The Agile Systems Pattern

A Reference Model for Agility in Systems

Bill Schindel, ICTT System Sciences

schindel@ictt.com

Ecosystem | Education | Health Care | Information | Manufacturing | Transportation

Great Lakes Regional Conference 2015

V1.2.1

Abstract

- Human-engineered and other systems may be under pressure to adapt--whether they encounter new opportunities or are threatened by commercial competition, living predators or ecological competitors, physical military attack, or cyber threat, or other changes in their environment. The ongoing ability of individual systems or system families to adapt well enough as conditions change, especially in the presence of uncertainty about future conditions, is a highly-valued capability that may determine prosperity, lifespan, or survival. Systems (including developmental and other life cycle systems) that can adapt well enough, in terms of time, cost, and effectiveness, are sometimes referred to as “agile systems”. When the rate of environmental change or uncertainty increases, this sort of agility can become a basic framework for survival, competitive success, or failure.
- Reviewing and extending the work of earlier pioneers, this presentation should be of interest to those who are responsible for planning, designing, or analyzing systems with enhanced agility, including products and services, as well as development, manufacturing, operational, and other life cycle processes, and to those who will lead, execute, use, manage, or acquire them. Attendees should expect to become aware of an MBSE-based reference architecture that can serve as a tool for accomplishing these challenging tasks, and also learn more about the INCOSE Agile Systems Engineering Life Cycle Model (ASELCM) Project.

Is this your tomorrow, or a distant vision?

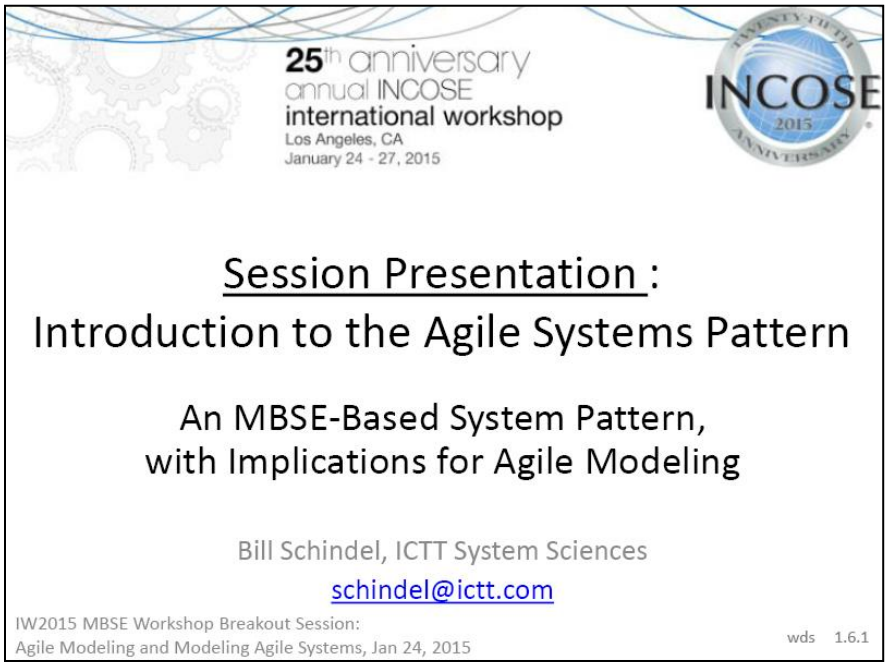
From "The Hardware Renaissance Arrives: A New Dawn for Gadgets", *The Wall Street Journal*, March 23, 2015:

"Recently, as I gazed into the prototype of a smart breast pump, I had a vision of the future. I saw an age in which new products—actual, physical electronics products—will go from idea to store shelves in a matter of months. A future in which warehouses and distribution centers cease to exist, because factories produce finished goods from raw materials on demand, and they never stop moving through the supply chain. *Only it turns out all of this is possible today*. The "hardware renaissance" that began in Silicon Valley in just the last five years, born of rapid prototyping technologies, has become something much larger and more important. It has been a sea change in every stage of producing physical objects, from idea to manufacturing to selling at retail . . ."

-- Christopher Mims, *The Wall Street Journal*, p B1,6, March 23, 2015

-- emphasis added

Agile Systems Engineering Life Cycle (ASELC) Pattern

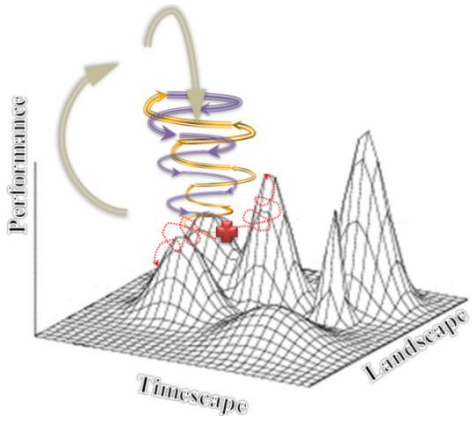


See --

http://www.omgwiki.org/MBSE/doku.php?id=mbse:incose_mbse_iw_2015:breakout_out_session_agile_modeling

Digest of the ASELC IW2015 MBSE Workshop Materials

- What are Agile Systems, and why do they matter?
- How are Agile Systems related to MBSE?
- What is the INCOSE Agile Systems Engineering Life Cycle Model Discovery Project?
- What is the Agile Systems Engineering Life Cycle Pattern?
- Health Care: Example Agile System Reference Boundaries
- What is the INCOSE Patterns Challenge Team?
- Where can I learn more?
- Discussion
- Attachments



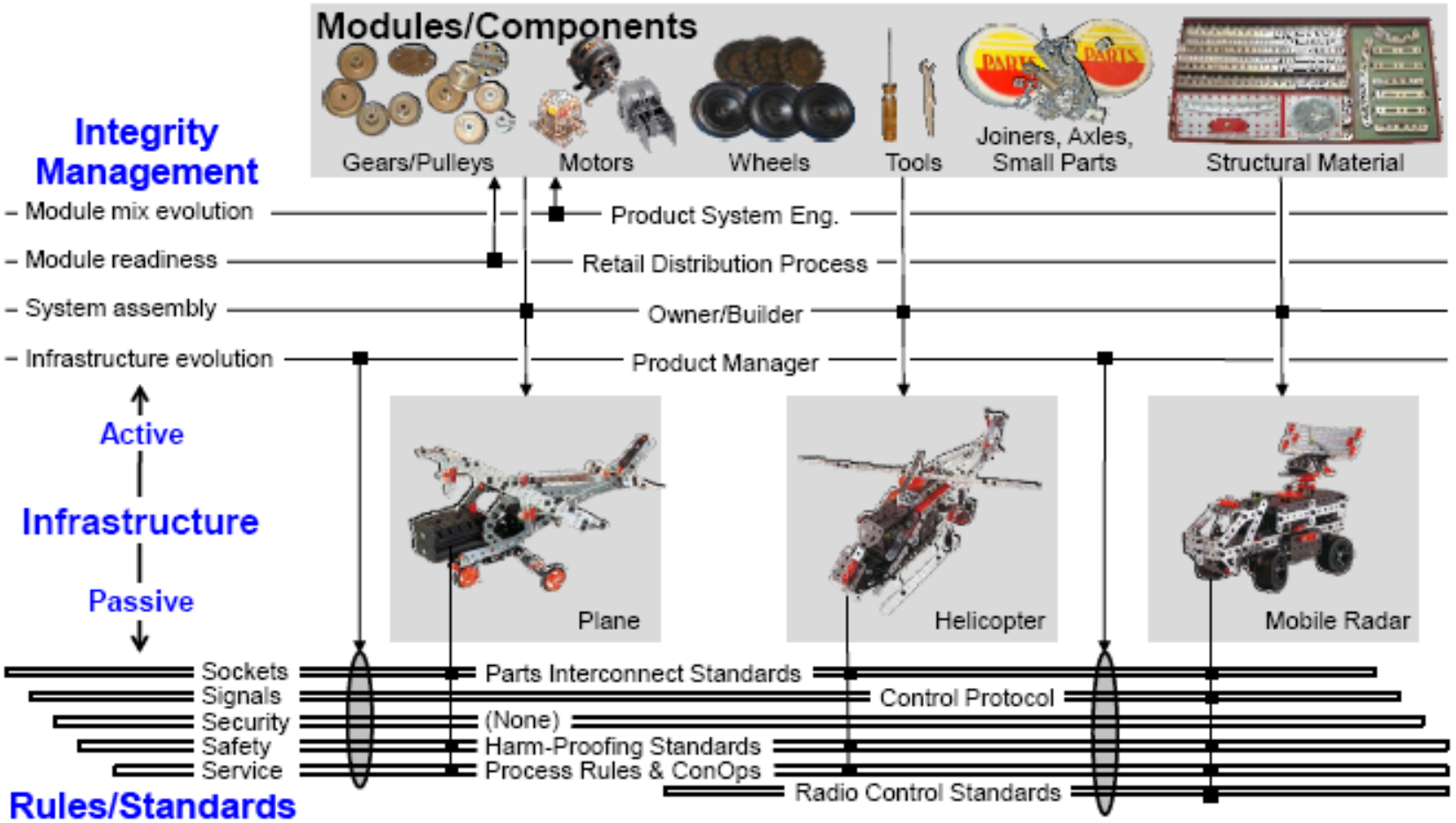
What are Agile Systems, and why do they matter?

- A longer history than just Agile Methods in software development:
  - See Dove and LaBarge, 2014
  - Multiple streams of thought
  - Subject of the INCOSE ASELCM Project

Agile Systems Informal Pattern (R. Dove)

The S*ASELCM Pattern captures (in a formal S*Model) the key ideas associated with the pre-MBSE Agile System Architecture:

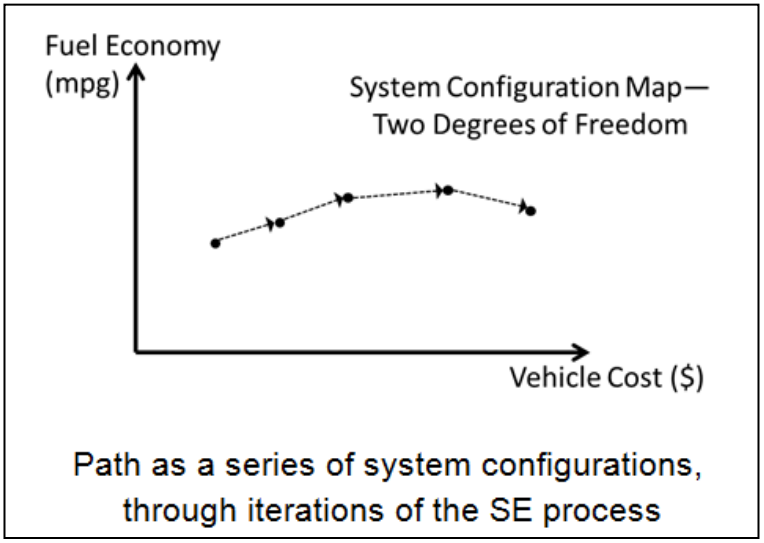
– As in (Dove and LaBarge, 2014)



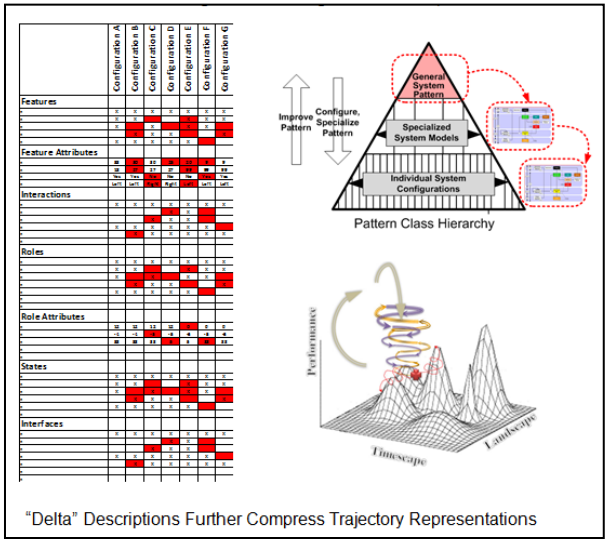
System Life Cycle Trajectories in S*Space

- Configurations change over life cycles, during development and subsequently
- Trajectories (configuration paths) in S*Space
- Effective tracking of trajectories
- History of dynamical paths in science and math
- Differential path representation: compression, equations of motion

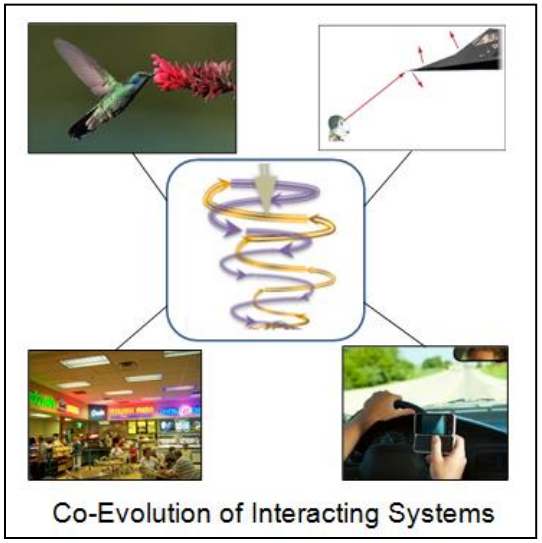
Evolving Systems Over Multiple Life Cycles



Path as a series of system configurations, through iterations of the SE process

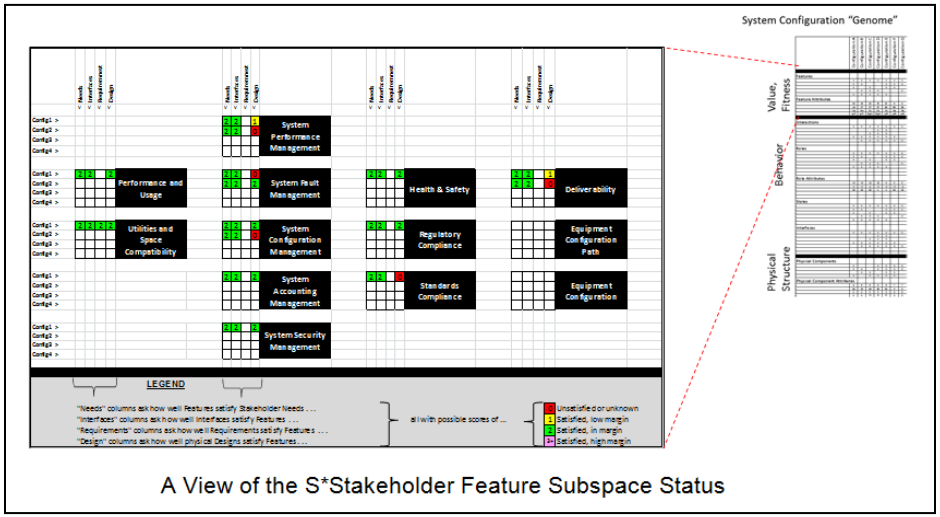


"Delta" Descriptions Further Compress Trajectory Representations

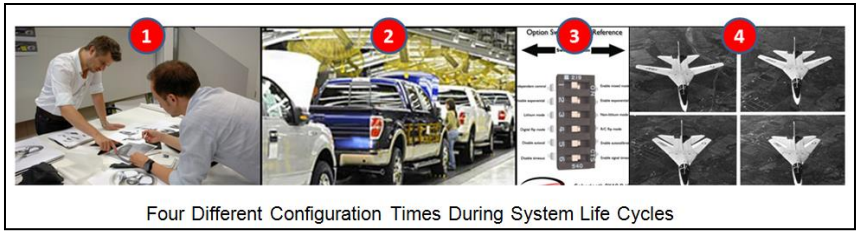


Co-Evolution of Interacting Systems

System Life Cycle Trajectories in S*Space

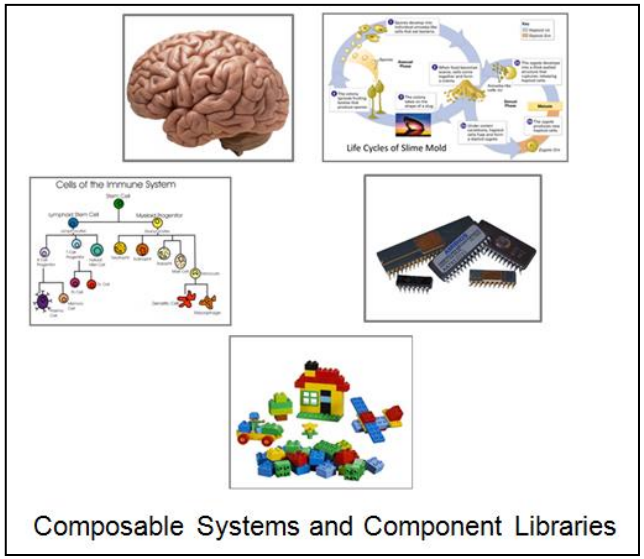


A View of the S*Stakeholder Feature Subspace Status



Four Different Configuration Times During System Life Cycles

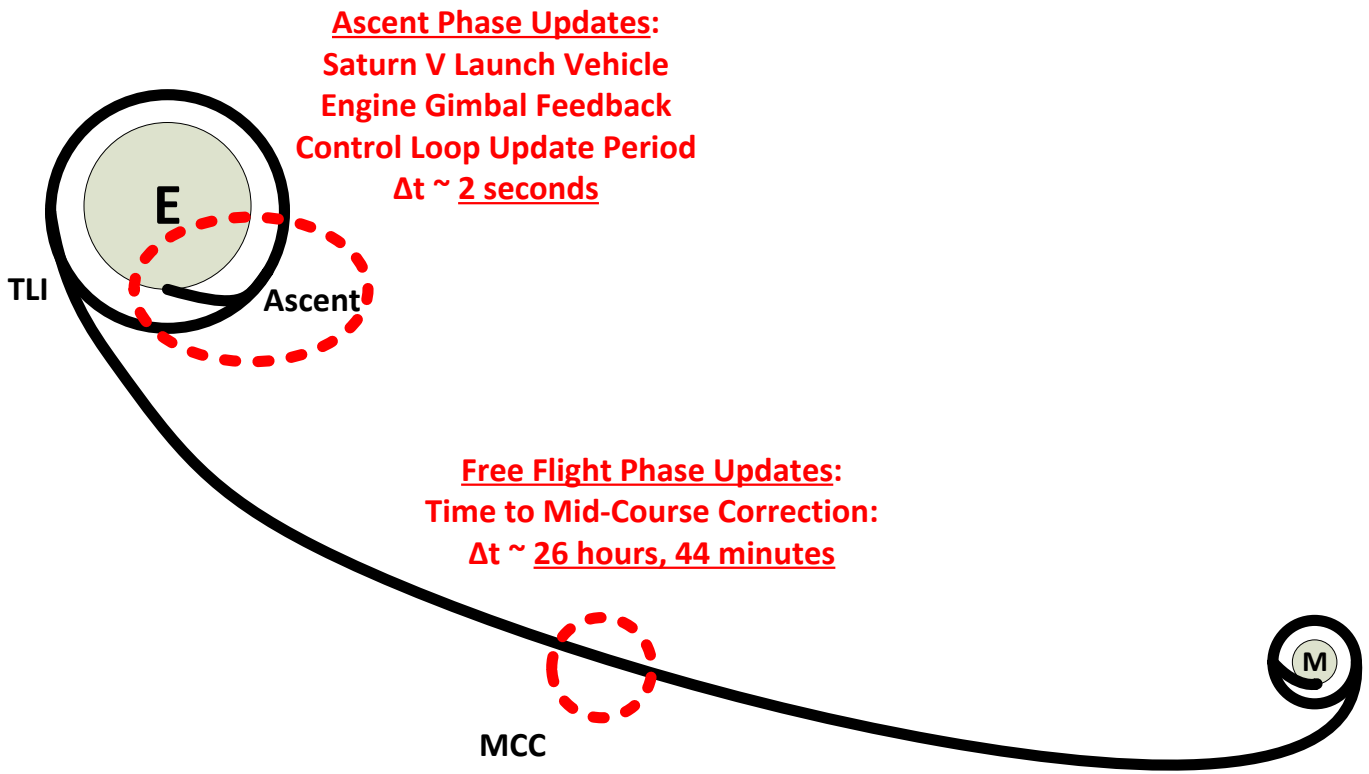
- There are productive "views" of those trajectories, which may be implemented on most any general systems modeling tool or PLM system:
  - a risk management application of SE tracing—
  - projecting detected gaps onto Stakeholder Feature space to understand their significance.
  - productive "views" help know "where" we are, and manage trajectory direction, critical to scrum empiricism
- Progressive advances in configurability:
  - Deferred times of reconfiguration (but S*Space applies to all of them!)
  - Addition of information to architecture
  - Composable architecture



Composable Systems and Component Libraries

Feedback & Correction Cycle Rate: A Hallmark of Agile Methods

An Apollo 11 Mission Question: Why was the Saturn V rocket engines' directional gimbals update cycle period throughout the Ascent Phase ~ 2 seconds, but the update cycle period of course direction during the Free Flight Phase was ~ 26 hours?



How are Agile Systems Related to MBSE?

1. **Basics**: Using explicit models, MBSE/PBSE adds clarity to pre-model descriptions of Agile Systems and Agile SE-- improves understanding of Agile Systems.
2. **More important**: MBSE/PBSE complements and improves the capability of Agile Systems and Agile Systems Engineering—
   - Agility requires persistent memory & learning—*being forgetful/not learning impacts agility*.
   - Patterns capture & retain learning, as persistent, re-usable, configurable, models, *updated as experience accumulates*.
   - S*Patterns are configurable, reusable S*Models.

"*PBSE as Agile MBSE*" emerges as essential when competing on agility becomes reality for competing, competent players:

- Improved: "Where are we?"
- Improved: "Where are we going?"
- Improved: "We've been here before."
- Improved: Understanding of response.
- Improved: Understanding of mission envelopes.
- Improved: Ability to assess agility
- Improved: Ability to plan agility

Vital for Scrum, other approaches

Vital for Response Situation Analysis (RSA)

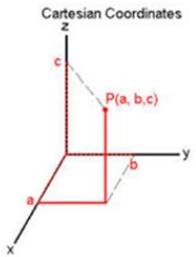
Maps *vs*. Itineraries -- SE Information *vs*. SE Process

Itinerary ≠ Map!

(What am I doing?) *(Where am I?)*

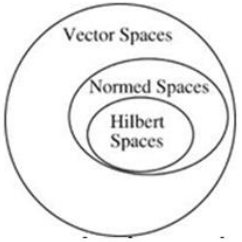
When they eventually did emerge, maps represented a newer idea of the nature of "where".

- The SE Process consumes and produces information.
- But, SE historically emphasizes process over information. (Evidence: Ink & effort spent describing standard process versus standard information.)
- Ever happen?-- Junior staff completes all the process steps, all the boxes are checked, but outcome is not okay.
- Recent discoveries about ancient navigators: Maps *vs*. Itineraries.
- The geometrization of Algebra and Function spaces (Descartes, Hilbert)
- Knowing where you are, not just what you are doing.
- Knowing where you are going, not just what you are doing.
- Distance metrics, inner products, projections, decompositions.



Rene Descartes
1596 - 1650

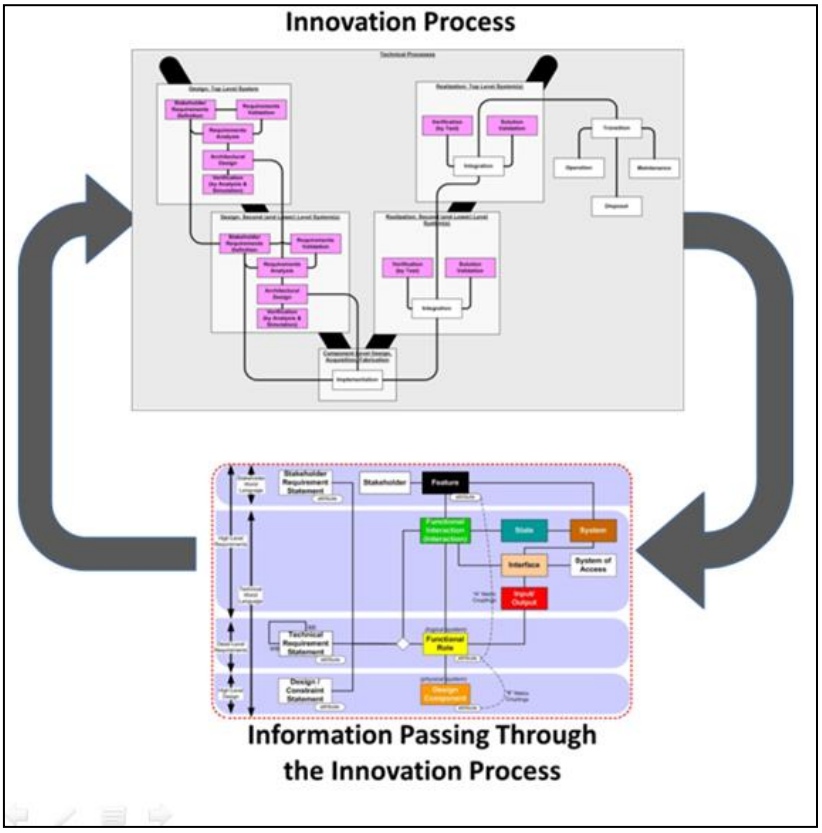
Geometrization of Algebra, by Rene Descartes

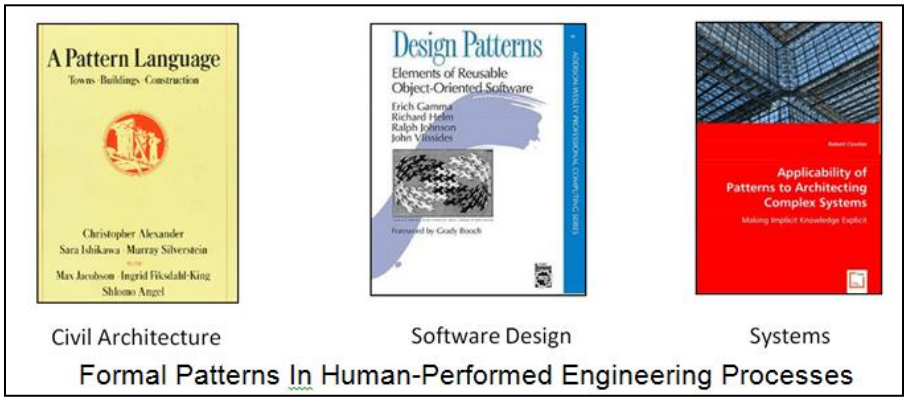


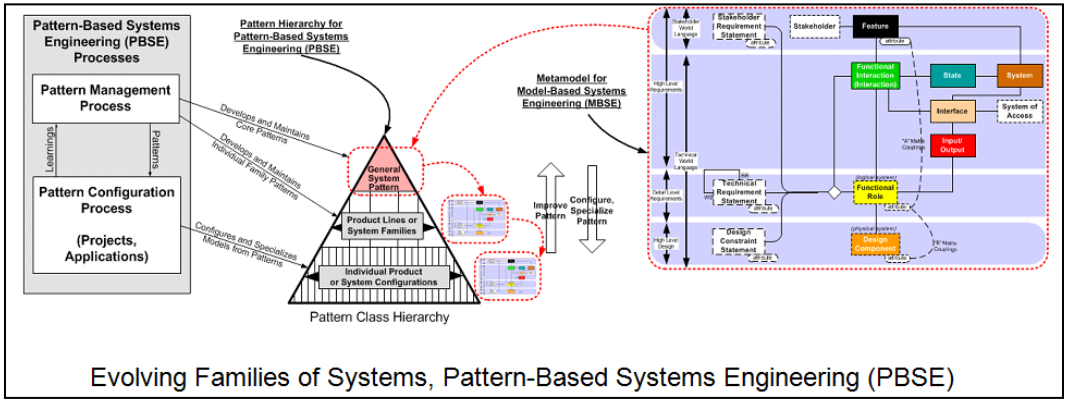
David Hilbert
1862 - 1943

Geometrization of Function Space, by David Hilbert

Maps *vs*. Itineraries -- SE Information *vs*. SE Process



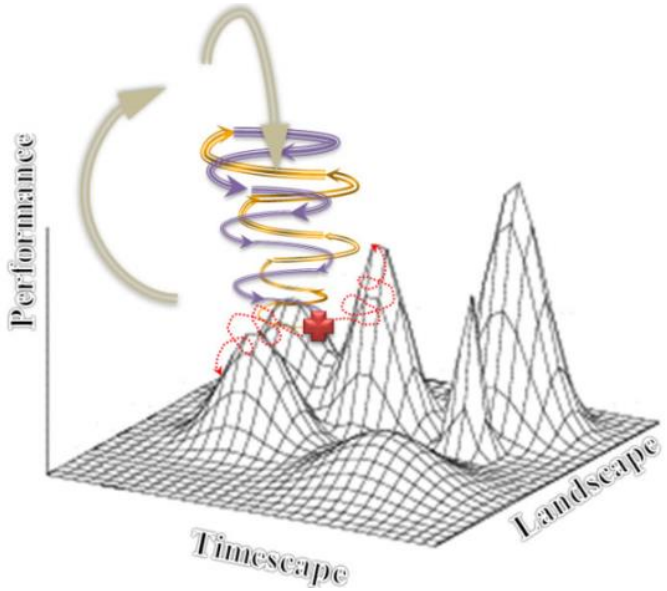




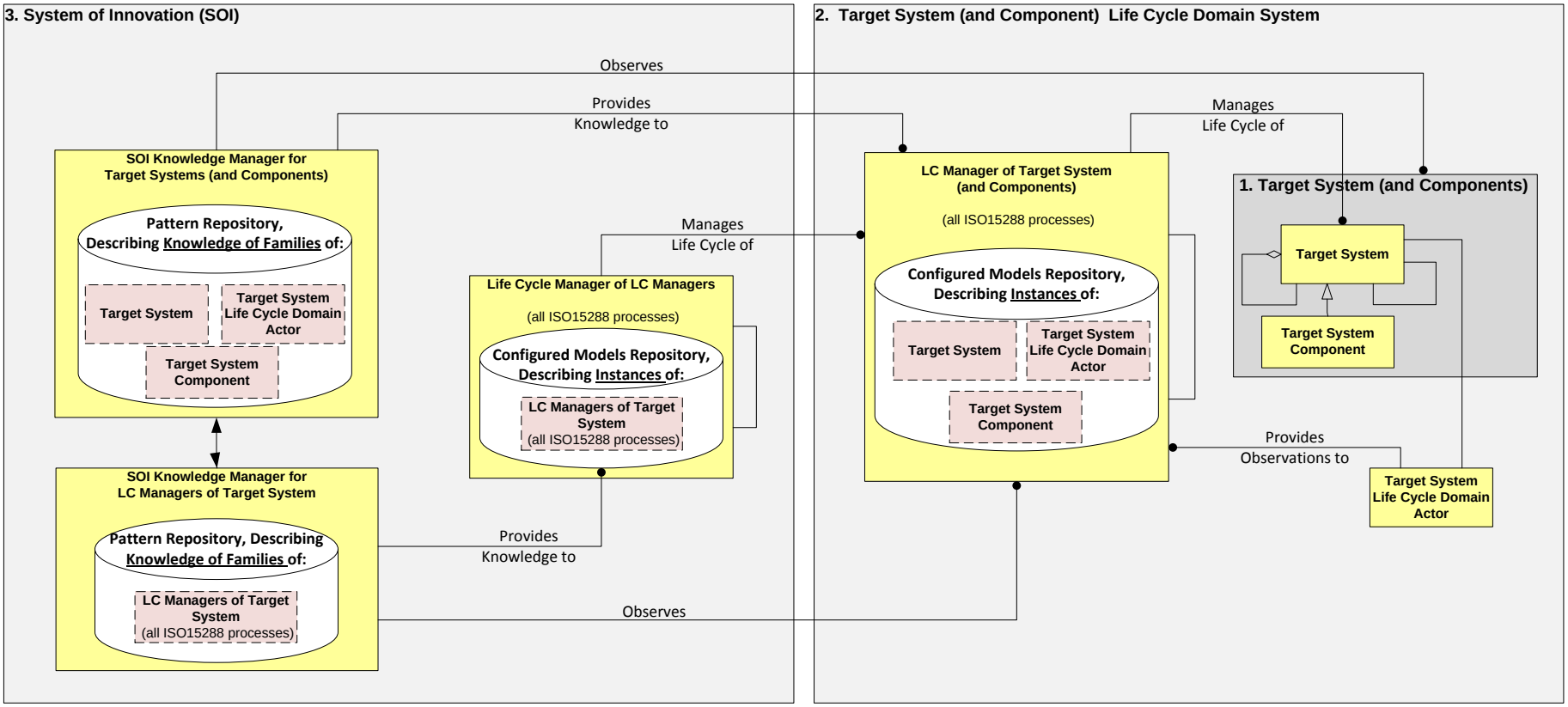
- Model-based Patterns in S*Space.
- Interactions as the basis of all laws of physical sciences.
- Relationships, not procedures, are the fruits of science used by engineers: Newton's laws, Maxwell's Equations.
- Immediate connection to Agility: knowing where you are--starting with better definition of what "where" means. There is a minimal "genome" (S*Metamodel) that provides a practical way to capture, record, and understand—the "smallest model of a system".
- Not giving up process: MBSE/PBSE version of ISO/IEC 15288.

What is the INCOSE Agile Systems Engineering Life Cycle Model Discovery Project?

http://www.parshift.com/ASELCM/Home.html



What is the Agile Systems Engineering Life Cycle Pattern?



- A key subset of the ASELCM Pattern:

  the system reference boundaries . . .

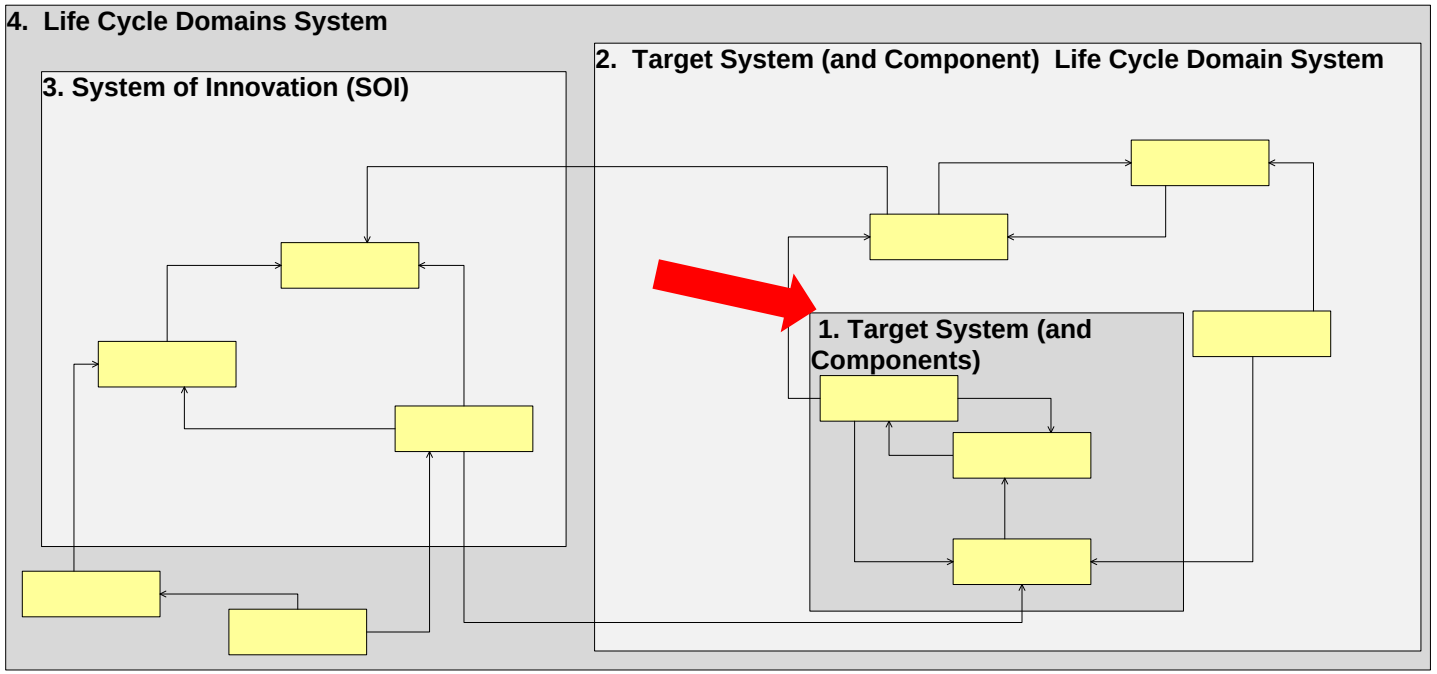
The Agile System Domain Model

- We will particularly refer to **four major system boundaries**:
  - To avoid a confusion bog of loaded terms, we could have just named them “System 1”, “System 2”, “System 3”, and “System 4” and proceeded to define them behaviorally.
  - The definitions are behavioral because these are logical systems, performing defined roles.
  - However, we will also give them more specific names — but make sure you understand the definitions of these systems, which are more important than their names . . .

The Agile System Domain Model

System 1: The **Target System (and Components):** (Definition) The logical system of interest, which results from, or is subject to, innovation.

- Its behavior, characteristics, or performance are targets of the innovation (change, adaptation) process we'll introduce later.
- It is potentially agile.
- Examples include aircraft, satellites, the human immune system, restaurants, birds, and the health care delivery system.

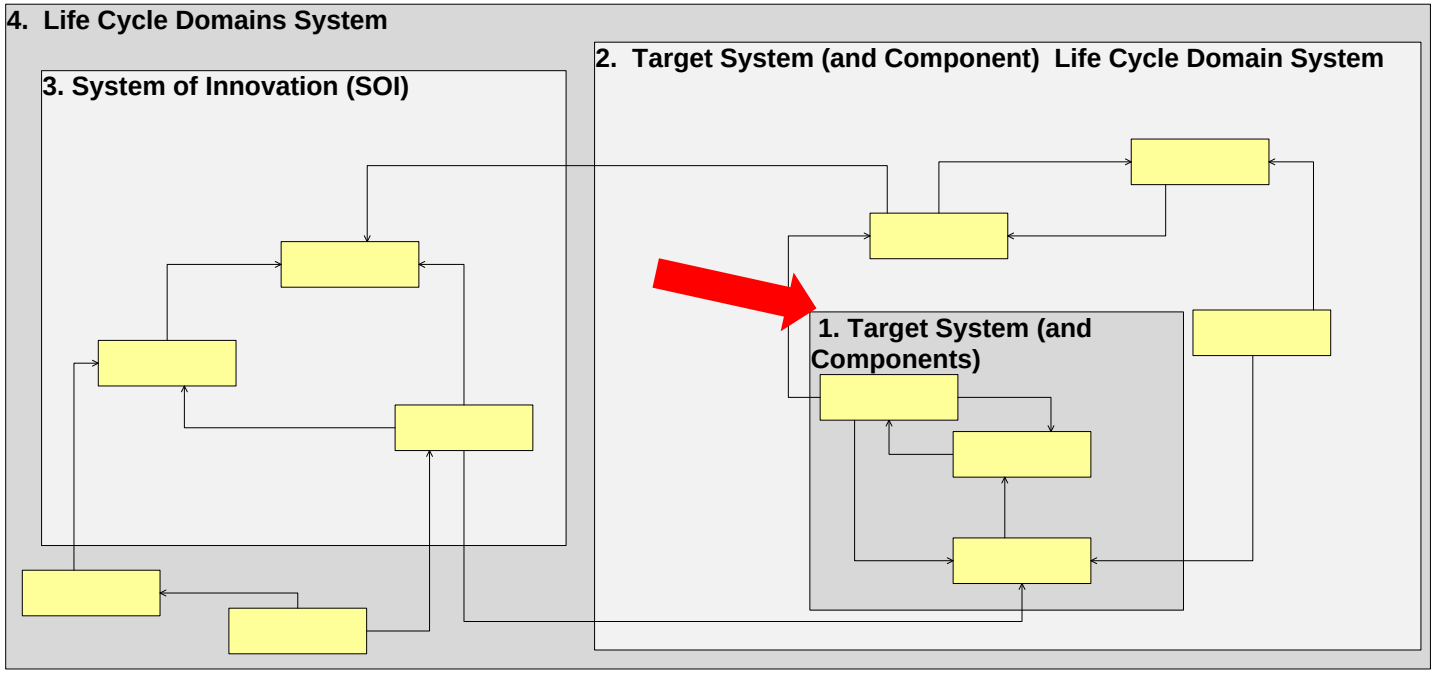


Internal component roles shown (yellow shapes) are notional, and will be identified and defined later.

The Agile System Domain Model

System 1: The **Target System (and Components):** (Definition) The logical system of interest, which results from, or is subject to, innovation.

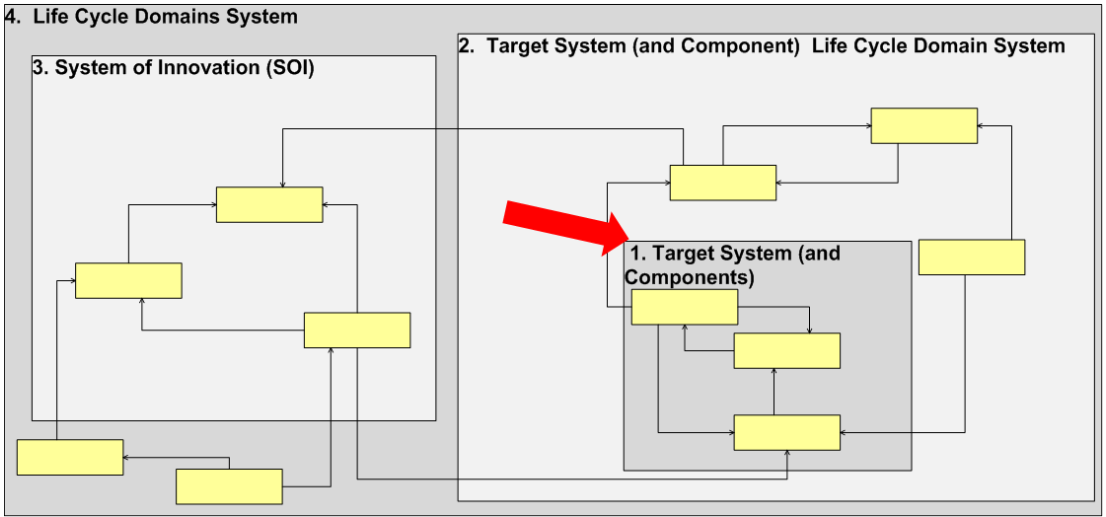
- The Components maintained for integration into a Target System, but not yet integrated, are included in this domain.
- Notice that this idea can apply at multiple additional levels (e.g., SOS, System, Component, etc.)



Internal component roles shown (yellow shapes) are notional, and will be identified and defined later.

Example Target System (for System 1): Home Entertainment System Example

- Wide Area Transport Media (RF, Cable, DSL, etc.)
- Receivers/Tuners (AM, FM, Satellite, TV, Modem, etc.)
- Recorded Media (Vinyl, Mag Tape, CD, etc.)
- Media Players (Record, CD, Tape, DVD)
- Amplifiers
- Speakers
- Display Media (CRT, Plasma, LCD, OED, etc.)
- Local Transport Media (Wiring, Power Line Carrier, Bluetooth, etc.)
- User Controls (Panel, Specialized Remote, Universal Remote, Smart Phone)

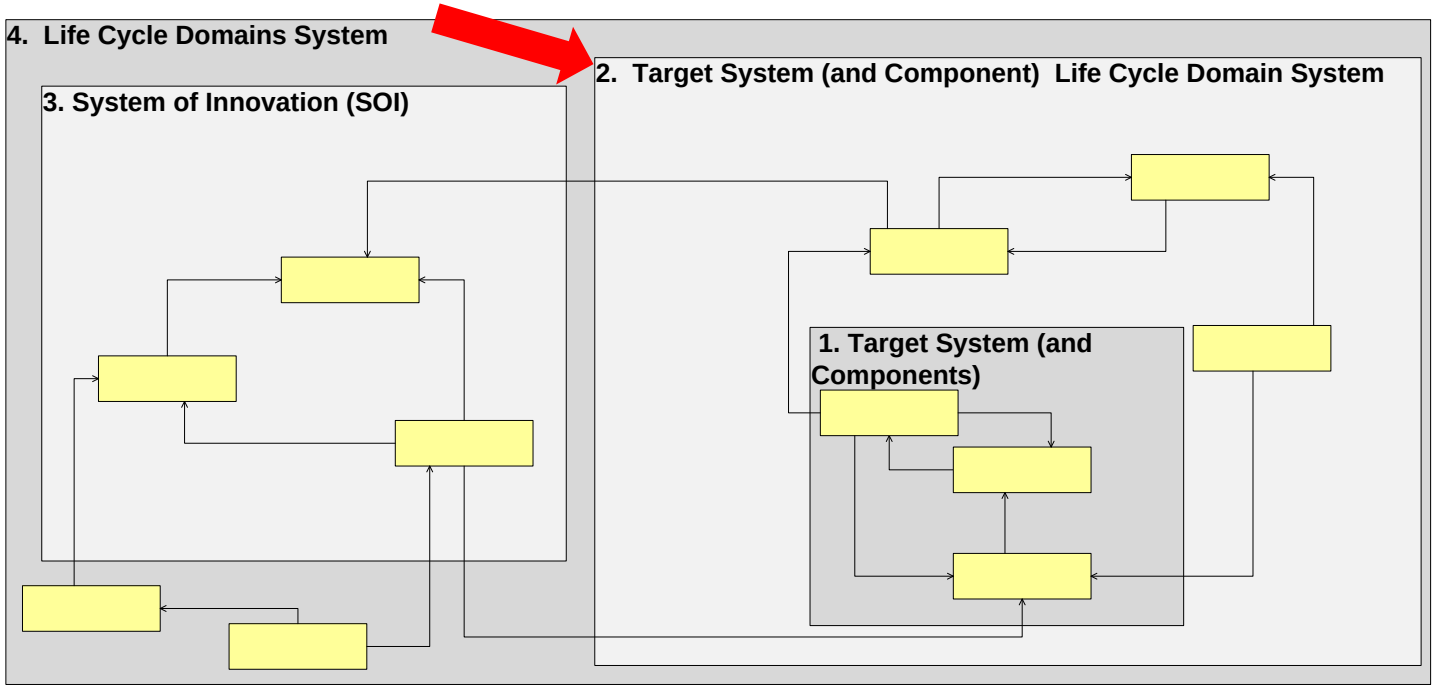


The Agile System Domain Model

System 2: The **Target System (and Component) Life Cycle Domain System:** (Definition) The logical system within which the Target System will exist during its life cycle, when "in service" or otherwise. This domain includes all actors[1] with which the Target System will directly interact during its life cycle:

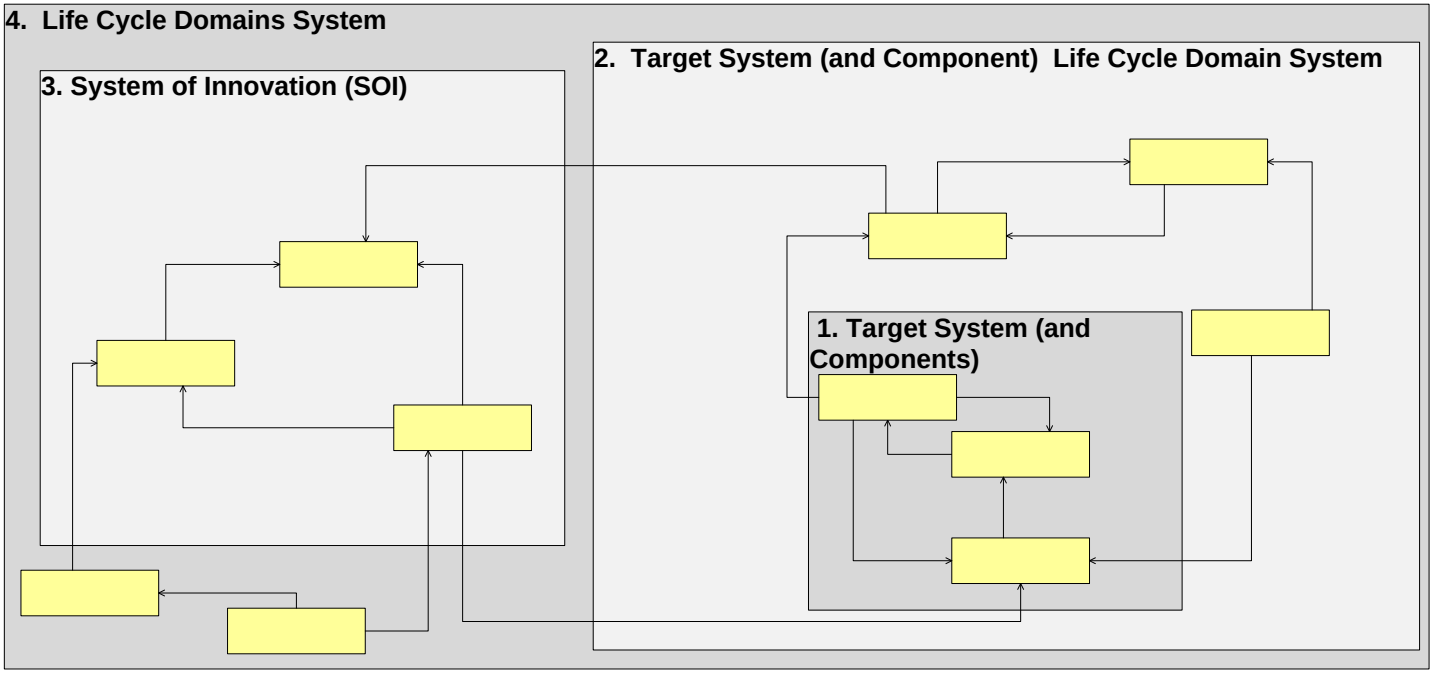
- This includes any system that directly manages the life cycle of an instance of a Target System (or a Component)—production and integration systems, maintenance and operations systems, and others.

1. "Actors" are environmental entities that interact with a system of interest.



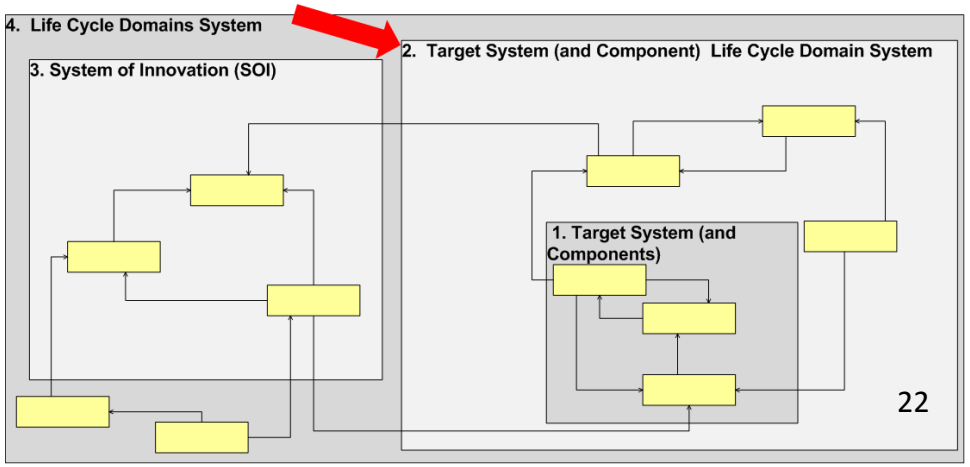
Internal component roles shown (yellow shapes) are notional, and will be identified and defined later.

- Again, remember that these are logical (behavioral) roles. In realized physical systems, a single physical system may behave as both a Target System and a system that produces, modifies, reconfigures, or otherwise manages a Target System, by having roles from each allocated to it.
- For purposes of this logical roles description, they have been identified separately.
- We will add the physical components to the model shortly.



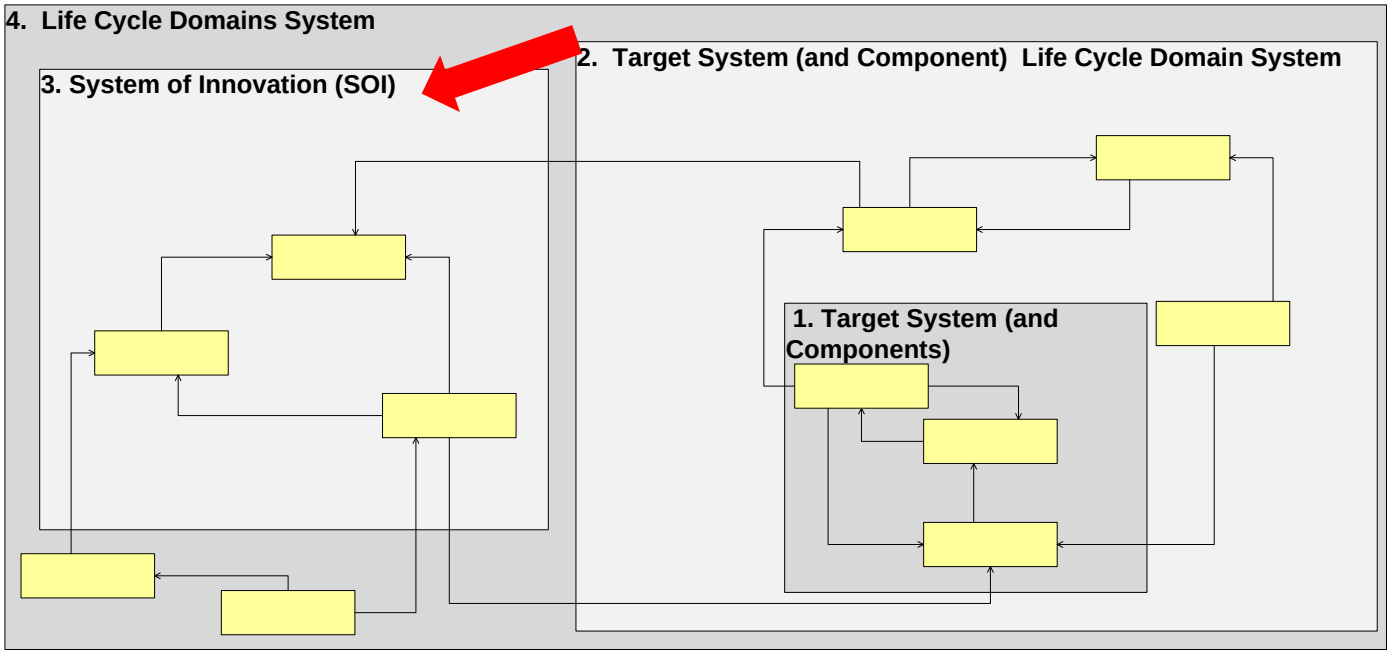
Example Target System (and Component) Life Cycle Domain System (System 2): Home Entertainment System Example

- Supply Chain: Electronics Systems Manufacturer, Distributor, Retailer, Electronic Components Manufacturer
- Operations, Maintenance, Configuration, Performance Management: Home User, Installation Technician, Hand & Electronic Tools, Repair Shop, Manufacturer Warranty Service Center, Manufacturer MES, PLM, CAD Information Systems & Tools
- Security Management: Physical Security, Authentication, Authorization, Encryption
- Other Environmental Actors: Power System, Home Environment, Broadcasters, Media Companies, Content Producers, Content Sellers



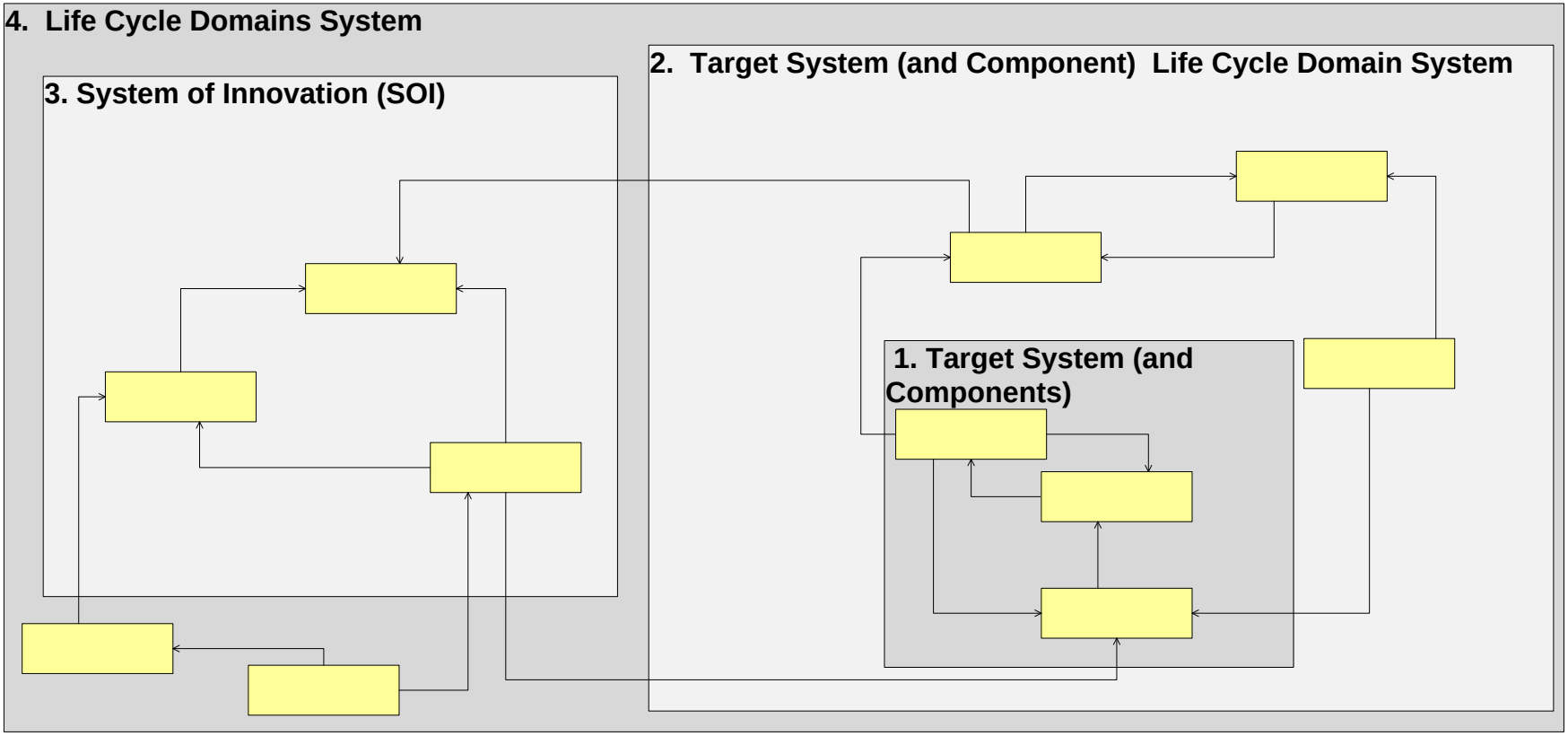
System 3: The **System of Innovation:** The logical system responsible for creating the possibility of (not production of) instances of Target System(s) with new or modified capabilities:

- Includes distillation of new knowledge (by observation) about Target Systems, their life cycle management, and their environmental domains, for future use.
- Also includes creation of instances of new production or other life cycle management capabilities for Target Systems, but not new instances of Target Systems.
- Engineers might think of this as the Engineering Process or the Development Process, but we have given it a more general name--to remind us that an innovation "competitor" may be operating from a cave or kitchen table, lacking a "recognized" engineering process; or, it might be a biological process that did not attend engineering school; or it might be some other type of innovation process, which we will study here.



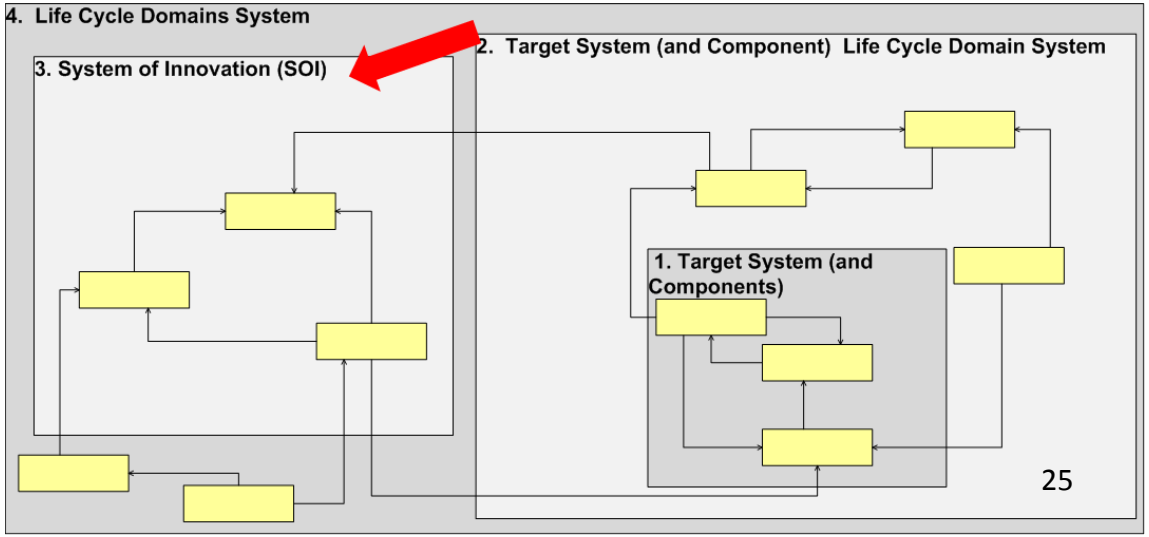
Internal component roles shown (yellow shapes) are notional, and will be identified and defined later.

- Summary so far:
  - System 2, the Target System Life Cycle Domain System produces and modifies instances of System 1, the Target Systems (and Components).
  - System 3, the System of Innovation, produces new abilities to do so, including knowledge.



Example System of Innovation (System 3): Home Entertainment System Example

- System Researchers, Designers: Electronic System Architects, Media and Device Basic Research Physical Scientists, Product & Process Designers, Network Architects, Computer Scientists, Standards Bodies.
- Configuration Management: CAD and PLM Tools and Information Systems.
- Security Management: Physical Security, Authentication, Authorization, Encryption



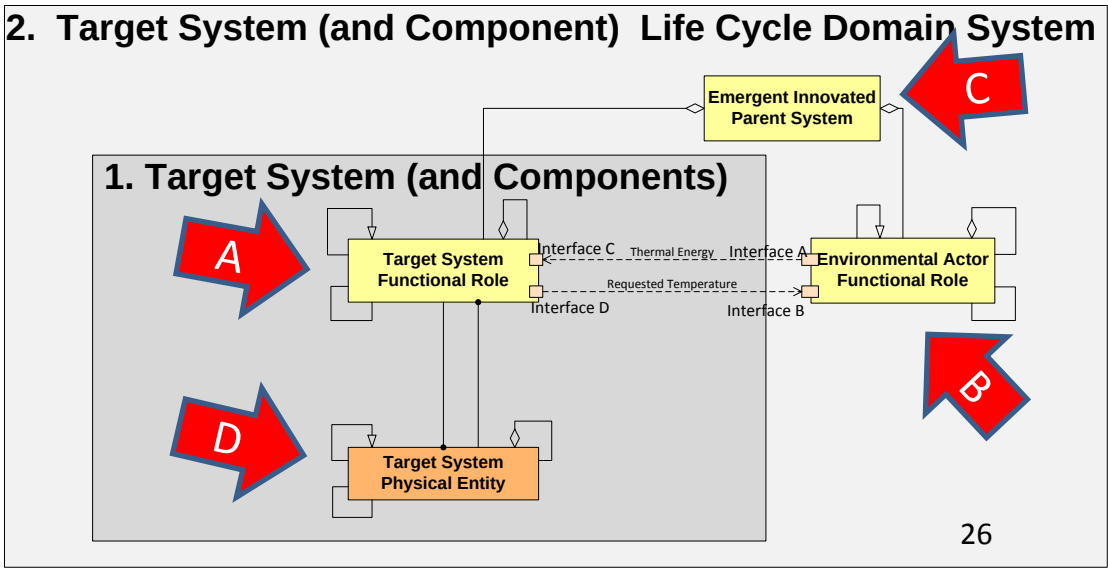
Logical Architecture and Physical Architecture of the Target System

A. The Innovated (Target) System is partitioned into a collection of **Target System Functional Roles.** These interact with each other to create the externally visible "black box" behavior of the Target System:
   - The web of connected Functional Roles within that system is its Logical Architecture.
   - These logical systems can also be in two types of hierarchy: A part-whole hierarchy and a special-general hierarchy.

B. The Innovated (Target) System interacts with external **Environmental Actor Functional Roles** played by environmental actors in the Target System (and Component) Life Cycle Domain System.

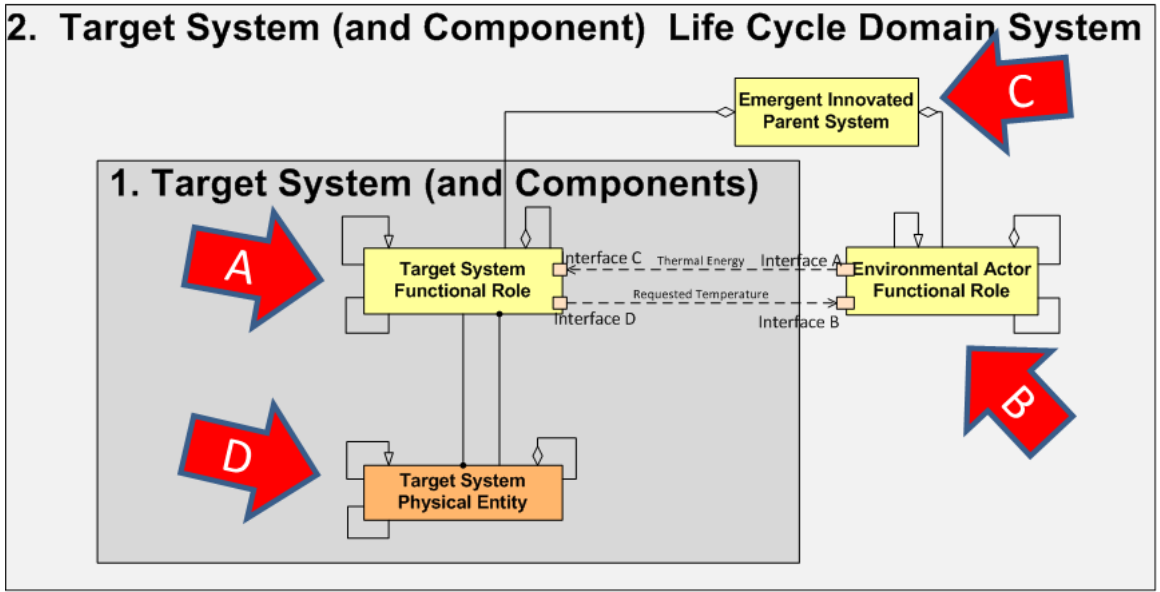
C. An **Emergent Innovated Parent System** is composed of the interacting Target System and its Environment.

D. The Target System Functional Roles are allocated to **Target System Physical Entities** that perform those roles:
   - There can also be hierarchies of these.



Example: Target System Roles, Environmental Actors, Physical Components, Emergent Innovated Parent System

A. Target System Functional Role: Downloadable Media Player
B. Environmental Actor Role: Music Library Supplier
C. Emergent Innovated Parent System: Post i-Tunes Music Industry
D. Physical Component: Apple iPhone6

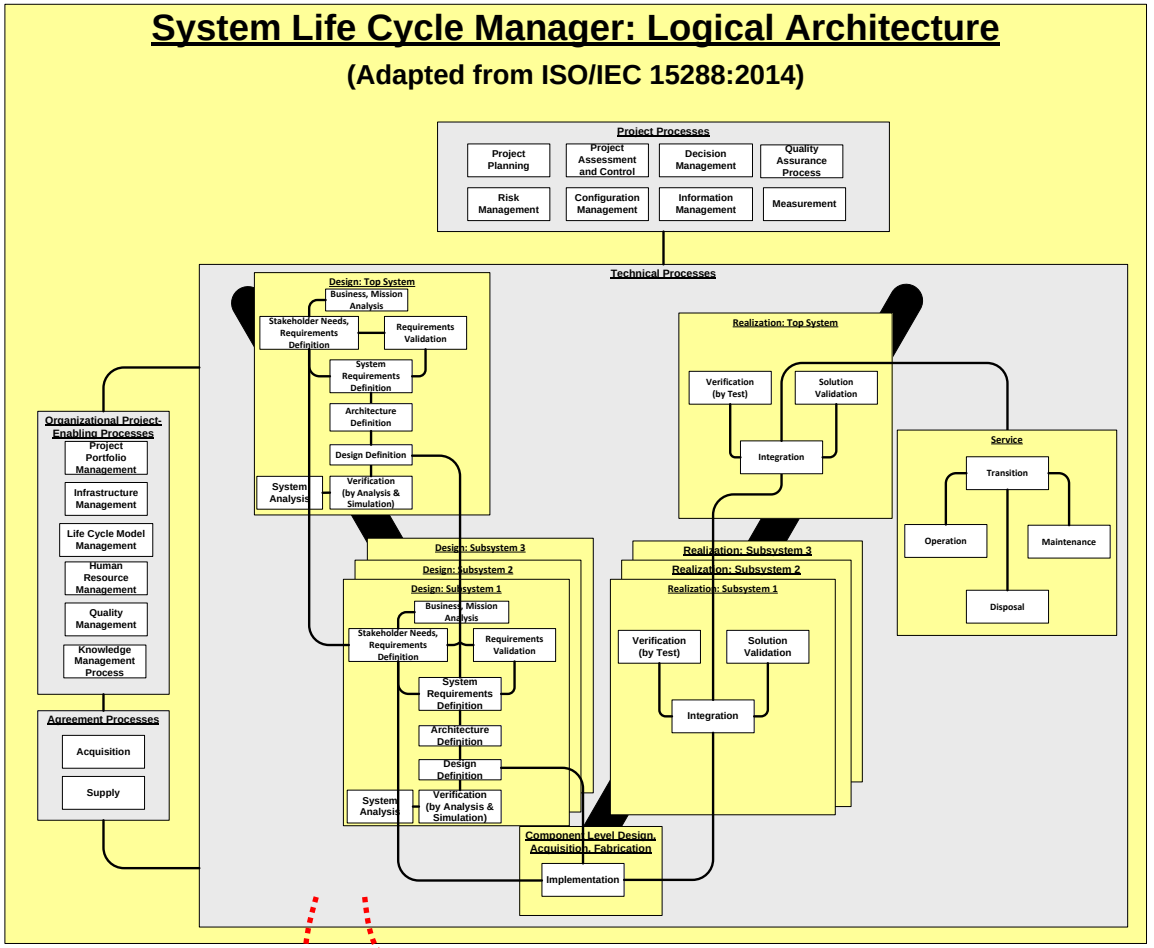


Relating Scrum and ISO 15288 Process Models

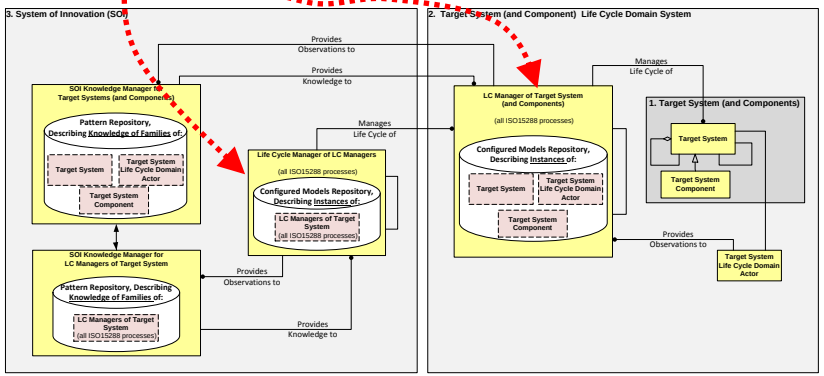
- More Than One Representation (Model View) of the Same Underlying (Process) Reality . . .

(See Attachment I for more.)

ISO15288
Reference Processes



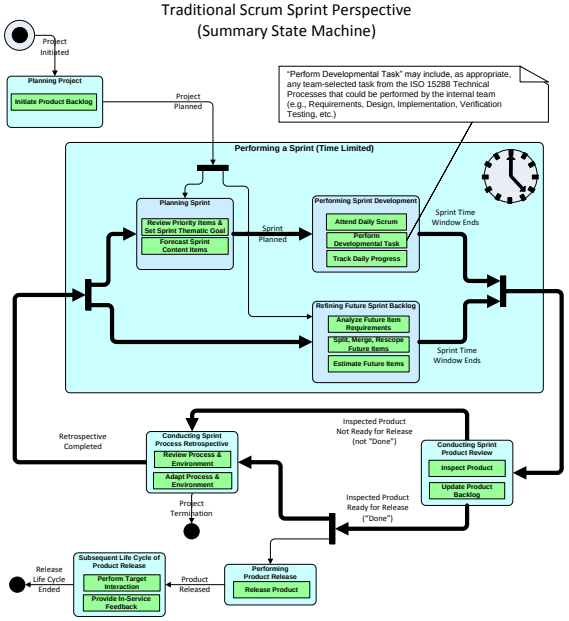
ISO15288 Technical Processes appear in System 2 (for target) and System 3 (for LC managers), as (potentially concurrent) “Vee” processes.



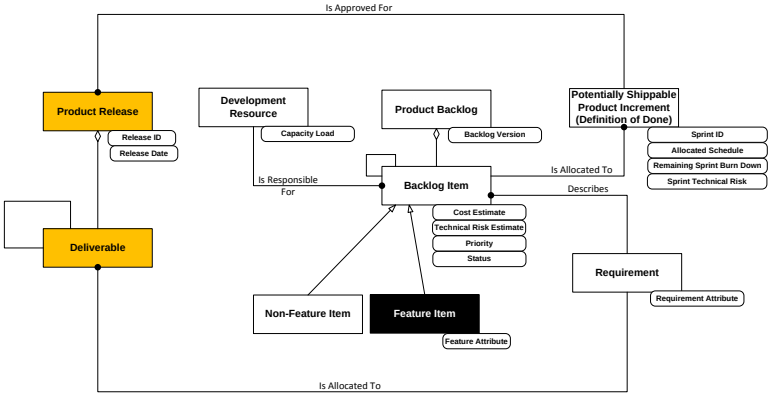
Agile Scrum Model

(See MBSE Workshop Attachment I for more.)

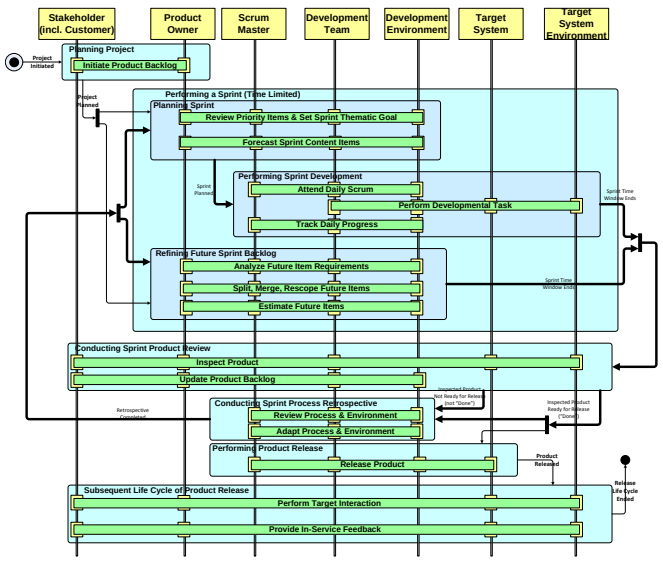
Traditional Scrum Sprint Perspective
(Summary State Machine)



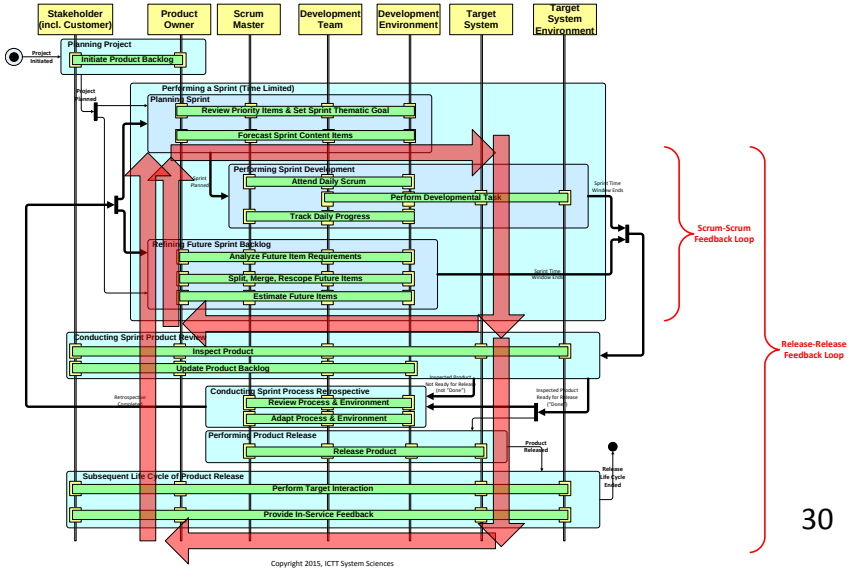
Traditional Scrum Sprint Perspective
(Simplified Model of Managed Information)



Traditional Scrum Sprint Perspective
(Activity Diagram, with Swim Lane Roles)



Traditional Scrum Sprint Perspective
(Activity Diagram, with Swim Lane Roles)

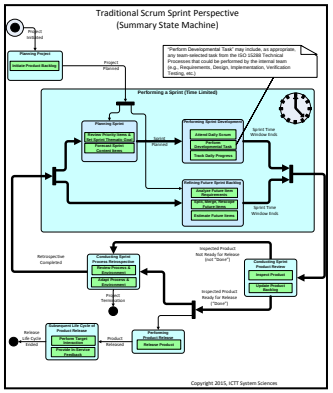


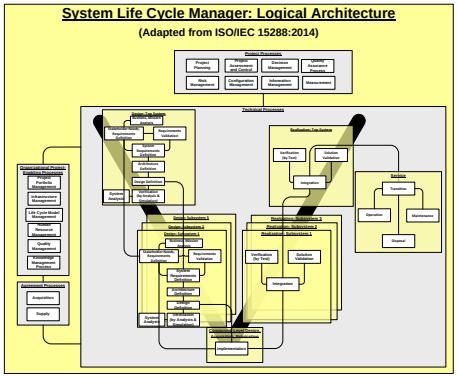
More Than One Representation (Model View) of the Same Underlying Reality

We are dealing with four different representations of the same underlying reality:

1. The Scrum Pattern: Emphasizes time and feedback, focusing on processes for learning and management of risk
2. The ISO15288 Pattern: Emphasizes types of processes, focusing on management of processes
3. The Agile Systems Engineering Life Cycle Pattern : Shows how (1) and (2) above may be seen as one
4. The S*Metamodel: Emphasizes the information flowing through all three of them: (1), (2), and (3)

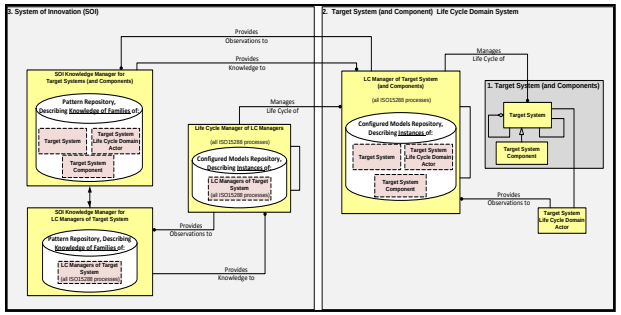
Scrum Pattern

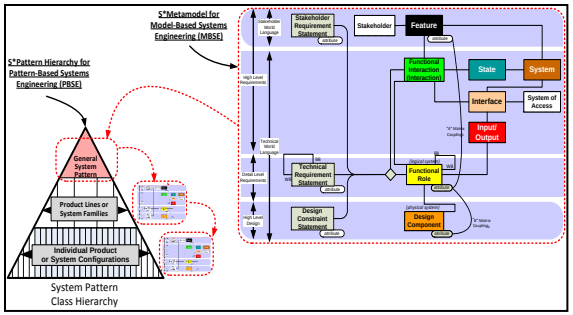




ISO15288 Pattern

ASELC Pattern



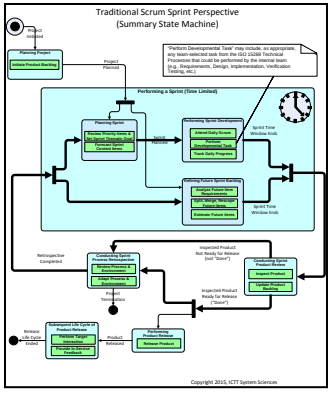


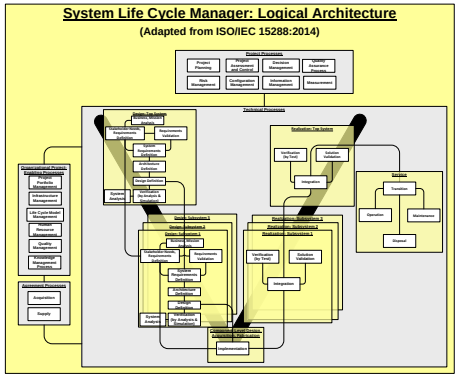
S*Metamodel

More Than One Representation (Model View) of the Same Underlying Reality

- The Scrum Model is actually an abstraction of the more complex-looking multiple Processes of the ISO15288 System Life Cycle reference model:
  - As indicated in the Agile literature, nothing about the Scrum Model is intended to prevent things like Requirements Analysis, Verification (Test), or even aspects of Project Management, . . .
  - But those activities are shared by the small team members who play many individual roles, and the simpler-looking Scrum model "gives us permission" to "do what is needed" in a given situation, in an "agile way".

Scrum Pattern





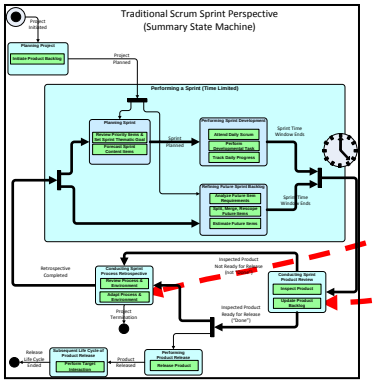
ISO15288 Pattern

More Than One Representation (Model View) of the Same Underlying Reality

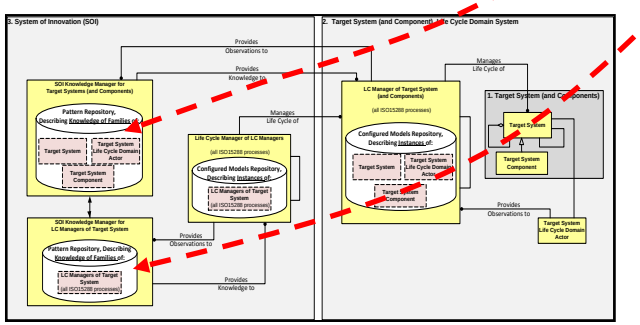
- The Scrum Model also abstracts complex learning behavior, into simple-looking form—but it is still strongly expected to occur as part of the Agile Process, and is more explicitly represented in the ASELC Pattern, as capture of Pattern information—not assumed to be only in human minds.

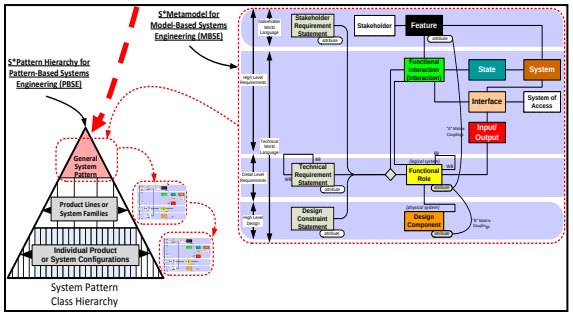
Learning

Scrum Pattern



ASELC Pattern





S*Metamodel

Learning often in upper-most S1,2,3 Pattern, but can also be in specializations and configurations below it.

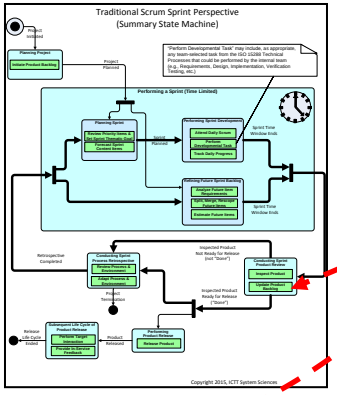
More Than One Representation (Model View) of the Same Underlying Reality

- Notice that the division of the System 3 roles in the ASELCM Pattern corresponds to the Scrum division of (review and learning about target system) versus (review and learning about development process):

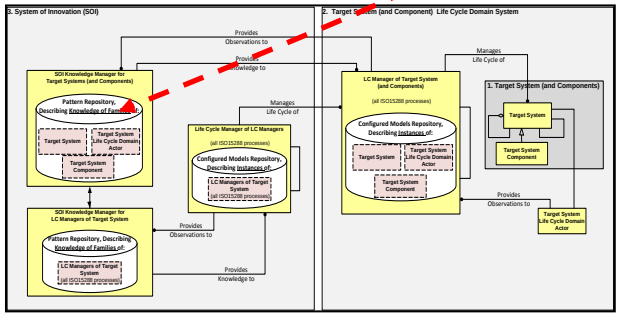
Pattern: Learnings about Target System (Product & Its Environment)

Pattern: Learnings about Development / Fielding System & Its Environment

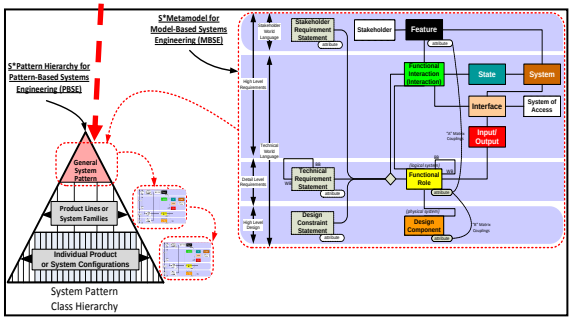
Scrum Pattern



ASELC Pattern



S*Metamodel

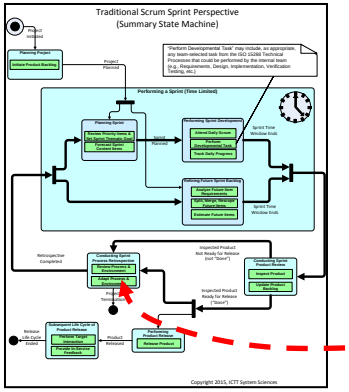


More Than One Representation (Model View) of the Same Underlying Reality

- Notice that the division of the System 3 roles in the ASELCM Pattern corresponds to the Scrum division of (review and learning about target system) versus (review and learning about development process):

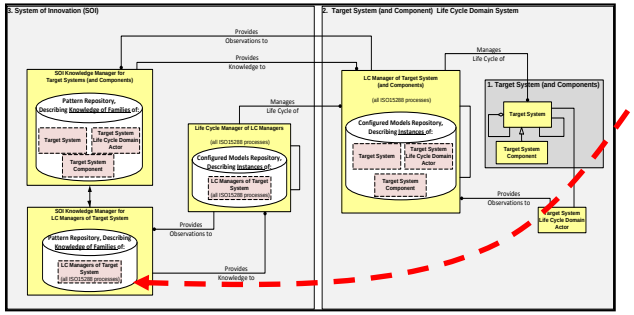
Pattern: Learnings about Target System (Product & Its Environment)

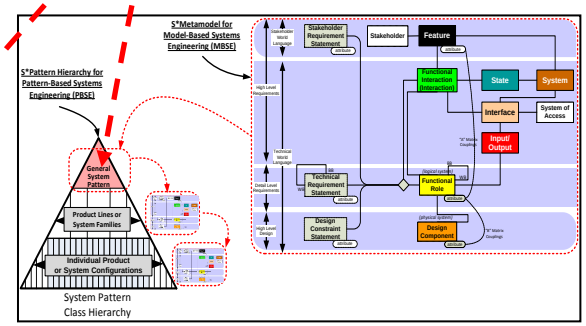
Scrum Pattern



Pattern: Learnings about Development / Fielding System & Its Environment

ASELC Pattern

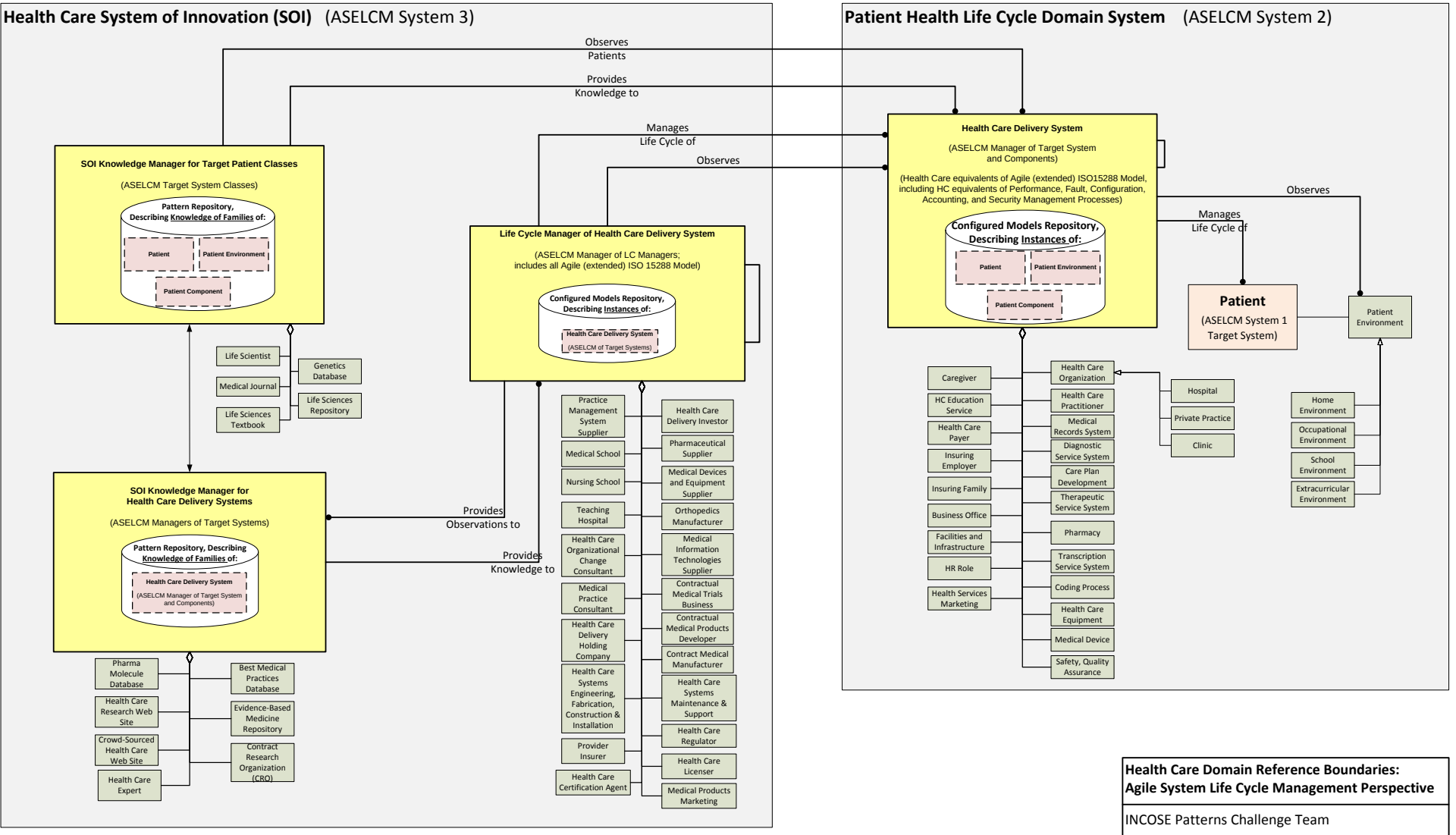




S*Metamodel

Health Care as an example:

ASELCM perspective on health care domain, before starting a model



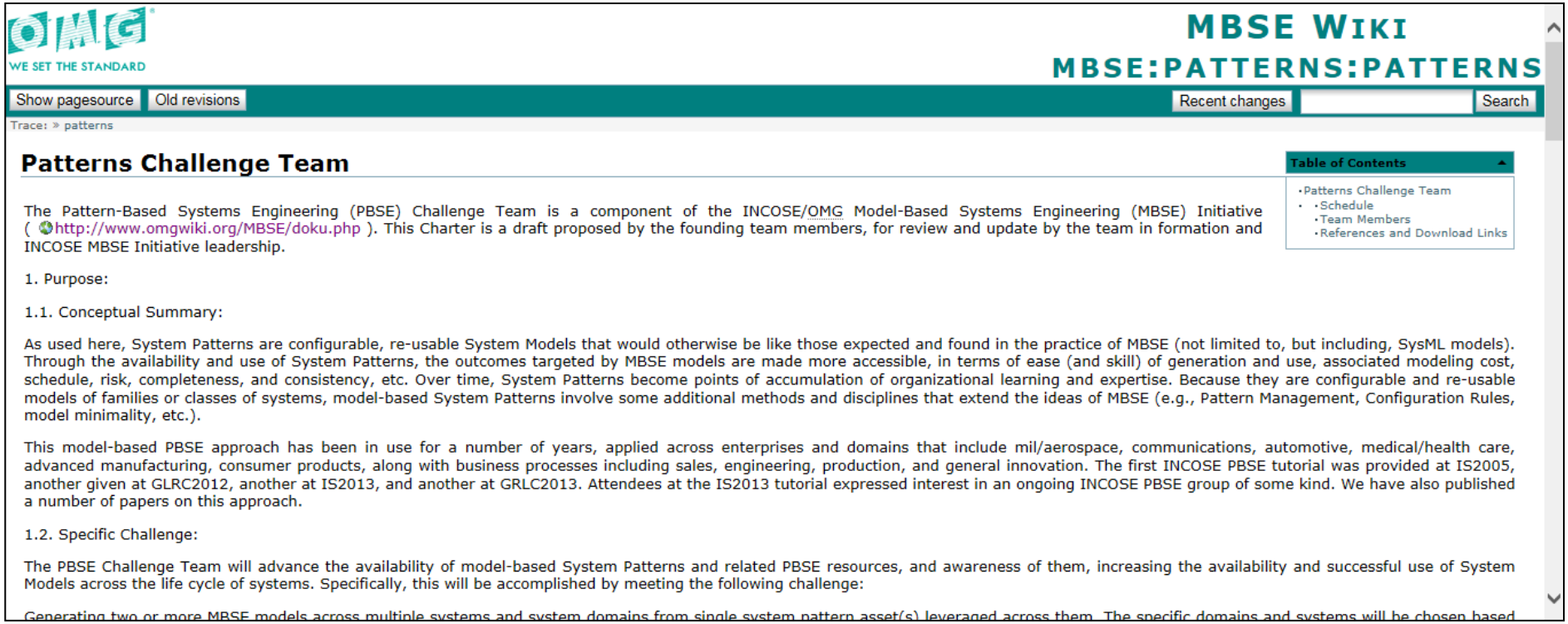
Health Care Domain Reference Boundaries: Agile System Life Cycle Management Perspective

INCOSE Patterns Challenge Team

V1.2.7 04.30.2015

What is the INCOSE Patterns Challenge Team?

http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns



OMG
WE SET THE STANDARD

MBSE WIKI
MBSE:PATTERNS:PATTERNS

Show pagesource | Old revisions | Recent changes | Search

Trace: » patterns

Patterns Challenge Team

Table of Contents
- Patterns Challenge Team
  - Schedule
  - Team Members
  - References and Download Links

The Pattern-Based Systems Engineering (PBSE) Challenge Team is a component of the INCOSE/OMG Model-Based Systems Engineering (MBSE) Initiative (http://www.omgwiki.org/MBSE/doku.php). This Charter is a draft proposed by the founding team members, for review and update by the team in formation and INCOSE MBSE Initiative leadership.

1. Purpose:

1.1. Conceptual Summary:

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

This model-based PBSE approach has been in use for a number of years, applied across enterprises and domains that include mil/aerospace, communications, automotive, medical/health care, advanced manufacturing, consumer products, along with business processes including sales, engineering, production, and general innovation. The first INCOSE PBSE tutorial was provided at IS2005, another given at GLRC2012, another at IS2013, and another at GRLC2013. Attendees at the IS2013 tutorial expressed interest in an ongoing INCOSE PBSE group of some kind. We have also published a number of papers on this approach.

1.2. Specific Challenge:

The PBSE Challenge Team will advance the availability of model-based System Patterns and related PBSE resources, and awareness of them, increasing the availability and successful use of System Models across the life cycle of systems. Specifically, this will be accomplished by meeting the following challenge:

Generating two or more MBSE models across multiple systems and system domains from single system pattern asset(s) leveraged across them. The specific domains and systems will be chosen based

INCOSE PBSE Challenge Team's PBSE Methodology Summary for INCOSE includes overview and many references:

"Pattern-Based Systems Engineering (PBSE), Based On S*MBSE Models", INCOSE Patterns Challenge Team, 2015.

Introduction to INCOSE MBSE Patterns Challenge Team

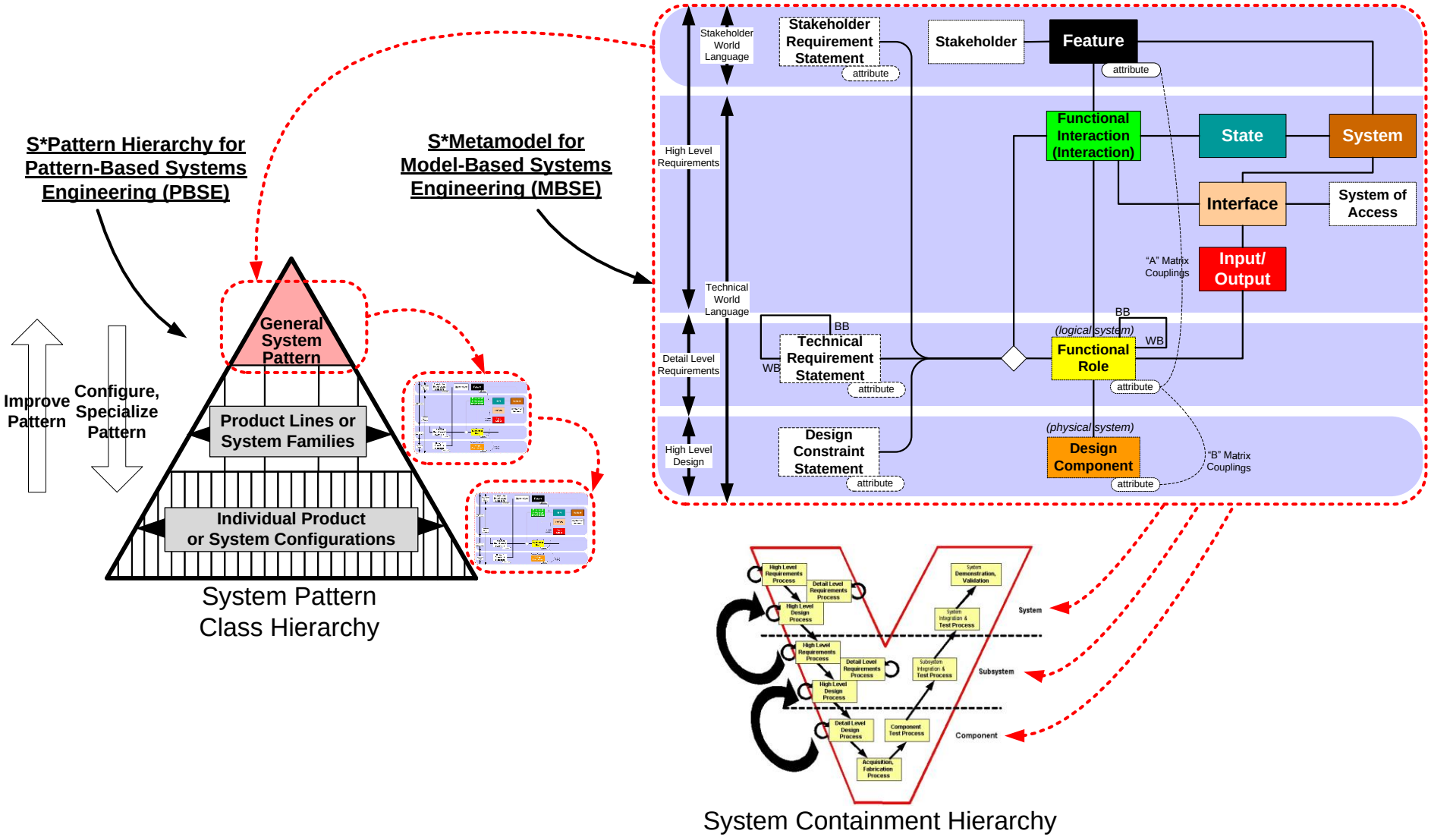
- Started in 2014, meeting approximately monthly, membership across domains.
- Team Co-chairs: Bill Schindel, Troy Peterson
- Six accepted IS2015 Challenge Team member papers.
- Re-usable, configurable, MBSE models ("Patterns").
- Based on S*Metamodel.
- Language and tool independent—frequently in SysML.
- Methodology practiced across domains ~ 20 years.
- For more information . . .

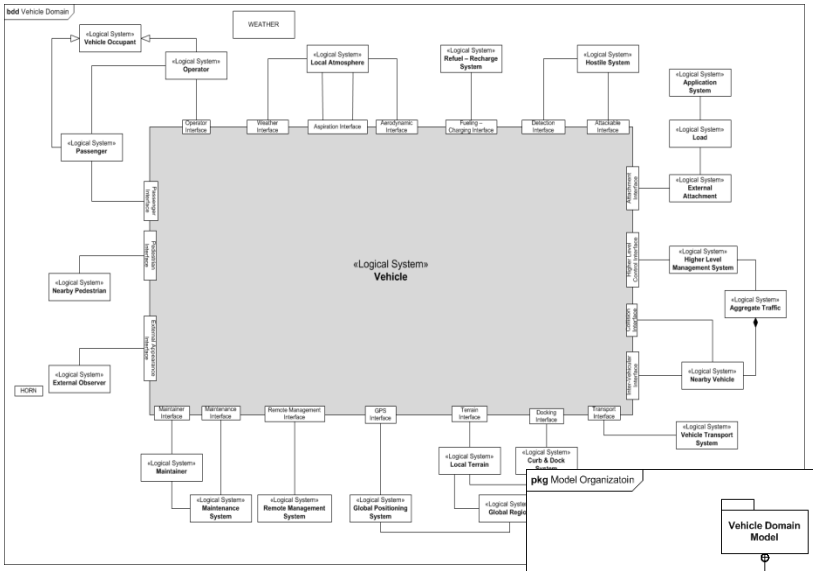
http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns

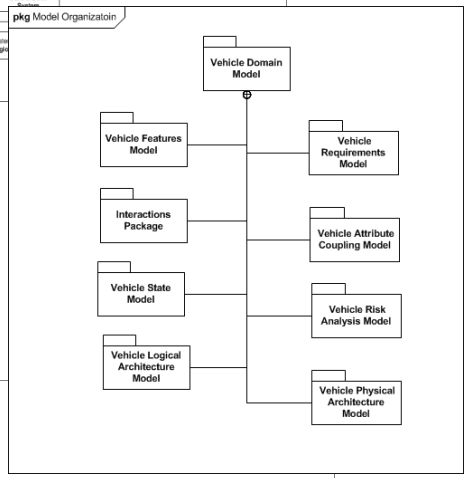
Cooperative cross-team/working group projects

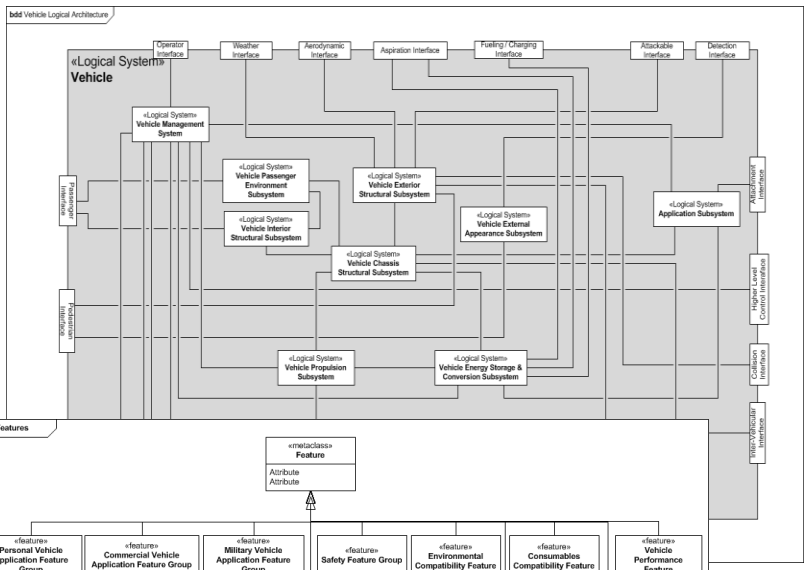
- The Patterns Challenge Team has been reaching out to other INCOSE and industry working groups:
  - Joint projects of interest.
- Example: We are jointly supporting, with the INCOSE Agile Systems Working Group (Rick Dove, chair), the Agile Systems Engineering Life Cycle Model Project
  - Sponsored by INCOSE
  - During 2015-16
  - Announced at IW2015

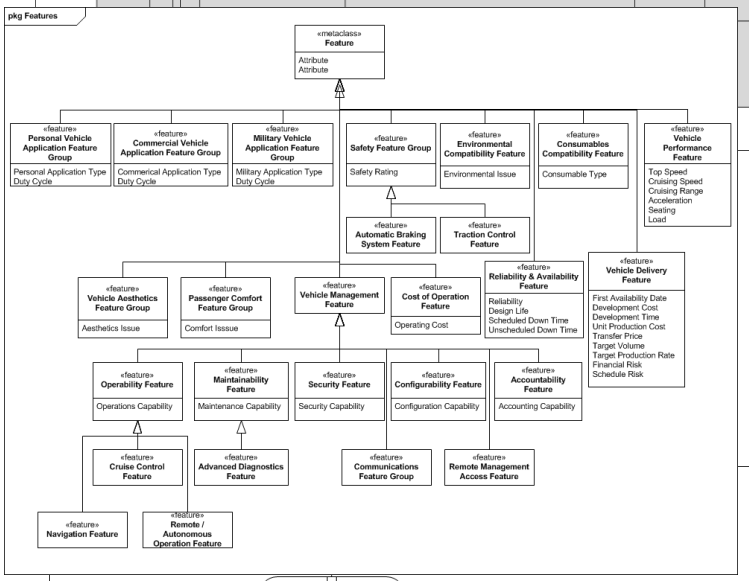
S*Models, S*Patterns

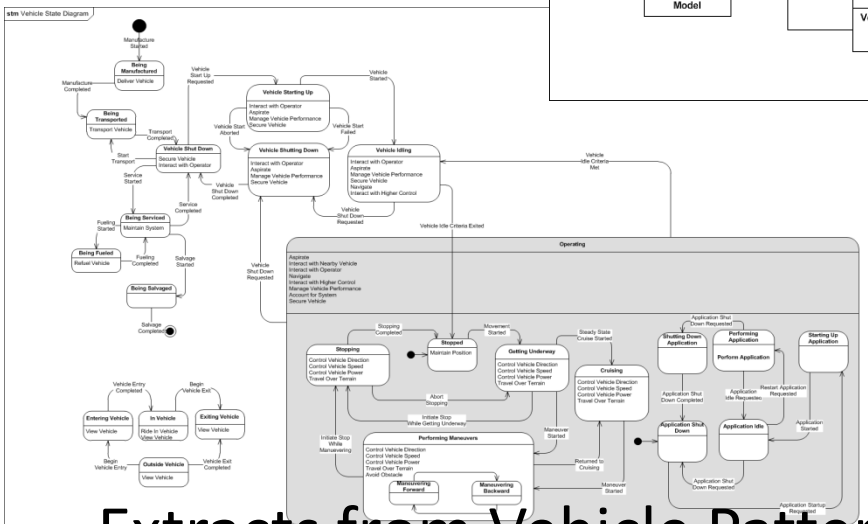


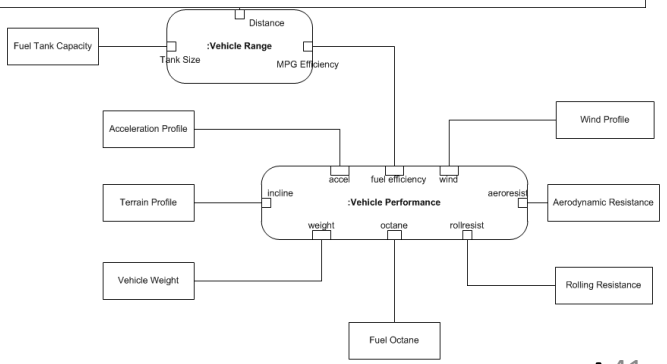






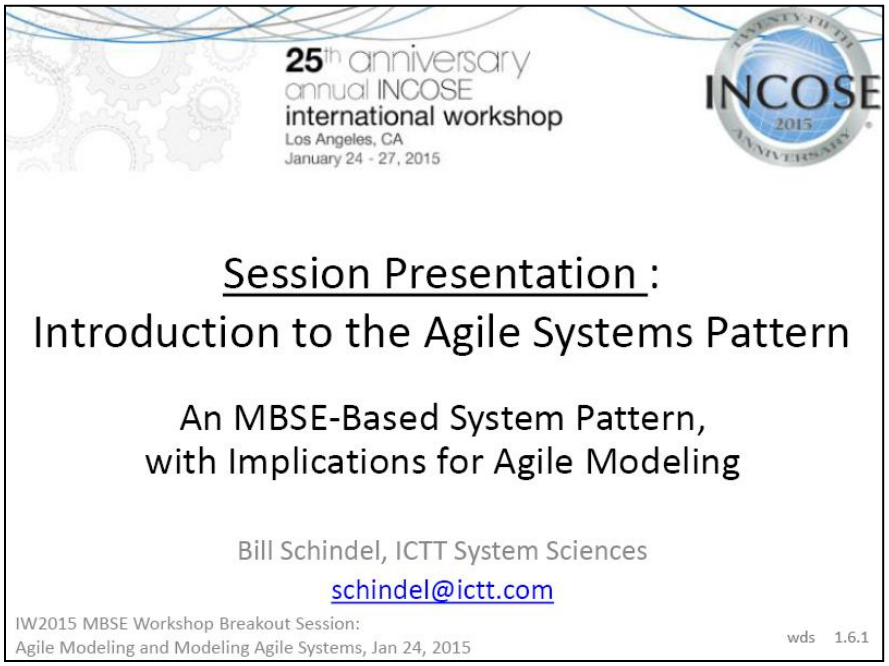






Extracts from Vehicle Pattern in 2013 INCOSE PBSE Tutorial

Where can I learn more?



See --

http://www.omgwiki.org/MBSE/doku.php?id=mbse:incose_mbse_iw_2015:breakout_out_session_agile_modeling

Primary References

1. Rick Dove, Ralph LaBarge, “Fundamentals of Agile Systems Engineering—Part 1” and “Part 2”, INCOSE IS2014, July, 2014.
2. Rick Dove, “Agile Systems 101”, “102” and “103”, Paradigm Shift, International, http://www.parshift.com/s/AgileSystems-101.pdf
3. ----------, *Response Ability: The Language, Structure, and Culture of the Agile Enterprise,* Wiley, 2001.
4. W. Schindel, “What Is the Smallest Model of a System?”, *Proc. of the INCOSE 2011 International Symposium*, International Council on Systems Engineering (2011).
5. Bill Schindel, Troy Peterson, “Introduction to Pattern-Based Systems Engineering (PBSE): Leveraging MBSE Techniques”, in Proc. of INCOSE 2013 International Symposium, Tutorial, June, 2013.
6. W. Schindel, “Maps or Itineraries?: A Systems Engineering Insight from Ancient Navigators”, to appear in Proc. of INCOSE International Symposium 2015, July, 2015.
7. ----------, “System Life Cycle Trajectories: Tracking Innovation Paths Using System DNA”, to appear in Proc. of INCOSE International Symposium 2015, July, 2015.
8. Schindel and Beihoff: “Systems of Innovation I: Models of Their Health and Pathologies”, Proc. of INCOSE International Symposium, 2012.
9. W. Schindel, “Systems of Innovation II: The Emergence of Purpose”, Proceedings of INCOSE 2013 International Symposium (2013)

Secondary References

1. Rick Dove, “Agile Systems and Processes—Driving Architecture with ConOps and Response Situation Analysis (Agile 102)”, Paradigm Shift, International, September 16, 2013.
2. Rick Dove, “Security R Us: Systems Engineering is the High Ground”, INCOSE Biomedical Healthcare Working Group, April 24, 2014.
3. W. Schindel, “Requirements statements are transfer functions: An insight from model-based systems engineering”, *Proceedings of INCOSE 2005 International Symposium*, (2005).
4. “OMG Systems Modeling Language, Version 1.3”, Object Management Group, June, 2012.
5. 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).
6. W. Schindel, “Pattern-Based Systems Engineering: An Extension of Model-Based SE”, INCOSE IS2005 Tutorial TIES 4, 2005.
7. 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).
8. W. Schindel, “The Impact of ‘Dark Patterns’ On Uncertainty: Enhancing Adaptability In The Systems World”, in Proc. of INCOSE Great Lakes 2011 Regional Conference on Systems Engineering, Dearborn, MI, 2011
9. INCOSE/OMG Patterns Working Group 2013-14 http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns
10. 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, American Association for Engineering Education, (2011).
11. ISO/IEC 15288: Systems Engineering—System Life Cycle Processes. International Standards Organization (2014).
12. *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, Version 4, International Council on Systems Engineering (2014).
13. “A World in Motion: Systems Engineering Vision 2025”, INCOSE, 2014.
14. “Instrument Unit Fact Sheet: Saturn V News Reference”, IBM Federal Systems Division, 1968.
15. W. David Woods, Kenneth D. MacTaggart and Frank O'Brien, “Apollo 11 Flight Journal: Day 2, Part 1: Mid Course Correction”, http://history.nasa.gov/ap11fj/05day2-mcc.htm , updated 2009.