THE IMPACT OF "DARK PATTERNS" ON UNCERTAINTY:

ENHANCING ADAPTABILITY IN THE SYSTEMS WORLD

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

Great Lakes Regional Conference, November, 2011

Abstract

- There are increasing examples of the formal use of System Patterns in the development, delivery, and life cycle of systems, including patterns of requirements, design architectures, failure modes and effects, verification processes, and other aspects. Nevertheless, today many systems engineering efforts (even when model-based) still occur without use of explicit Pattern-Based SE methods. In spite of this apparent lack of formal or explicitly visible patterns, most contemporary systems projects do in fact have pattern content--but in the form of “Dark Patterns”.
- This presentation examines the role of Dark Patterns in commonly-practiced contemporary systems engineering projects. Like the “dark matter” of physics, Dark Patterns (1) are not directly visible, (2) exert substantial forces that can make their presence known, and (3) are widespread throughout the SE universe. Understanding Dark Patterns and their explicitly visible counterparts can enhance the adaptability of systems and systems projects in a dynamic world of risks.

Contents

- Dark matter
- The systems engineering connection
- System patterns—dark versus explicit
- Leveraging adaptability to tame uncertainty

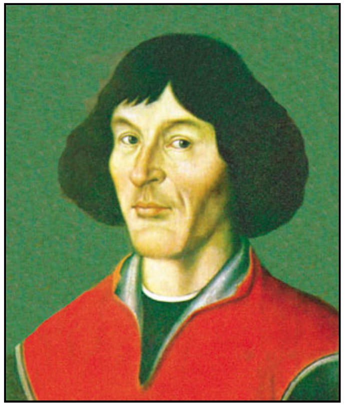
- References

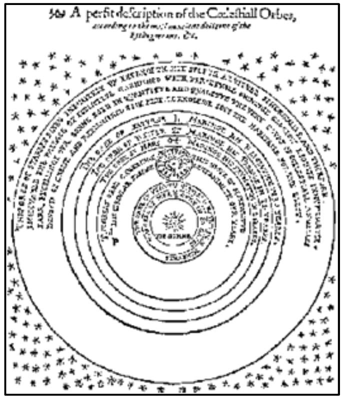
Dark Matter

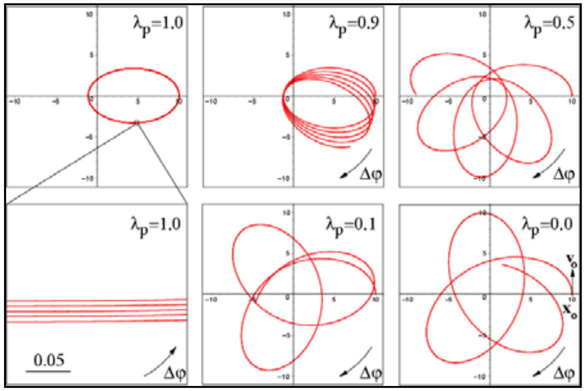
- Some cosmologists believe “Dark Matter” exists:
  - Is invisible (optically),
  - Exerts gravitational force on the rest of matter,
  - Is a major & widespread component of the universe.
- Otherwise unexplained behavior of the universe seems explained by Dark Matter.

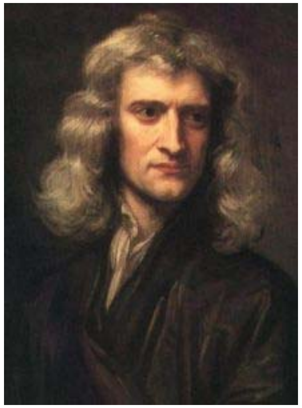
Science Seeks Models

- Will Dark Matter become fully accepted by the scientific community to explain the patterns of observed behavior?
  - It is still early to say.
- Earlier, the Copernican Revolution required generations to be the accepted explanation of planetary behavior patterns:
  - Others after Copernicus (Galileo, Brahe, Kepler, Newton) were aided by finding improved methods of representation to better explain regularities of observed behavior.



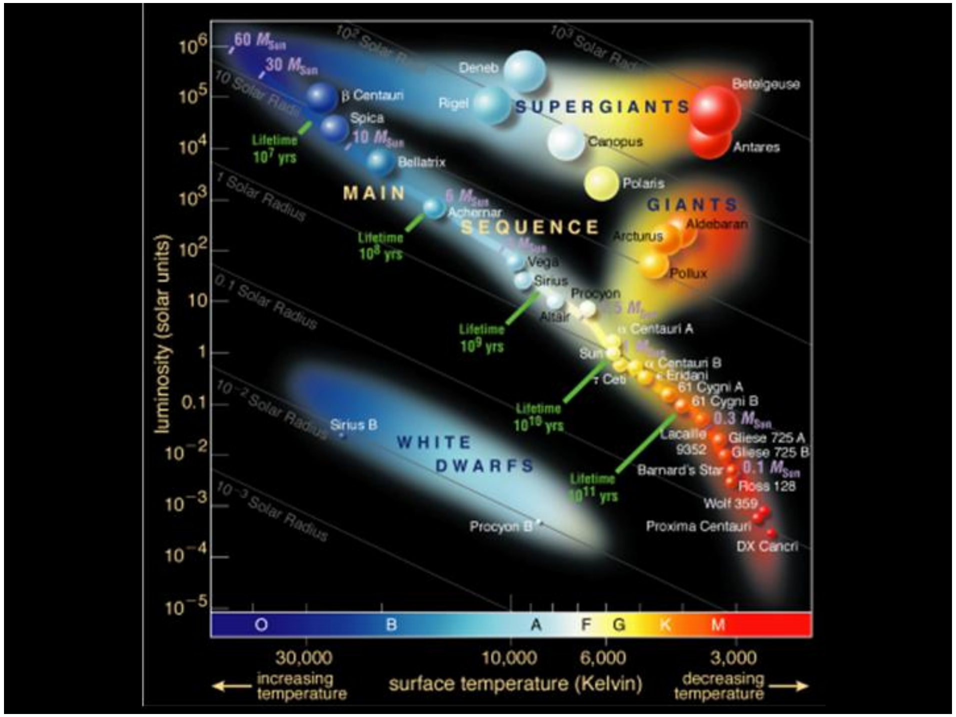






One-off Models versus Family Patterns

- These improved understandings depended not just on better Models of individual situations, but also on . . .
- Representation of Families, helping to understand different types of behavior, organizing the universe further into “types”:



The systems engineering connection

- *Discovering regularities and how to represent them* has been at the heart of science and engineering progress:
  - The INCOSE System Sciences Working Group (SSWG) bridges the interests of engineering and science.
  - Next meeting will be at IW2012.
  - https://sites.google.com/site/syssciwg/
- *Ability to manage risk and adapt* are related to our *awareness* and *understanding* of the *regularities* (patterns) around us:
  - Whether in the systems we engineer, or the markets and operational environments in which their life cycle unfolds.
  - They exert “forces” on us, whether are aware of them or not.

What repeating regularities are of interest?

- Smaller-Scale Regularities:
  - Patterns of Stakeholder Features (e.g., in vehicles, energy systems, etc.)
  - Patterns of Requirements
  - Patterns of Design Solutions
  - Patterns of Failure Modes and Effects
  - Patterns of Functional Roles, Interactions, States
  - Patterns of Interfaces, Input-Outputs, and Access
  - Patterns of Technologies
- Larger-Scale Regularities:
  - Patterns of how all the above are related to each other
  - Patterns in couplings across systems, domains, SOS's
  - Systems of Material Handling, Production, Distribution, Sustainment
  - Systems of Innovation
  - Patterns of Systems Pathologies

INCOSE SSWG

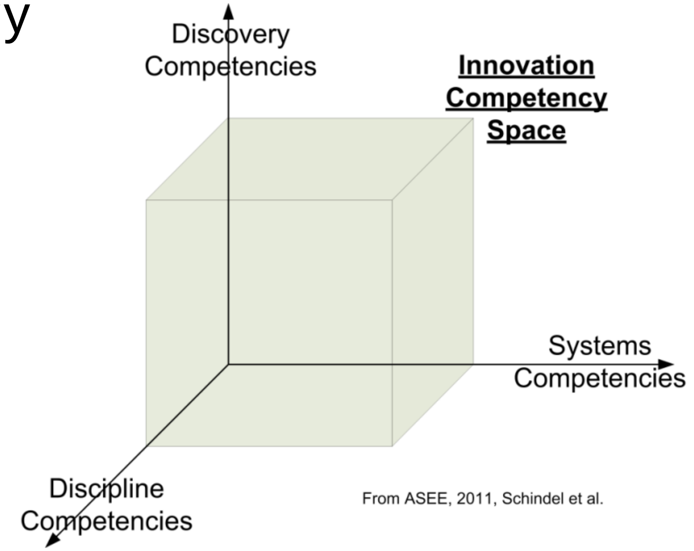
Is this “just of academic interest”?

- Hardly! Lack of awareness of these regular patterns leaves products, programs, enterprises at serious risk:
  - Re-experiencing the same mis-steps and reworks;
  - Just because we have made one system work, how do we know what will happen when we deploy more of them, as markets, conditions, & technologies evolve?
  - Just because our system has human experts on hand today, how do we know what will happen when they move on?
- Example cases and responses:
  - FDA push to the pharmaceutical manufacturing industry to improve the science-based understanding of underlying process transformations, provable ranges, and control strategies, etc.
  - The generation of system requirements families for globally-deployed product families and their production, distribution, and support systems.
  - The generation of system verification plans from underlying patterns of system requirements.
  - The use of System Patterns to generate Risk Analyses (e.g., FMEAs, etc.) for a variety of domain systems.

“Chance favors the prepared mind”

- Louis Pasteur

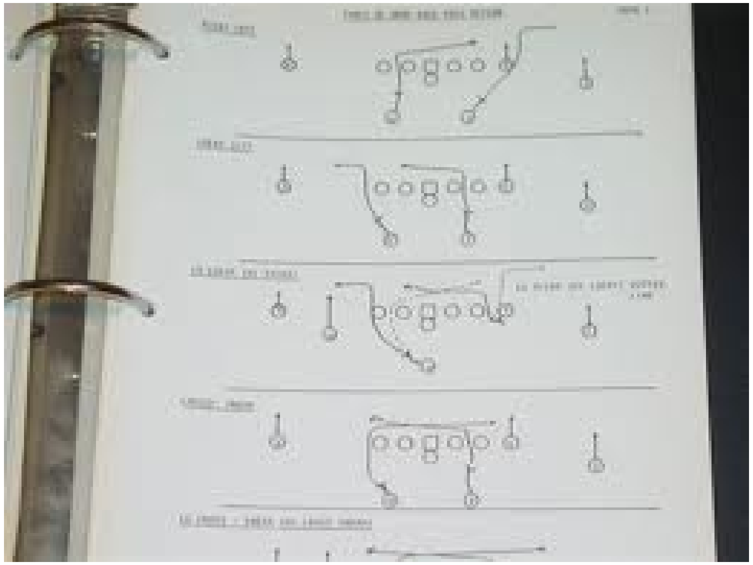
- Explicit patterns help us organize what we know--as well as what we don’t.
- Explicit preparation for:
  - System & program risks
  - Market & competitive shifts
  - New science & technology
  - Life cycle extensions
- Adaptability!



From ASEE, 2011, Schindel et al.

Adaptation Response Time

- Explicit pattern awareness helps us to:
  - Recognize the situation has changed.
  - Know the best alternate pattern configuration.



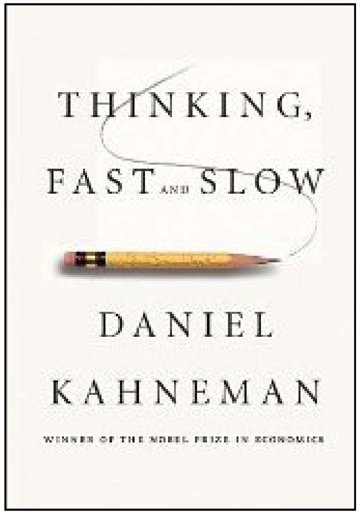
INCOSE
International Council on Systems Engineering

Irrationality: Human beings' behaviorally-preferred mode?

- A broad issue across human life:
  - The science of irrationality
  - Daniel Kahneman, Nobel Laureate, "Thinking, Fast and Slow")
  - "Moneyball", Oakland A's, Billy Beane.

- Engineering teams more rational than others?
  - Ever encounter a bad decision?
  - A significant fraction of requirements are left unstated

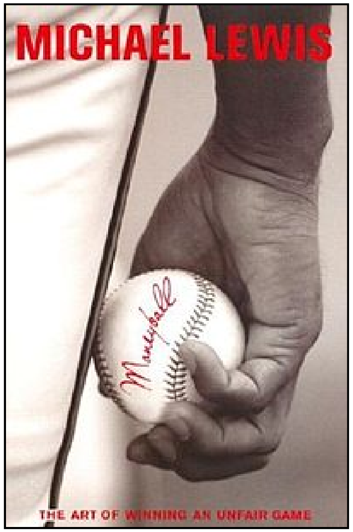
- Patterns existing in Nature do not mean the patterns are recognized by humans



THINKING, FAST AND SLOW

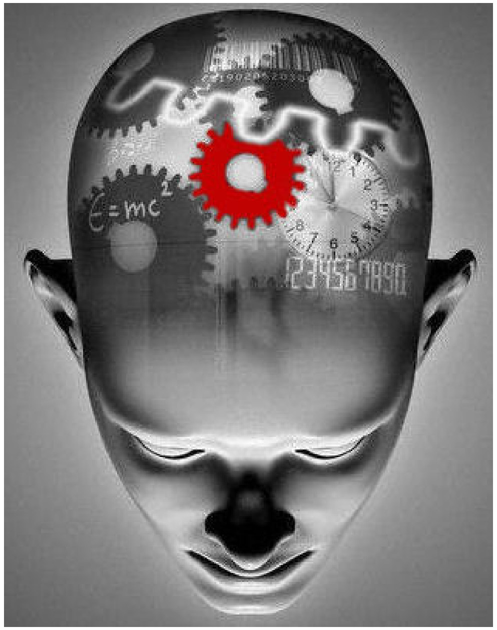
DANIEL KAHNEMAN

WINNER OF THE NOBEL PRIZE IN ECONOMICS



One way people cope . . .

- "Domain experts" internalize patterns:
  - These human experts influence our projects, using their experience, intuition, informed judgment.



System Patterns: Dark and Visible

- The regularities are "out there", whether we represent them or not:
  - In particular, they impact our ability to deal with uncertainty and adaptability.
- We use the term Dark Pattern to refer to system regularities that have not been explicitly represented:
  - They are in a sense "invisible", but still impact our systems, customers, programs, enterprises, institutions, and society.
- By contrast, when we represent those System Patterns formally, they become "visible", as Explicit Patterns:
  - Our method for doing this is Pattern-Based Systems Engineering (PBSE);
  - PBSE is an extension of Model-Based Systems Engineering (MBSE);
  - PBSE creates and applies configurable, re-usable models, called Patterns;
  - They typically include much more than just the "subject system".

How many patterns are Dark?

- Most systems programs involve Patterns, such as:
  - Patterns of available technologies and parts
  - Patterns of candidate solution architectures
  - Patterns of interfaces
  - Patterns of system states or modes
  - Patterns of customers, or market expectations
  - Patterns of competitive offerings
  - Patterns of system failures modes and effects
- Most systems engineering efforts—even model-based--still occur without use of explicit Pattern-Based methods:
  - This is the world of Dark Patterns.
  - Example: Nearly universally missed requirements.
- Explicit Patterns prepare us to adapt by describing key objects, relationships, and variables—including multiple types of risk.

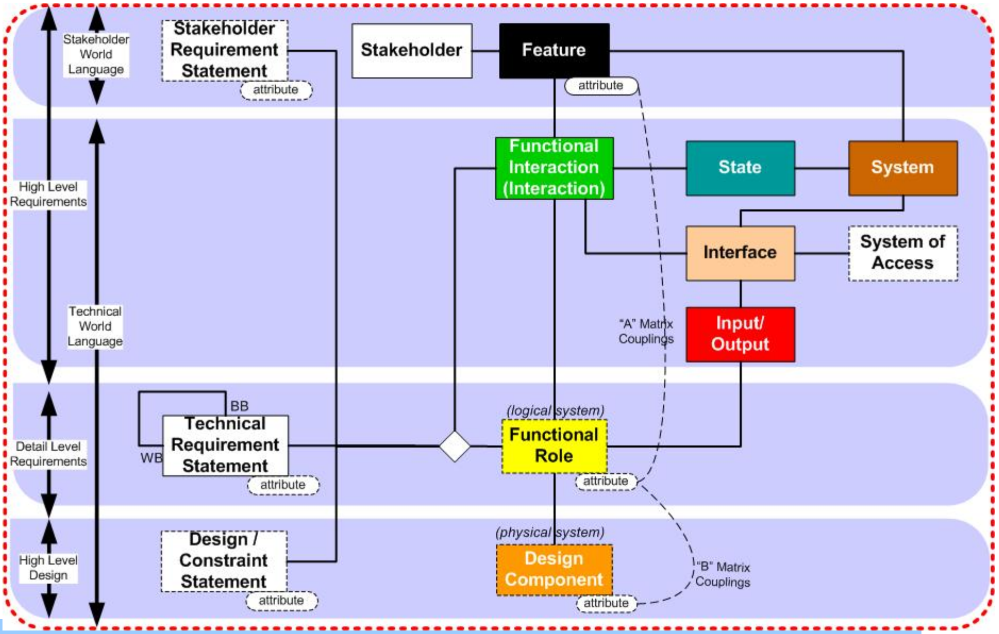
Representing System Patterns

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

Constructing an efficient representation

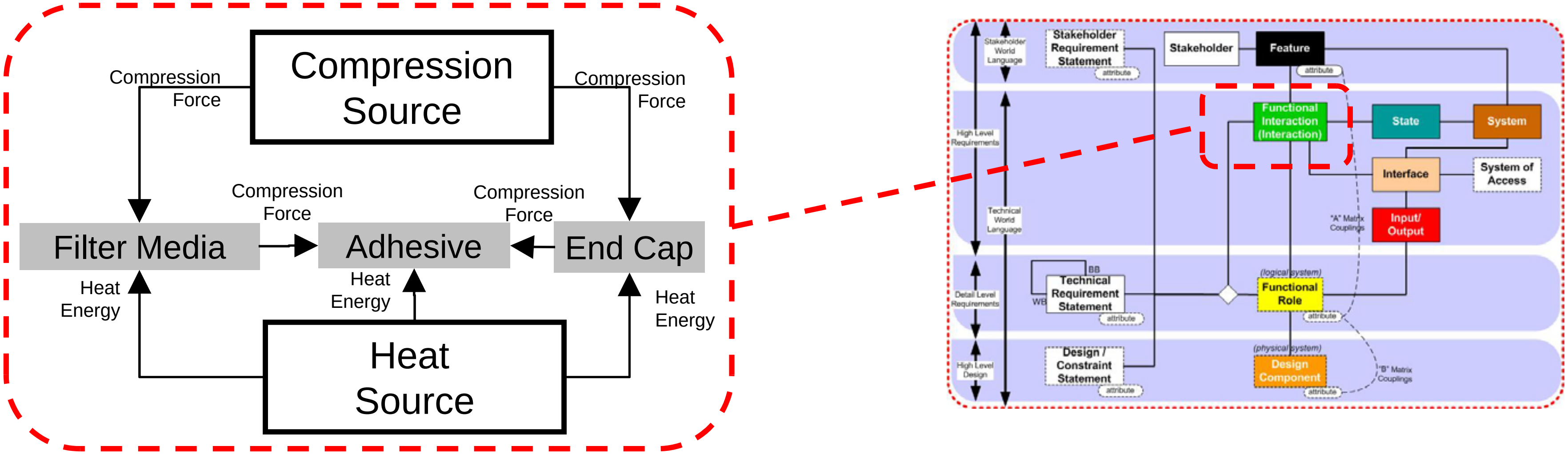
- A <u>metamodel</u> is a model of other models;
  - Sets forth how we will represent Requirements, Designs, Verification, Failure Analysis, Trade-offs, etc.;
  - We utilize the (language independent) S* Metamodel from Systematica™ Methodology:
- The resulting system models may be expressed in SysML™, other languages, DB tables, etc.
- Has been applied to systems engineering in aerospace, transportation, medical, advanced manufacturing, communication, construction, other domains.

Simple summary of detailed S* Metamodel.



Physical Interactions: At the heart of S* models

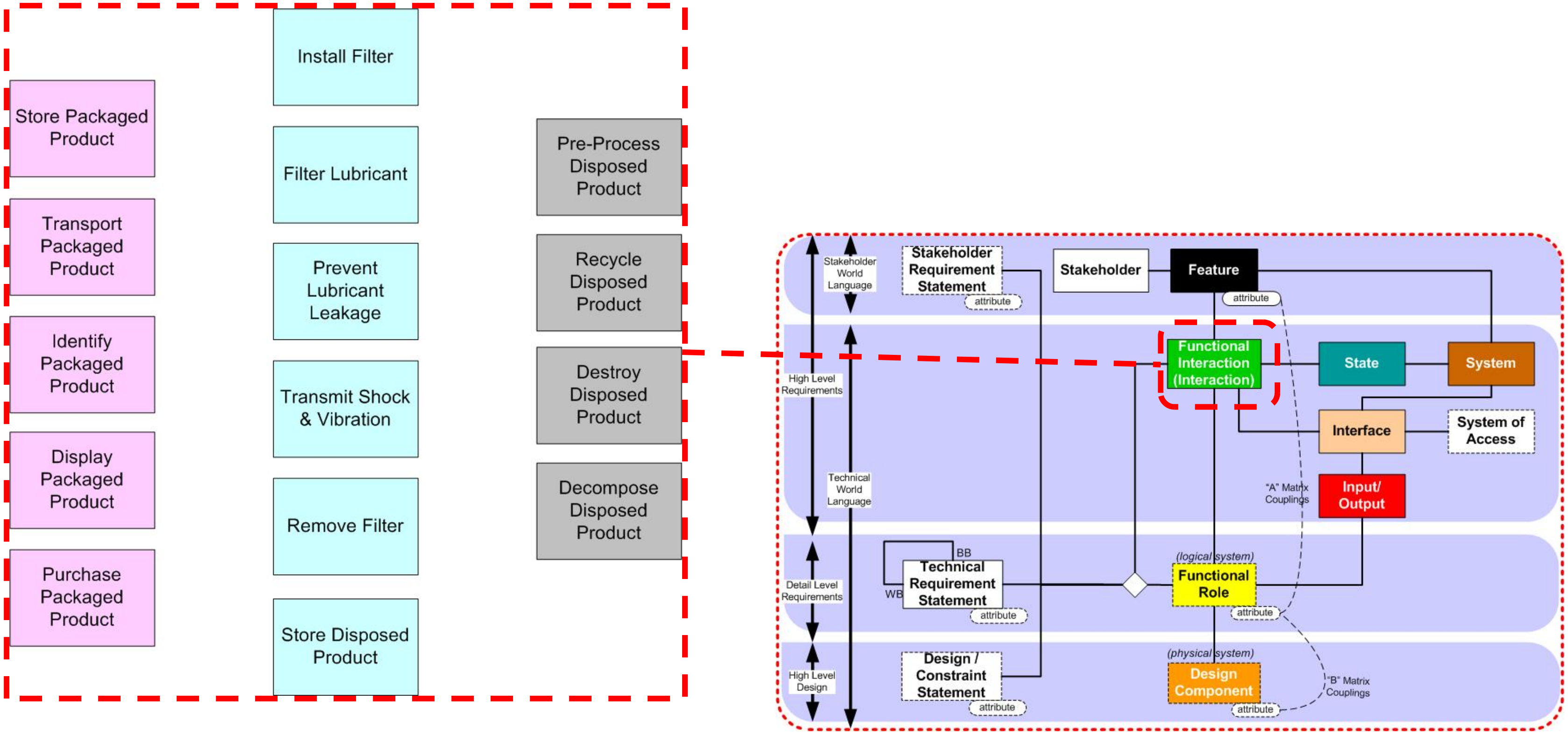
- S* models represent Interactions as explicit objects:
  - Goes to the heart of 300 years of natural science of systems as a foundation for engineering, including emergence.
  - All functional requirements are revealed as external interactions [Schindel].



- Other Metamodel parts: See the References.

Physical Interactions: At the heart of S* models

- S* models represent <u>Physical Interactions</u> as explicit objects:
  - Example: Pattern of Oil Filter Interactions:



Store Packaged Product

Transport Packaged Product

Identify Packaged Product

Display Packaged Product

Purchase Packaged Product

Install Filter

Filter Lubricant

Prevent Lubricant Leakage

Transmit Shock & Vibration

Remove Filter

Store Disposed Product

Pre-Process Disposed Product

Recycle Disposed Product

Destroy Disposed Product

Decompose Disposed Product

Stakeholder World Language

Stakeholder Requirement Statement

attribute

Stakeholder

Feature

attribute

High Level Requirements

Functional Interaction (Interaction)

State

System

Interface

System of Access

Technical World Language

"A" Matrix Couplings

Input/ Output

Detail Level Requirements

BB

WB

Technical Requirement Statement

attribute

(logical system)

Functional Role

attribute

High Level Design

Design / Constraint Statement

attribute

(physical system)

Design Component

attribute

"B" Matrix Couplings

Pattern-based systems engineering (PBSE)

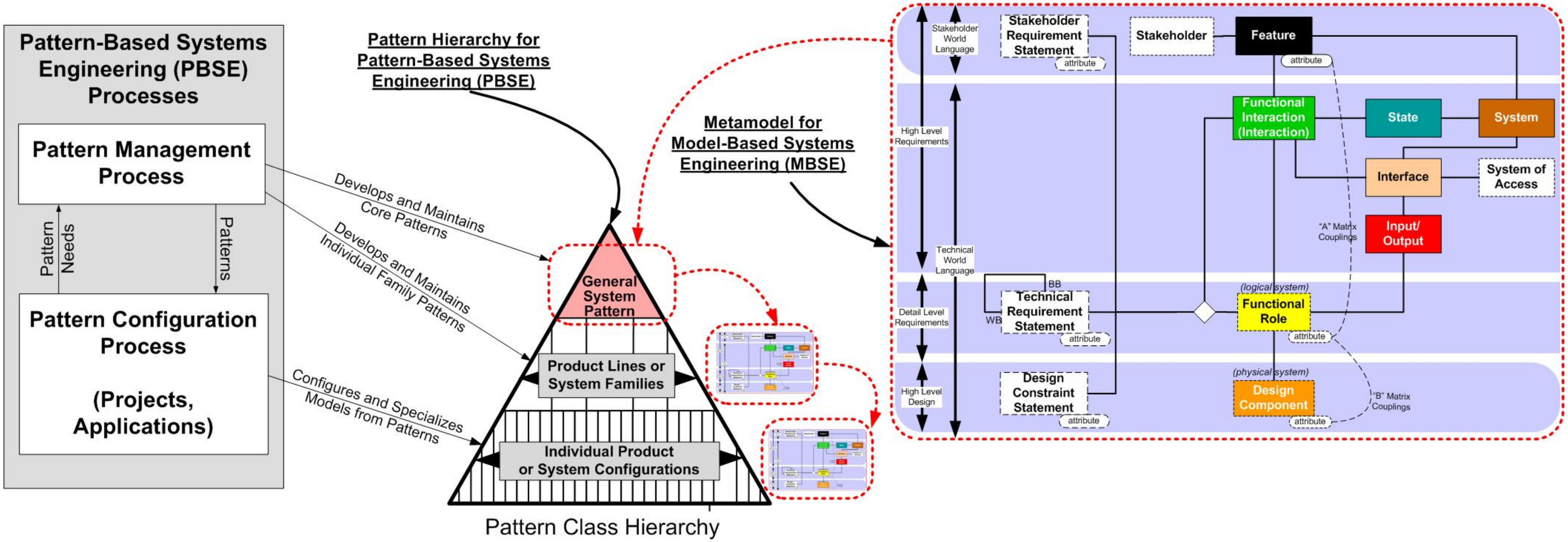
- Model-based Patterns:
  - In this approach, Patterns are reusable, configurable S* models of families (product lines, sets, ensembles) of systems.
- These Patterns are ready to be configured to serve as Models of individual systems in projects.
- Configured here is specifically limited to mean that:
  - Pattern model components are populated / de-populated, and
  - Pattern model attribute (parameter) values are set

. . . both based on configuration rules that are part of the Pattern.

- Patterns are based on the same Metamodel as “ordinary” Models

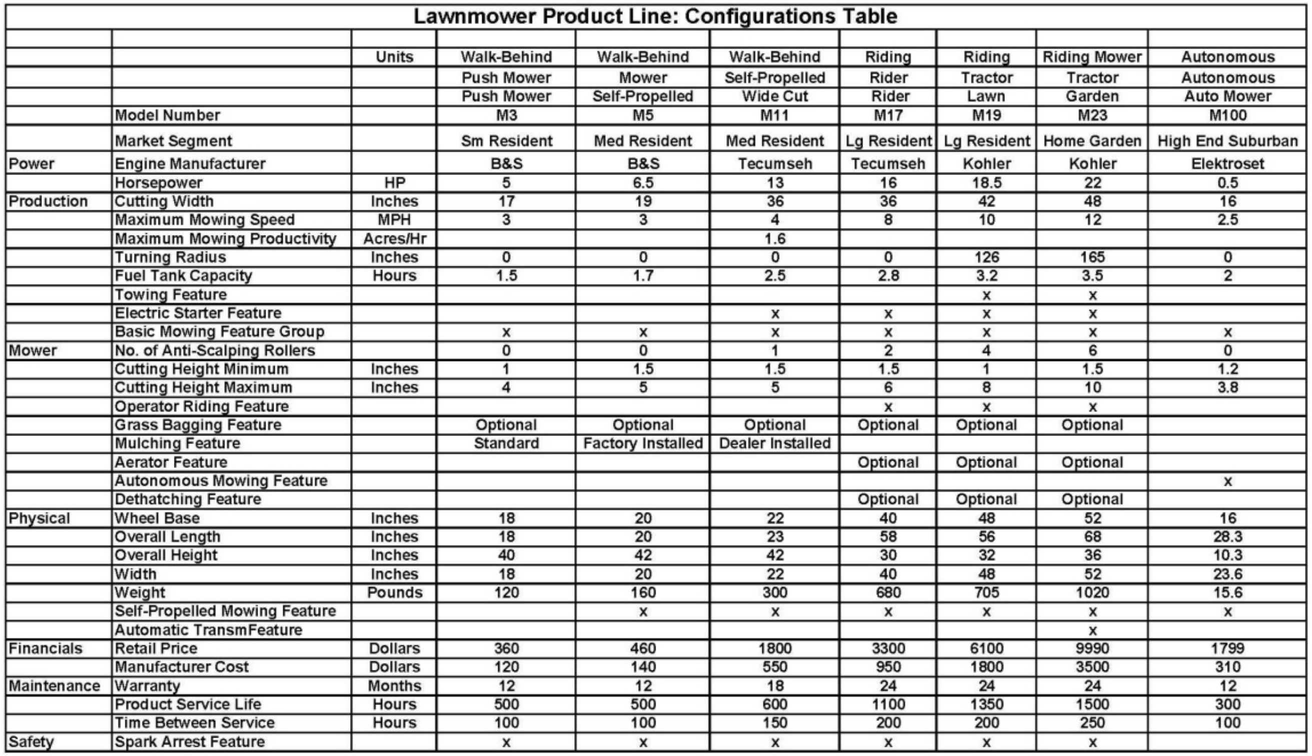
Pattern-based systems engineering (PBSE)

- Pattern-Based Systems Engineering (PBSE) has two overall processes:
  - **Pattern Management Process**: Generates the underlying family model, and periodically updates it based on application project discovery and learning;
  - **Pattern Configuration Process**: Configures the pattern into a specific model for application in a project.


Pattern-Based Systems Engineering (PBSE) Processes
Pattern Management Process
Pattern Needs
Patterns
Pattern Configuration Process
(Projects, Applications)
Develops and Maintains Core Patterns
Develops and Maintains Individual Family Patterns
Configures and Specializes Models from Patterns
Pattern Hierarchy for Pattern-Based Systems Engineering (PBSE)
General System Pattern
Product Lines or System Families
Individual Product or System Configurations
Pattern Class Hierarchy
Metamodel for Model-Based Systems Engineering (MBSE)
Stakeholder World Language
Stakeholder Requirement Statement
attribute
Stakeholder
Feature
attribute
High Level Requirements
Functional Interaction (Interaction)
State
System
Interface
System of Access
"A" Matrix Couplings
Input/ Output
Technical World Language
BB
WB
Detail Level Requirements
Technical Requirement Statement
attribute
(logical system)
Functional Role
attribute
High Level Design
Design Constraint Statement
attribute
(physical system)
Design Component
attribute
"B" Matrix Couplings

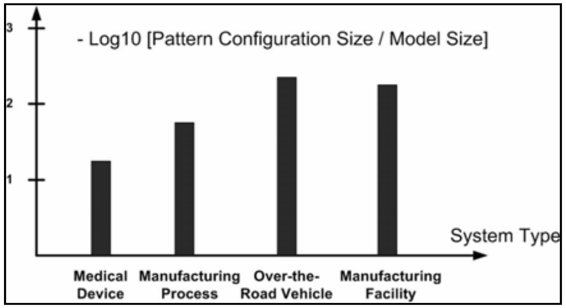
Pattern configurations

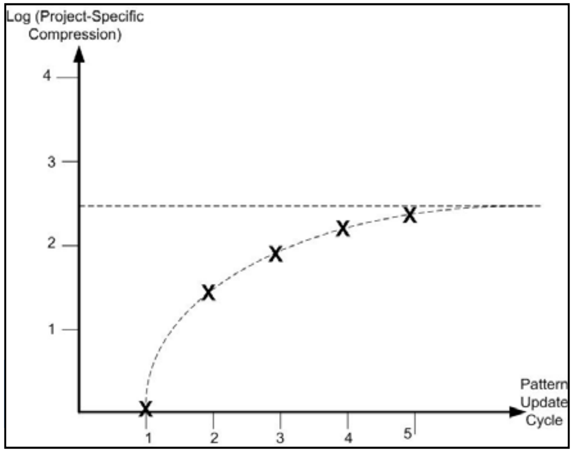
- A table of configurations illustrates how patterns facilitate compression;
- Each column in the table is a compressed system representation with respect to ("modulo") the pattern;
- The compression is typically very large;
- The compression ratio tells us how much of the pattern is variable and how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



Lawnmower Product Line: Configurations Table

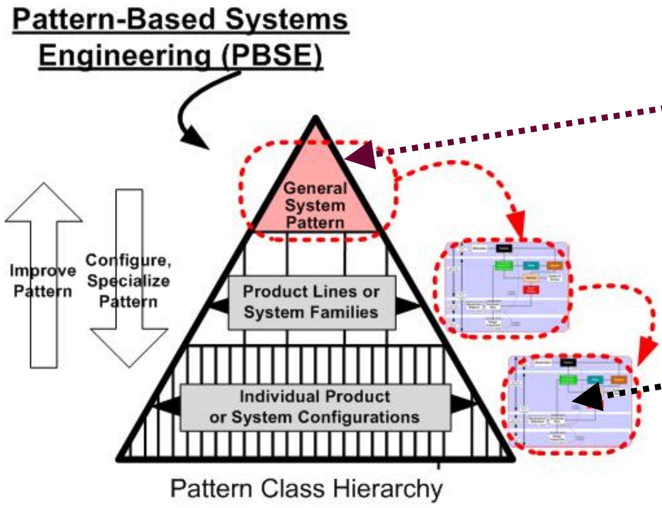
		Units	Walk-Behind	Walk-Behind	Walk-Behind	Riding	Riding	Riding Mower	Autonomous
			Push Mower	Mower	Self-Propelled	Rider	Tractor	Tractor	Autonomous
			Push Mower	Self-Propelled	Wide Cut	Rider	Lawn	Garden	Auto Mower
	Model Number		M3	M5	M11	M17	M19	M23	M100
	Market Segment		Sm Resident	Med Resident	Med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





Checking holistic alignment to a pattern

- Gestalt Rules express what is meant by holistic conformance to a pattern:
  - – Expressing regularities of whole things, versus same "parts"

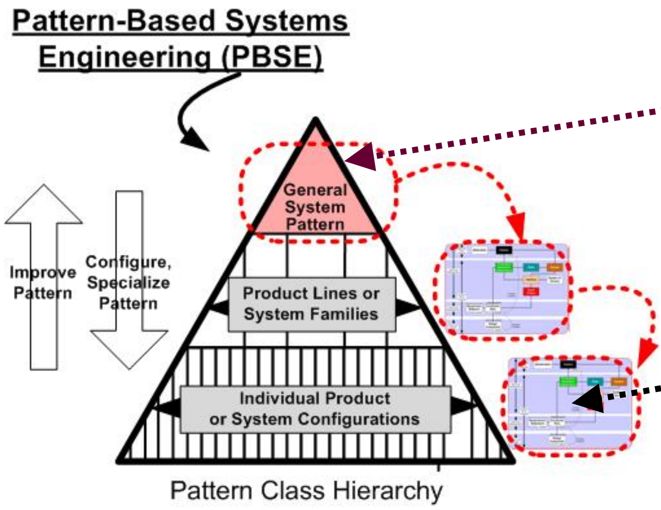

Pattern-Based Systems Engineering (PBSE)
General System Pattern
Improve Pattern
Configure, Specialize Pattern
Product Lines or System Families
Individual Product or System Configurations
Pattern Class Hierarchy

Governing pattern

Candidate model configuration—does it conform to pattern?

The Gestalt Rules

1. Every component class in the candidate model must be a subclass of a parent superclass in the pattern—no “orphan classes”.
2. Every relationship between component classes must be a subclass of a parent relationship in the pattern, and which must relate parent superclasses of those same component classes—no “orphan relationships”.
3. Refining the pattern superclasses and their relationships is a permissible way to achieve conformance to (1) and (2).

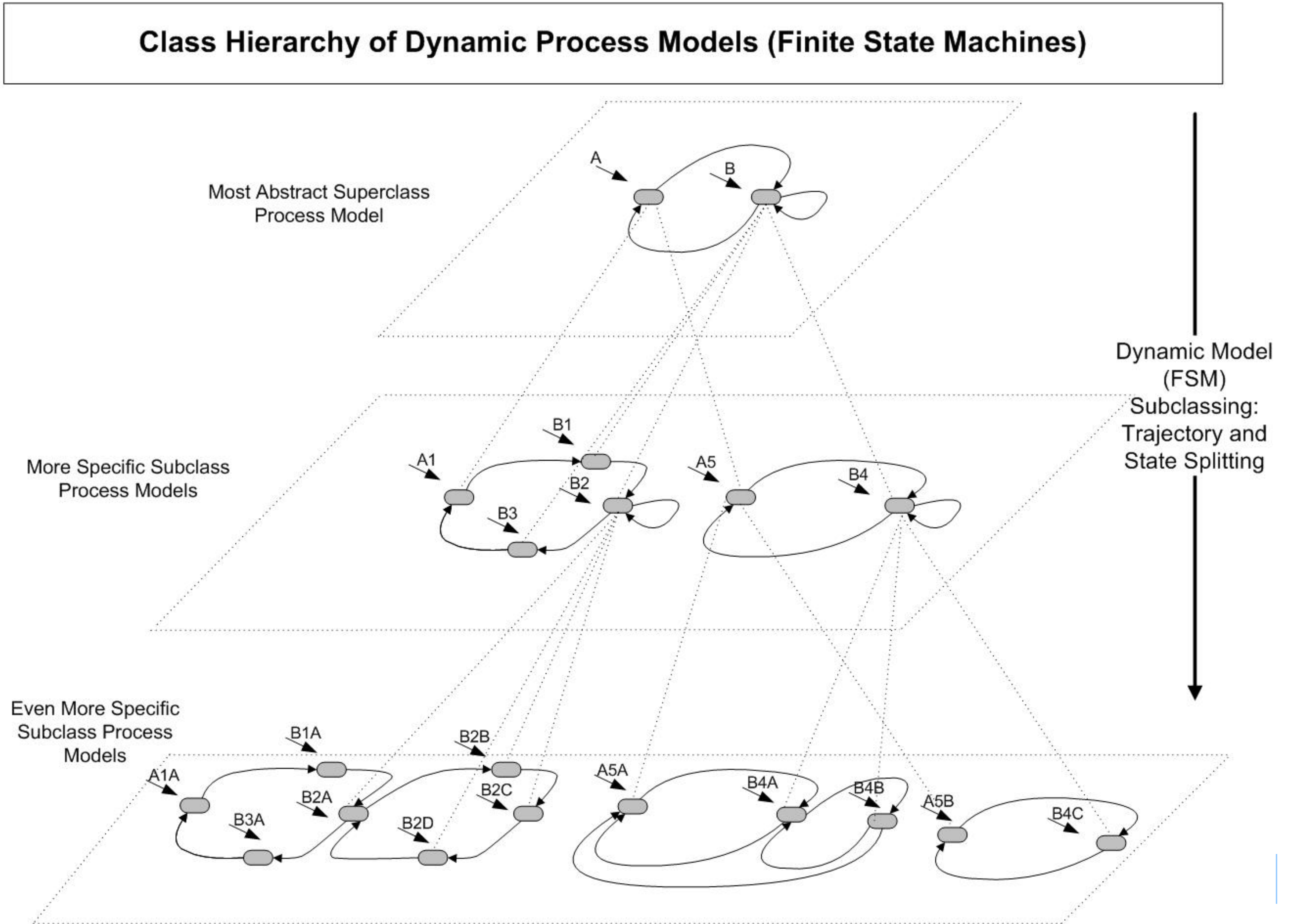

Pattern-Based Systems Engineering (PBSE)
General System Pattern
Improve Pattern
Configure, Specialize Pattern
Product Lines or System Families
Individual Product or System Configurations

Pattern Class Hierarchy

Governing pattern

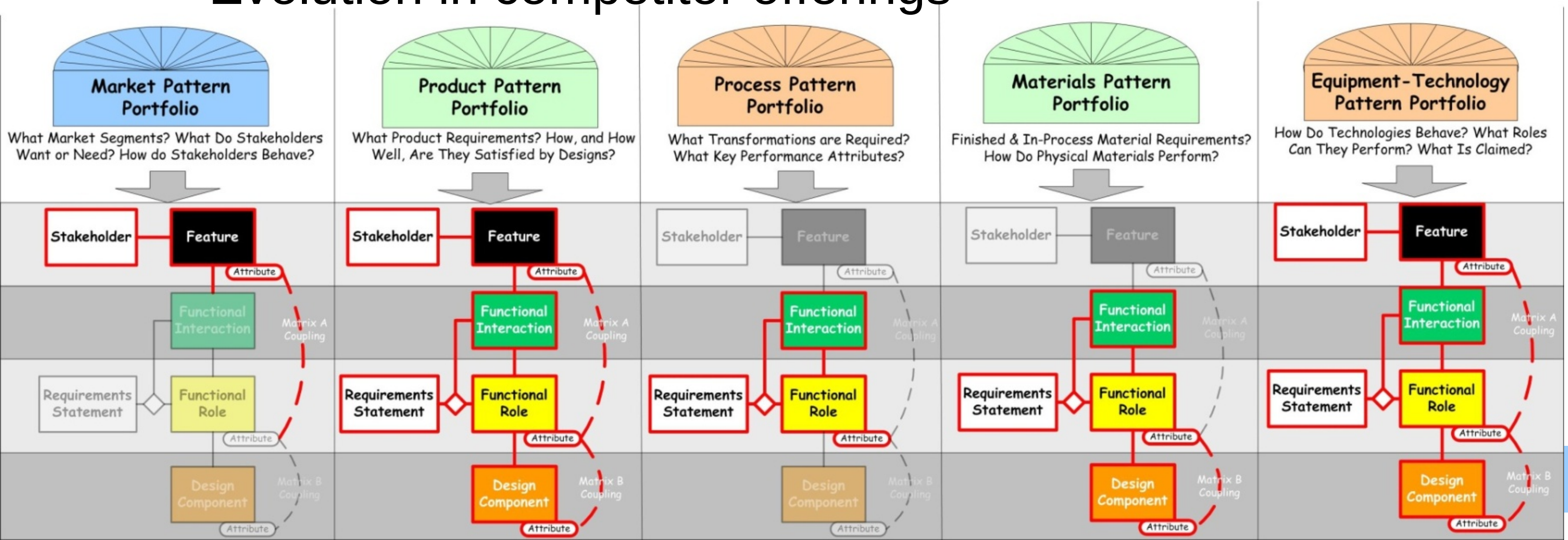
Candidate model configuration—does it conform to pattern?

Example: State Model Pattern—illustrates how *visual* is the "class splitting" and "relationship rubber banding" of the Gestalt Rules


Class Hierarchy of Dynamic Process Models (Finite State Machines)
Most Abstract Superclass Process Model
A
B
More Specific Subclass Process Models
A1
B1
B2
B3
A5
B4
Even More Specific Subclass Process Models
A1A
B1A
B2A
B3A
B2B
B2C
B2D
A5A
B4A
B4B
A5B
B4C
Dynamic Model (FSM) Subclassing: Trajectory and State Splitting

Leveraging adaptability to tame uncertainty

- Patterns express “envelopes” around “point situations”.
- Patterns help us discover, explore, and record what we may have to adapt to, along with adaptation plans:
  - Evolution in available technologies and parts
  - Evolution in system requirements, interfaces, modes, etc.
  - Evolution in the larger systems in which we operate
  - Evolution in customer or market expectations
  - Evolution in competitor offerings



Market Pattern Portfolio

What Market Segments? What Do Stakeholders Want or Need? How do Stakeholders Behave?

Product Pattern Portfolio

What Product Requirements? How, and How Well, Are They Satisfied by Designs?

Process Pattern Portfolio

What Transformations are Required? What Key Performance Attributes?

Materials Pattern Portfolio

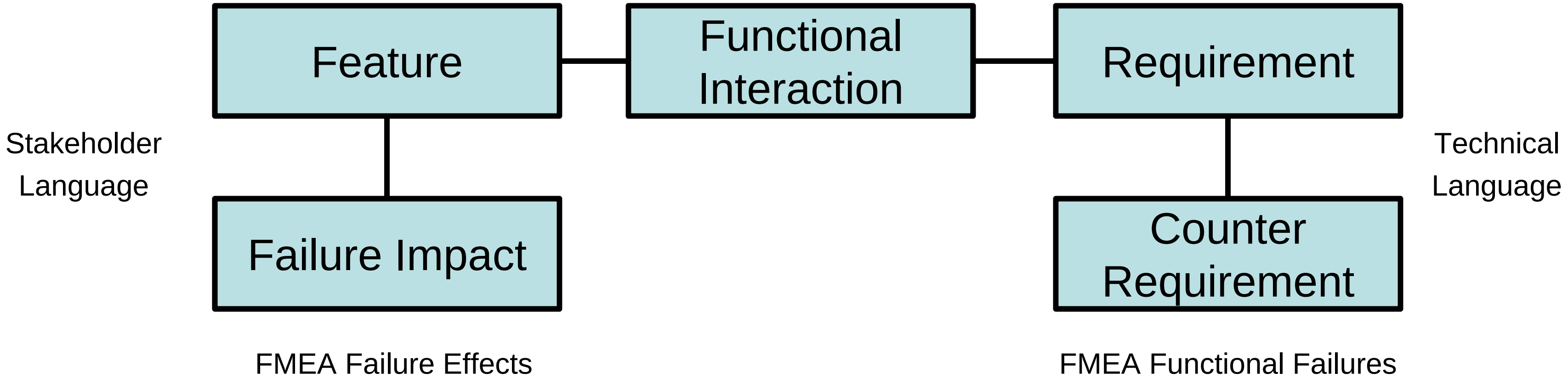
Finished & In-Process Material Requirements? How Do Physical Materials Perform?

Equipment-Technology Pattern Portfolio

How Do Technologies Behave? What Roles Can They Perform? What Is Claimed?

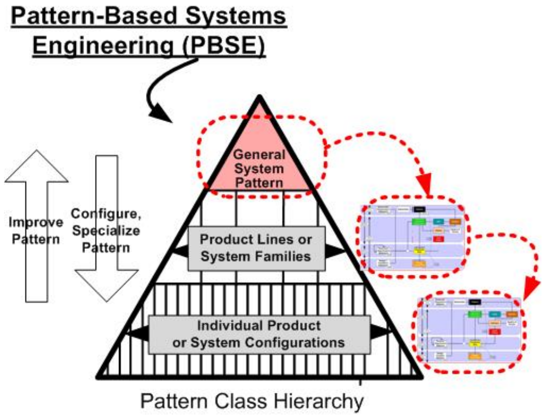
Leveraging adaptability to tame uncertainty

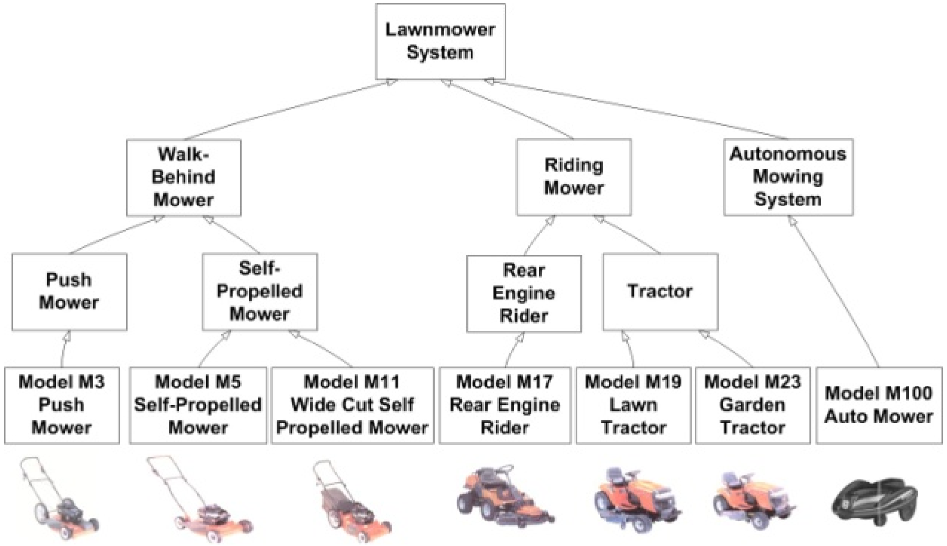
- Patterns also express risks and mitigations for:
  - Patterns of system failure modes and effects (d-FMEA)
  - Patterns of operator failure modes and effects (a-FMEA)
  - Patterns of production & distribution failures (p-FMEA)



Leveraging adaptability to tame uncertainty

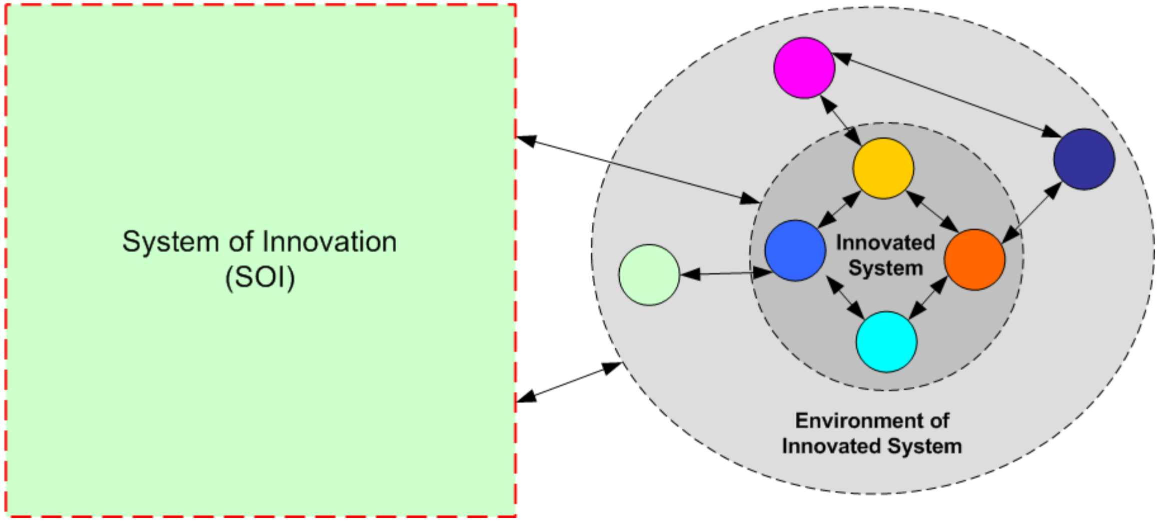
- Descriptions of SE processes typically appear to describe engineering a “new” system “from scratch” [e.g., ISO 15288, INCOSE SE Handbook]:
  - However, real projects are often concerned with engineering similar (but different) systems across different product generations, applications, configurations, or market segments.
  - Patterns provide the IP basis to make Platform Management a discipline, not just an attractive idea:


Pattern-Based Systems Engineering (PBSE)
Improve Pattern
Configure, Specialize Pattern
General System Pattern
Product Lines or System Families
Individual Product or System Configurations
Pattern Class Hierarchy



Leveraging adaptability to tame uncertainty

- The science behind the emergence of re-usable building blocks in natural innovation:
  - The Systems of Innovation Project of the INCOSE System Science Working Group
  - Next meeting will be at IW2012.
  - https://sites.google.com/site/syssciwg/



Conclusions

1. Patterns abound in the world of systems engineering.
2. These patterns extensively impact our projects, whether we take advantage of them as Explicit Patterns, or we are negatively impacted by Dark Patterns.
3. Pattern-Based Systems Engineering (PBSE) offers specific ways to extend MBSE to exploit Patterns.
4. MBSE comes first—Patterns without Models is like orbital mechanics before Newton.
5. We've had good success applying pattern-based methods in mil/aerospace, automotive, medical/health care, advanced manufacturing, and consumer product domains.

References

1. W. Schindel, “Innovation as Emergence: Hybrid Agent Enablers for Evolutionary Competence” in *Complex Adaptive Systems*, Volume 1, Cihan H. Dagli, Editor in Chief, Elsevier, 2011
2. W. Schindel, S. Peffers, J. Hanson, J. Ahmed, W. Kline, “All Innovation is Innovation of Systems : An Integrated 3-D Model of Innovation Competencies ”, Proc. of ASEE 2011 Conference (2011).
3. ISO/IEC 15288: Systems Engineering—System Life Cycle Processes. International Standards Organization (2008).
4. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, Version 3.2, International Council on Systems Engineering (2010).
5. Kahneman, D., *Thinking, Fast and Slow*, Farrar, Straus and Giroux, Publishers, 2011, ISBN**-10:** 0374275637.
6. Lewis, Michael, *Moneyball: The Art of Winning an Unfair Game*, Norton, New York, 2004.
7. NASA on dark matter http://science.nasa.gov/astrophysics/focus-areas/what-is-dark-energy/
8. *NASA Systems Engineering Handbook,* NASA/SP-2007-6105, Rev 1, U.S. National Aeronautics and Space Administration (2007).
9. W. Schindel, “Requirements statements are transfer functions: An insight from model-based systems engineering”, Proceedings of INCOSE 2005 International Symposium, (2005).
10. W. Schindel, “What Is the Smallest Model of a System?”, Proc. of the INCOSE 2011 International Symposium, International Council on Systems Engineering (2011).
11. B. Van Fraassen, *Scientific Representation: Paradoxes of Perspective*, Oxford U Press, 2008.
12. W. Schindel, “Pattern-Based Systems Engineering: An Extension of Model-Based SE”, INCOSE IS2005 Tutorial TIES 4, (2005).
13. J. Bradley, M. Hughes, and W. Schindel, “Optimizing Delivery of Global Pharmaceutical Packaging Solutions, Using Systems Engineering Patterns” Proceedings of the INCOSE 2010 International Symposium (2010).
14. W. Schindel, and V. Smith, “Results of applying a families-of-systems approach to systems engineering of product line families”, SAE International, Technical Report 2002-01-3086 (2002).
15. D. Williams, “How Concepts of Self-Regulation Explain Human Knowledge”, *The Bent of Tau Beta Pi*, Winter (2011) 16-21.
16. INCOSE System Sciences Working Group web site: https://sites.google.com/site/syssciwg/home

Speaker

Bill Schindel (schindel@ictt.com) is president of ICTT System Sciences (www.ictt.com), a systems engineering company, and developer of the Systematica™ Methodology for model and pattern-based systems engineering. His 40-year engineering career began in mil/aero systems with IBM Federal Systems, Owego, NY, included service as a faculty member of Rose-Hulman Institute of Technology, and founding of three commercial systems-based enterprises.

He has consulted on improvement of engineering processes within automotive, medical/health care, manufacturing, telecommunications, aerospace, and consumer products businesses. Schindel earned the BS and MS in Mathematics, and was awarded an Hon. D.Eng by Rose-Hulman Institute of Technology for his systems engineering work. At the 2005 INCOSE International Symposium, he was recognized as the author of the outstanding paper on Modelling and Tools. Bill is an INCOSE CSEP, and president-elect of the Crossroads of America INCOSE chapter.

V 1.2.1