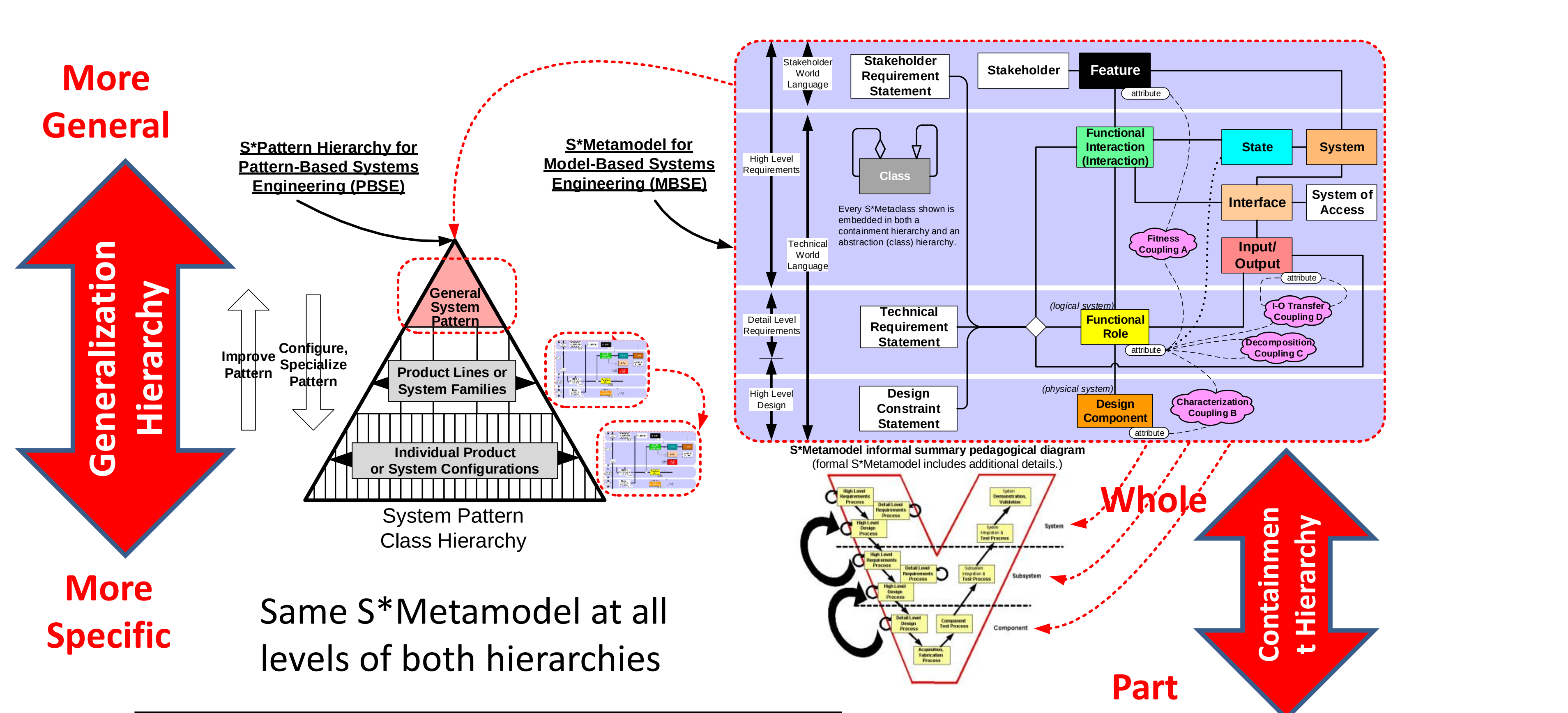
Lawnmower Product Line: Configurations Table

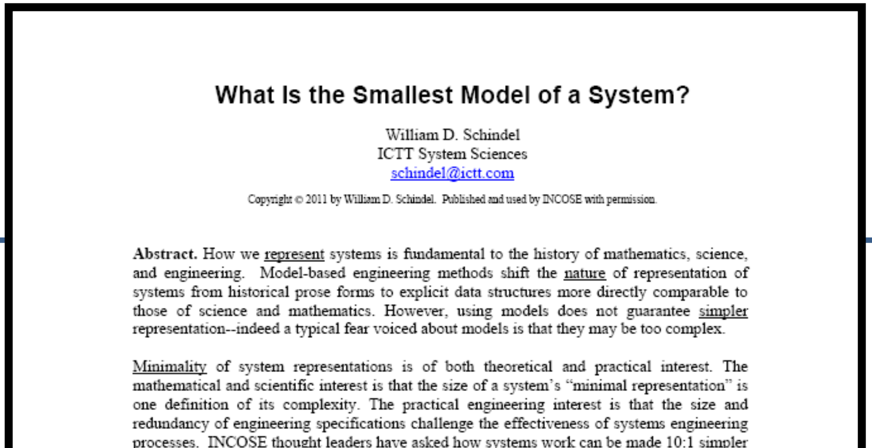
		Units	Walk-Behind	Walk-Behind	Walk-Behind	Riding	Riding	Riding Mower	Autonomous
			Push Mower	Mower	Self-Propelled	Rider	Tractor	Tractor	Autonomous
			Push Mower	Self-Propelled	Wide Cut	Rider	Lawn	Garden	Auto Mower
	Model Number		M3	M5	M11	M17	M19	M23	M100
	Market Segment		Sm Resident	Med Resident	Med 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:





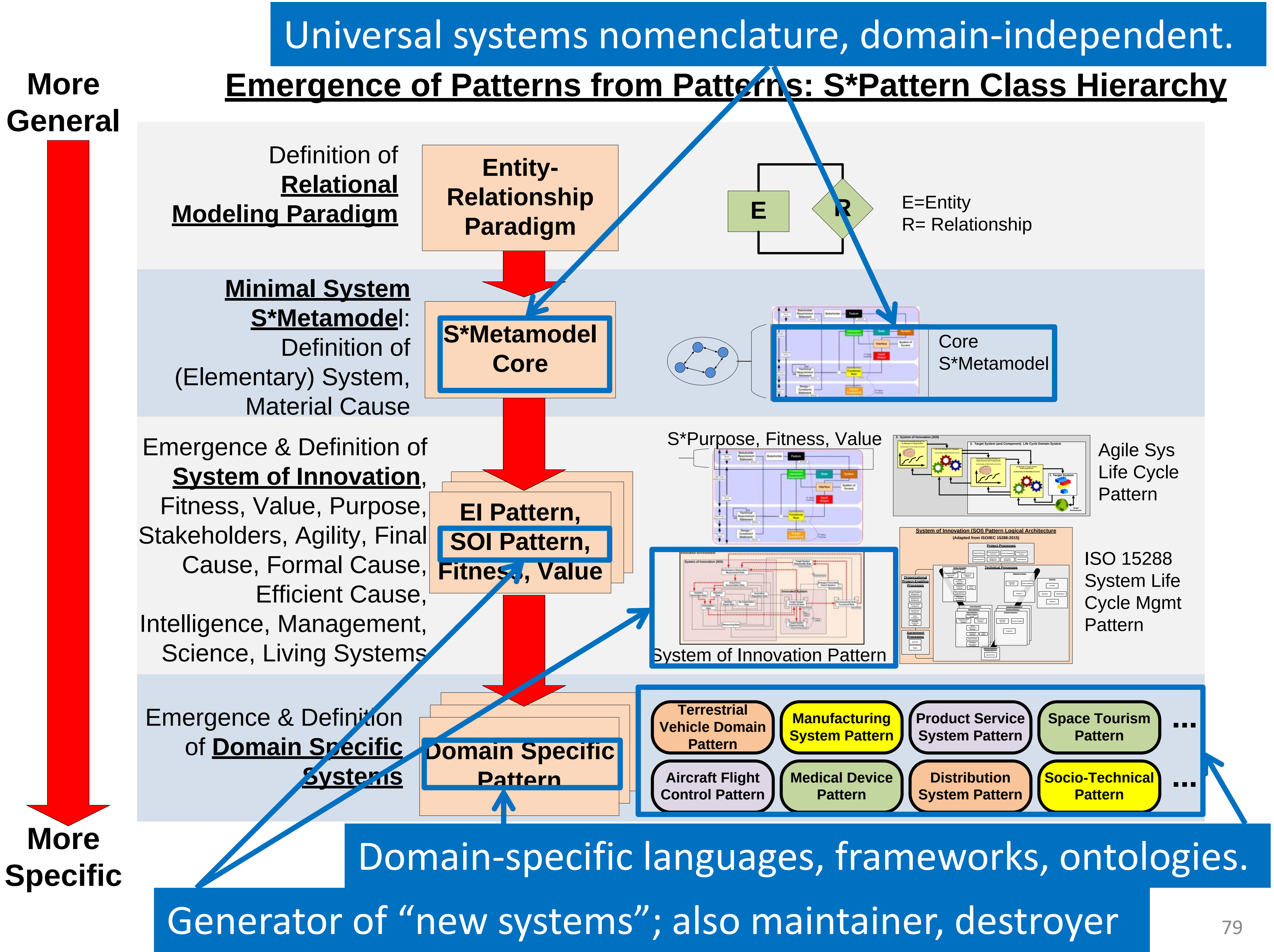
What Is the Smallest Model of a System?

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

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

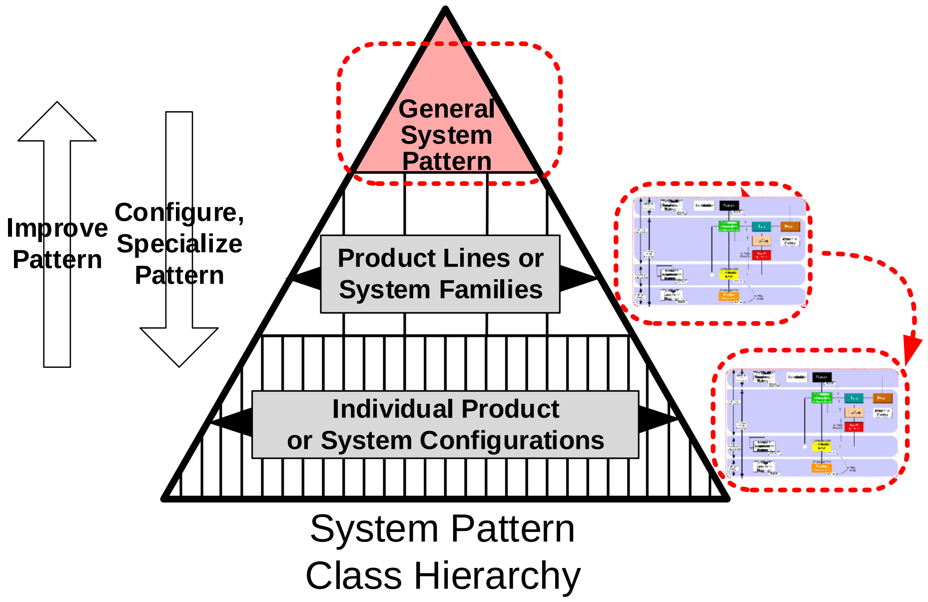
Abstract. How we represent systems is fundamental to the history of mathematics, science, and engineering. Model-based engineering methods shift the nature of representation of systems from historical prose forms to explicit data structures more directly comparable to those of science and mathematics.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

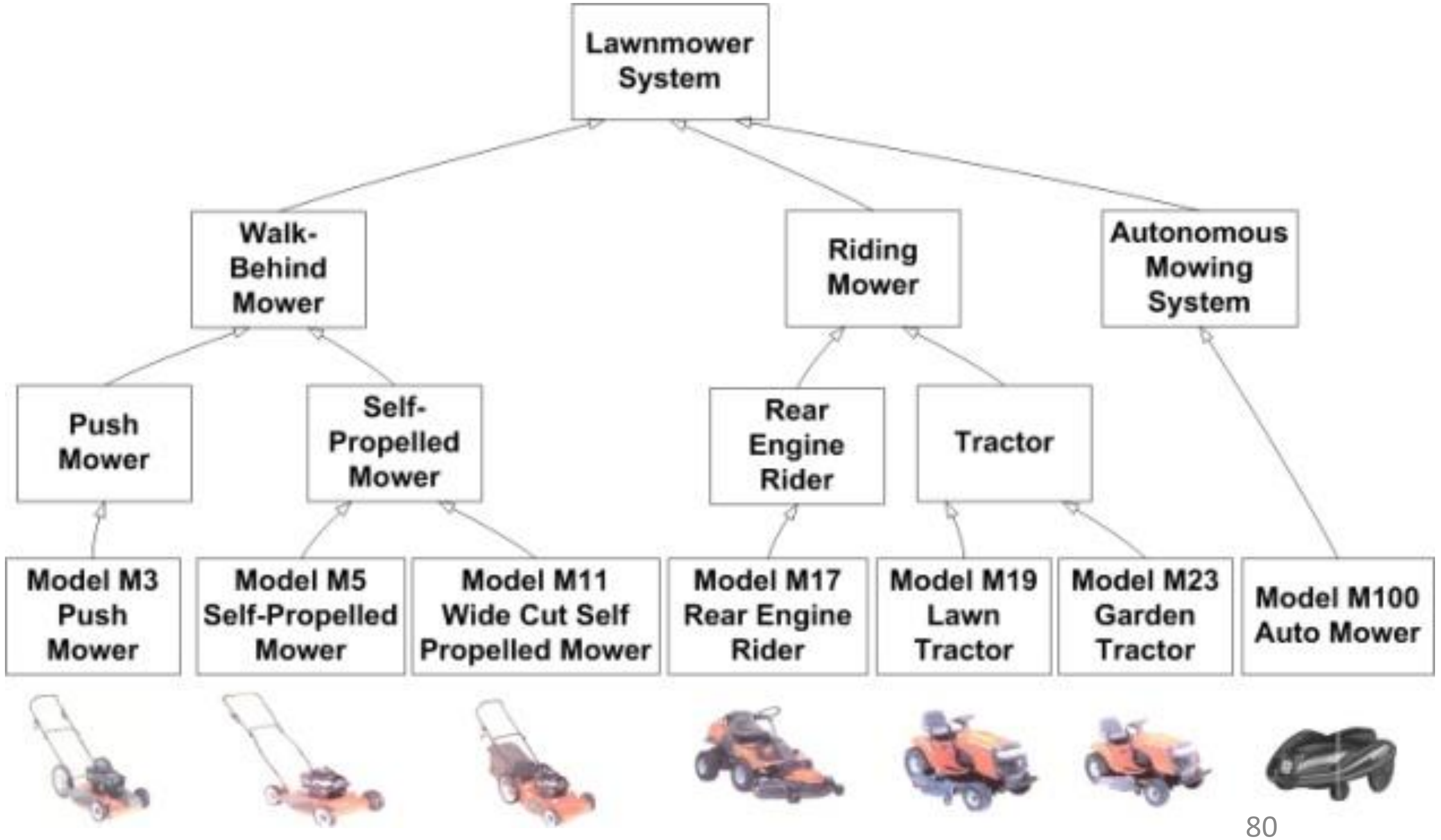
Minimality of system representations is of both theoretical and practical interest. The mathematical and scientific interest is that the size of a system's "minimal representation" is one definition of its complexity. The practical engineering interest is that the size and redundancy of engineering specifications 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
Definition of Relational Modeling Paradigm
Entity-Relationship Paradigm
E
R
E=Entity
R= Relationship
Minimal System S*Metamodel: Definition of (Elementary) System, Material Cause
S*Metamodel Core
Core S*Metamodel
Emergence & Definition of System of Innovation, Fitness, Value, Purpose, Stakeholders, Agility, Final Cause, Formal Cause, Efficient Cause, Intelligence, Management, Science, Living Systems
EI Pattern,
SOI Pattern,
Fitness, Value
S*Purpose, Fitness, Value
Agile Sys Life Cycle Pattern
System of Innovation Pattern
System of Innovation (SOI) Pattern Logical Architecture
ISO 15288 System Life Cycle Mgmt Pattern
Emergence & Definition of Domain Specific Systems
Domain Specific Pattern
Terrestrial Vehicle Domain Pattern
Manufacturing System Pattern
Product Service System Pattern
Space Tourism Pattern
Aircraft Flight Control Pattern
Medical Device Pattern
Distribution System Pattern
Socio-Technical Pattern
…
…
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



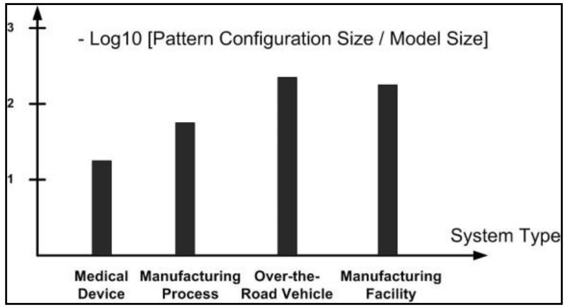


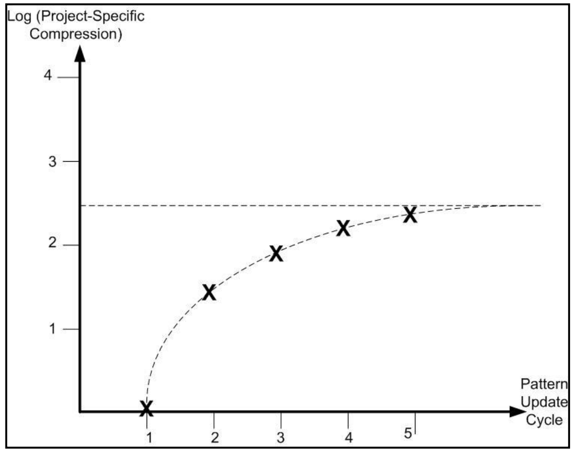
Pattern configurations

- 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

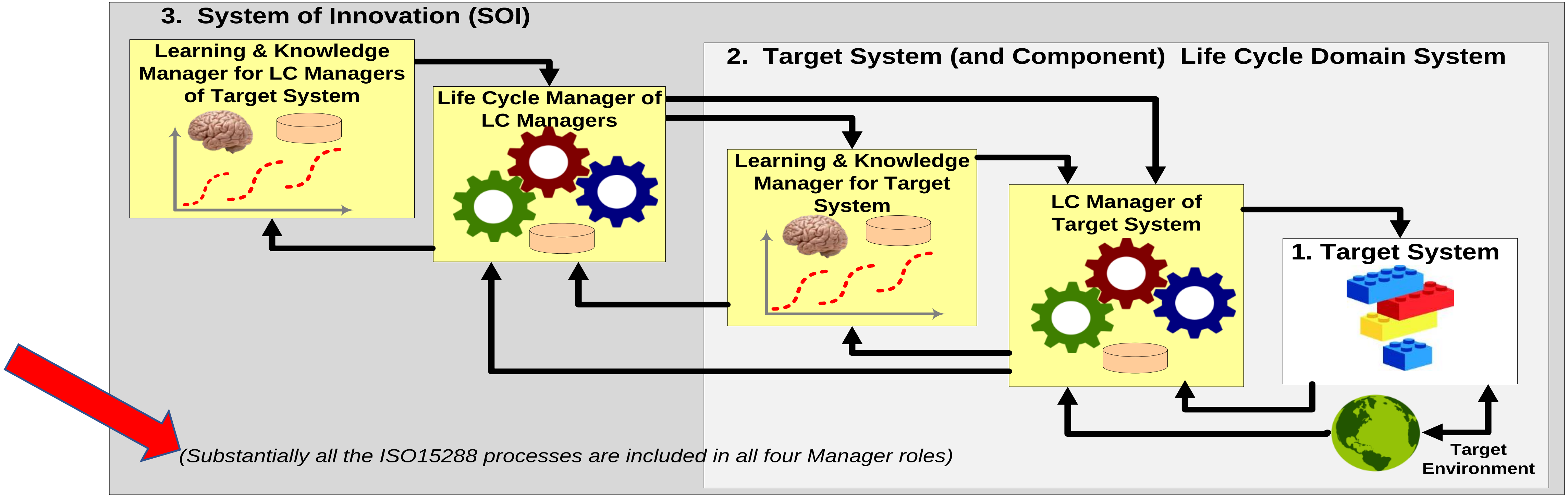
		Units	Walk-Behind	Walk-Behind	Walk-Behind	Riding	Riding	Riding Mower	Autonomous
			Push Mower	Mower	Self-Propelled	Rider	Tractor	Tractor	Autonomous
			Push Mower	Self-Propelled	Wide Cut	Rider	Lawn	Garden	Auto Mower
	Model Number		M3	M5	M11	M17	M19	M23	M100
	Market Segment		Sm Resident	Med Resident	Med 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 IV: ASELCM Pattern, Trusted Models, Effective Group Learning

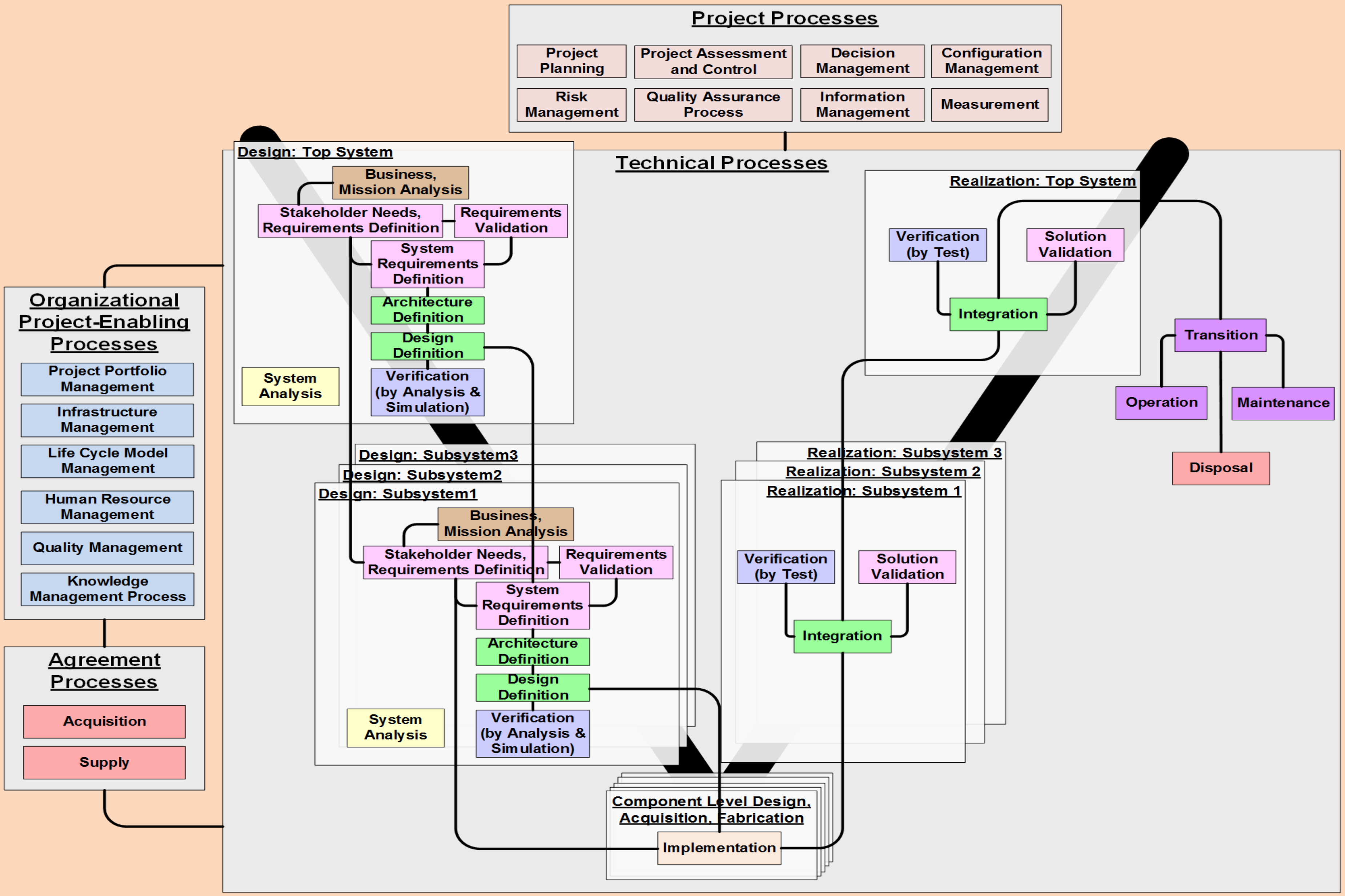
(Used for INCOSE Agile SE Project, INCOSE CIPR WG, etc. Generic innovation reference model: Descriptive, not prescriptive.)



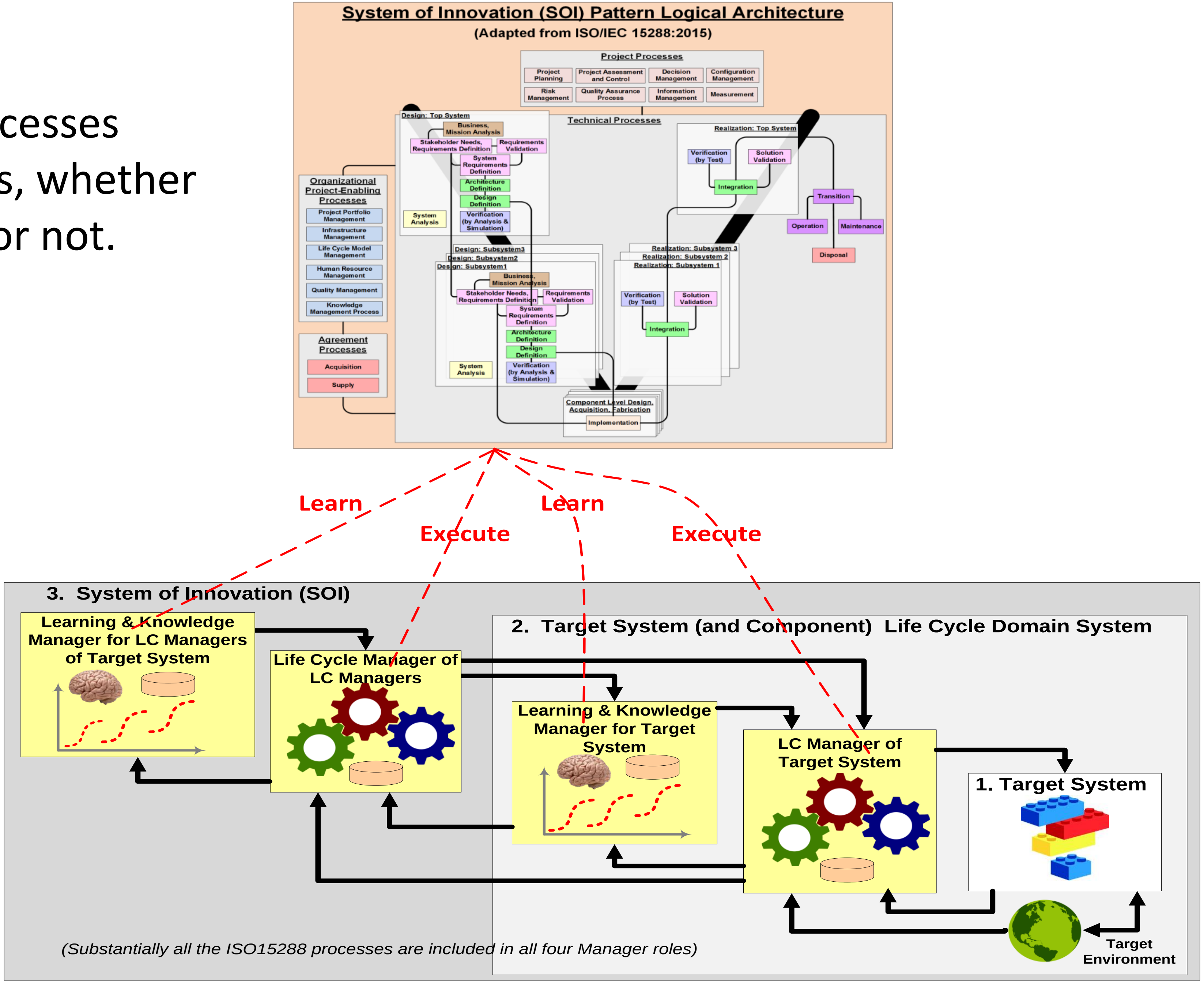
- System 1: Target system of interest, to be engineered or improved.
- System 2: The environment of (interacting with) S1, including all the life cycle management systems of S1, including learning about S1.
- System 3: The life cycle management systems for S2, including learning about S2.

System of Innovation (SOI) Pattern Logical Architecture

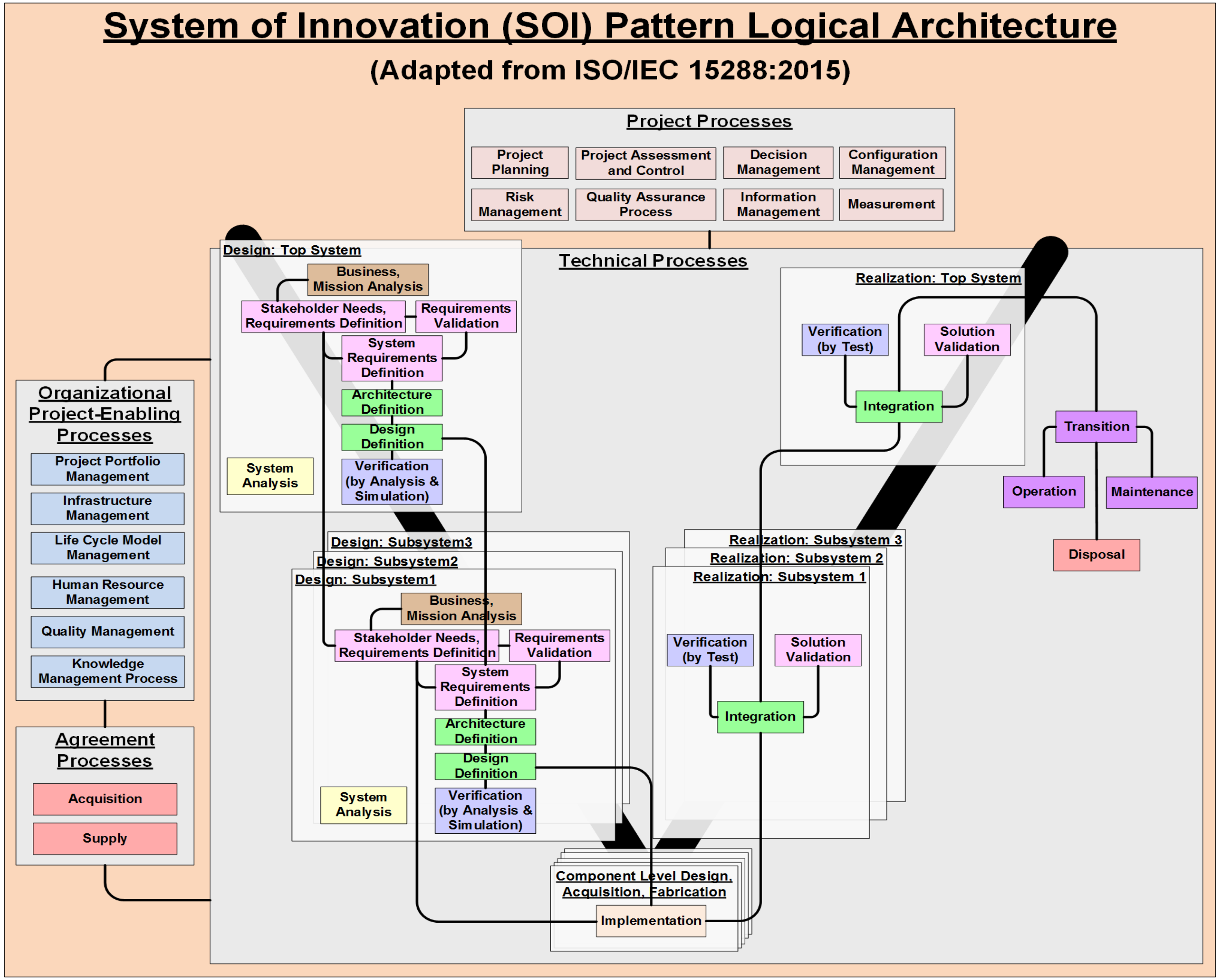
(Adapted from ISO/IEC 15288:2015)



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



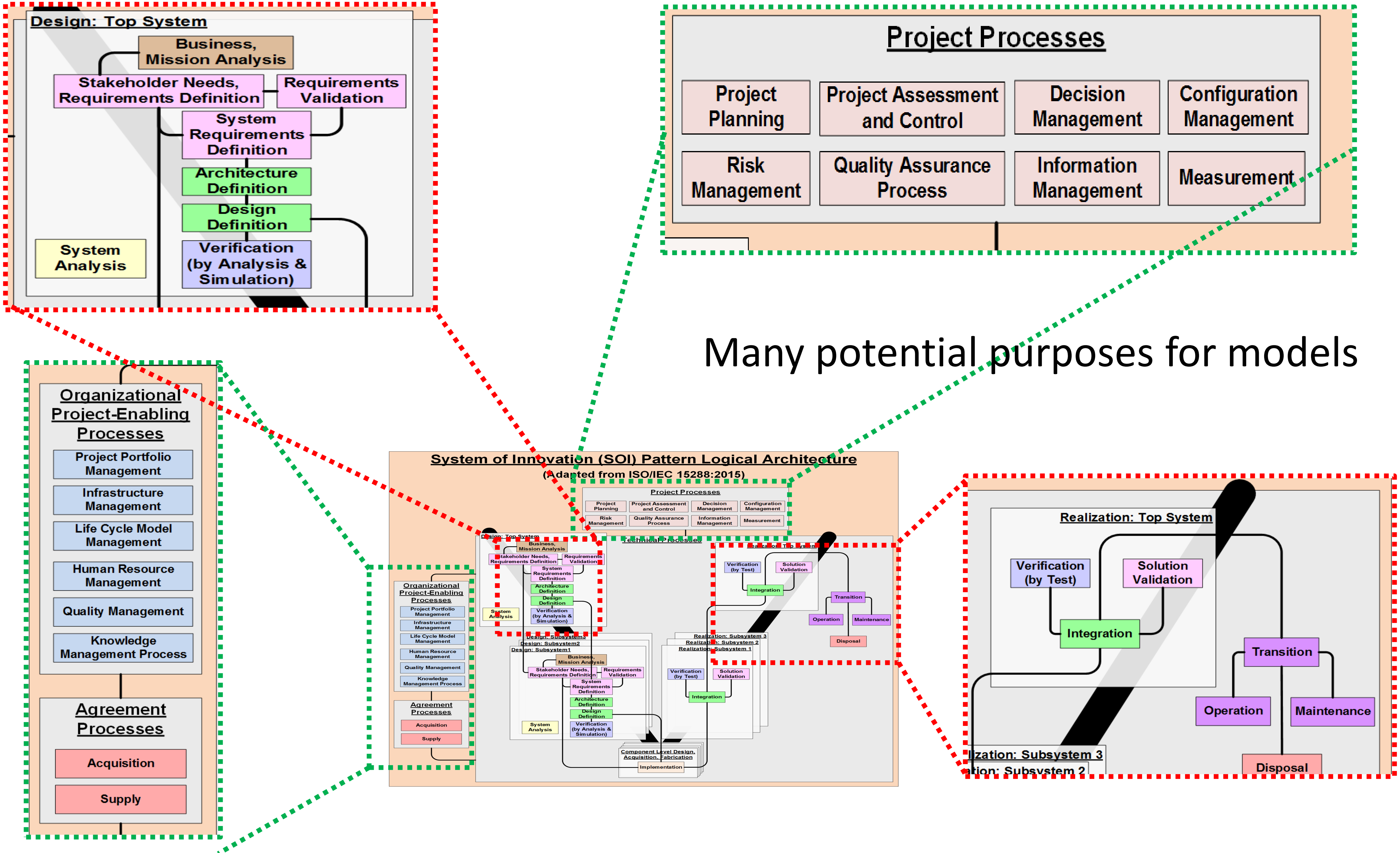
Models for what purposes? Possible ISO15288 answers



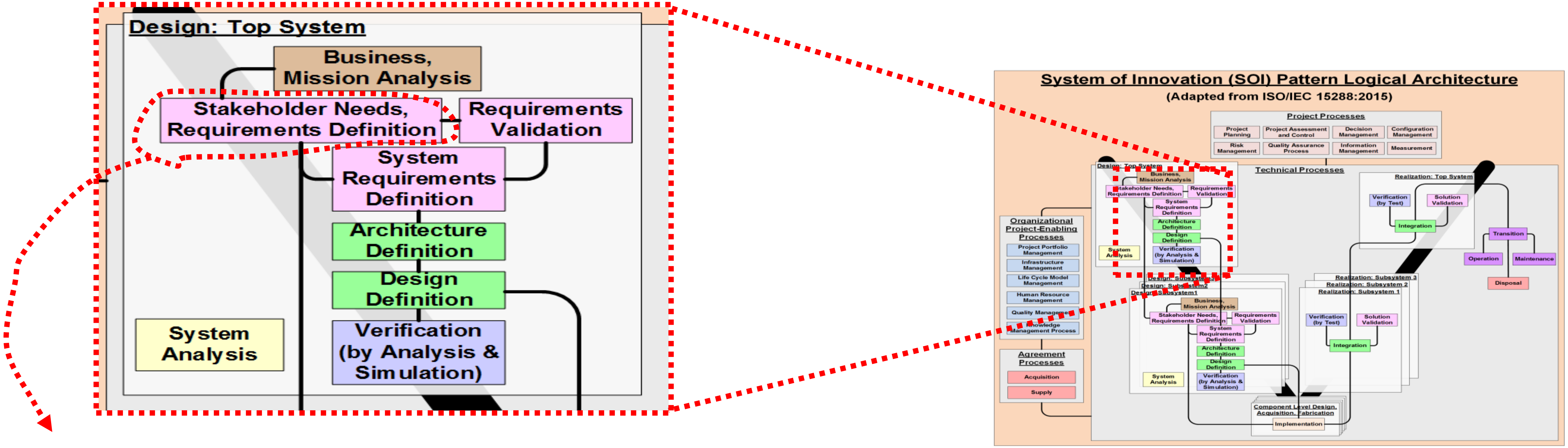
Potentially for any ISO 15288 processes:

- 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



a) “Stakeholders of the system are identified.
b) Required characteristics and context of use of capabilities and concepts in the life cycle stages, including operational concepts, are defined.
c) Constraints on a system are identified.
d) Stakeholder needs are defined.
e) Stakeholder needs are prioritized and transformed into clearly defined stakeholder requirements.
f) Critical performance measures are defined.
g) Stakeholder agreement that their needs and expectations are reflected adequately in the requirements is achieved.
h) Any enabling systems or services needed for stakeholder needs and requirements are available.
i) 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

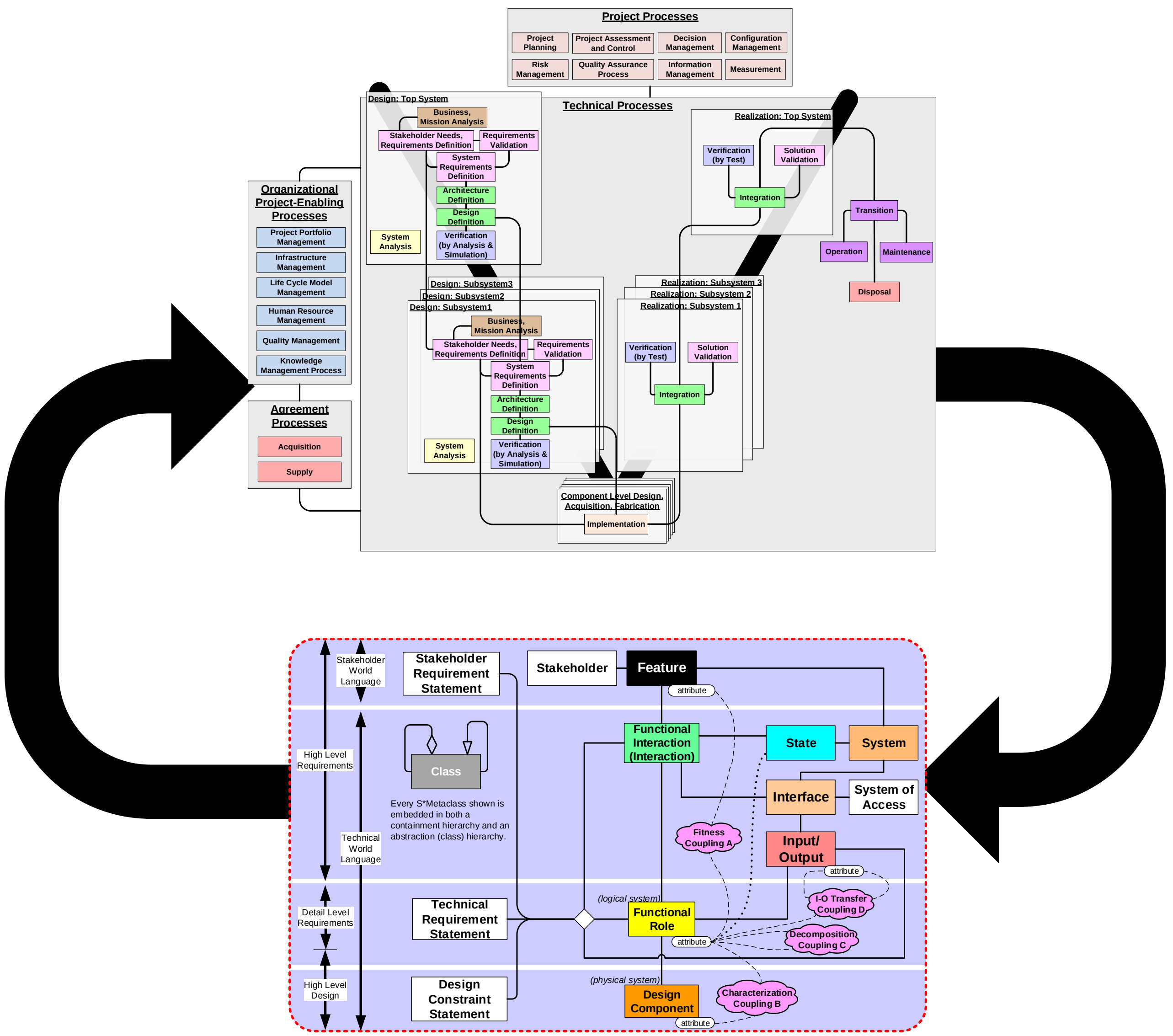
The INCOSE MBSE Transformation products are based on identification of --

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

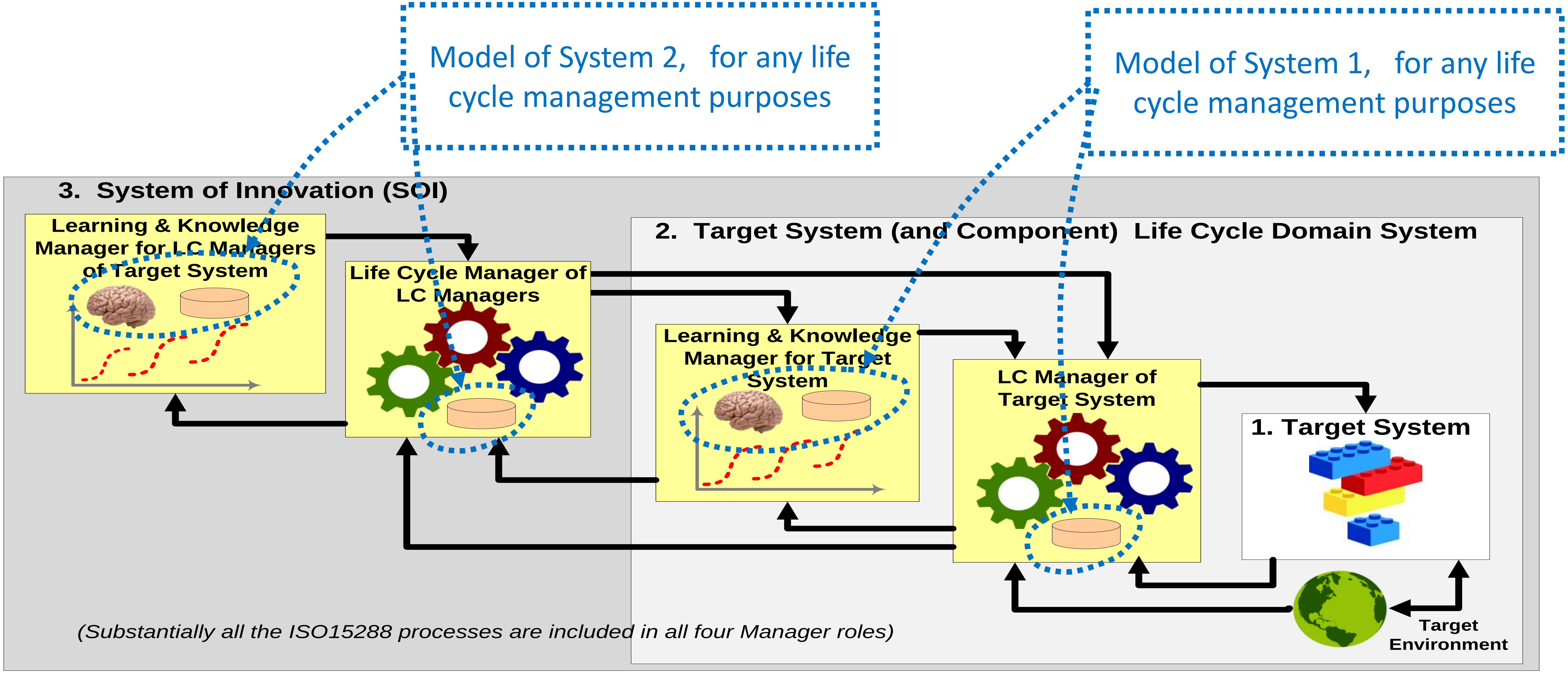


Models help make this real:

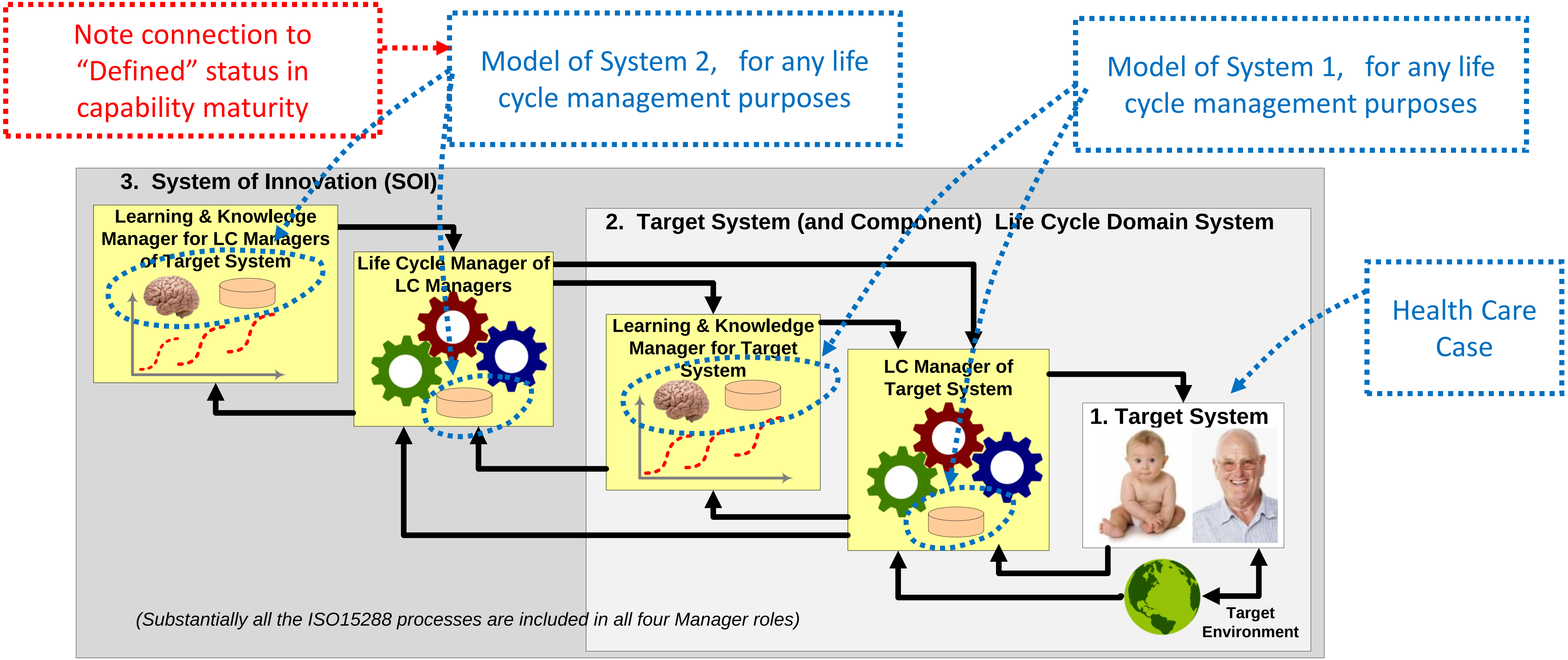
Shifting the emphasis from traditional focus on process and procedure, to greater emphasis on the state of the web of information passing through that process and procedure.

Compare to the traditional engineering disciplines.

Information Passing Through Innovation Process

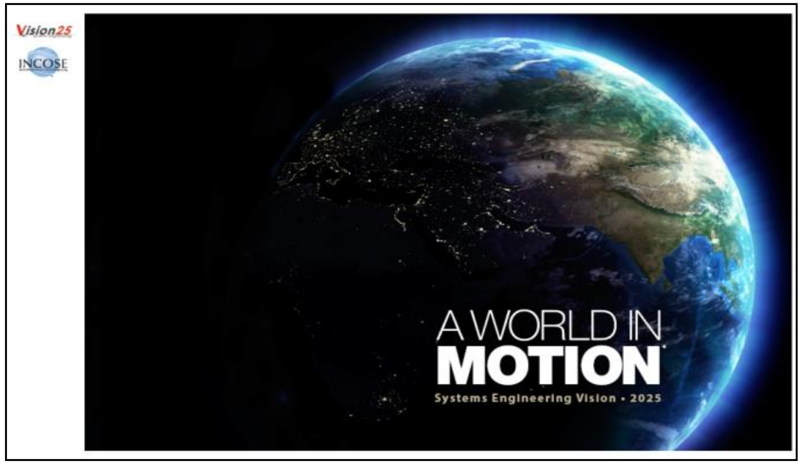


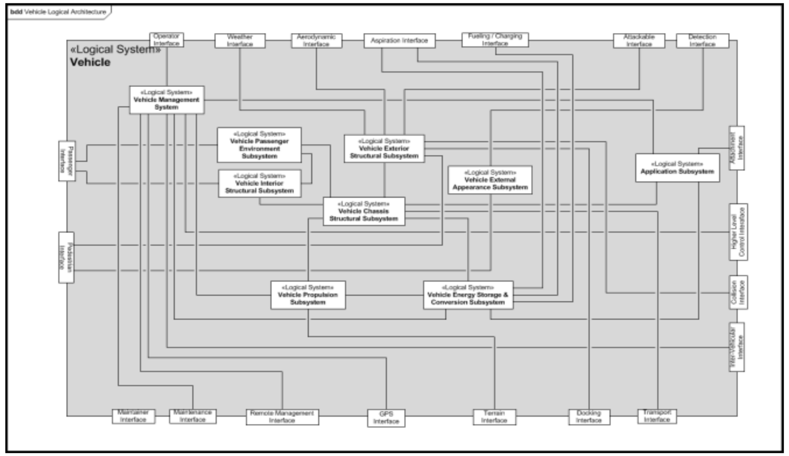
- System 1: Target system of interest, to be engineered or improved.
- System 2: The environment of (interacting with) S1, including all the life cycle management systems of S1, including learning about S1.
- System 3: The life cycle management systems for S2, including learning about S2.

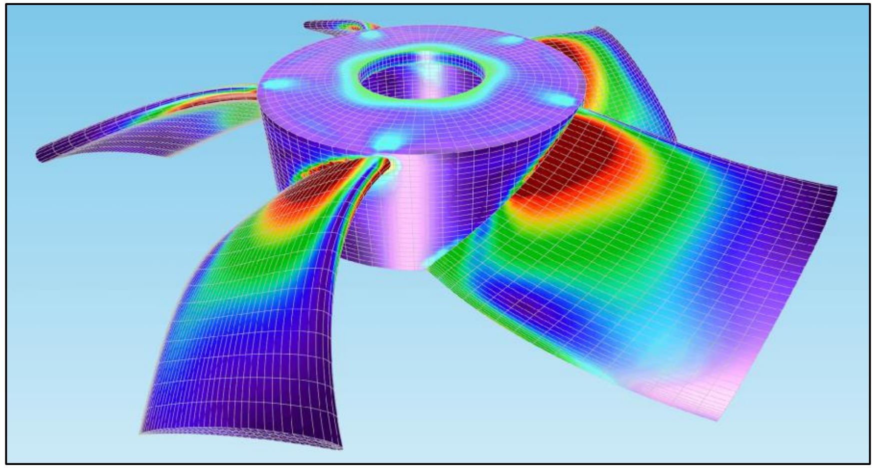


- System 1: Target system of interest, to be engineered or improved.
- System 2: The environment of (interacting with) S1, including all the life cycle management systems of S1, including learning about S1.
- System 3: The life cycle management systems for S2, including learning about S2.

Enthusiasm for Models



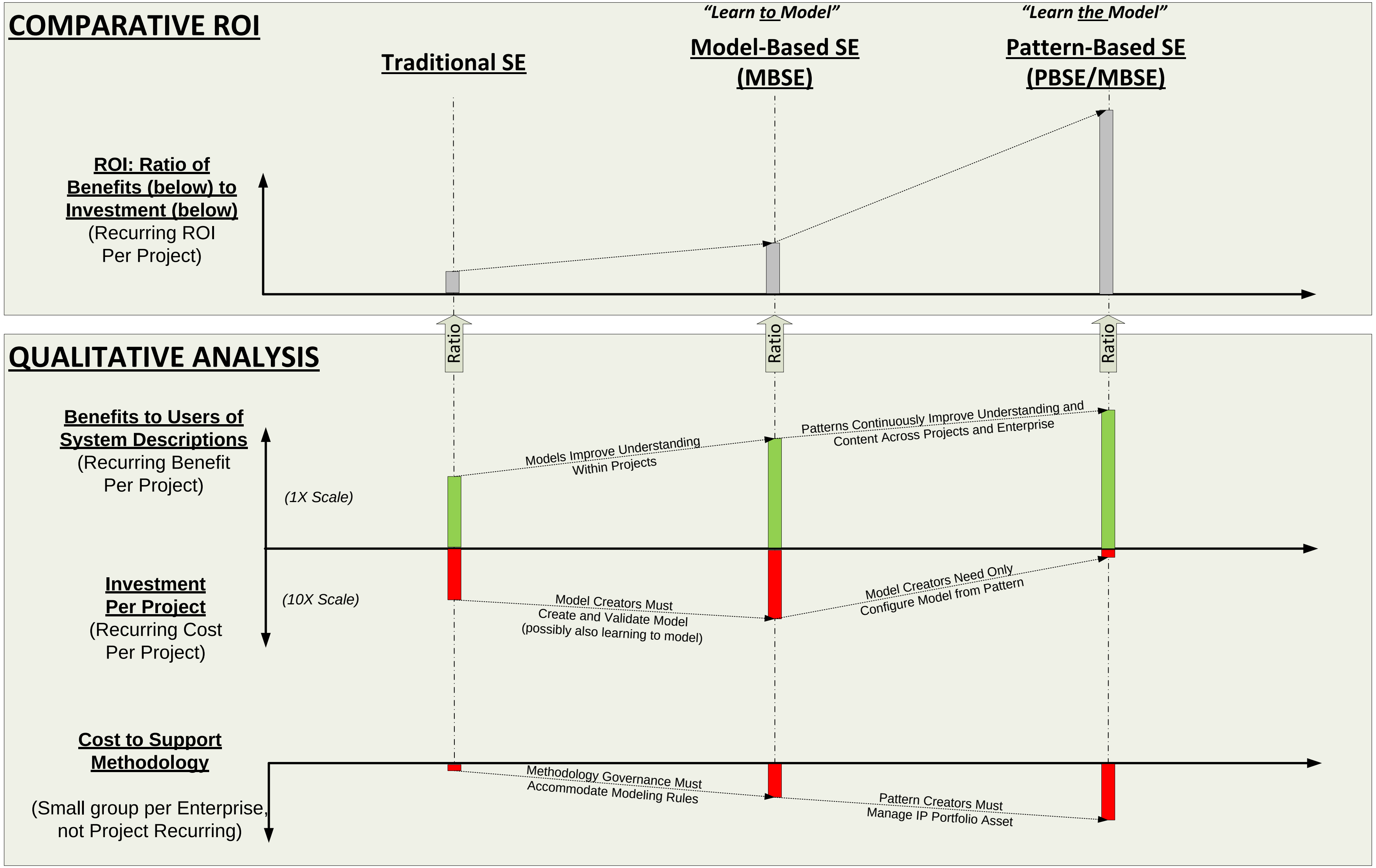




The INCOSE systems community has shown growing enthusiasm for "engineering with models" of all sorts:

- 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 & Gvmt. Initiatives	Organizations Internalizing MBSE, Including Gvmt Contractors & Commercial	Vendors of MBSE Tooling and Services	Academia and Researchers	Technical Societies, Other Non-Technical Organizations	
<u>Model Consumers (Model Users):</u>							
****	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	
<u>Model Creators (including Model Improvers):</u>							
*	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	
<u>Complex Idea Communicators (Model "Distributors"):</u>							
**	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	
<u>Model Infrastructure Providers, Including Tooling, Language and Other Standards, Methods:</u>							
*	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	
<u>INCOSE and other Engineering Professional Societies</u>							
*	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	

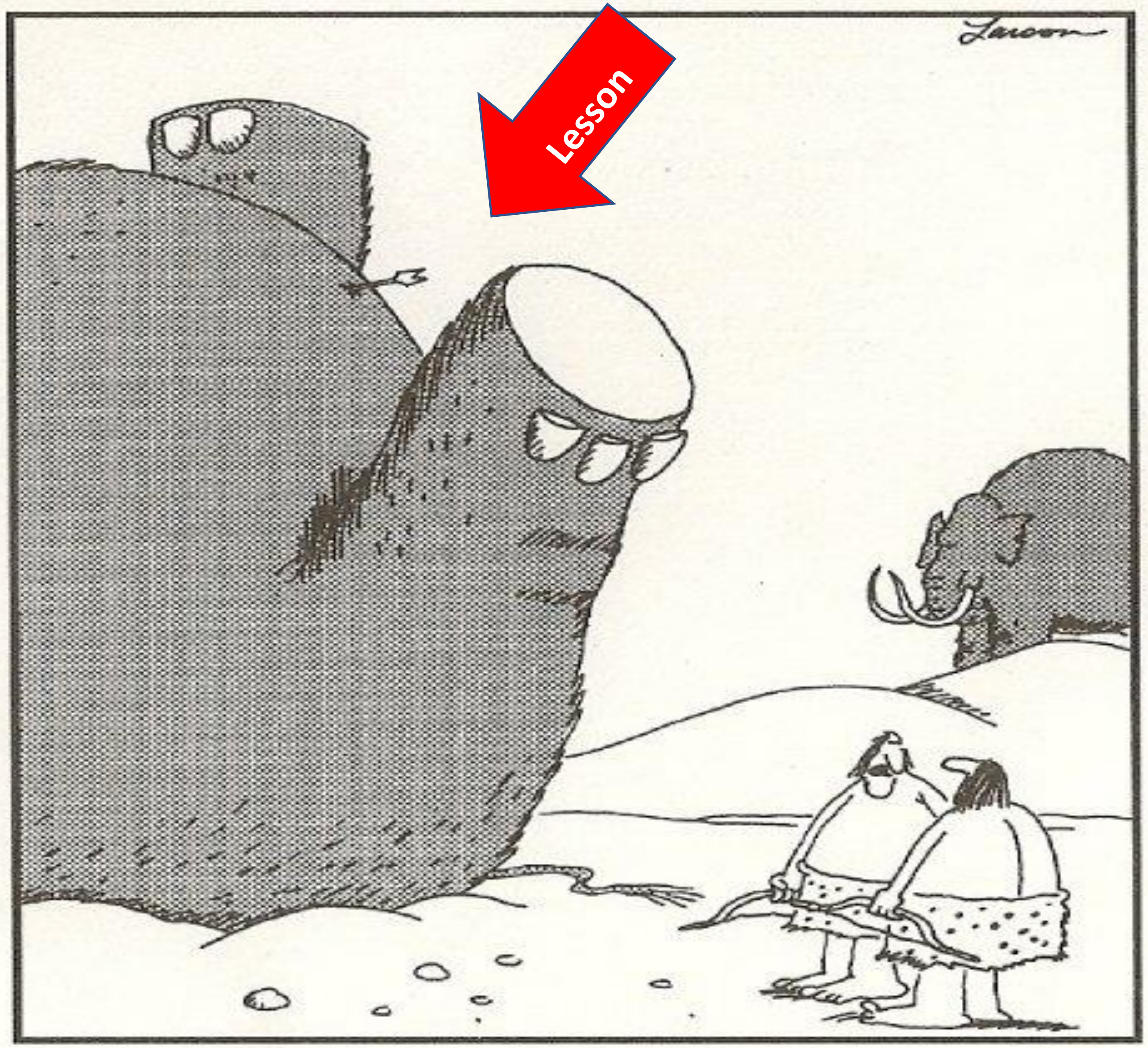
Lessons Learned: Effective Learning?

- 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.

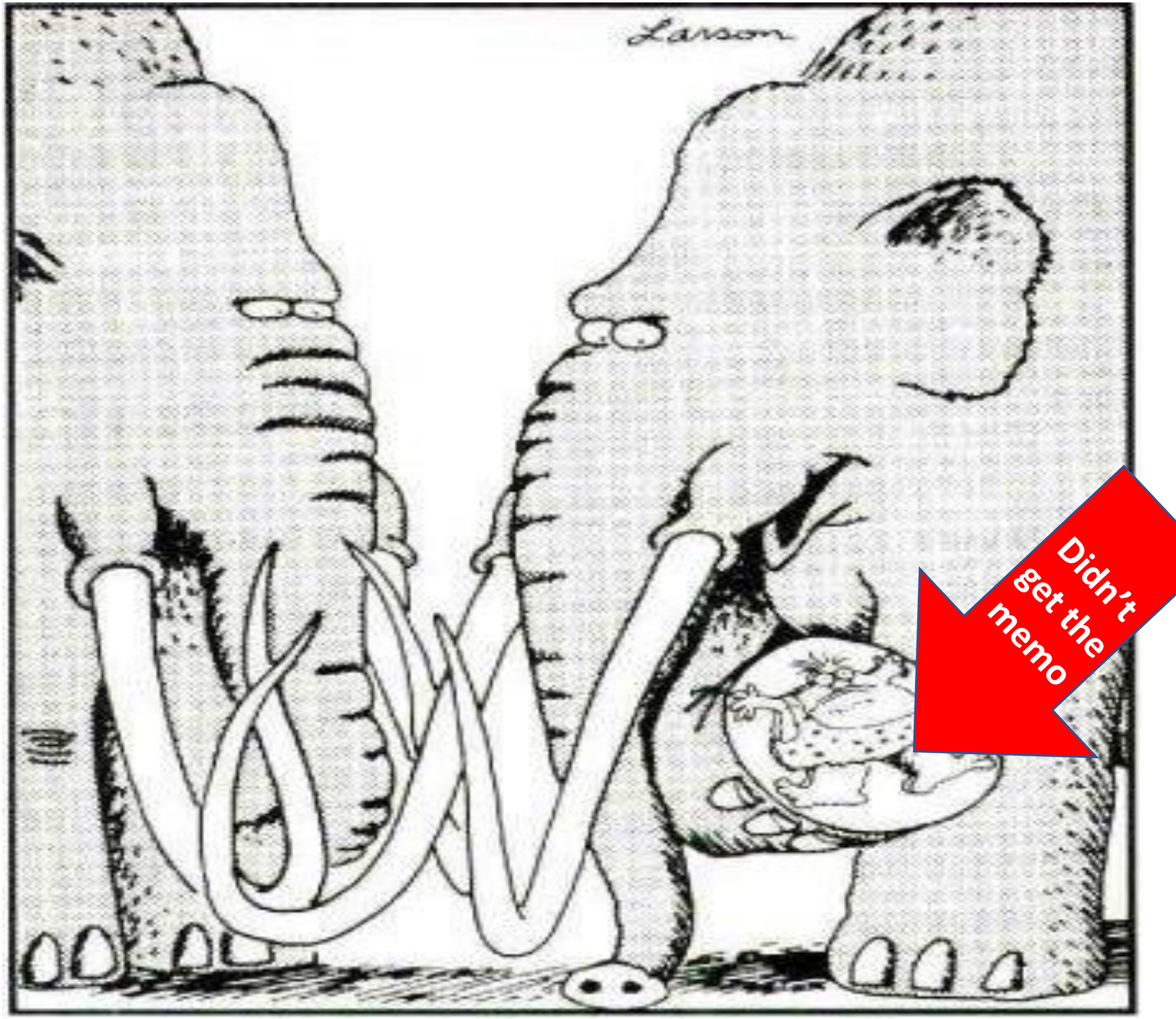


"We should write that spot down."

Lessons Effectively Learned?

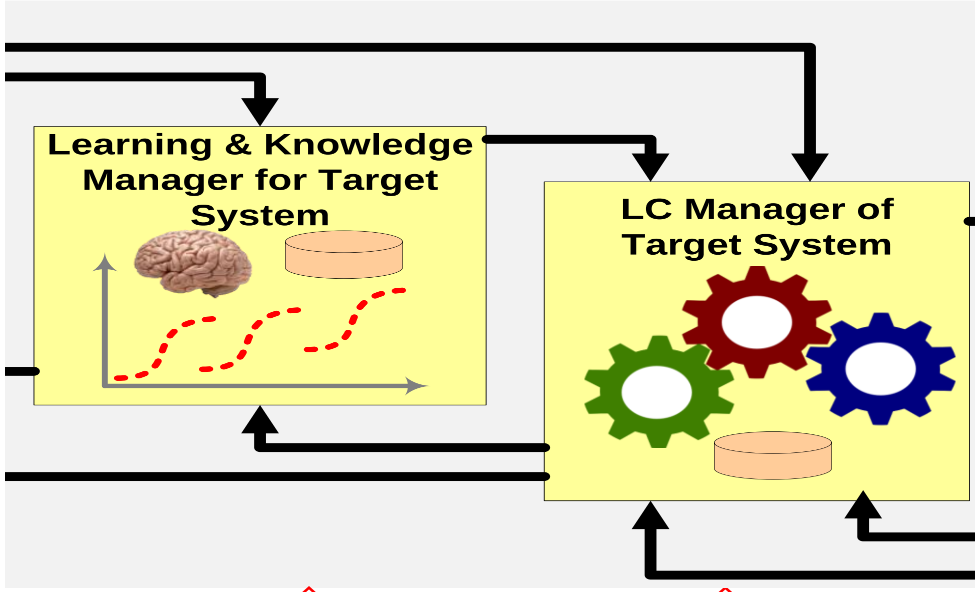
Lessons Learned Report

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.

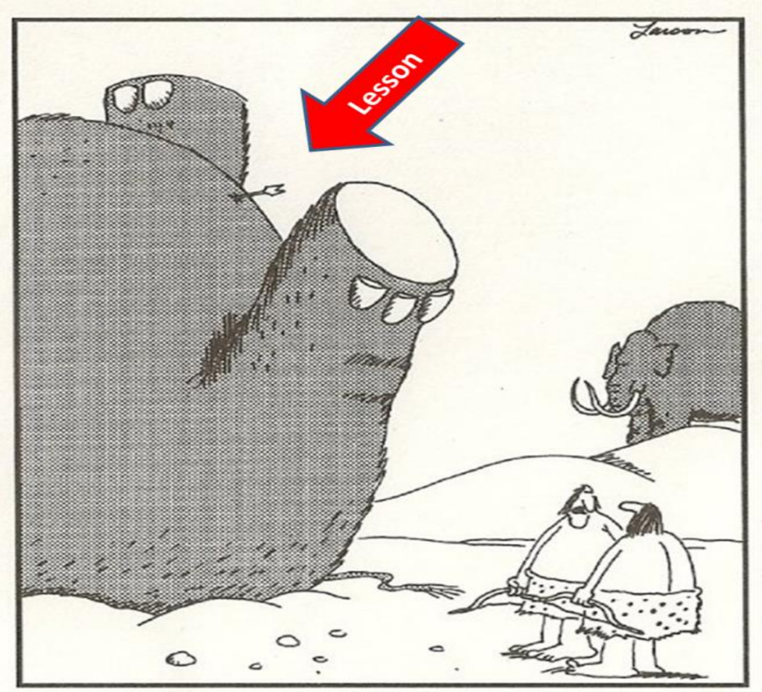


"Well, what the? ... I *thought* I smelled something."

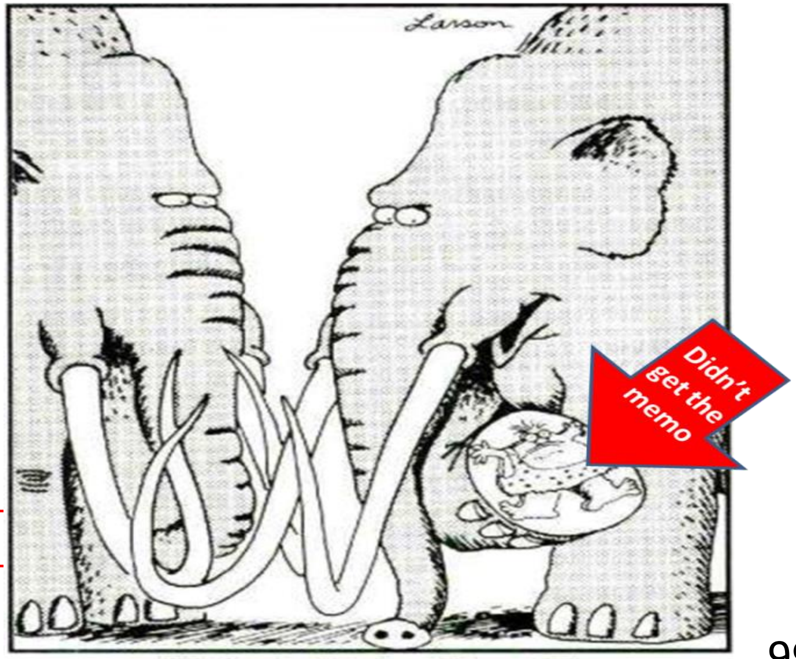
Learning



Executing

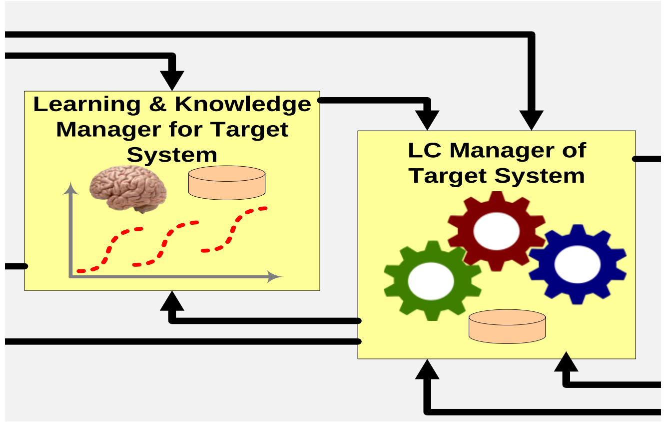


"We should write that spot down."



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

Lessons Learned: <u>Effective</u> Learning?



- <u>Where</u> are the "lessons learned" encoded? What would cause them to be <u>accessed</u>?
- Compare to biology:
  - "Muscle Memory" builds "motor" learning directly <u>*into a future situation*</u>, for future unconscious use, <u>*vs*</u>. 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.

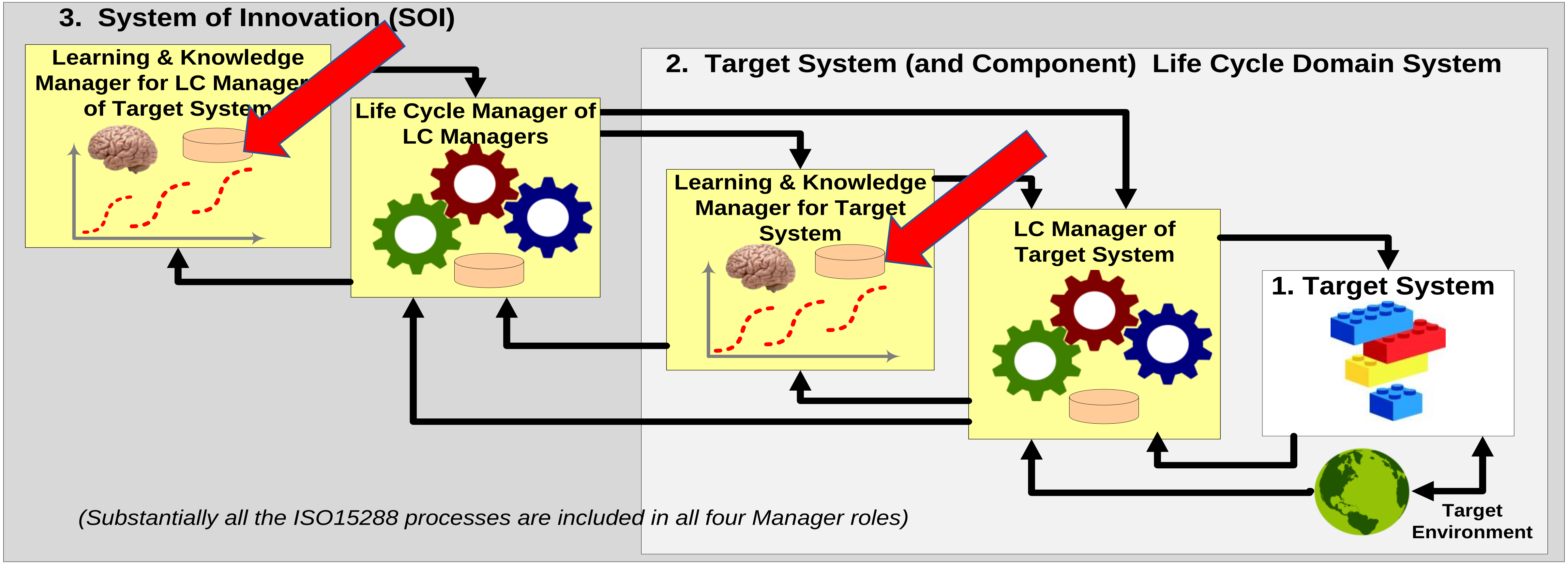
Learned models from STEM (~300 years) offer the most dramatic example of positive collaborative impact of effectively shared and validated models

- 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