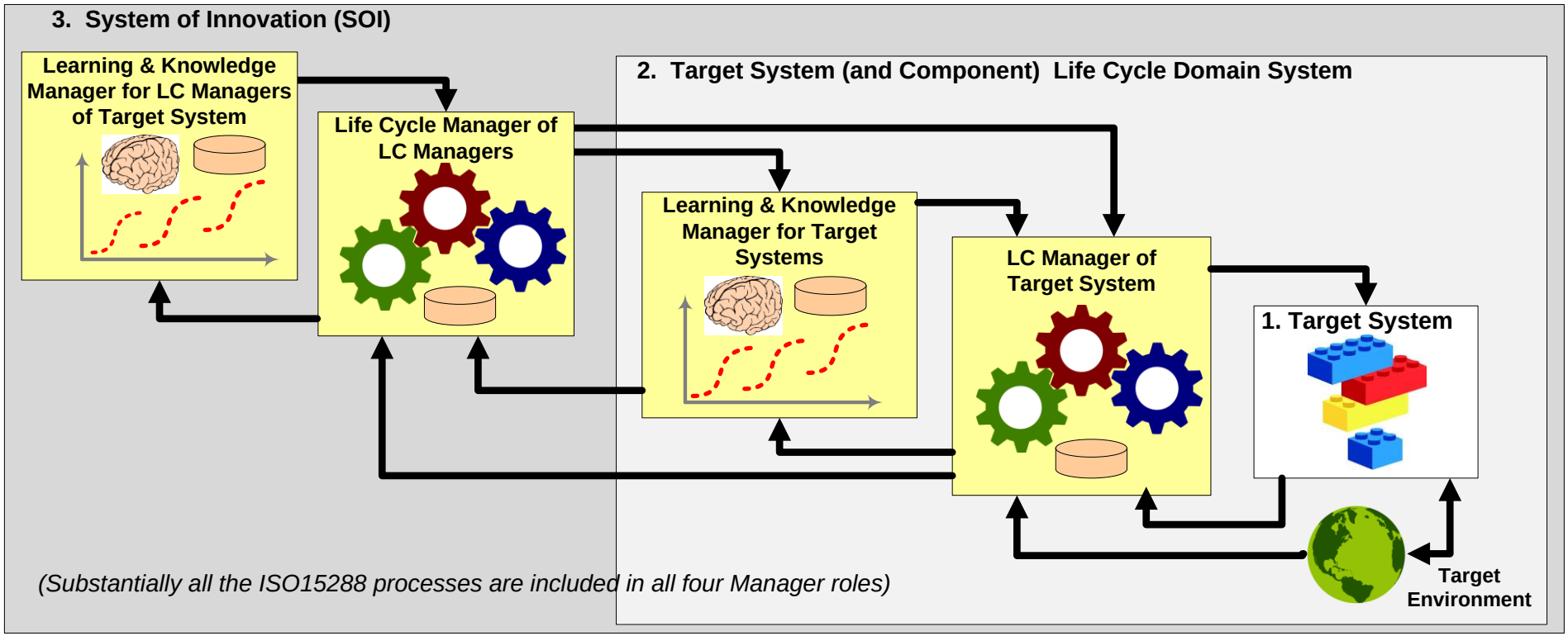
The INCOSE ASELCM Pattern:
A Reference Model for Agility in Systems



INCOSE 2016 international workshop
Los Angeles, CA, USA
January 30 - February 2, 2016

REVIEW OF PATTERN CONTENT:
January 31, 2016, Patterns WG Meeting

V1.3.1

schindel@ictt.com 812-232-2062,

Collaborating INCOSE Working Groups:

- Agile Systems Working Group

- Patterns Working Group

Contents

- Preliminaries:
  - Background
  - Current status
  - Next steps
- Current draft of the ASELCM Pattern:
  - Extracts from the general ASELCM Pattern
  - Observations from specific host site archetype configurations of ASELCM Pattern
- References

Preliminaries

- What is the INCOSE ASELCM Discovery Project?
- What are Agile Systems, and why do they matter?
- What is the INCOSE ASELCM Pattern?
- Different users, and different views, of this model.
- S*Models, S*Patterns: Using the S*Metamodel
- Pattern hierarchy and model configurations
- Before, during, and after the initial ASELCM workshops
- What is the status of the ASELCM Pattern?
- What comes next?
- Where can I learn more?

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

- During 2015-16, the INCOSE parent society is sponsoring the Agile Systems Engineering Life Cycle Model (ASELCM) Discovery Project, based on a series of workshop clinics being held at host example discovery sites across the U.S. and Europe.
- This project, now underway, will provide INCOSE inputs to a future version of ISO 15288, to improve explicit understanding of principles and practices of agility as applicable to systems engineering across different domains.

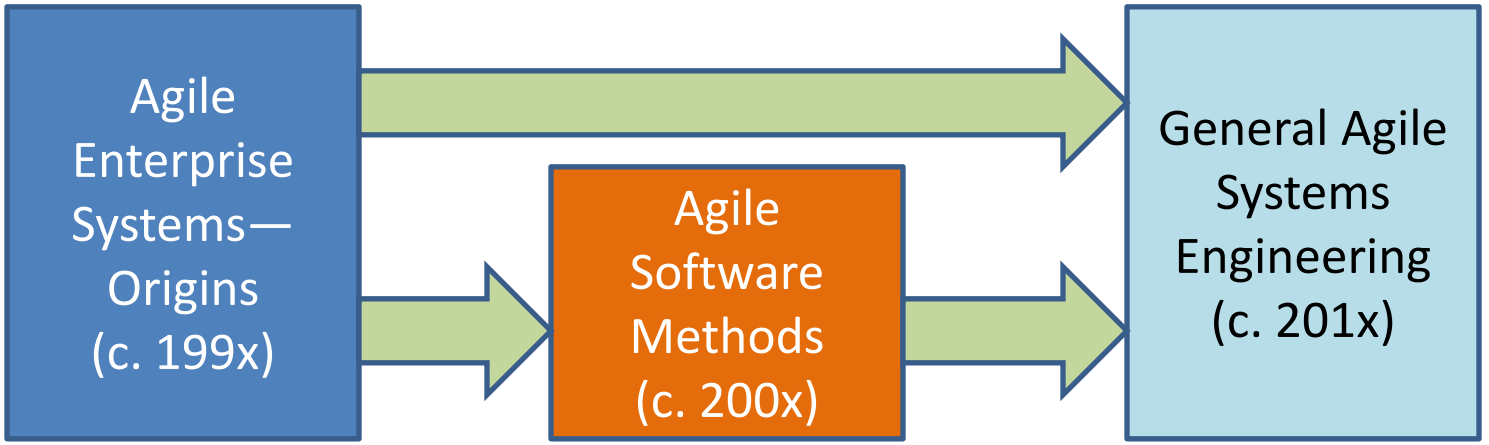
http://www.parshift.com/ASELCM/Home.html

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

- Announced at IW2015 January
- Built around a series of discovery workshops being conducted by example host sites during 2015-16
- Discovery clinics in 2015:
  - Navy SpaWar/MITRE, San Diego, CA, August
  - Northrop Grumman, Vienna, VA, August
  - Rockwell Collins, Cedar Rapids, IA, September
  - Lockheed Martin, Ft. Worth, TX, October
- You and your company can host or participate in 2016!
- Support from Agile Systems WG and Patterns WG:
  - R. Dove, project lead, co-leads K.Forsberg, H. Lawson, J. Ring, G. Roedler, B. Schindel

What are Agile Systems? Why do they matter?

Longer history than just Agile Software Development Methods :

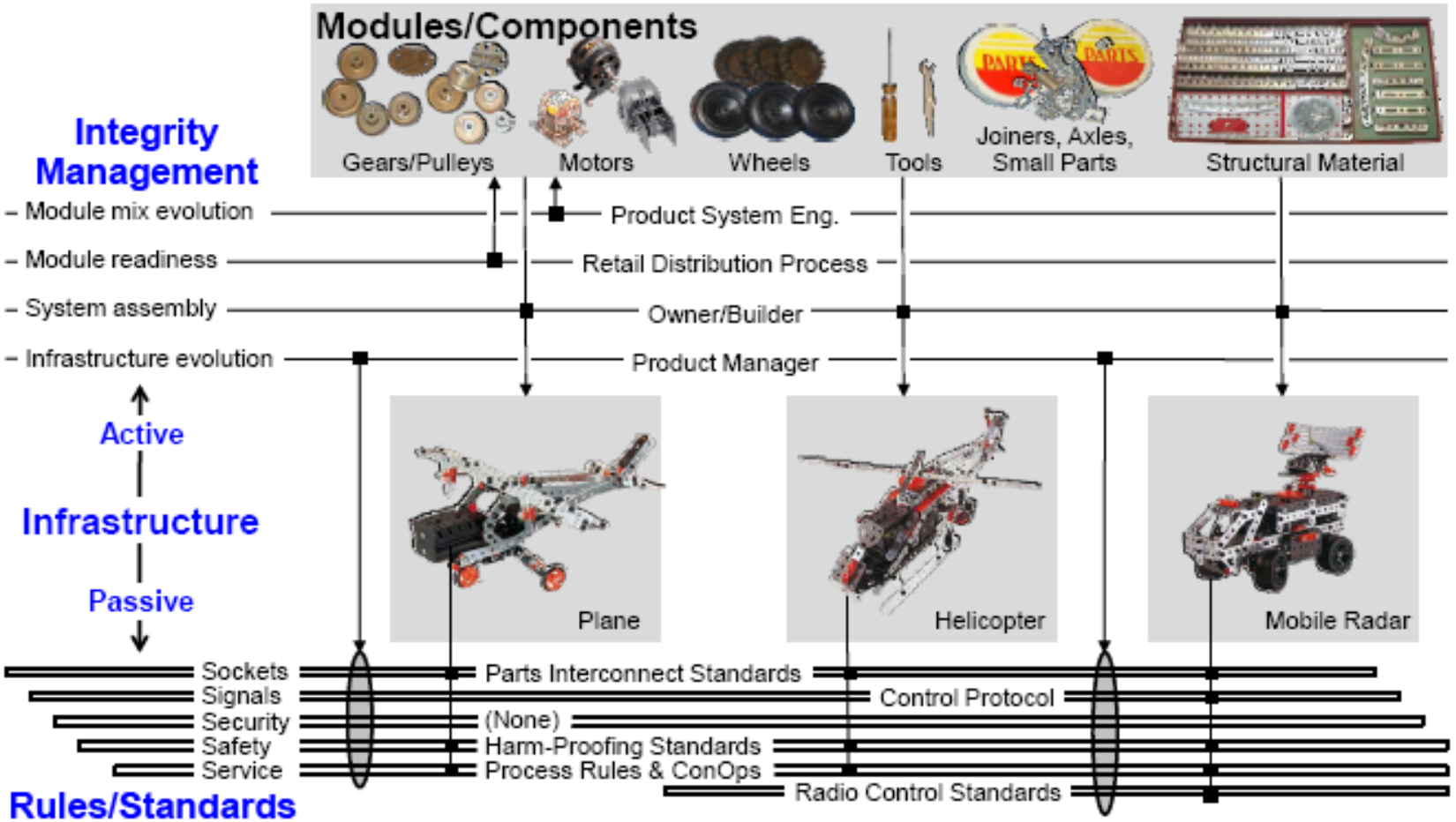


- For history and background, see Dove and LaBarge, 2014
- Agile software methods, by far better known, are related.
- General Agile Systems Engineering is the related broader subject of the INCOSE ASELCM Project.
- Problem space: Challenges of uncertainty and rates of change in environment, stakeholders, competition, technologies, capacities, capabilities. Not just “going faster”.

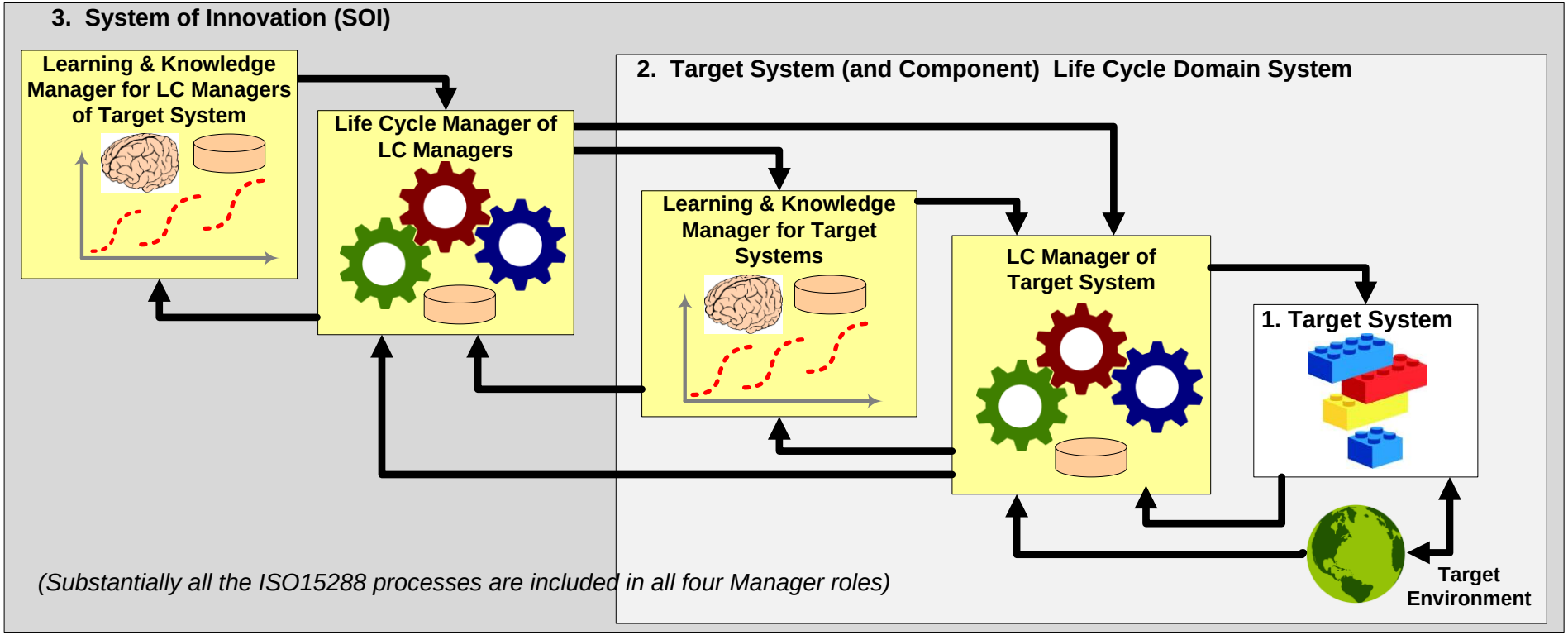
Agile Systems, Pre-MBSE Pattern (R. Dove)

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)



What is the Agile Systems Engineering Life Cycle Pattern?



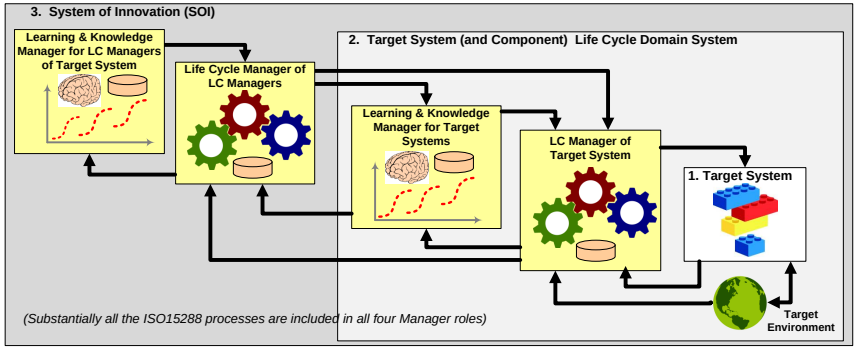
- **A general description of agile life cycle management systems (S2) supporting an agile target system (S1), plus the systems for adapting and managing those life cycle processes (S3).**
- **In the context of an arbitrary target system (S1) in a challenging environment (S2).**

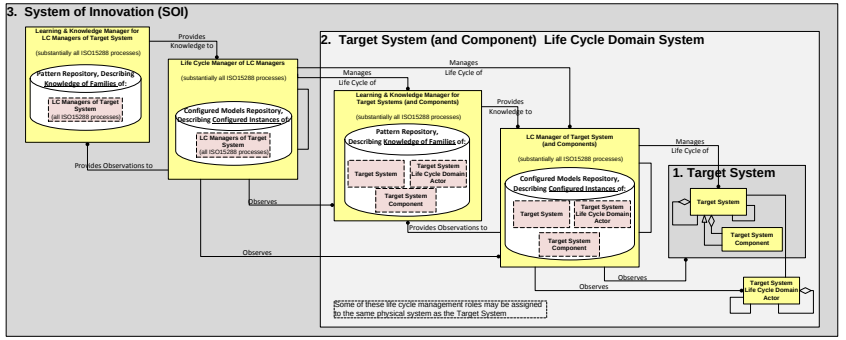
What is the Agile Systems Engineering Life Cycle Pattern?

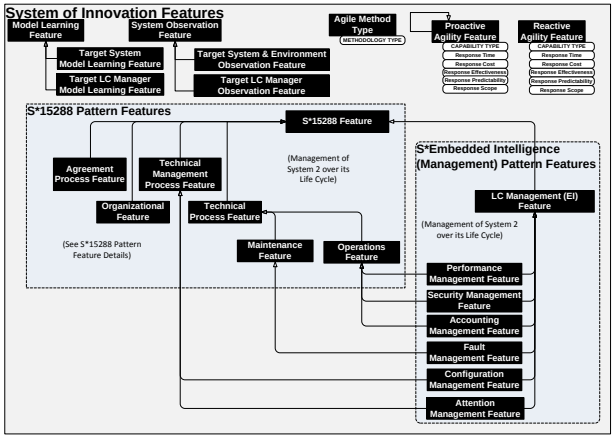
- Expressed as a re-usable, configurable, MBSE Pattern, capable of being specialized into different Agile System Engineering Life Cycle Model archetypes and configurations for different situations, methods, domains, and forms of agility.
- Directly tied to the ISO 15288 life cycle management processes standard, also expressed in the model.
- A point of accumulation for what we are learning about Agile Systems Engineering Life Cycles.
- Part of the related project inputs to INCOSE and future generations of ISO 15288 or other standards.

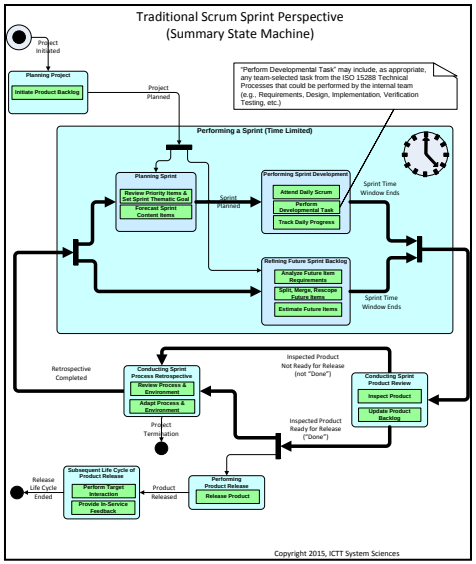
- The underlying ASELCM Pattern is expressed as an S*Pattern, independent of specific modeling languages and tools.
- Expressible in popular modeling languages and COTS tools,
- Scope includes multiple levels of model view detail, for different purposes and audiences.

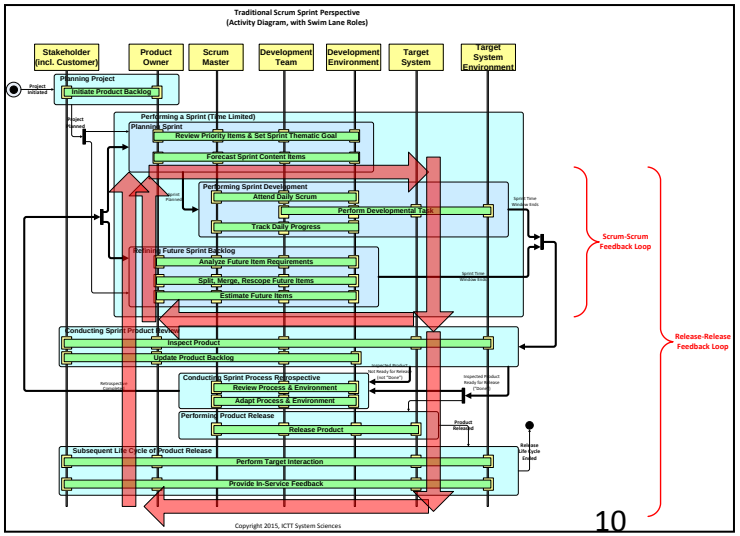
Detail views and descriptions are in references.





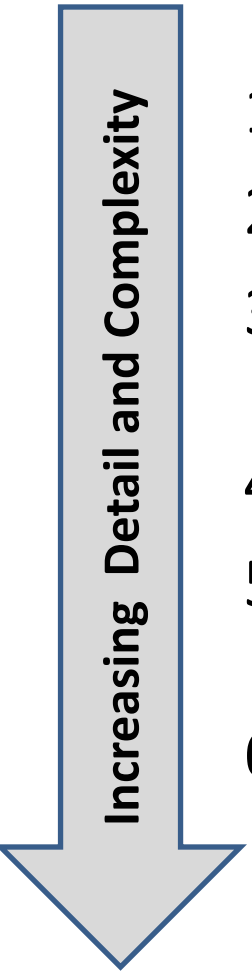






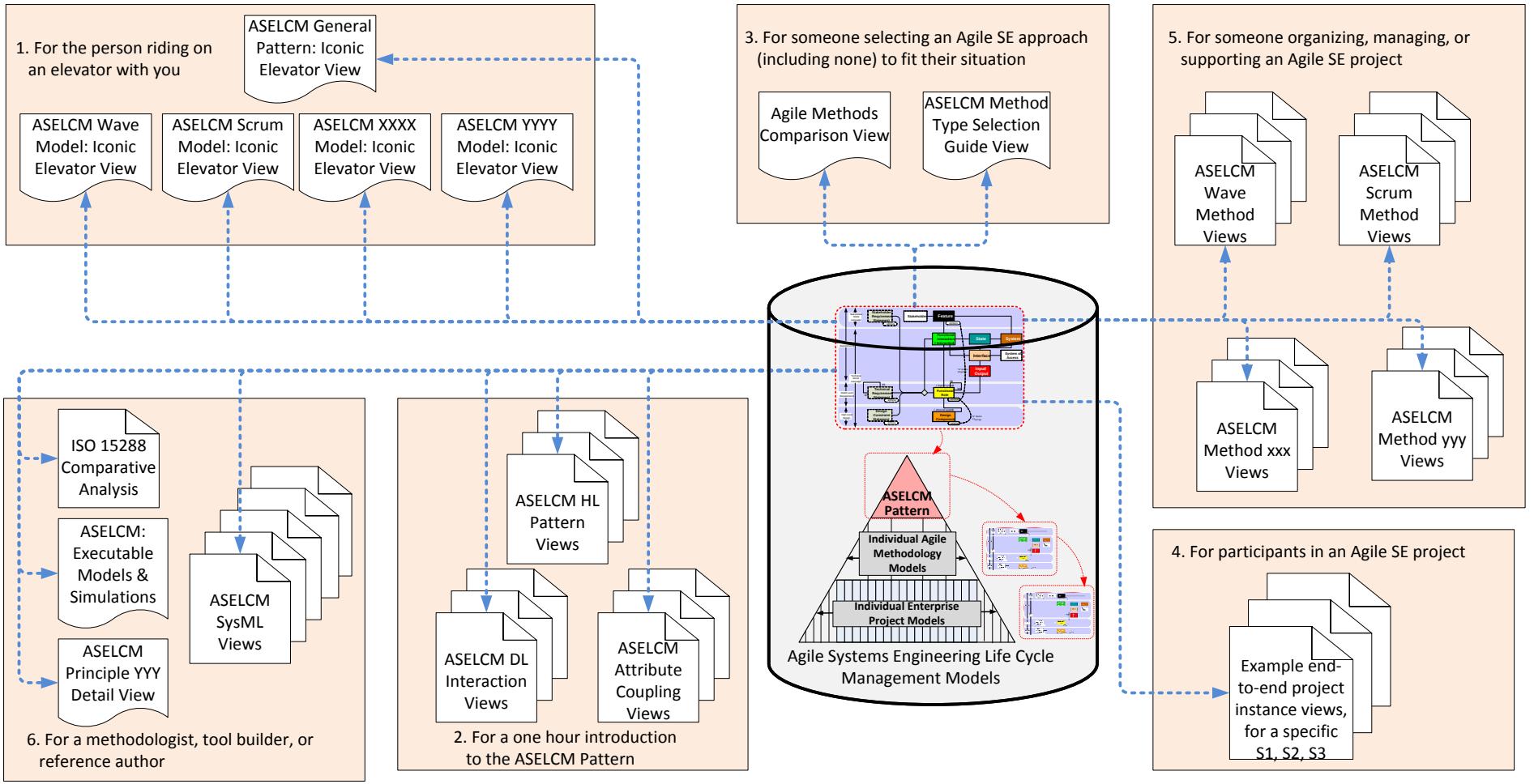
schindel@ictt.com 812-232-2062, attributed copies permitted

Different model users will need (very) different model views, of the same underlying model



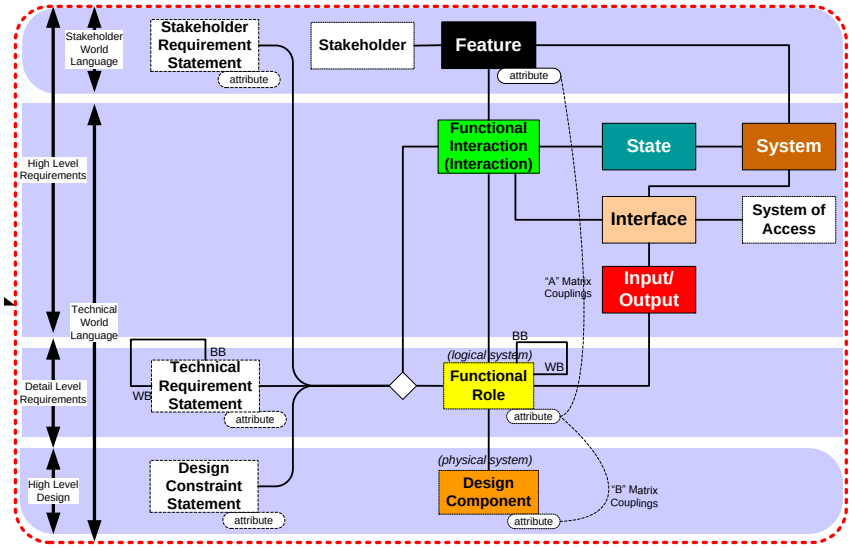
1. For the person riding on an elevator with you;
2. For a one hour introduction to Agile SE;
3. For someone selecting an Agile SE approach (including none) to fit their situation
4. For participants in an Agile SE project
5. For someone organizing, managing, or supporting an Agile SE project
6. For a methodologist, tool builder, or reference author

- Our core interest is in the general ASELC Model, but then …
- Then different configurations and views of that model, for different methodologies, users, and situations:

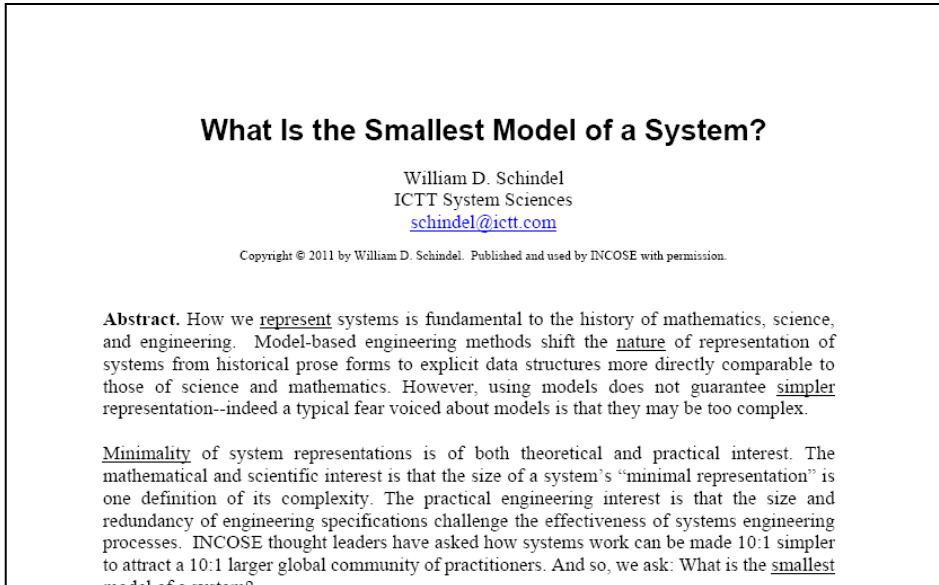


S*Models, S*Patterns: Using the S*Metamodel

- The model is being constructed as an S*Model:
  - That means it will conform to the underlying S*Metamodel (see Refs);
  - This is about including the minimal underlying concepts necessary to describe any system for engineering or scientific purposes;
  - Not about a modeling language: S*Models can be expressed in any of a number of contemporary modeling languages and toolsets; we may provide at least two such renderings for appeal to different groups;
  - The S*Metamodel is also the basis of the INCOSE Patterns WG's work.



Summary extract from S*Metamodel



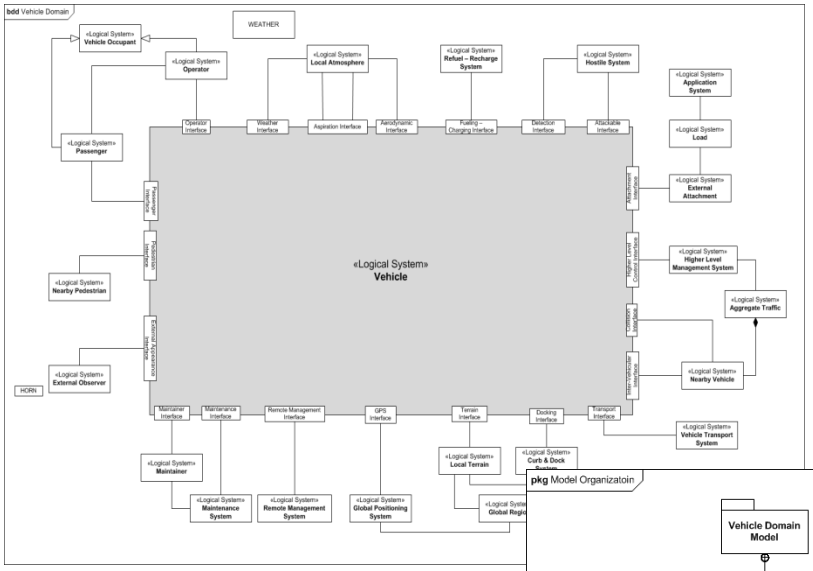
What Is the Smallest Model of a System?

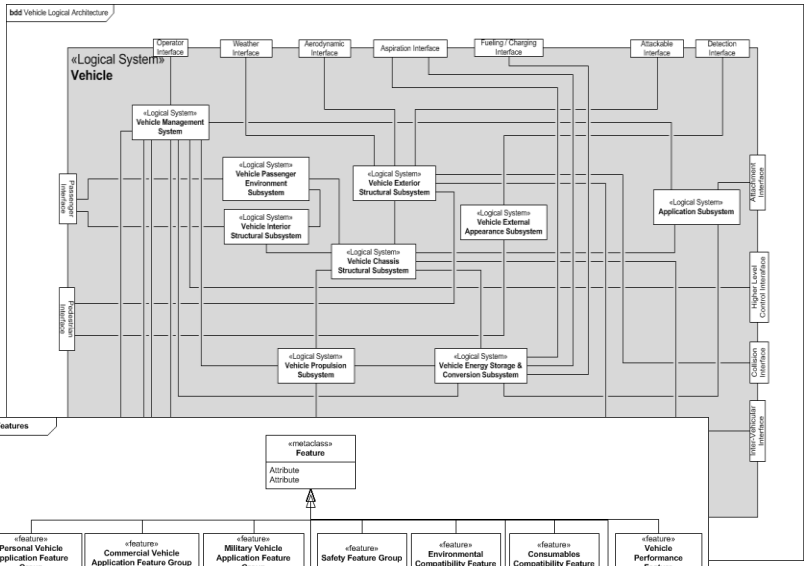
William D. Schindel
ICTT System Sciences
schindel@ictt.com

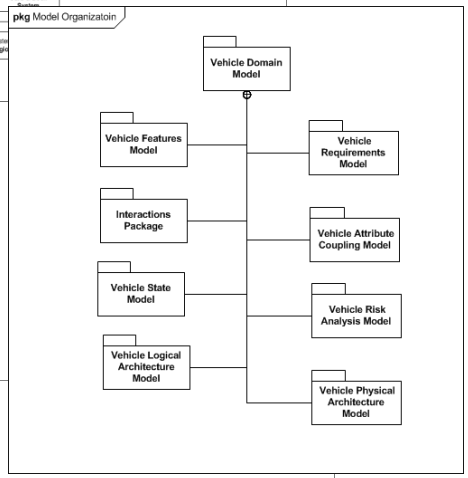
Abstract. How we represent systems is fundamental to the history of mathematics, science, and engineering. Model-based engineering methods shift the nature of representation of systems from historical prose forms to explicit data structures more directly comparable to those of science and mathematics.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

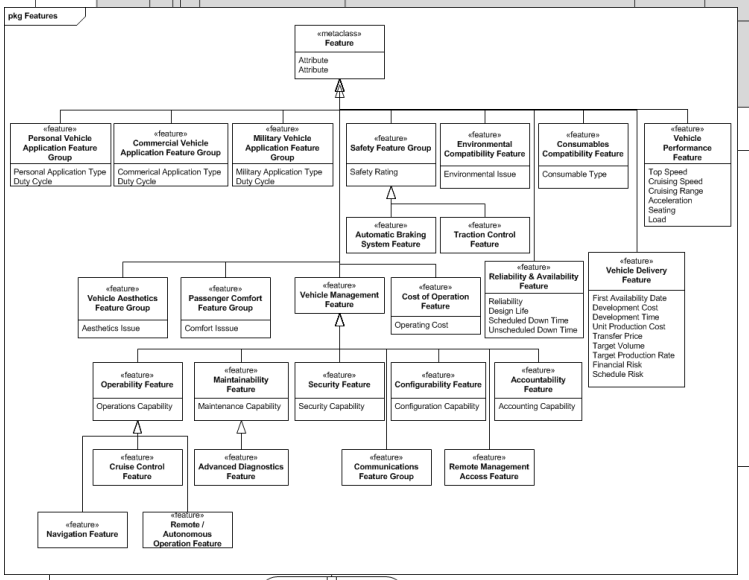
Minimality of system representations is of both theoretical and practical interest. The mathematical and scientific interest is that the size of a system's "minimal representation" is one definition of its complexity. The practical engineering interest is that the size and redundancy of engineering specifications 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?

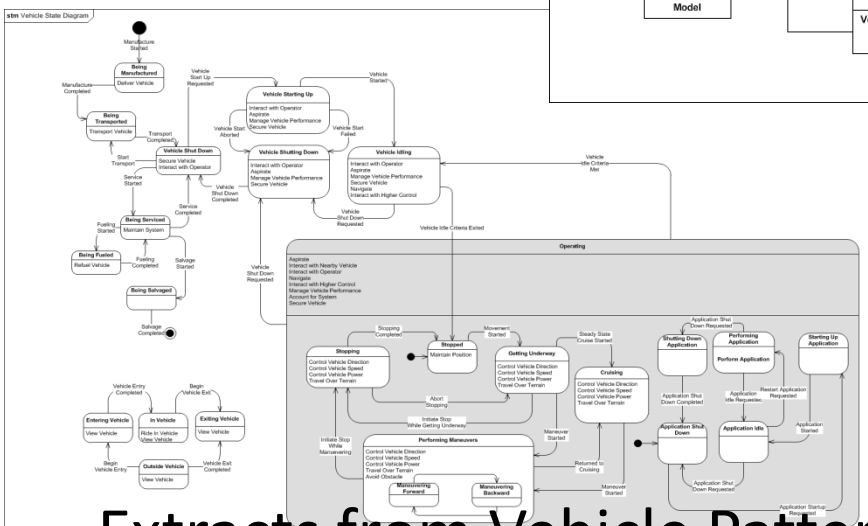
Introduction and Background: Size Matters!

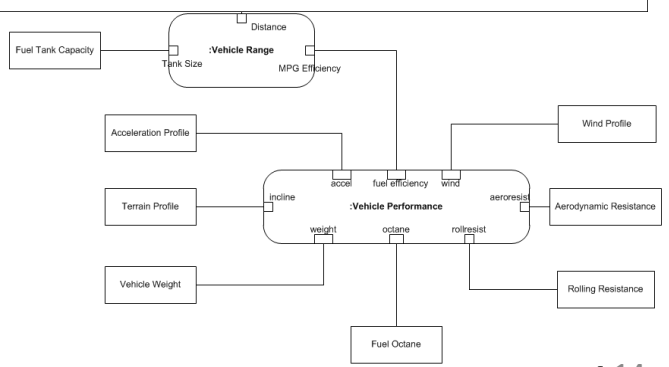










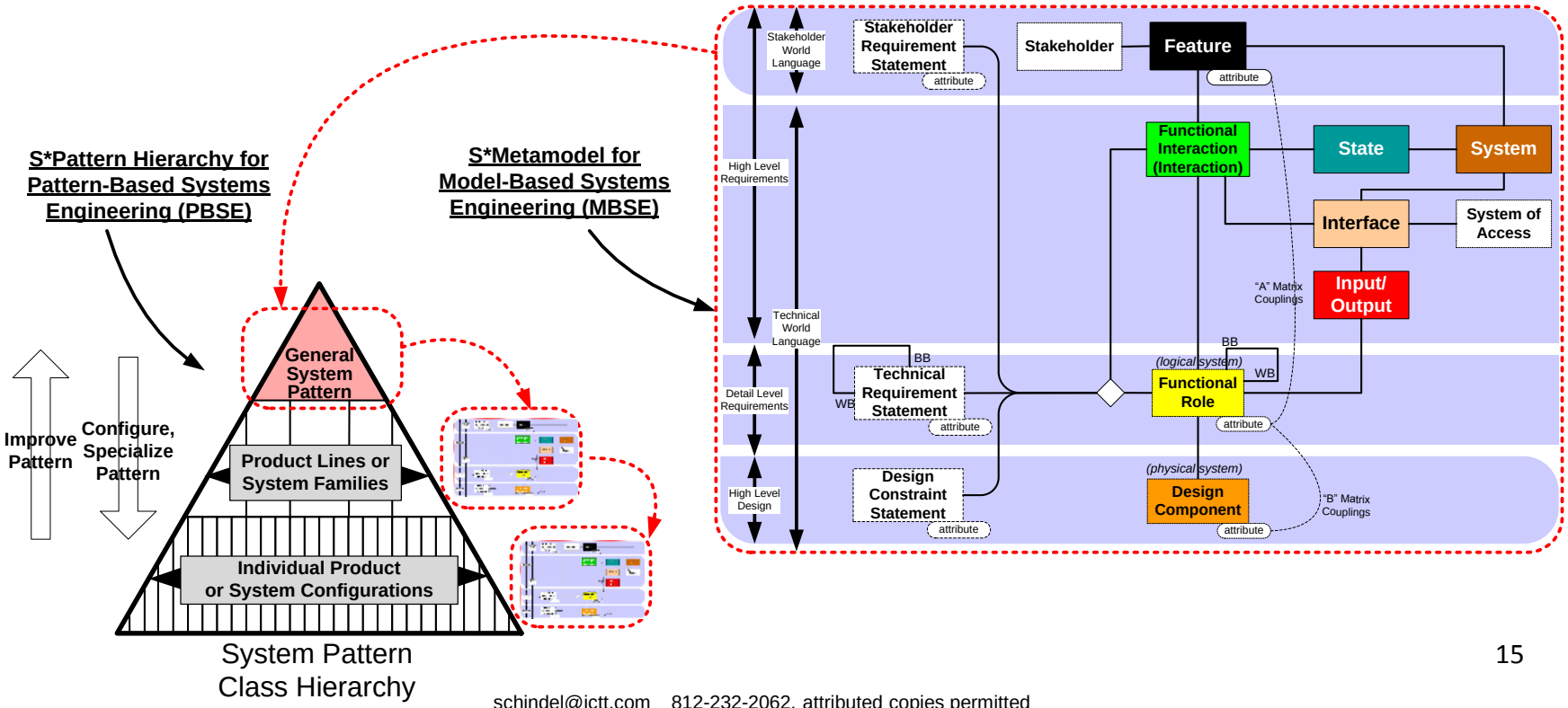


Extracts from Vehicle Pattern in 2013 INCOSE PBSE Tutorial

schindel@ictt.com 812-232-2062, attributed copies permitted

S*Models, S*Patterns: Using the S*Metamodel

- An S*Pattern is an S*Model of a system family, product line, platform, or other similar systems.
- An S*Pattern can be configured to produce an S*Model of a specific system type within the family.
- The ASELCM Pattern, represented as an S*Pattern, will be configurable to each of the special case forms of Agile Systems to be studied:

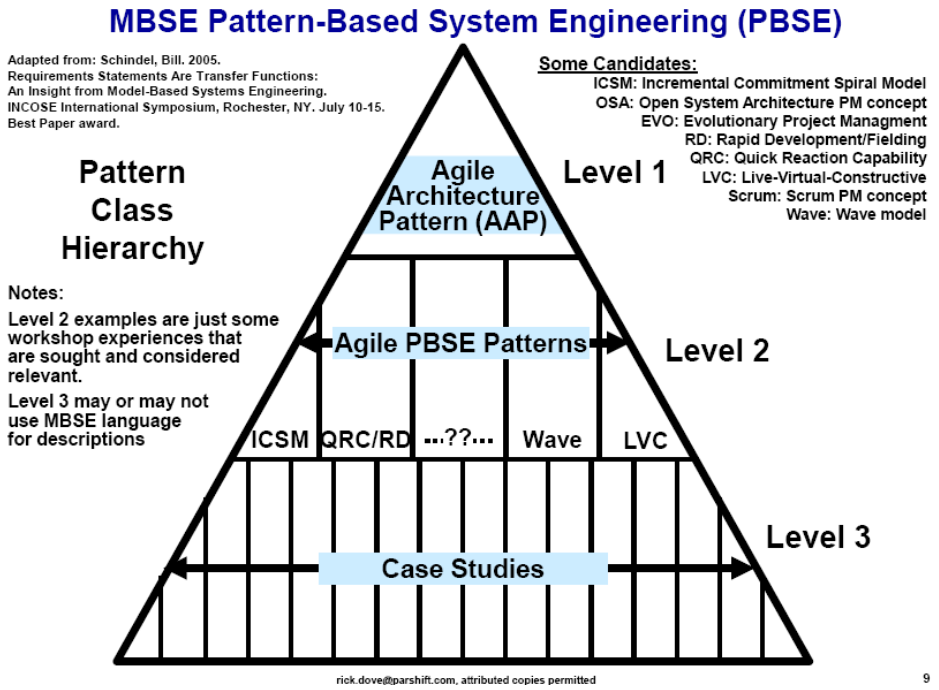


schindel@ictt.com 812-232-2062, attributed copies permitted

S*Models, S*Patterns: Using the S*Metamodel

- An S*Pattern is an S*Model of a system family, product line, platform, or other similar systems.
- An S*Pattern can be configured to produce an S*Model of a specific system type within the family.
- The ASELCM Pattern, represented as an S*Pattern, will be configurable to each of the special case forms of Agile Systems to be studied:

From R. Dove's ASELCM Project Summary:



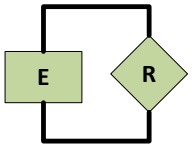
S*Pattern Class Hierarchy

More Abstract/General

More Specific

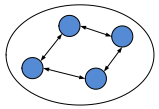
Definition of Relational Modeling

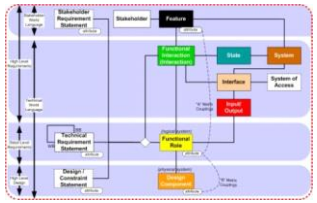
Entity-Relationship Paradigm



Minimal System S*Metamodel: Definition of (Elementary) System, Material Cause

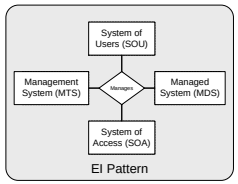
S*Metamodel Core

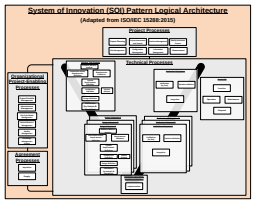




Emergence & Definition of Intelligence, Purpose, Stakeholders, Innovation, Final Cause, Formal Cause, Efficient Cause

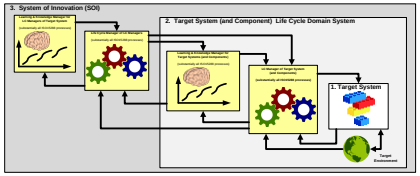
EI Pattern, SOI Pattern





Emergence & Definition of Agile Systems

General ASELCM Pattern



Incremental Commitment Spiral Model (ISCM) Pattern

Open System Architecture (OSA) Pattern

Evolutionary Project Management (EVO) Pattern

Product Line Engineering (PLE) Pattern

Rapid Development/ Fielding (RD) Pattern

Quick Reaction Capability (QRC) Pattern

Live-Virtual-Constructive (LVC) Pattern

Scrum Pattern

Wave Pattern

schindel@ictt.com 812-232-2062, attributed copies permitted

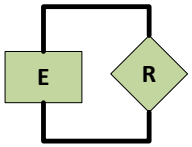
S*Pattern Class Hierarchy

More Abstract/General

More Specific

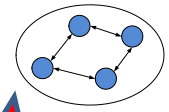
Definition of Relational Modeling

Entity-Relationship Paradigm



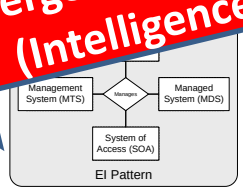
Minimal System S*Metamodel: Definition of (Elementary) System, Material Cause

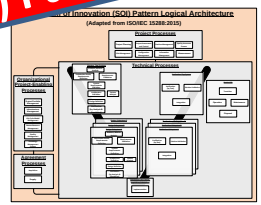
S*Metamodel Core



Emergence & Definition of Intelligence, Purpose, Stakeholders, Innovation, Final Cause, Formal Cause, Efficient Cause

EI Pattern, SOI Pattern

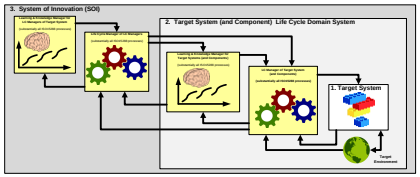




Emergence of Management (Intelligence) Pattern

Emergence & Definition of Agile Systems

General ASELCM Pattern



Incremental Commitment Spiral Model (ISCM) Pattern

Open System Architecture (OSA) Pattern

Evolutionary Project Management (EVO) Pattern

Product Line Engineering (PLE) Pattern

Rapid Development/ Fielding (RD) Pattern

Quick Reaction Capability (QRC) Pattern

Live-Virtual-Constructive (LVC) Pattern

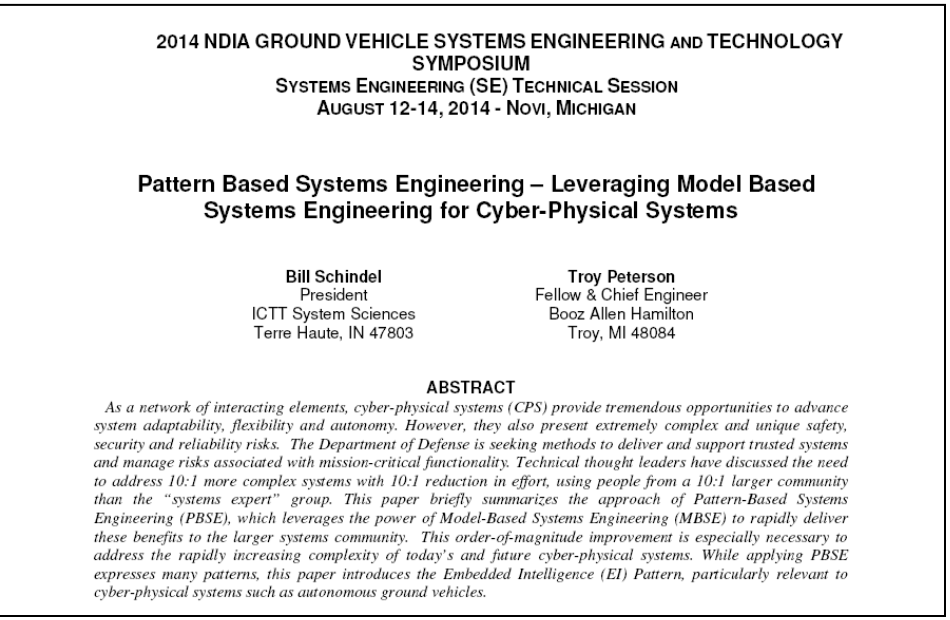
Scrum Pattern

Wave Pattern

schindel@ictt.com 812-232-2062, attributed copies permitted

Embedded Intelligence (EI) Pattern

- Express patterns important to life cycle management, complementary to ISO 15288 (also used here)
- Management of system Performance, Configuration, Security, Faults, and Accounting (SMFAs, from ISO 10040)



2014 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
SYSTEMS ENGINEERING (SE) TECHNICAL SESSION
AUGUST 12-14, 2014 - NOVI, MICHIGAN

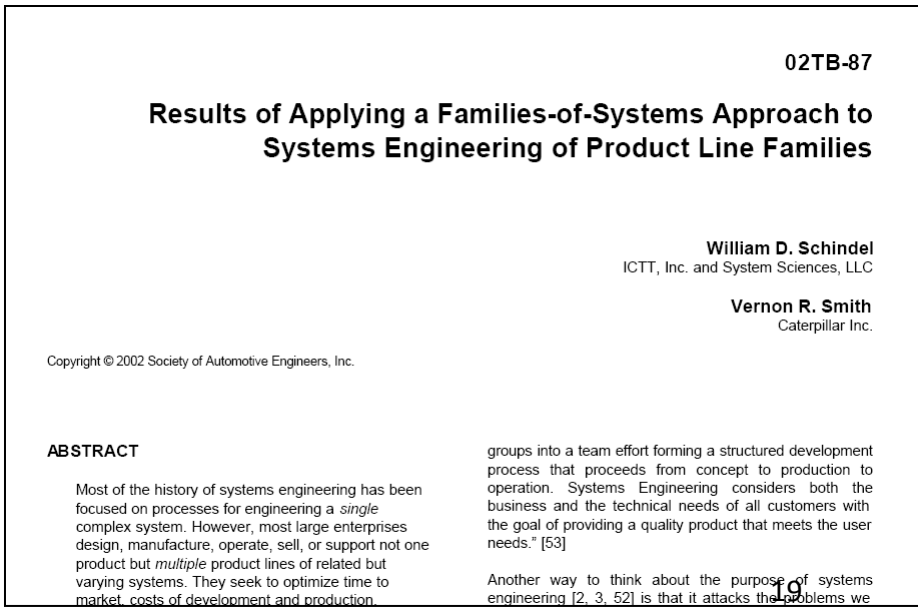
Pattern Based Systems Engineering – Leveraging Model Based Systems Engineering for Cyber-Physical Systems

Bill Schindel
President
ICTT System Sciences
Terre Haute, IN 47803

Troy Peterson
Fellow & Chief Engineer
Booz Allen Hamilton
Troy, MI 48084

ABSTRACT

As a network of interacting elements, cyber-physical systems (CPS) provide tremendous opportunities to advance system adaptability, flexibility and autonomy. However, they also present extremely complex and unique safety, security and reliability risks. The Department of Defense is seeking methods to deliver and support trusted systems and manage risks associated with mission-critical functionality. Technical thought leaders have discussed the need to address 10:1 more complex systems with 10:1 reduction in effort, using people from a 10:1 larger community than the "systems expert" group. This paper briefly summarizes the approach of Pattern-Based Systems Engineering (PBSE), which leverages the power of Model-Based Systems Engineering (MBSE) to rapidly deliver these benefits to the larger systems community. This order-of-magnitude improvement is especially necessary to address the rapidly increasing complexity of today's and future cyber-physical systems. While applying PBSE expresses many patterns, this paper introduces the Embedded Intelligence (EI) Pattern, particularly relevant to cyber-physical systems such as autonomous ground vehicles.



02TB-87

Results of Applying a Families-of-Systems Approach to Systems Engineering of Product Line Families

William D. Schindel
ICTT, Inc. and System Sciences, LLC

Vernon R. Smith
Caterpillar Inc.

ABSTRACT

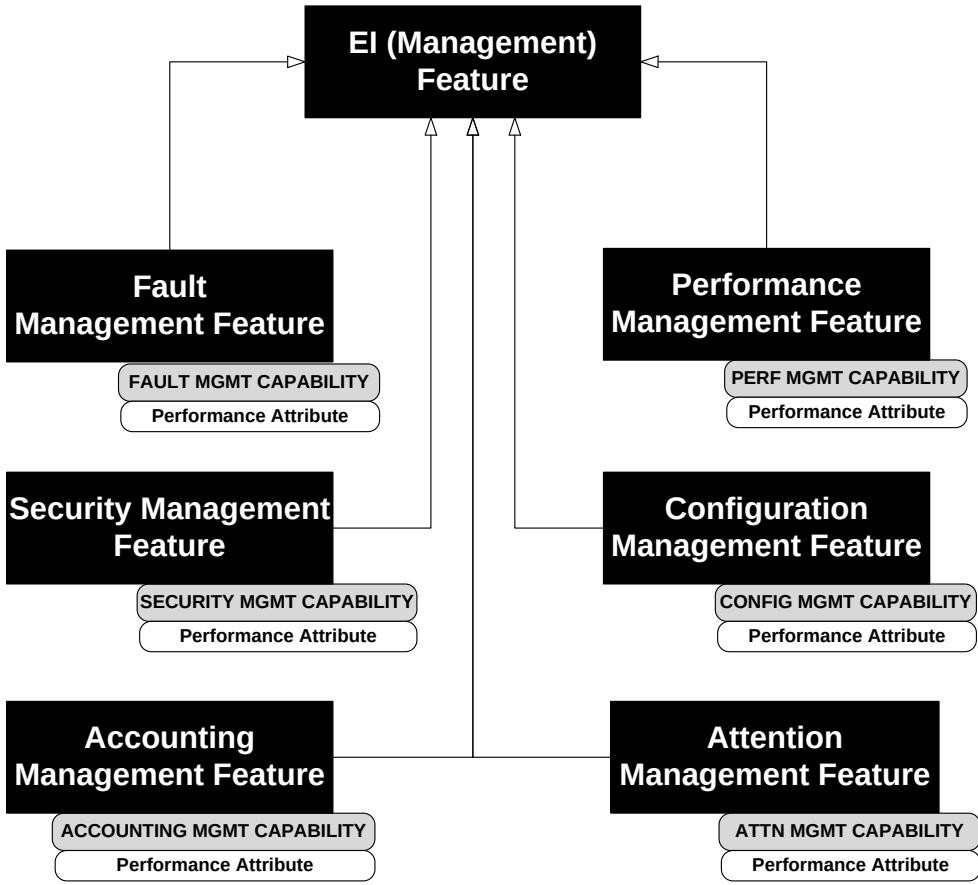
Most of the history of systems engineering has been focused on processes for engineering a *single* complex system. However, most large enterprises design, manufacture, operate, sell, or support not one product but *multiple* product lines of related but varying systems. They seek to optimize time to market, costs of development and production,

groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs." [53]

Another way to think about the purpose of systems engineering [2, 3, 52] is that it attacks the problems we

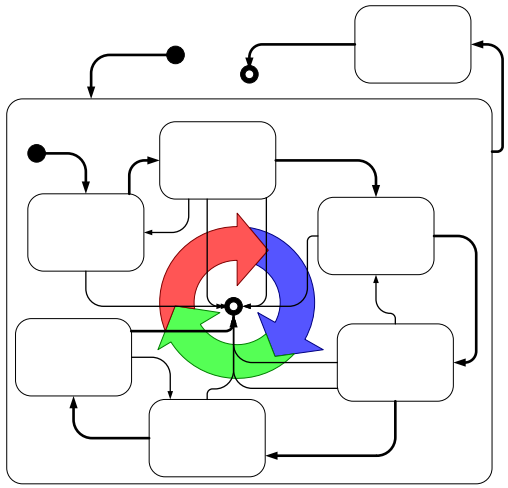
Embedded Intelligence (EI) Pattern—Summary of Feature Portion

- Features express selectable stakeholder values, system capabilities.
- Expressing value of performance of development / innovation / life cycle management processes.
- Specialized in ASELCM Pattern to capabilities that differentiate various agile methods, and how to select the most fitting one in different circumstances.



Embedded Intelligence (EI) Pattern

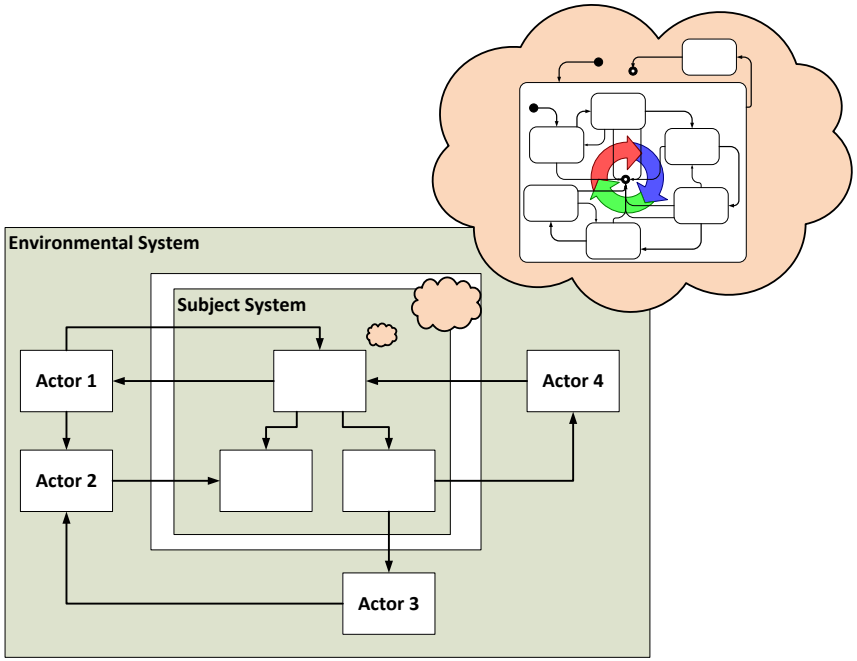
- The State Model portion of the EI Pattern provides insight into the nature of the “regulatory” role of embedded intelligence.
- These show numerous “situation resolution cycles” that drive the managed system to nominal states, when various situations are encountered:
  - Major mission cycles, from 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
- Describe agile responses



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.



Attention Management Feature

Next, we add learning (accumulation of experience) through the emergence of the Systems of Innovation Pattern . . . .

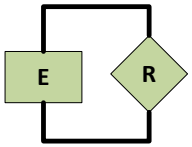
S*Pattern Class Hierarchy

More Abstract/General

More Specific

Definition of Relational Modeling

Entity-Relationship Paradigm

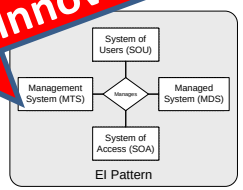


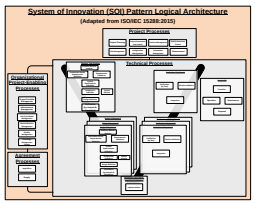
Minimal System S*Metamodel: Definition of (Elementary) System, Material Cause

S*Metamodel Core

Emergence & Definition of Intelligence, Purpose, Stakeholders, Innovation, Final Cause, Formal Cause, Efficient Cause

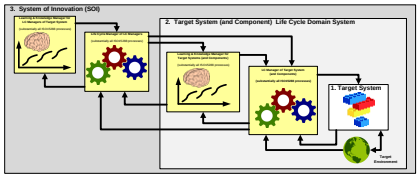
EI Pattern, SOI Pattern





Emergence & Definition of Agile Systems

General ASELCM Pattern



Emergence of Systems of Innovation (SOI) Pattern

Incremental Commitment Spiral Model (ISCM) Pattern

Open System Architecture (OSA) Pattern

Evolutionary Project Management (EVO) Pattern

Product Line Engineering (PLE) Pattern

Rapid Development/ Fielding (RD) Pattern

Quick Reaction Capability (QRC) Pattern

Live-Virtual-Constructive (LVC) Pattern

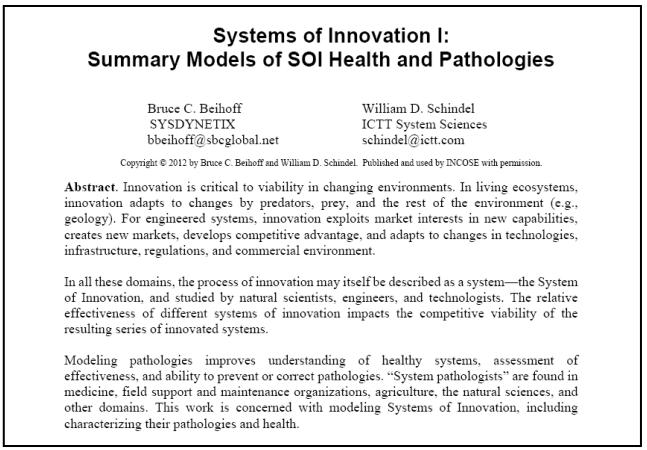
Scrum Pattern

Wave Pattern

schindel@ictt.com 812-232-2062, attributed copies permitted

Systems of Innovation (SOI) Pattern

- Focuses on Feature "selection" as paradigm for exploratory learning and improvement—whether by markets, designers, managers, competition, or Nature, in fitness (Feature) space.
- Emergence of Purpose



Systems of Innovation I:
Summary Models of SOI Health and Pathologies

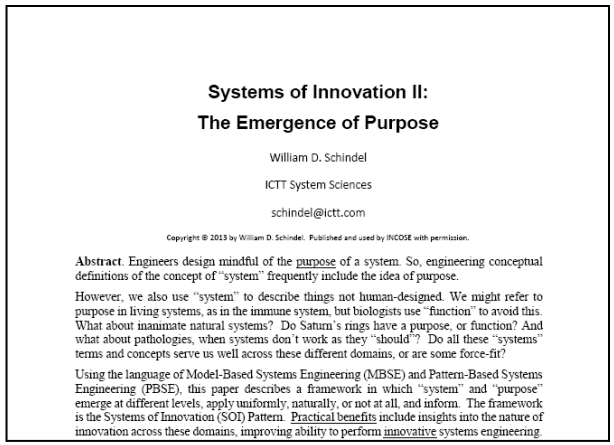
Bruce C. Beihoff
SYSDYNETIX
bbeihoff@sbcglobal.net

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

Abstract. Innovation is critical to viability in changing environments. In living ecosystems, innovation adapts to changes by predators, prey, and the rest of the environment (e.g., geology). For engineered systems, innovation exploits market interests in new capabilities, creates new markets, develops competitive advantage, and adapts to changes in technologies, infrastructure, regulations, and commercial environment.

In all these domains, the process of innovation may itself be described as a system—the System of Innovation, and studied by natural scientists, engineers, and technologists. The relative effectiveness of different systems of innovation impacts the competitive viability of the resulting series of innovated systems.

Modeling pathologies improves understanding of healthy systems, assessment of effectiveness, and ability to prevent or correct pathologies. "System pathologists" are found in medicine, field support and maintenance organizations, agriculture, the natural sciences, and other domains. This work is concerned with modeling Systems of Innovation, including characterizing their pathologies and health.



Systems of Innovation II:
The Emergence of Purpose

William D. Schindel

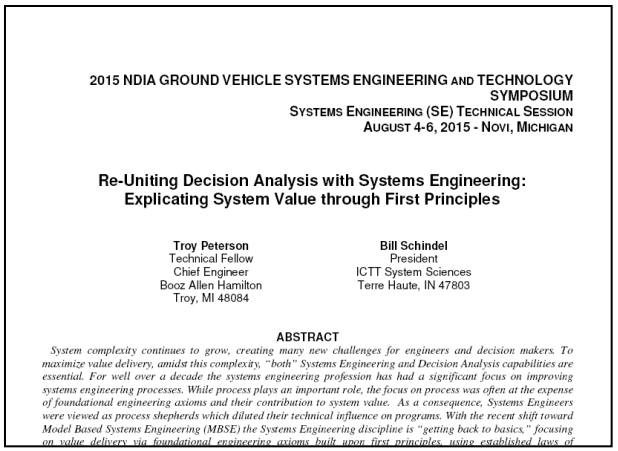
ICTT System Sciences

schindel@ictt.com

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 is the Systems of Innovation (SOI) Pattern. Practical benefits include insights into the nature of innovation across these domains, improving ability to perform innovative systems engineering.



2015 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
SYSTEMS ENGINEERING (SE) TECHNICAL SESSION
AUGUST 4-6, 2015 - NOVI, MICHIGAN

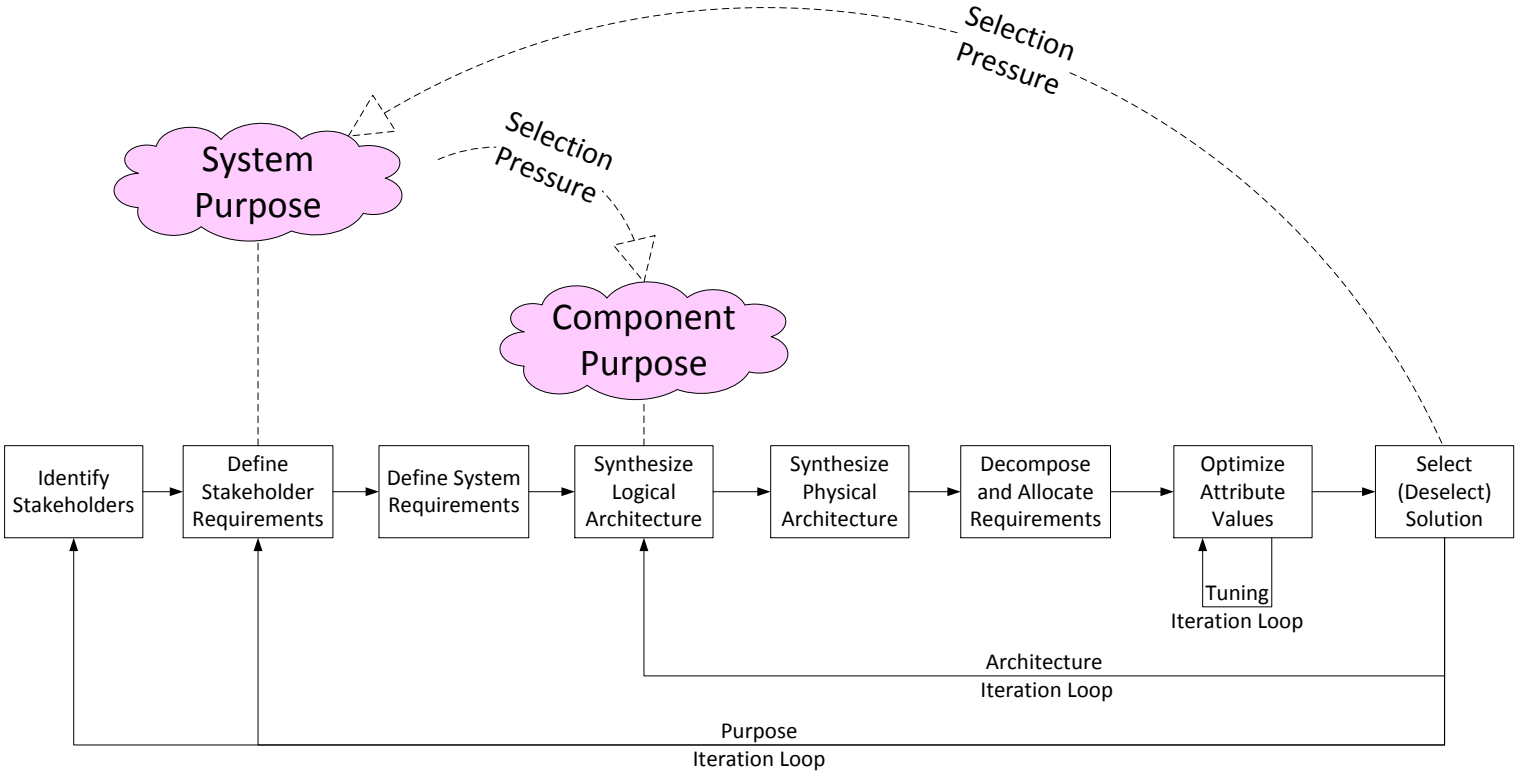
Re-Uniting Decision Analysis with Systems Engineering:
Explicating System Value through First Principles

Troy Peterson
Technical Fellow
Chief Engineer
Booz Allen Hamilton
Troy, MI 48084

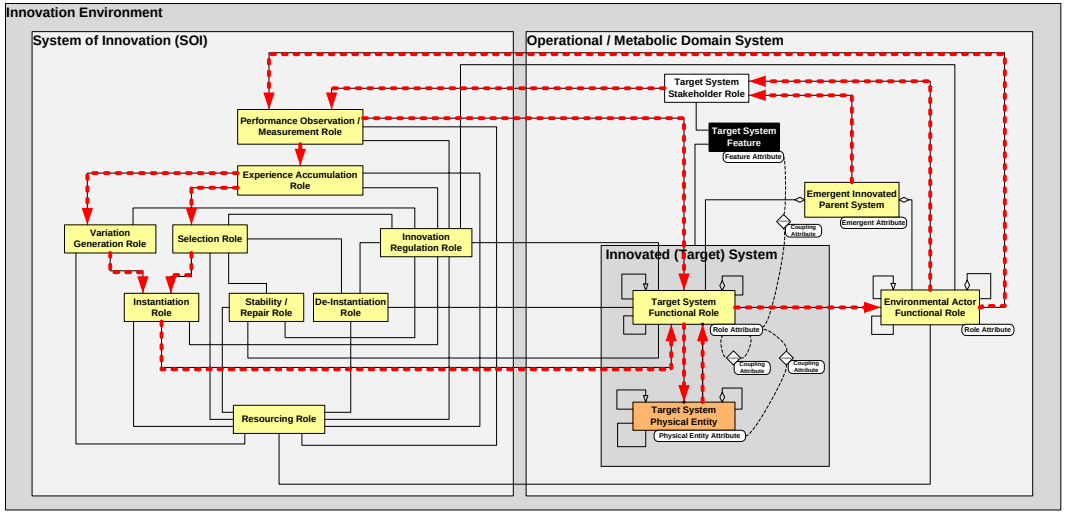
Bill Schindel
President
ICTT System Sciences
Terre Haute, IN 47803

ABSTRACT

System complexity continues to grow, creating many new challenges for engineers and decision makers. To maximize value delivery, amidst this complexity, "both" Systems Engineering and Decision Analysis capabilities are essential. For well over a decade the systems engineering profession has had a significant focus on improving systems engineering processes. While process plays an important role, the focus on process was often at the expense of foundational engineering axioms and their contribution to system value. As a consequence, Systems Engineers were viewed as process shepherds which diluted their technical influence on programs. With the recent shift toward Model Based Systems Engineering (MBSE) the Systems Engineering discipline is "getting back to basics," focusing on value delivery via foundational engineering axioms built upon first principles, using established laws of

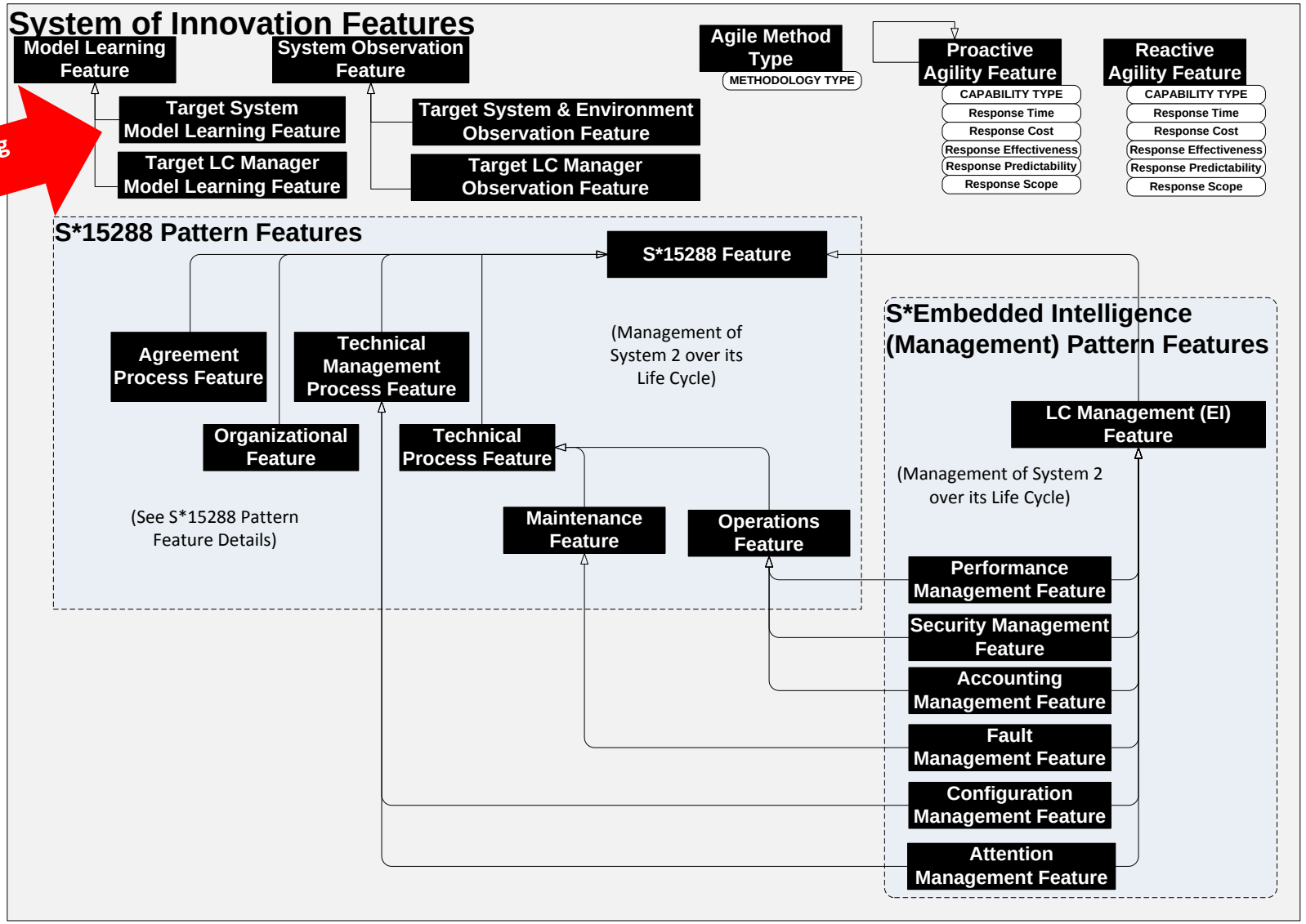


Signaling loop diagram



Systems of Innovation (SOI) Pattern

(more on this later in ASELCM Pattern)



schindel@ictt.com 812-232-2062, attributed copies permitted

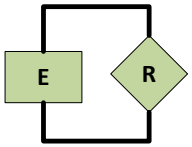
S*Pattern Class Hierarchy

More Abstract/General

More Specific

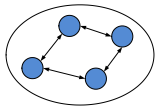
Definition of Relational Modeling

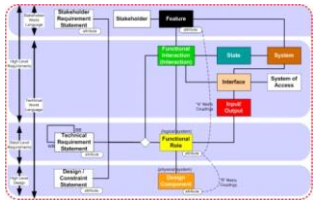
Entity-Relationship Paradigm



Minimal System S*Metamodel: Definition of (Elementary) System, Material Cause

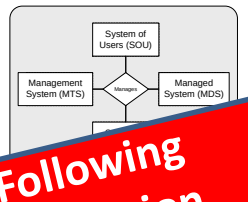
S*Metamodel Core

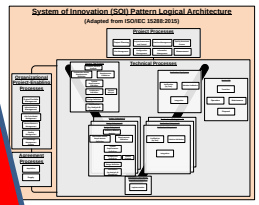




Emergence & Definition of Intelligence, Purpose, Stakeholders, Innovation, Final Cause, Formal Cause, Efficient Cause

EI Pattern, SOI Pattern

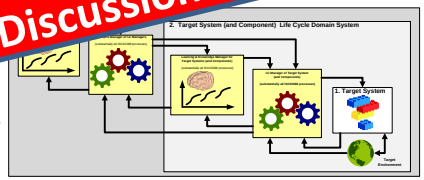




Following Discussion

Emergence & Definition of Agile Systems

General ASELCM Pattern



Incremental Commitment Spiral Model (ISCM) Pattern

Open System Architecture (OSA) Pattern

Evolutionary Project Management (EVO) Pattern

Product Line Engineering (PLE) Pattern

Rapid Development/ Fielding (RD) Pattern

Quick Reaction Capability (QRC) Pattern

Live-Virtual-Constructive (LVC) Pattern

Scrum Pattern

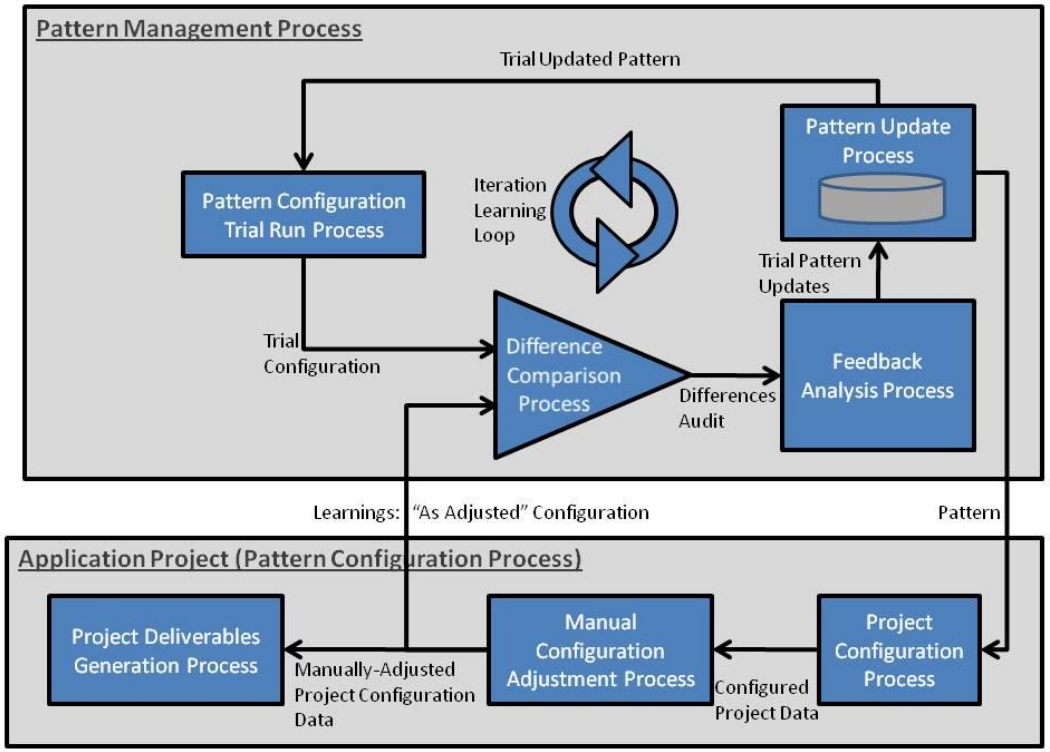
Wave Pattern

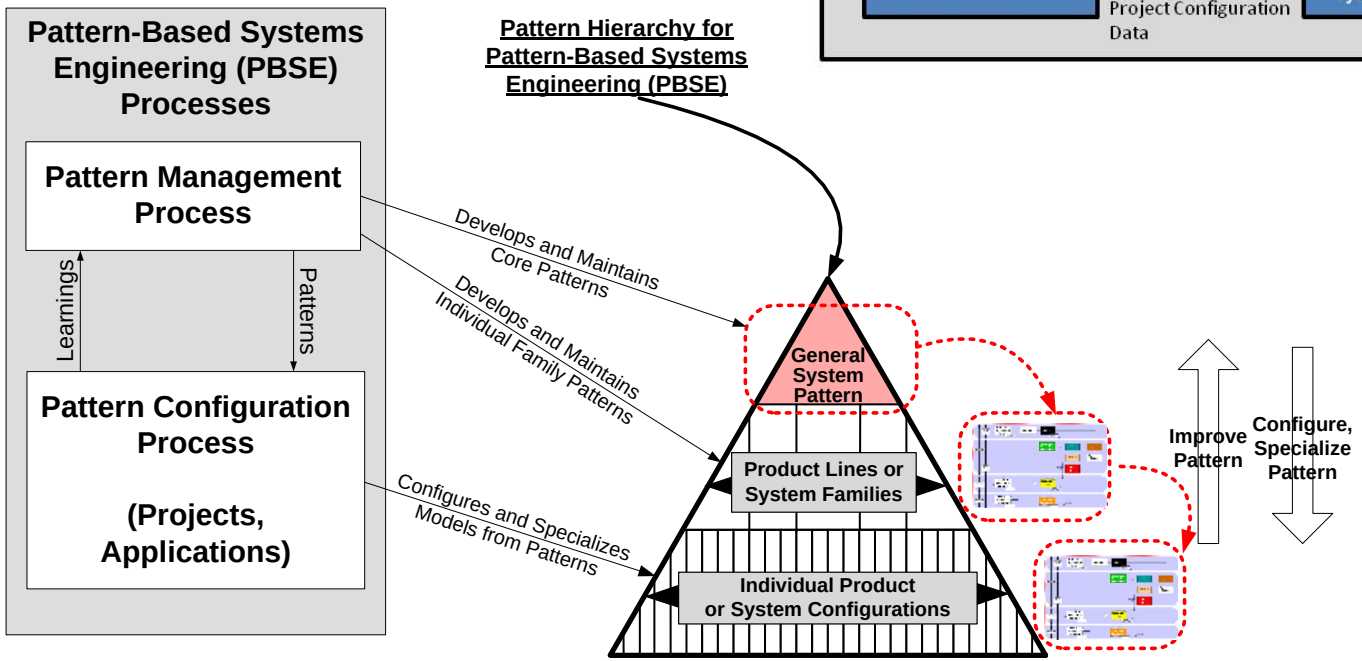
schindel@ictt.com 812-232-2062, attributed copies permitted

Before, during, after initial ASELCM host workshops

- **2H2014**: Initial work by leads of ASELCM Project (and the Patterns WG) to start a preliminary ASELCM Pattern draft, for testing. Several reviews in late 2014, before IW2015.
- **Jan, 2015**: Presented part of it (system boundaries) at the IW2015 MBSE Workshop session on Agile Systems, for feedback.
- Initial workshops contributed to, or affirmed:
  - **Host workshop 1**: System 1, 2, 3 reference boundaries; awareness and attention model (management and team)
  - **Host workshop 2:** Explicit partner risk allocation and spreading; Scrum model
  - **Host workshop 3:** Product Line Engineering (PLE / PBSE)
  - **Host workshop 4:** ASE assimilation in a defense acquisition environment; Information Debt in addition to Technical Debt
- **Oct, 2015:** Presented Introduction version to INCOSE Great Lakes SE Conference, obtaining feedback
- **Jan, 2016**: Joint review with Patterns WG and Agile WG at IW2016, seeking feedback.
- **Future workshops**: Will similarly configure and test against what we see, while extending and refining.

Testing & improving the model





Pattern Class Hierarchy

schindel@ictt.com 812-232-2062, attributed copies permitted

Testing & improving the model

- Test planning, criteria, fitness for use
- Checking against:
  - The studied site hosts' information
  - The existing agile literature
  - Other sources
- The same comparison feedback loop is also used to improve the pattern, ongoing.
- Underlying model aspects to test:
  - Qualitative elements, relationships, structures
  - Including causal loops (what is impacted by what), principles
  - Quantitative gaps: If we can obtain quantitative data from hosts, or plan other experiments
- Testing of the "views" of the model:
  - Simple enough? Expressive enough? For the intended uses and users
  - Recognition of configured cases by the host organizations we study
  - Recognition of the principles and configured cases by the study team

What is the status of the ASELCM Pattern?

- Pattern high level (HLR) constructs:
  - Includes System boundaries, Features/Feature Attributes, Architectural Relationships, Interactions, States, Interfaces
  - Identified and relatively stable
- Pattern detail level (DLR) constructs:
  - Includes Requirements, I/Os, Systems of Access, Attribute Couplings
  - Much of this still to be worked, although details less interesting to some audiences
- Pattern specializations to Agility Archetypes:
  - Likely to include at least Wave, Scrum, LVC, PLE, others
  - Will continue to evolve as host sites are visited
  - We are only part way into visiting variant sites—need commercial examples and more domains to test
- Also reviewed in IS2016 papers.

What is the status of the ASELCM Pattern?

- Languages and tools:
  - Most likely language targets are SysML and IDEF (because of handbook), but others are possible if there is strong enough interest
  - The underlying S*Metamodel on which the ASELCM Pattern is based is mapped to target schema, making the ASELCM Pattern relatively portable.
  - Probably will target two COTS tools in resulting deliverables.
- Portability and access:
  - We don't expect to make this pattern directly available on public COTS tools until reported out of the project
  - Initial pattern snapshots availability within the two collaborating INCOSE Working Groups

What comes next?

- More host sites or other access to commercial and other domains, agility approaches.
- More detail level (DLR) insertions to model
- Migration to targeted languages and COTS tools.
- Specialization to agility archetypes
- Generation of report, with configured examples
- Access

Where can I learn more?

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 | Transportatio

Great Lakes Regional Conference 2015

INCOSE V1.3

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

Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern

Bill Schindel
ICTT System Sciences
schindel@ictt.com

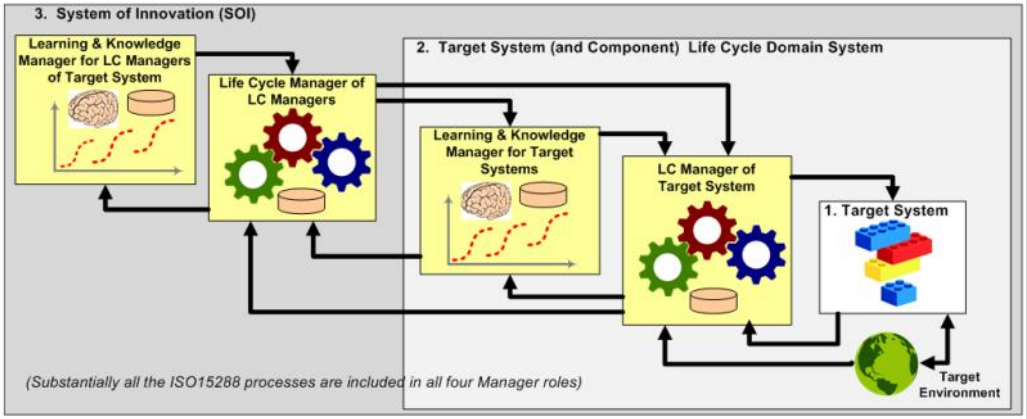
Rick Dove
Paradigm Shift, International Inc.
rick.dove@parshift.com

Abstract. Engineered and other systems are under pressure to adapt, from opportunities or competition, predators, changing environment, and physical or cyberattack. Ability to adapt well enough as conditions change, especially in presence of uncertainty, is valued. Systems (including developmental and life cycle management) that adapt well enough, in time, cost, and effectiveness, are sometimes called "agile". As environmental change or uncertainty increase, agility can mean survival. Agile systems and agile systems engineering are subjects of an INCOSE 2015-16 discovery project, described elsewhere. This paper introduces the underlying MBSE-based Agile Systems Engineering Life Cycle Pattern being used to capture, analyze, and communicate key aspects of systems being studied. More than an ontology, this model helps us understand necessary and sufficient conditions for agility, different approaches to it, and underlying relationships, performance couplings, and principles. This paper introduces the framework, while specific findings about methods and practicing enterprises studied will be reported separately.

Agile Systems, Agile Systems Engineering, and Why They Matter

Agile systems-engineering and agile-systems engineering are two different concepts (Haberfellner and de Weck 2005), but both are designed for change. They can be augmented

The INCOSE ASELCM Pattern: A Reference Model for Agility in Systems



INCOSE 2016 international workshop
Los Angeles, CA, USA
January 30 - February 2, 2016

REVIEW OF PATTERN CONTENT:
January 31, 2016, Patterns WG Meeting
Bill Schindel

V1.1.1

Current draft of the ASELCM Pattern

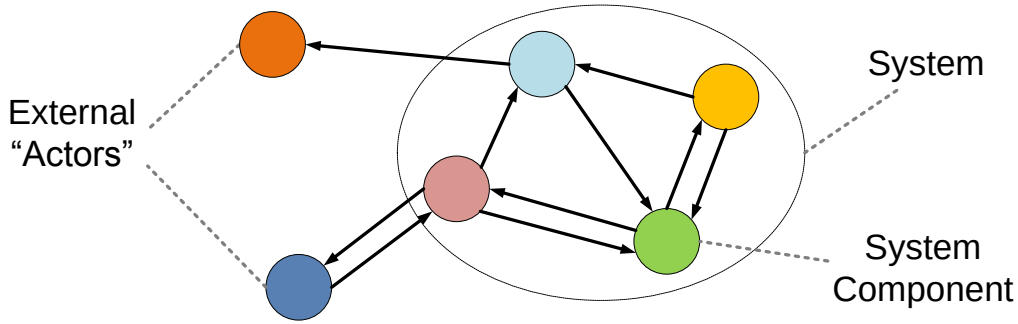
- The general ASELCM Pattern
- Host site archetypes/configurations of pattern (observations):
  - Wave/Navy SPAWAR
  - Scrum/Northrup Grumman
  - PLE/Rockwell-Collins
  - Adapted SAFe®/Lockheed Martin

Current draft: The general ASELCM Pattern

- Views included in current draft:
  - ☑ Managed Information Model
  - ☑ Patterns Hierarchy
  - ☑ Domain Model / System Reference Boundaries
  - ☑ Multiple S*Metaclass overview across systems
  - ☑ Logical Architecture Model
  - ☑ Physical Architecture Model
  - ☑ Stakeholder Features Model
  - ☑ Attribute Couplings Model
  - ☑ States & Interactions Model
  - □ *Requirements Model*

General ASELCM Pattern: Domain Model / System Ref. Boundaries

- S*Models define a system as a set of interacting components:



- By "interact", we mean exchange of energy, force, mass, or information, resulting in component change of state.
- By "state", we mean the condition of a thing that influences its behavior in subsequent interactions.

The Agile System Domain Model

- This portion of the Agile Pattern identifies system boundaries (scope) and a few main system names:
  - Rick's differentiation of the two systems in IS2014 Agile Systems, Parts 1 and 2 papers, important to the domain model--even though we intend to "make both of those systems agile"--

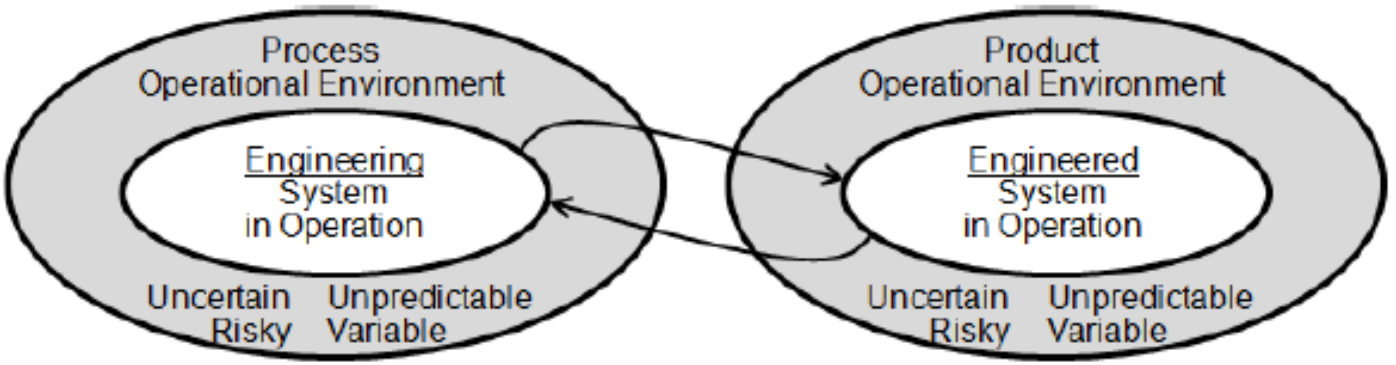


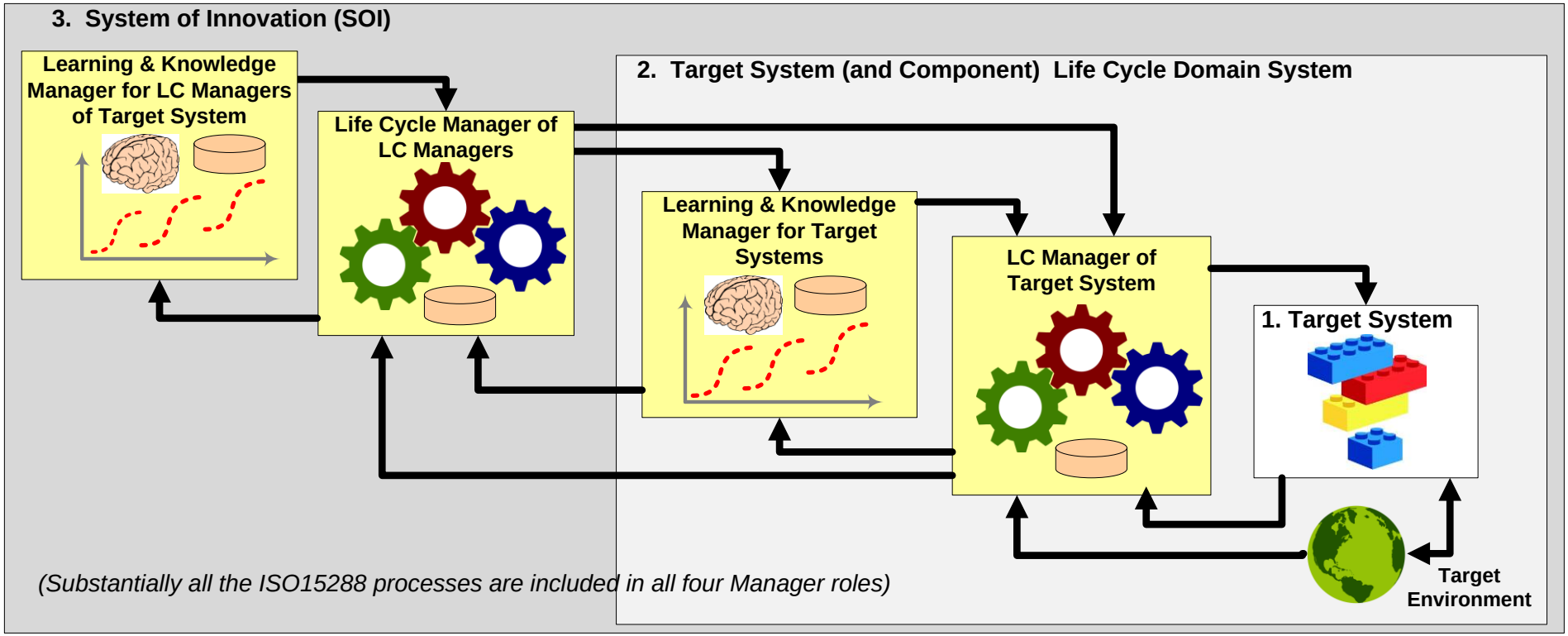
Figure 1: Two different operational environments defining necessary agile counterpoint for the systems they encompass.

From: Dove & LaBarge, "Fundamentals of Agile Systems Engineering—Part 2", IS2014.

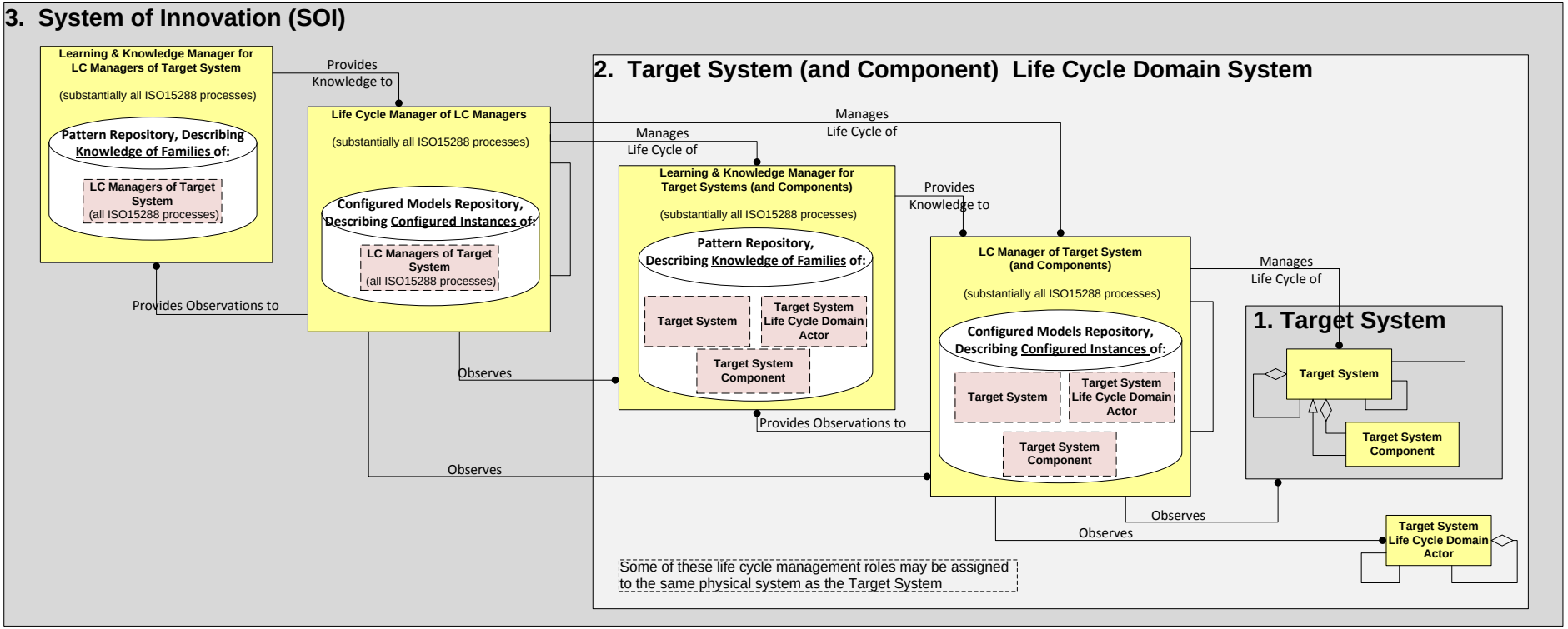
The Agile System Domain Model

- We will particularly refer to **three major system boundaries**:
  - To avoid a confusion bog of loaded terms, we could have just named them “System 1”, “System 2”, and “System 3” 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 informal “iconic” view



Behind the “iconic” diagram, there is a formal MBSE model that describes the ASELCM Pattern

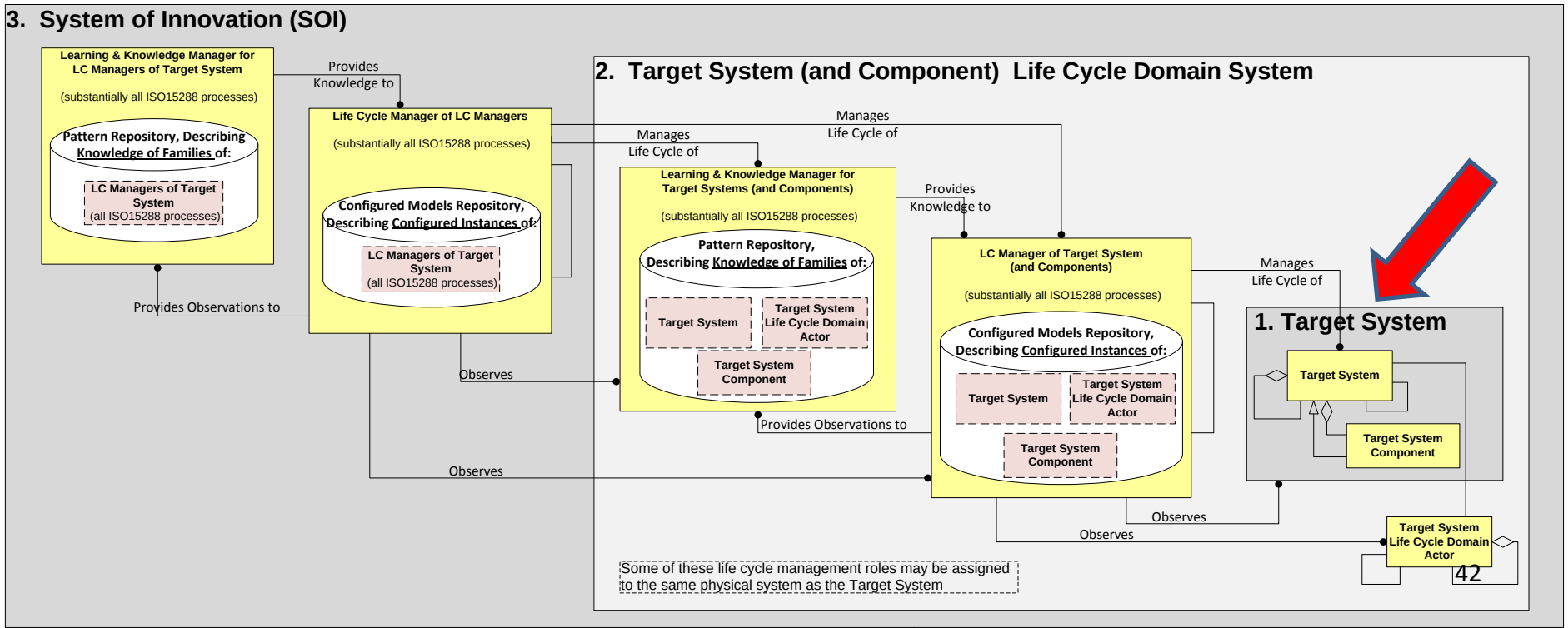


schindel@ictt.com 812-232-2062, attributed copies permitted

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.

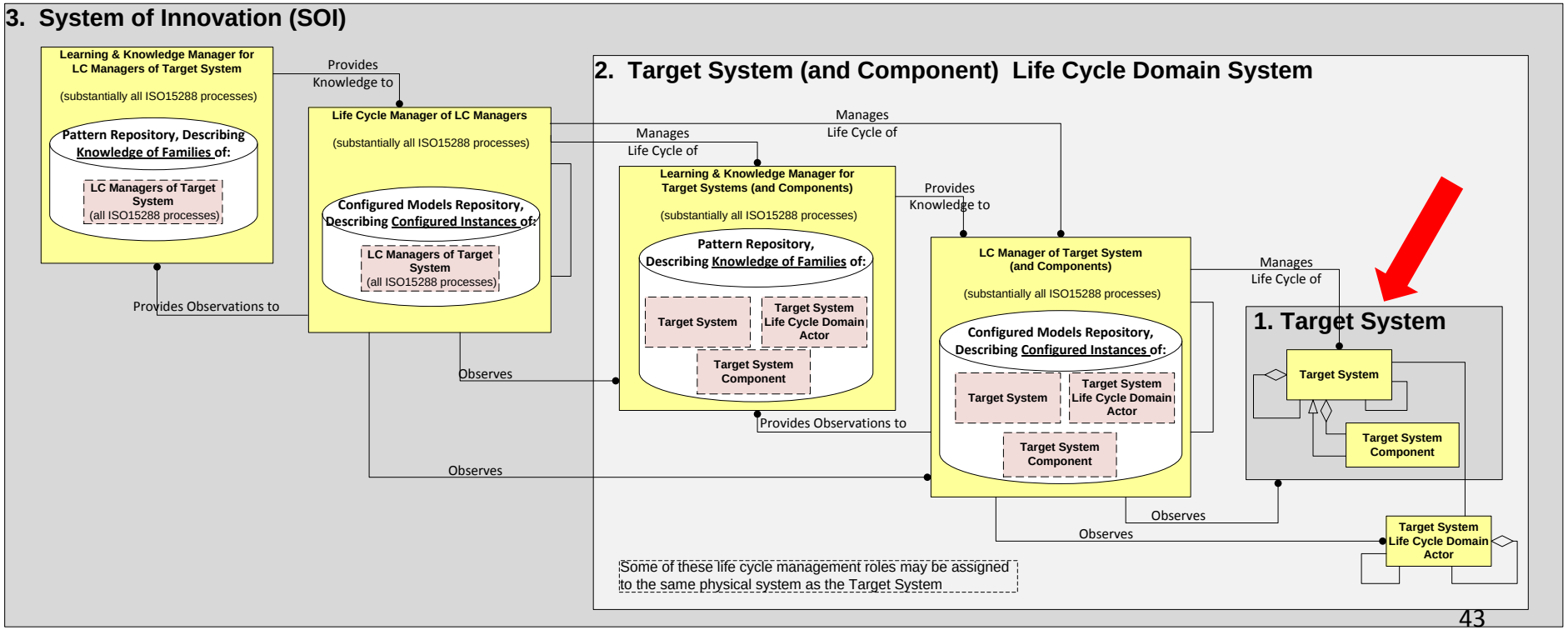
- Its behavior, characteristics, or performance are targets of the innovation (change, adaptation) process we'll introduce later.
- It is potentially agile. (Assertion: for SE to be fully agile, so must its target)
- Examples potentially include aircraft, satellites, the human immune system, software, restaurants, birds, and the health care delivery system.



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., Parent System of Systems, Target System, Target System Component, etc.)

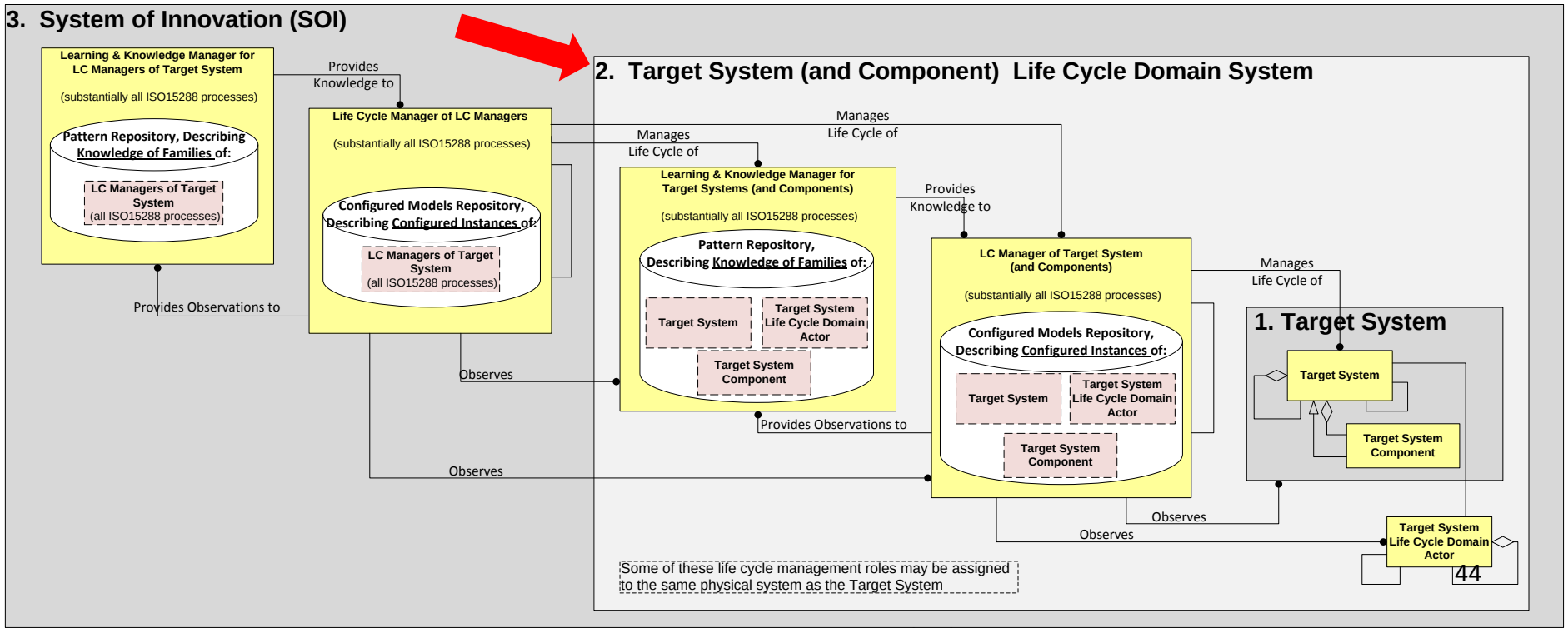


schindel@ictt.com 812-232-2062, attributed copies permitted

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 with which the Target System will directly interact any time during its life cycle*:

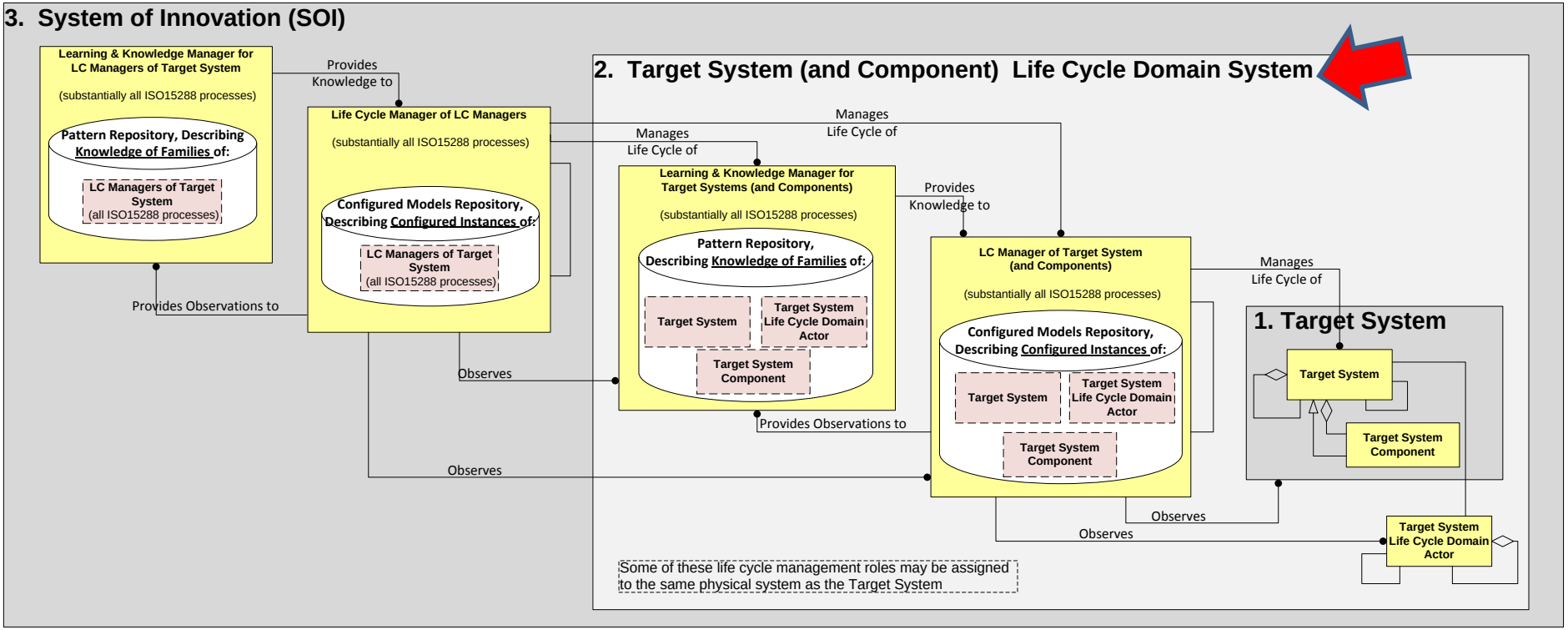
- This includes (among others) any system that directly manages the life cycle of an instance of a Target System (or a Component)—development, production and integration systems, maintenance and operations systems, and others.



The Agile System Domain Model

The System 2 model recognizes three systems besides the Target System:

- Target System Life Cycle Domain Actors
- LC Manager of Target Systems (and Components)
- Learning & Knowledge Managers for Target System (and Components)

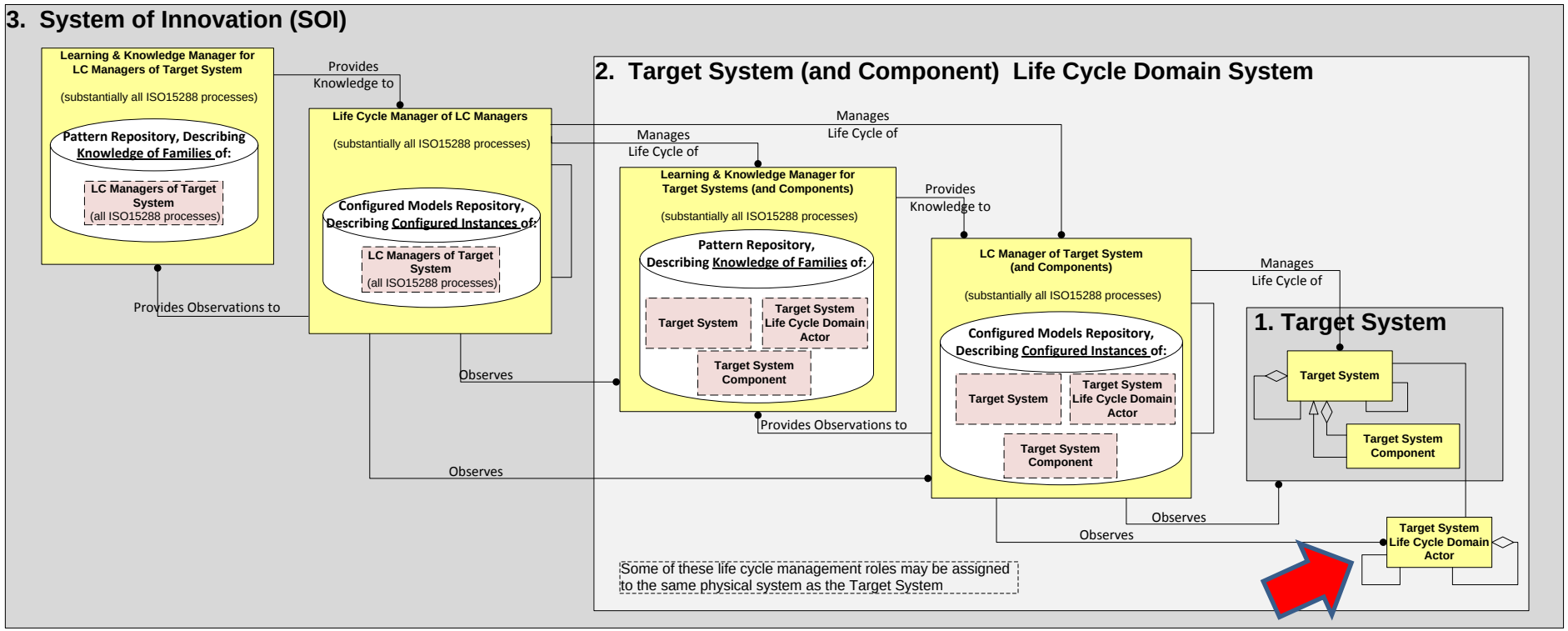


schindel@ictt.com 812-232-2062, attributed copies permitted

The Agile System Domain Model

The System 2 model recognizes three sub-systems besides the Target System:

- Target System Life Cycle Domain Actors: All actors with which the Target System will directly interact during its life cycle—those in its operational domain as well as all other direct actors.
- The next system is a special case of those actors . . .

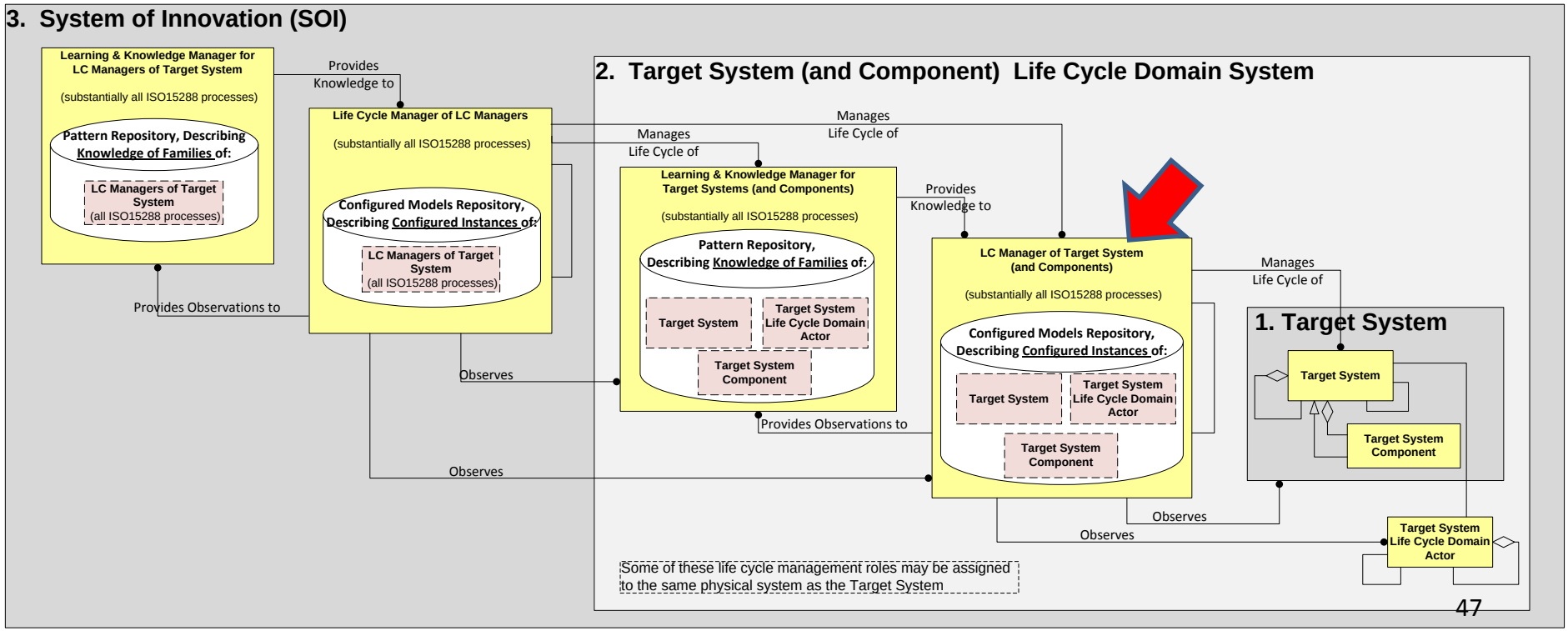


schindel@ictt.com 812-232-2062, attributed copies permitted

The Agile System Domain Model

The System 2 model recognizes three systems besides the Target System:

- LC Manager of Target System: Manages all life cycle aspects of the Target System, as recognized by ISO 15288. Note that this is more than just development or systems engineering—it includes manufacturing or acquisition, operations, maintenance, configuration management, and all the ISO System Management Functional Areas.
  - However, it includes only "already known" aspects of System 1 and Domain Actors—it does not include responsibility of learning new things about them . . .

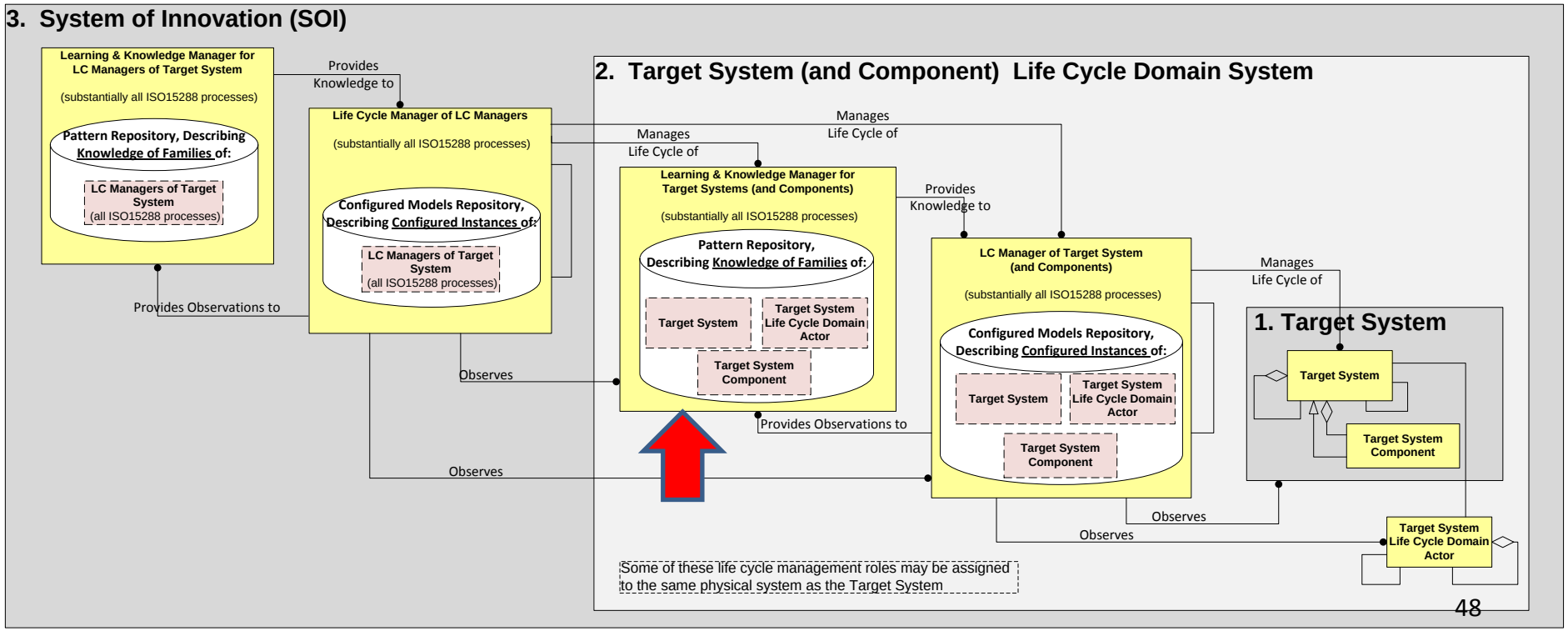


schindel@ictt.com 812-232-2062, attributed copies permitted

The Agile System Domain Model

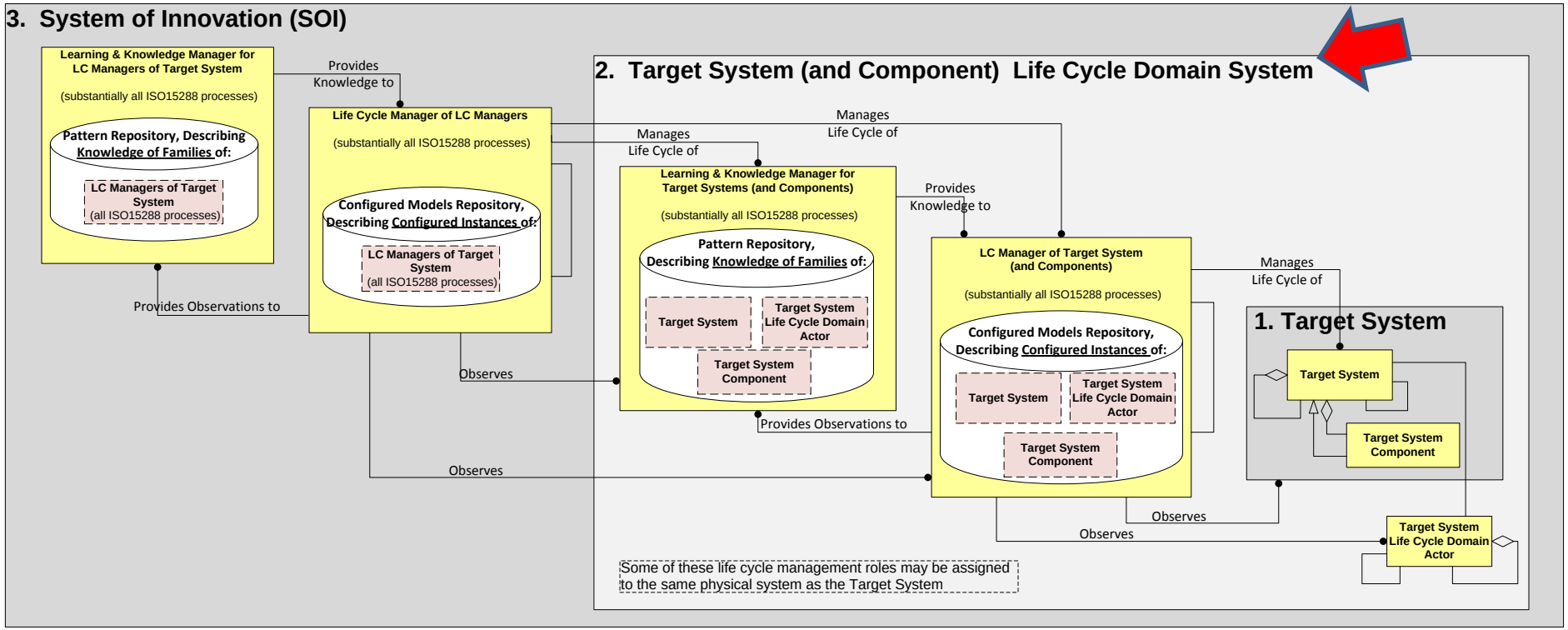
The System 2 model recognizes three systems besides the Target System:

- Learning & Knowledge Manager for Target System (and Components): Responsible for learning new things about the Target System, its Components, and its Environment. This may include extraction of patterns or other knowledge from observations, planning experiments and extracting conclusions from their results, and other forms of learning. It also includes responsibility for accumulation and persistent memory of those learnings, and for providing the resulting knowledge for use by the LC Managers of the Target System.



schindel@ictt.com 812-232-2062, attributed copies permitted

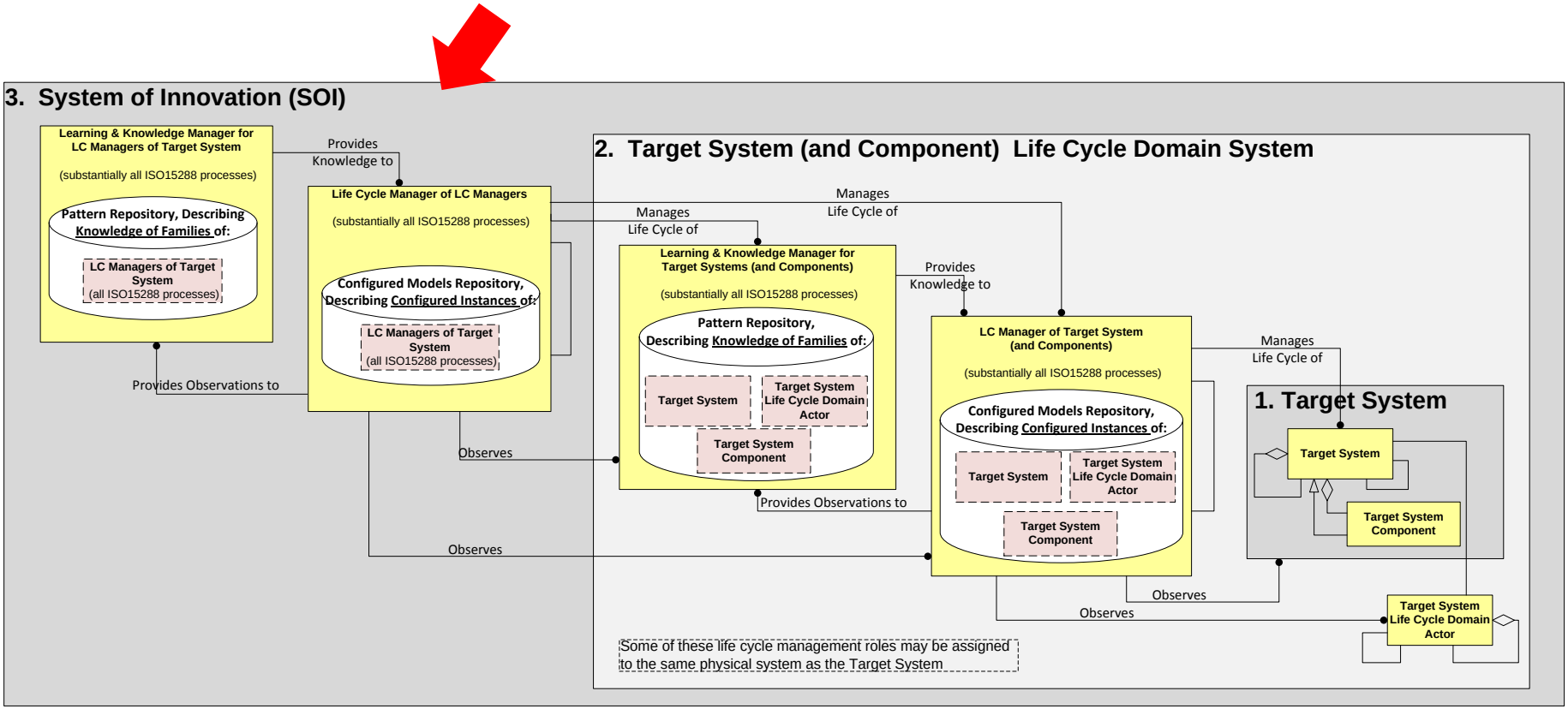
- Again, remember that these are logical (behavioral) roles. In realized physical systems, a single physical system may behave as both a Target System and a system that produces, modifies, reconfigures, or otherwise manages a Target System, by having roles from each allocated to it.
- For purposes of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



schindel@ictt.com 812-232-2062, attributed copies permitted

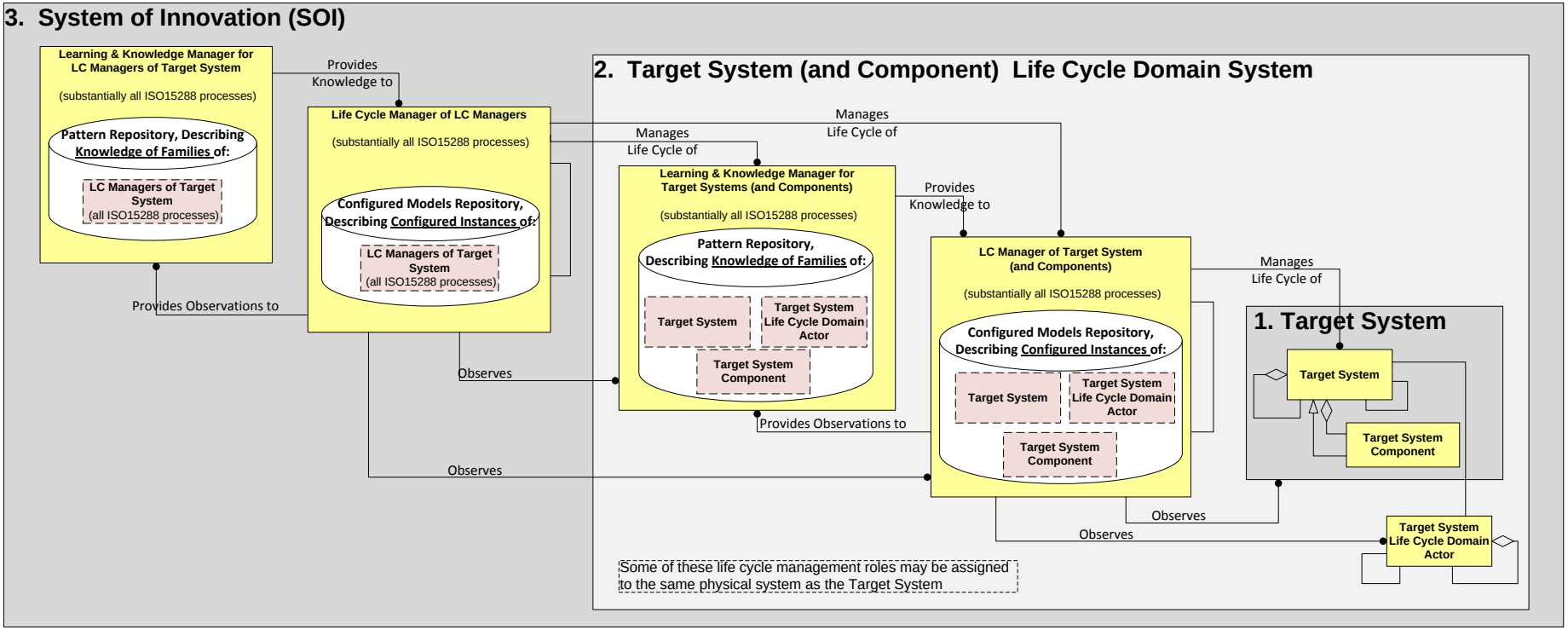
System 3: The **System of Innovation: (Definition)** The logical system responsible for managing the life cycles of instances of any (System 2) Target System LC Manager.

- (Recall that those System 2 Target System LC Managers include Target System development, production, integration, maintenance, operations, and other management systems.)



schindel@ictt.com 812-232-2062, attributed copies permitted

- Summary so far:
  - System 2, the Target System Life Cycle Domain System produces and modifies instances of System 1, the Target Systems (and Components), and also learns new things about System 1 and its environment.
  - System 3, the System of Innovation, produces and modifies instances of System 2, the Target LC Managers, and also learns new System 2 things

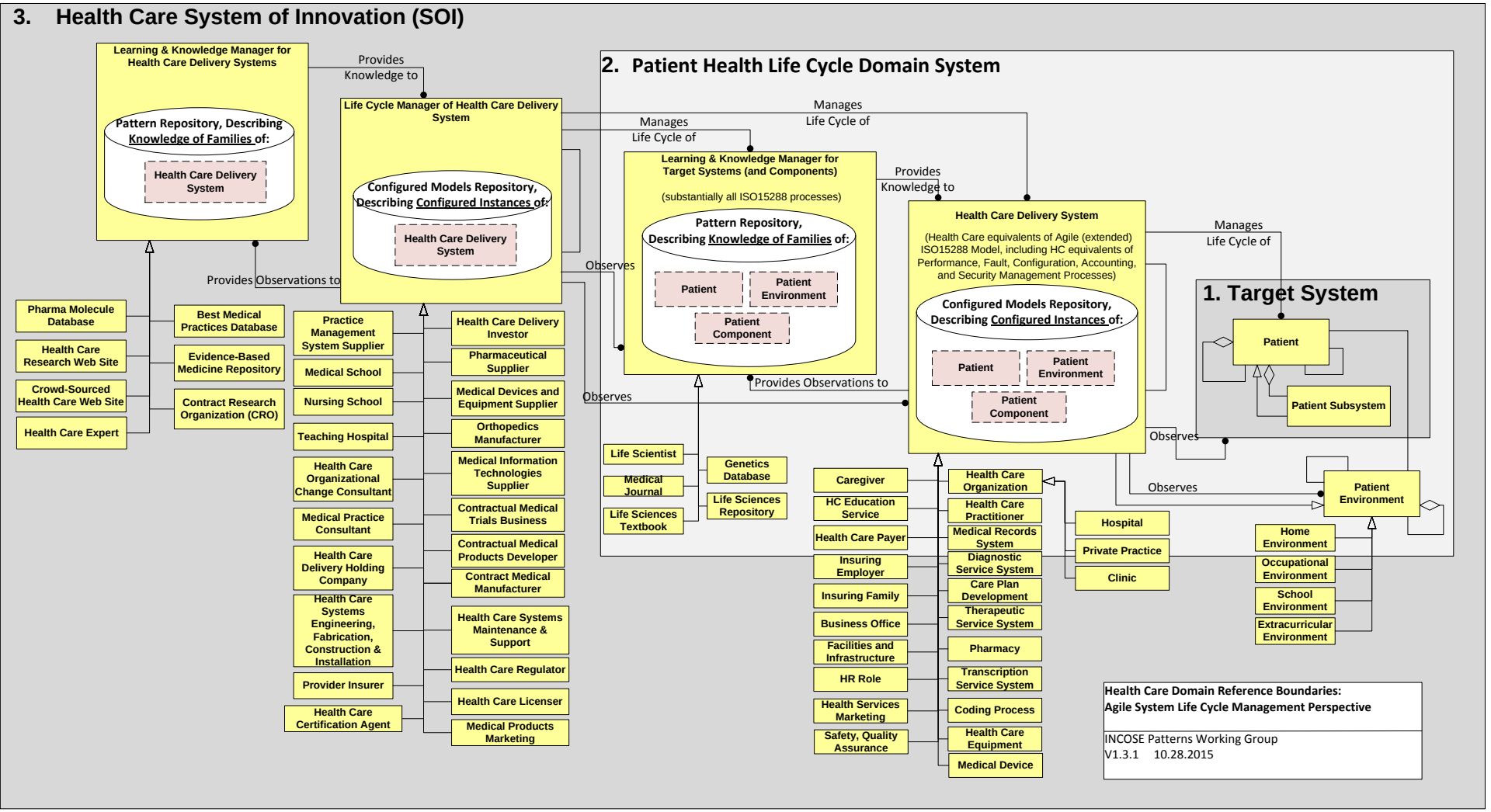


schindel@ictt.com 812-232-2062, attributed copies permitted

Why are the learning capabilities of System 2 and System 3 differentiated from other capabilities in System 2 & 3 models?

- Especially for Agile reasons:
  - We want to understand what capabilities can exist for "agile movement within what is already known", for both System 2 and System 3, in nearly all of the ISO 15288 process areas, and . . .
  - We also want to explicitly understand what is meant by "learning" in nearly all of the ISO 15288 process areas.
- Note that learning includes ideas such as experimental discovery, questioning, observation, noticing differences from what is known, etc.

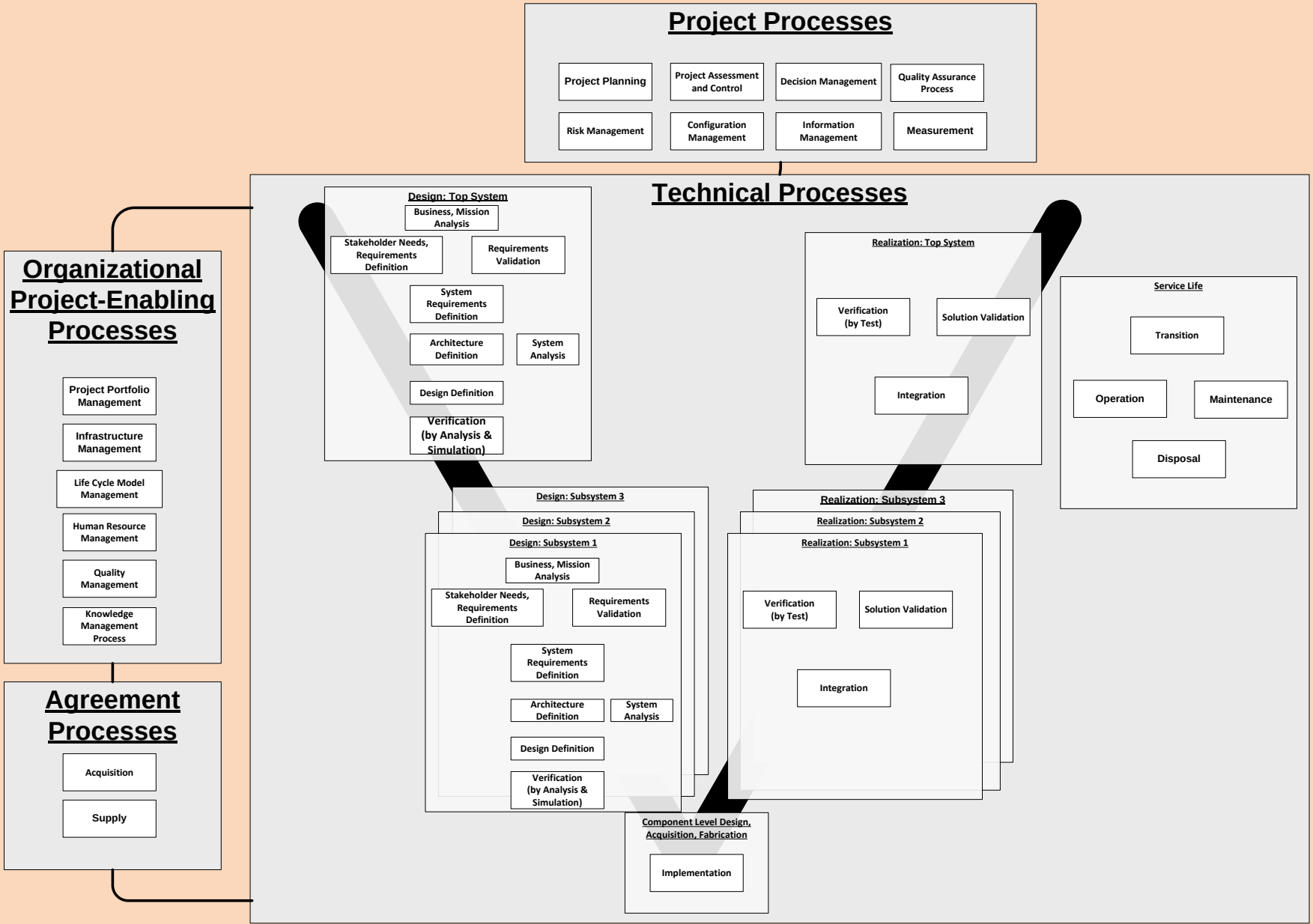
Example: Health care domain, top level



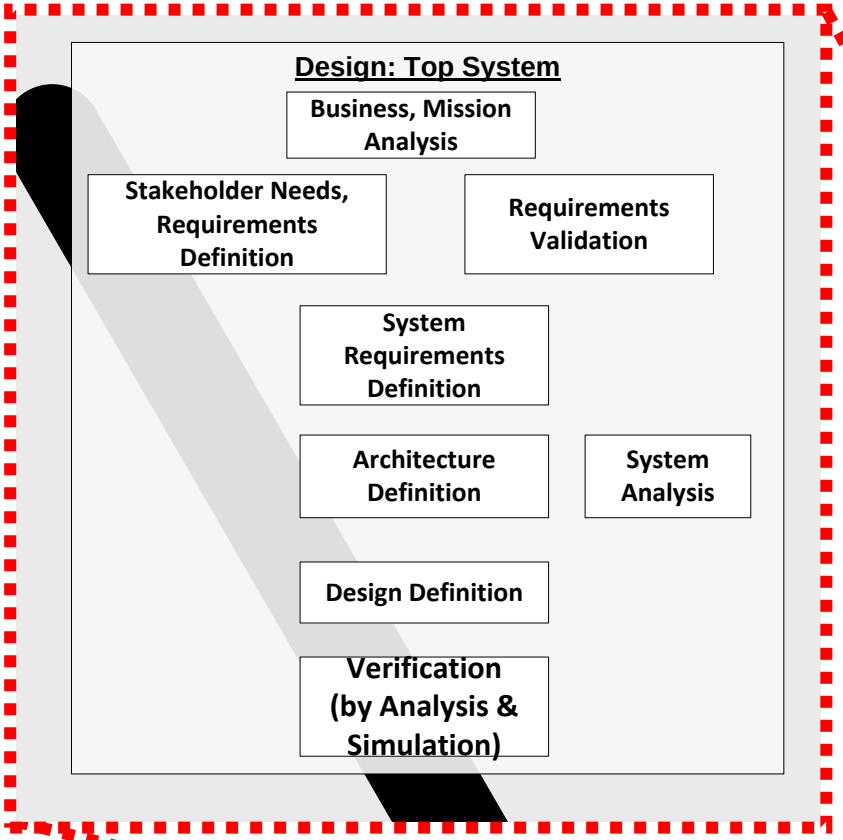
schindel@ictt.com 812-232-2062, attributed copies permitted

System of Innovation (SOI) Pattern Logical Architecture

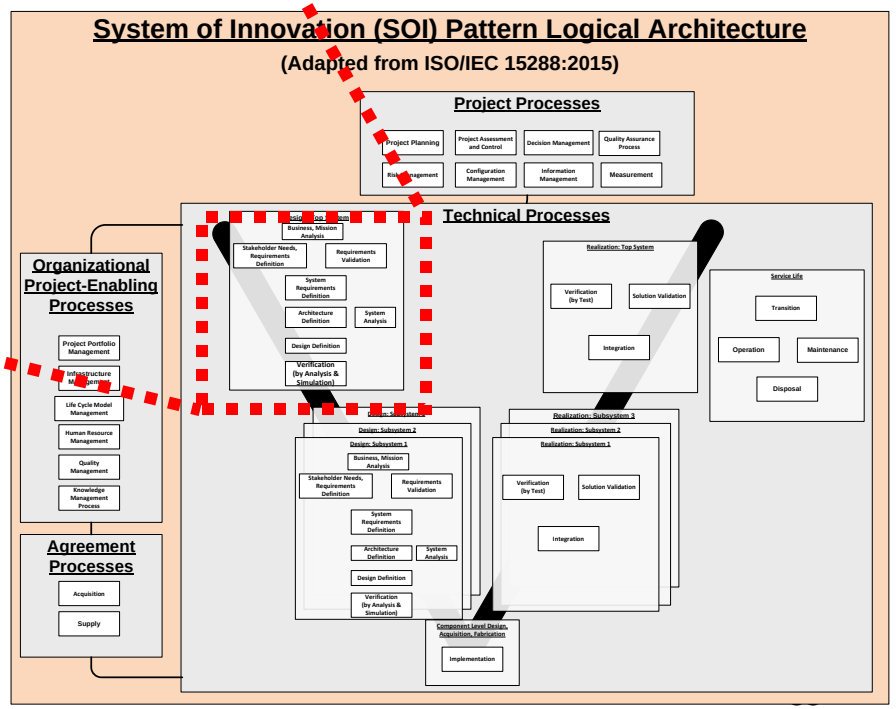
(Adapted from ISO/IEC 15288:2015)



schindel@ictt.com 812-232-2062, attributed copies permitted



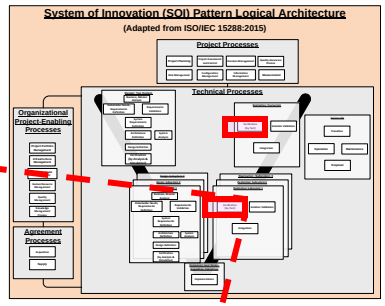
The ISO 15288 System Life Cycle Management framework outlines much of what the LC Management Systems in the ASELCM Pattern are supposed to do—without demanding any particular sequence, while still indicating information inter-dependencies.

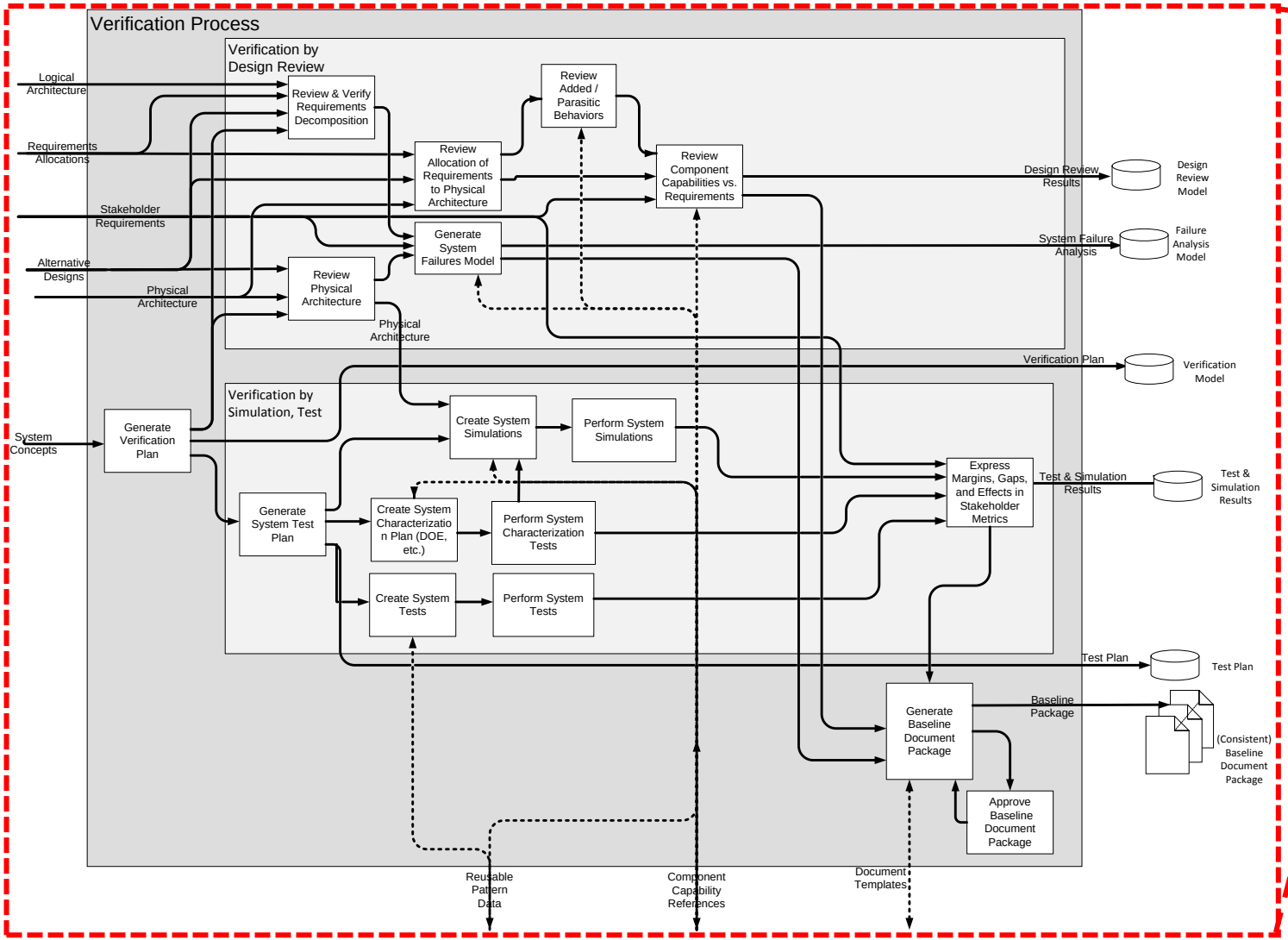


schindel@ictt.com 812-232-2062, attributed copies permitted

SOI Pattern includes a model for each of the ISO15288 Processes

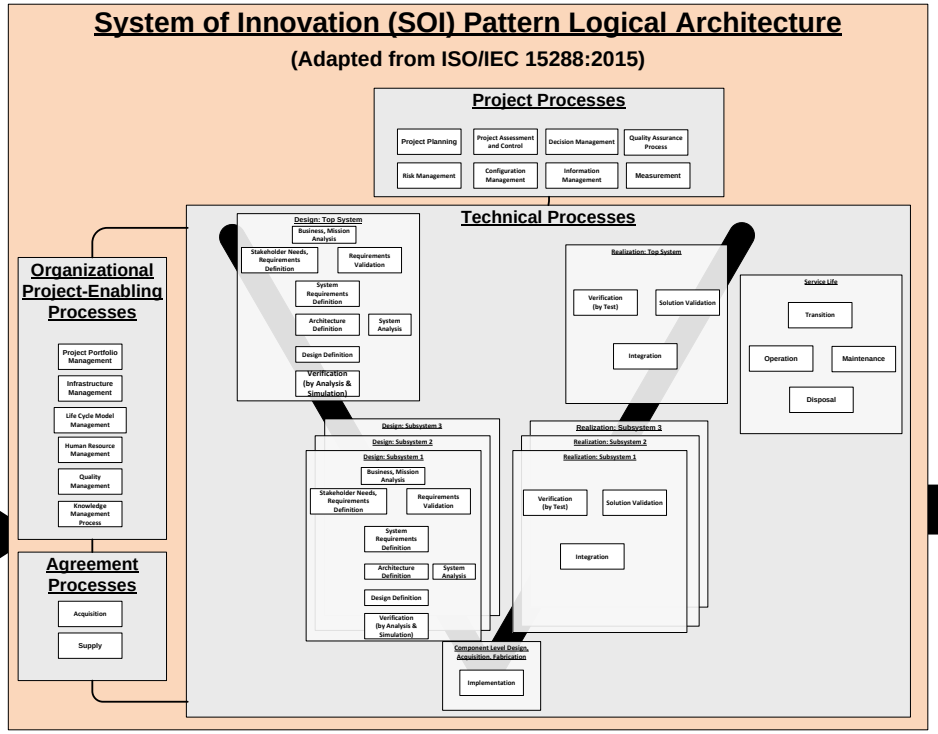
- Example: Verification Process
- Each also includes options for MBSE and PBSE methods

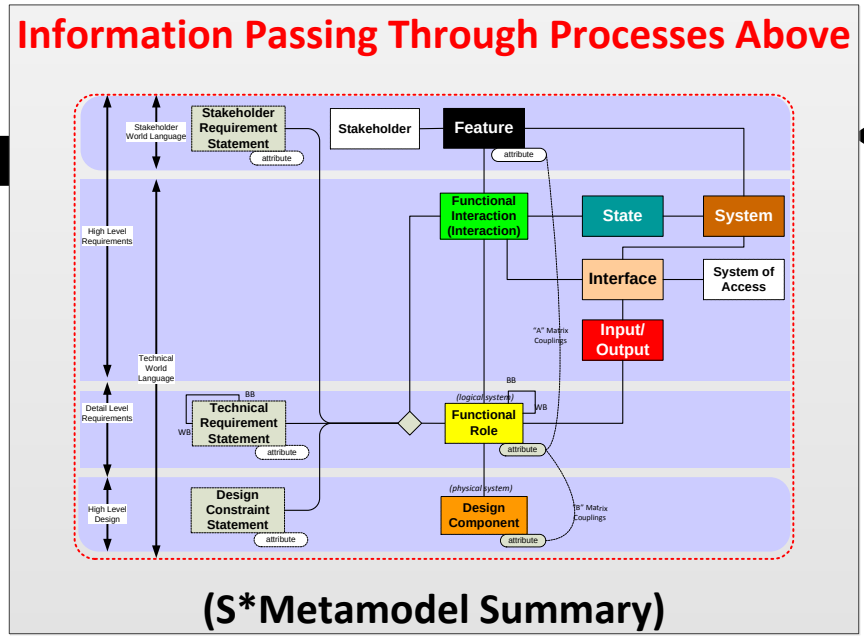




Process vs. Information:

- The S*Metamodel describes the MBSE information that passes through life cycle processes, as S*Models.
- Using PBSE accelerates this process, by basing S*Model information on knowledge-managed S*Patterns.
- None of this requires any specific sequence or order of processes, which may be concurrent or otherwise, depending on strategy.
- What is the “agile trajectory” through S*Space?
- Agile strategy typically to advance short distances in S*Space, with limited time, resource, and risk budgets.
- How are the “trajectory deltas” planned for each incremental advance?

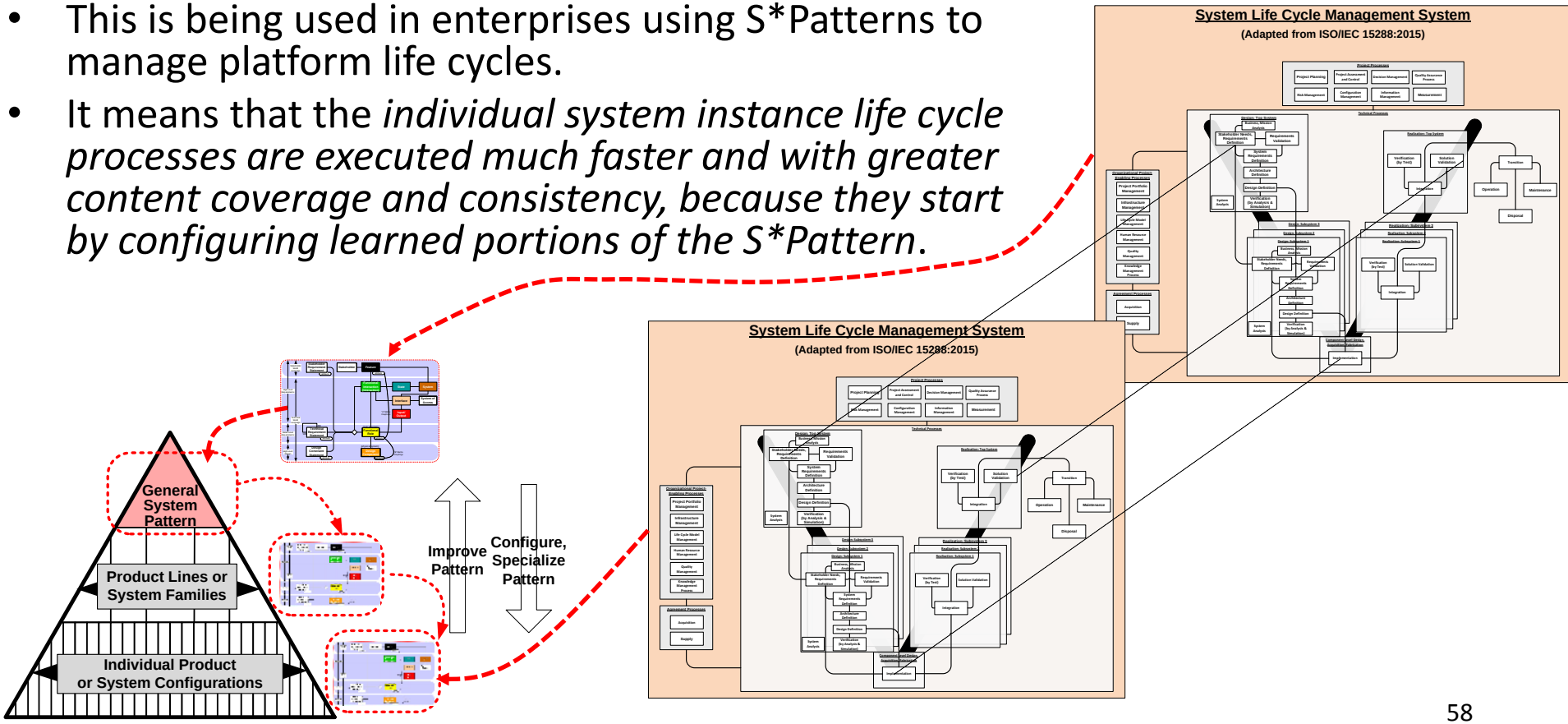




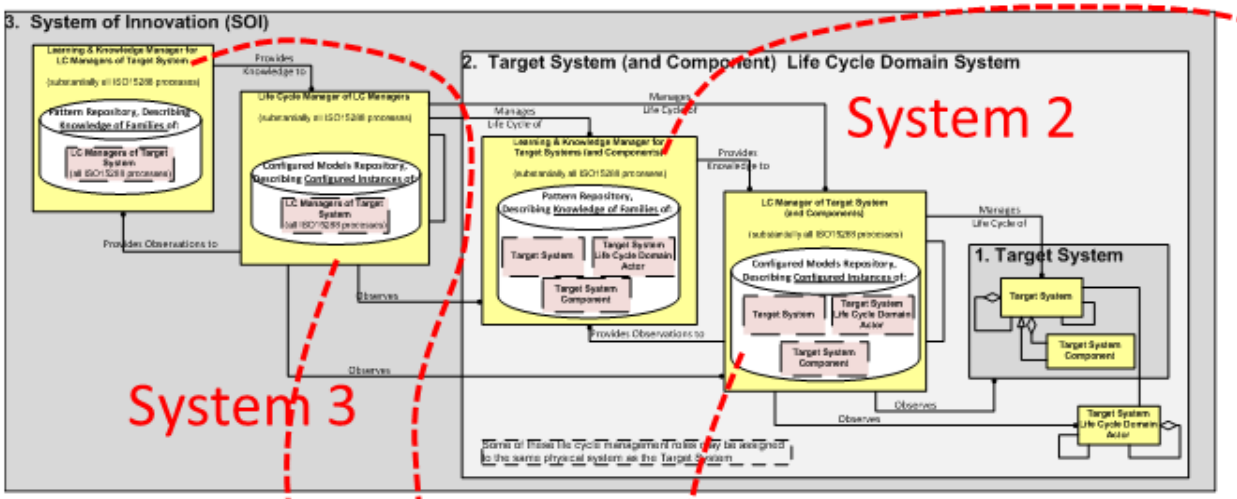
schindel@ictt.com 812-232-2062, attributed copies permitted

Managing Pattern Life Cycles

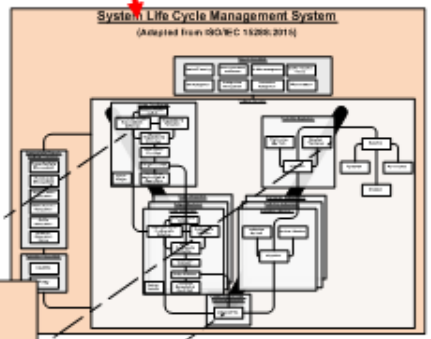
- In effect, a System Pattern is subject to most of the same Life Cycle Processes.
- ISO 15288 KM represents the notion of managing that knowledge, but in PBSE this can be viewed as "unfolding" the KM process to find another copy of the LC management processes, operating on the pattern.
- This is being used in enterprises using S*Patterns to manage platform life cycles.
- It means that the *individual system instance life cycle processes are executed much faster and with greater content coverage and consistency, because they start by configuring learned portions of the S*Pattern*.



schindel@ictt.com 812-232-2062, attributed copies permitted

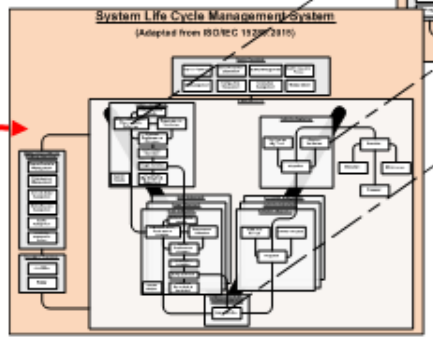


Performs most of these ISO processes, to manage what is being learned about S1 space.

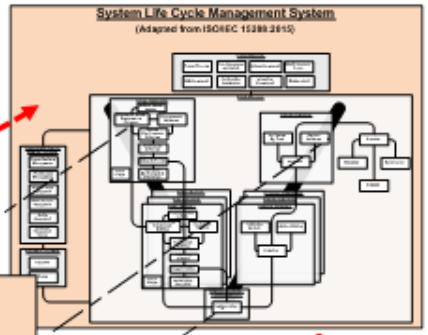


Manages Target System Family Patterns

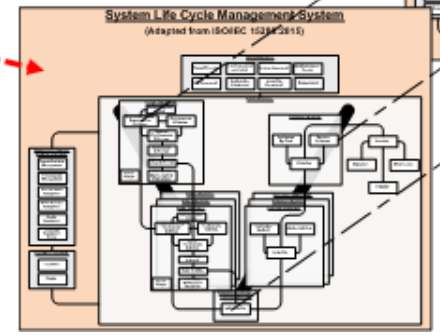
Performs most of these ISO processes, to manage S1 instances , using S1 patterns



Manages LC Management System Family Patterns



Performs most of these ISO processes, to manage S2 instances , using S2 patterns



Performs most of these ISO processes, to manage what is being learned about S2 space.

The ASELCM Project is about patterns of agility in both System 1 (Target System) and System 2 (Life Cycle Management)

schindel@ictt.com 812-232-2062, attributed copies permitted

General ASELCM Pattern: Stakeholder Features Model

- Since the Stakeholder Features for a system are used to configure that system's pattern for a specific case, we can expect to add System 2 features during this project:
  - For selecting an Agile methodology type (or non Agile approach), depending on external environment, internal capabilities, and other Agile Systems parameters identified by Rick Dove and colleagues in Agile SE literature
- This project also brings increased emphasis to earlier System of Innovation interest in System 2 features that help us know:
  - Where are we in S1 space? In S2 space?
  - Where are we going in S1? In S2?
  - What do we "optimally" adjust / vector, in S1? In S2?, in light of risk and other parameters.

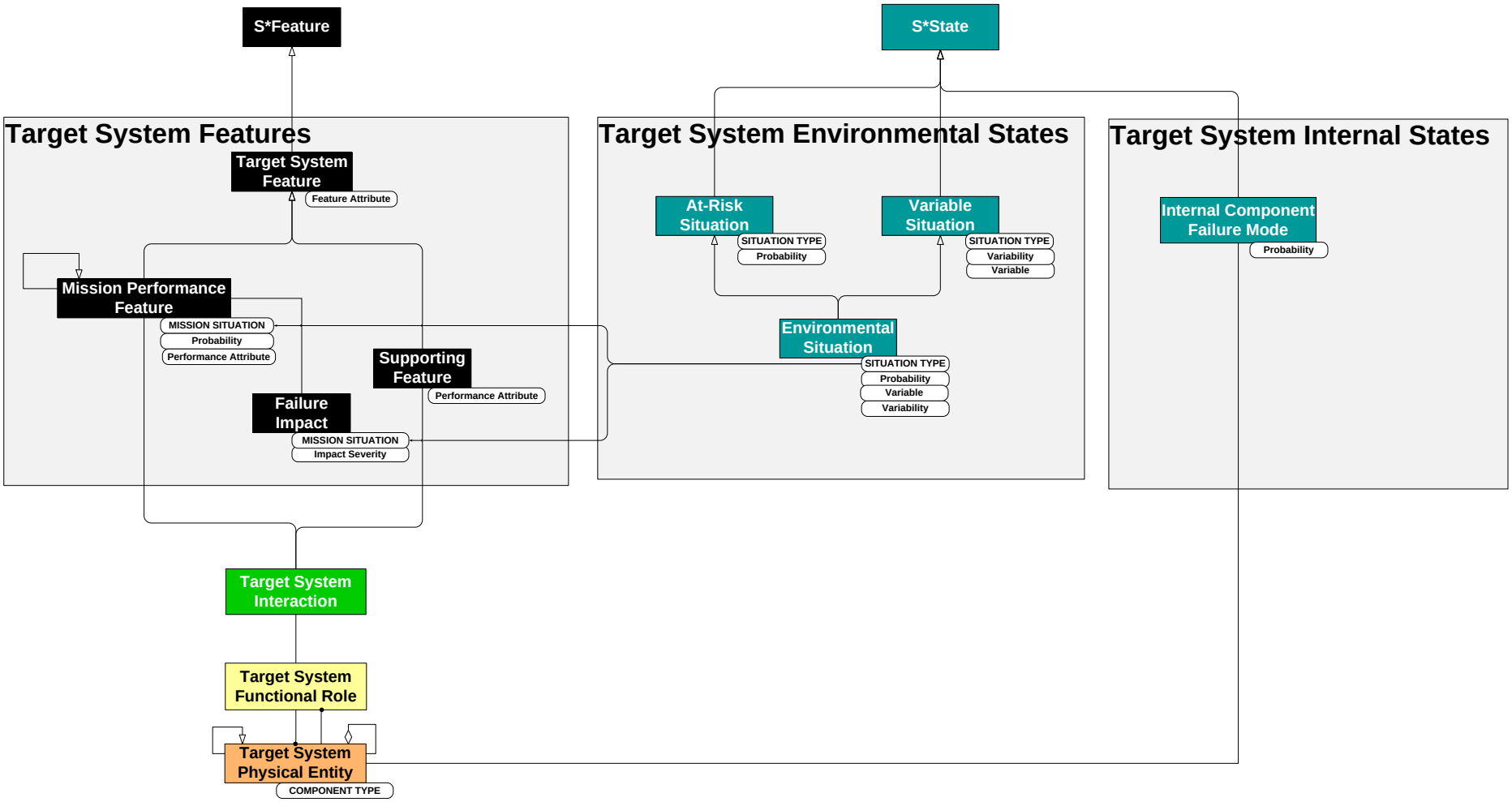
Agile System Pattern: Major Feature Groups

- **System 1 Features**: Stakeholder capabilities of the Target System—the system we ultimately want to respond (with help from Systems 2 and 3) in agile fashion:
  - Example: Autonomous Vehicle Navigation Feature

- **System 2 Features**: Stakeholder capabilities of the Target System Life Cycle Management System. This includes all aspects of its LC, a subset of which are relevant to the Agile Systems LC Pattern.
  - Example: Vehicle Environmental Reconfigurability Feature

- **System 3 Features**: Stakeholder capabilities of the three subsystems of System 3—concerned with observing and learning about the Target System and its Environment, and about the Target System LC Manager; also responsible for managing the LC of the Target System LC Manager.
  - Example: Vehicle Route Learning Feature

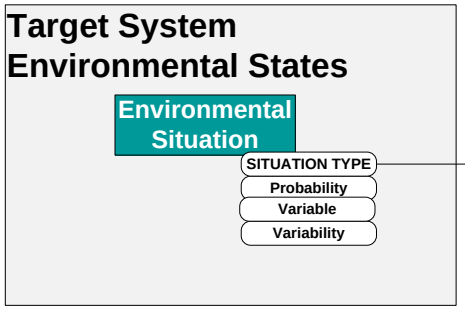
Overview of System 1 Features (and some related S*Metaclasses)

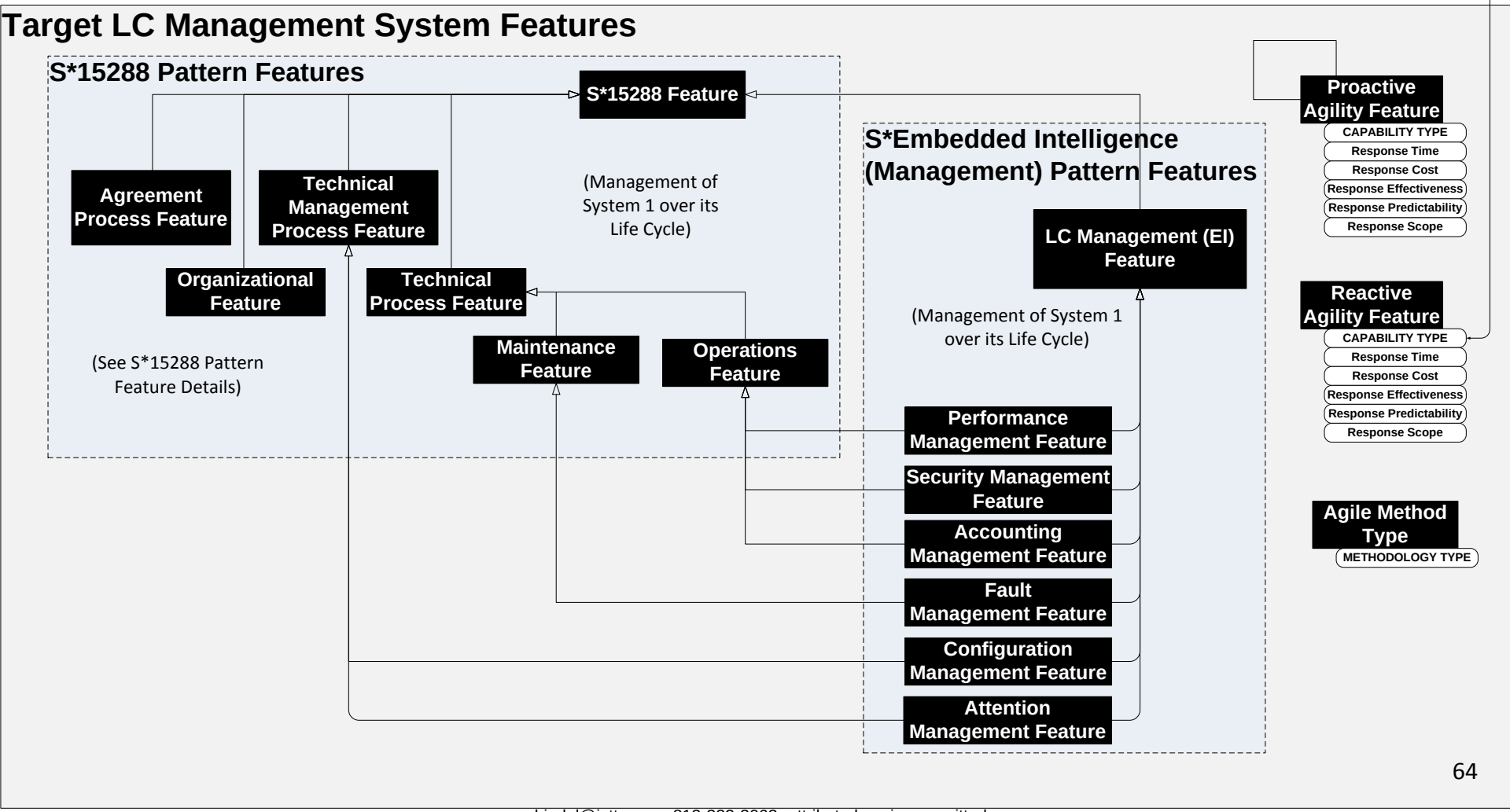


Subset of Features: Life Cycle Management Features

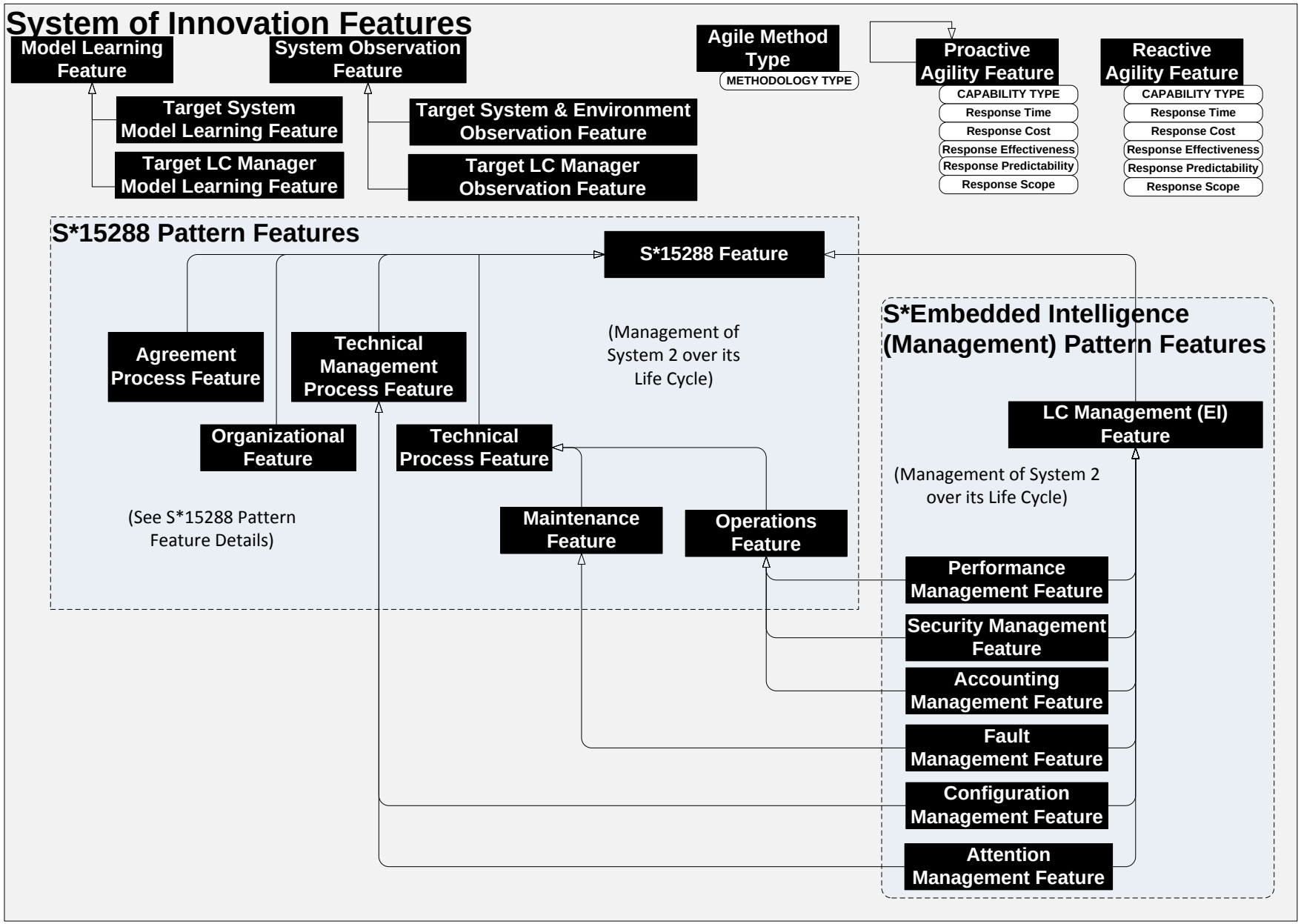
- These represent the stakeholder level view of capabilities to manage life cycles of either System 1 or System 2.
- We are making use of the S*15288 Pattern, which models ISO 15288 process capabilities:
  - This pattern is a specialization of the more abstract features of the generic Systems of Innovation (SOI) Pattern (Beihoff and Schindel, 2012)

Overview of System 2 Features



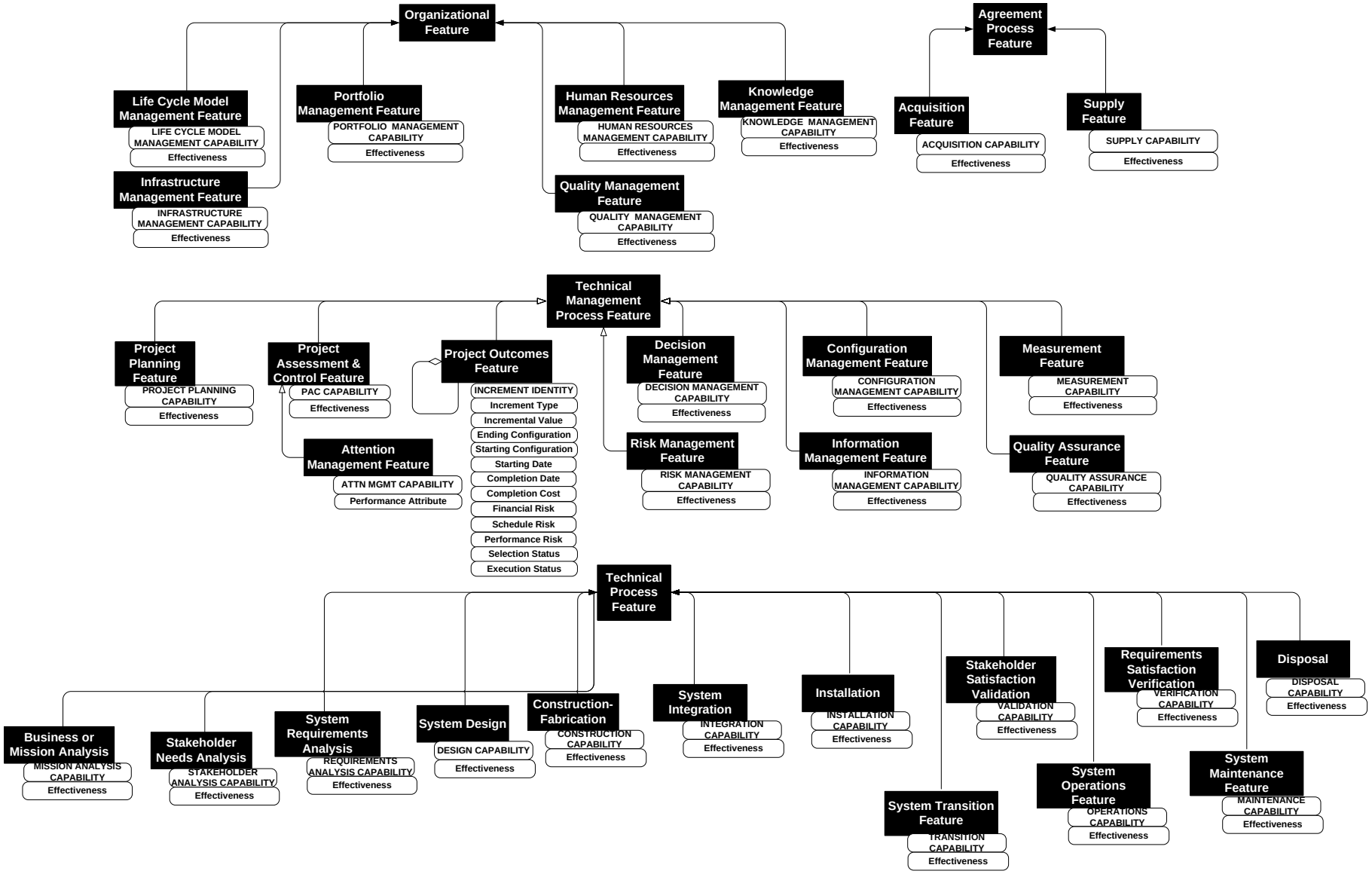


Overview of System 3 Features



schindel@ictt.com 812-232-2062, attributed copies permitted

Overview of S*15288 Pattern Features



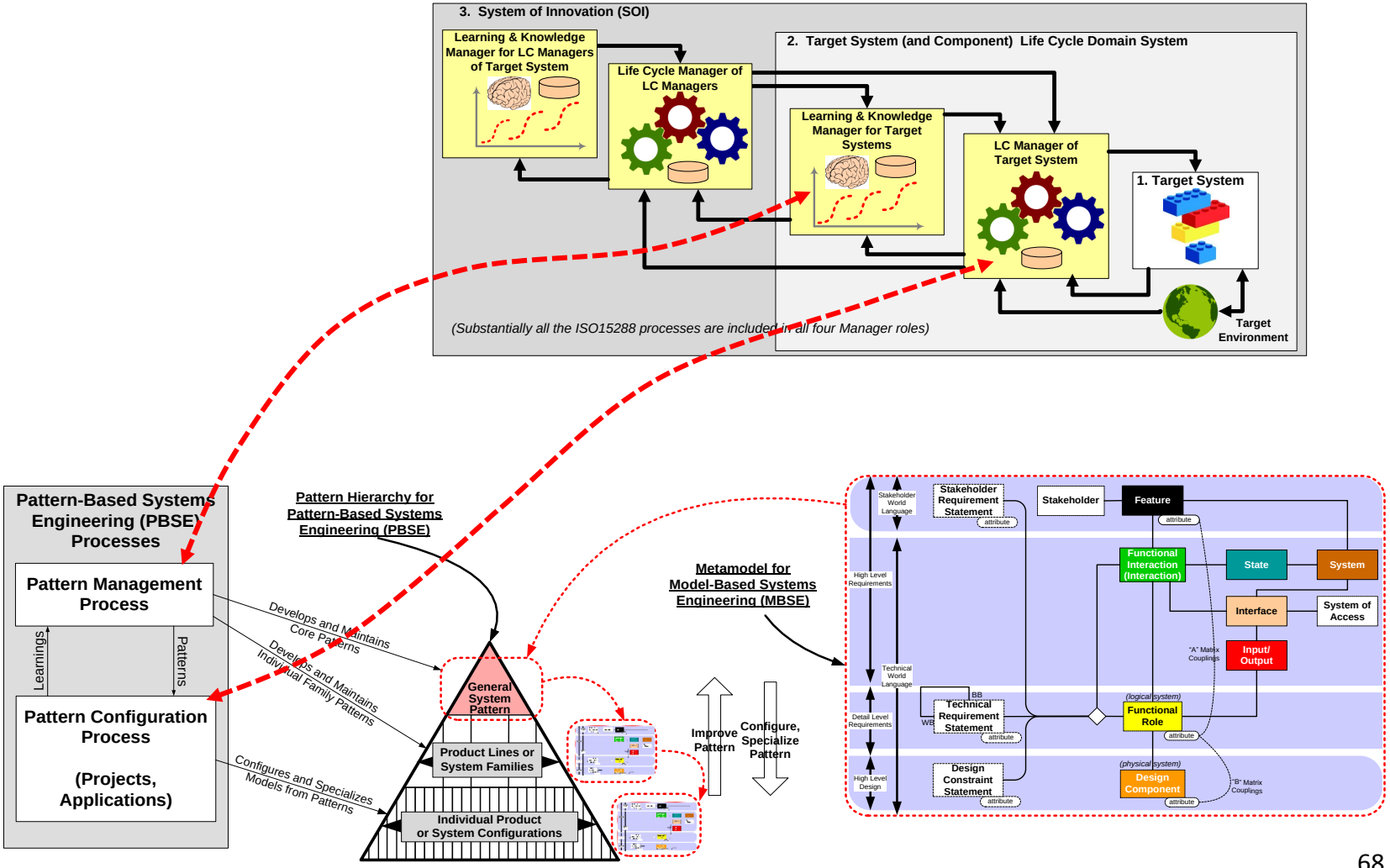
schindel@ictt.com 812-232-2062, attributed copies permitted

System Patterns Answer a Key Challenge to Agile Methods Adopters

- Another hallmark of agile methods is the repeated iterative release of a "complete enough" deliverables for some use to be made of them by the customer.
- For those adopting agile methods, this can raise a key question / challenge:
  - How to produce a complete enough deliverable in each (time limited) sprint, for a complex system?
- Answer: Configured Patterns as draft deliverables—S*Patterns may be very quickly configured.

PBSE is agile MBSE:

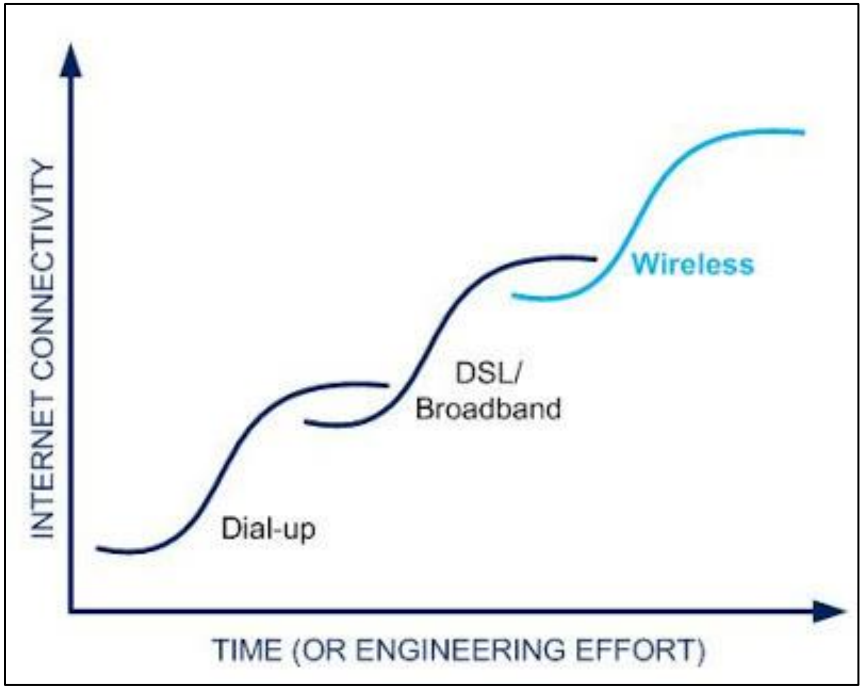
– A very small number of people can make a very large number of System 2 people much more agile: Learn the S1 *Model*, not S1 Model*ing*.

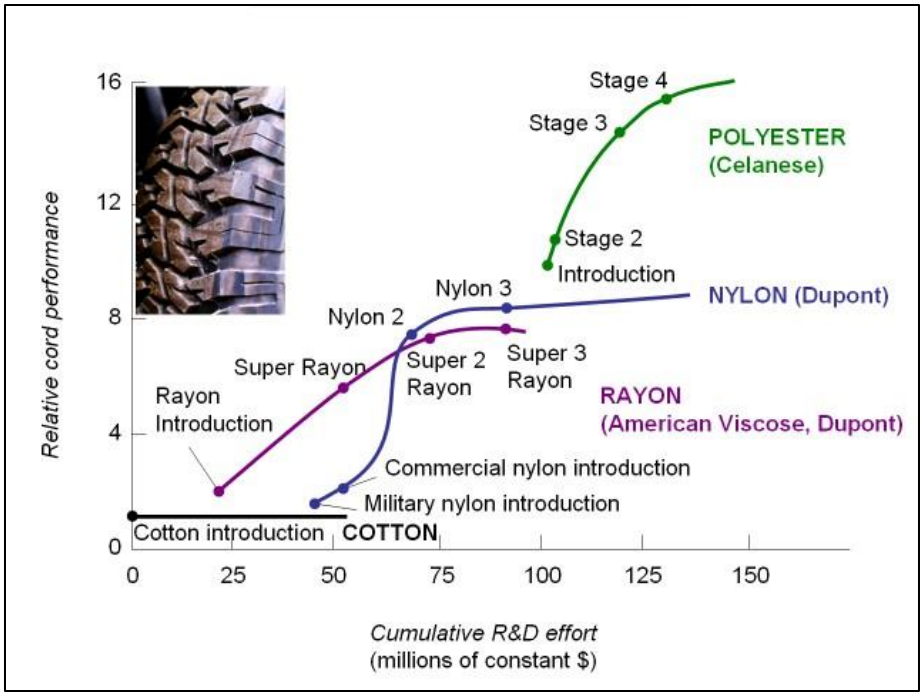


Pattern Class Hierarchy

schindel@ictt.com 812-232-2062, attributed copies permitted

Additional Trajectory-Based Concept: Learning Curves, “S” Curves





(insert references)

General ASELCM Pattern: States & Interactions Model

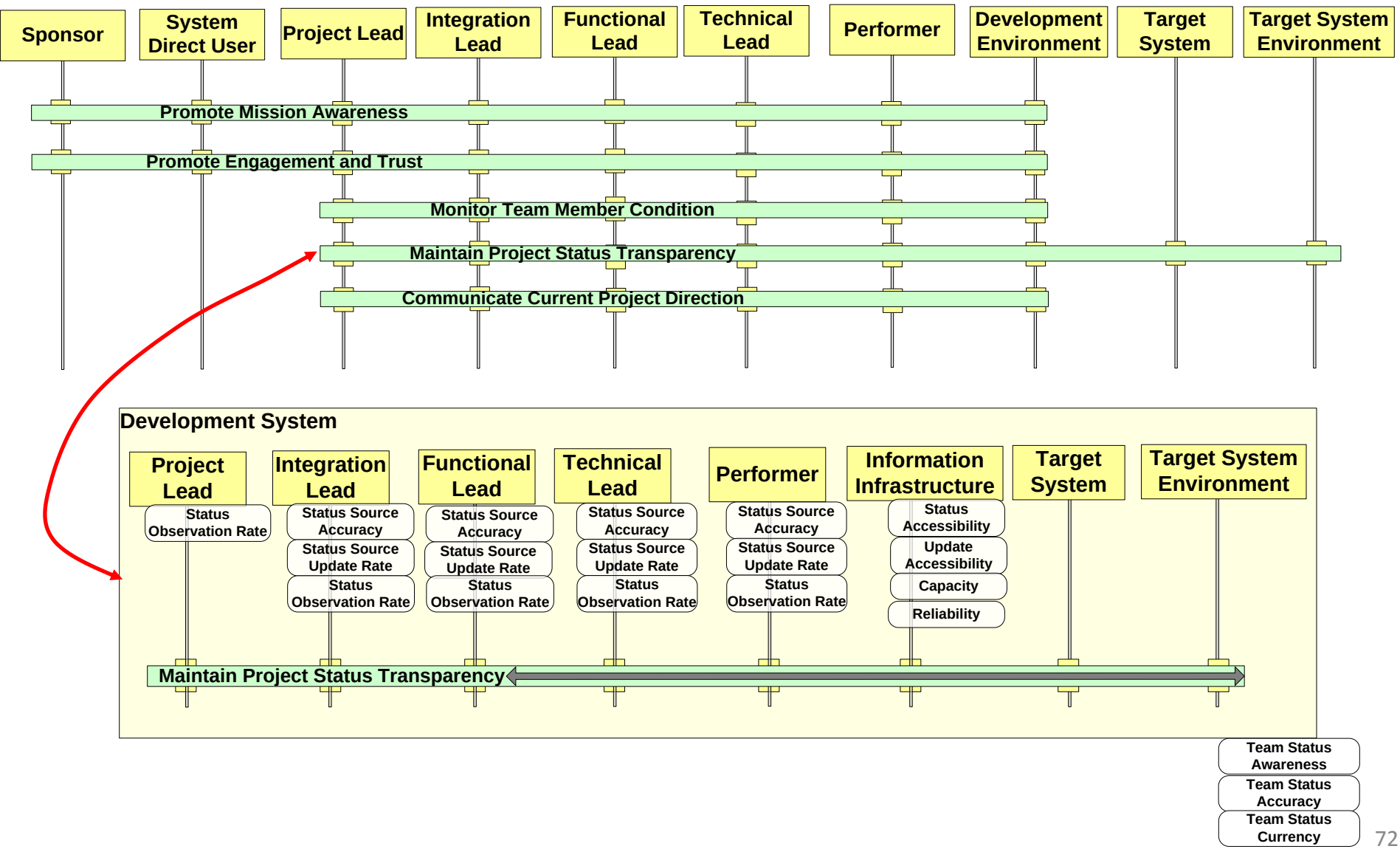
- The States & Interactions Model is currently being pursued in the Wave and Scrum specializations of the ASELCM Pattern:
  - See later sections below
  - After some basic questions here are settled, will insert a general one for the parent pattern

ASELCM Pattern Configurations: Initial workshops contributed to, or affirmed:

- **Host workshop 1:** System 1, 2, 3 reference boundaries; awareness and attention model (management and team)
- **Host workshop 2:** Explicit partner risk allocation and spreading; Scrum model
- **Host workshop 3:** Product Line Engineering (PLE / PBSE)
- **Host workshop 4:** ASE assimilation in a defense acquisition environment; Information Debt in addition to Technical Debt

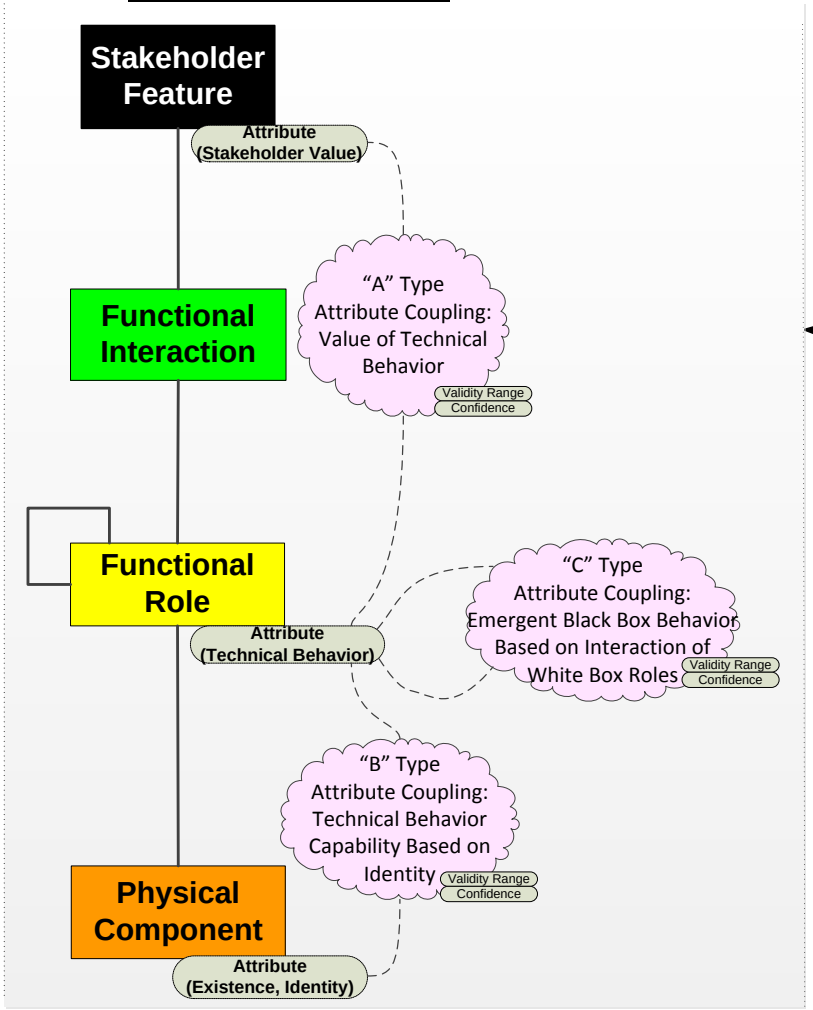
Site 1 / Wave configuration of ASELCM Pattern

Illustrates five interactions learned from Workshop 1 visit

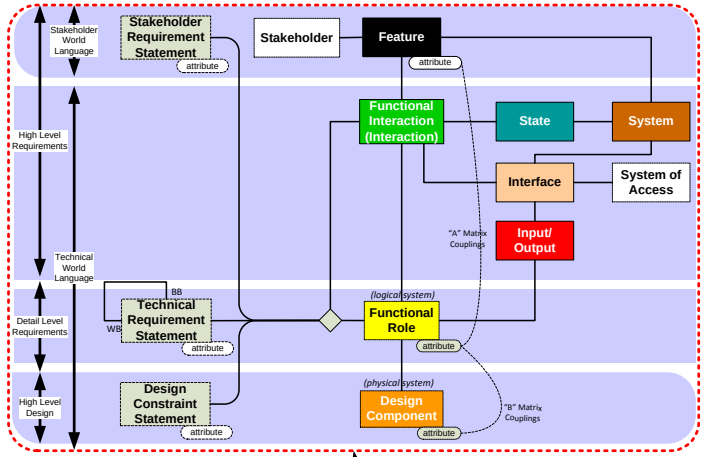


General ASELCM Pattern: Attribute Couplings Model

From S*Metamodel



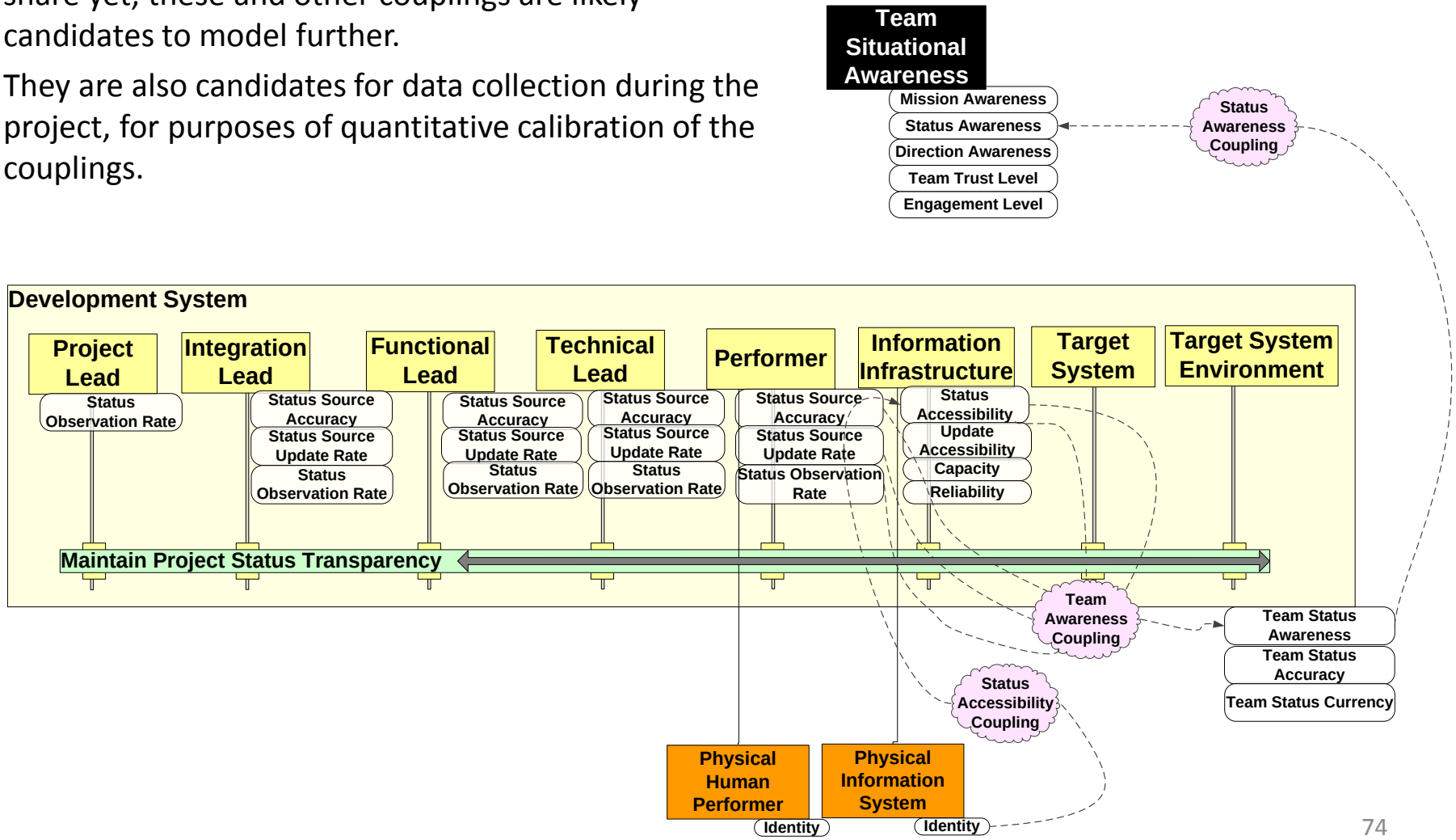
Subset of S*Metamodel at the focus of this use case



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

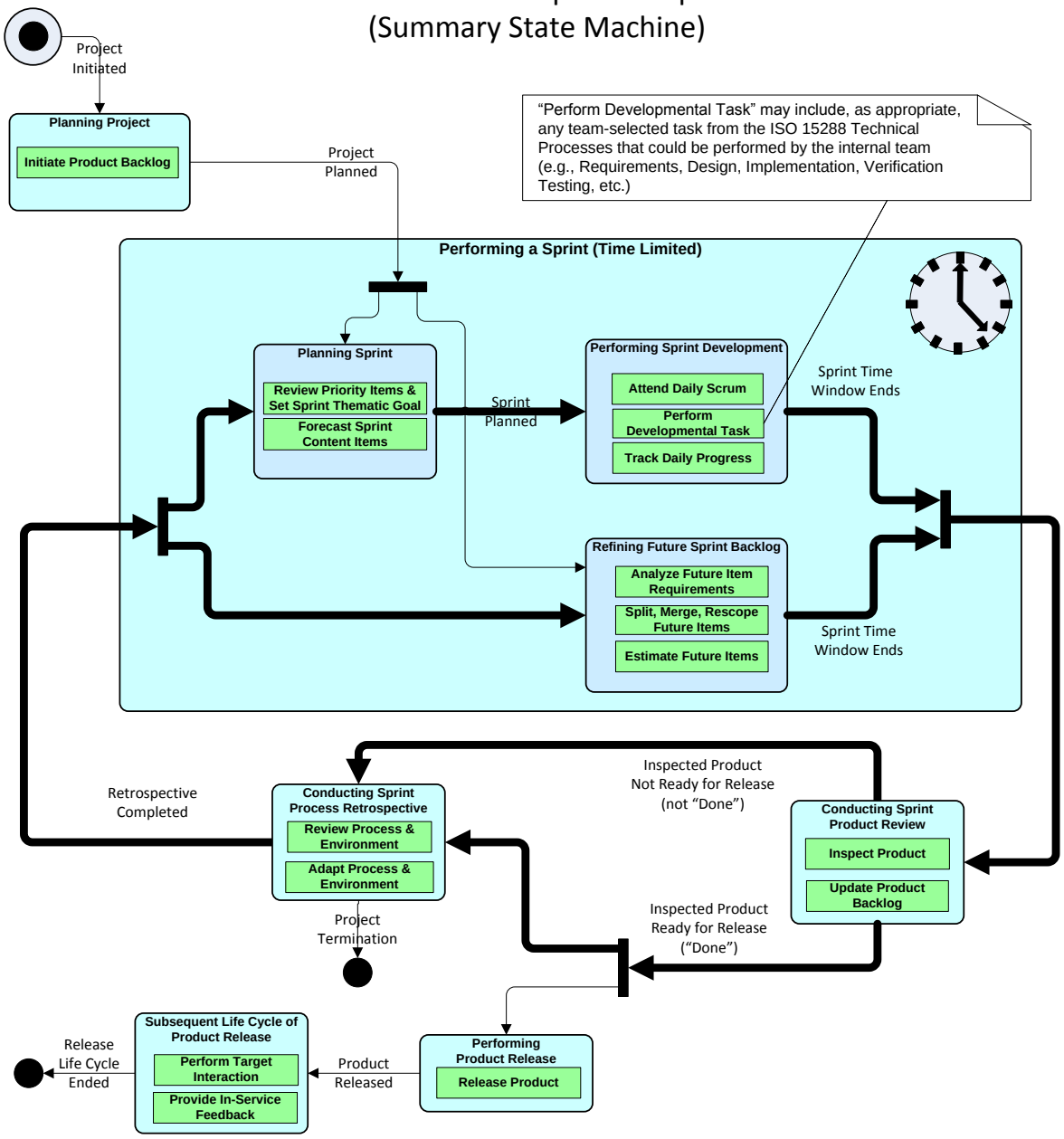
Site 1 / Wave configuration of ASELCM Pattern: Attribute Couplings Model

- Although we don't have quantitative values data to share yet, these and other couplings are likely candidates to model further.
- They are also candidates for data collection during the project, for purposes of quantitative calibration of the couplings.

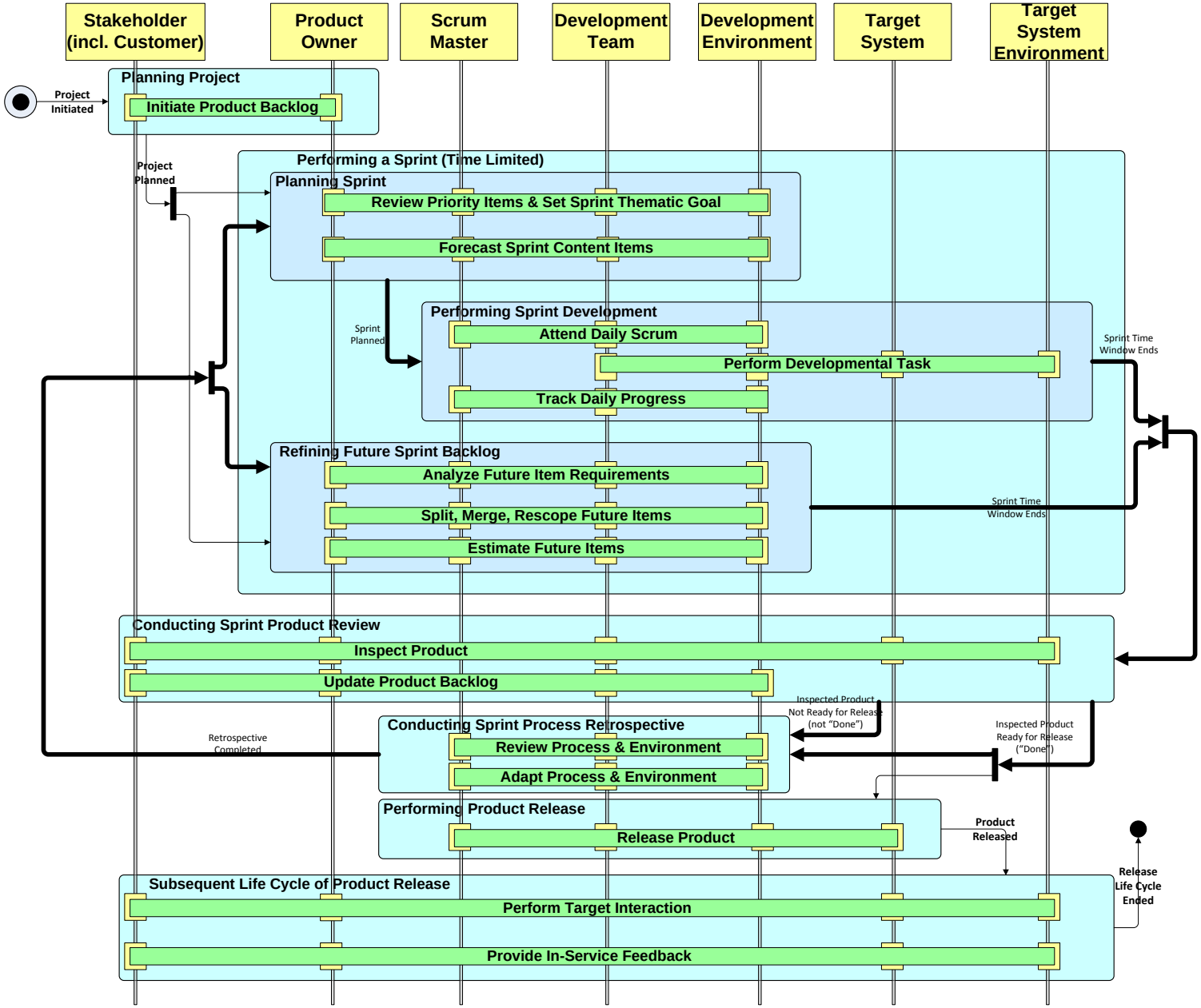


schindel@ictt.com 812-232-2062, attributed copies permitted

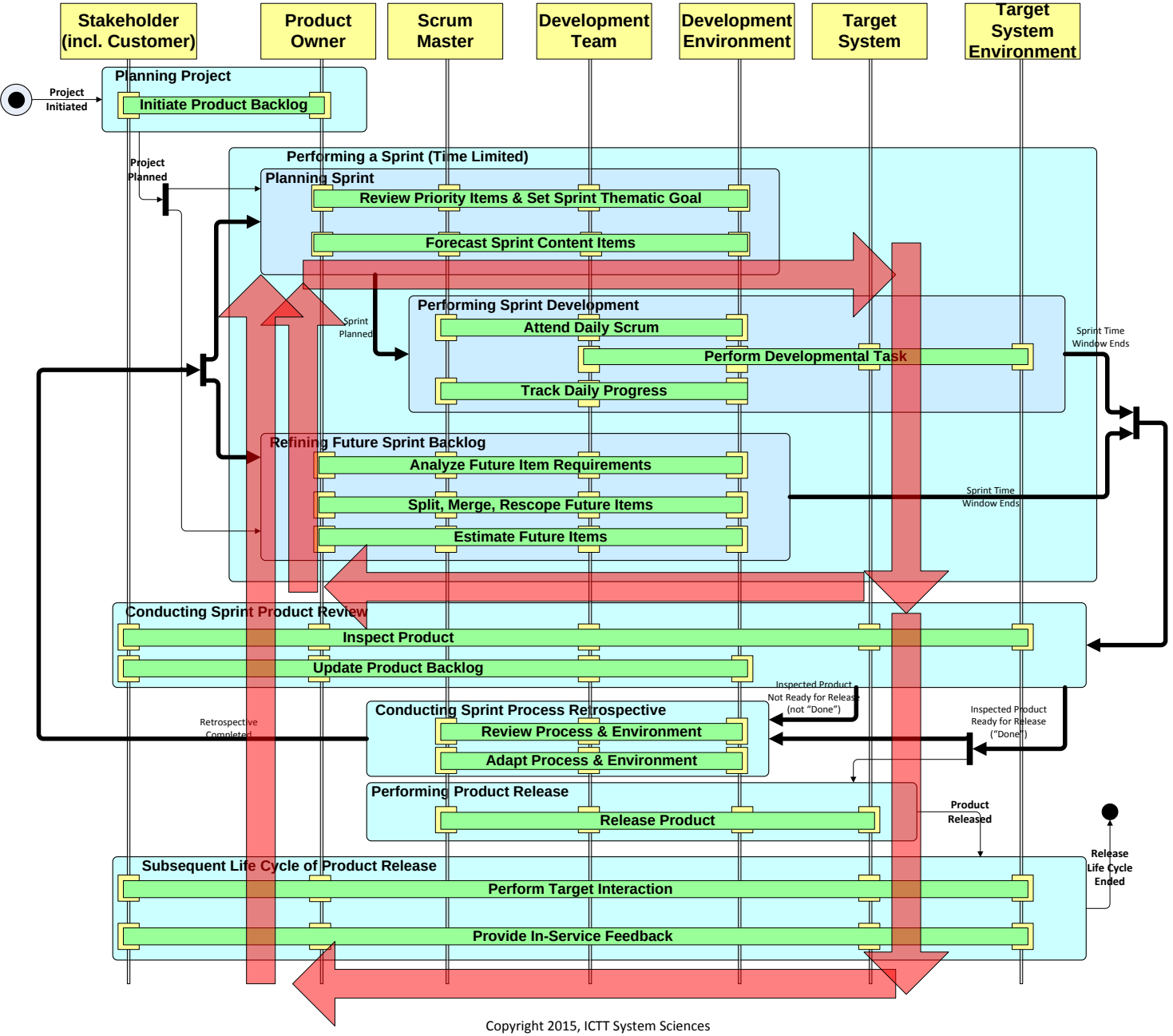
Traditional Scrum Sprint Perspective (Summary State Machine)



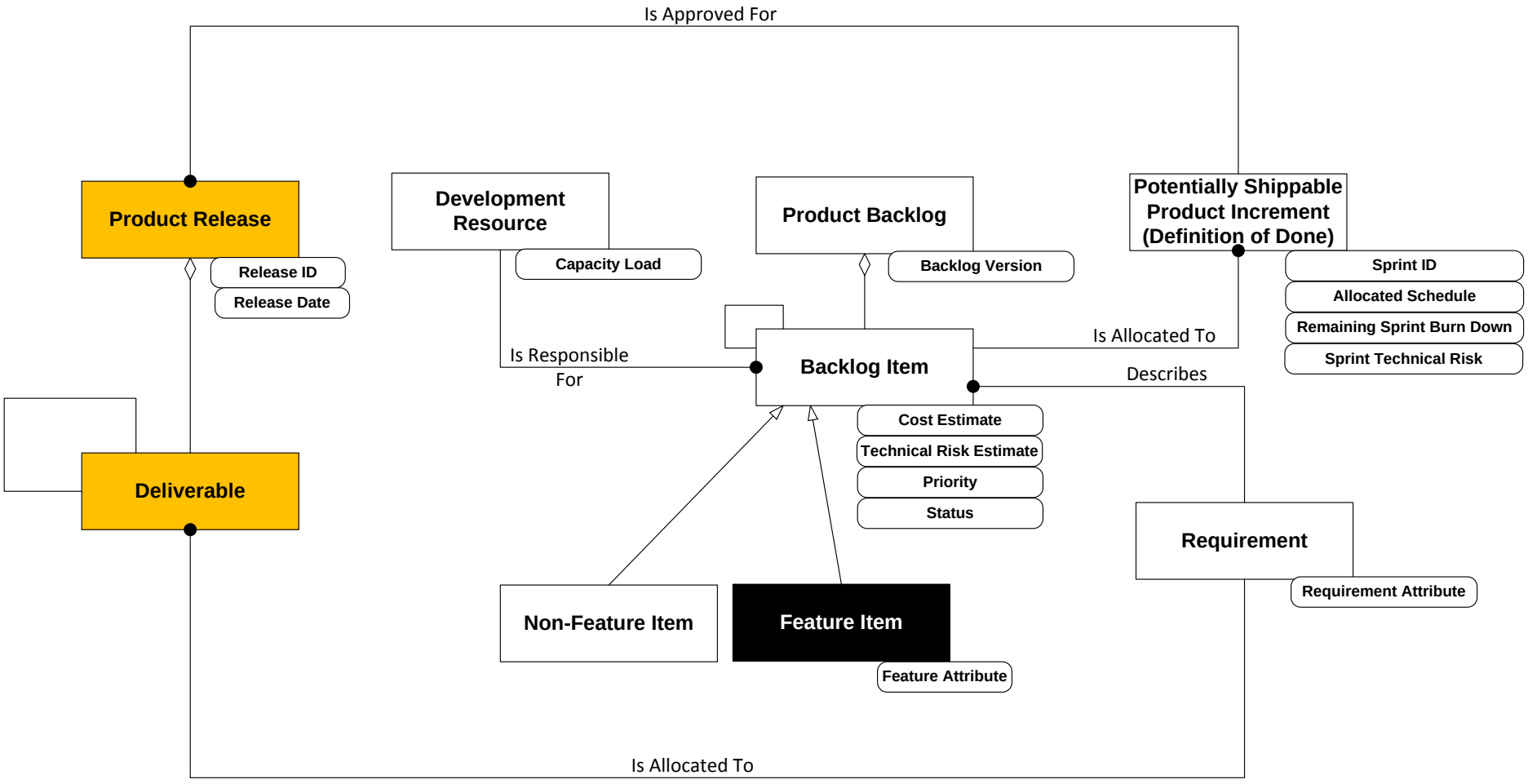
Traditional Scrum Sprint Perspective
(Activity Diagram, with Swim Lane Roles)



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



Traditional Scrum Sprint Perspective (Simplified Model of Managed Information)

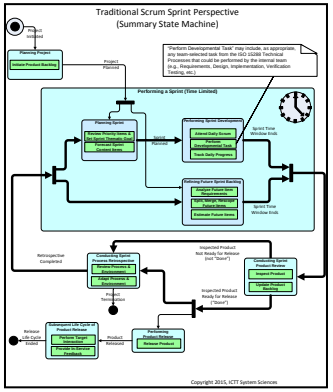


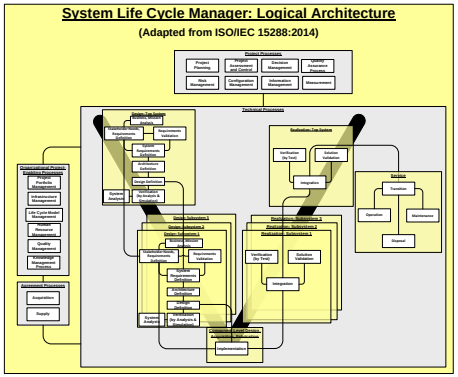
An ASELCM Insight: Technical Debt vs. Information Debt-- More Than One Representation (Model View) of the Same Underlying Reality

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)

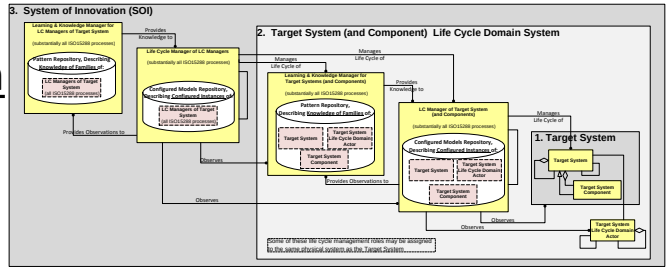
Scrum Pattern

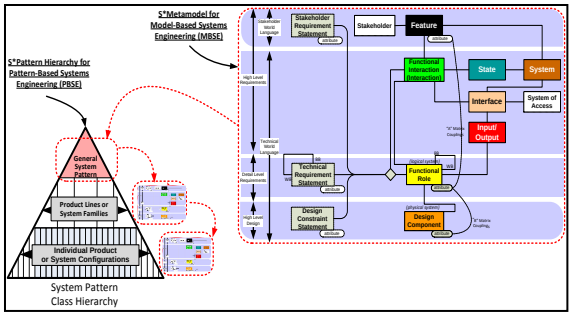




ISO15288 Pattern

ASELC Pattern



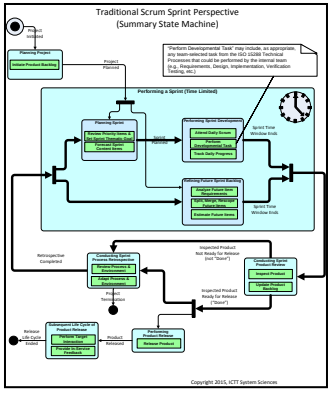


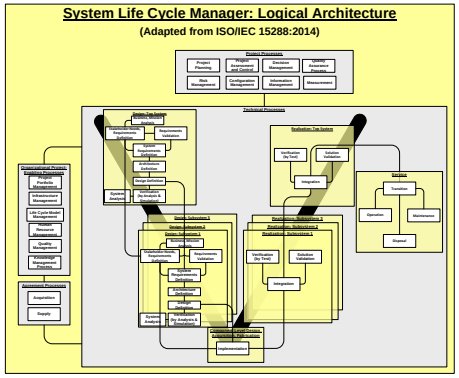
S*Metamodel

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





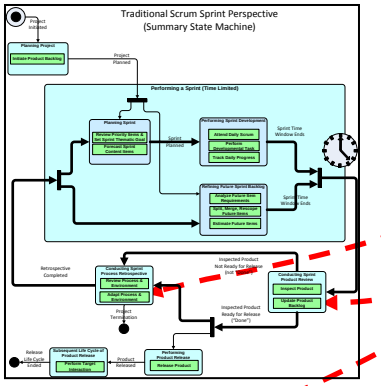
ISO15288 Pattern

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

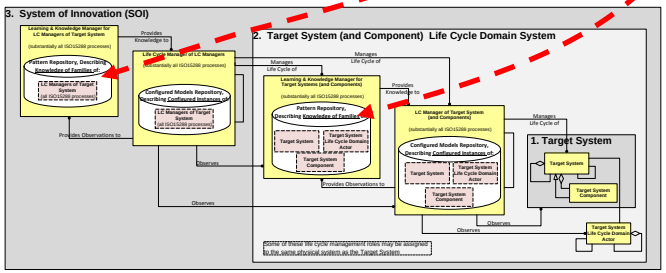
- The Scrum Model also abstracts complex learning behavior, into simple-looking form—but it is still strongly expected to occur as part of the Agile Process, and is more explicitly represented in the ASELC Pattern, as capture of Pattern information—not assumed to be only in human minds.

Learning

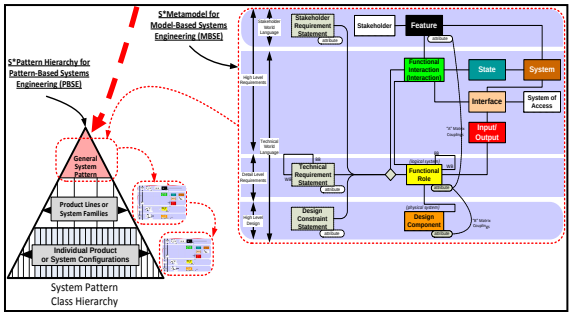
Scrum Pattern



ASELC Pattern



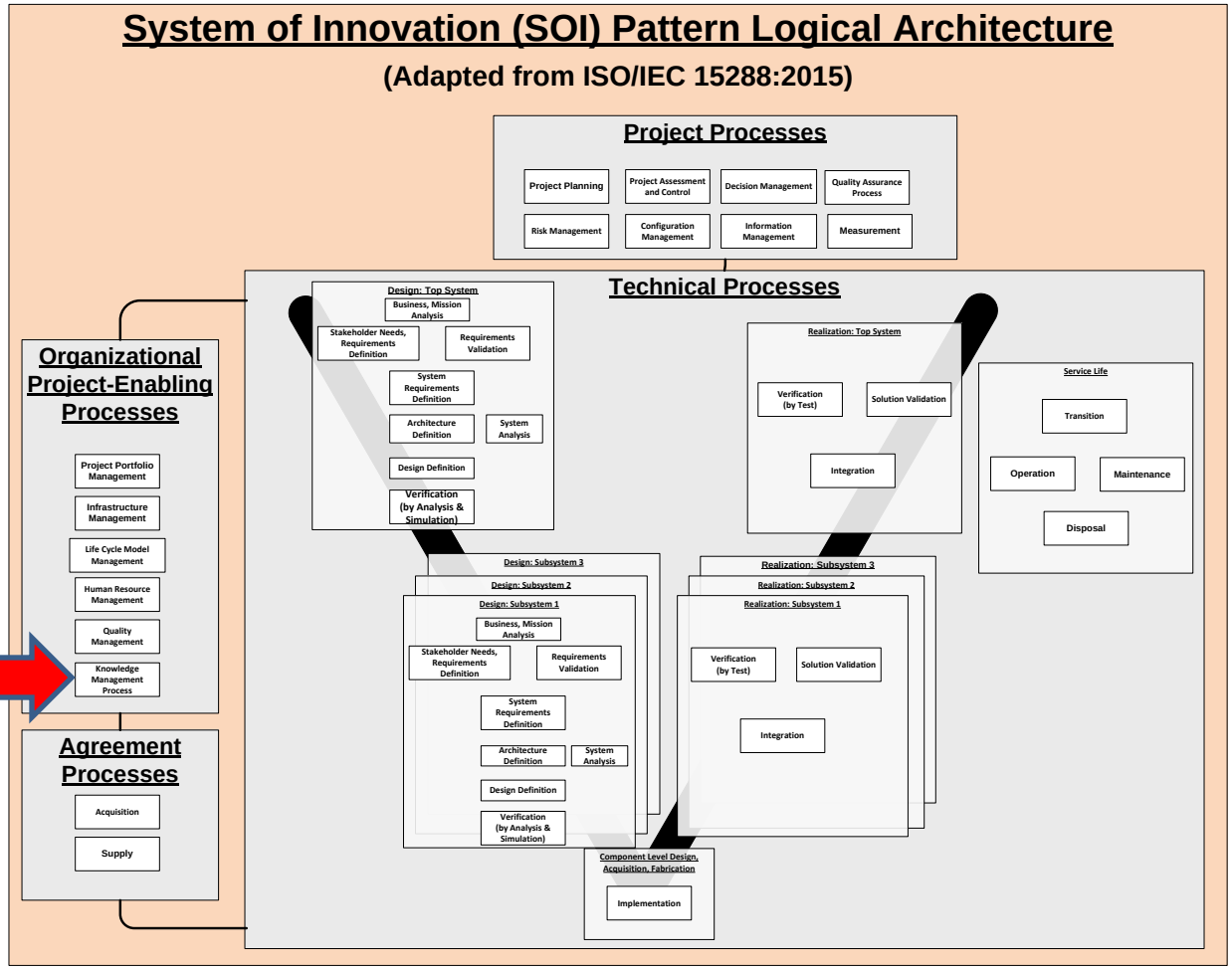
S*Metamodel



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

Learning and 15288 Processes

- Although “Knowledge Management” appears as a single process in the 2015 version of ISO15288, the System of Innovation Pattern shows that (technical) learning actually occurs across substantially <u>all</u> of the Technical Processes of ISO 15288.



1. Schindel, W., Dove, R, “Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern”, submitted to INCOSE IS2016 Symposium.
2. Dove, R., Schindel, W., Scrapper, C., “Agile Systems Engineering Process Features Collective Culture, Consciousness, and Conscience at SSC Pacific Unmanned Systems Group”, submitted to INCOSE IS2016 Symposium.
3. Rick Dove, Agile Systems Project Definition: http://www.parshift.com/s/AgileSELifeCycleModelProject-INCOSE.pdf
4. Rick Dove, Ralph LaBarge, “Fundamentals of Agile Systems Engineering—Part 1” and “Part 2”, INCOSE IS2014, July, 2014.
5. Rick Dove, *Response Ability: The Language, Structure, and Culture of the Agile Enterprise,* Wiley, 2001.
6. MBSE Methodology Summary: Pattern-Based Systems Engineering (PBSE), Based On S*MBSE Models, INCOSE survey of MBSE methodologies: http://www.omgwiki.org/MBSE/doku.php?id=mbse:pbse
7. W. Schindel, T. Peterson, “Introduction to Pattern-Based Systems Engineering (PBSE): Leveraging MBSE Techniques”, in *Proc. of INCOSE 2013 International Symposium*, Tutorial, June, 2013.
8. INCOSE/OMG Patterns Working Group 2013-14 http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns
9. W. Schindel, “System Life Cycle Trajectories: Tracking Innovation Paths Using System DNA”, in *Proc. of INCOSE International Symposium 2015*, July, 2015.
10. W. Schindel, T. Peterson, “Pattern Based Systems Engineering – Leveraging Model Based Systems Engineering for Cyber-Physical Systems”, in *Proc. of NDIA Ground Vehicle SE and Technology Symposium*, Aug, 2014.
11. ISO/IEC 15288: Systems Engineering—System Life Cycle Processes. ISO (2015).
12. Martin Fowler and Jim Highsmith. “The Agile Manifesto”. *Dr. Dobb's Journal*, August, 2001. www.drdobbs.com/open-source/the-agile-manifesto/184414755 .
13. J. Highsmith. *Agile Software Development Ecosystems*. Addison-Wesley Professional, 2002.
14. W. Schindel, “Maps or Itineraries?: A Systems Engineering Insight from Ancient Navigators”, n Proc. of INCOSE International Symposium 2015, July, 2015.
15. ----------, “System Life Cycle Trajectories: Tracking Innovation Paths Using System DNA”, in Proc. of INCOSE International Symposium 2015, July, 2015.
16. Schindel and Beihoff: “Systems of Innovation I: Models of Their Health and Pathologies”, Proc. of INCOSE International Symposium, 2012.
17. W. Schindel, “Systems of Innovation II: The Emergence of Purpose”, Proceedings of INCOSE 2013 International Symposium (2013)