GLRC 2018

2018 Annual INCOSE
Great Lakes Regional Conference
SYSTEMS AT THE CROSSROADS
17 - 20 October 2018 | Indianapolis, Indiana

Tutorial: Emerging Issues in Application of Model-Based Systems Engineering (MBSE)

Bill Schindel
ICTT System Sciences
schindel@ictt.com

1.4.2

Abstract: This tutorial is concerned with emerging issues in applying Model-Based Systems Engineering (MBSE), in two categories, and is divided into two half-day sessions:

- **Part I (Morning)**: Planning and Assessing Your Path to Value from MBSE--
  - In its earliest years, MBSE enthusiasm has been focused on technical model content and methodology, tools, languages, and standards. As MBSE reaches for mainstream use, larger groups of non-technical stakeholders are involved, and larger questions of strategy and paths forward for propagation appear. This tutorial session will address key developments emerging from efforts toward standardization and transformation, being pursued in two professional societies in particular (ASME and INCOSE). In Part I, attendees will learn how to apply the planning framework, and take a copy home to use. Attendees will also learn about introducing re-usable MBSE Patterns into work processes, and learn how to get started addressing model credibility issues.

- **Part II (Afternoon)**: Applying MBSE Patterns for Increased Leverage: Examples from Smart Manufacturing and the Internet of Things (IoT)--
  - Models are interesting to construct, and modelers are enthusiastic to do so. However, the business case for originating a "clean sheet" model for each project grows weaker as systems become more complex, as more is at stake, and as the demands for model content and credibility grow. This tutorial session will address the use of MBSE Patterns—formal models that are configurable and re-usable for different projects—as pursued in recent years by the INCOSE MBSE Patterns Working Group. In Part II, attendees will learn about the Embedded Intelligence Pattern and the Smart Manufacturing Pattern. Attendees will also learn about the strategy of financial capitalization of MBSE Patterns.

Introduction of Tutorial Participants

Thanks to Harry Potter.

Tutorial Summary Outline

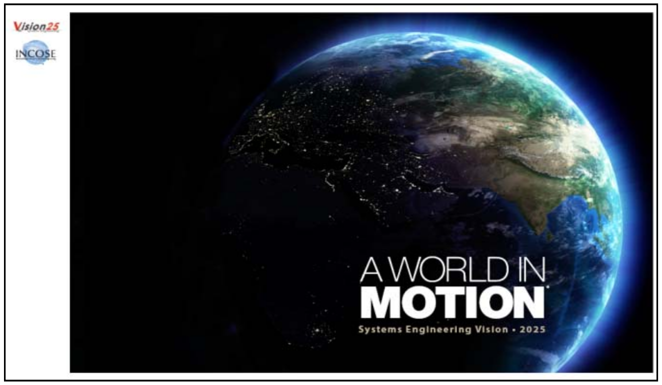
Part I (Morning):

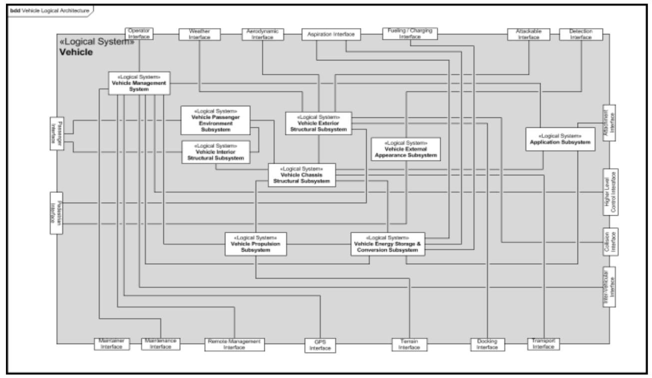
- **Targeting Purpose**: Planning development, use, and life cycle of models based on a standard model planning framework, neutral as to modeling tools, languages, methods
- **Institutionalizing Learning**: Practical steps to improve on organizational learning, using models as a focus of organizational learning and knowledge, based on model-based Learning Systems and Autonomous Systems.
- **Enabling Trust**: Can You Trust Someone Else's Model? Your Model? Planning for Model Verification, Validation, and Uncertainty Quantification (VVUQ)

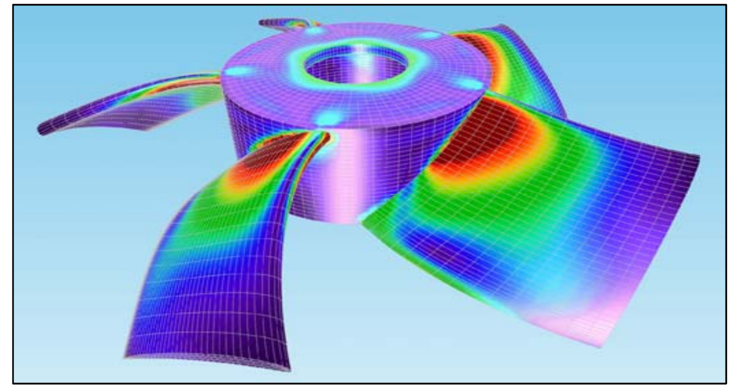
Part II (Afternoon):

- **Representing Intelligence**: The Embedded Intelligence (EI) Pattern, for any embedding of intelligence, in the form of automation, human operators, or other systems of management, feedback, regulation.
- **Advancing Production**: The Smart Manufacturing Pattern, for the IoT Age, for any manufacturing process, and with varied forms of instrumentation and management.
- **Capitalizing IP** of MBSE Patterns as Financial Assets, to shift the burden of model cost to the time of model use and benefit.

Enthusiasm for Models



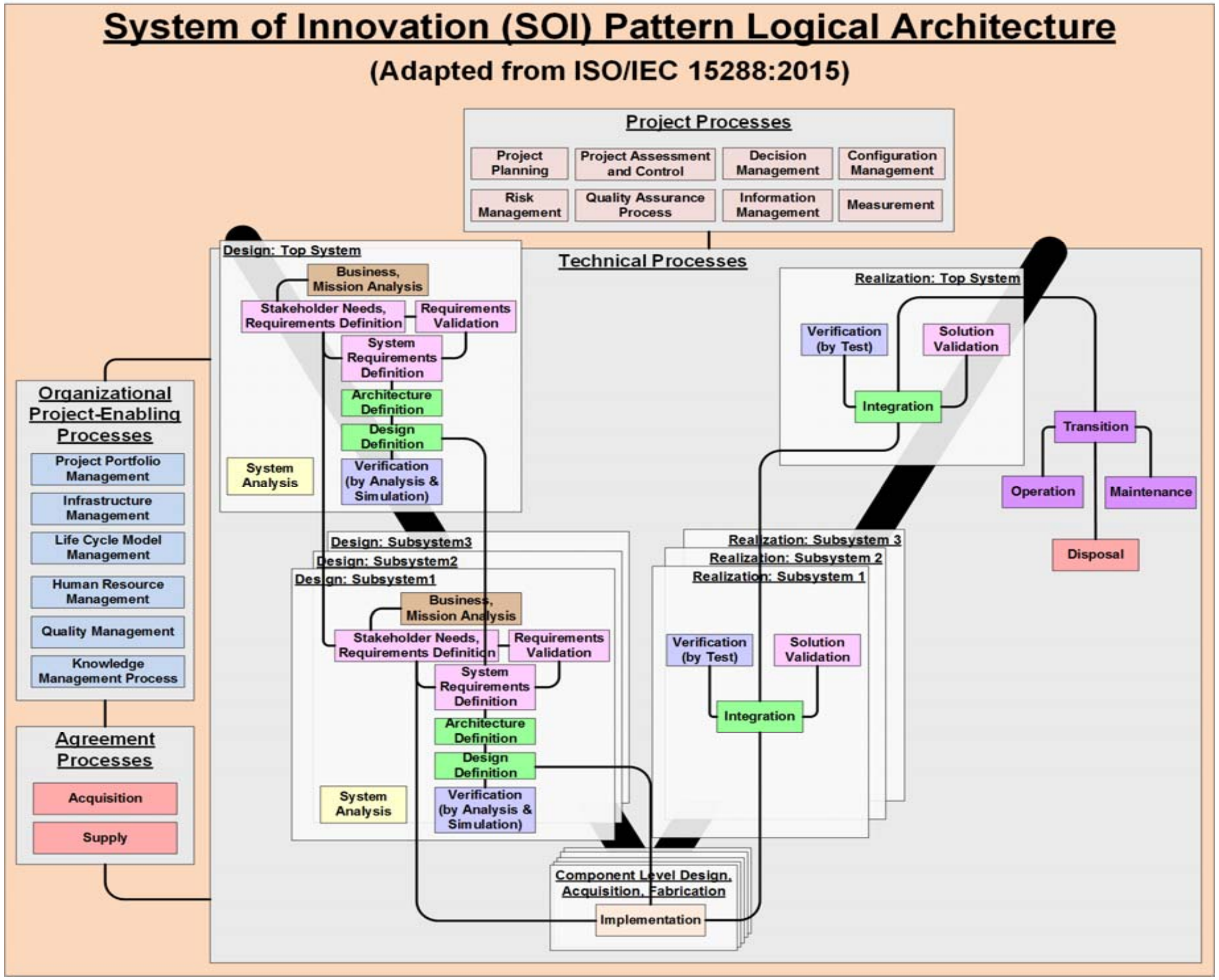




The INCOSE systems community has shown growing enthusiasm for "engineering with models" of all sorts:

- 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

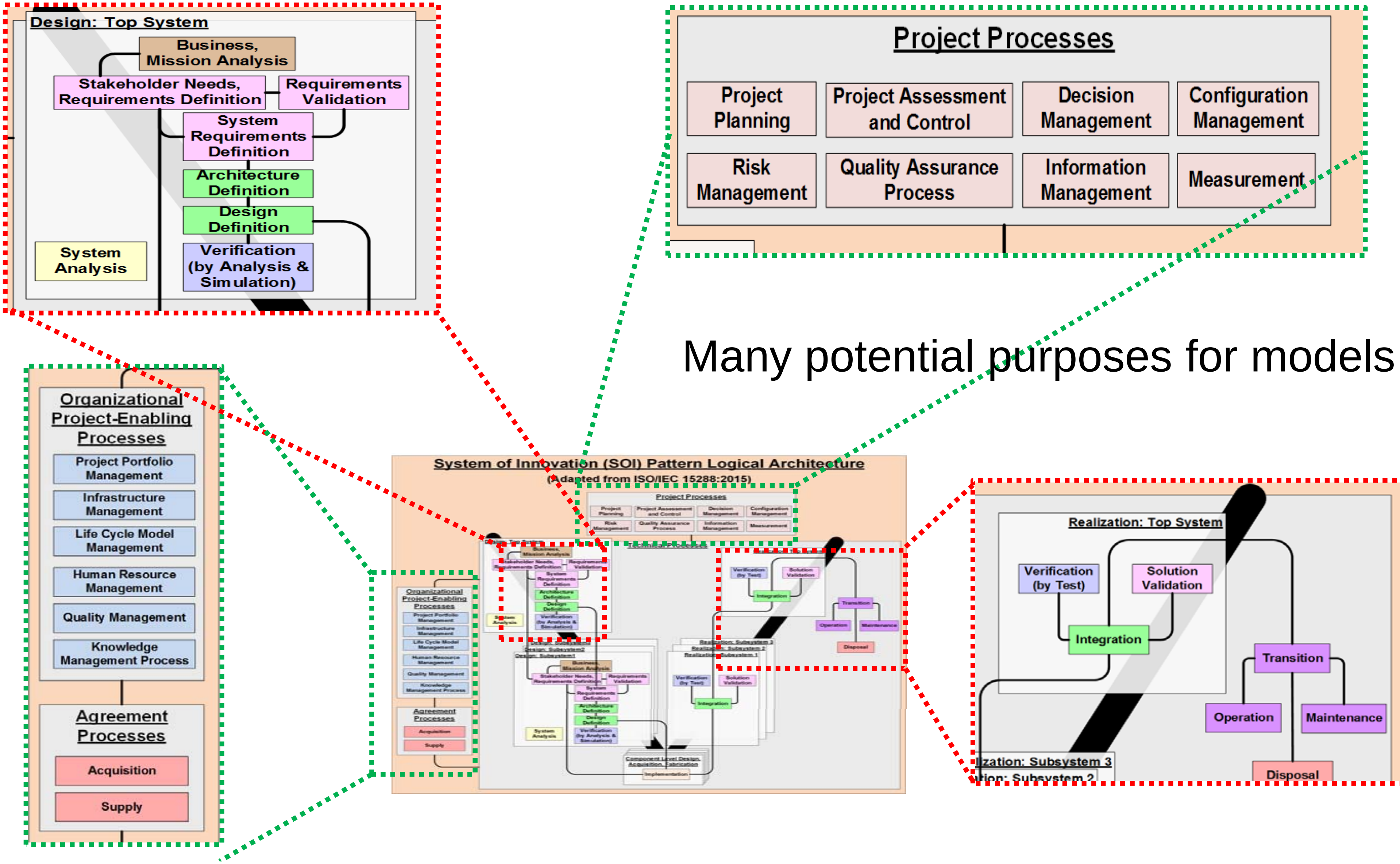
Models for what <u>purposes</u>? Possible ISO15288 answers.



<u>Potentially</u> for any ISO 15288 processes:

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

Many potential purposes for models



Targeting Purpose: Connections to ISO15288

- Model-based methods have multiple connections to ISO15288 system life cycle management practices:
  - The INCOSE Model-Based Transformation project provides means for assessing and planning the migration of ISO15288 practices to model-based approaches.
  - The INCOSE Agile SE Life Cycle Management Discovery Project provides inputs to a future version of ISO15288 including agile SE, and includes the model-based ASELCM Pattern and its representation of the roles of models in innovation.
  - The INCOSE MBSE Patterns Working Group supports improving the leverage of model-based practices using formal S*Patterns, and is partnering with ASME toward standards for the verification and validation of computational models for ISO15288 purposes.
- This tutorial will summarize how these efforts are being fit together to provide usable practitioner value, and how to get involved.

Targeting Purpose: Connections to ISO15288

- Maturity in MBSE is not only about our models, methods, and tools--although it includes them:
  - What will we use models <u>for</u> (intended purpose)?   Who is "we"?
  - How do we go about <u>trusting</u> our model?
  - Is our <u>learning</u> effectively enhanced?
- State of art & practice in some of these areas still low:
  - So, expect significant continuing change.
  - Measuring against current base may not reflect "maturity".
- There are overall requirements we can use to measure our MBSE maturity:
  - Based on, but enlarging, the interpretation of ISO 15288, existing maturity models, and computational models.
  - Providing a foundation for future maturity assessment, planning.
- The emerging foundation opens up thinking about scope of impacts, and therefore scope of maturity assessment.

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.

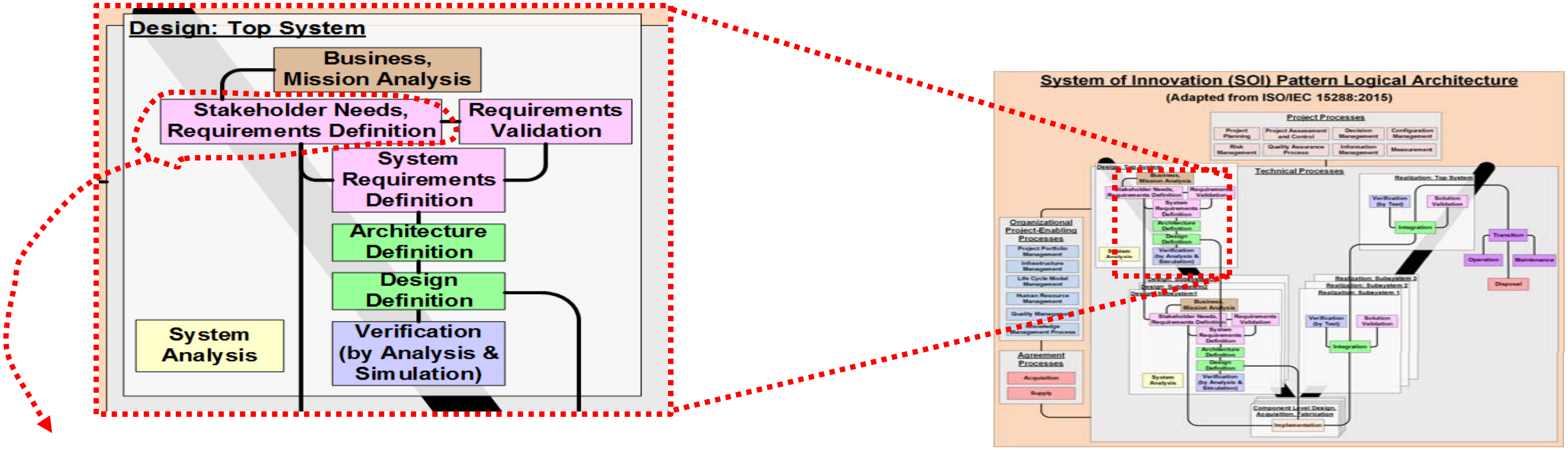
Further analysis of the Transformation Stakeholders

(also shows Energy Tech 2016 Conference ratings of needs, opportunities)

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

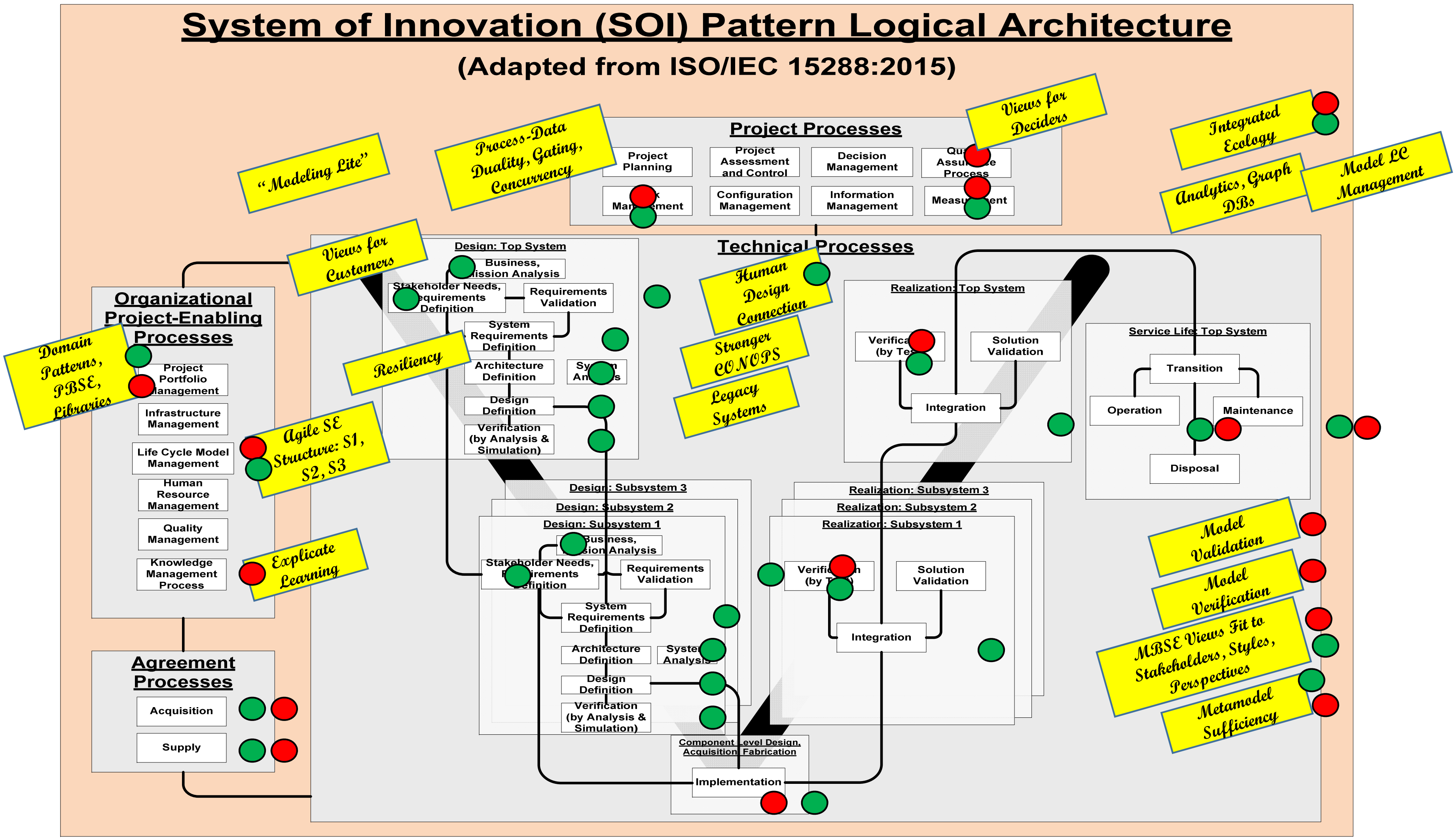
12

Each 15288 process definition suggests potentially assessable model impacts



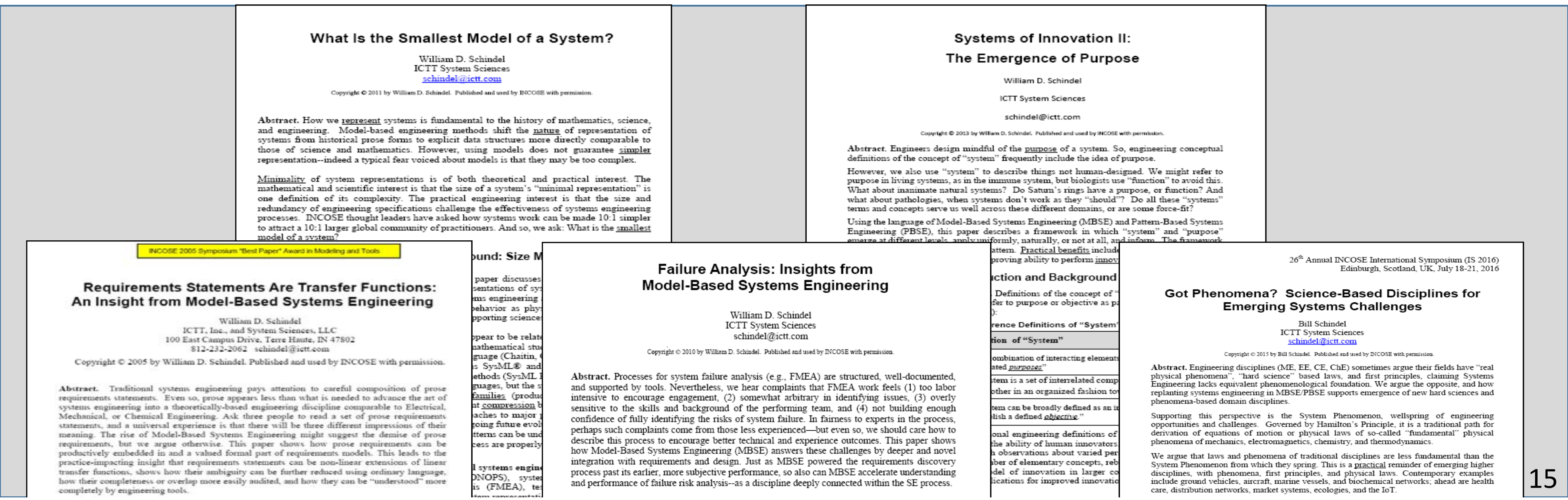
a) “Stakeholders of the system are identified.
b) Required characteristics and context of use of capabilities and concepts in the life cycle stages, including operational concepts, are defined.
c) Constraints on a system are identified.
d) Stakeholder needs are defined.
e) Stakeholder needs are prioritized 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.”

Each ISO15288 process offers higher level targeting, assessment
(Example: Energy Tech 2016 Feedback on MBSE in ISO15288)



Sufficiency for Purposes; Minimality

- Systems of Modeling, practiced, must be sufficient for their intended purposes, and preferably minimal / not overly complex, proliferated:
  - A lot of (continuing) effort by the modeling community being invested in sufficiency and also minimality.
  - Understanding of what is needed improving, but lists of future capabilities are long.
- More is involved than modeling languages, tools, methods, alone; for example:
  - Fitness to non-technical users and uses
  - Strong enough conceptual foundation, based on STEM, not just information models.
  - Credibility of model content (trust in the model)


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 to attract a 10:1 larger global community of practitioners. And so, we ask: What is the smallest model of a system?

Systems of Innovation II:
The Emergence of Purpose

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

Copyright © 2013 by William D. Schindel. Published and used by INCOSE with permission.

Abstract. Engineers design mindful of the purpose of a system. So, engineering conceptual definitions of the concept of "system" frequently include the idea of purpose.

However, we also use "system" to describe things not human-designed. We might refer to purpose in living systems, as in the immune system, but biologists use "function" to avoid this. What about inanimate natural systems? Do Saturn's rings have a purpose, or function? And what about pathologies, when systems don't work as they "should"? Do all these "systems" terms and concepts serve us well across these different domains, or are some force-fit?

Using the language of Model-Based Systems Engineering (MBSE) and Pattern-Based Systems Engineering (PBSE), this paper describes a framework in which "system" and "purpose" emerge at different levels, apply uniformly, naturally, or not at all, and inform. The framework

INCOSE 2005 Symposium "Best Paper" Award in Modeling and Tools

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

William D. Schindel
ICTT, Inc., and System Sciences, LLC
100 East Campus Drive, Terre Haute, IN 47802
812-232-2062 schindel@ictt.com

Copyright © 2005 by William D. Schindel. Published and used by INCOSE with permission.

Abstract. Traditional systems engineering pays attention to careful composition of prose requirements statements. Even so, prose appears less than what is needed to advance the art of systems engineering into a theoretically-based engineering discipline comparable to Electrical, Mechanical, or Chemical Engineering. Ask three people to read a set of prose requirements statements, and a universal experience is that there will be three different impressions of their meaning. The rise of Model-Based Systems Engineering might suggest the demise of prose requirements, but we argue otherwise. This paper shows how prose requirements can be productively embedded in and a valued formal part of requirements models. This leads to the practice-impacting insight that requirements statements can be non-linear extensions of linear transfer functions, shows how their ambiguity can be further reduced using ordinary language, how their completeness or overlap more easily audited, and how they can be "understood" more completely by engineering tools.

Failure Analysis: Insights from
Model-Based Systems Engineering

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

Copyright © 2010 by William D. Schindel. Published and used by INCOSE with permission.

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

26th Annual INCOSE International Symposium (IS 2016)
Edinburgh, Scotland, UK, July 18-21, 2016

Got Phenomena? Science-Based Disciplines for Emerging Systems Challenges

Bill Schindel
ICTT System Sciences
schindel@ictt.com

Copyright © 2015 by Bill Schindel. Published and used by INCOSE with permission.

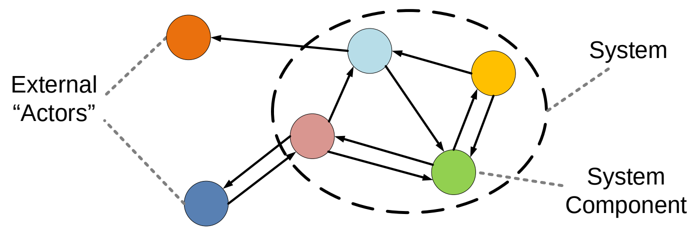
Abstract. Engineering disciplines (ME, EE, CE, ChE) sometimes argue their fields have "real physical phenomena", "hard science" based laws, and first principles, claiming Systems Engineering lacks equivalent phenomenological foundation. We argue the opposite, and how replanting systems engineering in MBSE/PBSE supports emergence of new hard sciences and phenomena-based domain disciplines.

Supporting this perspective is the System Phenomenon, wellspring of engineering opportunities and challenges. Governed by Hamilton's Principle, it is a traditional path for derivation of equations of motion or physical laws of so-called "fundamental" physical phenomena of mechanics, electromagnetics, chemistry, and thermodynamics.

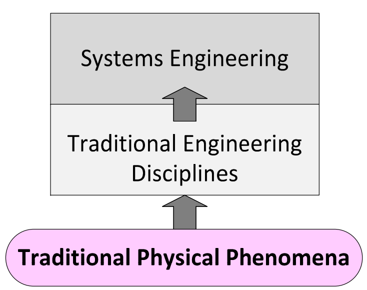
We argue that laws and phenomena of traditional disciplines are less fundamental than the System Phenomenon from which they spring. This is a practical reminder of emerging higher disciplines, with phenomena, first principles, and physical laws. Contemporary examples include ground vehicles, aircraft, marine vessels, and biochemical networks; ahead are health care, distribution networks, market systems, ecologies, and the IoT.

Scientific heritage (~300 years)

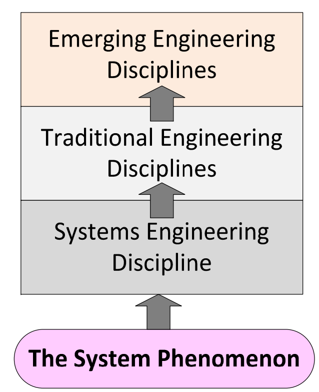
- The eventual flowering of the physical sciences depended upon the emergence of strong enough underlying model constructs (of math, physics) to better represent Nature.
- Specifically, the System Phenomenon (Newton, Lagrange, Hamilton, Noether):



A traditional view of systems engineering

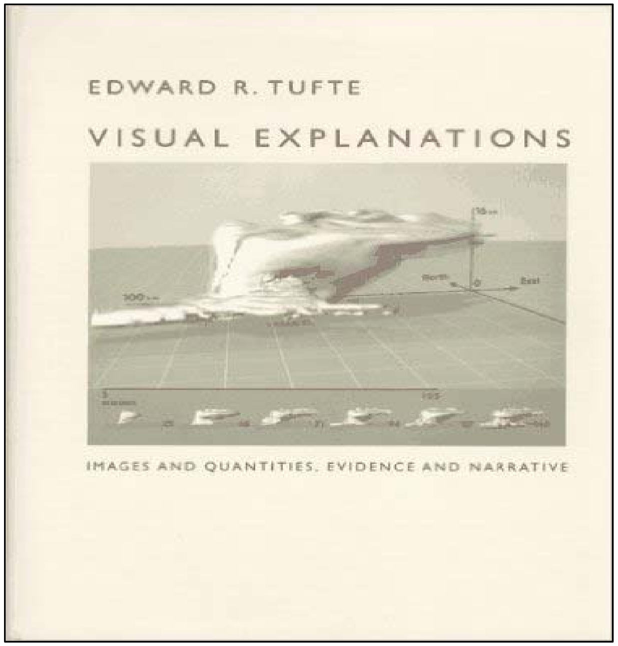


Our view of systems engineering



Sufficiency for Purposes; Minimality

- Example: Fitness of model to use
  - Includes fitness of model views to intended uses, users.
- See discussions by E. Tufte, N Levinson, concerning NASA shuttle model views
- Culture plays a key part in this.
- So, measuring maturity of MBSE will take us across more subjects than technical practitioners might expect.



- Modeling more than just the "engineered" System 1
- Intended model uses and users, along with culture, are "System 2" issues . . . .

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.

INCOSE MBSE Assessment and Planning Pattern: Model Stakeholder Features Overview

Model Identity and Focus

- **Modeled System of Interest**
  - System of Interest
- **Modeled Environmental Domain**
  - Domain Type

Model Utility

- **Model Intended Use**
  - LIFE CYCLE PROCESS SUPPORTED (ISO15288)
- **Perceived Model Value and Use**
  - USER GROUP SEGMENT
  - Level of Annual Use
  - Value Level
- **Third Party Acceptance**
  - ACCEPTING AUTHORITY
- **Model Ease of Use**
  - Perceived Model Complexity

Model Scope and Content

- **Modeled Stakeholder Value**
  - STAKEHOLDER TYPE
- **Modeled System External (Black Box) Behavior**
- **Explanatory Decomposition**
- **Failure Modes and Effects**
- **Parametric Couplings--Fitness**
- **Parametric Couplings--Decomposition**
- **Parametric Couplings--Characterization**
- **Trusted Configurable Pattern**
  - CONFIGURATION ID
  - Pattern Type
- **Physical Architecture**
- **Managed Model Datasets**
  - DATASET TYPE

Model Credibility

- **Model Envelope**
  - MODEL APPLICATION ENVELOPE
- **Validated Conceptual Model Credibility**
  - Quantitative Accuracy Reference
  - Function Structure Accuracy Reference
  - Uncertainty Quantification (UQ) Reference
  - Model Validation Reference
- **Verified Executable Model Credibility**
  - Quantitative Accuracy Reference
  - Function Structure Accuracy Reference
  - Uncertainty Quantification (UQ) Reference
  - Speed
  - Quantization
  - Stability
  - Model Validation Reference

Model Representation

- **Conceptual Model Representation**
  - Conceptual Model Representation Type
  - Conceptual Model Interoperability
- **Executable Model Representation**
  - Executable Model Representation Type
  - Executable Model Interoperability

Model Life Cycle Management

- **Model Versioning and Configuration Management**
  - CM CAPABILIY TYPE
- **Model Maintainability**
  - Maintenance Method
- **Model Deployability**
  - Deployment Method
- **Model Cost**
  - Development Cost
  - Operational Cost
  - Maintenance Cost
  - Deployment Cost
  - Retirement Cost
  - Life Cycle Financial Risk
- **Executable Model Environmental Compatibility**
  - IT ENVIRONMENTAL COMPONENT
- **Model Design Life Cycle and Retirement**
  - Design Life
- **Model Availability**
  - First Availability Date
  - First Availability Risk
  - Life Cycle Availability Risk
- **VVUQ Pattern Learning**
  - VVUQ PATTERN EXCEPTION
  - Impacted VVUQ Feature
  - VVUQ Pattern Version
  - Project
  - Person

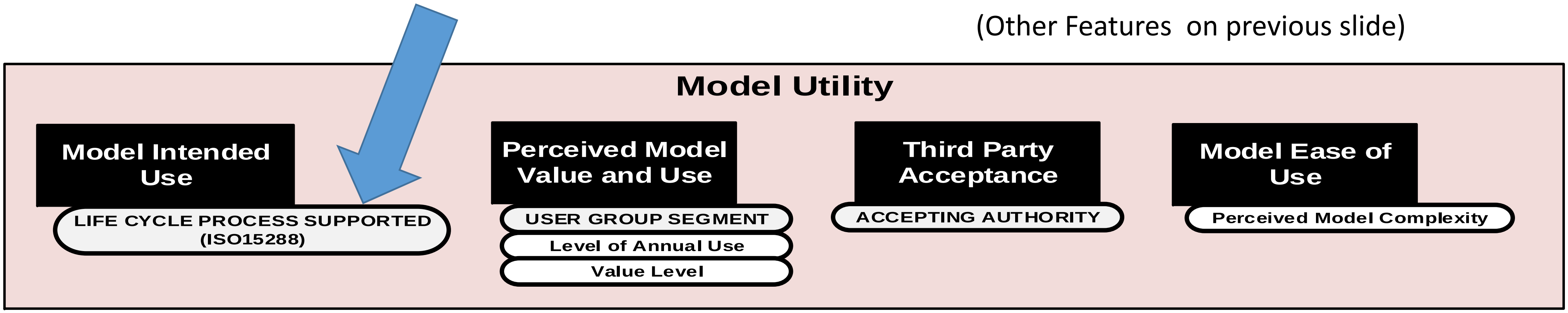
Legend:

- **STAKEHOLDER FEATURE**
  - FEATURE PK ATTRIBUTE
  - Other Feature Attribute
  - Other Feature Attribute

Stakeholder Feature Model for Computational Models		
Version: 1.5.4	Date: 31 Aug 2017	Drawn By: B Schindel

The ISO 15288 Processes provide the Model Stakeholder Feature Set for Planning & Assessment

(Other Features on previous slide)



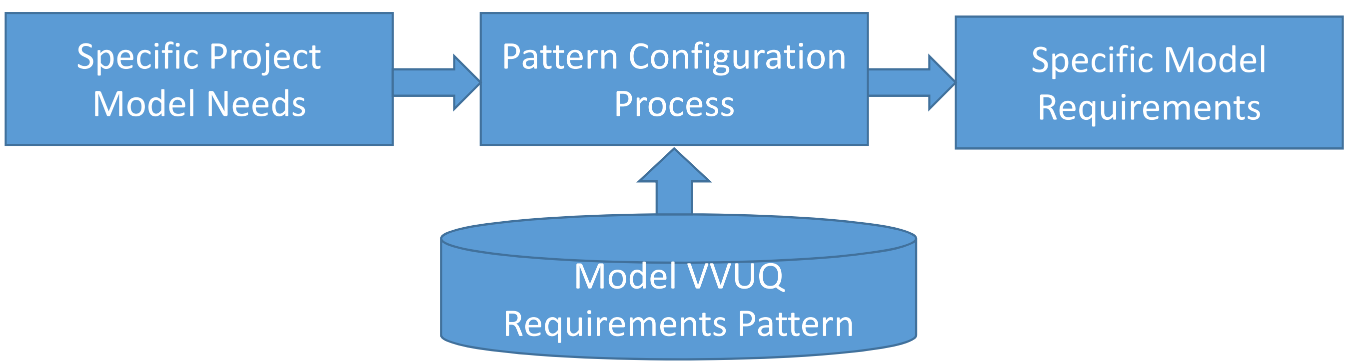
Feature Group	Feature Name	Feature Definition	Feature Attribute	Attribute Definition	Feature Stakeholder							Model Type	
					Model User	Model Developer	Model Maintainer	Mdl Deployer-Distributor	Model Use Supporter	Regulatory Authority	Mdl Investor-Owner	Physics Based	Data Driven
Describes the intended use, utility, and value of the model													
Model Utility	Model Intended Use	The intended purpose(s) or use(s) of the model.	Life Cycle Process Supported	The intended life cycle management process to be supported by the model, from the ISO15288 process list. More than one value may be listed.	**X**					**X**	**X**	x	x
	Perceived Model Value and Use	The relative level of value ascribed to the model, by those who use it for its stated purpose.	User Group Segment	The identify of using group segment (multiple)	**X**					**X**	**X**	x	x
			Level of Annual Use	The relative level of annual use by the segment	**X**					**X**	**X**	x	x
			Value Level	The value class associated with the model by that segment	**X**					**X**	**X**	x	x
	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	The identity (may be multiple) of regulators, agencies, customers, supply chains, accepting the model	**X**					**X**	**X**	x	x
	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	**X**					**X**		x	x

Vision for a Practical Aid to Model Community

- In establishing model credibility, a computational model is verified and validated (VV), including quantification of related uncertainties (UQ):
  - With respect to not just the system it represents, but also the Model Requirements, specifying the intended use(s), user(s), and characteristics of that model.
- This vision is to make the generation of those Model Requirements easier, more complete, and more successful than would otherwise be the case—using the Model VVUQ Pattern.

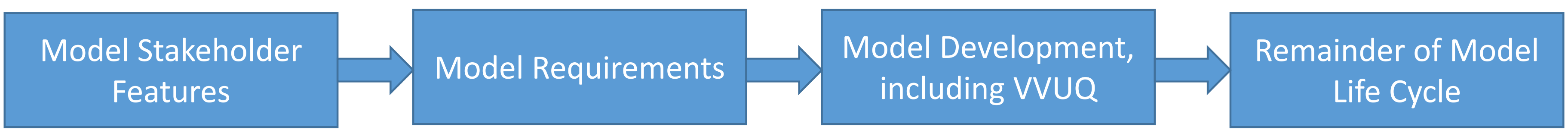
Vision for a
Practical Aid to Model Community

- Vision of a guideline that includes a practical pattern for the efficient and effective planning and generation of computational models that have a higher likelihood of VVUQ and successful service.
- The smallest set of ideas necessary to achieve that goal.
- Makes use of ideas used in Pattern-Based Systems Engineering, a form of MBSE, for configurable models:

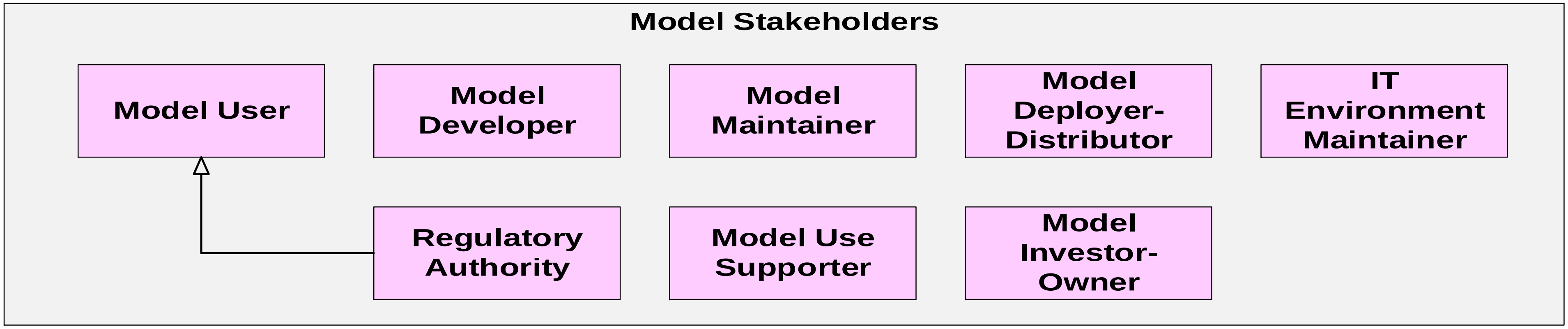


Vision for a Practical Aid to Model Community

- The foundation of this capability are the computational model's Stakeholder Features and the computational model's Requirements . . .



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.

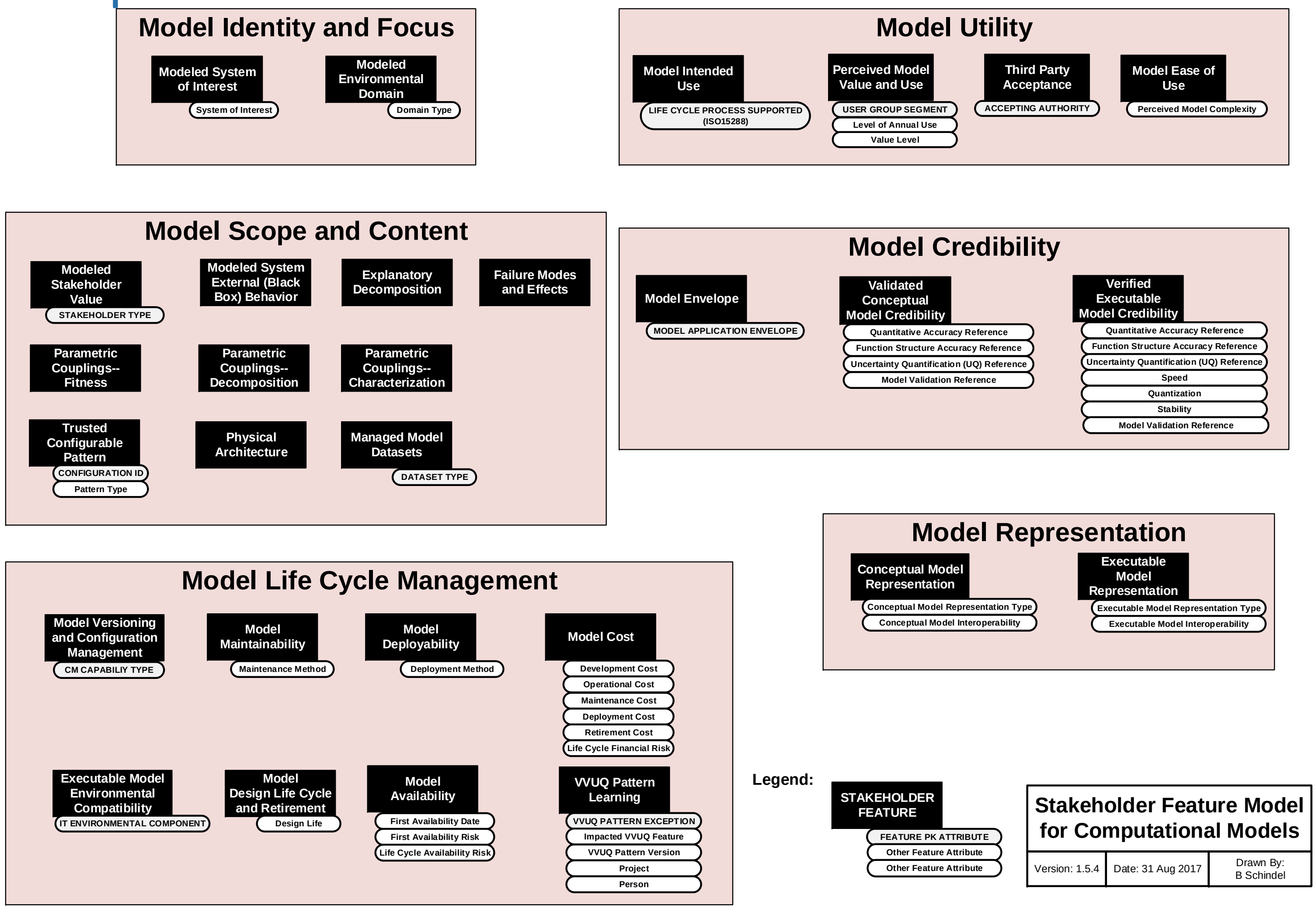
Model Life Cycle Management

Describes the related model life cycle management capabilities.

Model Representation

Describes the representation used by the model.

Computational Model Feature Groups: 27 Features, in 6 Feature Groups, Configurable for Specific Models

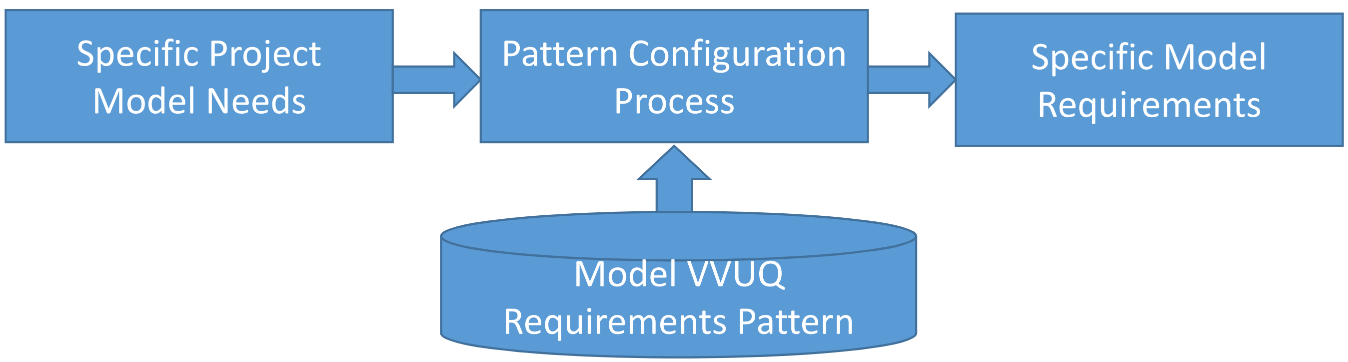


Computational Model Feature Groups: Configurable for Specific Models

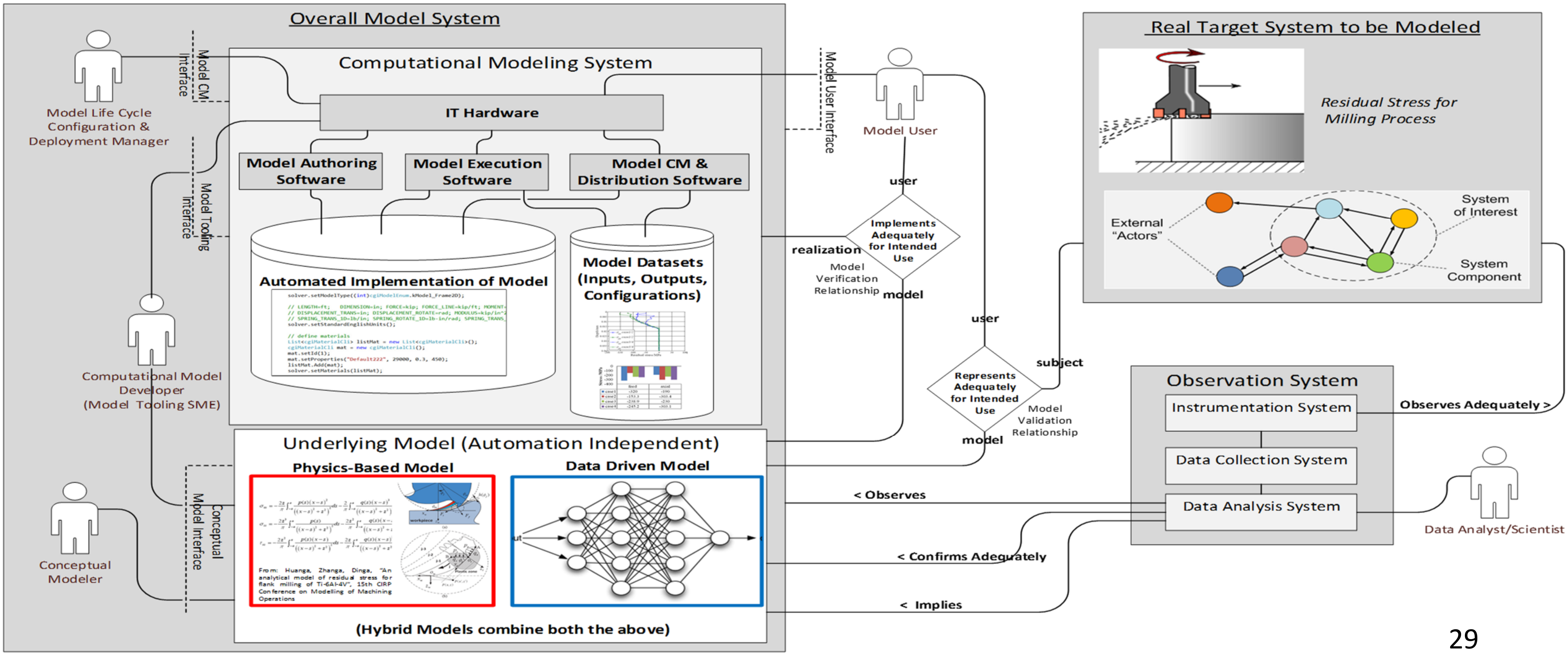
- The Stakeholder Features are configurable Stakeholder expectations, intentions, and valued aspects for a computational model:
  - These can be “configured” like Lego® blocks, as a form of checklist to rapidly create the stakeholder-level expectations for a computational model.
  - And from them, the more technical Requirements for the model follow.

Generation of Model Stakeholder Features

- The Model Stakeholder Feature Pattern is configured for a specific project by populating or depopulating the pattern's generic Features, and setting the values of its Feature Attributes:

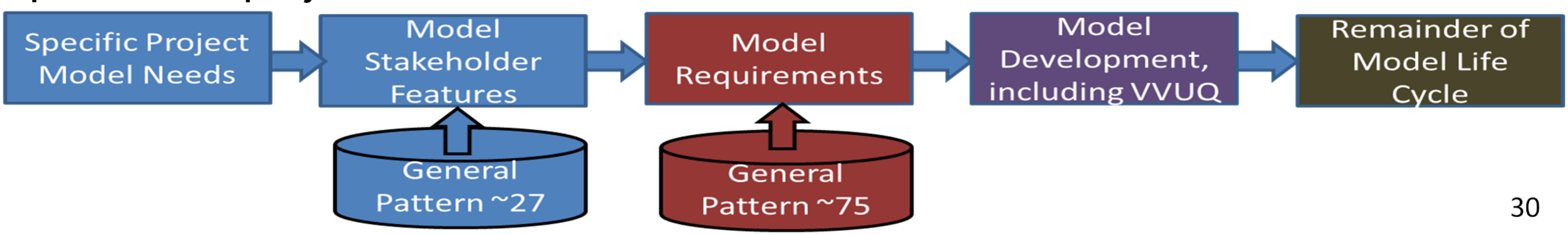


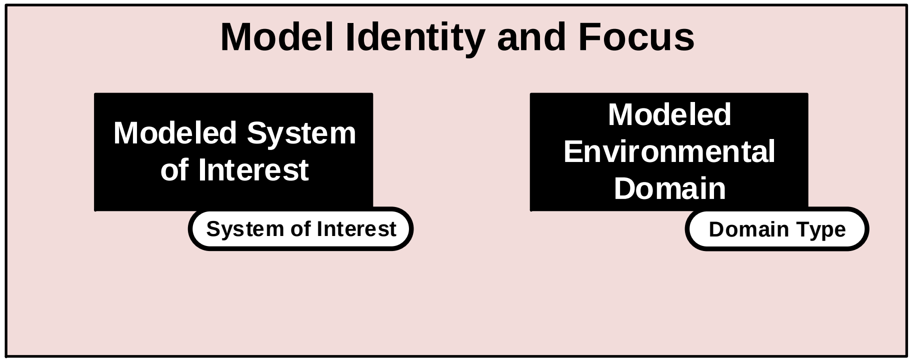
System Reference Boundaries: Computational Modeling Domain



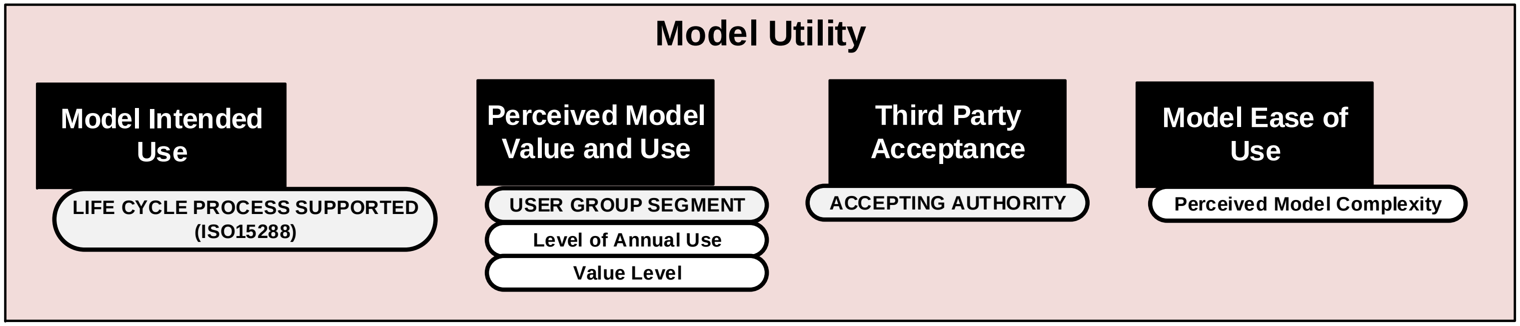
Requirements for Models

- Requirements for a specific computational model are the basis of subsequent validation and verification of the model.
- The Requirements for a computational model are implied by the Stakeholder Features (see above), but with more details configured into them.
- Approximately 75 configurable general Requirements for Models have been identified and traced to the Stakeholder Features, in the current draft of the Model VVUQ Pattern.
- After these have been further vetted and polished in this project, they provide a rapid start way to generate a high quality set of Model Requirements in a production project.

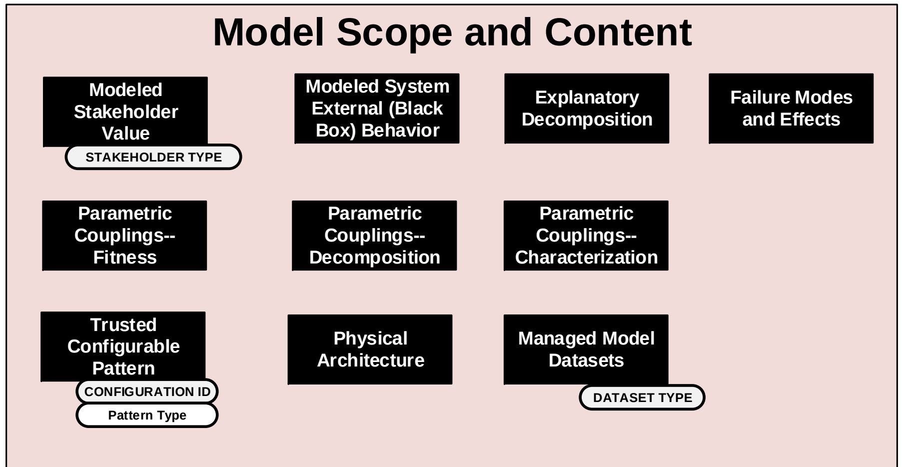




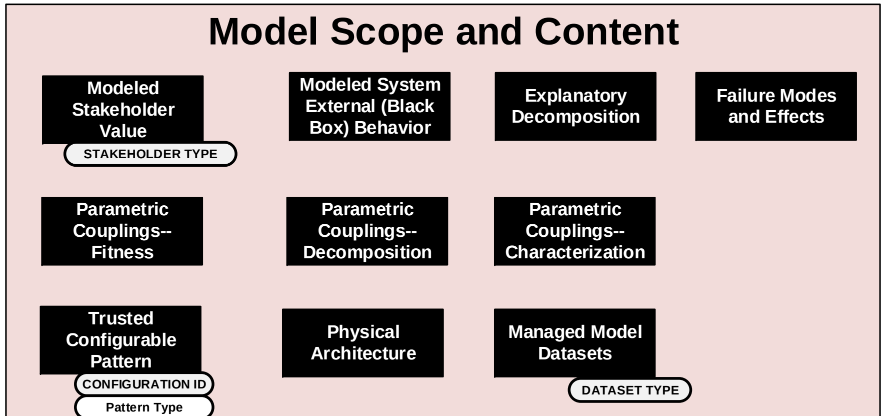
Feature Group	Feature Name	Feature Definition	Feature Attribute	Attribute Definition	Feature Stakeholder							Model Type	
					Model User	Model Developer	Model Maintainer	Mdl Deployer-Distributor	Model Use Supporter	Regulatory Authority	Mdl Investor-Owner	Physics Based	Data Driven
Identifies the main subject or focus of the model													
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	**X**					**X**	**X**	X	X
	Modeled Environmental Domain	Identifies the type of external environmental domain(s) that this model includes.	Domain Type(s)	Name(s) of modeled domains (manufacturing, distribution, use, etc.)	**X**					**X**	**X**	X	X



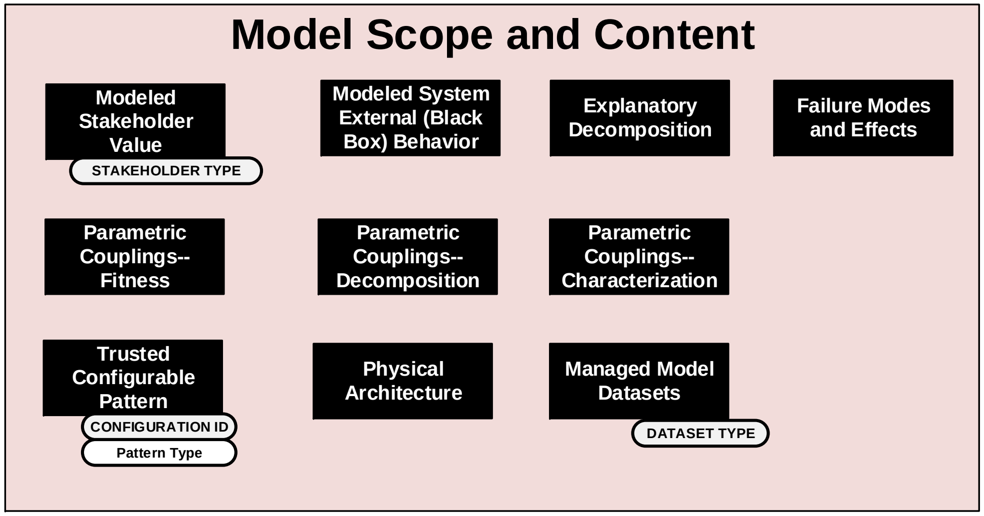
Feature Group	Feature Name	Feature Definition	Feature Attribute	Attribute Definition	Feature Stakeholder							Model Type	
					Model User	Model Developer	Model Maintainer	Mdl Deployer-Distributor	Model Use Supporter	Regulatory Authority	Mdl Investor-Owner	Physics Based	Data Driven
Describes the intended use, utility, and value of the model													
Model Utility	Model Intended Use	The intended purpose(s) or use(s) of the model.	Life Cycle Process Supported	The intended life cycle management process to be supported by the model, from the ISO15288 process list. More than one value may be listed.	X					X	X	X	X
	Perceived Model Value and Use	The relative level of value ascribed to the model, by those who use it for its stated purpose.	User Group Segment	The identify of using group segment (multiple)	X					X	X	X	X
			Level of Annual Use	The relative level of annual use by the segment	X					X	X	X	X
			Value Level	The value class associated with the model by that segment	X					X	X	X	X
	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	The identity (may be multiple) of regulators, agencies, customers, supply chains, accepting the model	X					X	X	X	X
	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	X					X		X	X



Feature Group	Feature Name	Feature Definition	Feature Attribute	Attribute Definition	Feature Stakeholder							Model Type	
					Model User	Model Developer	Model Maintainer	Mdl Deployer-Distributor	Model Use Supporter	Regulatory Authority	Mdl Investor-Owner	Physics Based	Data Driven
Describes the scope of content of the model													
Model Scope of 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	Classes of covered stakeholders (may be multiple)	X					X	X	X	X
	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.			X					X		X	X
	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.			X					X		X	
	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.			X					X		X	

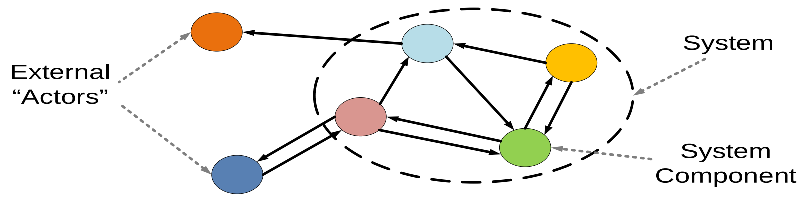


Feature Group	Feature Name	Feature Definition	Feature Attribute	Attribute Definition	Feature Stakeholder: Model User	Model Developer	Model Maintainer	Mdl Deployer-Distributor	Model Use Supporter	Regulatory Authority	Mdl Investor-Owner	Model Type: Physics Based	Data Driven
Describes the scope of content of the model													
	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.			X					X		X	X
	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.			X					X		X	X
	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).			X					X		X	
	Managed Model Datasets	The capability of the model to include managed datasets for use as inputs, parametric characterizations, or outputs	Dataset Type	The type(s) of data sets (may be multiple)	X		X			X		X	X
	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.	X		X			X	X	X	X
			Pattern ID	The identifier of the trusted configurable pattern.	X		X			X	X	X	X

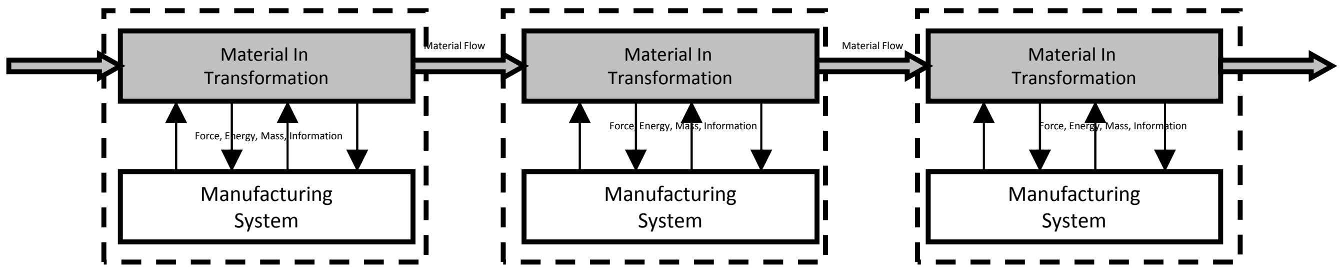


Feature Group	Feature Name	Feature Definition	Feature Attribute	Attribute Definition	Feature Stakeholder								Model Type	
					Model User	Model Developer	Model Maintainer	Mdl Deployer-Distributor	Model Use Supporter	Regulatory Authority	Mdl Investor-Owner		Physics Based	Data Driven
Describes the scope of content of the model														
Model Scope of 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.			X					X	X		X	

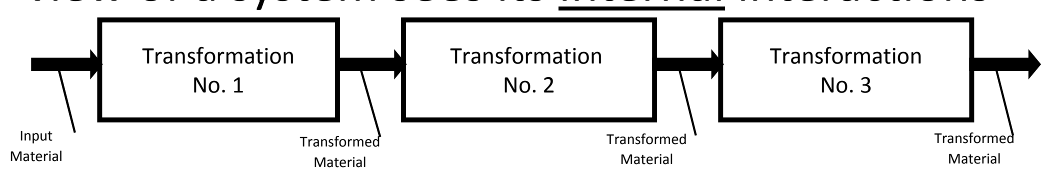
- A System is a set of interacting components:
  - By "interact", we mean exchanging energy, forces, mass flows, or information, resulting in changes of state:



  - So, a (Manufacturing or other) Process is a type of System (but not all Systems are such Processes):



- The "Black Box" view of a system sees only its <u>external</u> behavior
- The "White Box" view of a system sees its <u>internal</u> interactions

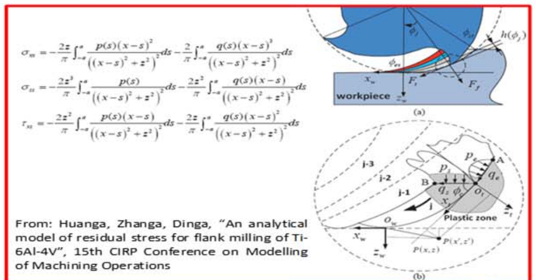


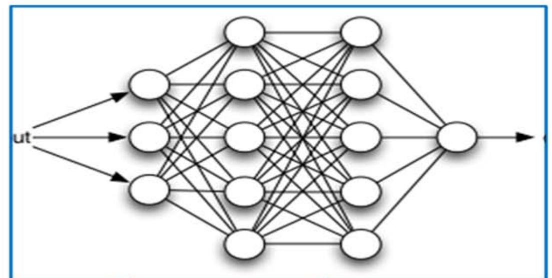
Physics-Based Model

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

Data Driven Model

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





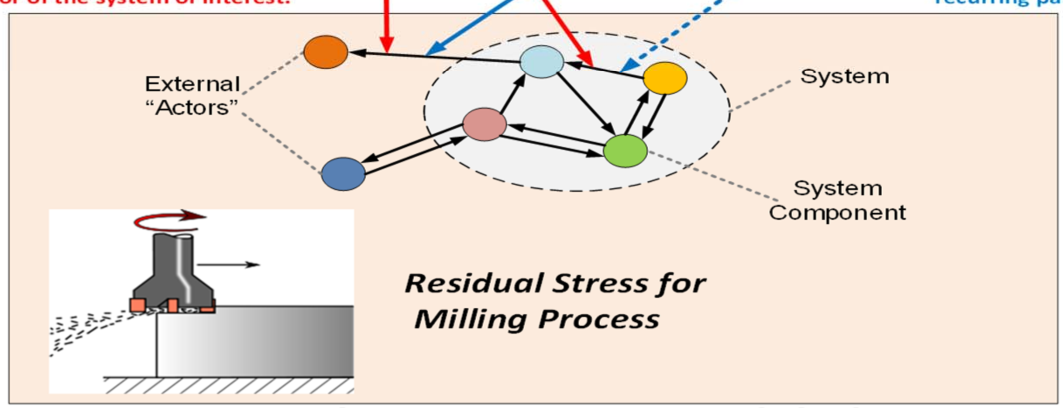
- **Physical scientists and phenomena models from their disciplines can apply here.**
- **The hard sciences physical laws, and how they can be used to explain the externally visible behavior of the system of interest.**

predicts, explains

predicts

optional

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



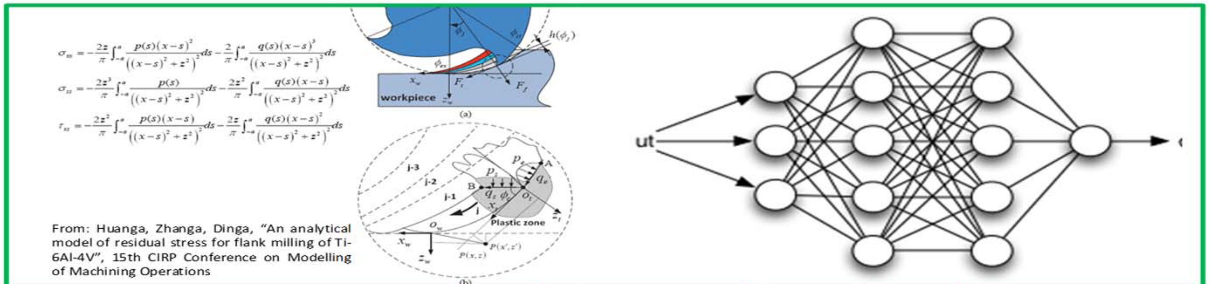
Real System Being Modeled

Hybrid Model: Both Data Driven and Physics-Based

- Predicts the external behavior of the System of Interest, visible externally to the external actors with which it interacts.

- Models (some aspects of) internal physical interactions of the System of Interest, and how they combine to cause/explain (some aspects of) externally visible behavior.
- Model has both external predictive value and (some) phenomena-based internal-to-external explanatory value.

- (Some) 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 (for some aspects) not internal explanatory value.



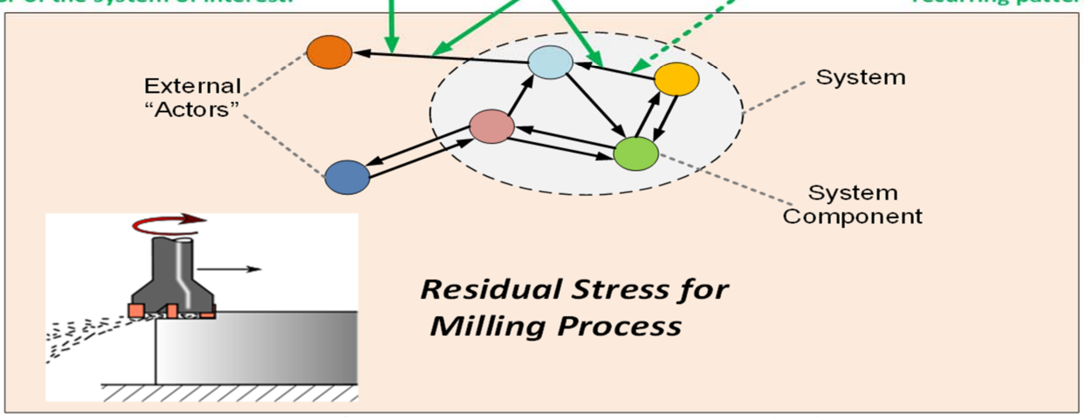
- **Physical scientists and phenomena models from their disciplines can apply here.**
- **The hard sciences physical laws, and how they can be used to explain the externally visible behavior of the system of interest.**

predicts, explains

predicts

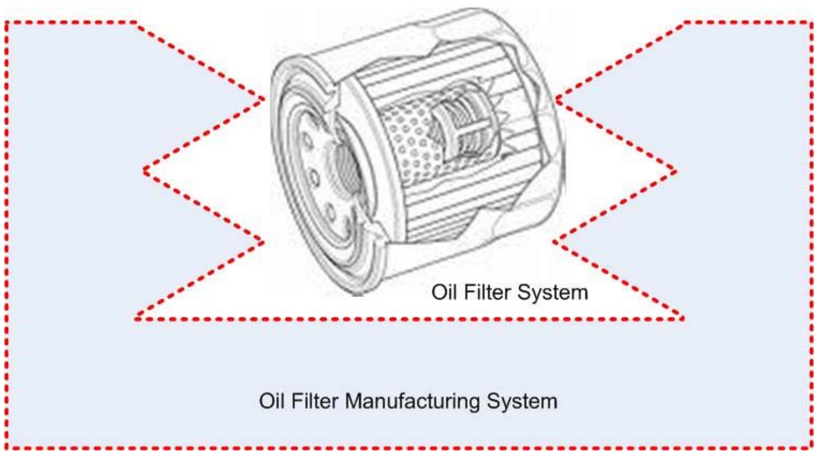
optional

- **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 System Being Modeled

Samples from a simple illustrative example

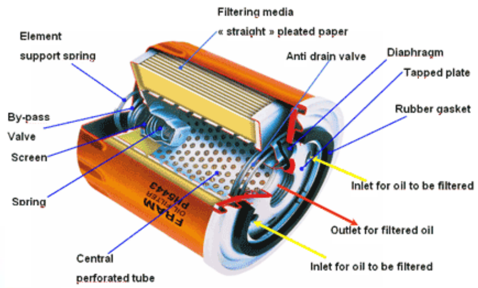


- Product: Oil Filter
- Manufacturing System: Oil Filter Mfg System

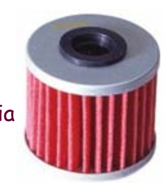
Physical Architecture Models describes the physical portion of the technology, to which Functional Roles will later be allocated and optimized . . .

Product Physical Architecture

Architecture 1: Laminated and Accordion Pleated Filtration Media, Flow Orthogonal to Plane of Media, Additive Impregnated



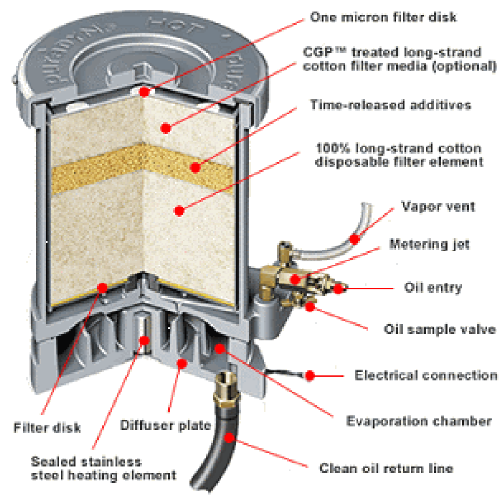
Paper Filter Media



Synthetic Filter Media

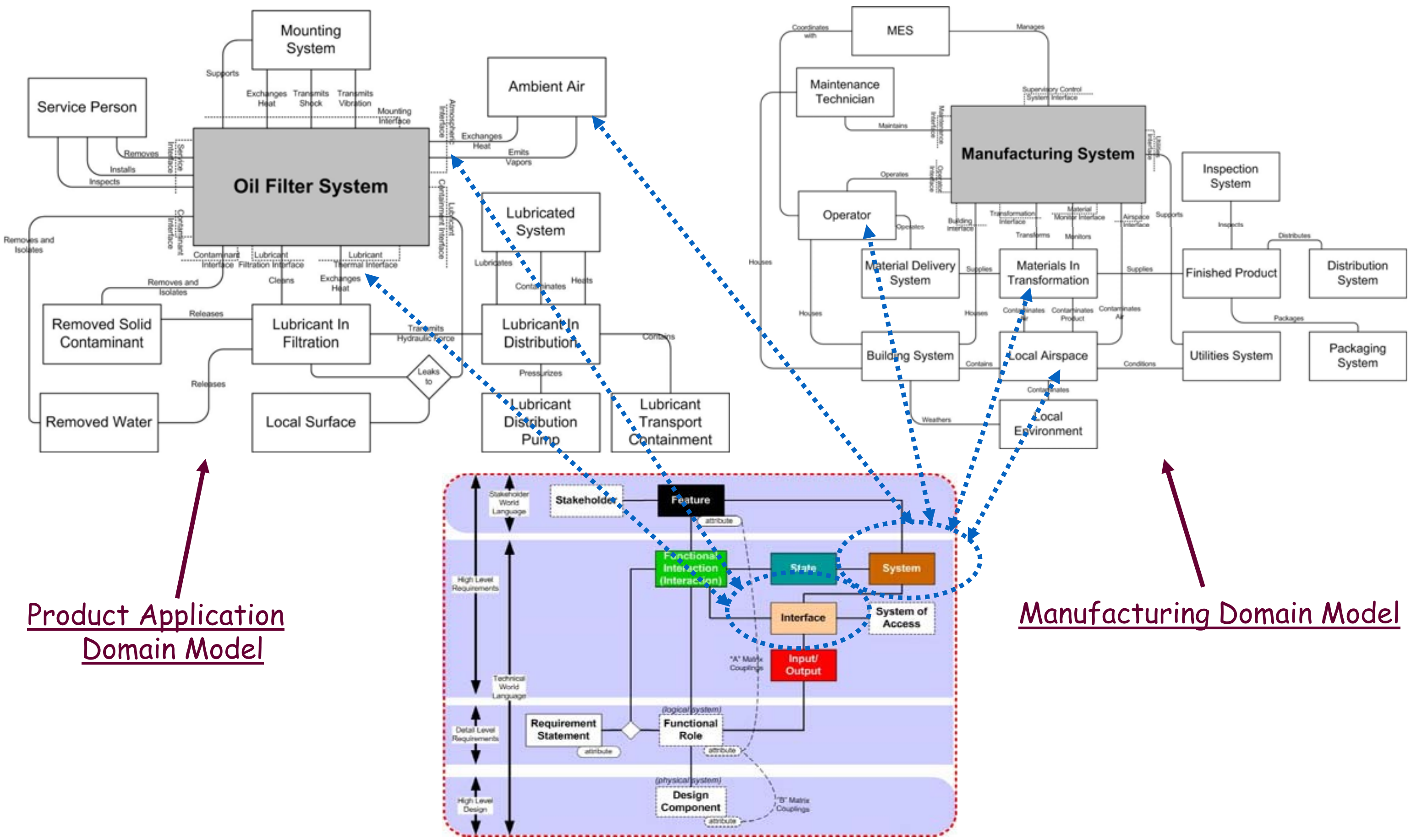
Stainless Steel Filter Media

Architecture 2: Wound Filtration Fiber, Flow Orthogonal to Plane of Windings, Additive Impregnated



Domain Models directly help by discovering and capturing all the external systems physically interacting with the Subject System—these are the source of all Functional Requirements.

Domain Models

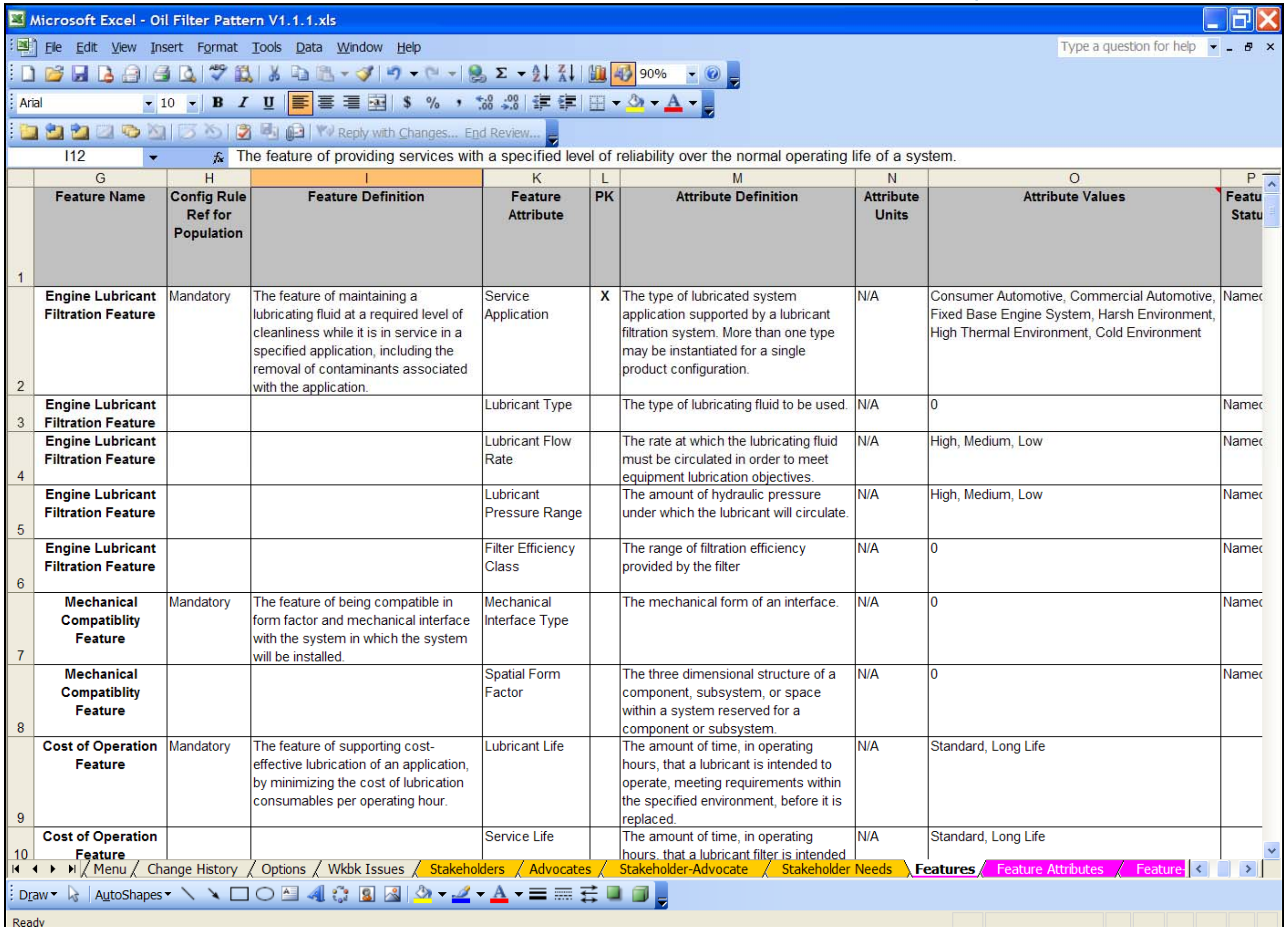


Product Application Domain Model

Manufacturing Domain Model

Stakeholder Feature Models address a key SE challenge by making explicit the ultimate stakeholder outcomes against which <u>all decisions, trade-offs, optimizations, and outcomes</u> will be <u>scored and selected</u>. This covers <u>all</u> Stakeholders, not just Customers (e.g., Shareholders, Community, etc.)

Product Stakeholder Features, Feature Attributes



Microsoft Excel - Oil Filter Pattern V1.1.1.xls

I12 — The feature of providing services with a specified level of reliability over the normal operating life of a system.

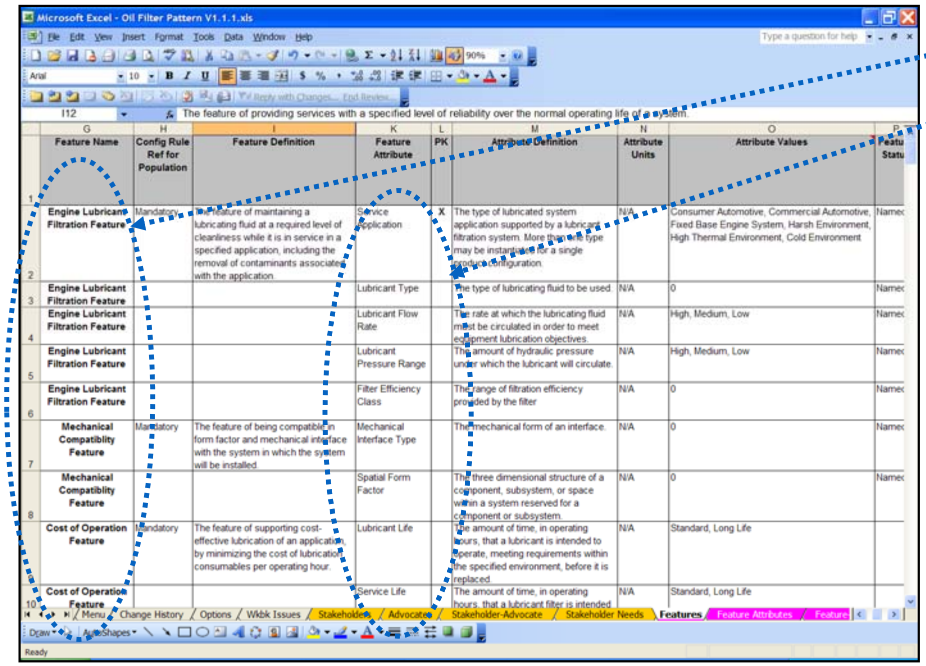
	G	H	I	K	L	M	N	O	P
1	Feature Name	Config Rule Ref for Population	Feature Definition	Feature Attribute	PK	Attribute Definition	Attribute Units	Attribute Values	Featu Statu
2	Engine Lubricant Filtration Feature	Mandatory	The feature of maintaining a lubricating fluid at a required level of cleanliness while it is in service in a specified application, including the removal of contaminants associated with the application.	Service Application	X	The type of lubricated system application supported by a lubricant filtration system. More than one type may be instantiated for a single product configuration.	N/A	Consumer Automotive, Commercial Automotive, Fixed Base Engine System, Harsh Environment, High Thermal Environment, Cold Environment	Namec
3	Engine Lubricant Filtration Feature			Lubricant Type		The type of lubricating fluid to be used.	N/A	0	Namec
4	Engine Lubricant Filtration Feature			Lubricant Flow Rate		The rate at which the lubricating fluid must be circulated in order to meet equipment lubrication objectives.	N/A	High, Medium, Low	Namec
5	Engine Lubricant Filtration Feature			Lubricant Pressure Range		The amount of hydraulic pressure under which the lubricant will circulate.	N/A	High, Medium, Low	Namec
6	Engine Lubricant Filtration Feature			Filter Efficiency Class		The range of filtration efficiency provided by the filter	N/A	0	Namec
7	Mechanical Compatiblity Feature	Mandatory	The feature of being compatible in form factor and mechanical interface with the system in which the system will be installed.	Mechanical Interface Type		The mechanical form of an interface.	N/A	0	Namec
8	Mechanical Compatiblity Feature			Spatial Form Factor		The three dimensional structure of a component, subsystem, or space within a system reserved for a component or subsystem.	N/A	0	Namec
9	Cost of Operation Feature	Mandatory	The feature of supporting cost-effective lubrication of an application, by minimizing the cost of lubrication consumables per operating hour.	Lubricant Life		The amount of time, in operating hours, that a lubricant is intended to operate, meeting requirements within the specified environment, before it is replaced.	N/A	Standard, Long Life	
10	Cost of Operation Feature			Service Life		The amount of time, in operating hours, that a lubricant filter is intended	N/A	Standard, Long Life	

Menu / Change History / Options / Wkbk Issues / Stakeholders / Advocates / Stakeholder-Advocate / Stakeholder Needs / Features / Feature Attributes / Feature

Ready

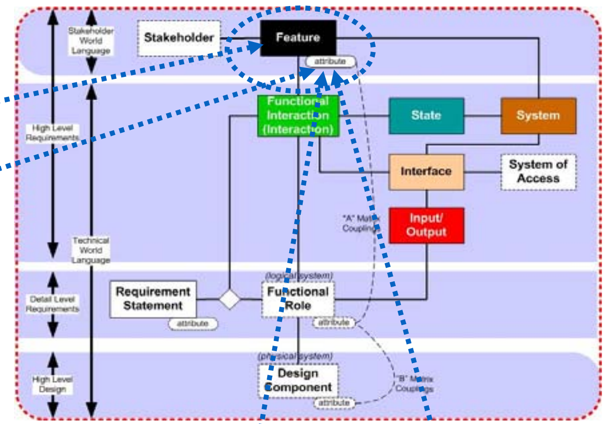
<u>Features</u> are collections of Functional Interactions (behaviors) having value to Stakeholders; their Attributes quantify that value impact. Features are in language of Stakeholders.

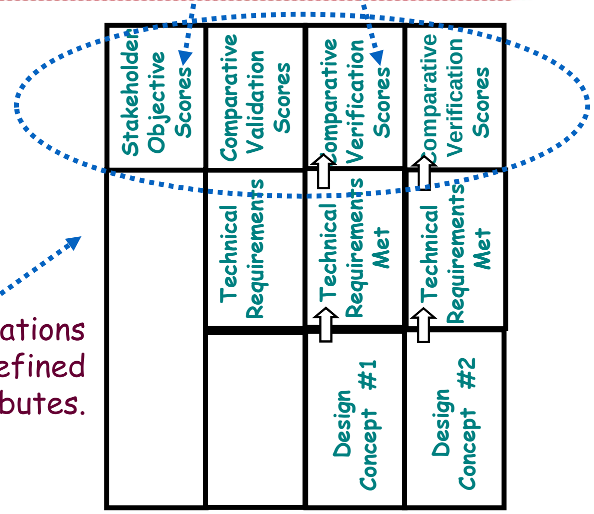
Product Stakeholder Features, Feature Attributes


Microsoft Excel - Oil Filter Pattern V1.1.1.xls

I12 The feature of providing services with a specified level of reliability over the normal operating life of a system.

	Feature Name	Config Rule Ref for Population	Feature Definition	Feature Attribute	PK	Attribute Definition	Attribute Units	Attribute Values	Featu Statu
2	Engine Lubricant Filtration Feature	Mandatory	The feature of maintaining a lubricating fluid at a required level of cleanliness while it is in service in a specified application, including the removal of contaminants associated with the application	Service Application	X	The type of lubricated system application supported by a lubricant filtration system. More than one type may be instantiated for a single product configuration.	N/A	Consumer Automotive, Commercial Automotive, Fixed Base Engine System, Harsh Environment, High Thermal Environment, Cold Environment	Namec
3	Engine Lubricant Filtration Feature			Lubricant Type		The type of lubricating fluid to be used.	N/A	0	Namec
4	Engine Lubricant Filtration Feature			Lubricant Flow Rate		The rate at which the lubricating fluid must be circulated in order to meet equipment lubrication objectives.	N/A	High, Medium, Low	Namec
5	Engine Lubricant Filtration Feature			Lubricant Pressure Range		The amount of hydraulic pressure under which the lubricant will circulate.	N/A	High, Medium, Low	Namec
6	Engine Lubricant Filtration Feature			Filter Efficiency Class		The range of filtration efficiency provided by the filter	N/A	0	Namec
7	Mechanical Compatiblity Feature	Mandatory	The feature of being compatible in form factor and mechanical interface with the system in which the system will be installed.	Mechanical Interface Type		The mechanical form of an interface.	N/A	0	Namec
8	Mechanical Compatiblity Feature			Spatial Form Factor		The three dimensional structure of a component, subsystem, or space within a system reserved for a component or subsystem.	N/A	0	Namec
9	Cost of Operation Feature	Mandatory	The feature of supporting cost-effective lubrication of an application, by minimizing the cost of lubrication consumables per operating hour.	Lubricant Life		The amount of time, in operating hours, that a lubricant is intended to operate, meeting requirements within the specified environment, before it is replaced.	N/A	Standard, Long Life	
10	Cost of Operation Feature			Service Life		The amount of time, in operating hours, that a lubricant filter is intended	N/A	Standard, Long Life	





<u>Configuration Score Sheet</u>

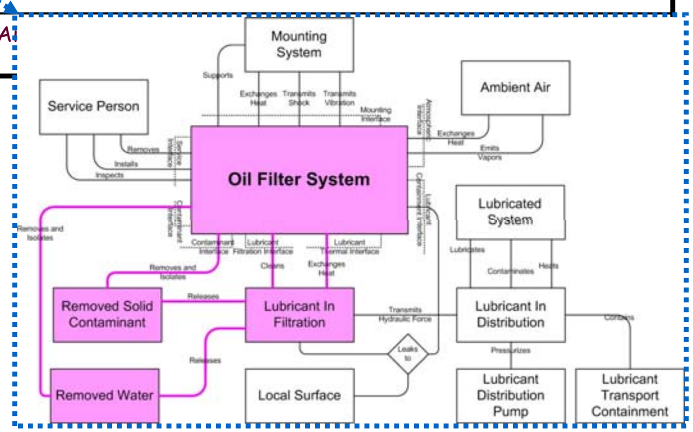
Alternate designs, different configurations, and technology generations are all ultimately "Scored" in lower-dimension trade-off space defined by the Stakeholder Feature Attributes.

<u>For example</u>: Every FMEA (Failure Mode Effects Analysis) failure impact can be expressed in terms of Feature Attributes.

Functional Interaction Models a key SE challenge by discovering and describing all external interactions of a Subject System. This leads to <u>all functional requirements</u> and thereafter all other requirements, in the Detail Requirements Model.

Product Functional Interactions, Roles

Functional Interaction	Functional Roles
Filter Lubricant	Lubricant in Filtration, Oil Filter System, Removed Solid Contaminant, Removed Water
Install Filter	Service Person, Filter
Monitor Filter	Filter, Monitor & Control System
Prevent Vapor Leakage	Lubricant, Vapor, Filter, Atmosphere
Prevent Lubricant Leakage	Lubricant, Filter, Local Surface
Transmit Shock & Vibration	Filter, Mounting System
Transmit Thermal Energy	Filter, Lubricant, Mounting System, Ambient A

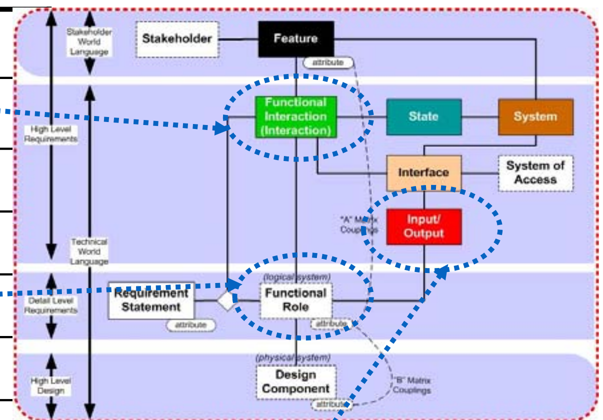


Every system <u>directly</u> interacting with the Subject System (Oil Filter System) contributes to its Requirements.

An Interaction of Systems, expressed as an external (outcome) relationship in which systems impact each other's states. Interacting systems fill Roles in the Interaction. Interactions technically characterize (model) the behaviors summarized by stakeholder-valued Features.

Product Functional Interactions, Roles

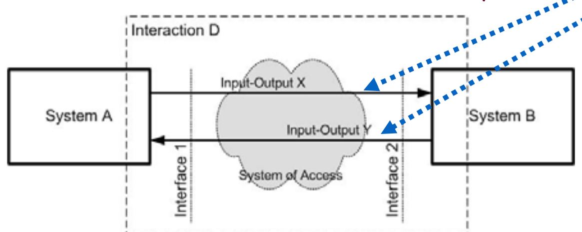
Functional Interaction	Functional Roles
Filter Lubricant	Lubricant in Filtration, Oil Filter System, Removed Solid Contaminant, Removed Water
Change Filter	Service Person, Filter
Monitor Filter	Filter, Monitor & Control System
Prevent Vapor Leakage	Lubricant, Vapor, Filter, Atmosphere
Prevent Lubricant Leakage	Lubricant, Filter, Local Surface
Transmit Shock & Vibration	Filter, Mounting System
Transmit Thermal Energy	Filter, Lubricant, Mounting System, Ambient Air



Input/Outputs exchanged during these interactions are:

- Energy
- Force
- Mass
- Information

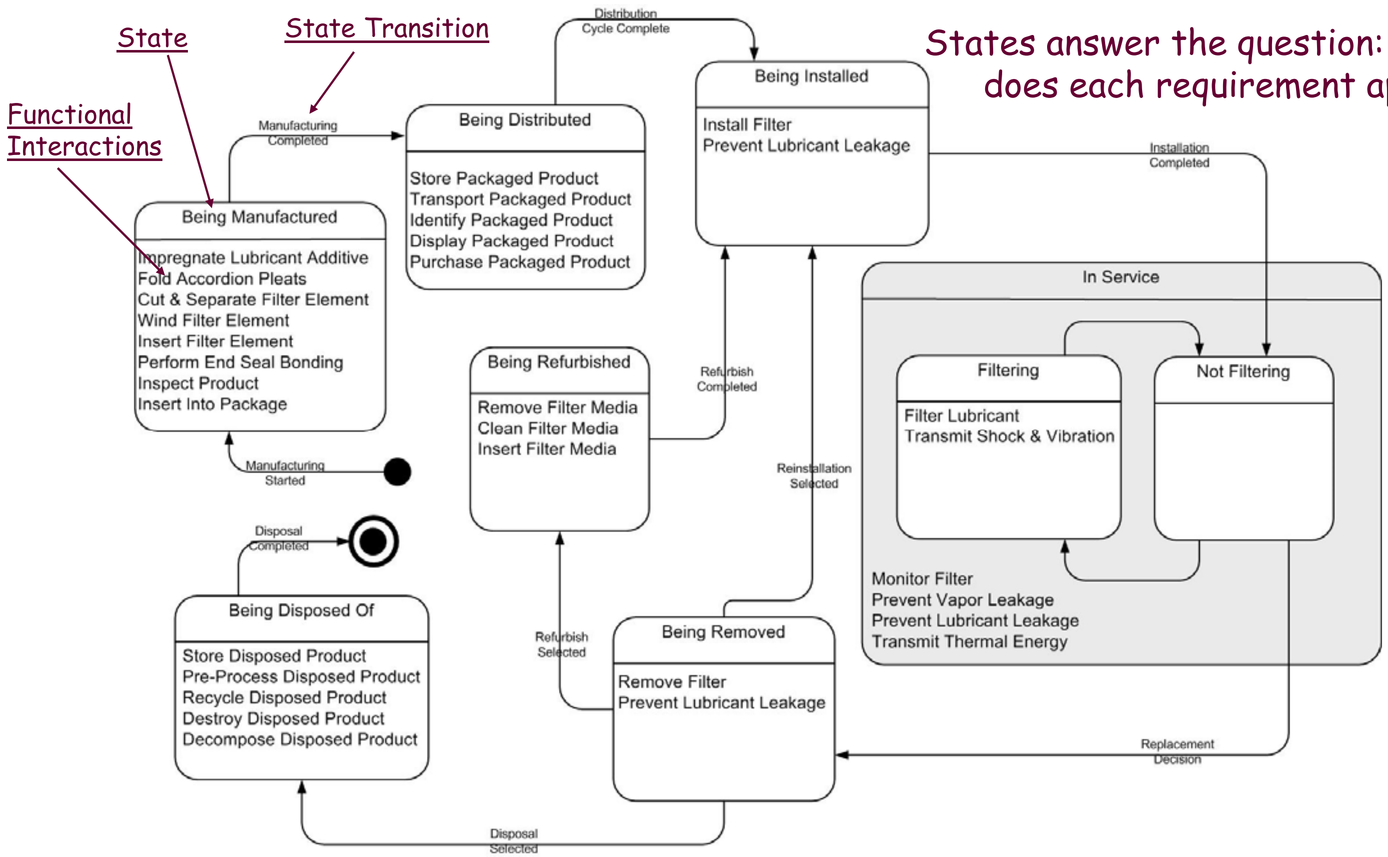
Interactions involve two or more systems.



State Models directly address a key SE challenge by discovering and describing all Situations, Modes, or Use Cases (environmental states) that a Subject System will encounter. These are associated with Functional Interactions that lead directly to requirements. State Models can also describe Designs.

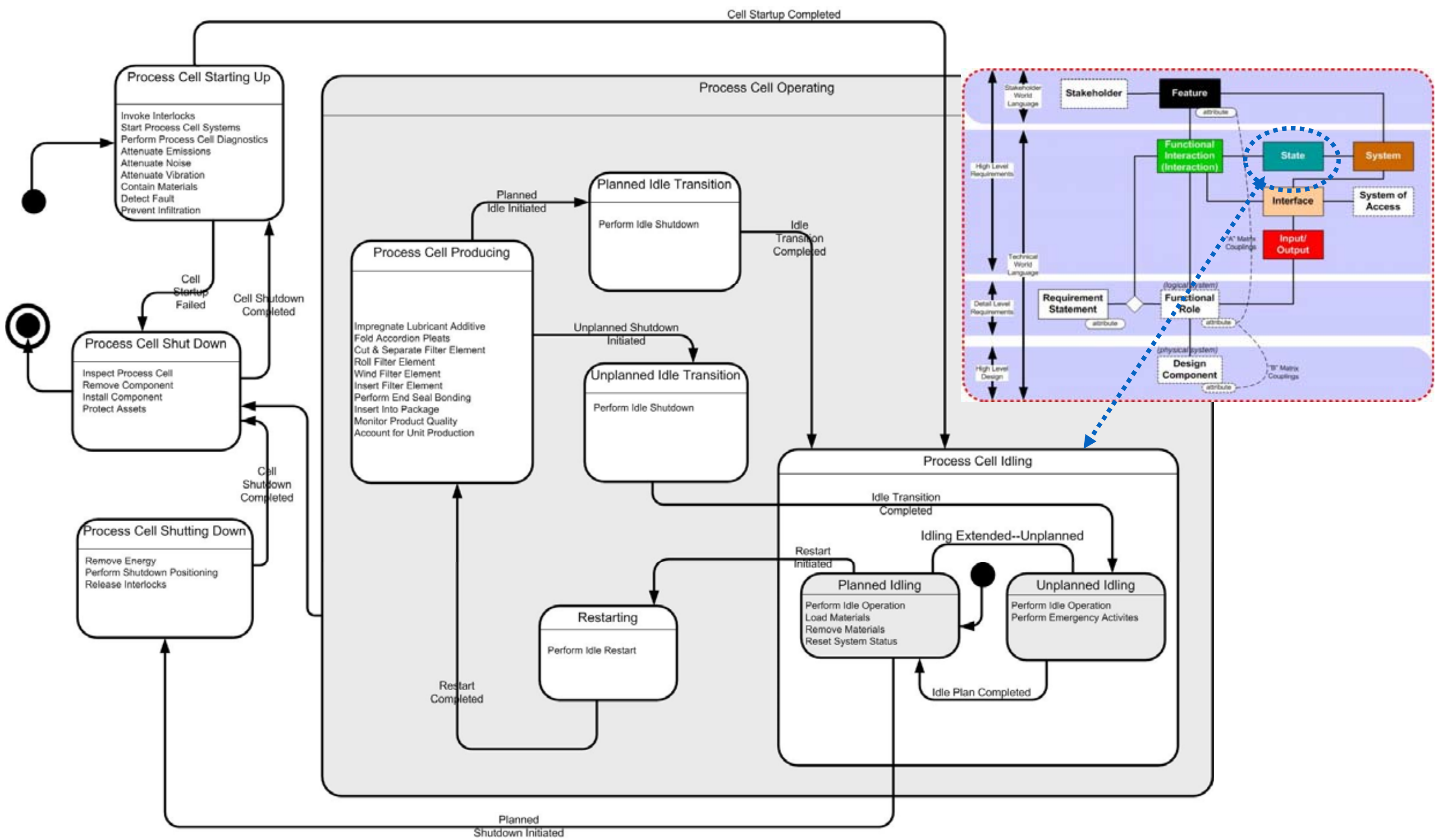
Product State Model

States answer the question: "When does each requirement apply?"



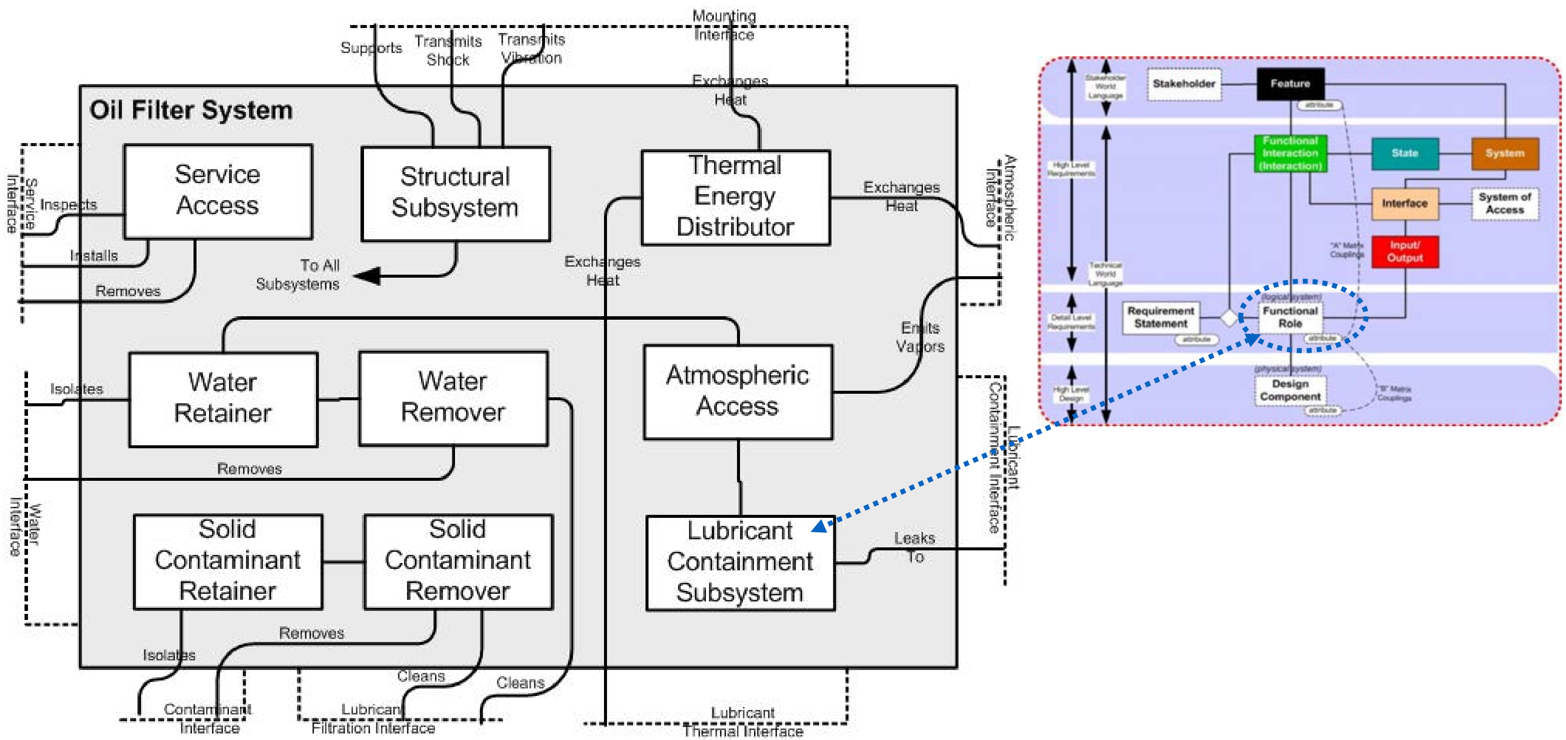
States are Situations (Modes, Use Cases, Phases) that will be encountered in the environment of a Subject System, in which it is required to meet certain requirements.

Manufacturing System State Model



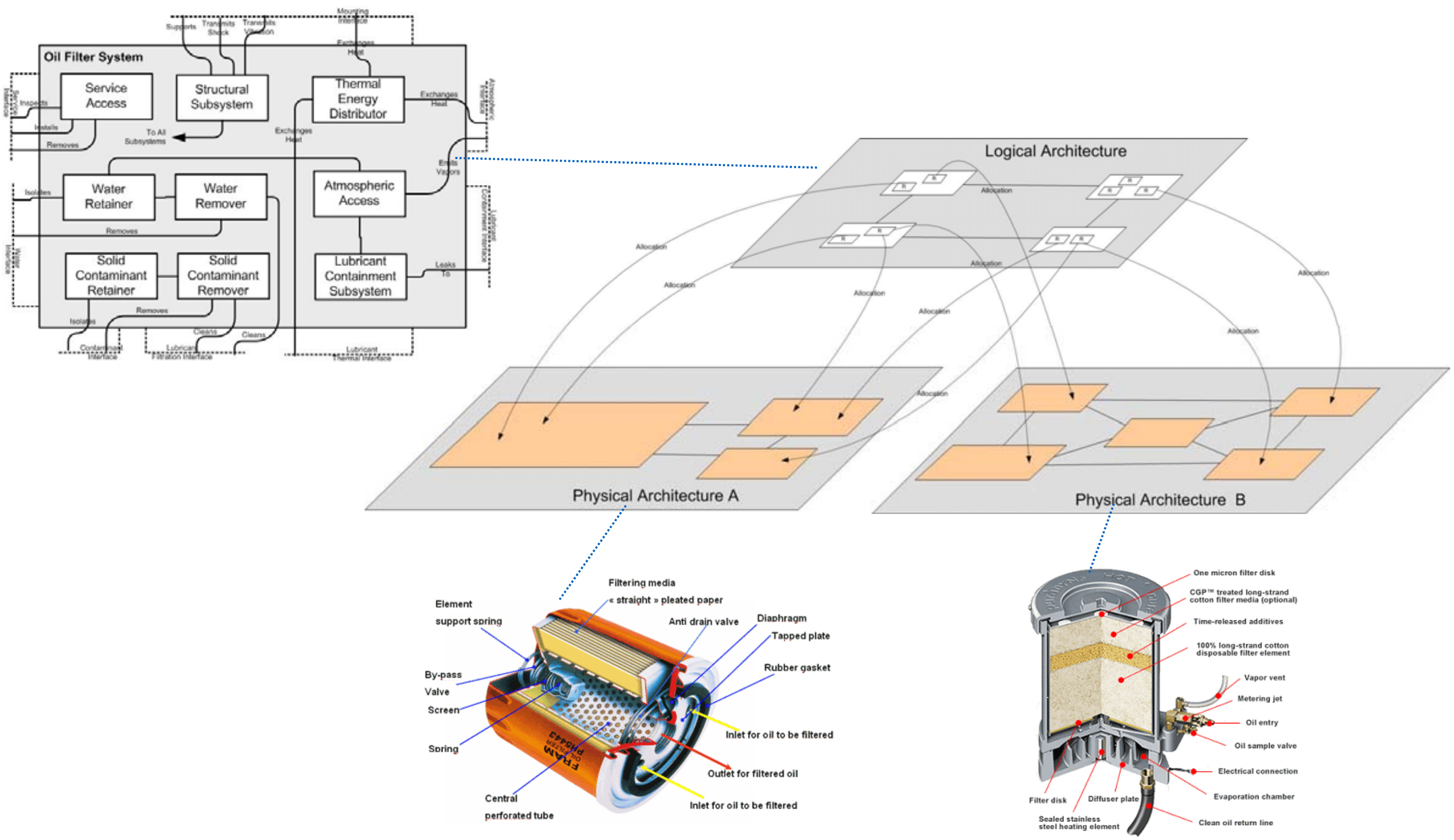
Logical Architecture Models directly address key SE challenges by partitioning the structure of requirements into Logical Roles independent of design, then address more SE challenges by stimulating design ideation and role allocation to physical designs and future technologies.

Product Logical Architecture Model



Directly addressing a key SE challenge, multiple alternate physical architectures are typically supported by a single Logical Architecture! This provides a powerful means for managing across Technologies & Configurations, and enhances Platform Management.

Alternate Technologies, Family Configurations, Roadmaps



What Is the Smallest Model of a System?

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

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 to attract a 10:1 larger global community of practitioners. And so, we ask: What is the smallest model of a system?

Introduction and Background

Representation Size, Purpose, Traditions. This paper
least) upper bounds on the sizes of effective representati
systems engineering. Compared to traditional systems en
directly on scientific traditions for representing behavi
engineering is still young, and its connections to supportir

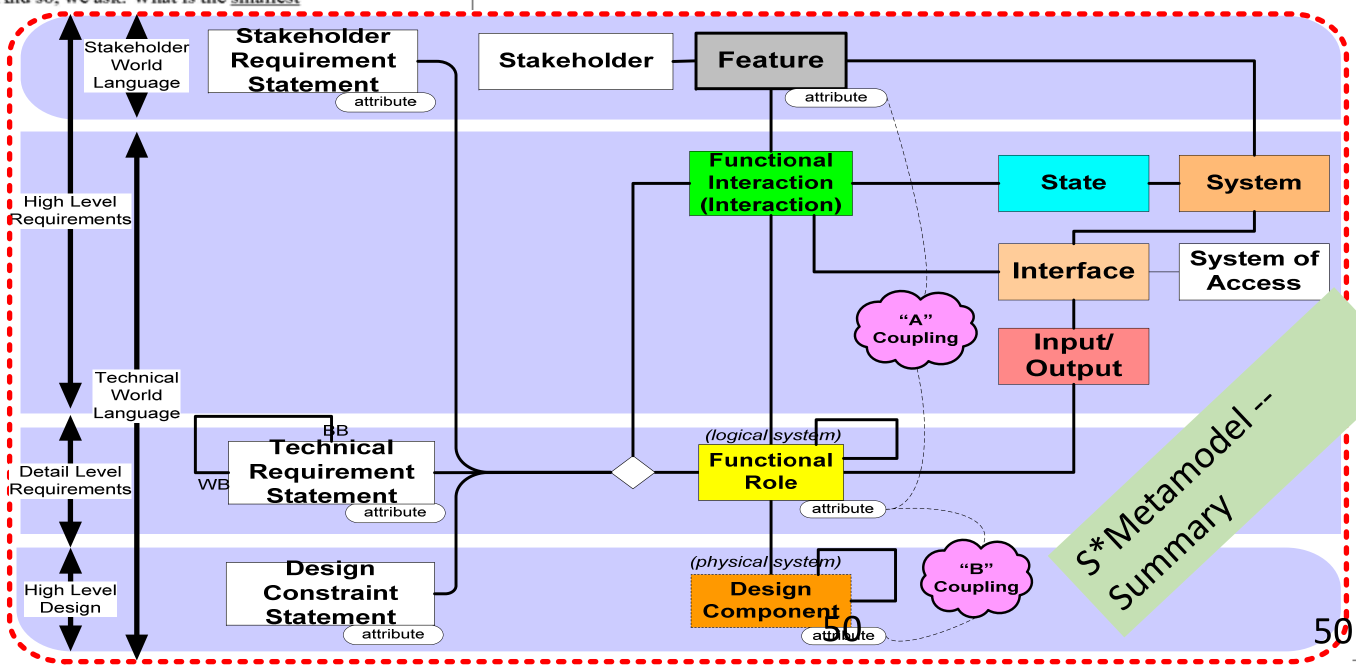
Language and Compression. This subject may appear t
describe systems, and an interesting thread in the mathen
whether minimality is in a sense independent of language
In any case, systems modeling languages such as Sys
valuable assets for the movement to model-based methods
is not the machinery of these specific modeling languages
models must address. When used for system familia
representation described here is subject to significant com
turns out to provide powerful insights about approaches
size of SE descriptions and processes, and about ongoing f
over time. These dynamics also suggest that such patterns
the interaction rules of the systems engineering process ar

Practical representation challenges of traditional syste
documentation of concept of operations (CONOP
specifications, failure mode and effects analysis (FM
maintenance procedures, and other task-specific system re
system can exceed thousands of pages. This does not e

INCOSE 2005 Symposium "Best Paper" Award in Modeling and Tools

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

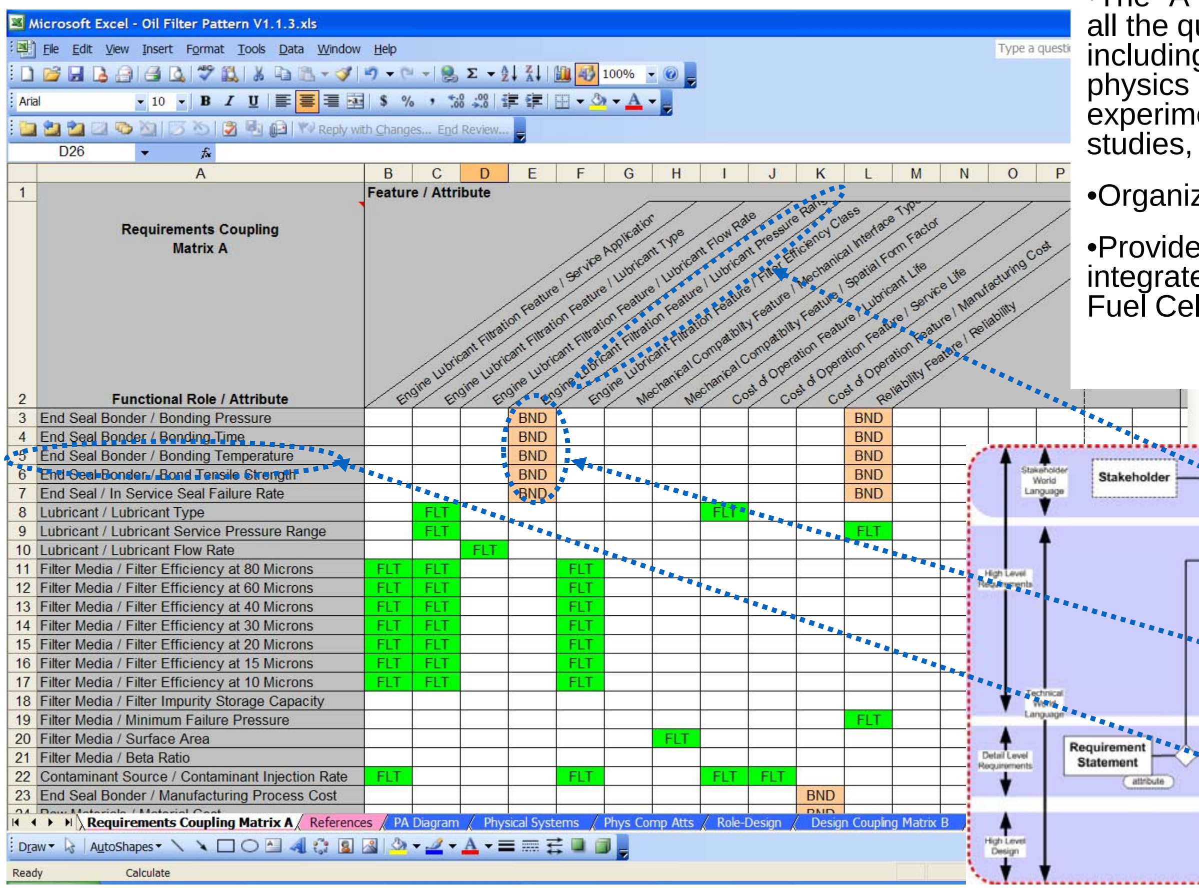
William D. Schindel
ICTT, Inc., and System Sciences, LLC
100 East Campus Drive, Terre Haute, IN 47802
812-232-2062 schindel@ictt.com

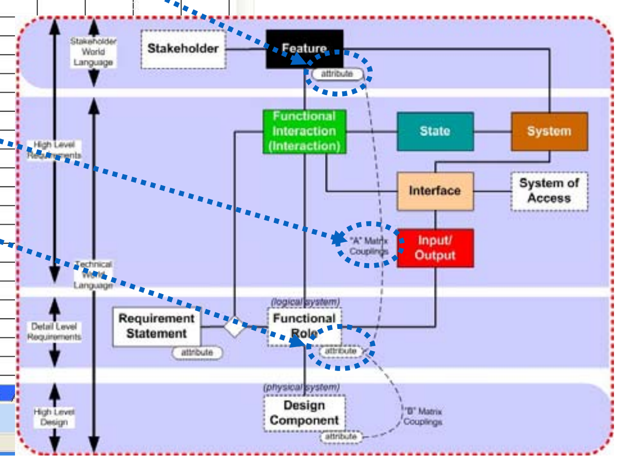


S*Metamodel -- Summary

The Attribute Coupling Model addresses a key SE challenge to understand the quantitative coupling of stakeholder preferences (Features) to technical requirements (Roles), establishing a Feature-based scoring space for trade-offs.

Attribute Coupling Model--Requirements

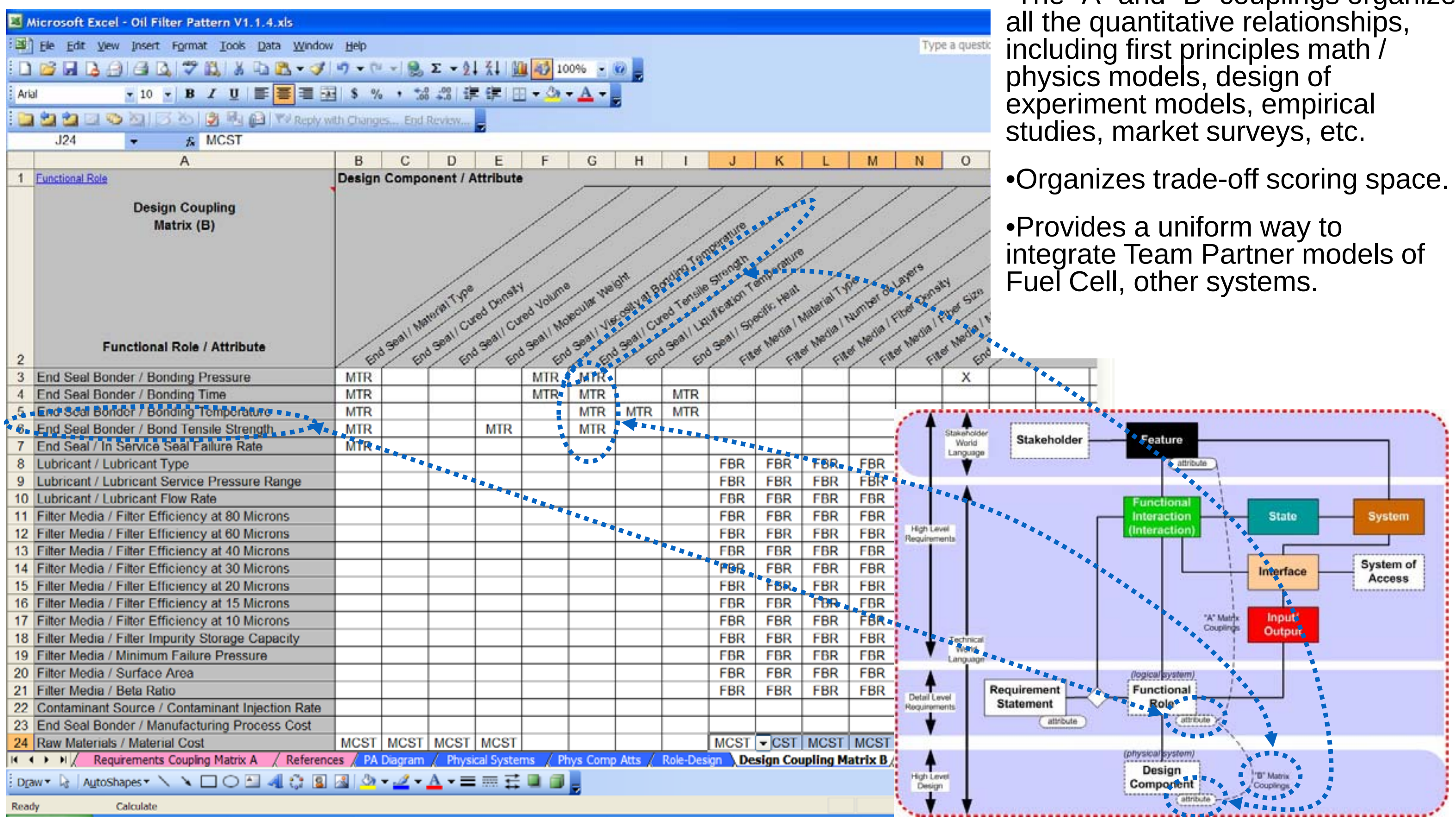




- The "A" and "B" couplings organize all the quantitative relationships, including first principles math / physics models, design of experiment models, empirical studies, market surveys, etc.
- Organizes trade-off scoring space.
- Provides a uniform way to integrate Team Partner models of Fuel Cell, other systems.

The Attribute Coupling Model addresses a key Challenge to describe the coupling of Design Component attributes to technical requirements (Role) attributes, provide scoring (in Feature Space) of Design Attribute solutions.

Attribute Coupling Model--Designs

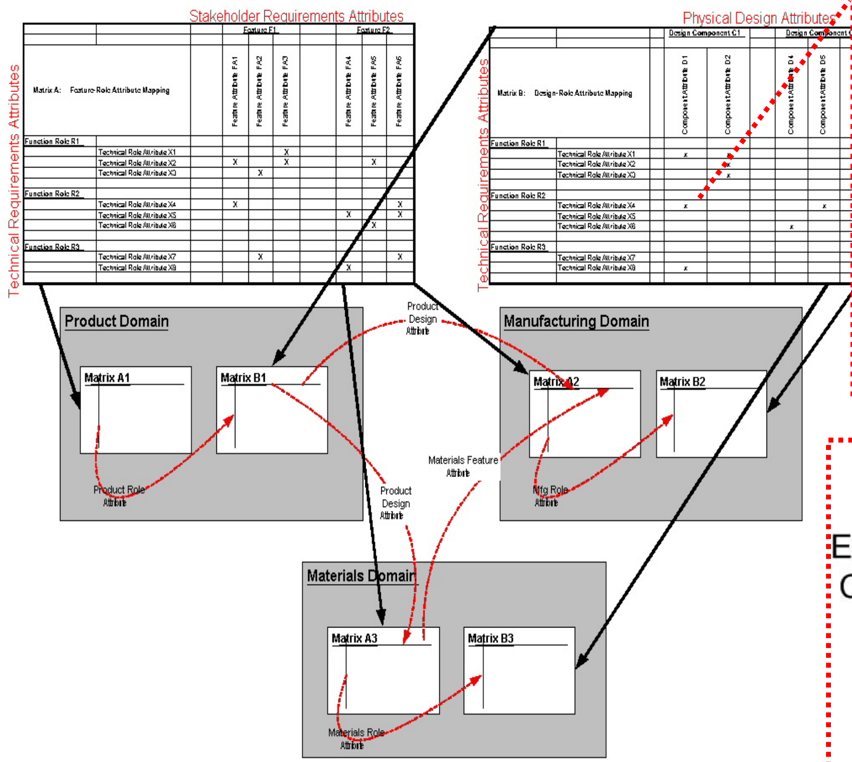


•The “A” and “B” couplings organize all the quantitative relationships, including first principles math / physics models, design of experiment models, empirical studies, market surveys, etc.

•Organizes trade-off scoring space.

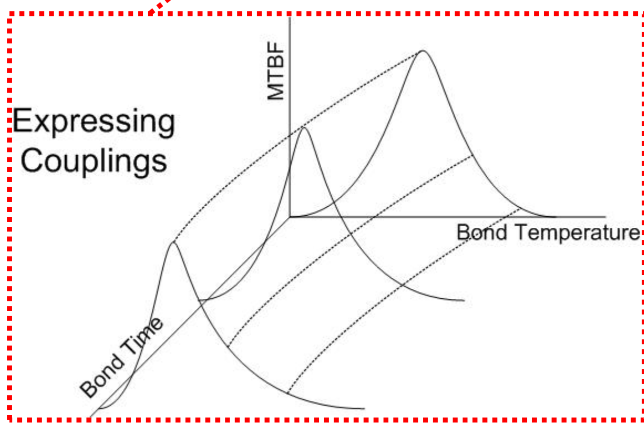
•Provides a uniform way to integrate Team Partner models of Fuel Cell, other systems.

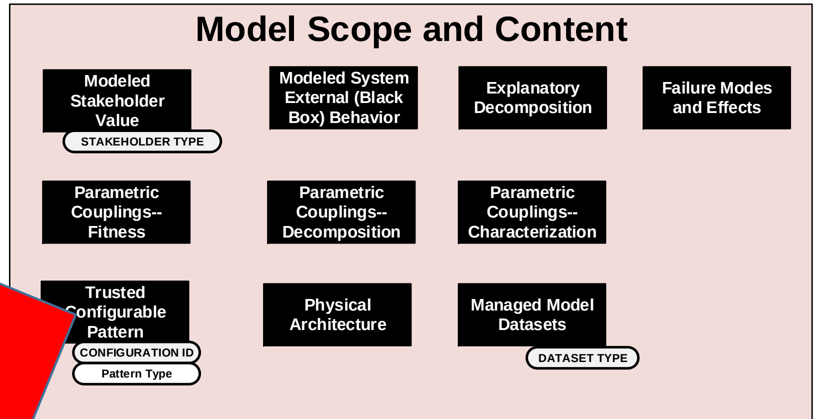
Attribute couplings cross domains



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

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





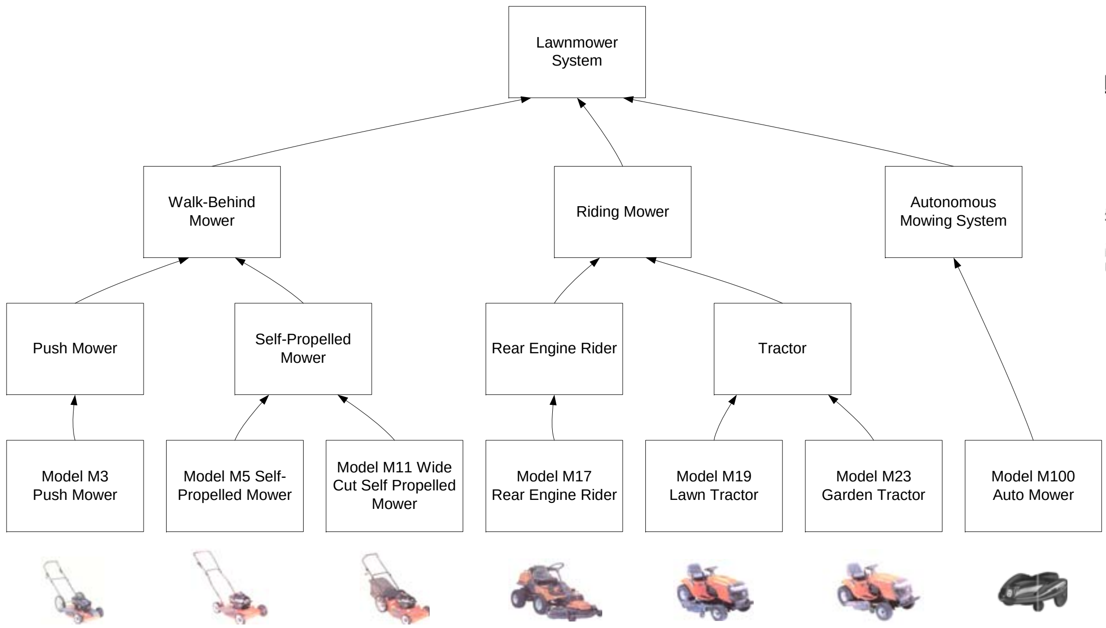
Of special importance to the economics of trust and VVUQ

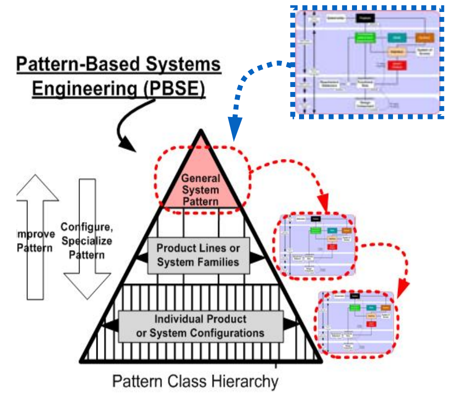
	[illegible]ame	Feature Definition	Feature Attribute	Attribute Definition	Feature Stakeholder							Model Type	
					Model User	Model Developer	Model Maintainer	Mdl Deployer-Distributor	Model Use Supporter	Regulatory Authority	Mdl Investor-Owner	Physics Based	Data Driven
Describes the scope of content of the model													
	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.			**X**					**X**		X	X
	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.			**X**					**X**		X	X
	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).			**X**					**X**		X	
	Managed Model Datasets	The capability of the model to include managed datasets for use as inputs, parametric characterizations, or outputs	Dataset Type	The type(s) of data sets (may be multiple)	**X**		**X**			**X**		X	X
	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.	**X**		**X**			**X**	**X**	X	X
			Pattern ID	The identifier of the trusted configurable pattern.	**X**		**X**			**X**	**X**	X	X

The Family Configurations Model directly addresses a key SE challenge by providing Class Hierarchy Models with Configuration Rules (Gestalt Rules) that govern Platforms and Portfolios of Products, Systems, and Technologies.

Family Configurations Model

- The Family Configurations Model supports multiple configurations, technologies:

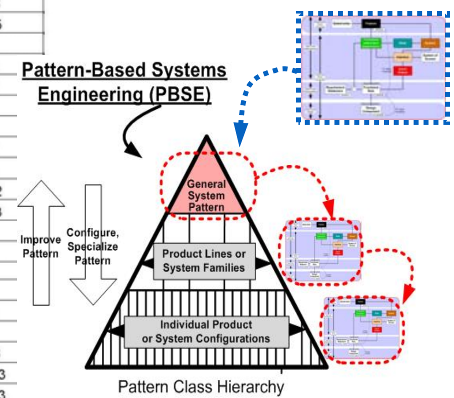




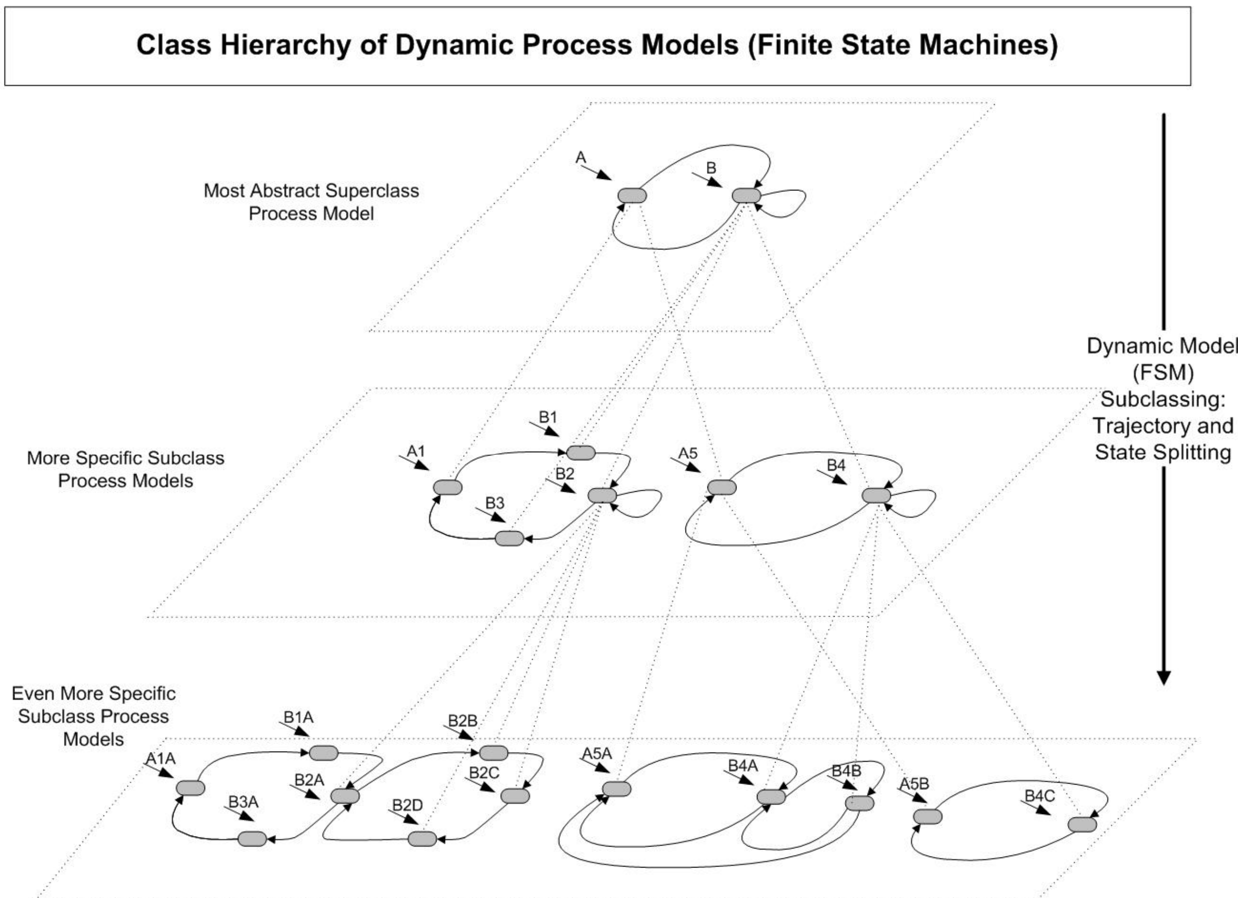
- This can be exploited by partitioning the model to integrate with existing Portfolio Roadmaps for Markets, Technologies, and Products

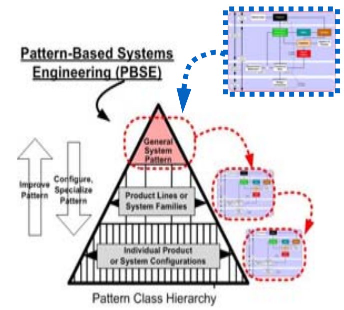
Family Configurations Model

Lawnmower Product Line: Configurations Table									
		Units							
			Walk-Behind Mower	Walk-Behind Mower	Walk-Behind Mower	Riding Mower	Riding Mower	Riding Mower	Autonomous Mowing System
			Push Mower	Self-Propelled Mower	Self-Propelled Mower	Rear Engine Rider	Tractor	Tractor	Autonomous Mowing System
			Push Mower	Self-Propelled	Wide Cut Self Propelled Mower	Rear Engine Rider	Lawn Tractor	Garden Tractor	Auto Mower
	Model Number		M3	M5	M11	M17	M19	M23	M100
	Market Segment		Small Residential	Medium Residential	Medium Residential	Large Residential	Large Residential	Home Garden	High End Suburban
Power	Engine Manufacturer		Briggs & Stratton	Briggs & Stratton	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	Number 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
	Fully Automatic Transmission Feature							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	

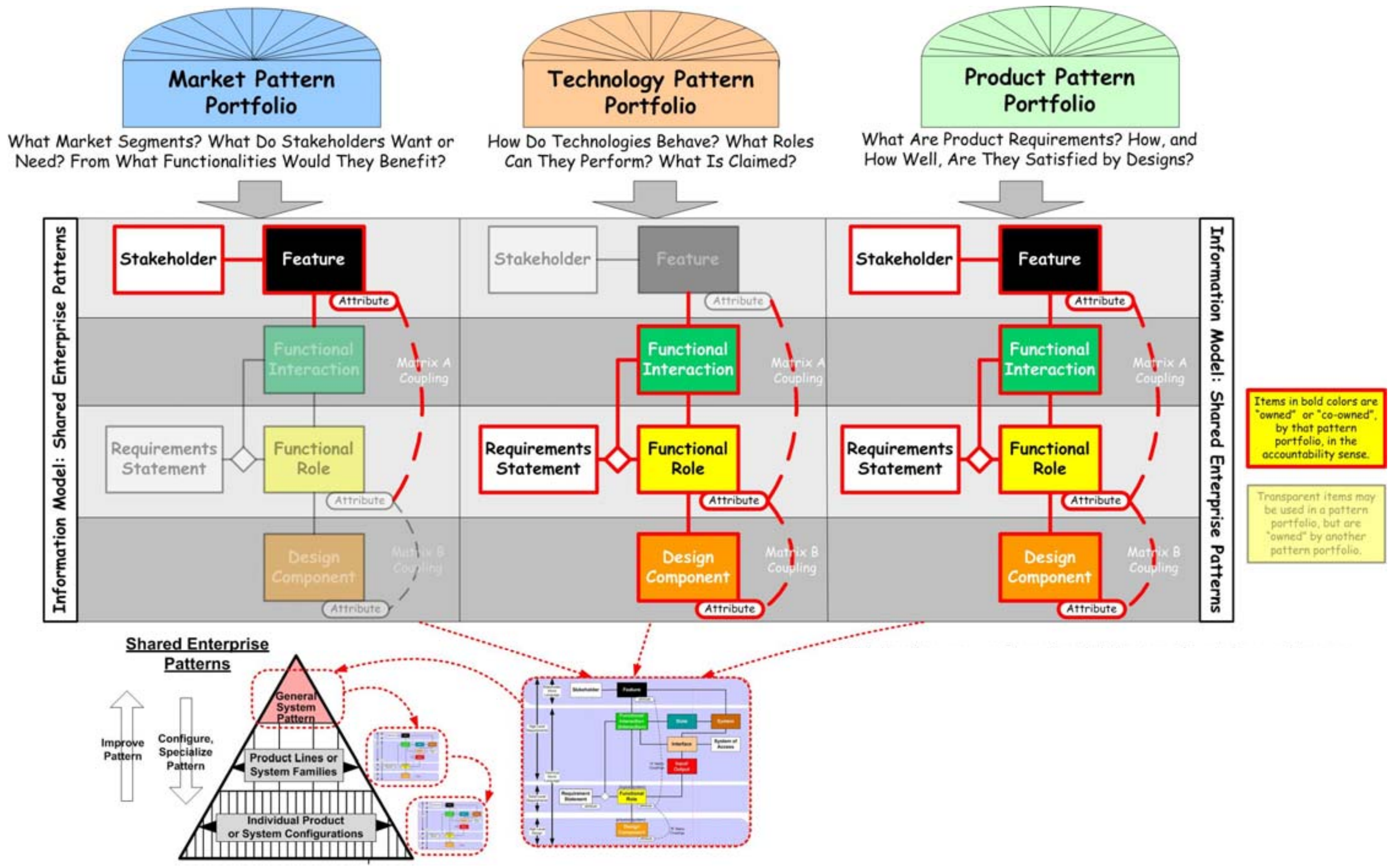


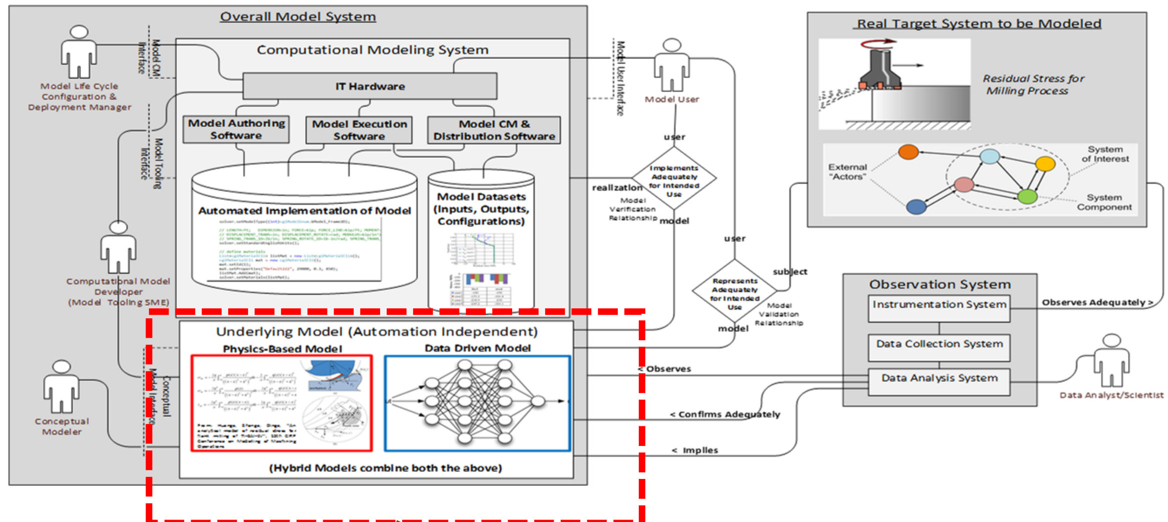
Family Configurations Model



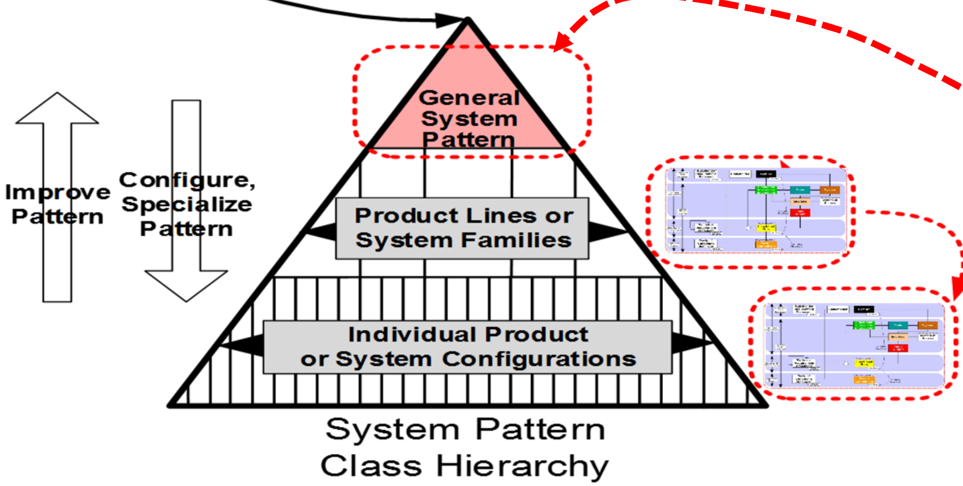


Family Configurations Model

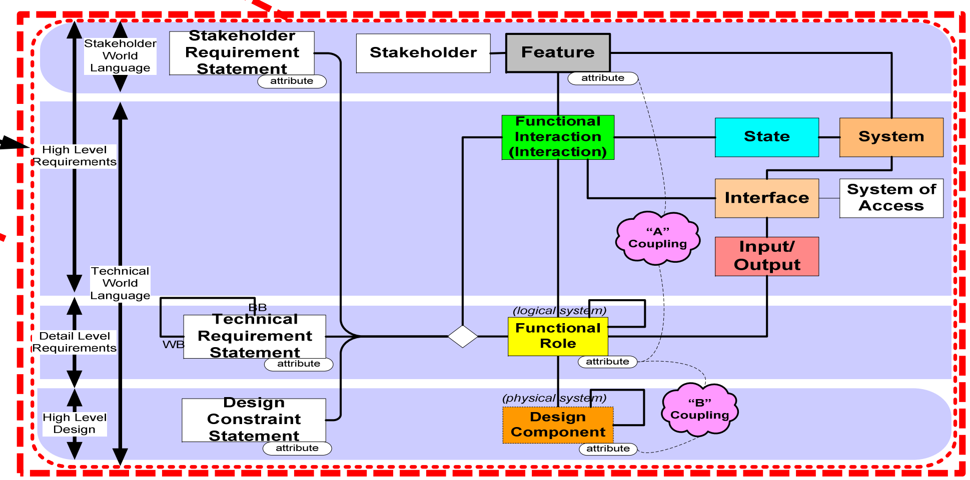




S*Pattern Hierarchy for Pattern-Based Systems Engineering (PBSE)



S*Metamodel for Model-Based Systems Engineering (MBSE)



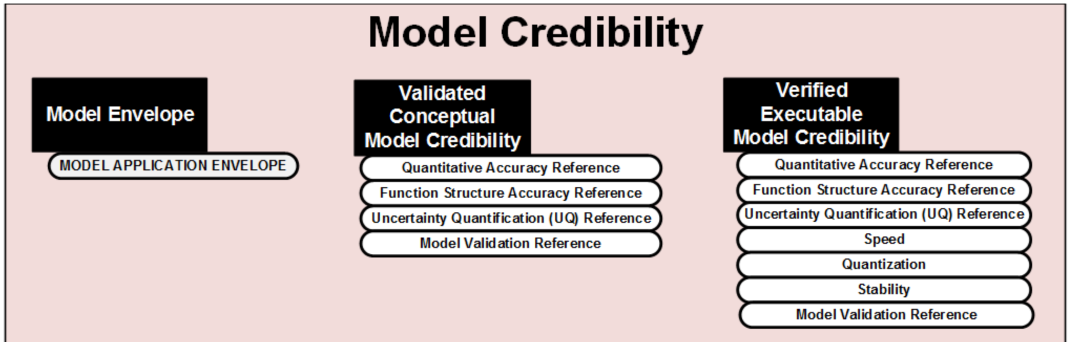
Model Credibility

Model Envelope
- MODEL APPLICATION ENVELOPE

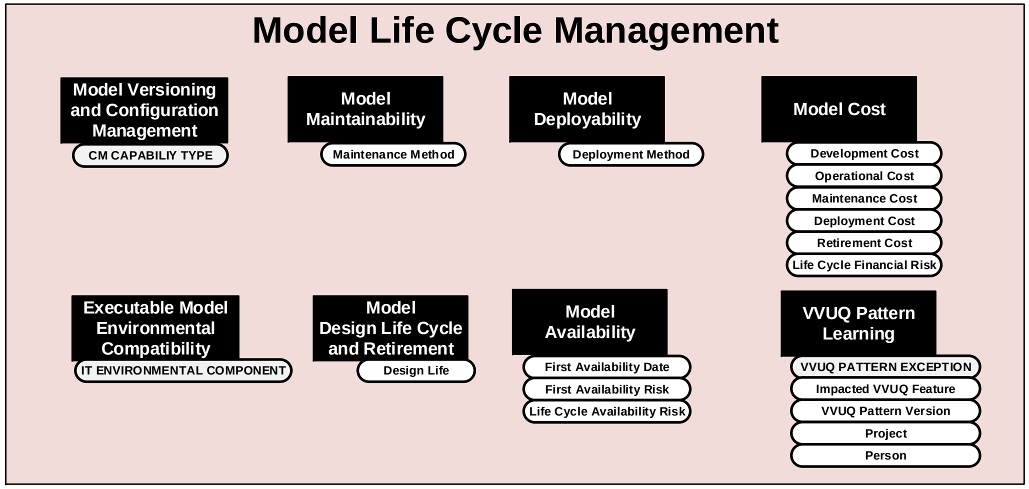
Validated Conceptual Model Credibility
- Quantitative Accuracy Reference
- Function Structure Accuracy Reference
- Uncertainty Quantification (UQ) Reference
- Model Validation Reference

Verified Executable Model Credibility
- Quantitative Accuracy Reference
- Function Structure Accuracy Reference
- Uncertainty Quantification (UQ) Reference
- Speed
- Quantization
- Stability
- Model Validation Reference

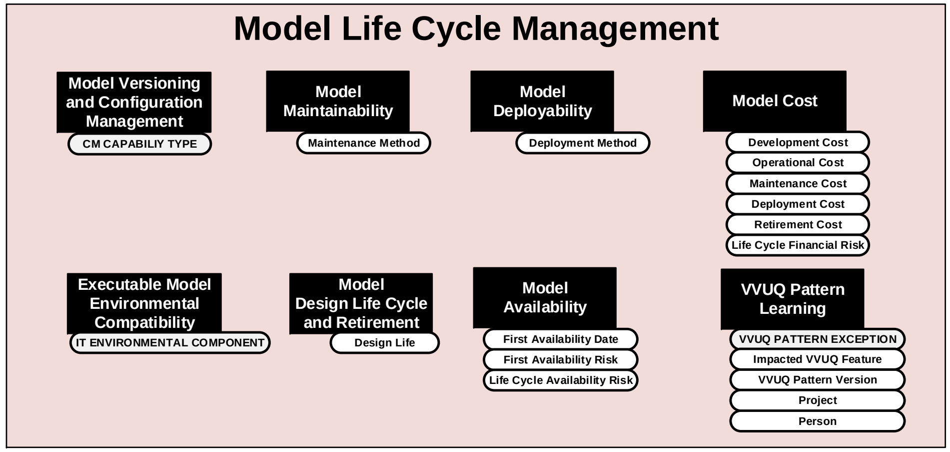
Feature Group	Feature Name	Feature Definition	Feature Attribute	Attribute Definition	Feature Stakeholder: Model User	Model Developer	Model Maintainer	Mdl Deployer -	Model Use Supporter	Regulatory Authority	Mdl Investor -	Model Type: Physics Based	Data Driven
Describes the credibility of the model													
	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.	X		X			X	X	X	X
	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.	X					X	X	X	X
			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.	X		X			X	X	X	X
			Uncertainty Quantification (UQ) Reference	The specification reference describing the degree of uncertainty of the Credibility of the conceptual model to the system of	X		X			X	X	X	X
			Model Validation Reference	The reference documenting the validation of the conceptual model's Credibility to the system of	X		X			X	X	X	X



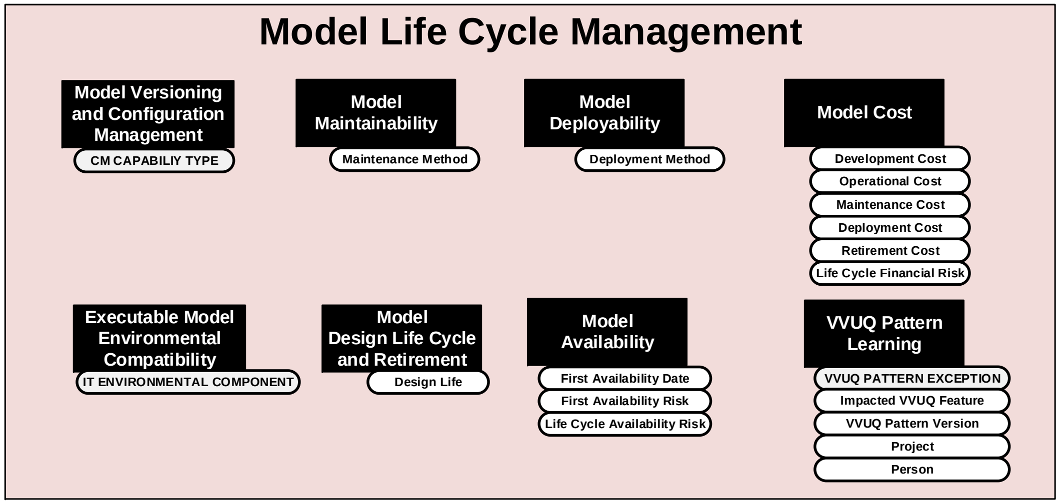
Feature Group	Feature Name	Feature Definition	Feature Attribute	Attribute Definition	Feature Stakeholder: Model User	Model Developer	Model Maintainer	Mdl Deployer-	Model Use Supporter	Regulatory Authority	Mdl Investor-	Model Type: Physics Based	Data Driven
Model Credibility	Verified Executable Model 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 executable model to the conceptual model.	X		X			X	X	X	X
			Structural Accuracy Reference	The specification reference describing the structural (presence or absence of elements) accuracy of the executable model to the conceptual model.	X		X			X	X	X	X
			Uncertainty Quantification (UQ) Reference	The specification reference describing the degree of uncertainty of the Credibility of the executable model to the conceptual model	X		X			X		X	X
			Speed	The specification reference describing the execution run time (speed) for the executable model.	X		X			X	X	X	X
			Quantization	The specification reference describing the quantization error of the executabl e model.	X		X			X	X	X	X
			Stability	The specification reference describing the level of stability of the accuracy and uncertainty of the executable model error characteristics.	X		X			X	X	X	X
			Model Validation Reference	The reference documenting the verification of the executable model's Credibility to the conceptual model.	X		X			X	X	X	X



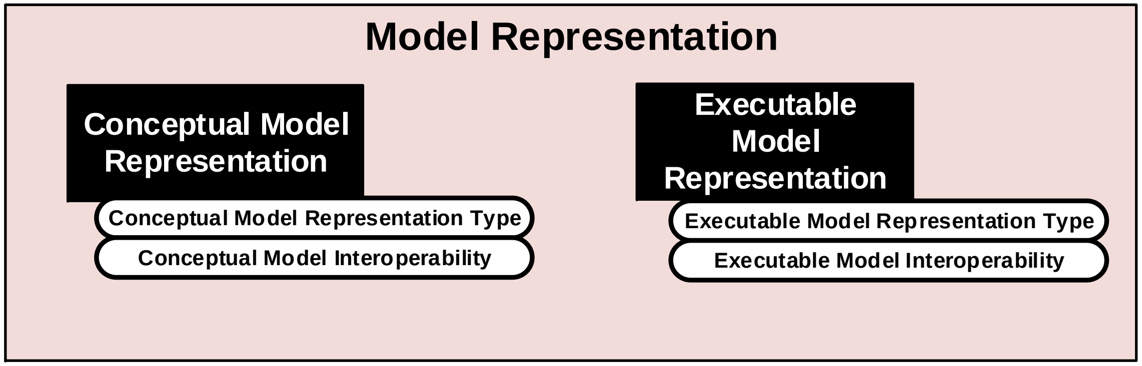
Feature Group	Feature Name	Feature Definition	Feature Attribute	Attribute Definition	Feature Stakeholder: Model User	Model Developer	Model Maintainer	Mdl Deployer-Distributor	Model Use Supporter	Regulatory Authority	Mdl Investor-Owner	Model Type: Physics Based	Data Driven
Describes related model life cycle management capabilities													
Model Life Cycle Management	Model Versioning and Configuration Management	The capability of the model to provide for version and configuration management.	CM Capability Type	The type(s) of CM capabilities included (may be multiple)	X		X			X		X	X
	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	The type(s) of IT environments or standards supported	X		X			X		X	X
	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	X		X			X		X	X
	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.	X		X			X	X	X	X
	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.	X			X			X	X	X



Feature Group	Feature Name	Feature Definition	Feature Attribute	Attribute Definition	Feature Stakeholder: Model User	Model Developer	Model Maintainer	Mdl Deployer-Distributor	Model Use Supporter	Regulatory Authority	Mdl Investor-Owner	Model Type: Physics Based	Data Driven
Describes related model life cycle management capabilities													
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		**X**					**X**	X	X
			Operational Cost	The cost to execute and otherwise operate the model, in standardized execution load units	**X**						**X**	X	X
			Maintenance Cost	The cost to maintain the model			**X**				**X**	X	X
			Deployment Cost	The cost to deploy, and redeploy updates, per cycle				**X**			**X**	X	X
			Retirement Cost	The cost to retire the model from service, in a planned fashion	**X**						**X**	X	X
			Life Cycle Financial Risk	Risk to the overall life cycle cost of the model							**X**	X	X
	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	**X**						**X**	X	X
			First Availability Risk	Risk to the scheduled date of first availability	**X**						**X**	X	X
			Life Cycle Availability Risk	Risk to ongoing availability after introduction	**X**						**X**	X	X



Feature Group	Feature Name	Feature Definition	Feature Attribute	Attribute Definition	Feature Stakeholder: Model User	Model Developer	Model Maintainer	Mdl Deployer-	Model Use Supporter	Regulatory Authority	Mdl Investor-	Model Type: Physics Based	Data Driven
	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	A summary of the exception noted to the current VVUQ Pattern (may be multiple exceptions)		X					X	X	X
			Impacted VVUQ Feature	The impacted existing, modified, or additional feature of the VVUQ Pattern.		X					X	X	X
			VVUQ Pattern Version	The version of the VVUQ Pattern in current use before change.		X					X	X	X
			Project	Identifies the project in which the exception was noted		X					X	X	X
			Person	Identifies the person describing the exception		X					X	X	X



Feature Group	Feature Name	Feature Definition	Feature Attribute	Attribute Definition	Feature Stakeholder							Model Type	
					Model User	Model Developer	Model Maintainer	Mdl Deployer-Distributor	Model Use Supporter	Regulatory Authority	Mdl Investor-Owner	Physics Based	Data Driven
Identifies the type of representation used by the model													
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.	X		X			X		X	X
			Conceptual Model Interoperability	The degree of interoperability of the conceptual model, for exchange with other environments	X		X			X		X	X
	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.	X		X			X		X	X
			Executable Model Interoperability	The degree of interoperability of the executable model, for exchange with other environments	X		X			X		X	X

Exercise 1: Model Planning, Targeting Business Values

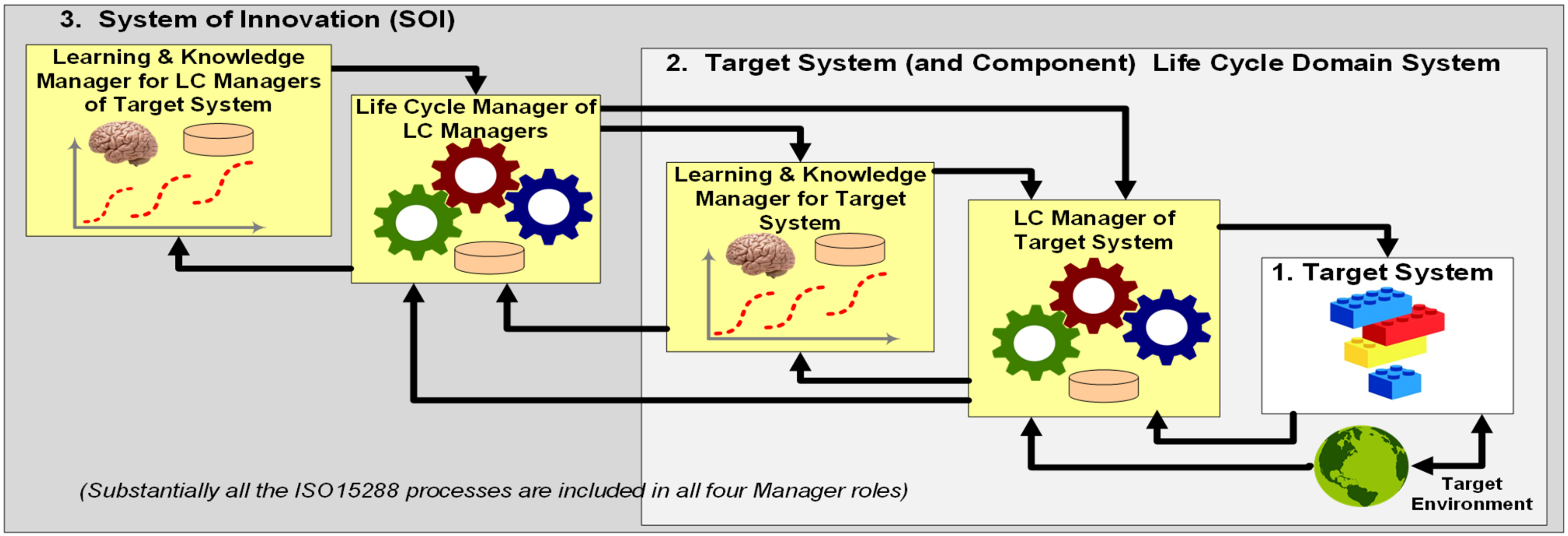
1. For a (real or hypothetical) use by your enterprise of a model-based approach, configure the VVUQ Model Features Pattern to describe your targeted outcomes – use the Model Features Pattern Form.
2. Did the VVUQ Features Pattern cover all your targeted improvement issues and concerns? Are there others?
3. What model credibility issues would have to be addressed by Model VVUQ?

Learning, versus Lessons Not Learned

- Practical steps to improve on organizational learning, using models as a focus of organizational learning and knowledge, based on model-based Learning Systems and Autonomous Systems.

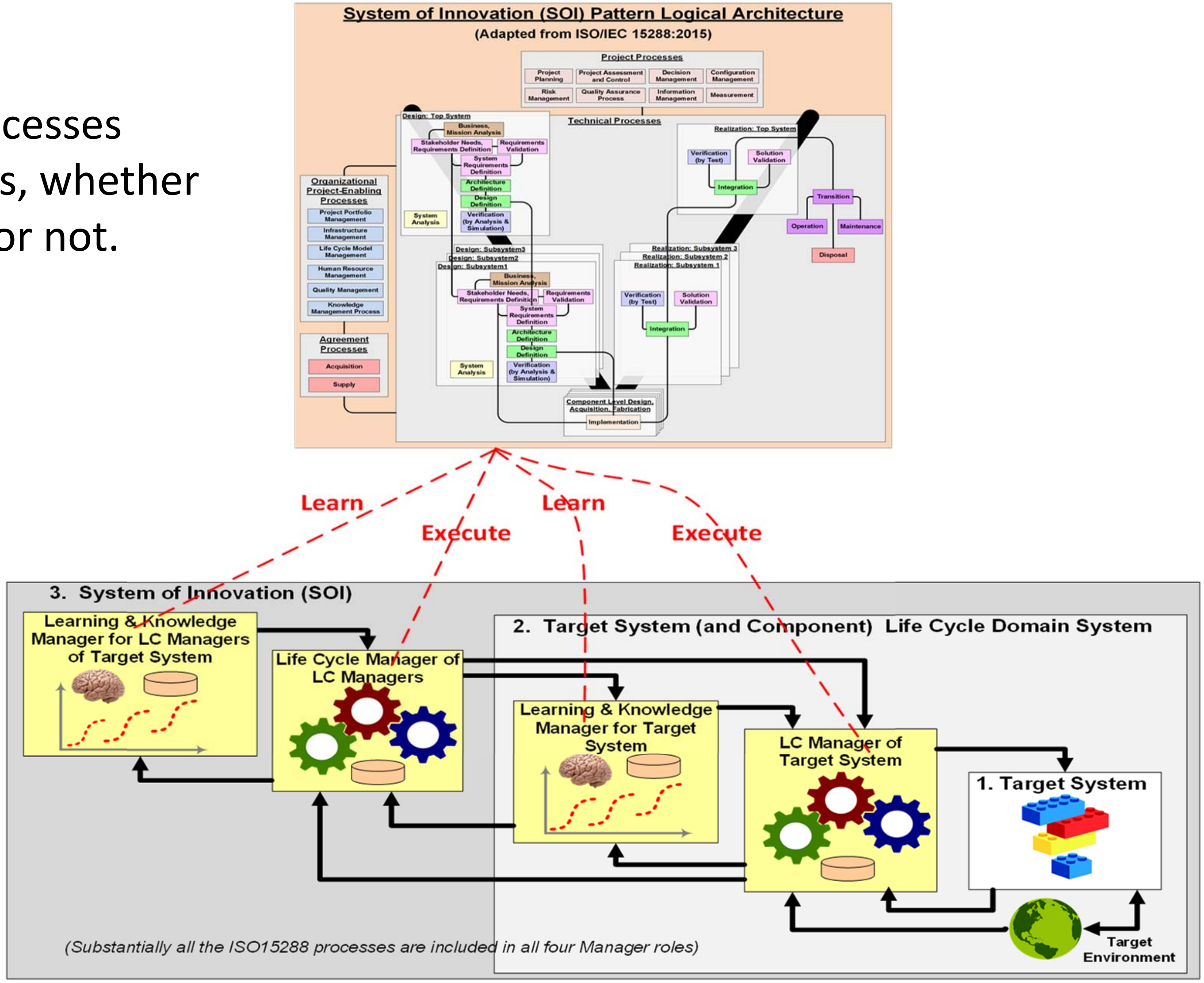
The System of Innovation (SOI) MBSE Pattern

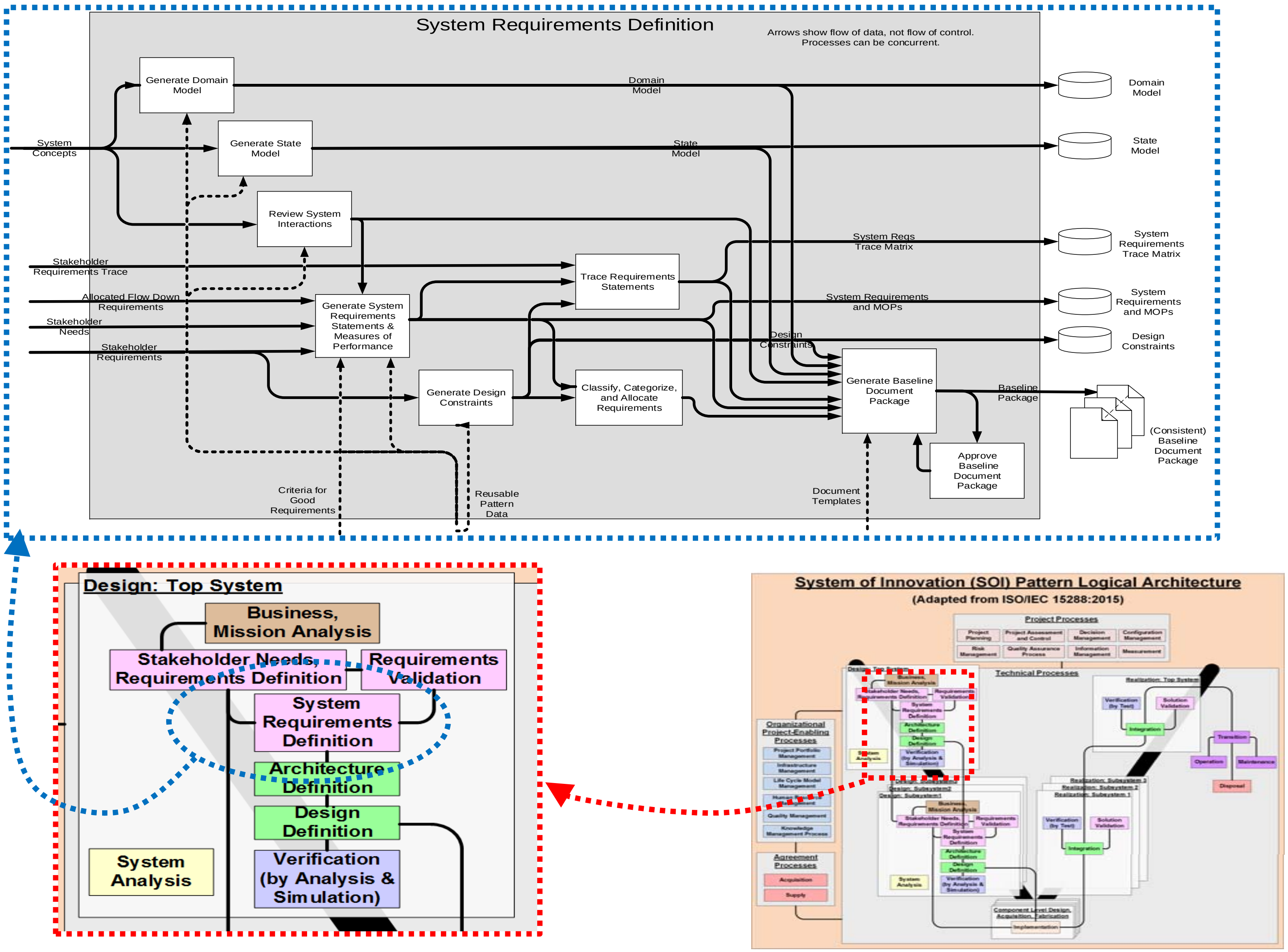
(Used for INCOSE Agile SE Project, INCOSE CIPR WG, etc.
Innovation reference model: Not prescriptive, but descriptive.)

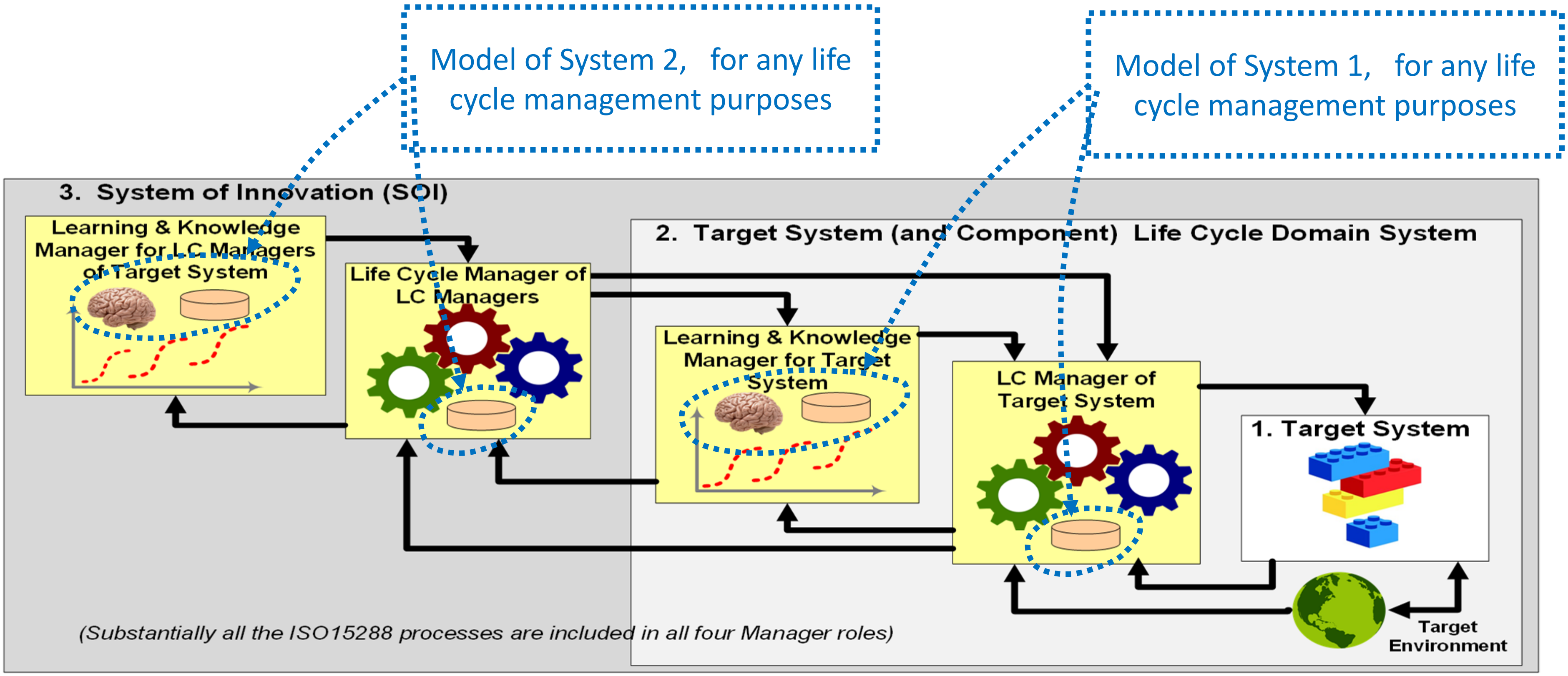


- System 1: Target system of interest, to be engineered or improved.
- System 2: The environment of (interacting with) S1, including all the life cycle management systems of S1, including learning about S1.
- System 3: The life cycle management systems for S2, including learning about S2.

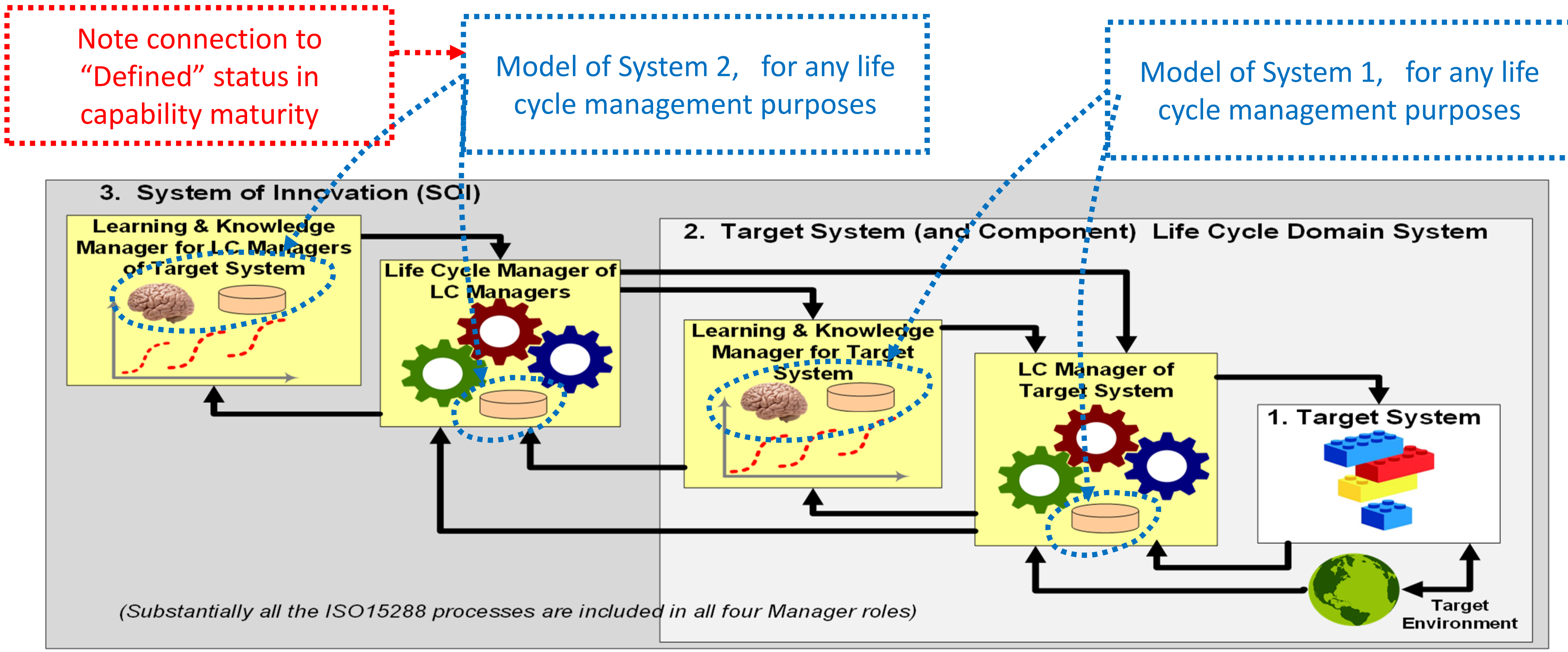
ISO 15288 processes appear 4 times, whether we recognize or not.




System Requirements Definition
Arrows show flow of data, not flow of control.
Processes can be concurrent.
Generate Domain Model
Generate State Model
Review System Interactions
Generate System Requirements Statements & Measures of Performance
Trace Requirements Statements
Generate Design Constraints
Classify, Categorize, and Allocate Requirements
Generate Baseline Document Package
Approve Baseline Document Package
System Concepts
Stakeholder Requirements Trace
Allocated Flow Down Requirements
Stakeholder Needs
Stakeholder Requirements
Domain Model
State Model
System Reqs Trace Matrix
System Requirements and MOPs
Design Constraints
Baseline Package
System Requirements Trace Matrix
(Consistent) Baseline Document Package
Criteria for Good Requirements
Reusable Pattern Data
Document Templates
Design: Top System
Business, Mission Analysis
Stakeholder Needs, Requirements Definition
Requirements Validation
System Requirements Definition
Architecture Definition
Design Definition
Verification (by Analysis & Simulation)
System Analysis
System of Innovation (SOI) Pattern Logical Architecture
(Adapted from ISO/IEC 15288:2015)
Project Processes
Technical Processes
Organizational Project-Enabling Processes
Agreement Processes
Realization: Top System
Transition
Operation
Maintenance
Disposal



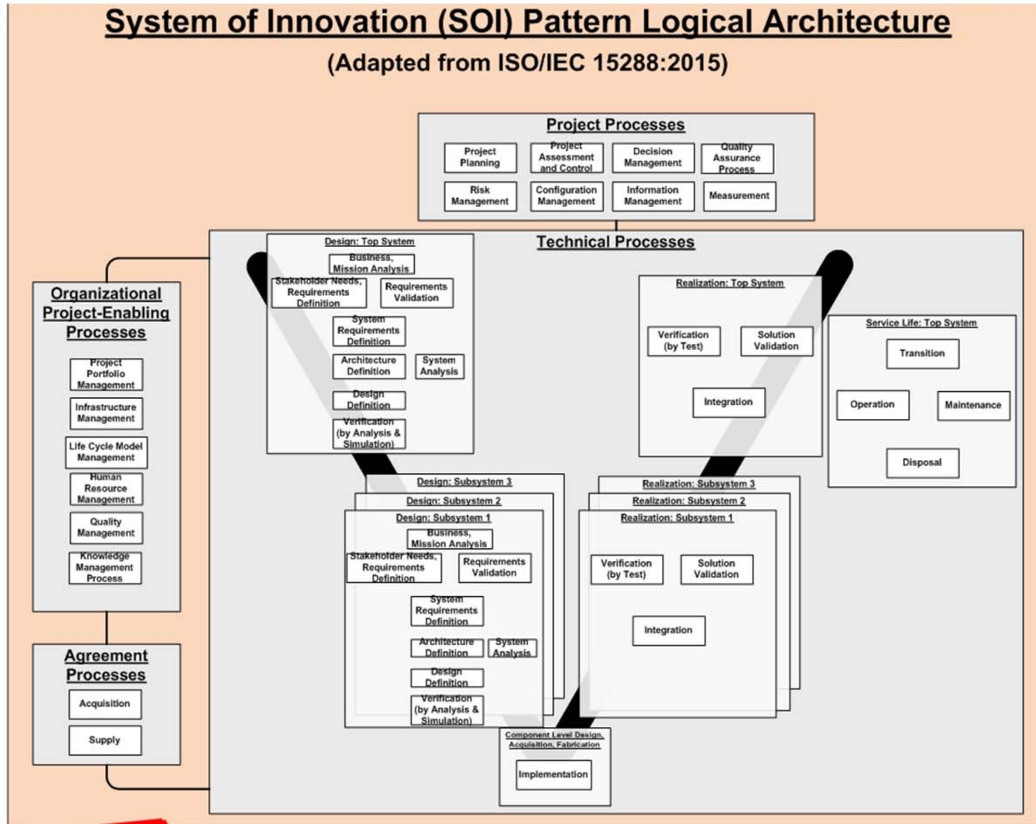
- System 1: Target system of interest, to be engineered or improved.
- System 2: The environment of (interacting with) S1, including all the life cycle management systems of S1, including learning about S1.
- System 3: The life cycle management systems for S2, including learning about S2.

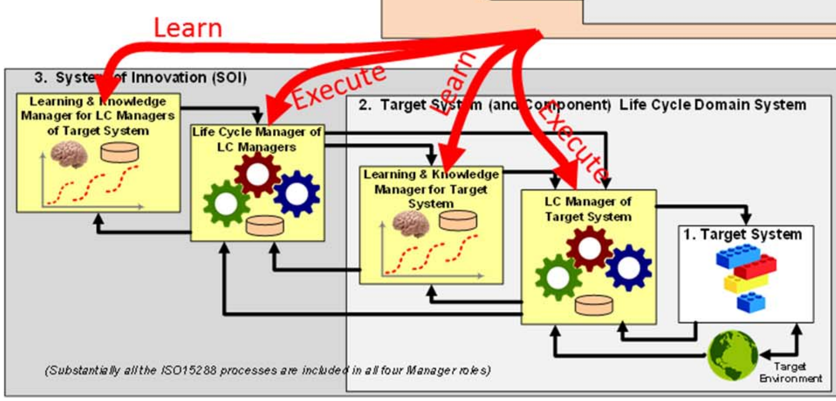


- System 1: Target system of interest, to be engineered or improved.
- System 2: The environment of (interacting with) S1, including all the life cycle management systems of S1, including learning about S1.
- System 3: The life cycle management systems for S2, including learning about S2.

Both System 1 and System 2 are potentially subject to learning.

System 2: Each of the ISO15288 Processes Appears repeatedly in the ASELCM Pattern:

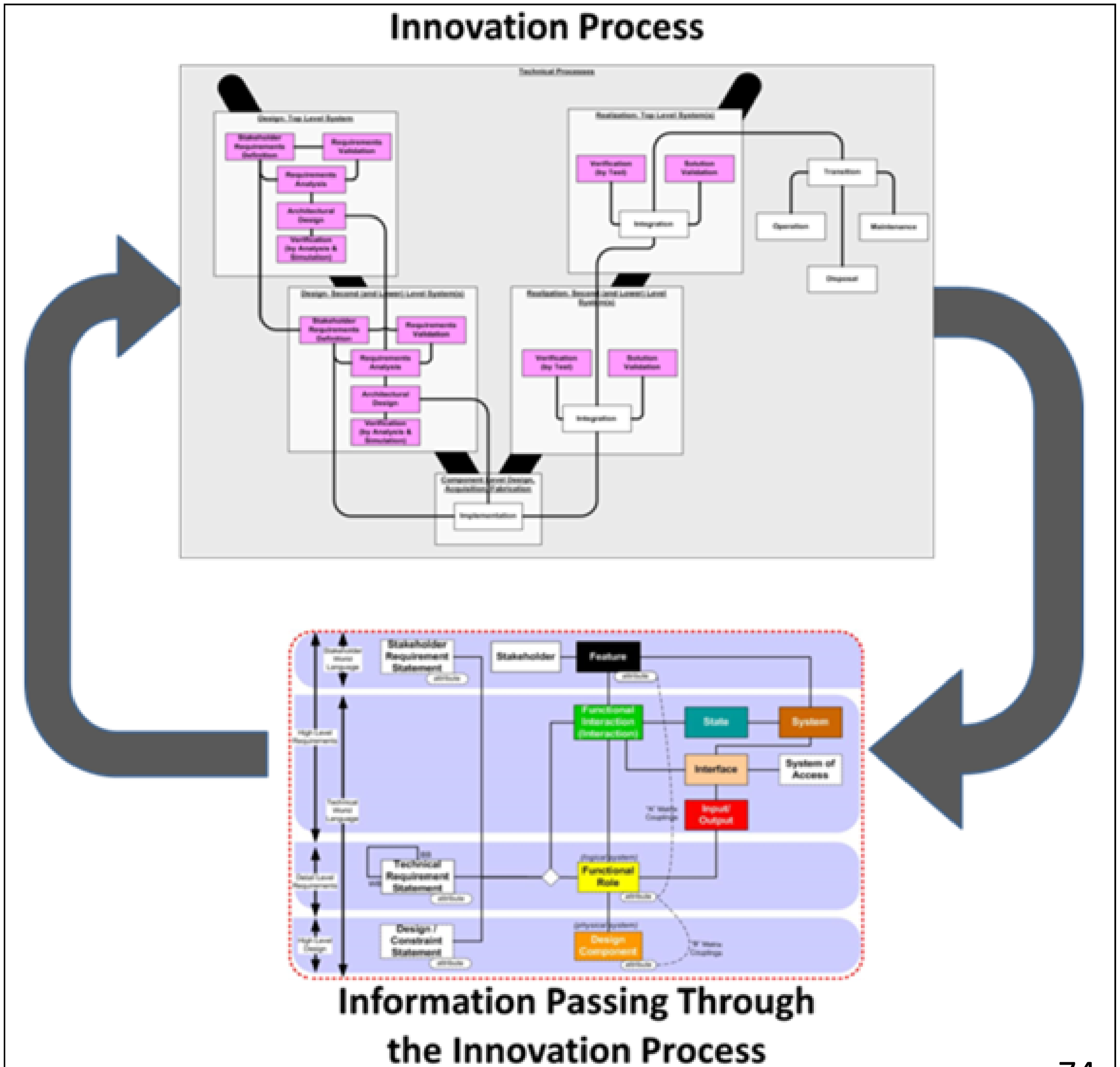




They appear repeatedly, in different ways in the SOI & ASELCM Patterns . . . . . .

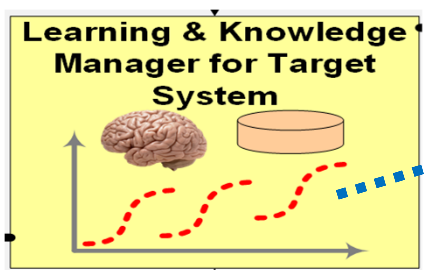
From Systems Engineering to Systems Innovation:

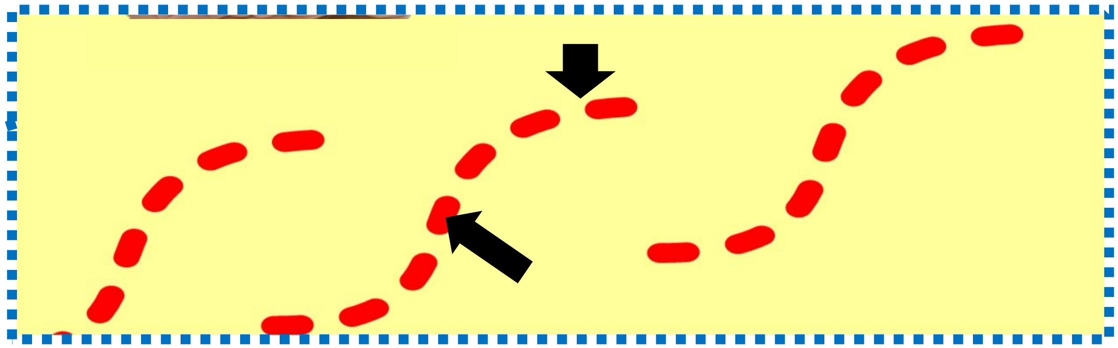
Shifting the emphasis from traditional focus on procedure, to greater emphasis on the state of the web of information passing through the process



When is immaturity valued?

- The progressive "S Curves" of waves of new technologies, paradigms, product families, scientific, and other discoveries represent learning.
- In this context, "maturity" is the flat part at the top of each generation of learning.
- The earlier, "steep" part of the curve represents higher rates of change, as we learn more rapidly and exploit discovery.





- So, where do we want to be on this curve?
- Notice the challenging trade-off!
- Applies to learning about System 2 (e.g., methodology) as well as Learning about System 1 (engineered system).

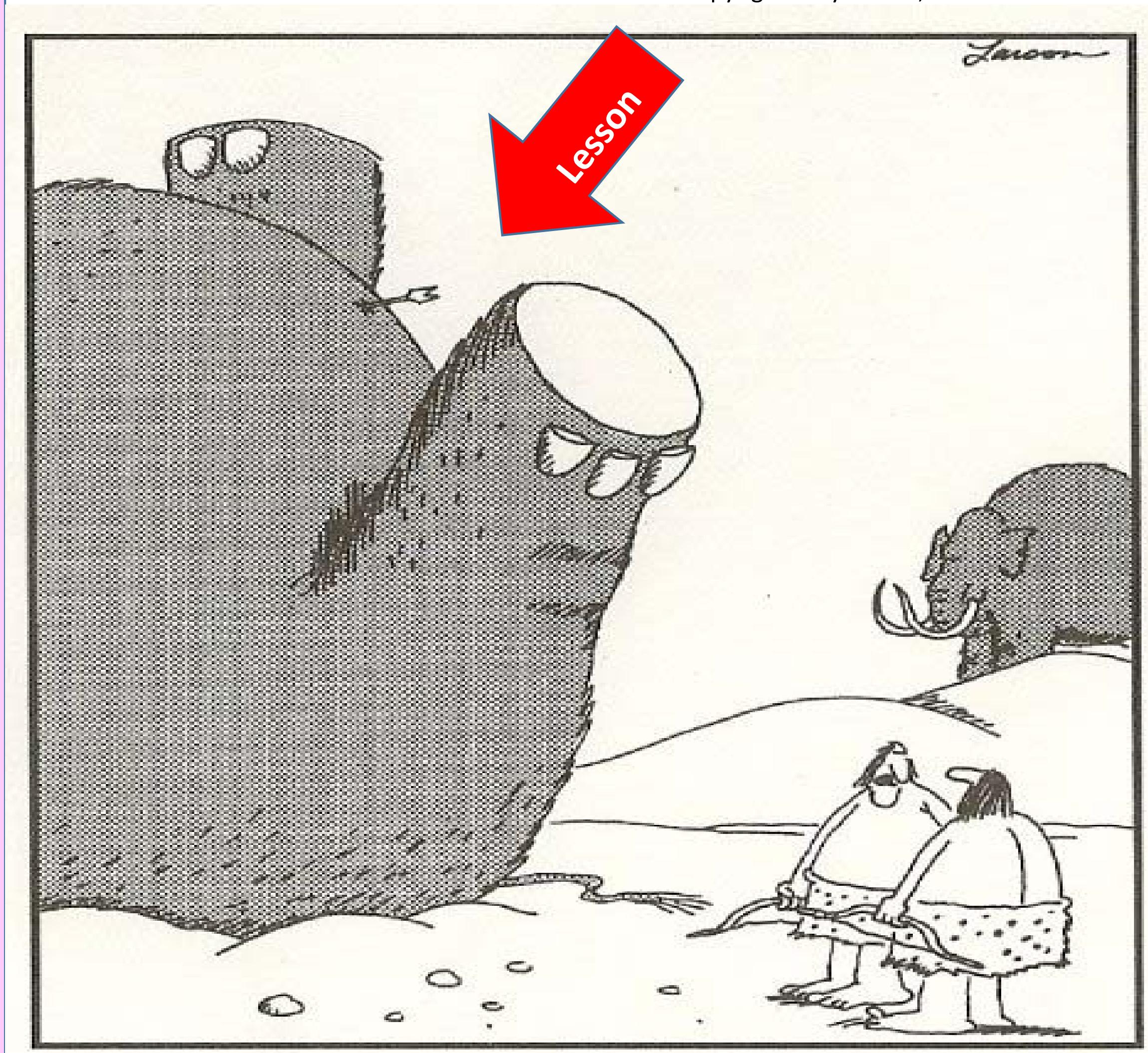
Lessons Learned: <u>Effective</u> Learning?

- 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 <u>could</u> be impacted <u>is</u> 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.

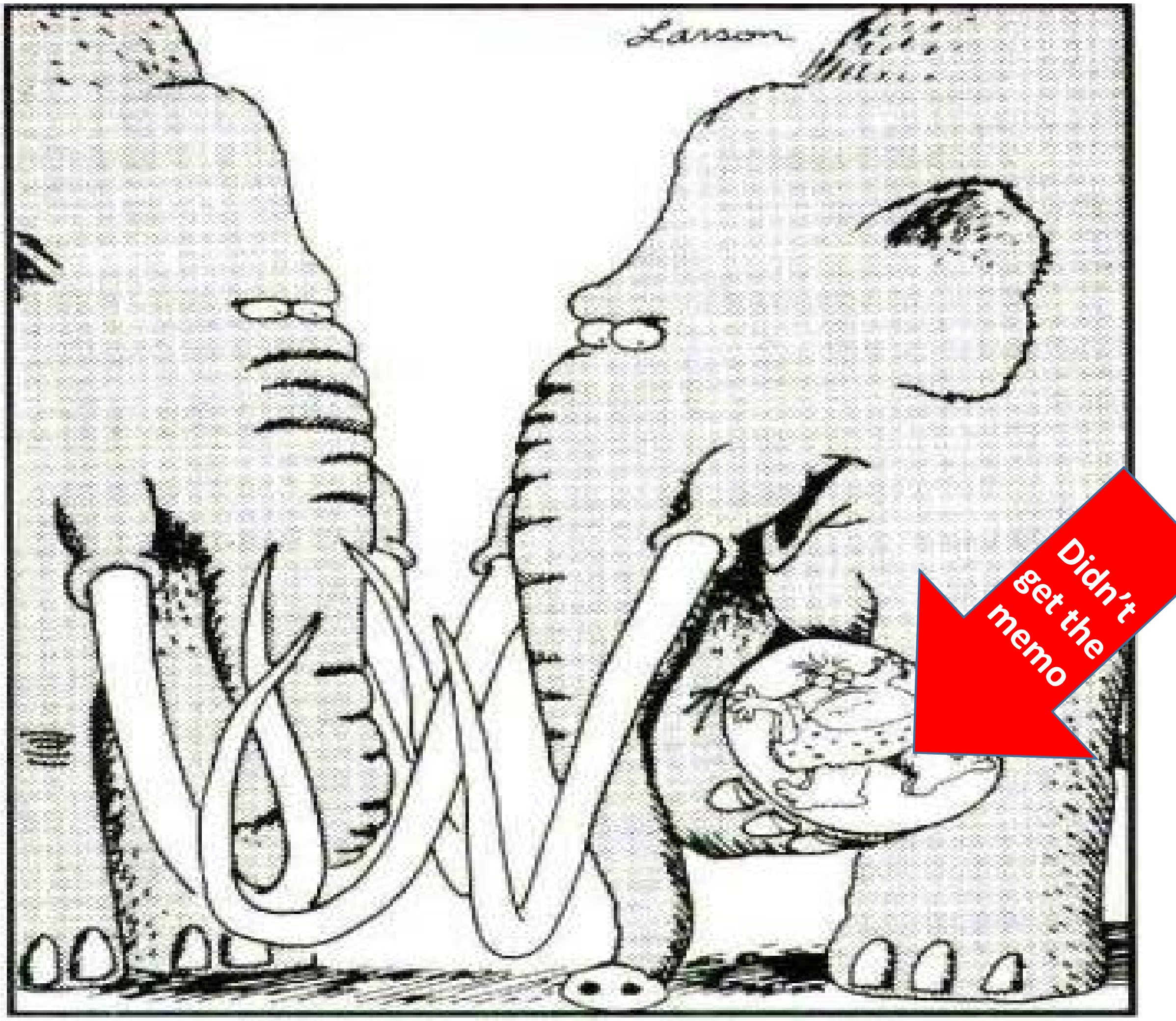


"We should write that spot down."

Lessons <u>Effectively</u> Learned?

Lessons Learned Report

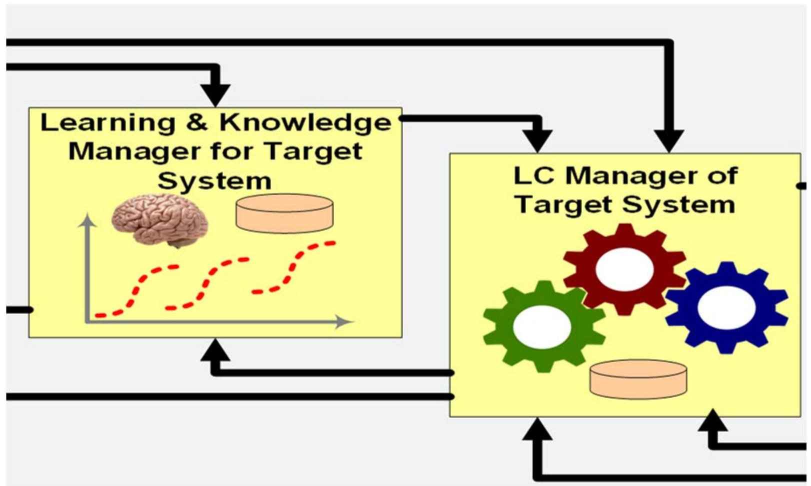
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed aliquam odio eget massa feugiat, at tincidunt quam ullamcorper. Nullam ac purus tortor. Duis a ullamcorper augue. Pellentesque eu eros hendrerit, tempor tellus vitae, suscipit.

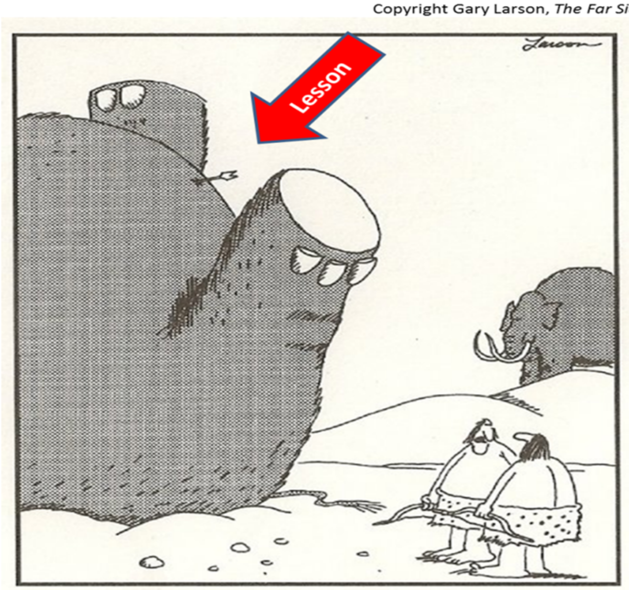


"Well, what the? ... I *thought* I smelled something."

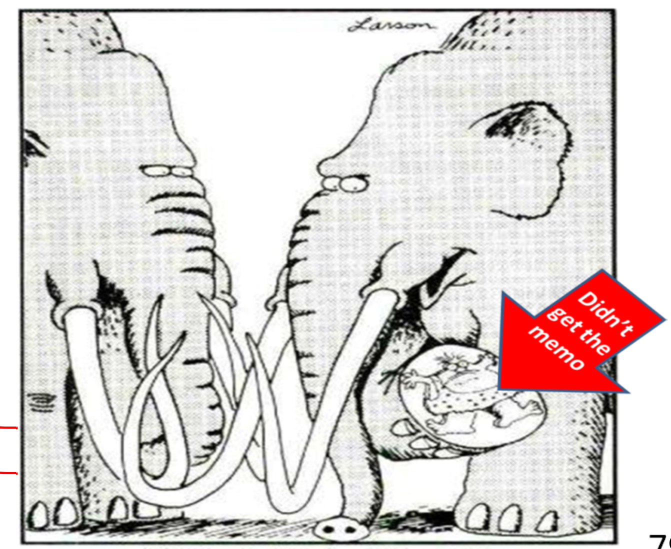
Learning

Executing



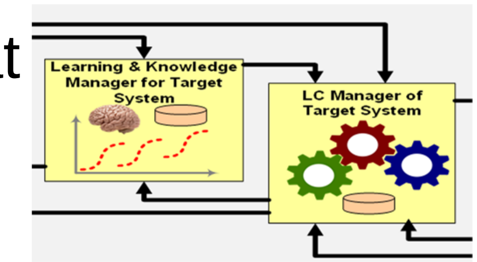


"We should write that spot down."



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

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

- <u>Where</u> are the "lessons learned" encoded? What 

  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.

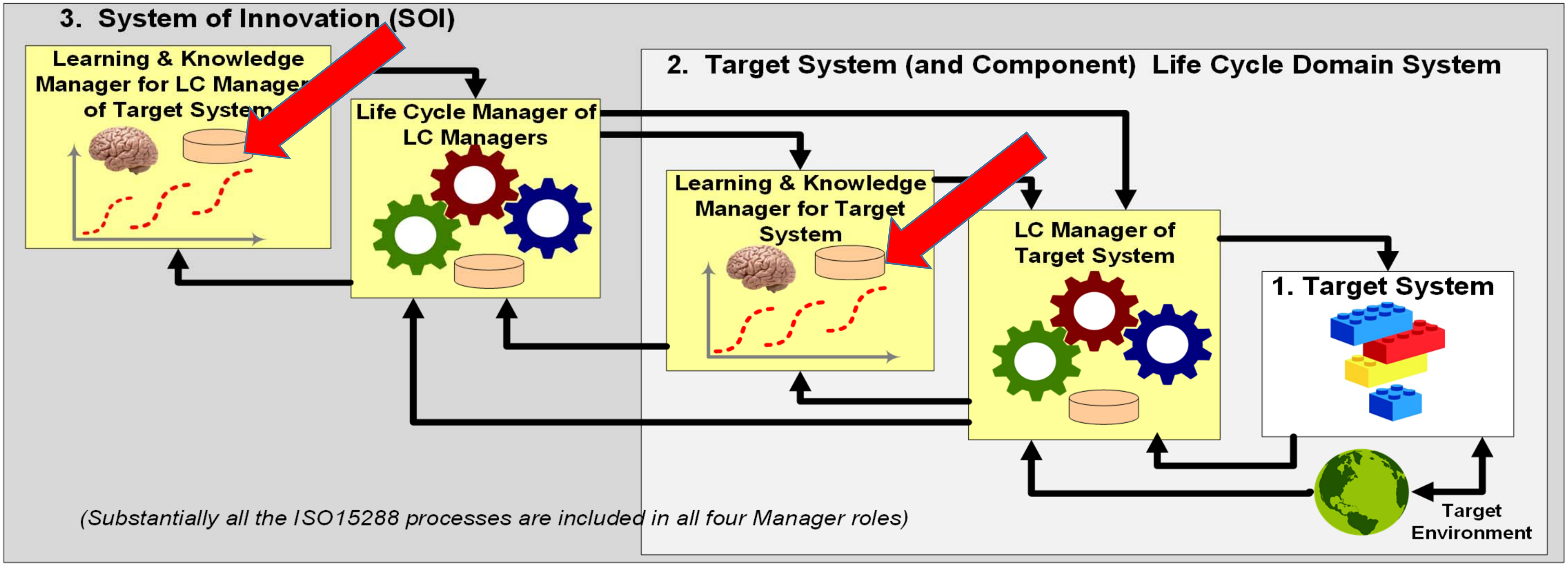
Learned models from STEM (~300 years) offer the most dramatic example of positive collaborative impact of effectively <u>shared</u> and <u>validated</u> models

- Effective Model <u>Sharing</u>:
  - 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 <u>Validation</u>:
  - Especially when shared, models demand that we <u>trust</u> 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 <u>add new costs to development of models</u> 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 <u>engineering professional societies</u>, 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 <u>trust in models</u> . . .

An emerging special case: Regulated markets



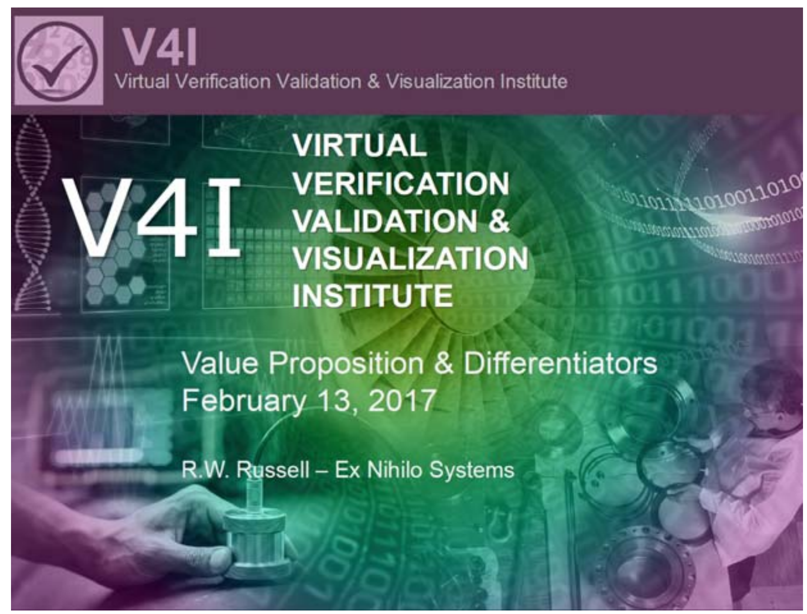
- Trusted shared MBSE Patterns for classes of systems
- Configurable for vendor-specific products
- With Model VVUQ frameworks lowering the cost of model trust for regulatory submissions

Exercise 2: Targeted Learning Areas

1. Identify and list the opportunities in your enterprise and process to capture what is learned in system patterns used as the basis of future projects.
2. Which are System 1 and which are System 2?

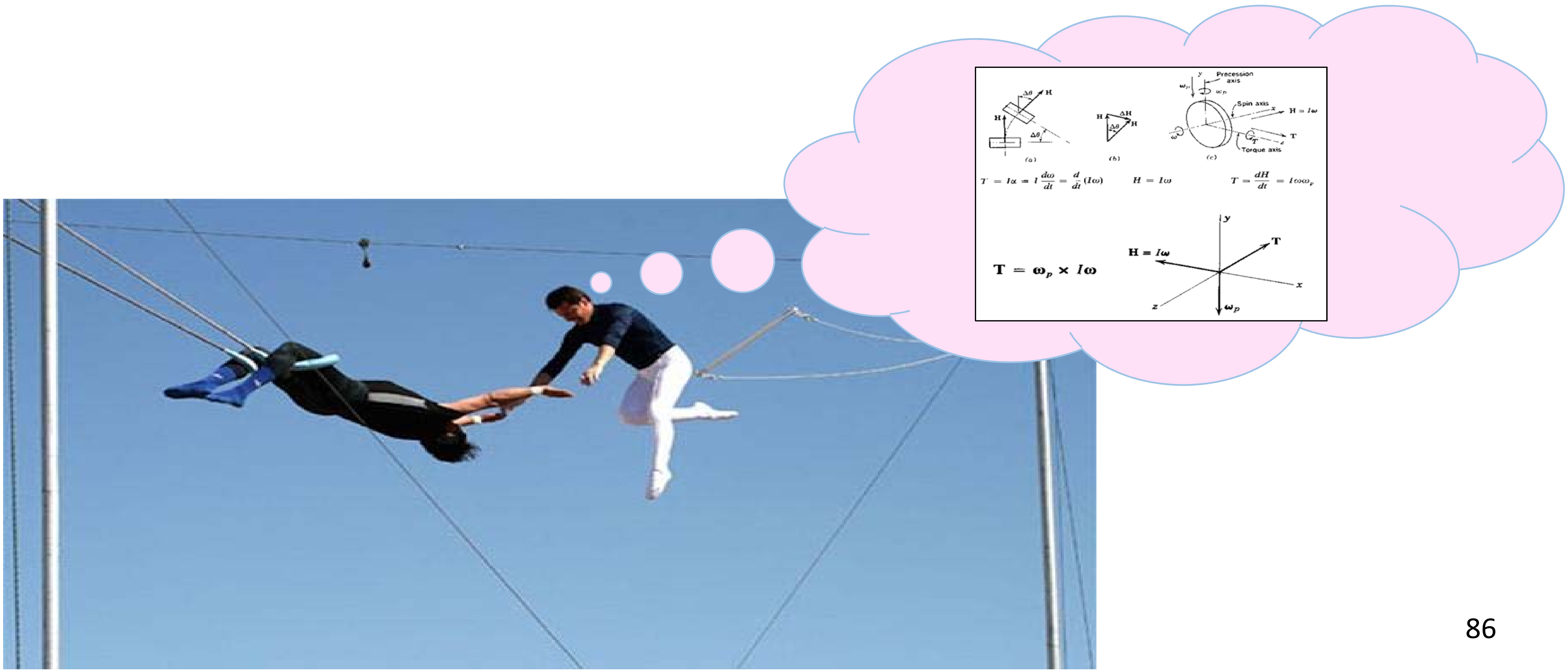
Can You Trust Someone Else's Model? Your Model?

- Planning for Model Verification, Validation, and Uncertainty Quantification (Model VVUQ)



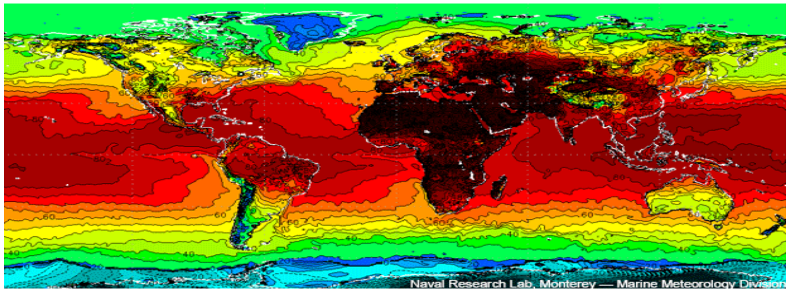
Requirements for trustable models

We cannot discuss maturity in development or use of models without discussing whether we can trust those models . . .



If we expect to use models to support critical decisions, then we are placing *increased trust in models*:

- Critical financial, other business decisions
- Human life safety
- Societal impacts
- Extending human capability



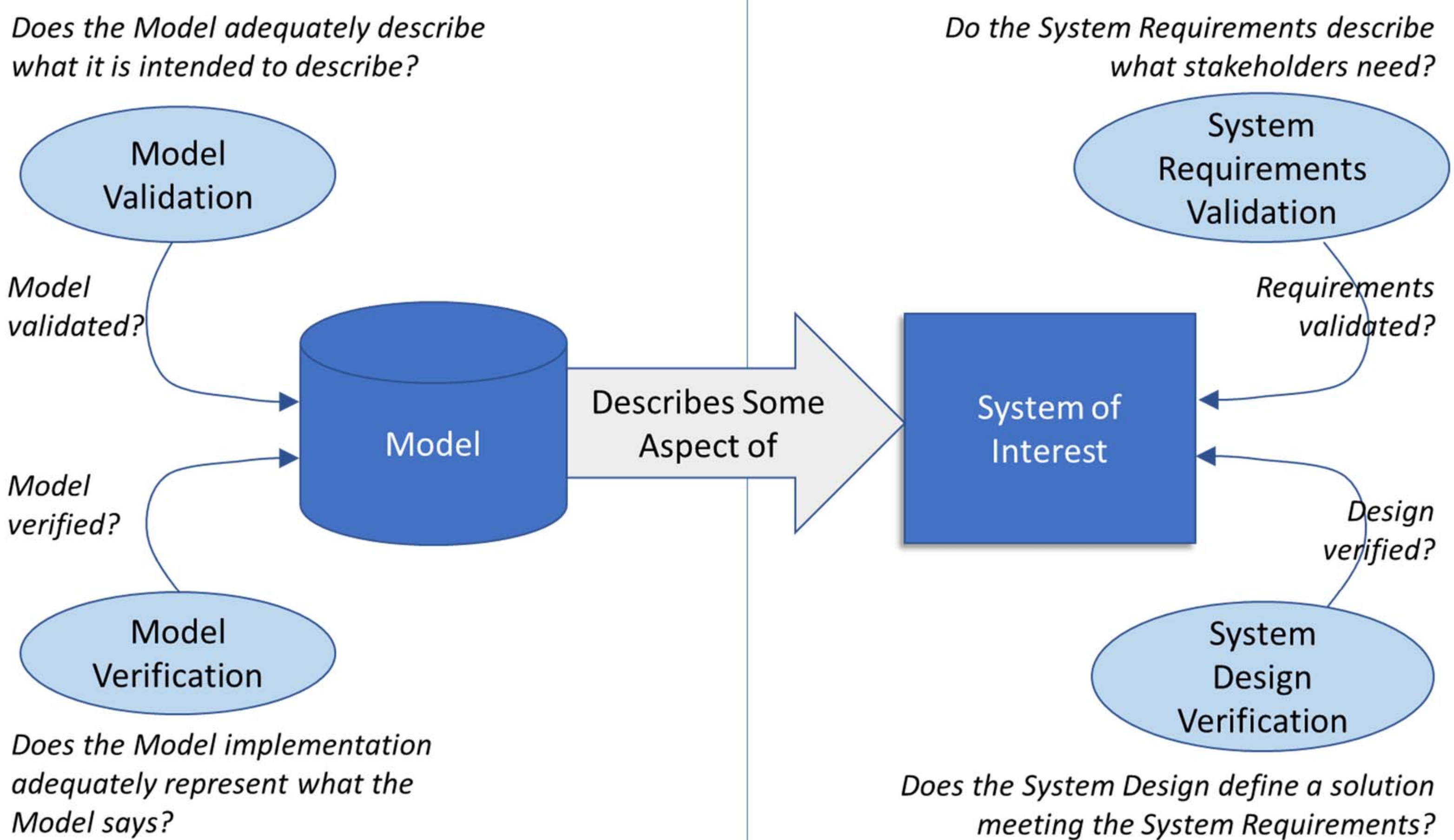
- MBSE Maturity requires that we characterize the structure of that trust and manage it:
  - The Validation, Verification, and Uncertainty Quantification (VVUQ) *of the models themselves*.

What is meant by VVUQ of a model?

- Model Validation (V)
- Model Verification (V)
- Model Uncertainty Quantification (UQ)
- Not just for numerical grid (FEA, CFD, Thermal) models—extension to system models at all levels.
- Bayesian Network aspects of UQ

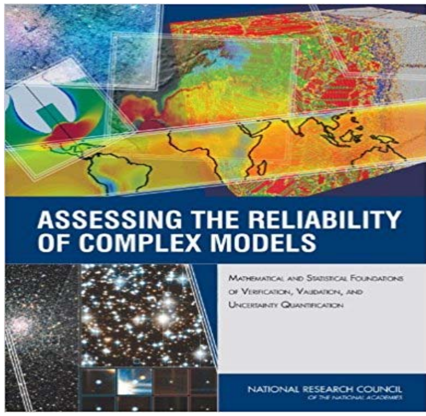
V&V of *Models*, Per Emerging ASME Model V&V Standards

V&V of *System Requirements, Designs*, Per ISO 15288 & INCOSE Handbook



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, in ways viewed as the heart of this work.
- But, some additional systems work is needed, and in progress, as to the more general VVUQ framework, suitable for general standards or guidelines.

General structure of uncertainty / confidence tracing:

- Do the modeled external Interactions qualitatively cover the modeled Stakeholder Features over the range of intended S1 situations of interest?
- Quantify confidence / uncertainty that the modeled Stakeholder Feature Attributes quantitatively represent the real system concerns of the S1 Stakeholders with sufficient accuracy over the range of intended situation envelopes.
- Quantify confidence / uncertainty that the modeled Technical Performance Attributes quantitatively represent the real system external behavior of the S1 system with sufficient accuracy over the range of intended situation envelopes.

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 models"
- Also conducts annual Symposium on Validation and Verification of Computational Models, in May.
- To participate in this work, in 2016 the speaker joined the ASME VV50 Committee:
  - 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.
- Also invited sub-team leader Joe Hightower (Boeing) to address the INCOSE IW2017 MBSE Workshop, on our related ASME activity.

ASME Verification & Validation Standards Committee

- V&V 10: Verification & Validation in Computational Solid Dynamics
- V&V20: Verification & Validation in Computational Fluid Dynamics and Heat Transfer
- V&V 30: Verification and Validation in Computational Simulation of Nuclear System Thermal Fluids Behavior
- V&V 40: Verification and Validation in Computational Modeling of Medical Devices
- V&V 50: Verification & Validation of Computational Modeling for Advanced Manufacturing
- V&V 60: Verification and Validation in Modeling and Simulation in Energy Systems and Applications

https://cstools.asme.org/csconnect/CommitteePages.cfm?Committee=100003367

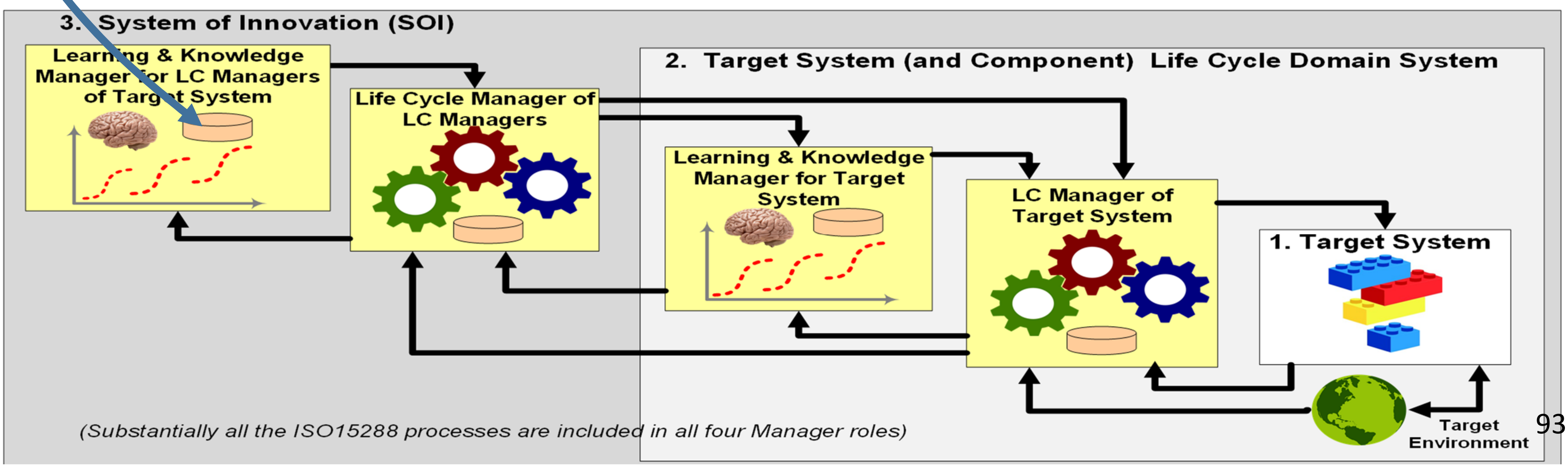
ASME SETTING THE STANDARD

Requirements for trustable, impactful models, as a basis for MBSE maturity

INCOSE MBSE Assessment and Planning Pattern

MBSE Maturity in general, and VVUQ for Models in particular, mean we have to understand:

- Stakeholders for Models
- Stakeholder Features of Models
- Technical Requirements for Models
- We are capturing these in an MBSE Pattern



Opportunities--what you can do

- Think larger about <u>intended uses and users of MBSE</u>, and judge its maturity in that light.
- Include how well MBSE enables group learning.
- Include the full breadth of model types in your thinking.
- Consider why you think a model should be trusted.
- Join the INCOSE MBSE Patterns Working Group, to advance practice.
- Join the ASME Computational VVUQ effort, to advance model trust.
- Exercise the emerging MBSE Planning and Assessment Framework, in your own company and work, and provide feedback.

Exercise 3: Identifying Credibility Needs for Trusted System Patterns

1. Where and when, in your enterprise organization and process, could a trusted system pattern be consulted as the basis for configuring system Requirements, Designs, Failure Analysis, Manufacturing, Distribution, Support, or otherwise?
   (Hint: Consider your answers to Exercise 2.)
2. What would be the model credibility issues that would need to be addressed? What could be the benefits of a trustable model?

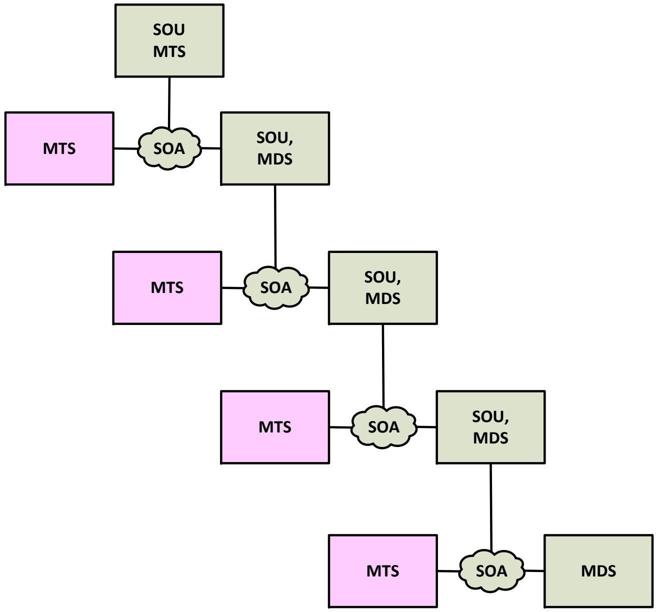
End of Part I

Part II (Afternoon)

- **The Embedded Intelligence (EI) Pattern**: For any embedding of intelligence, in the form of automation, human operators, or other systems of management, feedback, regulation.
- **The Smart Manufacturing Pattern, for the IoT Age**: For any manufacturing process, and with varied forms of instrumentation and management.
- **Capitalization of MBSE Patterns as Financial Assets**: How to shift the burden of model cost to the time of use and benefit.
- Exercises

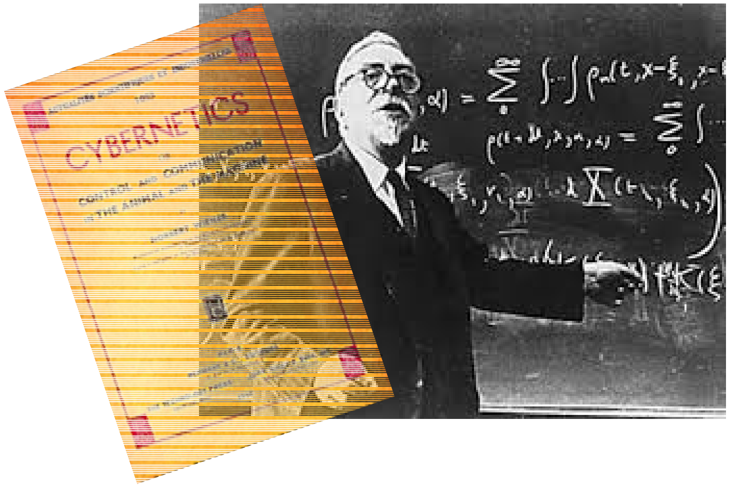
The Embedded Intelligence (EI) Pattern

- For any embedding of intelligence, in the form of automation, human operators, or other systems of management, feedback, regulation.



Embedded Intelligence (EI) Pattern

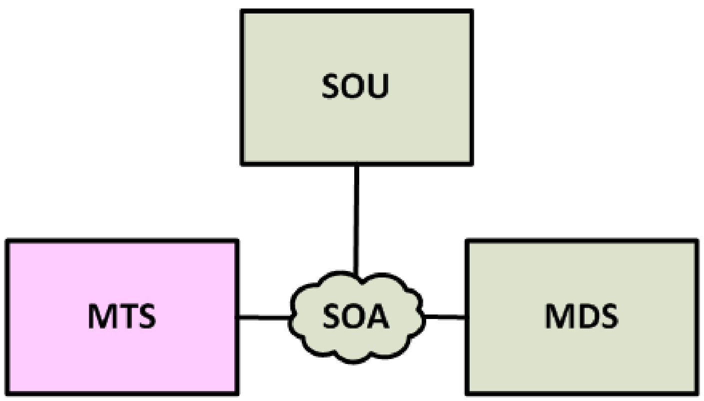
- The EI Pattern returns to the perspective of Norbert Wiener, who first coined the term “cybernetics” to refer to the study of communication and control in living and human-engineered systems:

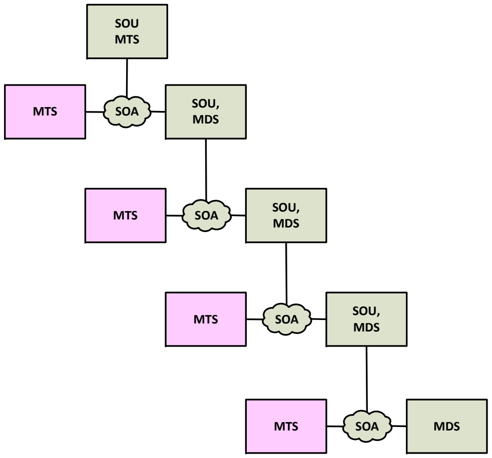


- Especially appropriate if we are interested in Cyber-Physical Systems – but now we are interested in more than just feedback and control performance (studied by Wiener) . . .

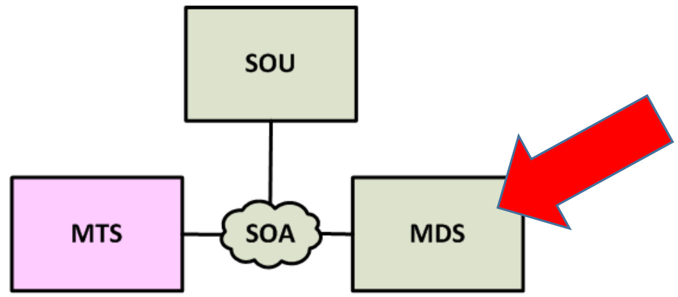
Embedded Intelligence (EI) Pattern

- The EI Pattern is an S*Pattern that emerges to describe intelligence in explicit models of evolving systems in the natural and man-made world:
  - Also referred to as the Management System Pattern.
  - Concerned with the emergence of four roles, emergent at multiple hierarchical levels:



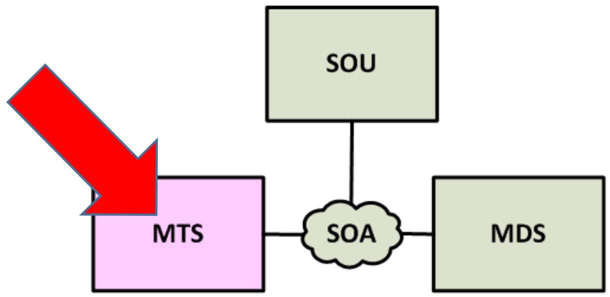


Embedded Intelligence (EI) Pattern



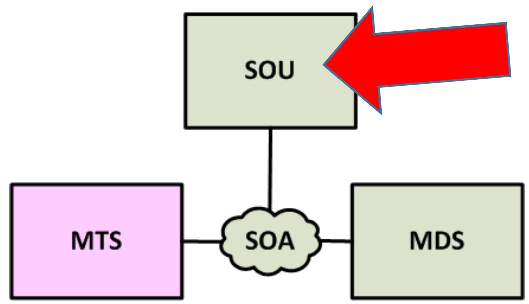
- Managed System (MDS): Any system behavior whose performance, configuration, faults, security, or accounting are to be managed--referred to as System Management Functional Areas (SMFAs) or in ISO terminology fault, configuration, accounting, performance, security (FCAPS).
- These are the roles played by the so-called “physical systems” in a cyber-physical system, providing physical services such as energy conversion, transport, transformation, or otherwise.

Embedded Intelligence (EI) Pattern



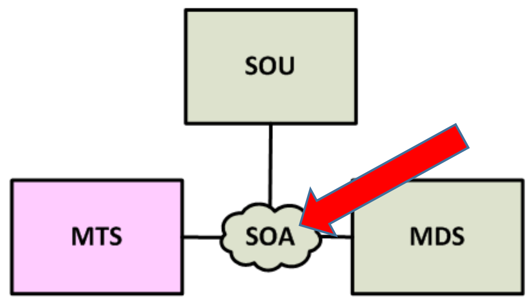
- Management System (MTS): The roles of performing management (active or passive) of any of the SMFAs of the managed system.
- These are so-called “cyber” roles in a cyber-physical system, and may be played by automation technology, human beings, or hybrids thereof, to accomplish regulatory or other management purposes.

Embedded Intelligence (EI) Pattern



- System of Users (SOU): The roles played by a system which consumes the services of an managed system and/or management system, including human system users or other service-consuming systems at higher levels.

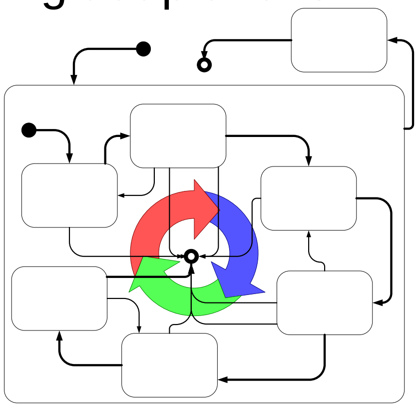
Embedded Intelligence (EI) Pattern



- System of Access (SOA): The roles providing a means of interaction between the other EI roles.
- Engineered sensors, actuators, the Internet, and human-machine interfaces have contributed greatly to the emergence of the “Internet of Things”..

Embedded Intelligence (EI) Pattern

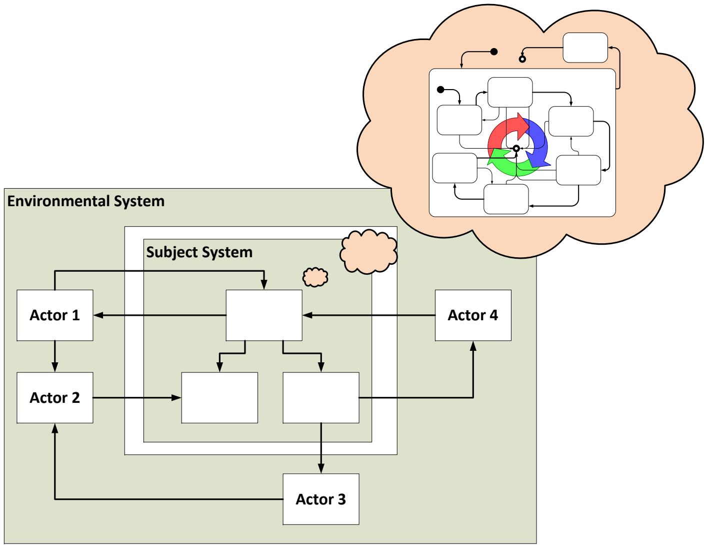
- The State Model portion of the EI Pattern provides insight into the nature of the "regulatory" role of embedded intelligence.
- These show numerous "situation resolution cycles" that drive the managed system to nominal states, when various situations are encountered:
  - Major mission cycles, from mission start to completion
  - Fault resolution cycles, other lesser or minor situation resolution cycles
  - Configuration change cycles, including adaptations
  - Fulfillment of requests for services
  - Security condition resolution cycles
  - Other situation resolution cycles
- Specific or general situations



Sample EI Situation Resolution Cycle

Embedded Intelligence (EI) Pattern

- A system that is capable of not only traversing a situation resolution cycle, but also recognizing that a triggering situation has arisen in the first place is said to be "Situationally Aware":
  - If a human operator control panel has a "mode switch", the system relies on the human to be aware of situations, launching the appropriate cycles
  - More advanced systems recognize these situations autonomously—also leading to EI Attention Model recognition of finite system resources.

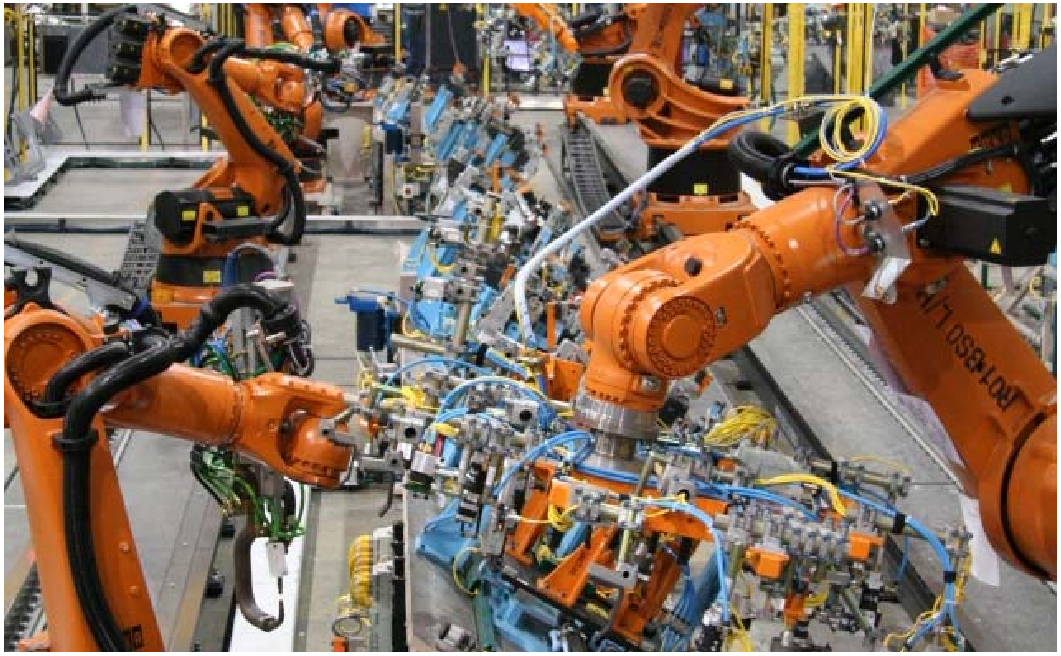


Exercise 4: Applying the Management System Pattern (EI Pattern)

1. Identify a possible Management Systems application for the EI Pattern, for some system of interest. What is the Managed System?
2. Are there multiple levels of control for your example? Draw a multi-level EI Hierarchy and identify the levels.
3. Are there human-filled Management System roles? Automation-filled Management System roles?
4. Which of the five System Management Functional Areas (SMFAs) are involved?
5. What types of Management Situations would occur, for resolution by the Management System?

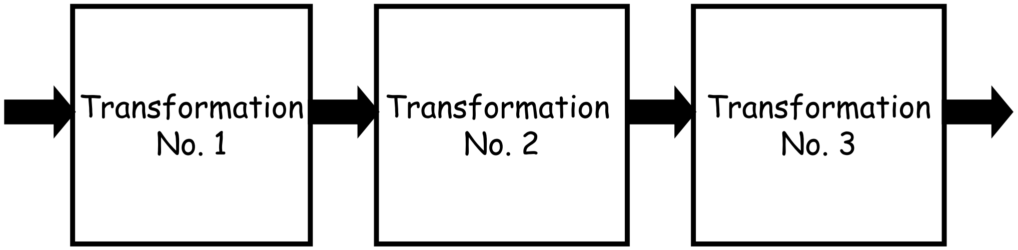
The Smart Manufacturing Pattern, for the IoT Age

- For any manufacturing process, and with varied forms of instrumentation and management.



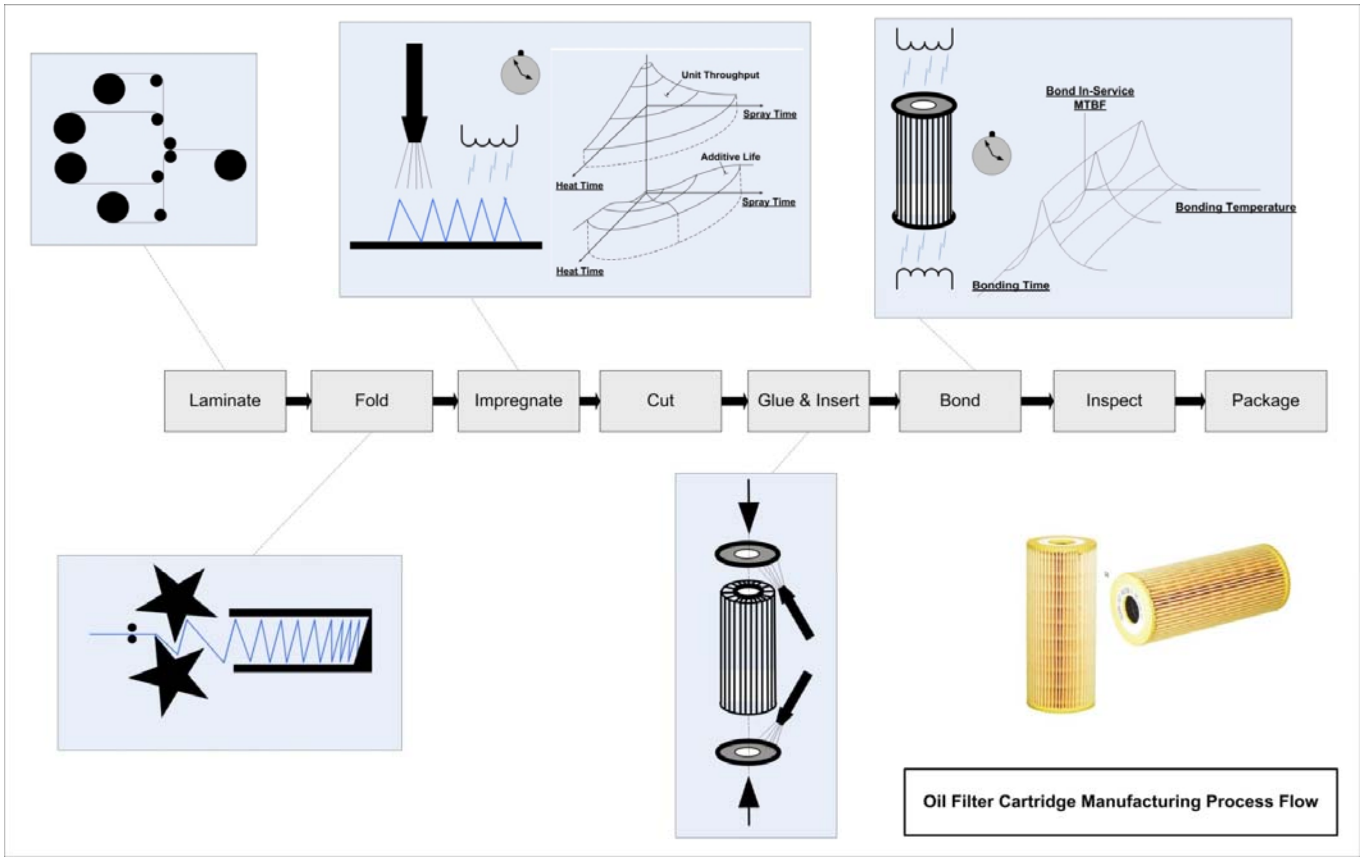
The Process Engineer's Perspective

- Process Engineers are trained to visualize manufacturing as transformations of material (or of information).
- This is frequently represented graphically using Process Flow Diagrams (PFDs):



- The material flowing out is different than the material flowing in--it is "transformed" chemically, structurally, thermodynamically, as information, visually, etc.

A Simple Example:
Manufacturing Oil Filter Cartridges



Process Engineering vs. Equipment Design

- By omitting equipment-specific design, the PFD perspective has the advantage of emphasizing what is required to be changed (transformed) about the material, without describing how manufacturing equipment, tools, people, or control systems will accomplish those transformations:

- Since it describes the required transformations, it is a form of partial <u>requirements on a manufacturing system</u>.

Process Engineering Challenges

- Process Engineering and Process Flow Diagrams provide powerful tools for conceptualizing manufacturing processes.
- However, the fact they use a perspective or language separate from design of equipment requires that the enterprise bridge a gap when integrating PE into the larger engineering context.
- For example, not all requirements on a manufacturing system are requirements of the process itself—they may even conflict.
- Various enterprises and trade groups have wrestled with the question of integrating the larger engineering process for manufacturing systems . . .

Integration with the larger engineering context: Challenges

1. How can the language and perspective of process engineers be more effectively coupled to those of equipment designers?
2. How do process requirements fit into overall manufacturing system requirements, which have larger scope?
3. What is the relationship of physical equipment design to these requirements?
4. How can process requirements for new or modified products be incorporated early enough in the equipment design cycle?
5. How are manufacturing system requirements that are not transformation of materials related to this?
6. How can we conceive new manufacturing solutions without being mentally trapped in assuming constraints of past designs?
7. How can candidate manufacturing designs, design changes, or design risks be evaluated in light of process engineering needs?
8. How are industry reference models of manufacturing (e.g., ISA, ISPE, etc.) related to these issues?
9. More generally, how can increasingly complex advanced manufacturing systems best be engineered, over their life cycles?

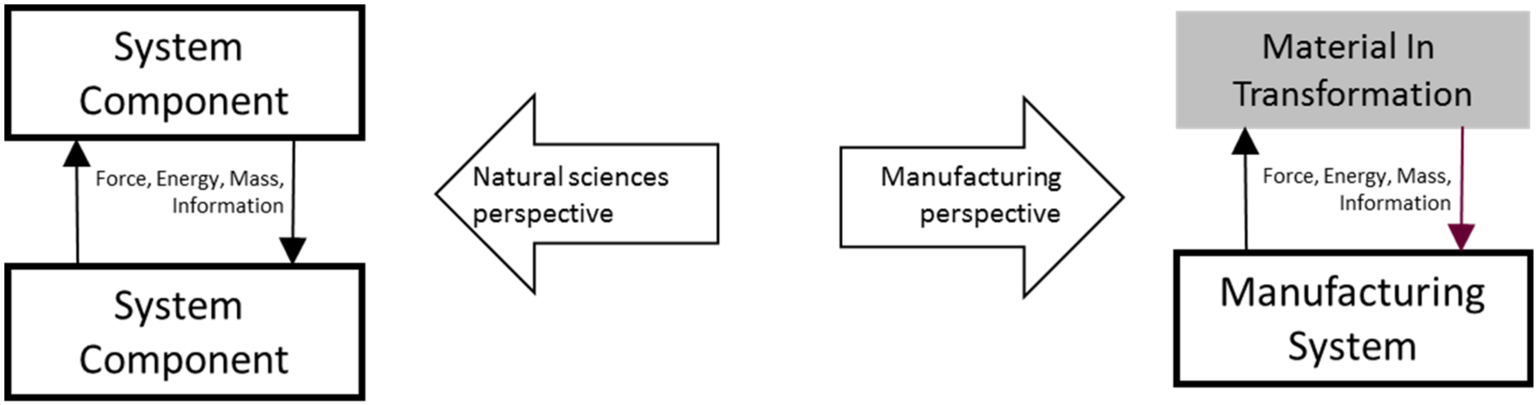
The need for a Science-based Understanding

- Industry trends increasingly emphasize science-based understanding of manufacturing processes:
  - Unit operations: key parametric relationships—materials science, chemistry, physics, etc.
  - First principle and empirical characterizations;
  - Mathematics of production flow;
  - Process capabilities and control laws;
  - Regulatory (e.g., FDA) pressures for a more science-based approach.

- How do we fit science-based understanding into an integrated framework of process and equipment engineering?

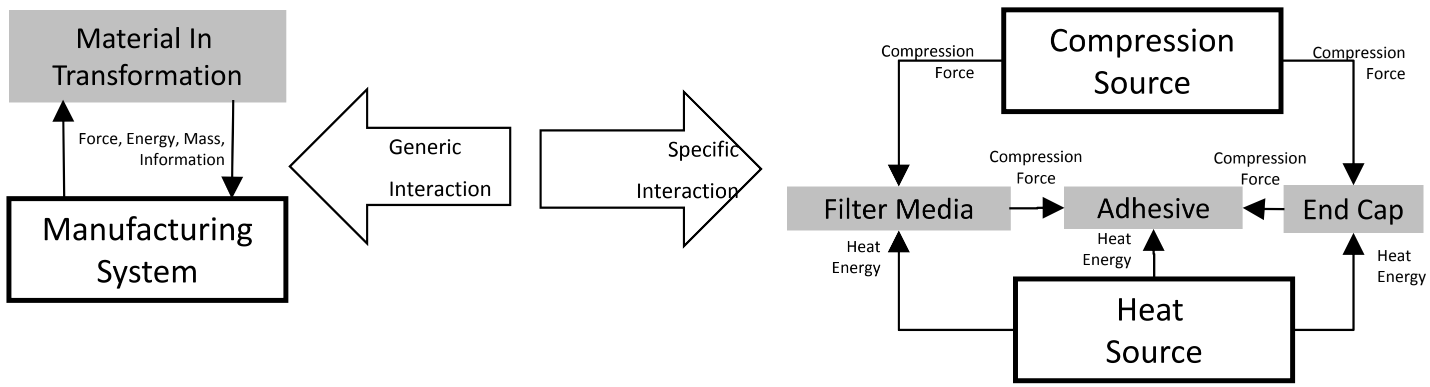
The need for a Science-based Understanding

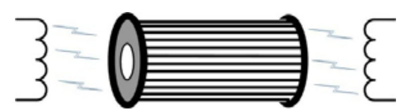
- *Literally everything we know* from the physical sciences is about the *behavior of interacting system components*—whether in chemical reactions, electromagnetics, acoustics, mechanics, thermodynamics, or other discipline-specific interactions:



- Accordingly, the *interactions* of Materials In Transformation with the Manufacturing System assign “roles” to the *Manufacturing System* and the *Materials*, which are required to be met by what we have learned from sciences and by the results we want.

An example Interaction





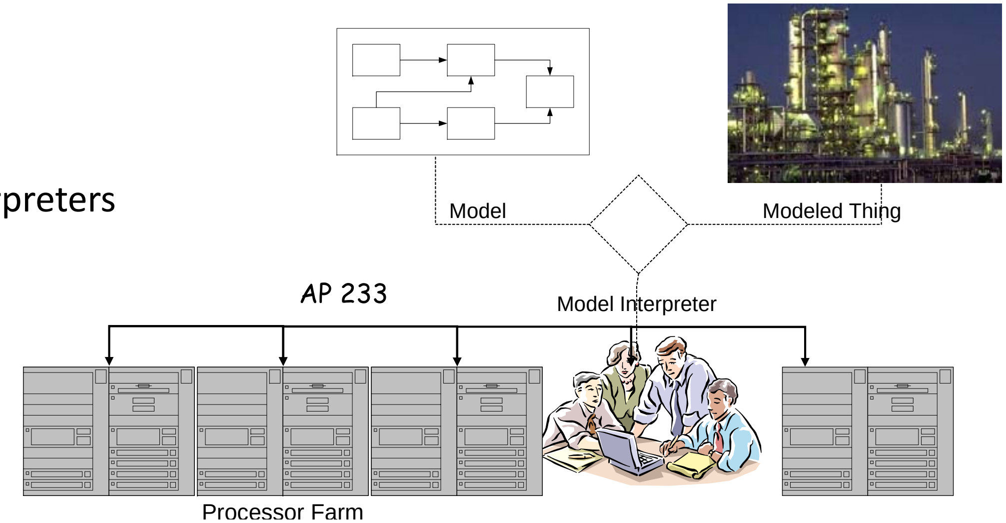
- Interaction = “Bond Filter Media to End Cap”
- Functional Roles (of materials and equipment):
  - Filter Media
  - End Cap
  - Adhesive
  - Heat Source
  - Compression Source

- Each of these “Roles” includes specific Required Behavior in order to meet expectations for the overall Interaction.
- The Physical Component to which the Role is allocated must meet those requirements—whether Equipment, Materials, or People

Model-Based Systems Engineering (MBSE)

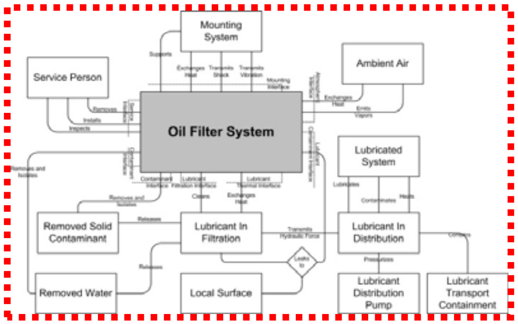
- Model-based systems engineering is an emerging approach to systems engineering:
  - See www.incose.org
- Uses explicit models where previously informal, intuitive, natural language prose (e.g., English) of documents was used

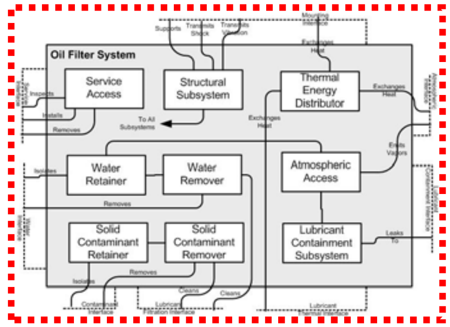
- Not all model interpreters need be human

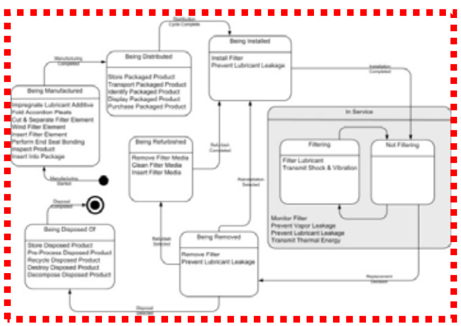


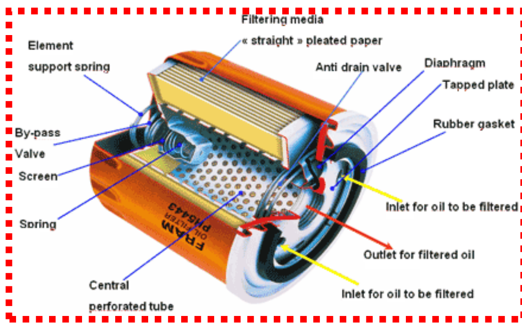
Assumed MBSE background we'll need

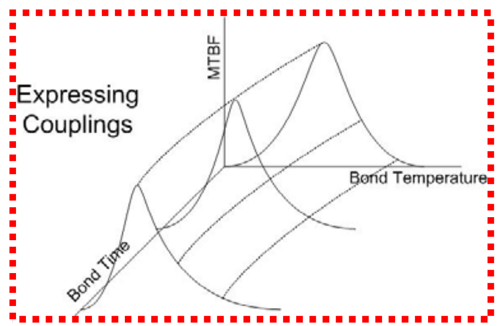
There is a growing practice and literature on Model-Based Systems Engineering.







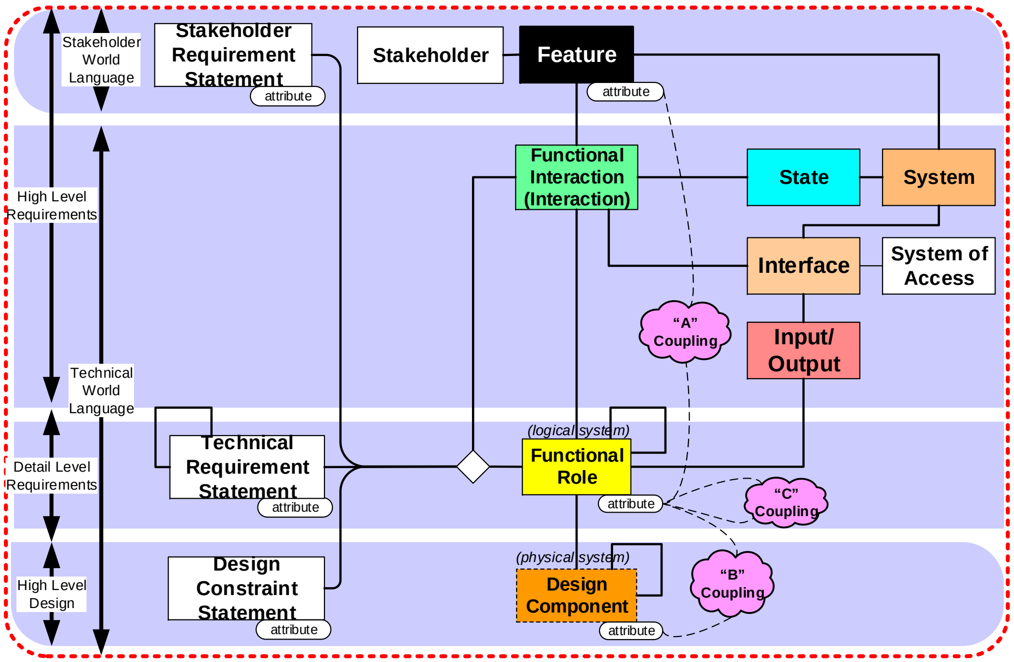




Systematica approach to MBSE

- Uses models ("blueprints") instead of prose, to specify requirements and design of complex systems (product systems, manufacturing systems, operations processes, the engineering process, etc.).
- Increases understanding while lowering costs.
- Establishes a common language and data model for all systems engineering , across people, tools, information systems—for leadership as well as technologists.
- Expresses model-based formal industry standard (e.g., ISA) descriptions of systems.
- Uses S*Metamodel to express underlying concepts.

Simple summary of detailed Systematica Metamodel.

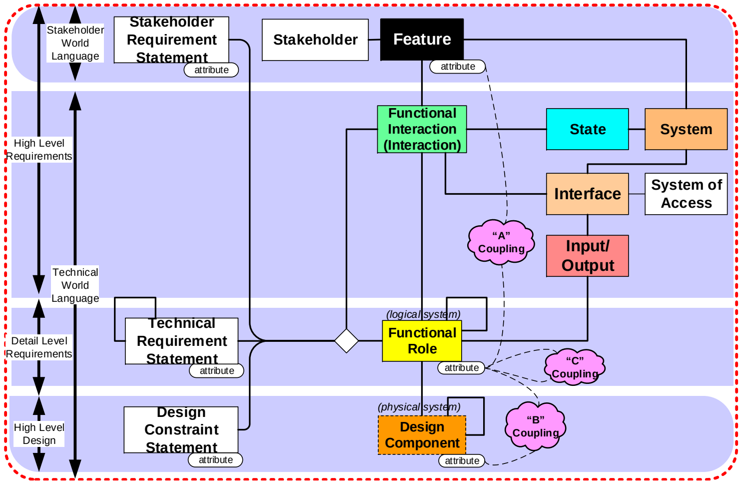


Model-based systems engineering MBSE)

What does Systematica mean by “Metamodel”?

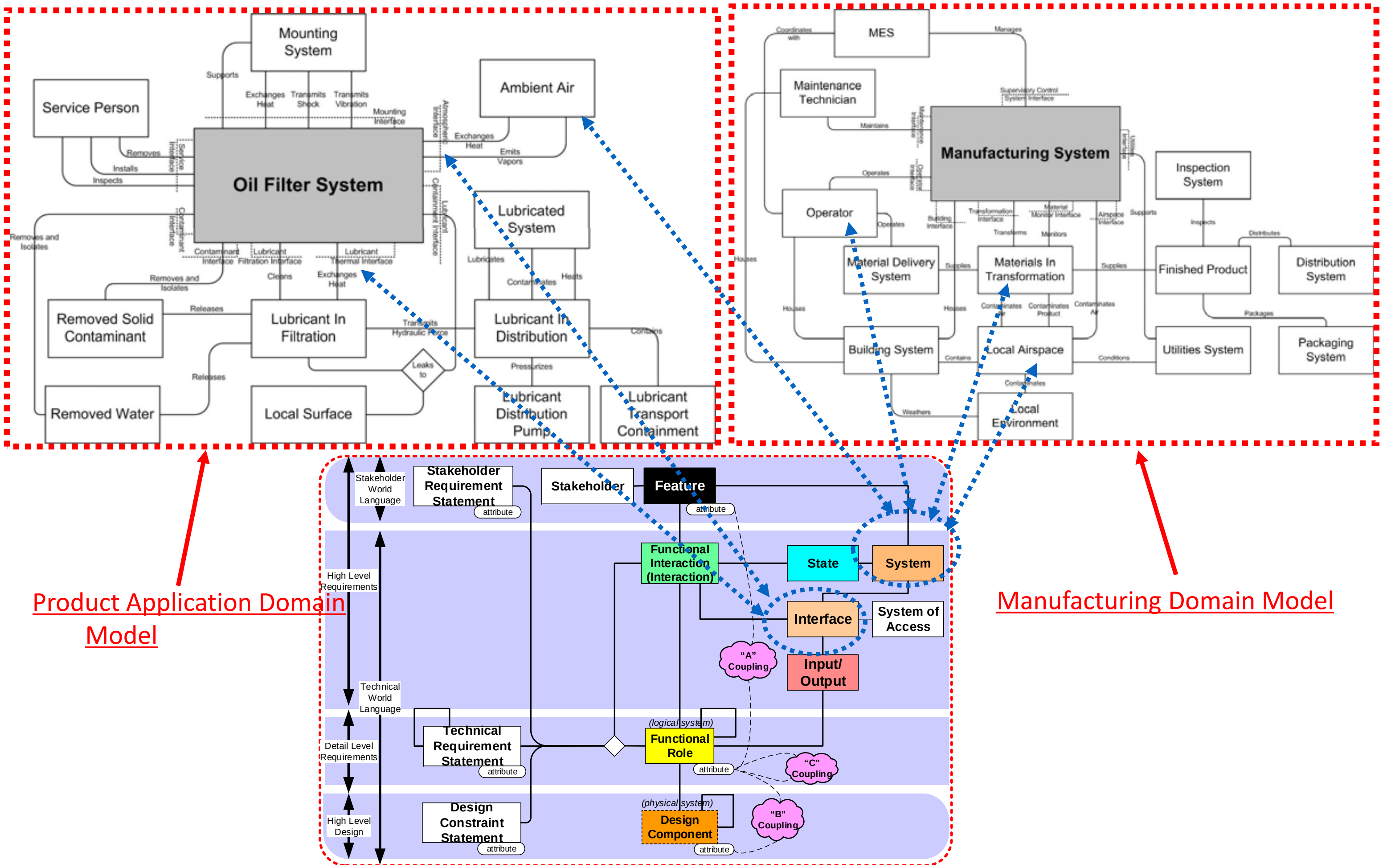
- The framework in which all models are described
- *The minimum set of ideas necessary to express all concepts of system requirements and design, independent of technology*
- The overall model to which any system model must conform
- Constrains community to a common framework, across technologies and functions

- Within this framework, we create an Enterprise or Industry Language for Shared Patterns, to consistently express system requirements, designs, validations and verifications, FMEAs, etc.
- Incorporating industry, enterprise, governmental standards as needed



Simple summary of detailed Systematica Metamodel.

Models can describe Manufacturing Systems, as well as Manufactured Products.

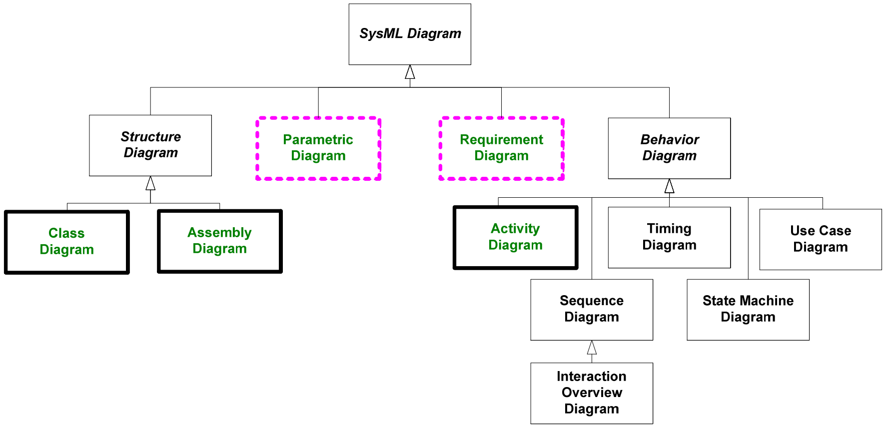


Product Application Domain Model

Manufacturing Domain Model

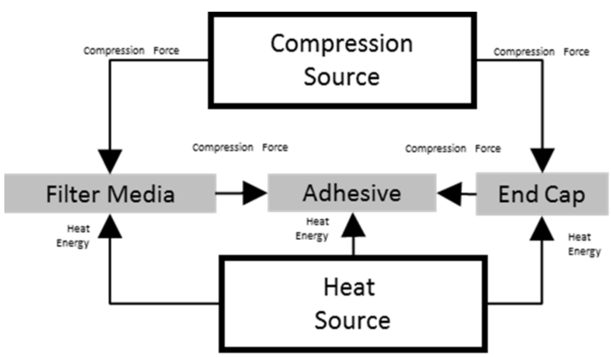
Assumed MBSE background we'll need

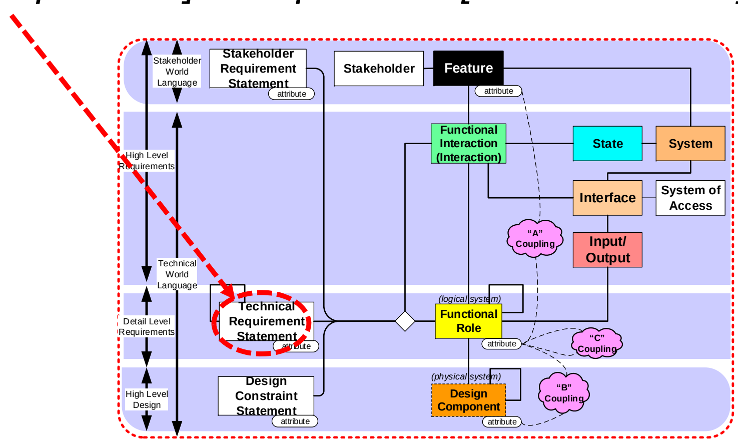
- Model-based methods supplement the use of natural language prose in traditional engineering documents with the use of "models" which are explicit data structures (typically relational tables and formal diagrams).
- The structure of these models can be exploited to create analyses and checks that would be much more difficult and subjective to perform using purely prose-based methods.
- When applied well, they can also more effectively convey shared meaning to human readers.
- We will focus here on how Manufacturing Transformations can be more deeply integrated as a part of such MBSE models.
- See the attached example for other aspects.



Modeling transformation behavior

- This Metamodel re-positions prose functional “Requirements Statements”:
  - These textual statements become a formal part of the model.
  - All functional requirements are modeled as external interaction behaviors.
  - They become input-output relationships describing external system “black box” behavior during Interactions with external actors—a “prose transfer function”:
    - *“The Manufacturing System shall deliver to the Materials In Process a <u>Compression Force</u> of [Min Bond Force] for a period of [Min Bond Time]”.*
    - *“The Manufacturing System shall deliver to the Materials in Process <u>Heat Energy</u> sufficient to maintain a bond temperature of [Min Bond Temperature] for a period of [Min Bond Time].”*

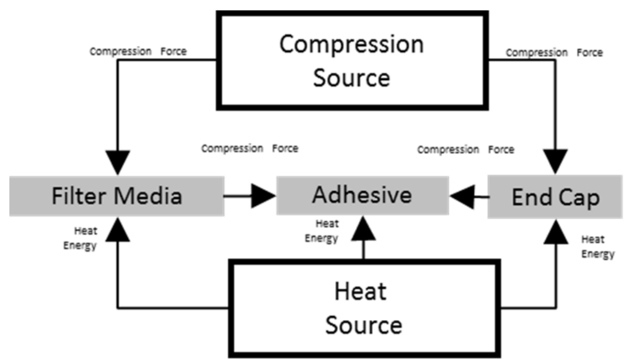


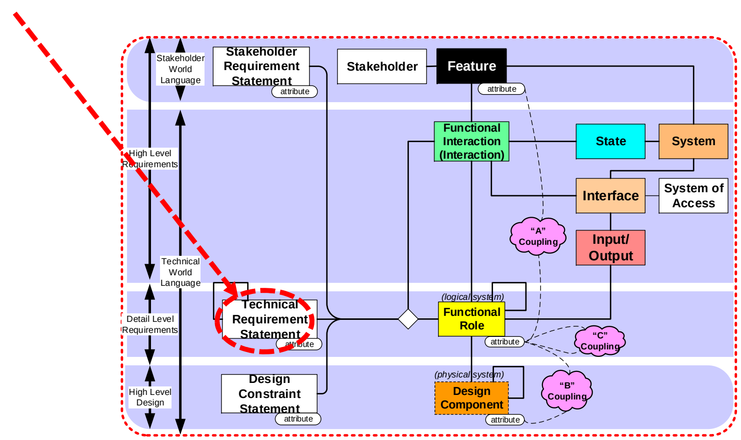


- Further described in (IS 2005).

It works for the [Materials in Process], as well as the Manufacturing System

- In the same way, in the same model we can describe the required behavior of the Materials in Process:
  - *"The Adhesive, Filter Media, and End Cap shall bond upon input of a Compression Force of [Min Bond Force] for a period of [Min Bond Time], accompanied by input of Heat Energy sufficient to maintain a bond temperature of [Min Bond Temperature] for a period of [Min Bond Time]."*
  - *"The Oil Filter shall operate in service at Lubricant Pressure of [Max Lubricant Pressure] with bond or other structural failure rates less than [Max Structural Failure Rate] over an in-service life of [Min Service Life]."*

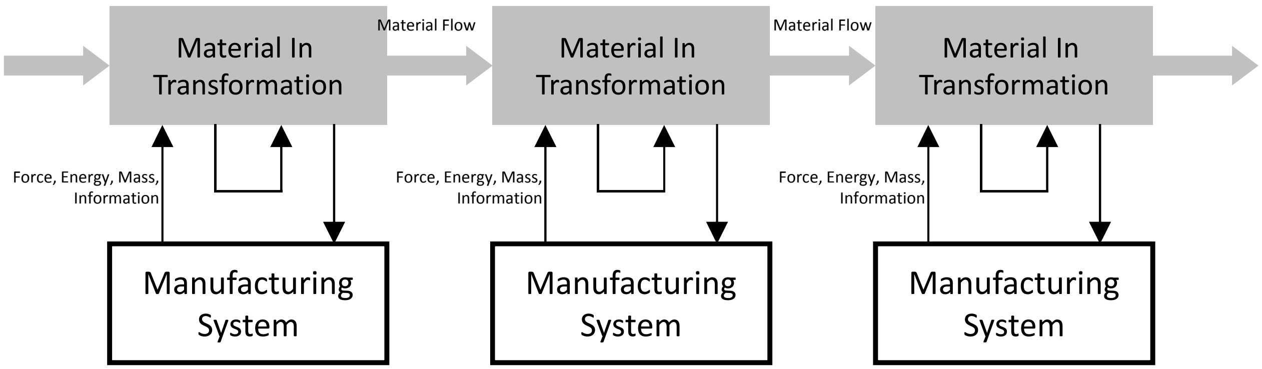




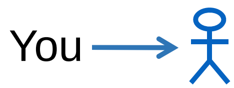
- Further described in (IS2005).

Applying the concepts to manufacturing processes

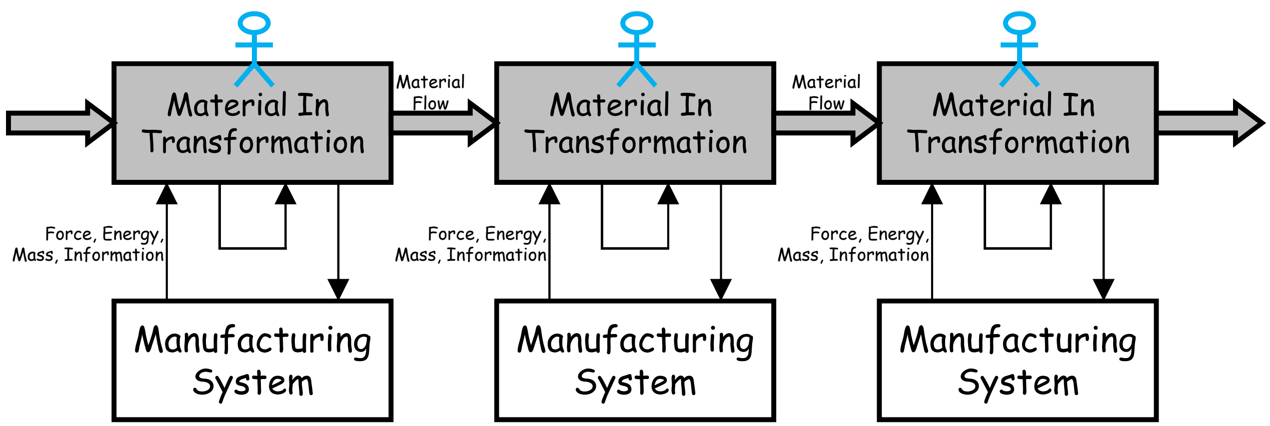
- For some process engineering specialists, material scientists, or other disciplines, an understanding of the behavior of the material during transformations is essential:
  - bending, forming, structural deformations, cutting, milling, extruding, compression
  - chemical, biochemical, electrochemical reactions, distillation, fermentation, etc.
  - heating, cooling, bonding, welding, fastening, mixing, blending
  - other transformations
- These specialists think about the “Material In Transformation”:
  - how the material behaves <u>during</u> each of a series of sequential unit operation transformations;
  - During each transformation, the Material will exchange <u>energy</u>, <u>force</u>, <u>mass</u>, or <u>information</u> with the Manufacturing System, as well as with itself--



Process: What the Material "Sees"

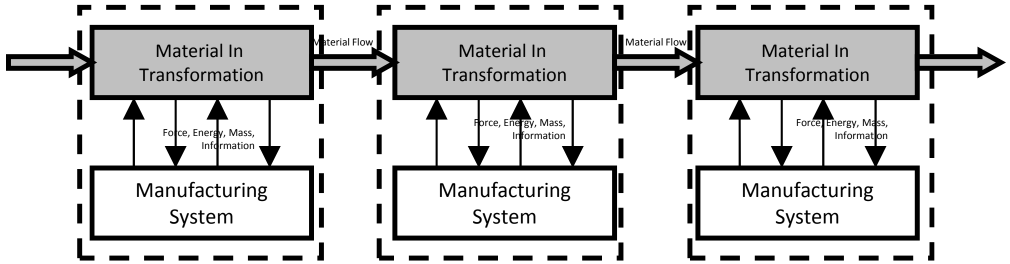


- Imagine that <u>you</u> could "ride through the process with the material".
- Imagine that <u>you</u> could "see what the material sees" (forces, temperatures, etc.).
- This is the "process view" of the process engineer, materials scientist, chemist, metallurgist, or other process-related specialist:

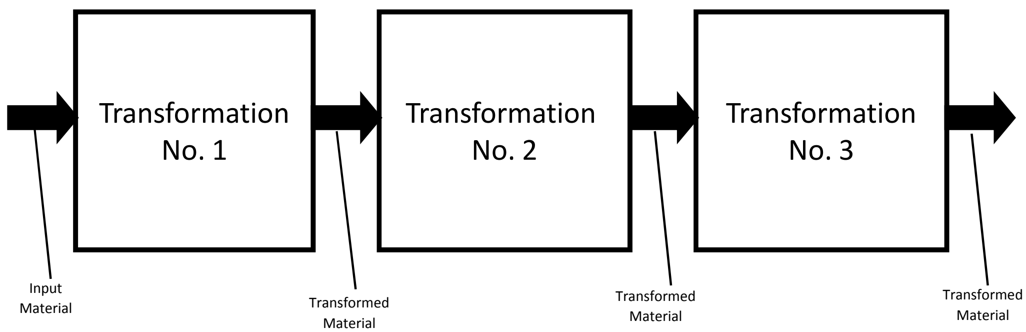


Less detailed PFD views

- Others people’s jobs don’t need that much detail, so they think of the transformations as “black boxes”; so that . . . .

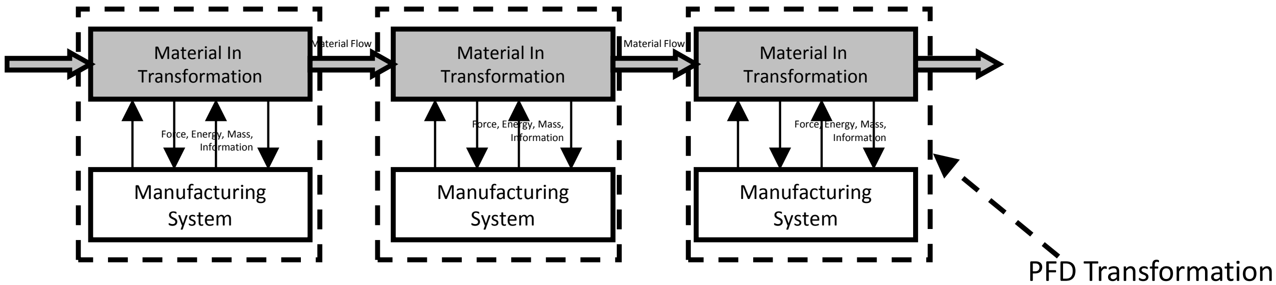


becomes a Process Flow Diagram (PFD):



Material In Transformation can be modeled as "logically outside" the equipment's transformation role

- Difference between these two representations:
  - the Material In Transformation is "logically outside" the Manufacturing System, but . . .
  - that Material In Transformation is "logically inside" the PFD Transformations:



- After all, the Material In Transformation is not a part of the BOM of the Manufacturing System!
- The advantage of this approach is that it allows us to use the MBSE technique that *all the functional requirements on the manufacturing system are found at the points of input-output boundary crossings of that system*

“Registered Process” As Requirements

- Many manufacturing “processes” have a kind of managed existence separate from their specific implementation with equipment:
  - When a PFD describes a process before there is equipment design;
  - When a “registered process” has been approved by a regulator, and a factory is constructed to implement that specific process;
  - When a low-volume process has come out of a laboratory to a pilot production line, but not yet been scaled up to production volume.

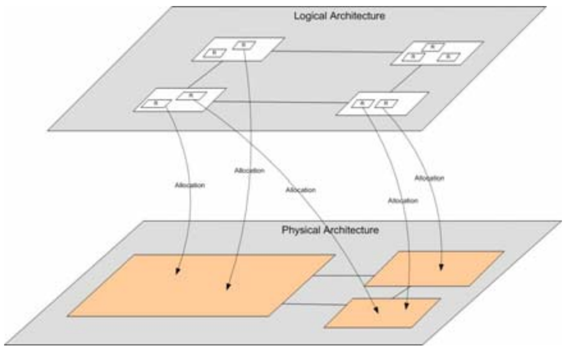
- This reflects the idea that the requirements of a manufacturing system are something more than producing the end outputs from the initial inputs—it is also expected to embody a specific targeted manufacturing process.

- This is why we model the “Materials In Process” as an external actor interacting with the equipment.

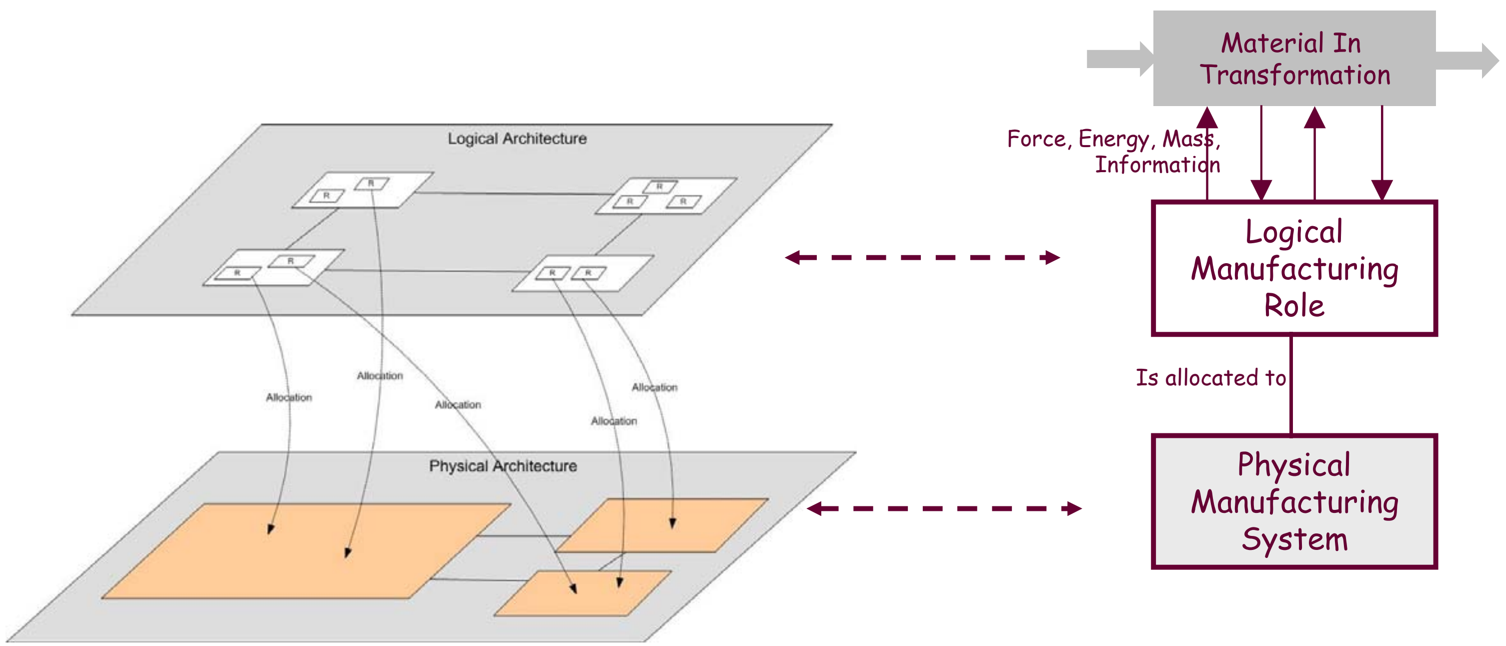
Logical Systems vs. Physical Systems

- MBSE expresses what the Manufacturing System contributes to the process, using Logical Systems:
  - Logical systems are defined by their required externally visible behavior, as seen by the other interacting actors, without regard to the physical design used to accomplish that behavior.

- Logical System Roles:
  - represent transformation or other behavior of the manufacturing system, without regard to its design.
  - Certain Logical Manufacturing Roles must produce (or consume) certain forces, energy, or information, exchanged with the Material In Transformation.
- Physical Manufacturing Systems:
  - Are defined by their physical identity, not their behavior.
  - Logical behaviors are then allocated to physical equipment.
- Logical Roles are allocated to Physical Systems

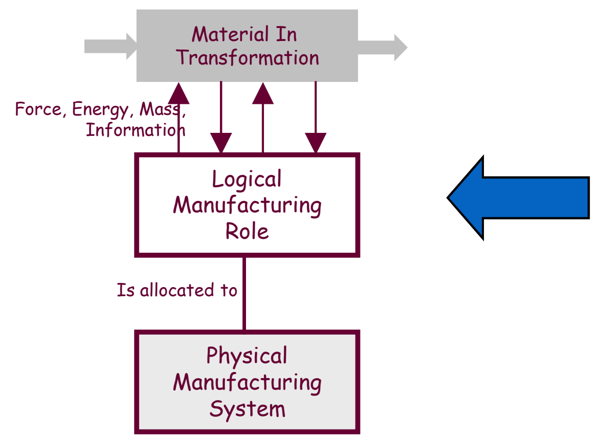


Logical Systems vs. Physical Systems



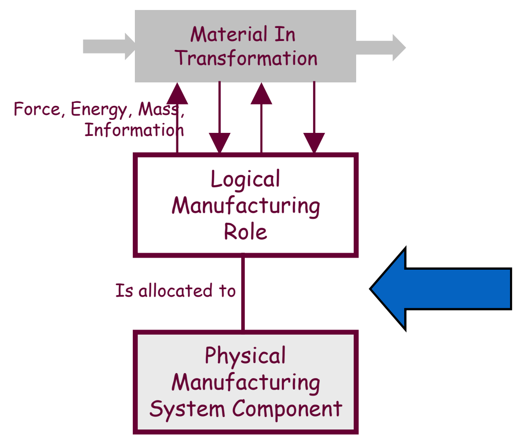
Manufacturing system requirements

- The input-output relationships (relationships between input-output Forces, Energies, Masses, Information that are exchanged with the Material In Transformation) of the Logical Manufacturing Roles turn out to express the requirements allocated to the Manufacturing System to accomplish the transformation:



Manufacturing equipment design

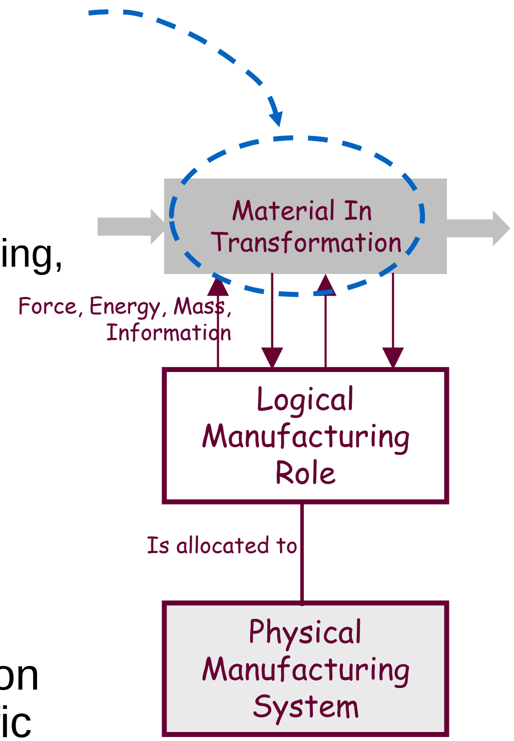
- The allocation of logical manufacturing roles to physical equipment components describes the high level design of the manufacturing system:



➢ This begins the embedding of process requirements into an integrated framework of system requirements.

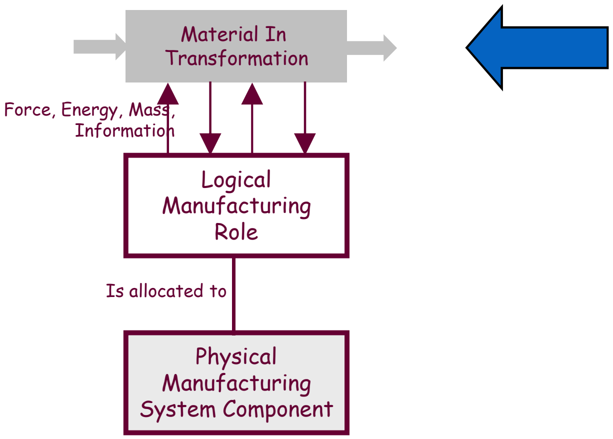
Materials roles

- For materials scientists, chemists, metallurgists, and other specialists in materials . . .
- These specialists seek out materials that have properties desirable for transformations:
  - bending, forming, structural deformations, cutting, milling, extruding, compression
  - chemical, biochemical, electrochemical reactions, distillation, fermentation, etc.
  - heating, cooling
  - bonding, welding, fastening
  - mixing, blending
  - other transformations
- The logical transformation model facilitates description of those properties, somewhat independent of specific materials:
  - Encourages understanding of materials requirements and opens thinking to new materials solutions.

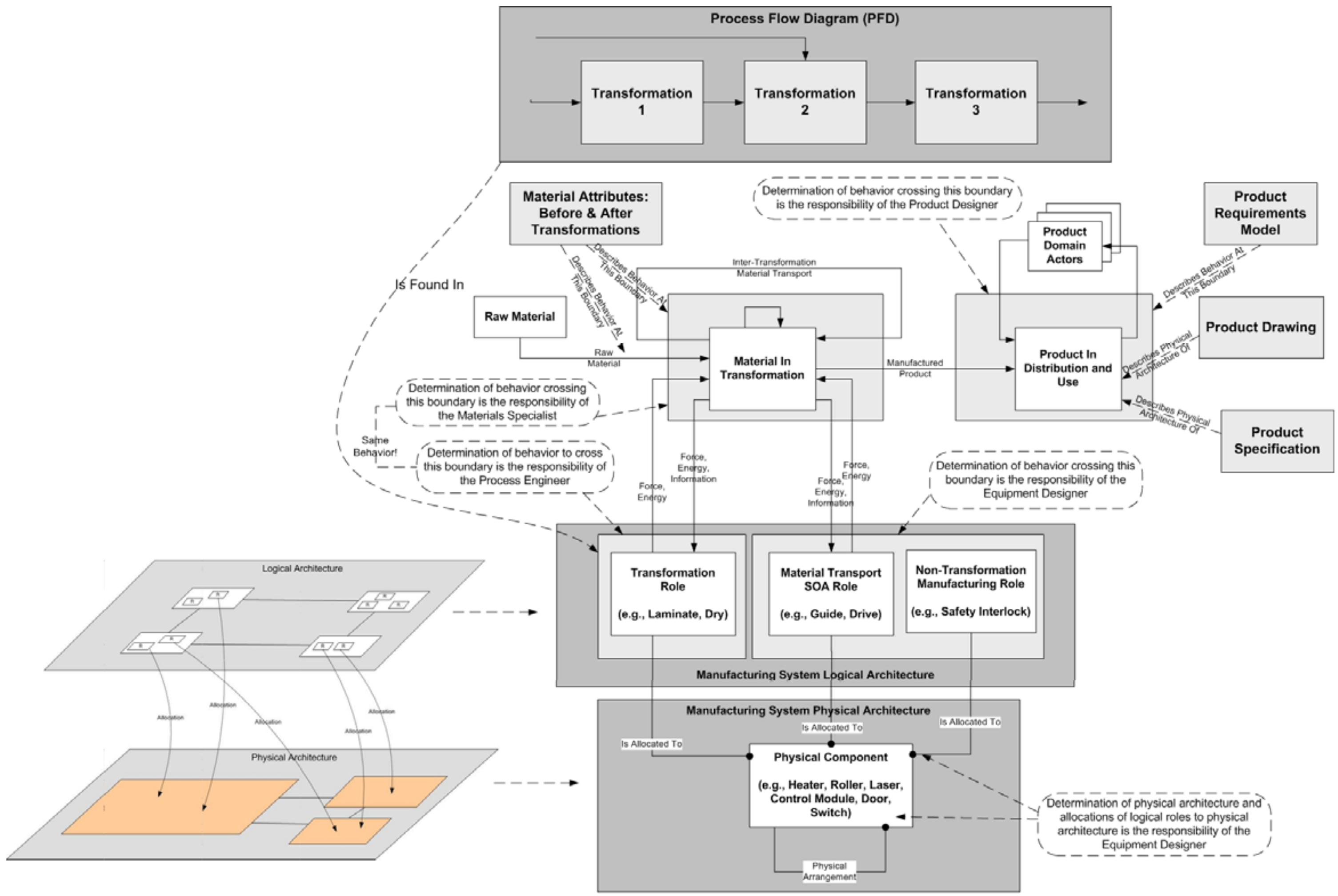


Materials roles

- Just like the equipment, logical roles are allocated to the Materials In Transformation, which they must satisfy in order for the transformation (or transport) to succeed:



- This means that we can create an integrated model that couples the roles of interest to the process engineer and equipment design with those of interest to the materials specialist . . . .



Logical and Physical Architecture, Cross-Domain Development & Engineering Roles

Conclusions

Applying this PBSE approach to manufacturing systems helps:

1. Integrate science-based understanding of processes, materials, and transformations into the life cycle engineering of manufacturing systems.
2. Improve integration of Process Engineering with other engineering disciplines.
3. Improve manufacturing process IP capture—particularly using PBSE.
4. Improve teams' and individuals' abilities to "think outside the box".
5. Speed discovery of new product and process implications for equipment design.
6. Improve understanding of newer references and standards for describing manufacturing processes that use the language of "models".
7. Improve the ability to perform long-range planning and portfolio management of manufacturing technologies, along with related product science and technologies.
8. Organize patterns of re-usable IP for processes, materials, technology, and design.

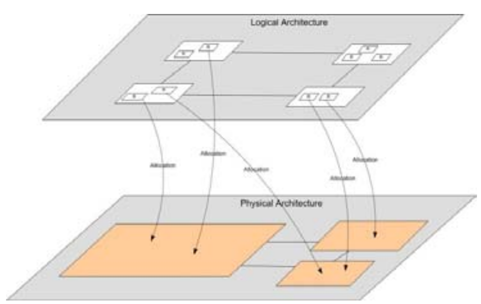
Additional information

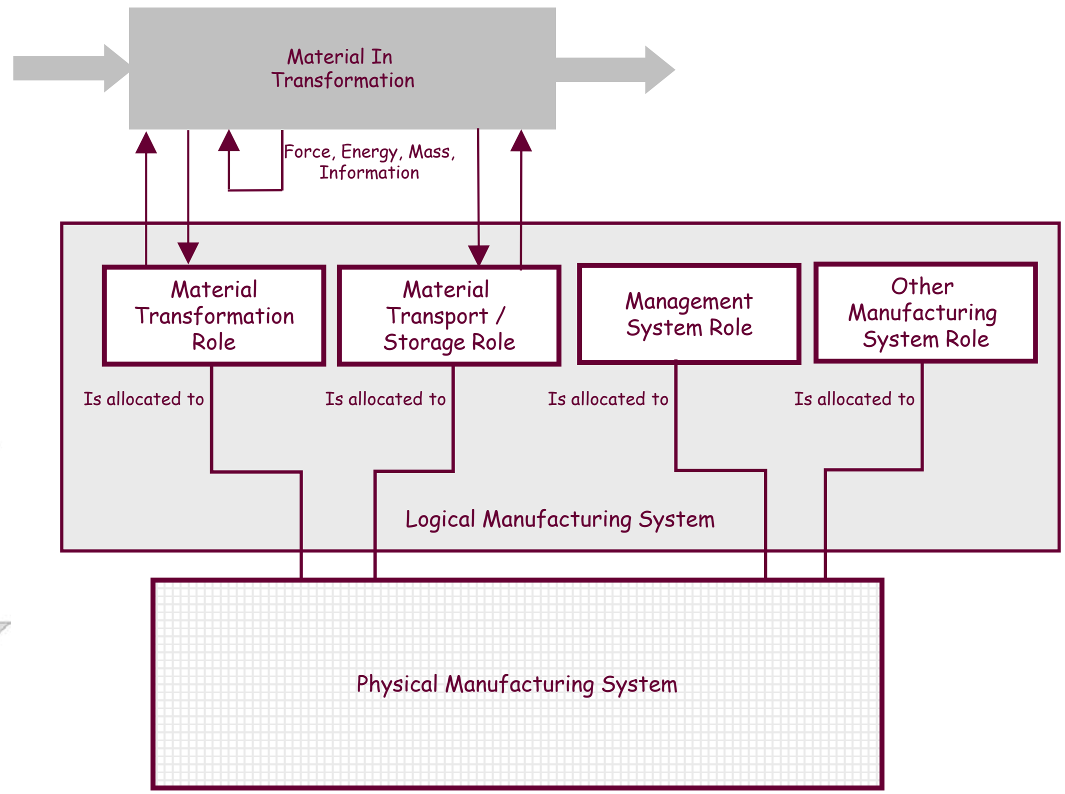
- Non-transformation manufacturing roles
- Manufacturing patterns, parameterized recipes
- Unit operations vs. higher level systems
- Portfolio management
- An extended example

Non-transformation manufacturing roles

- There are additional logical roles that the Manufacturing System must perform, beyond physical transformations.
- For example:
  - Transport and storage roles;
  - Material systems of access (interface) roles;
  - Infrastructure roles (utilities, etc.);
  - Management: Operations, maintenance, configuration, security, accounting roles

Non-transformation roles





Transport and storage roles

- Requirements on the manufacturing system for:
  - Transport (movement of material in process)
    - Liquid transport
    - Web transport
    - Powder, solid materials, gaseous transport
    - Logistics considerations, carriers, space, etc.
  - Storage
    - Roles typically filled by tanks, warehouses, shelves, etc.

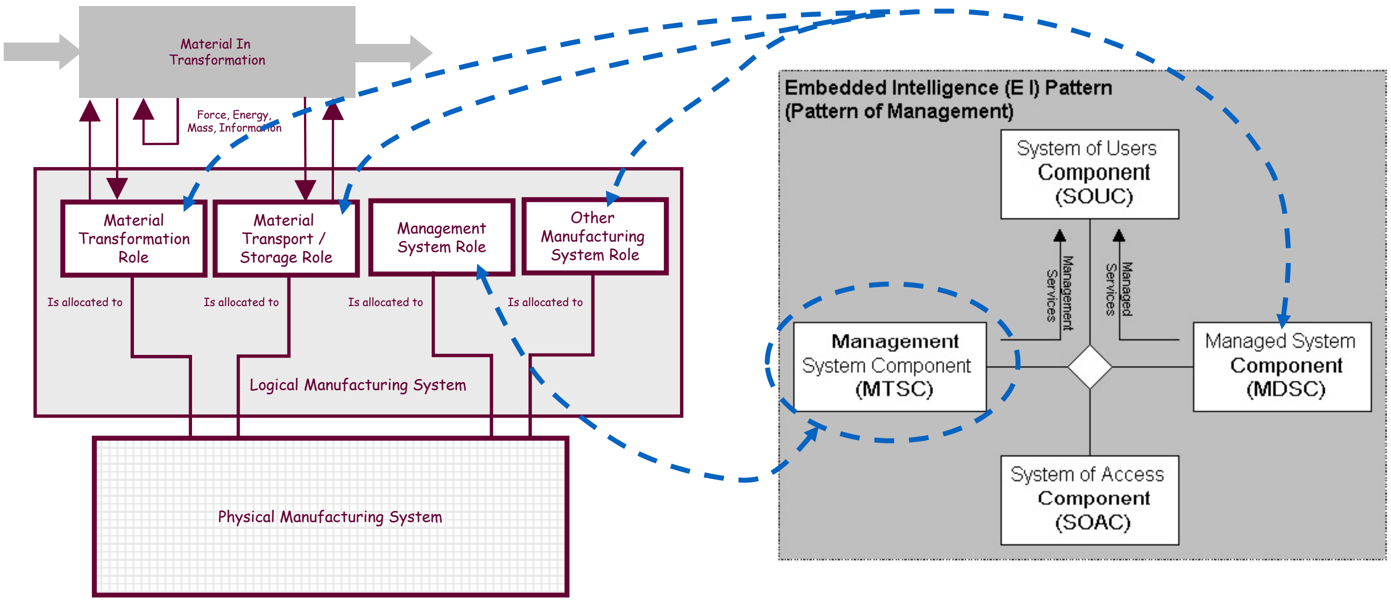
Material Systems of Access (SOA) Roles

- A System Of Access is part of an Interface Model—the system that enables physical interaction between two other systems.
- SOAs are important “glue” for practical engineering as well as scientific understanding of system interactions.
- Two SOA classes important to Process Engineering models:
  - Transformation Systems of Access--
    - Example: the logical roles played during material transformations, by heated tank jackets (heat transfer) or bubbling gas through liquids (maximize contact area), etc.
  - Transport Systems of Access—
    - Example: the logical roles played during material transport, by slurry pumps, conveyer belts, augers, rolling bins, etc.
- Separating SOAs in the model improves the ability of the underlying transformation and transport processes to be modeled independent of technology.

Infrastructure (utilities, infrastructure, etc.) roles

- Regular utilities (electrical & pneumatic power, heating & cooling media, etc.)
- HVAC
- Clean or specialized utilities
- Consumables treated as utilities
- Waste disposal, treatment, co-generation, or recovery streams
- Plant space, structural resources
- Site resources

Management roles: Operations, maintenance, configuration, security, accounting



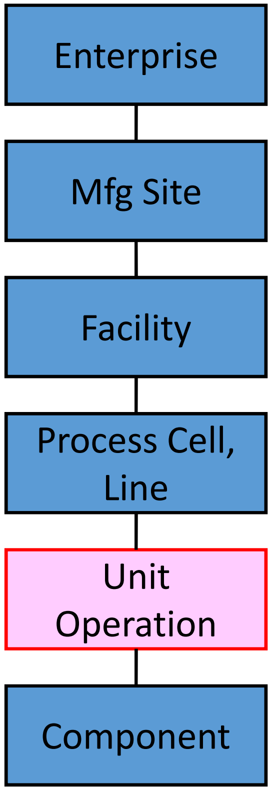
- Electronic controls and automation are “management system roles” that are part of the model.
- These roles are also played by humans (operators, etc.).
- They are usually organized into hierarchical controls patterns:
- For more on this, consult the Systematica materials on Embedded Intelligence (EI) Pattern of Intelligence-Based Systems Engineering (IBSE).

Manufacturing patterns, parameterized recipes

- MBSE “models” describe both requirements and design, for both equipment and materials;
- PBSE “patterns” are re-usable Models, requiring less effort to use than creating Models from scratch;
- Patterns can be configured for different needs and uses:
  - One reason to configure a general pattern is to describe a site specific system (e.g., a manufacturing system installed at a site).
    - A single configured system of this type might still be capable of carrying out many different recipes.
    - This type of configuration is “configuration at design time”.
  - Another reason to configure a pattern is to express a specific recipe:
    - This has the effect of configuring a site specific system for a single recipe, and is a “configuration at run time”
  - For more on this, see the Pattern Configuration Process, ISA S.88, etc.

Unit ops vs. higher level aggregations

- The “materials in transformation” approach to modeling particularly applies to the Unit Operation level, where the transformation occurs;
- There are many other requirements not about transformations, and other hierarchy levels, as well;
- This is all very typical SE hierarchy of decomposable requirements;
- Frequently addressed by multiple disciplines or specialties, and integrated together by SE;
- As usual, it also means that there are attributes (parameters) that are characteristics of the different levels—some are lower level process attributes, but couple to higher level product Quality, Capacity, Yield, Cost, or other critical attributes;
- MBSE attribute coupling models help to make the relationships between these attributes more evident—typically these couplings are characterized by DOE studies, first principles, process characterization, and other sources.



Exercise 5: Applying the Manufacturing System Pattern

1. What is new, changing, or challenging that might drive a need to more effectively model production/manufacturing systems in your or some other enterprise?
2. What types of production material transformations may need more attention? What interactions are involved (equipment-material, material-material, management-equipment-material)?
3. Draw the related Process Flow (transformations) Diagram and its underlying Interaction Diagram.
4. What additional instrumentation or embedding of networking or intelligence in the production process may be occurring, and what challenges to planning and representing this are expected?
5. What are the challenges to the organization or individuals to make this transition?

Capitalization of MBSE Patterns as Financial Assets:
How to shift the burden of model cost to the time of use and benefit

- Cost of innovation (development or (otherwise) is a major concern in the strategy and execution of R&D or other advancement.
  - These costs have most frequently been expressed as an expense, subtracting from the current bottom line.
  - The benefits (e.g., increased revenue, etc.) gained from this investment sometimes will not occur until somewhere in the future, making the investment harder.

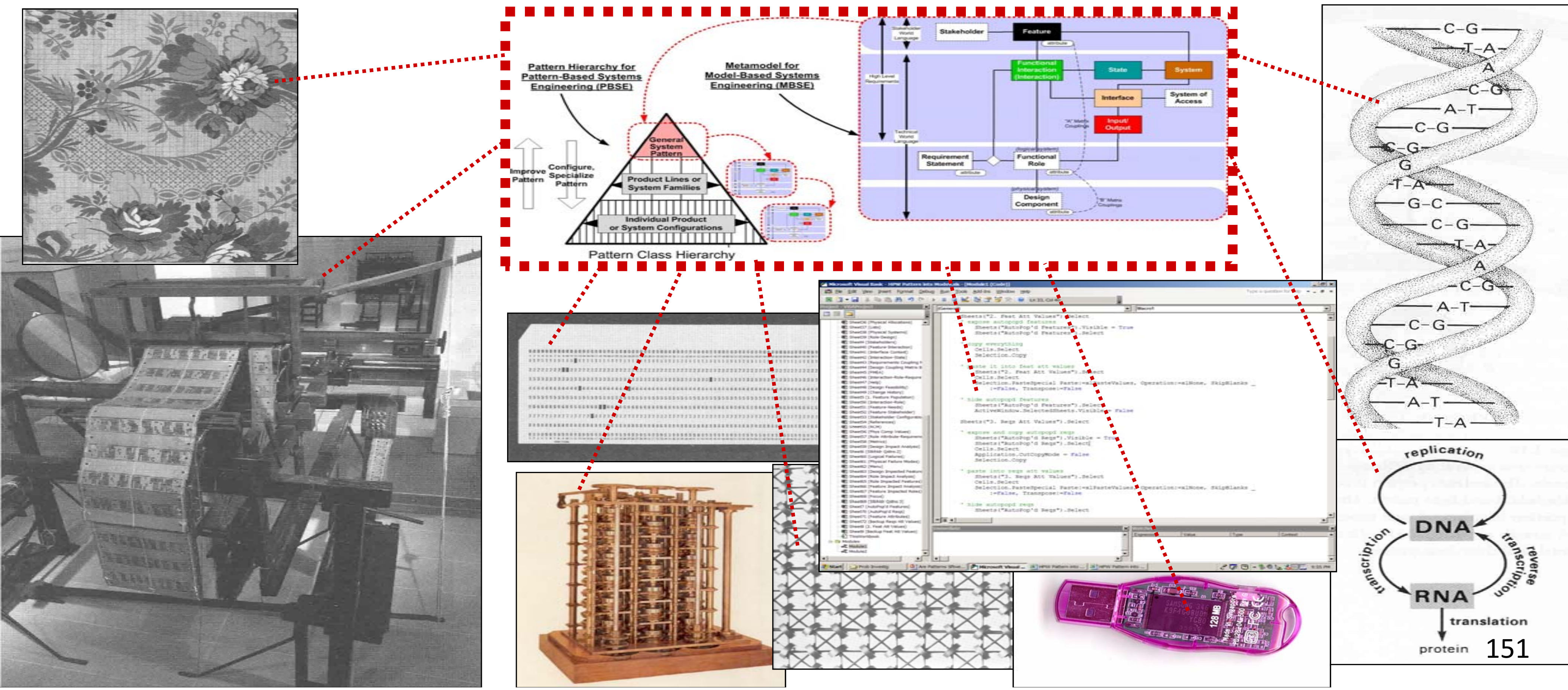
Capitalization of MBSE Patterns as Financial Assets: How to shift the burden of model cost to the time of use and benefit

- In the Construction and Capital Equipment businesses, this situation was addressed many decades ago, through <u>capitalization of assets</u>:
  - Construction or fabrication costs are shown on the balance sheet as creating new (tangible) financial assets—buildings or equipment
  - Those assets are then "expensed" (amortized) over future times, with the incremental amortization generating modest annual expenses, during the years of productive life of the (building or equipment) asset.
  - Those are the years that the asset is producing value (revenue or other benefits)
  - It is a little bit like renting an asset instead of buying it, but all carried out within the same financial statements.

Capitalization of MBSE Patterns as Financial Assets: How to shift the burden of model cost to the time of use and benefit

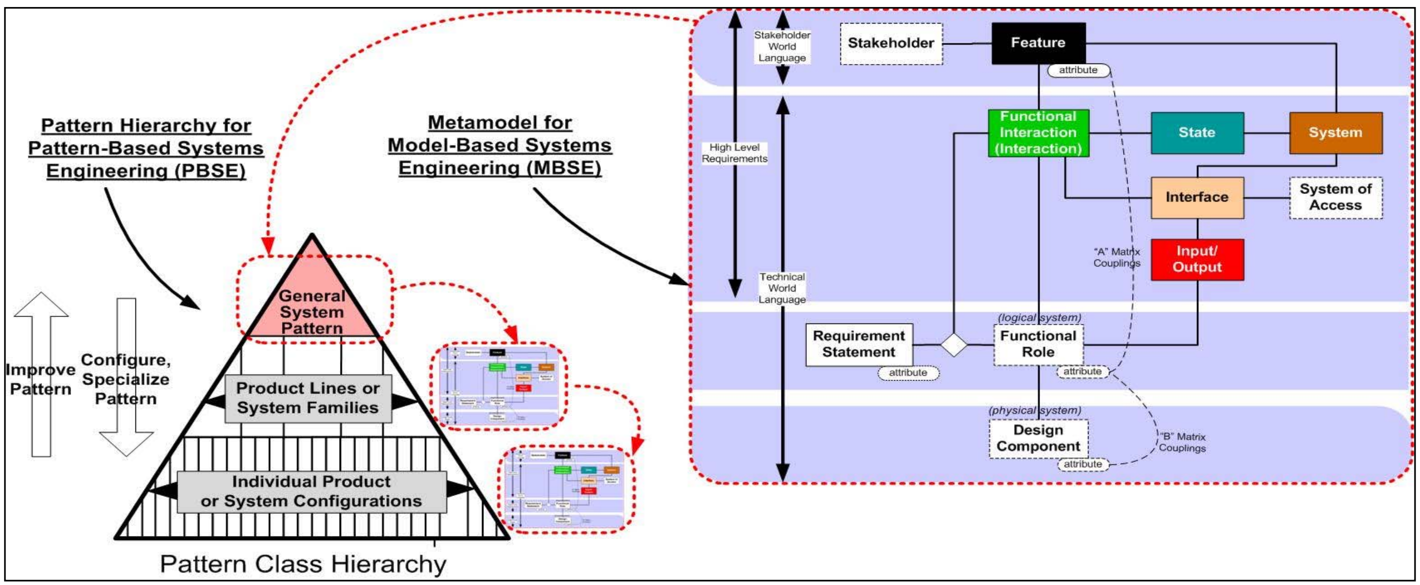
- Over the decades, capital investment in tangible (e.g., bricks and mortar) assets has been outpaced by investment in intangible (e.g., intellectual) assets.
- In the 1980's, this led to the adoption, by the Financial Accounting Standards Board (FASB) of accepted accounting standards for capitalization of <u>computer software.</u> (See FASB86)
- How are MBSE Patterns similar to, or different than, computer software?

The Question: Are MBSE Patterns Software?



What are MBSE Patterns?

- S*Models are *explicit* descriptions of systems:
  - Their Requirements, Design, and other aspects
  - Using data structures as models.
- S*Patterns are re-usable, configurable Models.



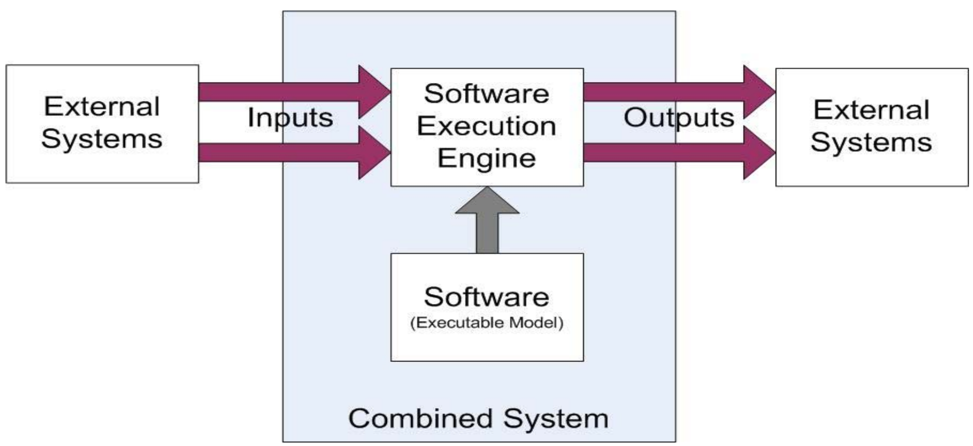
What is Software?

"It cannot be software unless it is written by a computer programmer in ALGOL 68 . . . "

Let's step back and gain a better perspective . . .

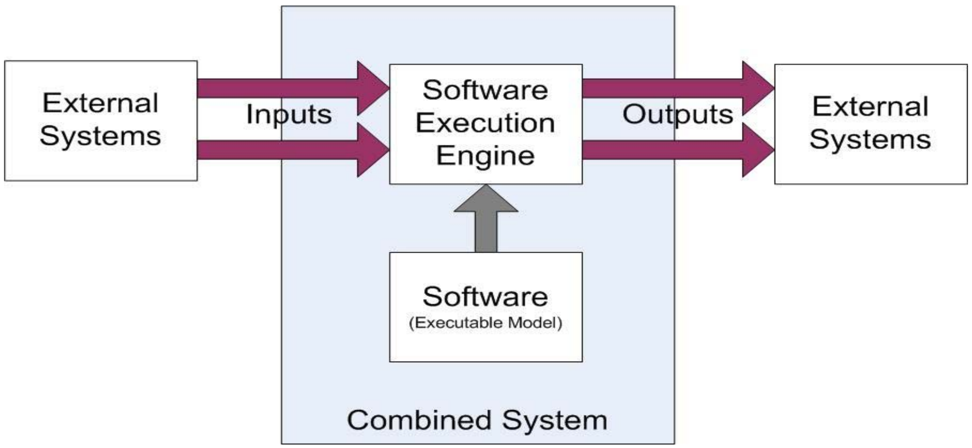
What is Software?

- Software is a special type of information:
  - Software unambiguously specifies the behavior, structure, and other aspects of certain types of systems.
  - Software is always “paired” with something that can interpret, or “execute”, the software.
    - Most typically -- a “Computer”
  - So, software is an executable (interpretable) model.



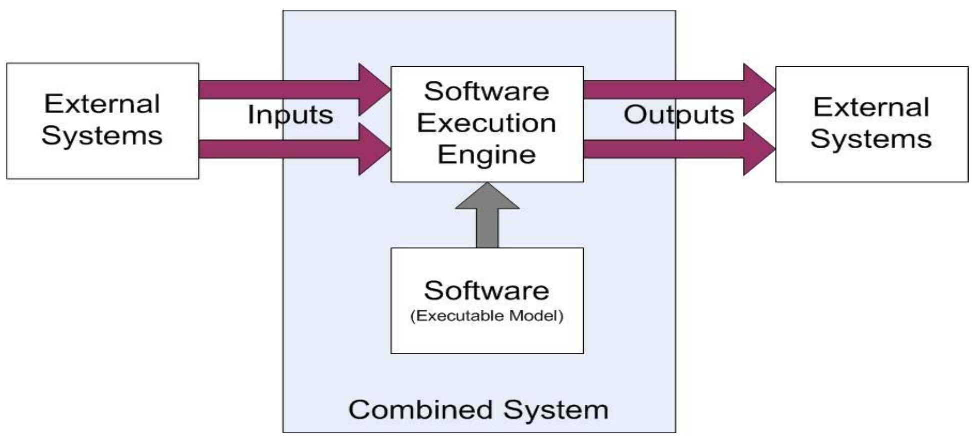
What is Software?

- The “execution engines” that interpret software transform Inputs into Outputs, under control of the Software:
  - These Inputs and Outputs can be Information, Mass, Force, or Energy.



What is Software?

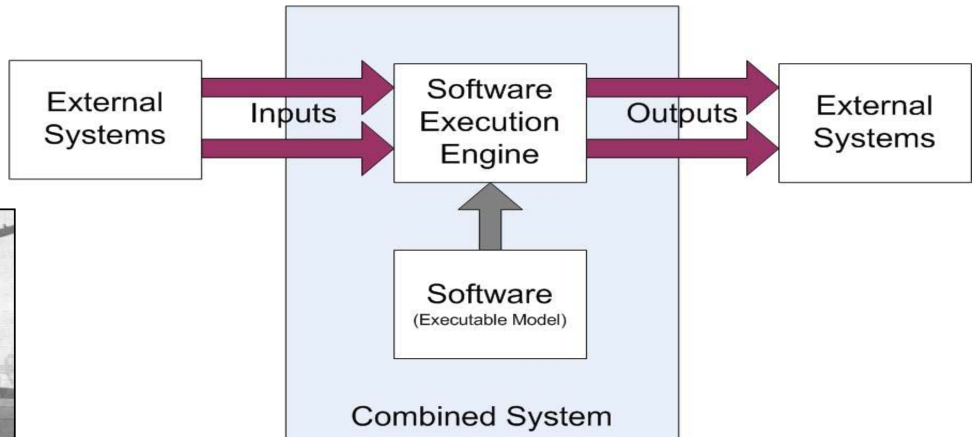
- The most familiar thing that can execute software is a “General Purpose Computer”.
  - But, it is not the only thing that can execute a model
  - And, there are many “data structures” that can represent the model . . . .



Progenitors of “modern” computers . . .

History of the technology tells us

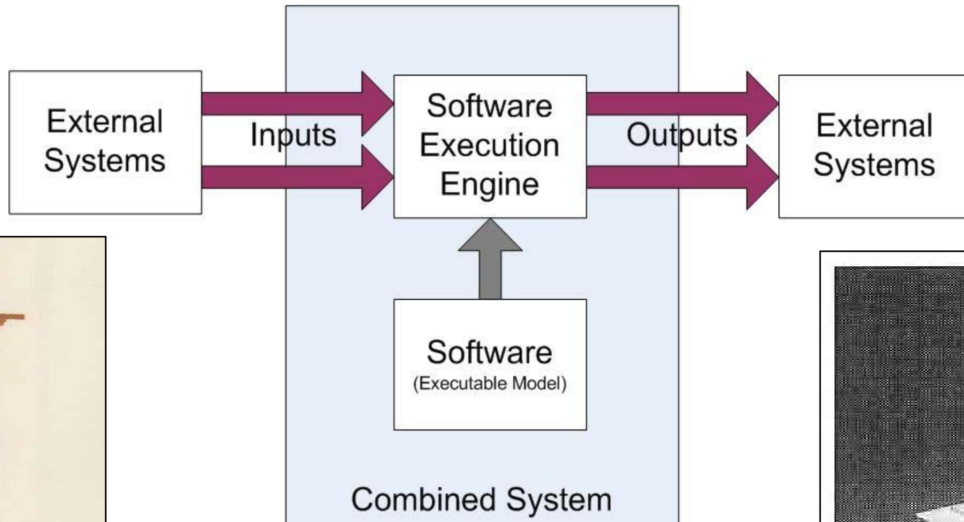
- The Jacquard Loom was programmed with an early version of punched cards to drive its weaving of textile patterns—a revolution in textiles. (1804)

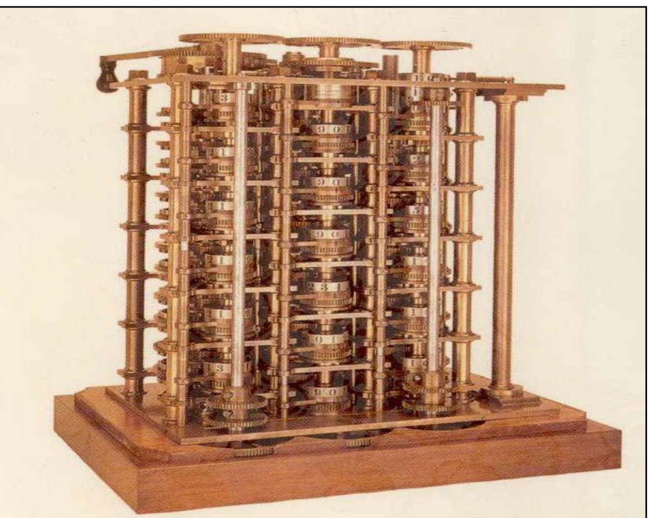


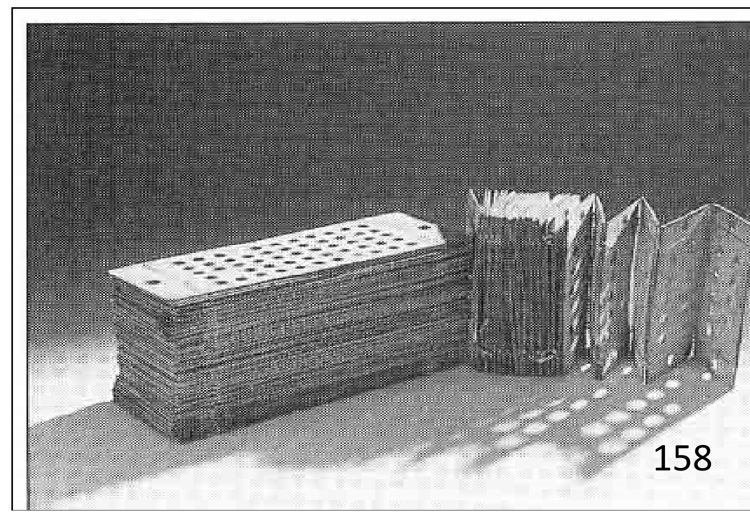
Series of punched cards

History of the technology tells us

- Charles Babbage designed the Difference Engine and the Analytical Engine, programmed by another form of punched cards to drive arithmetic calculations.
 (1821)

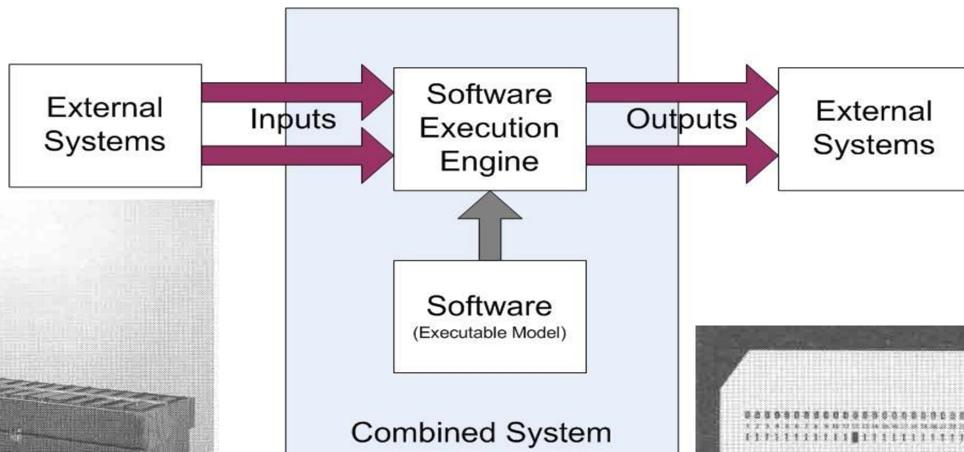


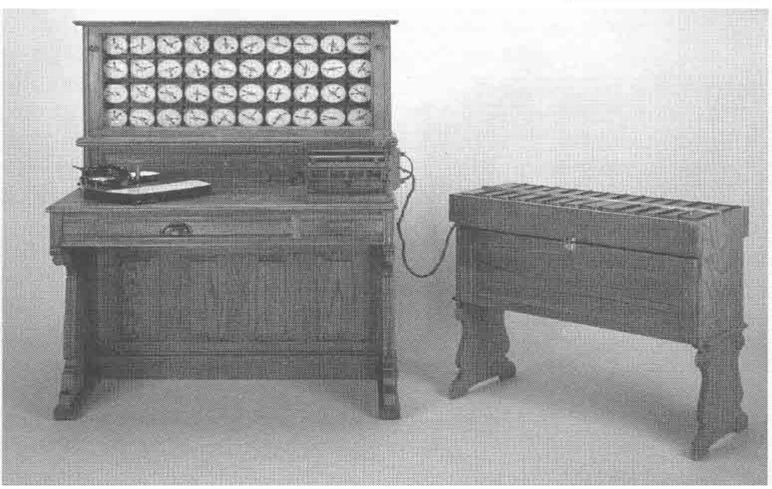


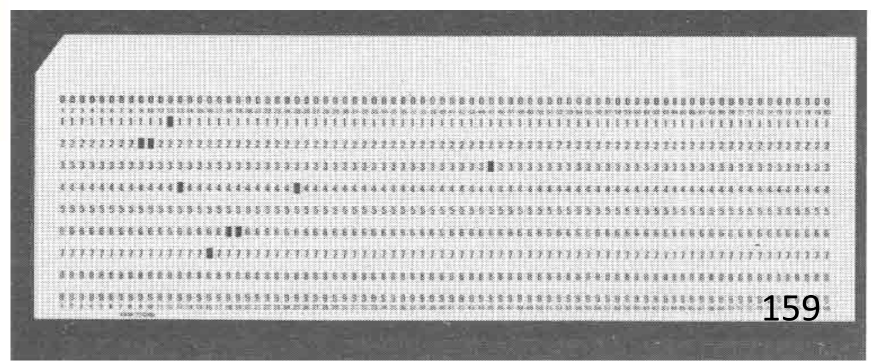


History of the technology tells us

- Herman Hollerith “re-invented” the punched card to develop mechanical and electrical sorters, tabulators, counters for statistical counting, leading to IBM and others.   (1900)
- The “programs” for these machines were typically in wired plugboard information, with the cards used for inputs and outputs.

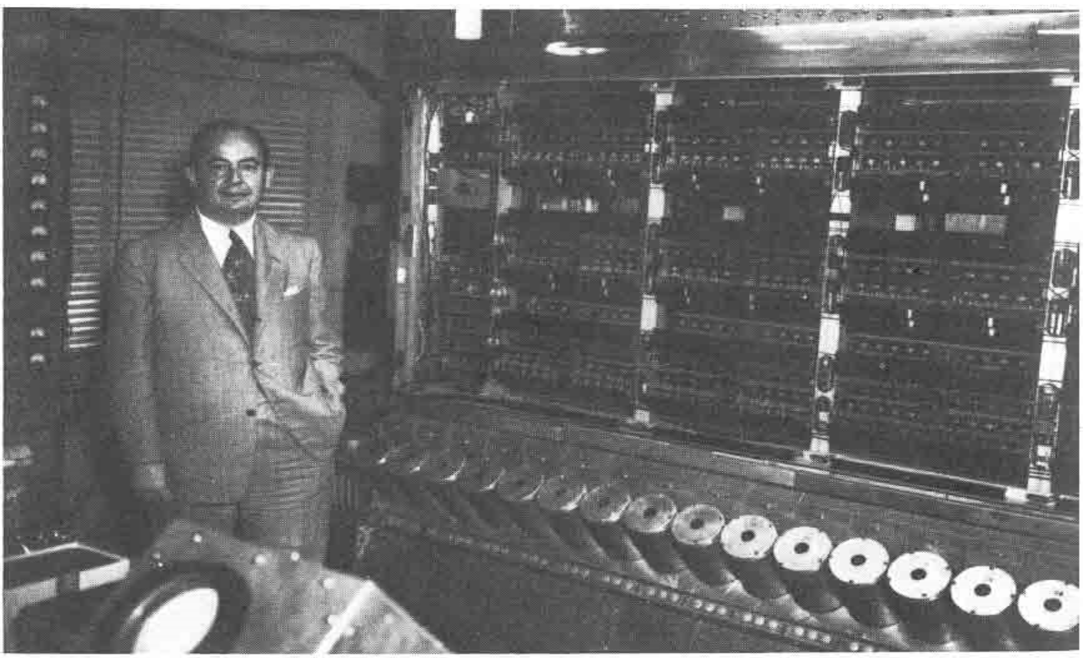


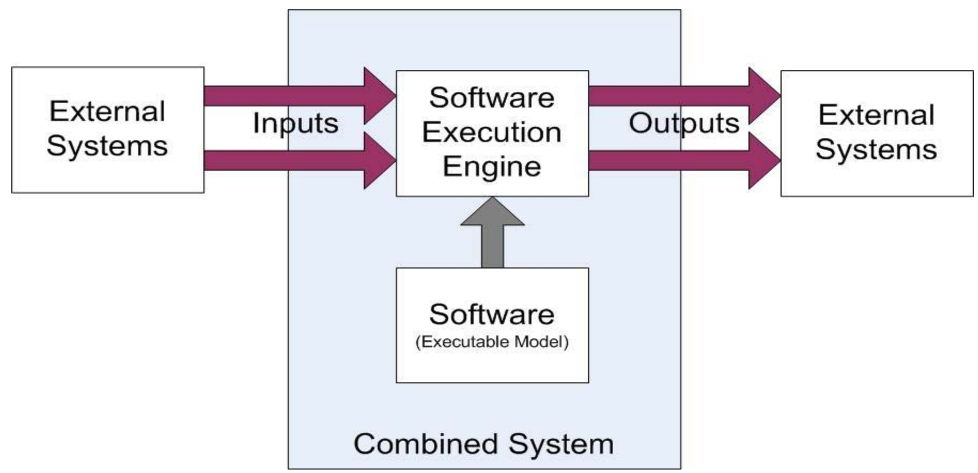




History of the technology tells us

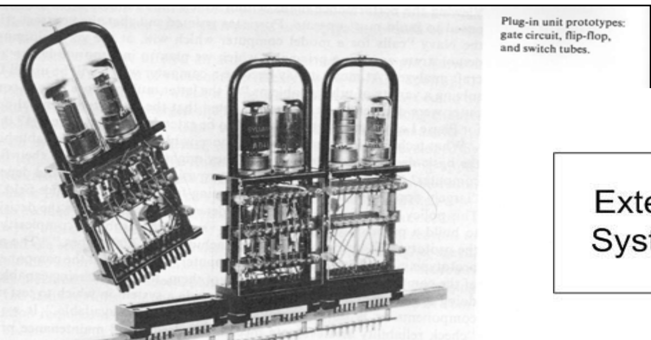
- John von Neumann and others developed the idea of storing the program information as part of the machine's other data—but did not invent the idea of program information, which was much older. (1940s)

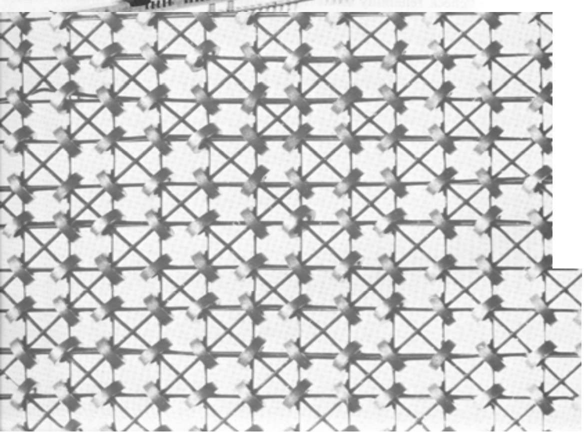


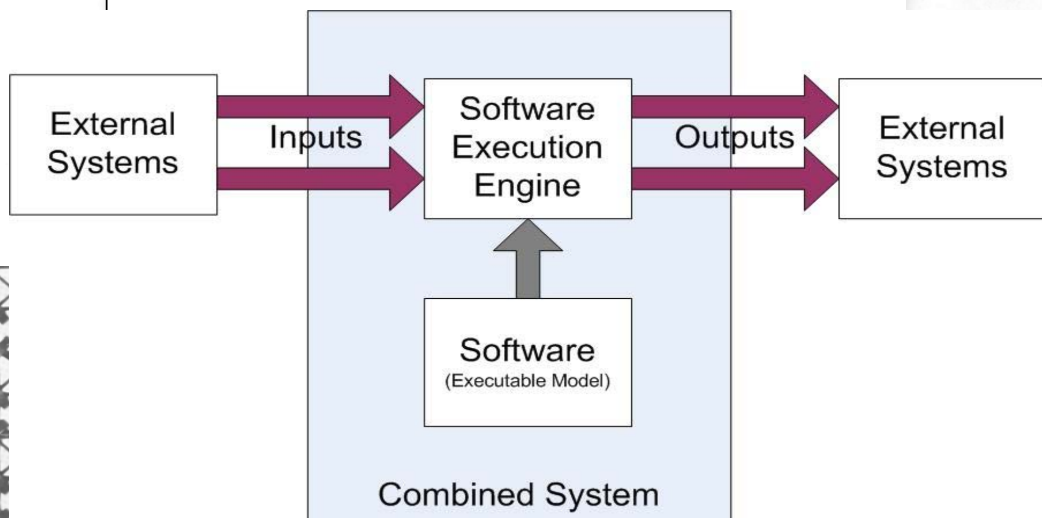


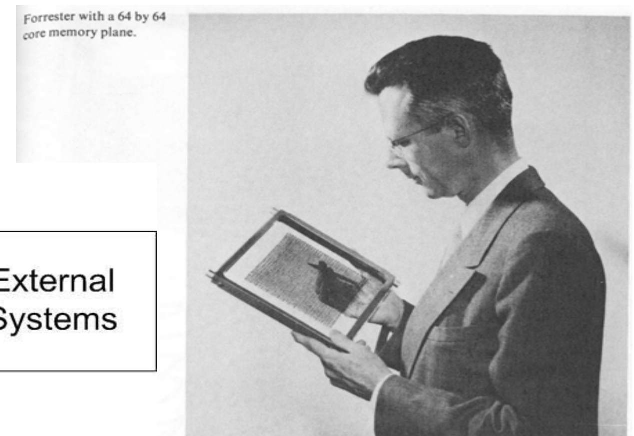
What history of technology tells us

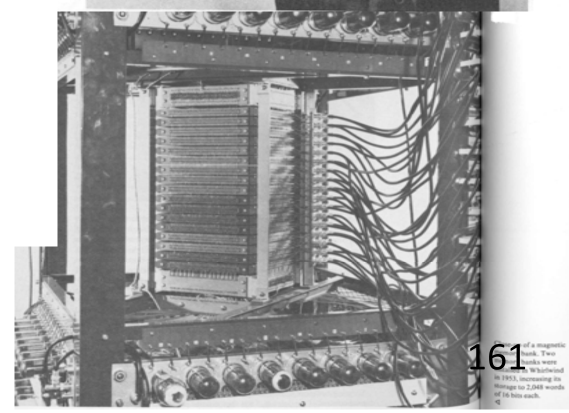
- Jay Forrester moved program data into magnetic core storage. (1950s)





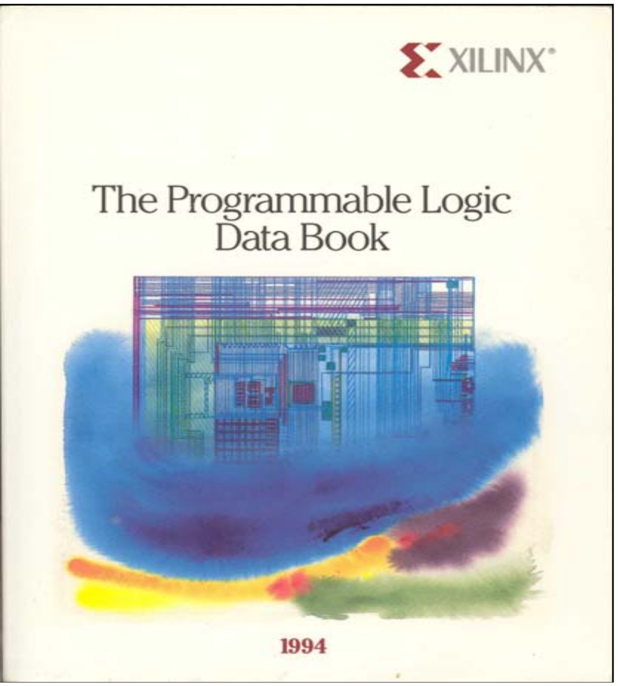


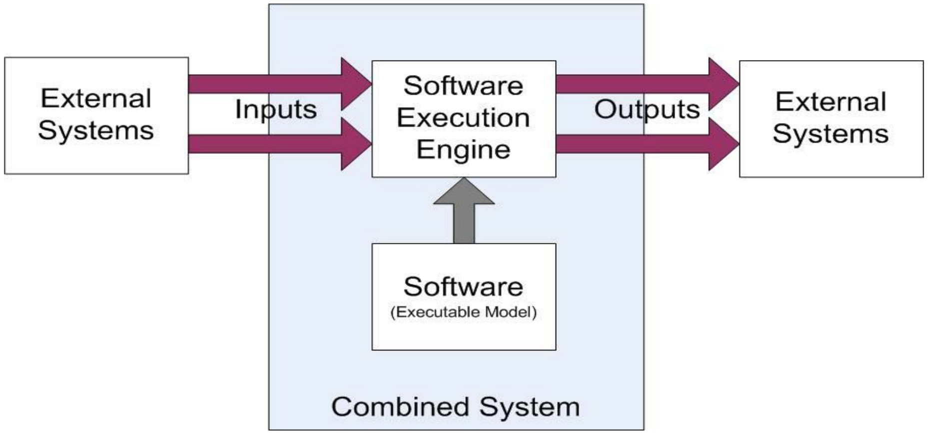




History of the technology tells us

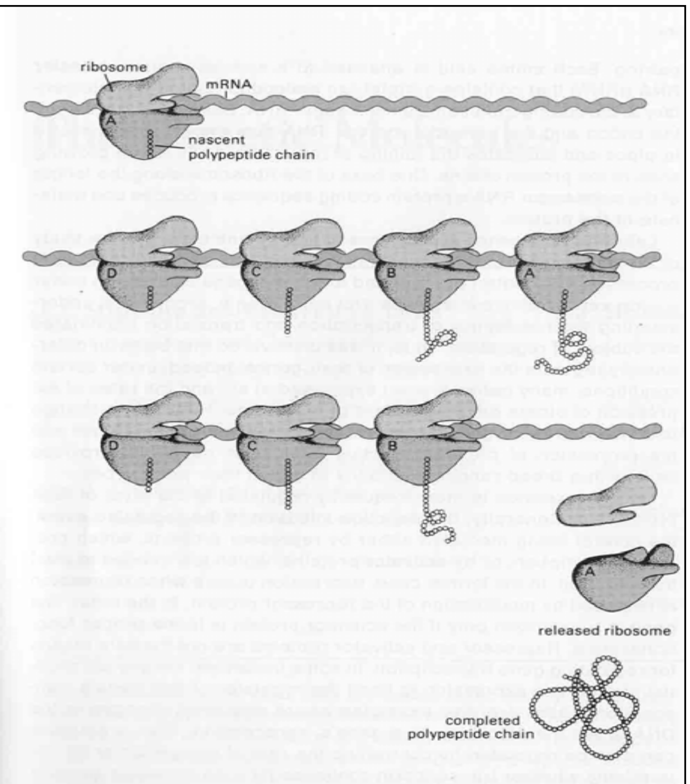
- Xilinx and other electronic hardware vendors develop “configurable hardware”: the idea of storing information as hardware, in very large scale high speed processors—ASICs and FPGAs.
- Other vendors developed VHDL, HDL, and RTL languages to define and test high complexity chip hardware.

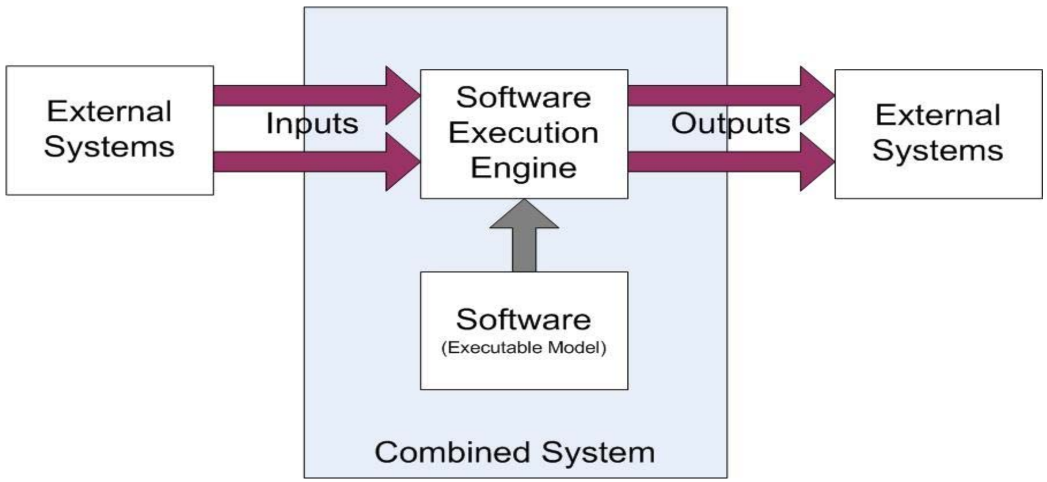


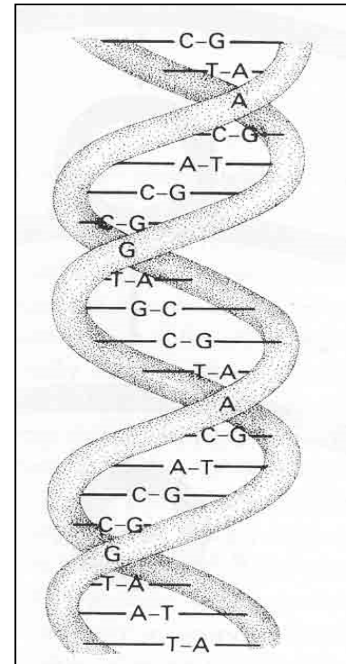


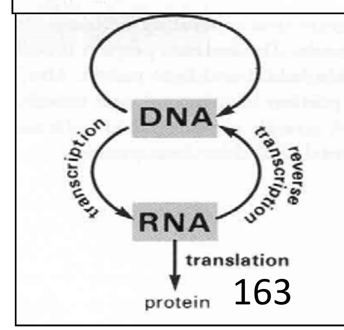
What the natural world tells us

- In nature, information stored in DNA is replicated, transcribed, and then used by ribosomes to generate protein molecular structures in living “epigenetic” systems.



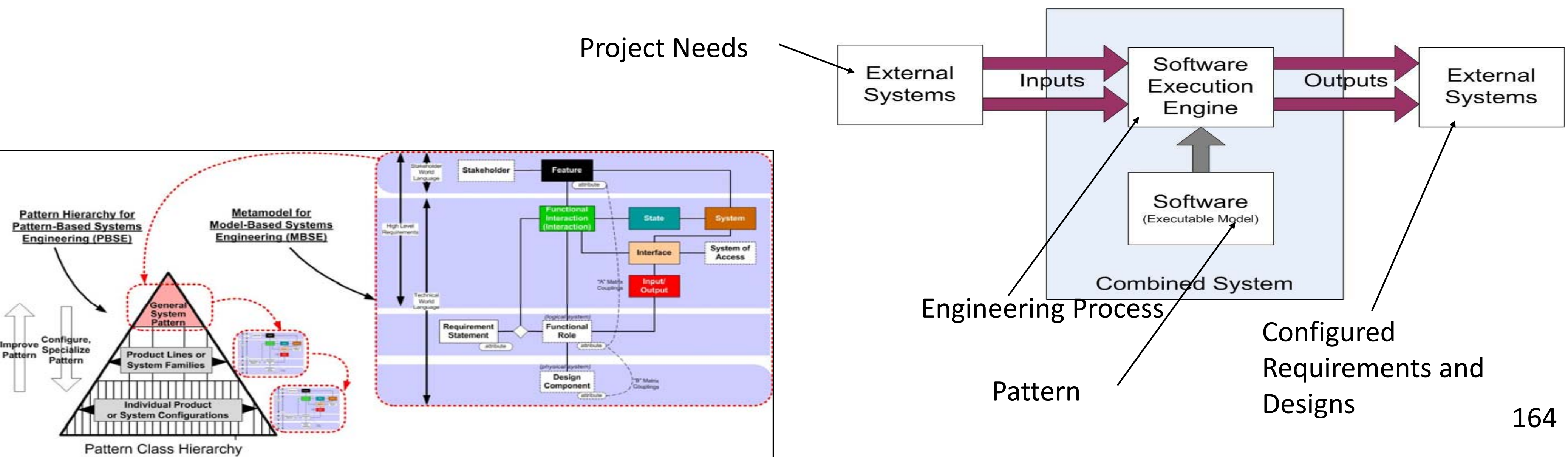






An engineering process application

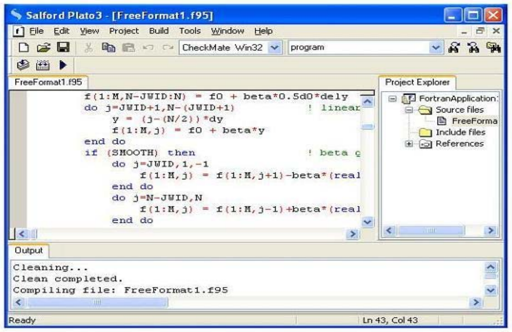
- Even the engineering process itself (along with its internal tools) is such an engine—using configured models to produce requirements and designs of new systems, in a never-ending cycle.



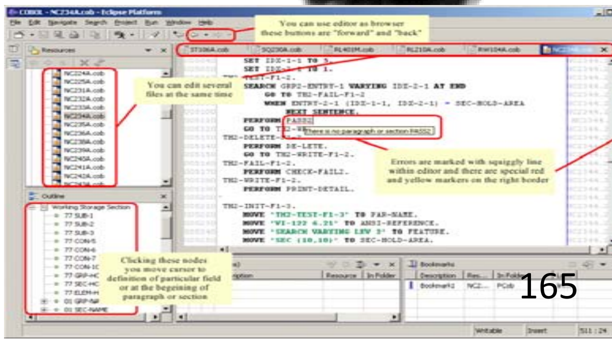
All these programming languages are themselves data structures – compiled or even run time interpreted by other programs

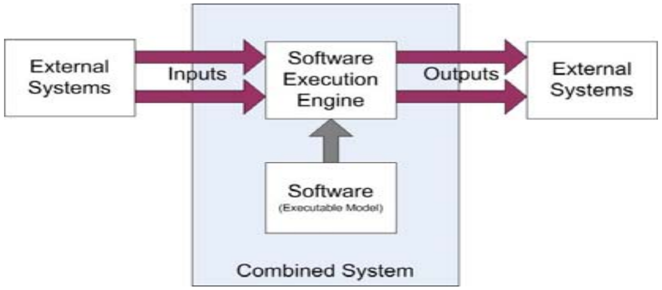
Software Languages as Data Structures

- FORTRAN (Formula Translation) Language:
  - A procedural programming language invented by John Backus to express mathematical formulas. (1950's)



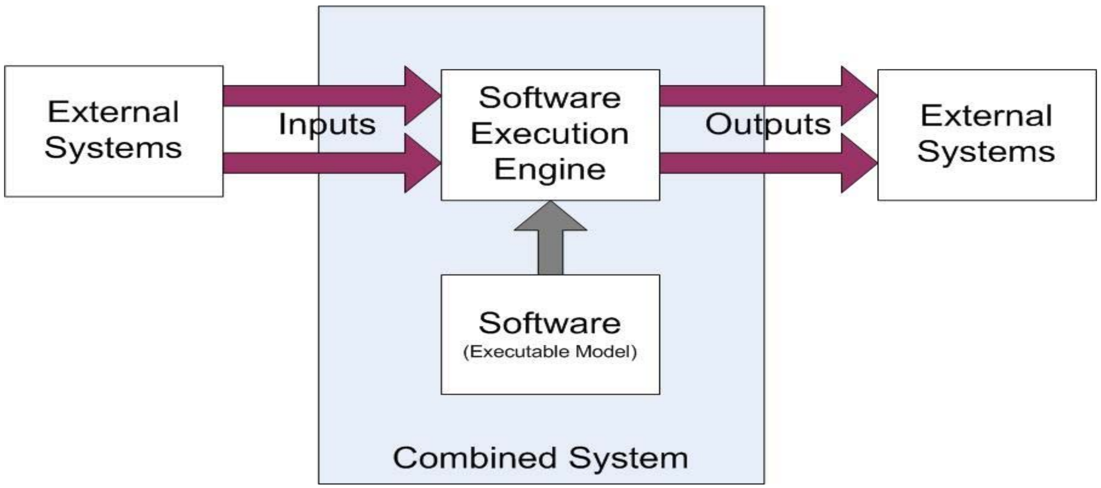
- COBOL (Common Business Oriented) Language:
  - A procedural language invented by Admiral Grace Hopper to express business algorithms. (1950-60's)





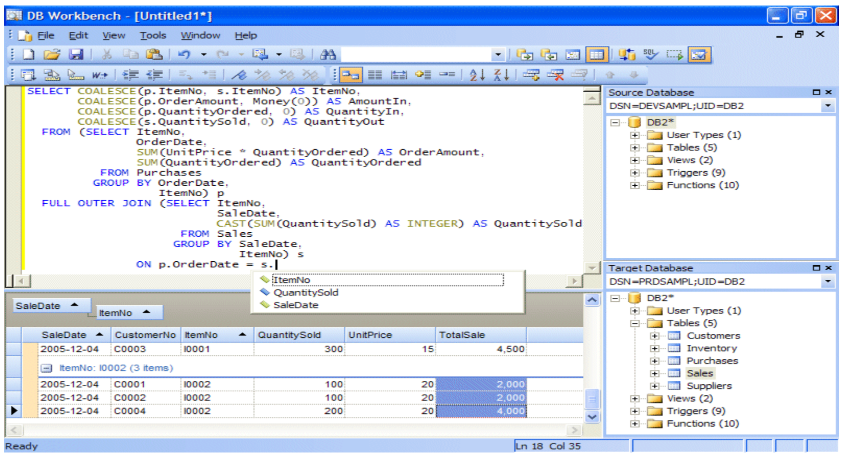
Not all software describes procedure

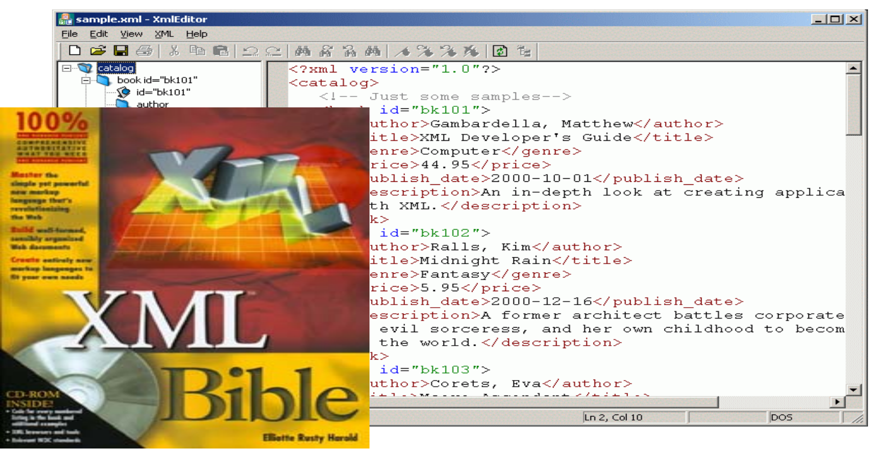
- It is not even safe to say that "software describes a sequence of operations" –
  - Because all Non-Procedural Languages are precisely <u>not</u> procedures!
  - Examples: SQL, XML, SCHEME, etc.



All these programming languages are themselves data structures – compiled or interpreted by other programs

Non-Procedural Software Languages



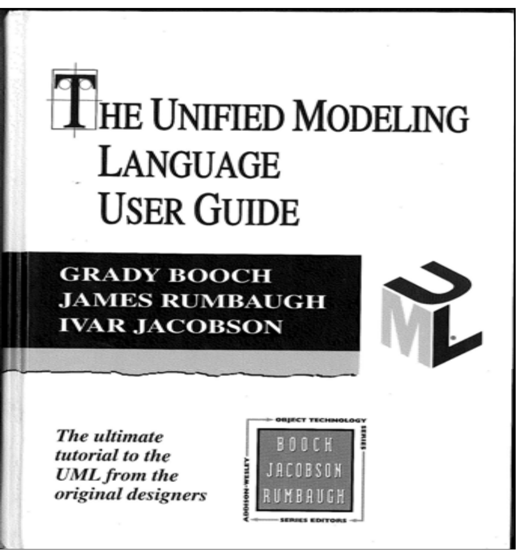


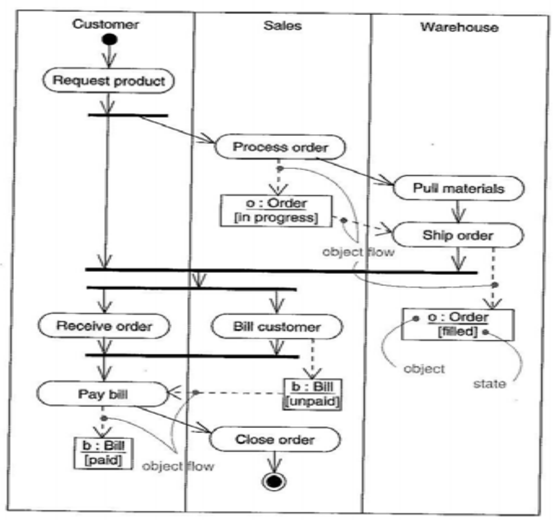
- These programming languages express the relationship of output data to input data without intermediate algorithms (D. Parnas):
  - <u>SQL</u> (Structured Query Language)
    - Invented by Codd and Date to express relational data models and operations on them. (1970's)
  - <u>XML</u> (Extensible Modeling Language)
    - Invented to express data models and transformations, as an evolution of SGML (1990's)
    - The foundation of many additional languages (e.g., Molecular Modeling Language, etc.)

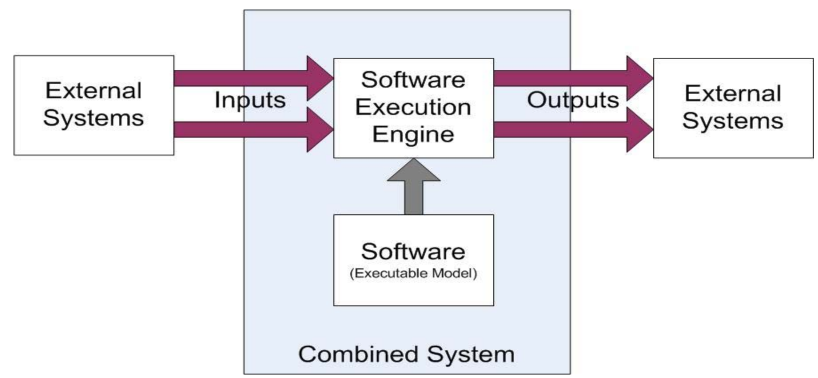
All these programming languages are themselves data structures – compiled or interpreted by other programs

Model-Defined Software

- More and more “traditional” software is now being developed by expressing both requirements and design in graphical data structures called “Models”:
  - UML™ (Unified Modeling Language) is the most popular current example (Booch, Rumbaugh, Jacobson; OMG™)



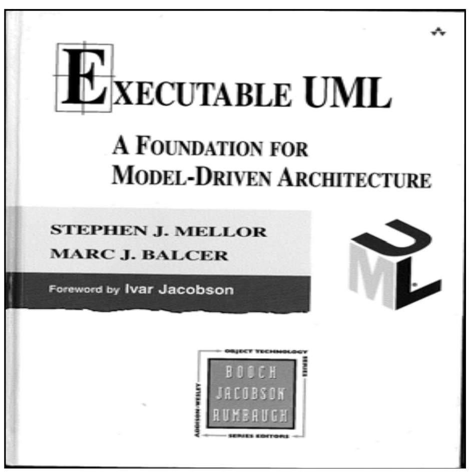


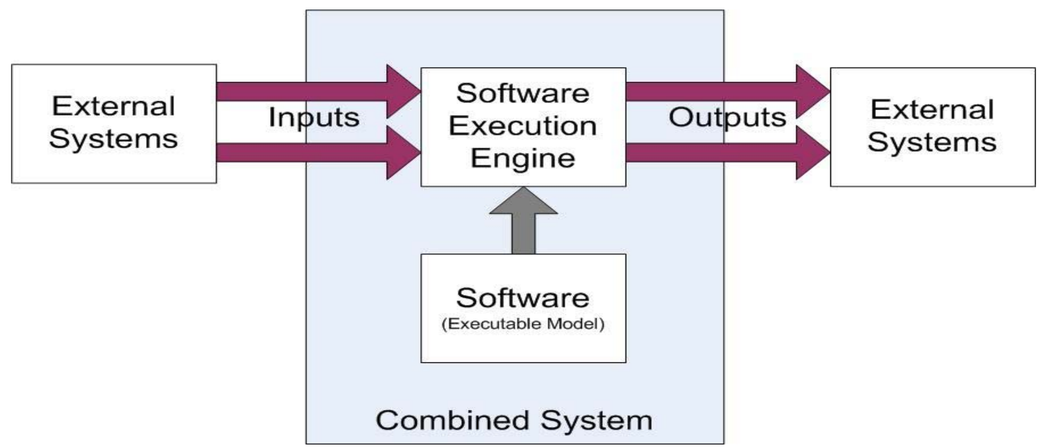


All these languages are themselves data structures – compiled or interpreted by other programs

Executable Models

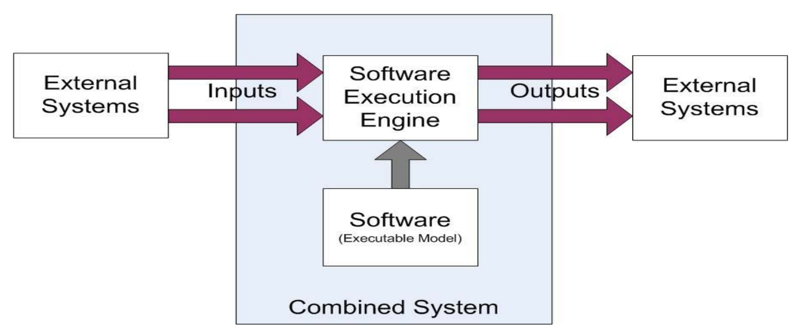
- Many "executable" models are being generated:
  - For traditional simulators (e.g., MATLAB™, etc.)
  - For requirements validation simulations (e.g., STATE MATE™, etc.)
  - For dual use as both source code generation as well as simulation execution (e.g., RHAPSODY™, etc.)





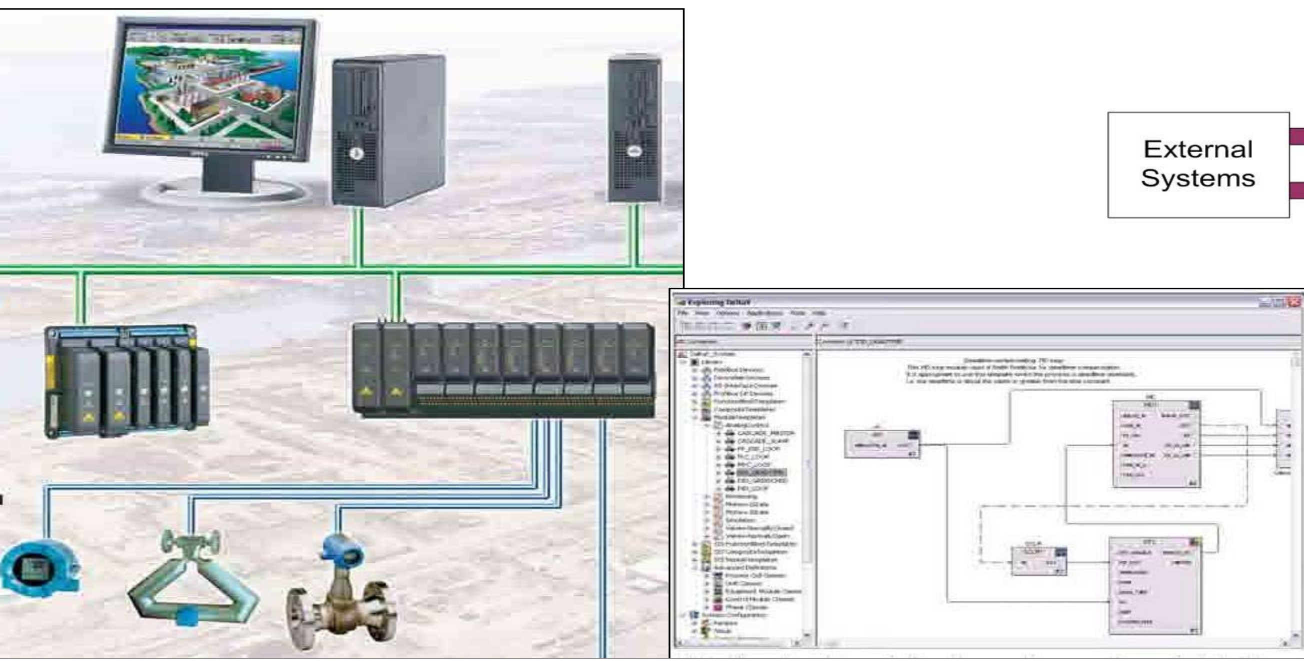
Not all software is “executed by hardware”

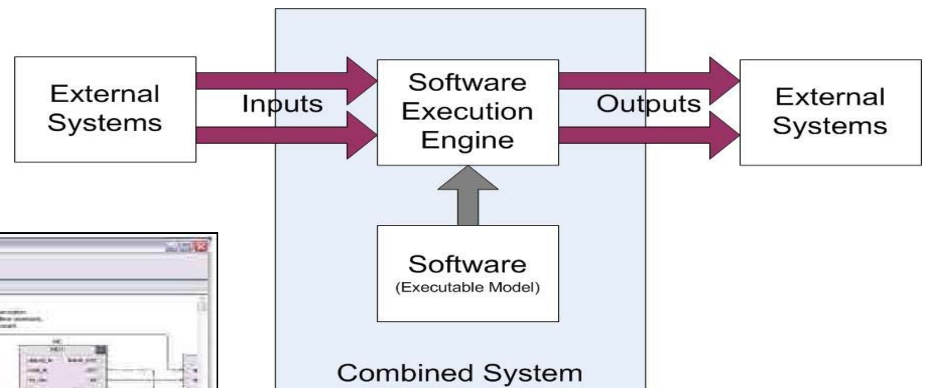
- Interpreted languages are very common:
  - e.g., BASIC is typically interpreted by software interpreters
  - “Virtual machines” are used to “execute” Java, etc.
  - This “code” is “executed” by other programs, not hardware!
- Microcoded emulators in chips:
  - Most modern microprocessors, PCs, servers, mainframes use “microcode” and are really “interpreters”.
- Emulators:
  - Many software and hardware debugging products use emulation by other code to “execute” the code being developed
- Spread sheets allow expression of complex relationships that are executed by software engines (e.g., Excel™, etc.)



An engineering process application

- Control systems suppliers and products (e.g., Emerson DeltaV™) now allow us to “program” our control systems using data structures created by inputting models.
- Many other programs are also created this way.





The economy itself becoming IP based

- US Government and Economists adopt increased capitalization of developed information assets:

THE WALL STREET JOURNAL. 9/29/06

THE ECONOMY

Economists Put a Number on R&D

Data on Knowledge Creation Point to an Increasing Role In U.S. GDP Expansion

By Greg Ip
And Mark Whitehouse

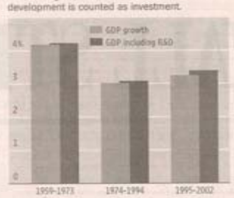
That research and development makes an important contribution to U.S. economic growth has long been obvious. But in an important advance, the nation's economic scorekeepers declared they can now measure that contribution and found that it is increasing.

The new data, part of ongoing efforts to pinpoint the role of knowledge creation in Americans' well-being, helps explain why the U.S. economy tends to expand faster than the contributions of capital and labor alone, a puzzle that has been the subject of Nobel-winning research.

In the current system of measuring gross domestic product, R&D is treated like a so-called intermediate expense. For example, salaries paid to research scientists are lumped in with wages paid to assembly-line workers. Under the new approach developed by the Commerce Department's Bureau of Economic Analysis and the National Science Foundation, R&D spending is treated like capital investment, such as the cost of a machine tool or an office building. By that measure, R&D would have accounted for nearly 7% of growth from 1995 to 2002, up from a little more than 4% from 1959 to 1994. (The rest comes from an expanding work force, increased capital and other, unexplained factors.) That exceeds by a wide margin the 2% contribution of investment in buildings and factories during the 1959-2002 period.

Brighter Picture

Economic growth is faster when research and development is counted as investment



Source: Commerce Department

Treating R&D as an investment would make the economy 3% larger and the national savings rate about two percentage points higher. "Our economy may be hundreds of billions of dollars larger than we currently calculate," said Commerce Secretary Carlos Gutierrez.

Since the 1950s, economists have explained economic output as the result of measurable inputs. Any increase in output that can't be explained by capital and labor is called "multifactor productivity" or "the Solow residual," after Robert Solow, the Nobel-prize winning economist considered the father of modern growth theory.

Between 1959 and 2002, this factor accounted for about 20% of U.S. growth. Between 1995 and 2002, when productivity growth accelerated sharply, that grew to about 33%. Accounting for R&D would explain about one fifth, by some measures, of the productivity mystery. It suggests companies have been investing more than the official data had previously shown—a good omen for future economic growth. "The slump in investment is not as drastic as people thought before they saw these figures," says Dale Jorgenson, professor of economics at Harvard University.

Mr. Jorgenson noted a lot of the multifactor productivity growth remains unexplained. "The great mystery of growth...is not eliminated."

Paul Romer, an economics professor at Stanford Business School, said the better the measurements of R&D become, the more economists and policy makers will realize other factors may be more important. "If you look at why we had rapid productivity growth in big-box retailing, there were lots of intangibles and ideas that...don't get recorded as R&D."

BEA director Steve Landefeld said for now, GDP will continue to be measured in the typical way, but the agency plans to start including R&D regularly in 2012.

Separately, the BEA said GDP grew at a 2.6% annual rate in the second quarter, down from an estimated 2.9%, in part because businesses didn't boost their inventories as much. In another report, the Labor Department said the number of new claims for unemployment insurance fell by 6,000 last week to 316,000.

Capitalizing on Systems Engineering

Jason J. Sherey
ICTT, Inc.
100 East Campus Drive, Terre Haute, IN 47802
(812) 232-5968 sherey@ictt.com

Abstract. Project managers often find it difficult to justify allocating significant resources and schedule to systems engineering tasks when "real" engineering has to be done. With ever-decreasing time to market demands, systems engineering continually loses out to design, integration, and test. A new method of systems engineering called Pattern-Based Systems Engineering (PBSE) enables companies to transfer portions of systems engineering costs out of project specific budgets and into company capital asset accounts. Such a change in accounting provides a series of benefits that include improved documentation and management of core corporate intellectual property, best practices, and standards as well as not having to reserve as much precious project money on tasks that need constant re-justification to product development managers with typically constrained budgets. This paper reviews PBSE, relevant accounting standards, and how much of systems engineering can be performed as a company-wide capital asset development program instead of as project overhead.

The Current Situation

Justifying Systems Engineering in Project Budgets. Systems engineering is sometimes described as a way of reducing risk to product development and support areas such as:

- Project Schedule,
- Life Cycle Cost,
- Integration and Test, and
- Product Performance (ICTT 2005b).

Most of these cost impacts, however, occur after product development phases and during support phases whose costs are often reported in completely different organizational accounts (Blanchard et al. 1998). The effects of not performing enough systems engineering are therefore not very visible to the product manager planning development budgets. This creates an "iceberg" effect in which the project manager figures a small allocation to systems engineering is all that is needed while the hidden yet much larger costs of maintenance, distribution, market share losses due to poor performance, and other life cycle support costs are ready to cripple the company (Blanchard 1998).

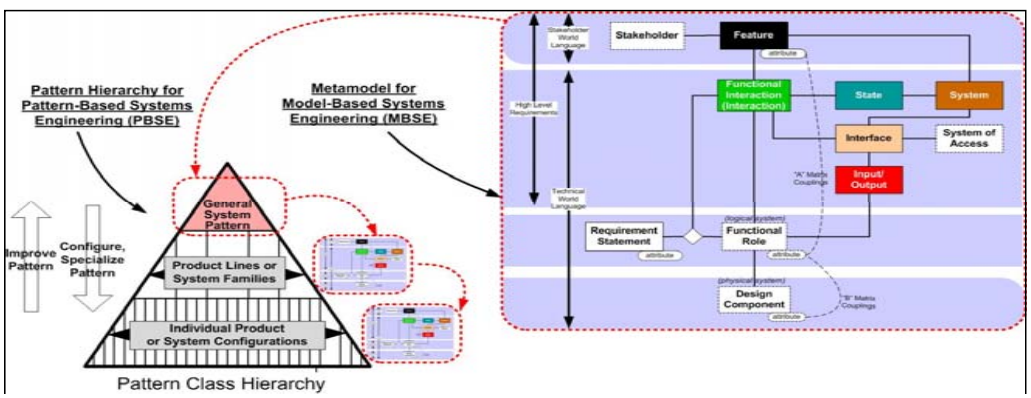
Arguments that explain that while program costs are spent mostly towards the end of the life-cycle the costs are actually committed based on the work very early in the program when most of the systems engineering is performed can be used to clarify the need for systems engineering (Blanchard 1998). However, systems engineering costs are usually accounted for in an overhead expense account in typical Work Breakdown Structures (WBS) that also includes costs for project management (DOD SMC 2001). When looking at the rest of the WBS, it is very difficult

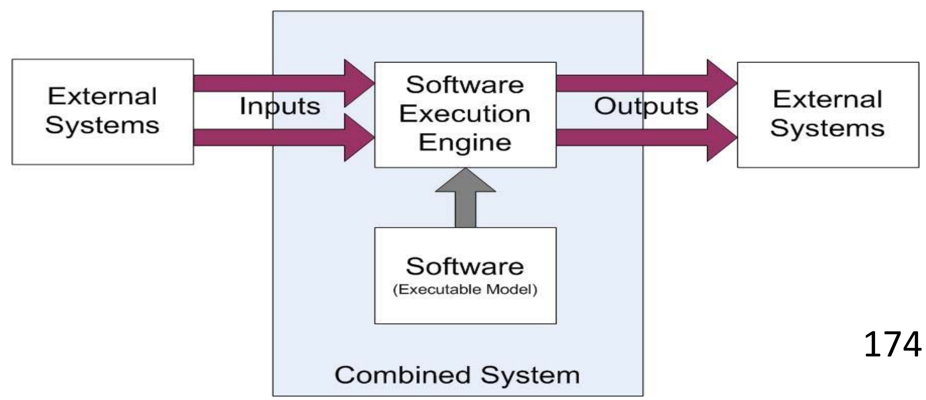
Capitalizing information assets

- Does this mean that all software deserves to be capitalized?
  - Of course not!
- There are many “hurdles” to capitalization, that only <u>some</u> software will clear:
  - For example, $ valuation and life of the asset;
  - And (especially for software) solid life cycle management of the asset: Its requirements, design, verification, maintenance, configuration and version management, support, etc.
  - FASB and other industry or professional criteria;
  - Specific capitalization criteria of the enterprise.

Pattern Capitalization: Implications

1. We are moving toward the Model Based Economy (MBE)—more of our assets are intellectual property (IP)—and many are models.
2. Software is information used as an executable model.
3. This model is interpreted by many types of “execution engines”.
4. Patterns are a form of software.
5. Even the engineering process is such an engine—configuring patterns produce requirements and designs of new systems.
6. Software life cycle management informs us about pattern life cycle management.





Exercise 6: Financial Capitalization of System Patterns

1. How much of your systems engineering costs might be dealing with variants around a common core theme?
2. How important would ability to afford more systems engineering cost be in your enterprise, moving cost to time of realizing value?
3. What pattern would you capitalize?
4. Who would care about moving cost of development off P&L and onto balance sheet?

Related INCOSE, ASME communities

- **INCOSE:**
  - Model-Based Engineering Transformation Initiative
  - INCOSE-NAFEMS Joint Working Group on Simulation
  - MBSE Patterns Working Group
  - Agile Systems & Systems Engineering Working Group
  - Tools Interoperability and Model Life Cycle Management Group
  - INCOSE-OMG MBSE Initiative: Challenge Teams, Activity Teams

- **ASME Computational Model V&V Committee / Working Groups:**
  - V&V 10: Verification & Validation in Computational Solid Dynamics
  - V&V20: Verification & Validation in Computational Fluid Dynamics and Heat Transfer
  - V&V 30: Verification and Validation in Computational Simulation of Nuclear System Thermal Fluids Behavior
  - V&V 40: Verification and Validation in Computational Modeling of Medical Devices
  - V&V 50: Verification & Validation of Computational Modeling for Advanced Manufacturing
  - V&V 60: Verification and Validation in Modeling and Simulation in Energy Systems and Applications

Additional Sources of Help:

- **S*Patterns Community:**
  - A member community of people, enterprises, and institutions employing advanced methods and assets for the world's most challenging systems issues—unlocked by Model-Based Patterns using the S*Metamodel
- **Virtual Verification, Validation, and Visualization Institute (V4I):**
  - A member community of people, enterprises, and institutions improving the effectiveness of product development and other life cycle processes, employing model-based verification, validation, and visualization
- **Uncover the Pattern™:**
  - A fast path to creation of the first draft of your organization's fundamental system S*Pattern in 90 days or less

End of Part II

Attachments

- Exercise hand-outs
- Pattern extract hand-outs

References

Model Planning and Purpose, Learning, Agility, Adaptation, and PBSE:

1. "INCOSE MBSE Transformation Planning & Assessment Framework: Beta Test": http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:planning_assessment_requirements_for_mbse_model_applications_v1.4.2.pdf
2. Schindel, W. "Pattern-Based Systems Engineering: An Extension of Model-Based SE", INCOSE Tutorial TIES 4. *Proceedings of INCOSE 2005 Symposium*. 2005.
3. Schindel, W. "Requirements Statements Are Transfer Functions: An Insight from Model-Based Systems Engineering", *Proceedings of INCOSE 2005 Symposium*. 2005.
4. Schindel, W., and Smith, V. "Results of Applying a Families-of-Systems Approach to Systems Engineering of Product Line Families", SAE International, Technical Report 2002-01-3086. 2002.
5. Beihoff, B., et al, "A World in Motion: INCOSE Vision 2025", INCOSE.
6. Schindel, W., "What Is the Smallest Model of a System?", Proc. of the INCOSE 2011 International Symposium, International Council on Systems Engineering (2011).
7. Schindel, W., and Dove, R., "Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern", in Proc. of INCOSE 2016 International Symposium, 2016.
8. Schindel, W., "Got Phenomena? Science-Based Disciplines for Emerging Systems Challenges PBSE methodology summary", Proc. of INCOSE IS2017 Symposium, Adelaide, UK, 2017.
9. INCOSE 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

Model VVUQ and Credibility

10. 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

11. *Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification* ISBN 978-0-309-25634-6 THE NATIONAL ACADEMIES PRESS, http://nap.edu/13395

12. Web site of ASME VV50 https://cstools.asme.org/csconnect/CommitteePages.cfm?Committee=100003367

13. "ASME V&V 10-2006: Guide for Verification and Validation in Computational Solid Mechanics", ASME, 2006.

14. "ASME V&V 20-2009: Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer", ASME, 2009.

15. "ASME V&V 10.1-2012: An Illustration of the Concepts of Verification and Validation in Computational Solid Mechanics", ASME, 2012.

16. *Journal of Verification, Validation, and Uncertainty Quantification*, ASME. https://verification.asmedigitalcollection.asme.org/journal.aspx

17. AIAA (American Institute for Aeronautics and Astronautics). 1998. *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations.* Reston, Va.

18. Box, G., and N. Draper. *Empirical Model Building and Response Surfaces*. New York: Wiley, 1987.

EI Pattern and IOT; Manufacturing Pattern and Digital Plant:

19. Peterson, T., and Schindel, W., "Pattern Based Systems Engineering – Leveraging Model Based Systems Engineering for Cyber-Physical Systems", Proc. Of NDIA 2014 NDIA GVSETS Symposium on Systems Engineering, Novi, MI, August, 2014.
20. Lu, Y., Morris, K., Frechette, S., "NISTIR 8107: Current Standards Landscape for Smart Manufacturing Systems", National Institute of Standards and Technology, US Dept. of Commerce, February 2016.
21. "Industry 4.0 / Smart Manufacturing Final Report", ISO TMB Strategic Advisory Group, September 2016.
22. "Plattform Industrie 4.0 Progress Report", German BMWi, April 2016, at http://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/digitization-of-industrie-plattform-i40.pdf?__blob=publicationFile&v=4
23. ANSI/ISA–88.00.01–2010, "Batch Control--Part 1: Models and Terminology", RTP, NC: International Society of Automation.
24. IBM Corporation. "The Metamorphosis of Manufacturing: From Art to Science. IBM Global Services, Somers, NY, 2005.
25. Ilegbusi, O., Iguchi, M., Wahnsiedler, W. *Mathematical and Physical Modeling of Materials Processing Operations*, Boca Raton: CRC Press, 2000.
26. U. S. Dept of Health and Human Services, FDA. "Innovation and Continuous Improvement in Pharmaceutical Manufacturing — Pharmaceutical cGMPs for the 21st Century". 2004. http://www.fda.gov/cder/gmp/gmp2004/manufSCIWP.pdf
27. U. S. Dept of Health and Human Services, FDA. "Final Report on Pharmaceutical cGMPs for the 21st Century— Risk-based Approach". 2004, http://www.fda.gov/cder/gmp/gmp2004/GMP_finalreport2004.htm
28. U. S. Dept of Health and Human Services, FDA. "Q8 Pharmaceutical Development ICH Draft: Step 2". 2004. http://www.fda.gov/cder/guidance/6672.dft.htm
29. "ICH Quality Guidelines 2009, ICHQ1 – ICHQ10", International Conference on Harmonisation web site, 2009: http://www.ich.org/cache/compo/276-254-1.html
30. U. S. Dept of Health and Human Services, FDA Science Board. Scherzer. R. "Quality by Design: A Challenge to the Pharma Industry". 2005. http://www.fda.gov/cder/OPS/Scherzer-Camp/index.htm
31. Gunyon, R., and Schindel, W. "Engineering Global Pharmaceutical Manufacturing Systems in the New Environment", *Proceedings of the 2010 INCOSE International Symposium*. International Council in Systems Engineering. 2010.
32. Berg, E., "Affordable Systems Engineering: An Application of Model-Based System Patterns To Consumer Packaged Goods Products, Manufacturing, and Distribution", INCOSE International Workshop, IW2014, Los Angeles, Jan, 2014.
33. Schindel, W. "Systems Engineering for Advanced Manufacturing: Unit Op Insights from Model-Based Methods". *Proceedings of the INCOSE 2011 International Symposium*. 2011.

References

Capitalization of Patterns

34. W. Aspray, *John von Neumann and the Origins of Modern Computing*, MIT Press, 1990.
35. K. Redmond and T. Smith, *Project Whirlwind: The History of a Pioneering Computer*, ISBN 0-932376-09-6, Digital Press History of Computing Series, 1980.
36. M Singer and P. Berg, *Genes & Genomes: A Changing Perspective*, University Science Books, 1991.
37. J. Essinger, *Jacquard's Web: How A Hand Loom Led to the Birth of the Information Age*, Oxford U. Press, 2004.
38. D. Swade, *The Difference Engine: Charles Babbage and the Quest to Build the First Computer*, Viking Press, 2000.
39. *The Programmable Logic Data Book*, Xilinx, 1994.
40. J. Sherey, "Capitalizing on Systems Engineering", *Proceedings of the INCOSE 2006 Symposium*, July, 2006.
41. "Economists Put a Number on R&D", *Wall Street Journal*, 09/29/06.
42. Gamma, Helm, John, and Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995
43. Financial Accounting Standards Board, *Statement of Financial Accounting Standards No. 2: Accounting for Research and Development Costs*, October 1974.
44. Financial Accounting Standards Board, *Statement of Financial Accounting Standards No. 86: Accounting for the Costs of Computer Software to Be Sold, Leased, Otherwise Marketed*, August 1985.
45. Financial Accounting Standards Board, *Statement of Financial Accounting Standards No. 142: Goodwill and Other Intangible Assets*, June 2001.
46. Schindel, William D., "Requirements Statements Are Transfer Functions: An Insight from Model-Based Systems Engineering", *Proceedings of INCOSE 2005 Symposium*, July, 2005.
47. Sherey, Jason J., "A New Method to Justify Systems Engineering", INCOSE Crossroads of America Chapter Fall Mini-Conference, Fort Wayne, IN, 2004.

Speaker

Bill Schindel chairs the MBSE Patterns Working Group of the INCOSE/OMG MBSE Initiative. He is president of ICTT System Sciences, and has practiced systems engineering for over thirty years, across multiple industry domains. Bill serves as president of the INCOSE Crossroads of America Chapter, and is an INCOSE Fellow and Certified Systems Engineering Professional. An ASME member, he is part of the ASME VV50 standards team's effort to describe the verification, validation, and uncertainty quantification of models.

ICTT System Sciences®
Understand your systems.