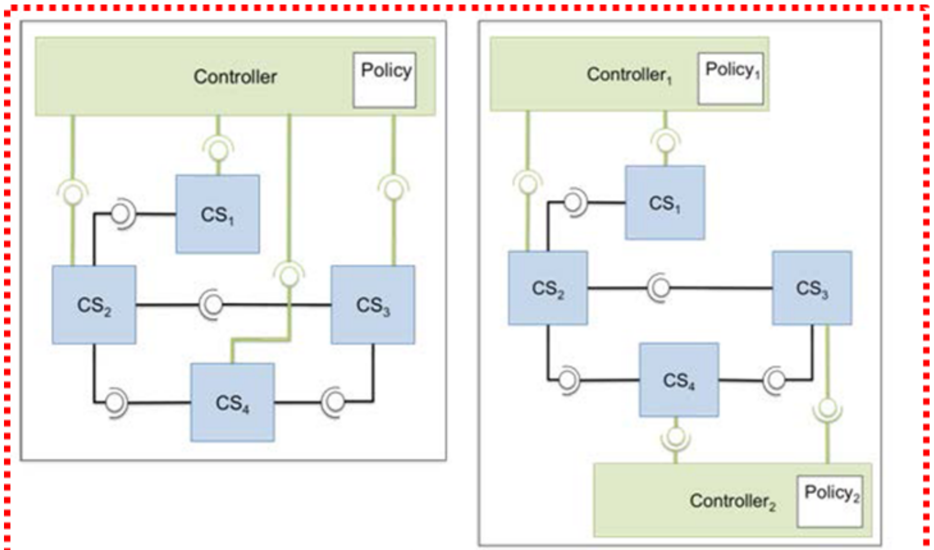
A Joint Workshop by:

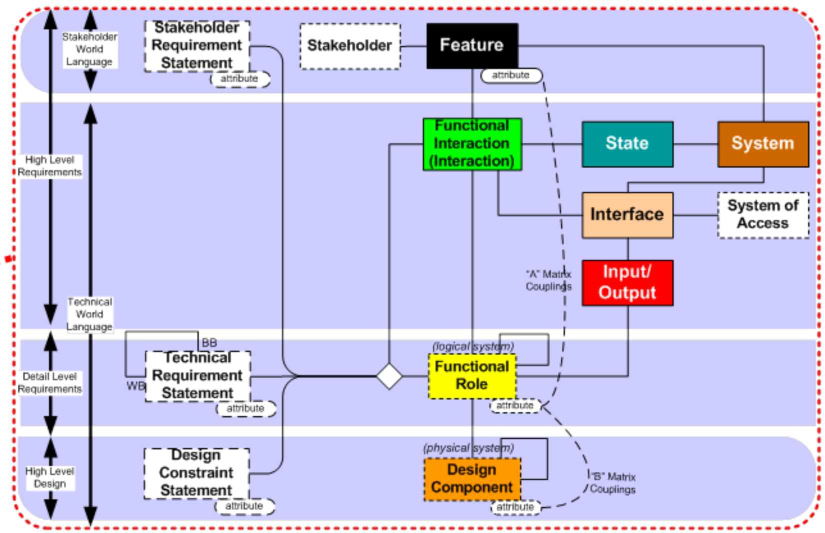
- INCOSE Patterns Working Group
- INCOSE Systems of Systems Working Group

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

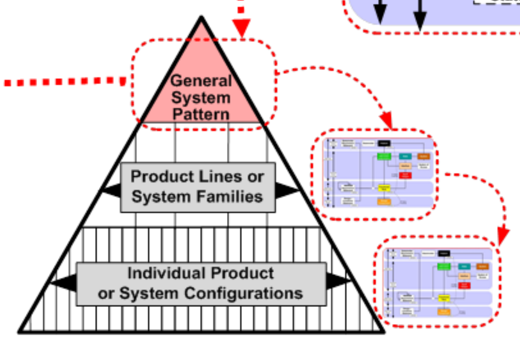
Patterns in Systems of Systems



SoS Pattern



S*Metamodel for (MBSE)



S*Patterns Class Hierarchy for PBSE

V1.3.8

Contents

- Workshop motivation, background, and objectives
- Agenda / time line

- Workshop introduction
- Introduction to Systems of Systems
- Introduction to S*Patterns, as applicable to Systems of Systems
- Teams Break-Out Session
- Plenary Discussion

- References

- Attachments:
  - Break out session materials
  - Workshop feedback materials

Motivation:

- Patterns have been very useful in other communities as a mechanism to identify, structure and share information and reusable products.
- Patterns are also fundamental to understanding system phenomena.

Working Group Background:

- The INCOSE Systems of Systems Working Group is several years into exploration of Systems of Systems, including principles and examples of SoS Patterns[1,2].
- The INCOSE Patterns Working Group is several years into exploration of S*Patterns, representing MBSE models of systems across domains, using the S*Metamodel[3,4].

Objectives of this Workshop:

- This workshop provides a combined perspective, beginning to illustrate SoS Patterns using the S*Metamodel.
- Start the process of collecting helpful information on patterns for SoS engineers.
- Begin building a community to develop and support SoSE patterns.
- Attendees will have the opportunity to compare their own experiences with these ideas, identifying key issues and ideas of future interest.

1. http://www.incose.org/docs/default-source/Working-Groups/System-of-Systems/sos-wg-presentation-oct-2015.pptx?sfvrsn=0
2. http://www.incose.org/ChaptersGroups/WorkingGroups/processes/system-of-systems
3. http://www.omgwiki.org/MBSE/doku.php?id=mbse:pbse
4. http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns

Workshop Agenda and Time Line

Workshop Session	Time	WG Lead
Introduction, review of workshop objectives and agenda	13:30 – 13:40	Joint
Introduction to Systems of Systems	13:40 – 14:20	SoS WG
S*Patterns and their Application to SoS	14:20 – 15:00	Patterns WG
IW-Wide Break	*15:00 - 15:30*	
Structured interactive small group breakout sessions—to brainstorm and discuss suggestions	15:30 – 17:00	Breakout Teams
Plenary discussion, reflecting on previous work and plans for potential future activities	17:00 – 17:30	Plenary
Adjourn	*17:30*	

Introduction to

Systems of Systems

Systems of Systems Working Group

Topics

- Definitions
- Characteristics
- SoS Types
- Comparing Systems with SoS
- Implications for Systems Engineering
- Challenges: “SoS Pain Points”
- Pain Points and Patterns

System of Systems

A set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities

Systems of Systems Engineering

The process of planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into a system-of-systems capability that is greater than the sum of the capabilities of the constituent parts

Definitions

Source	Definition
SE Body of Knowledge	A SoS is an integration of a finite number of constituent systems which are independent and operatable, and which are networked together for a period of time to achieve a certain higher goal. (Jamshidi 2009)
INCOSE SE Handbook	[A] system-of-interest whose elements are managerially and/or operationally independent systems. These interoperating and/or integrated collections of systems produce results unachievable by the individual systems alone.
Draft ISO 15288 SoS Annex	A system of systems (SoS) is a system-of-interest (SOI) whose elements are themselves systems. A SoS brings together a set of systems for a task that none of the systems can accomplish on its own. Each constituent system keeps its own management, goals, and resources while coordinating within the SoS and adapting to meet SoS goals.
US DoD SoS SE Guide	A set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities.


Energy

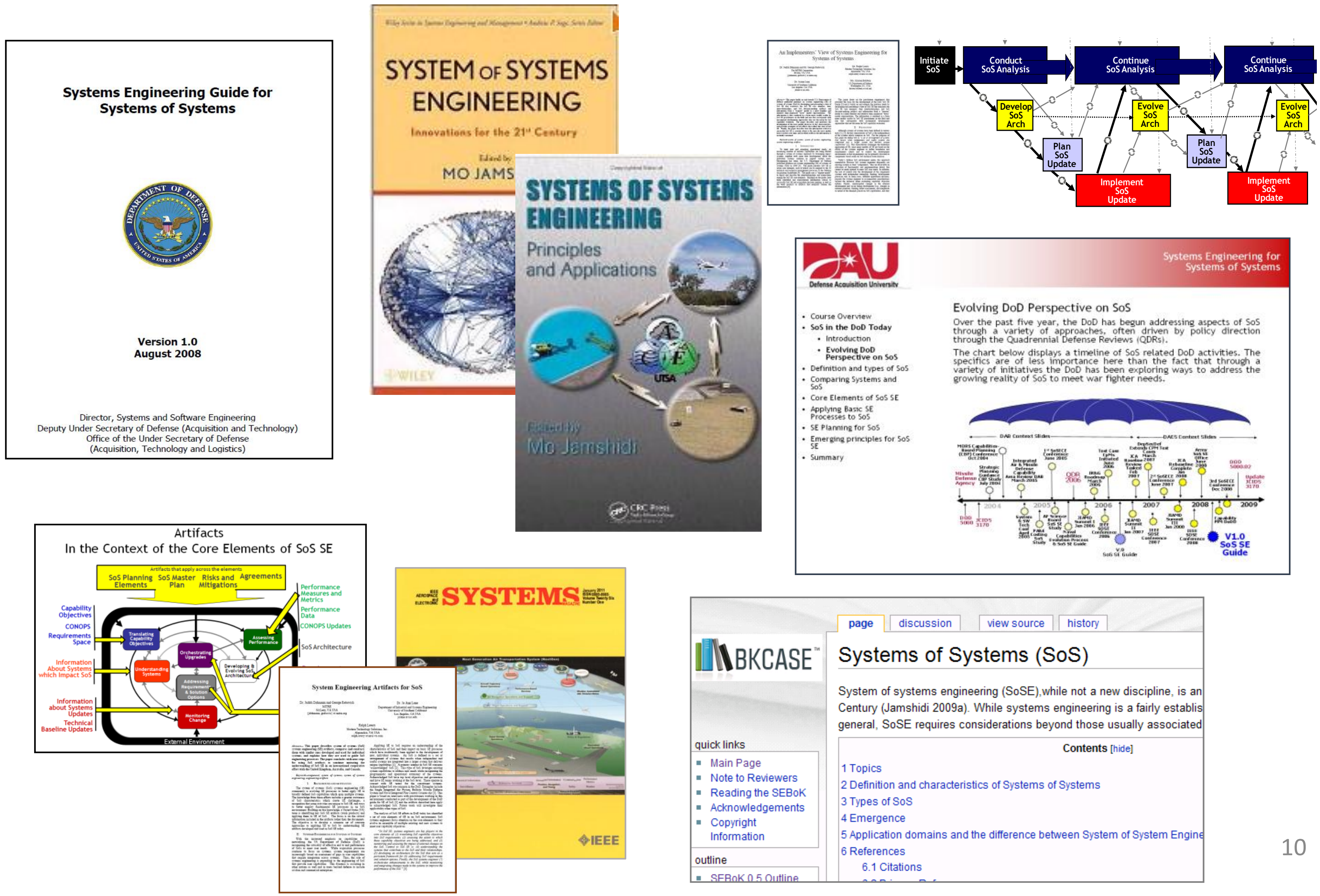

Water


Air Transportation


Missile Defense
SENSORS
BOOST / ASCENT
MIDCOURSE
TERMINAL
Command, Control, Battle Management, and Communications (C2BMC)

Investigations in to SoS SE for Defense & Beyond

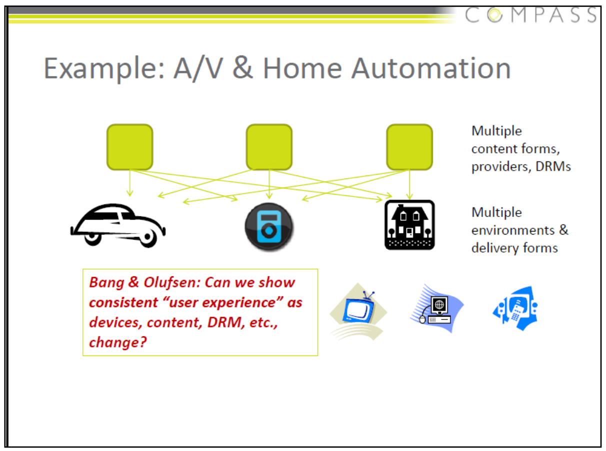
INCOSE 2016 international workshop
Los Angeles, CA, USA
January 30 - February 2, 2016

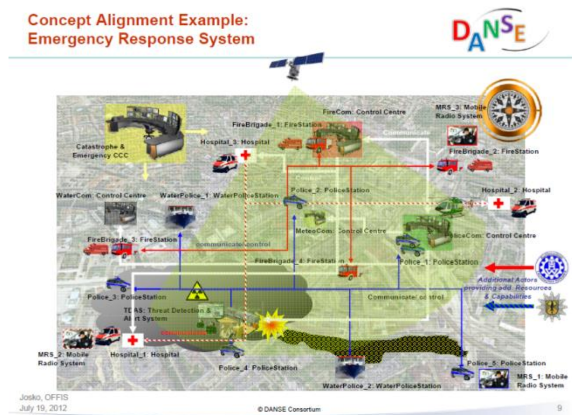


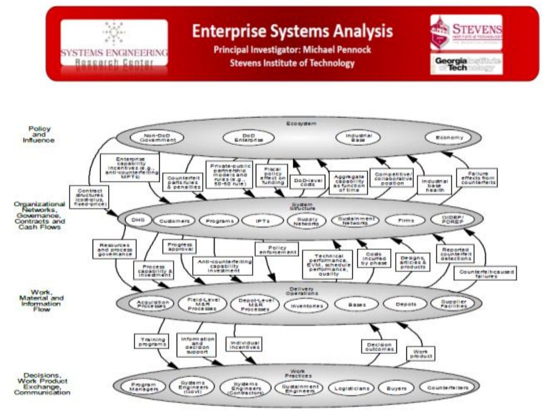
Maier SoS Characterization

- Maier (1998) postulated five key characteristics of SoS:
  - Operational independence of component systems
  - Managerial independence of component systems
  - Geographical distribution
  - Evolutionary development processes
  - Emergent behavior

Scale and Scope of SoS



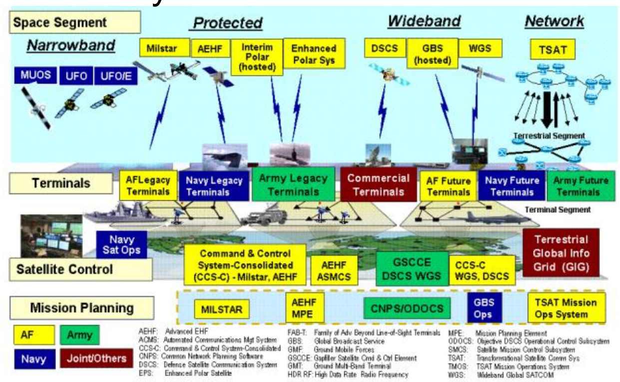




Technical ----- Socio -Technical ----- Enterprise

SoS Domains in Defense

Military Satellite Communications



Tactical Vehicle

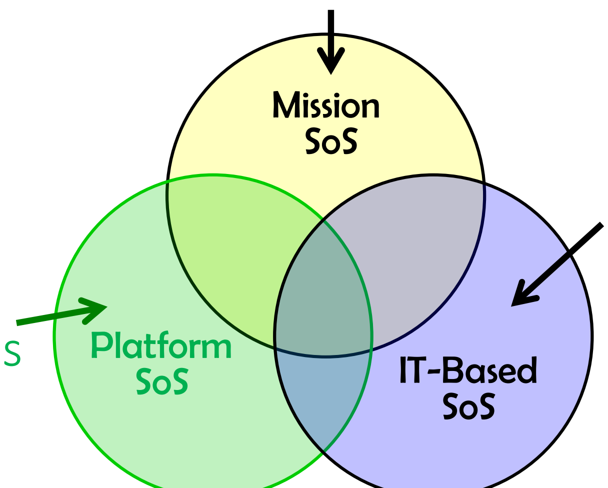
Operations Center

Missions

Sets of systems working together to provide a broader capability or mission

Platforms

A military platform (e.g. ship, aircraft, satellite, ground vehicle) equipped with independent systems (e.g. sensor, weapons, communications) needed to meet platform objectives



Information Technology

Networked information systems to support operations within or across platforms or systems to meet mission or capability objectives

SoS Types

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

Type	Definition
Directed	Directed SoS are those in which the SoS is engineered and managed to fulfill specific purposes. It is centrally managed during long-term operation to continue to fulfill those purposes as well as any new ones the system owners might wish to address. The component systems maintain an ability to operate independently, but their normal operational mode is subordinated to the centrally managed purpose.
Acknowledged	Acknowledged SoS have recognized objectives, a designated manager, and resources for the SoS; however, the constituent systems retain their independent ownership, objectives, funding, development, and sustainment approaches. Changes in the systems are based on cooperative agreements between the SoS and the system.
Collaborative	In collaborative SoS, the component systems interact more or less voluntarily to fulfill agreed-upon central purposes.
Virtual	Virtual SoS lacks a central management authority and a centrally agreed-upon purpose for the system of systems. Large-scale behavior emerges—and may be desirable—but this type of SoS relies upon relatively invisible, self-organizing mechanisms to maintain it.

- Many SoS exist but are not recognized and develop and evolve without benefit of SE
- Types apply when the SoS is recognized and treated as an SoS
- In reality, most actual SoS are a combination of these types

https://acc.dau.mil/dag4

Differences Between Systems and SoS as They Apply to SE

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

	Systems Engineering	Systems of Systems Engineering
Management & Oversight		
System	Physical engineering	Socio-technical management and engineering
Stakeholder Involvement	Clear set of stakeholders	Multiple levels of stakeholders with mixed and possibly competing interests
Governance	Aligned management and funding	Added levels of complexity due to management and funding for both SoS and systems; SoS does not have control over all constituent systems
Operational Focus (Goals)		
Operational Focus	Designed and developed to meet common objectives	Called upon to meet new SoS objectives using systems whose objectives may or may not align with the SoS objectives
Implementation		
Acquisition/Development	Aligned to established acquisition and [illegible]sses	Cross multiple system lifecycles across asynchronous acquisition and development efforts, involving legacy systems, developmental systems, and technology insertion
Process	Well-established	Learning and Adaptation
Test and Evaluation	Test and evaluation of the system is possible	Testing is more challenging due to systems' asynchronous life cycles and given the complexity of all the parts
Engineering and Design Considerations (Engineering & Design)		
Boundaries and Interfaces	Focuses on boundaries and interfaces	Focus on identifying systems contributing to SoS objectives and enabling flow of data, control and functionality across the SoS while balancing needs of the systems OR focus on interactions between systems. Difficult to define system-of-interest
Performance and Behavior	Performance of the system to meet performance objectives	Performance across the SoS that satisfies SoS use capability needs while balancing needs of the systems
Metrics	Well defined (e.g. INCOSE handbook)	Difficult to define, agree, and quantify

From SEBoK 1.1

SoS Pain Points

SoS Authority

What are effective collaboration patterns in SoS?

Leadership

What are the roles and characteristics of effective SoS leaders?

Capabilities & Requirements

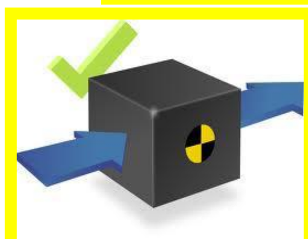
How can SE address SoS capabilities and requirements?

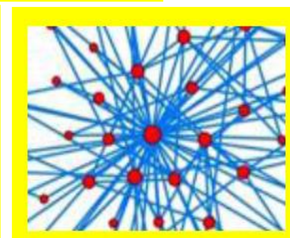
Constituent Systems

What are effective approaches to integrating constituent systems?

Testing, Validation & Learning

How can SE approach SoS validation, testing, and continuous learning in SoS?





SoS Principles

What are the key SoS thinking principles?

Autonomy, Interdependencies & Emergence

How can SE address the complexities of interdependencies and emergent behaviors?

Challenges and Opportunities

SoS Authorities

- **SoS have been characterized in terms of these authority relationships (SEBoK 1.0)**
  - Directed
  - Acknowledged
  - Collaborative
  - Virtual

- **In defense applications**
  - Authority conflicts often dominate discussion of SoS
  - Focus on how to legitimately arbitrate these opposing forces to balance **the values of the systems with those of the SoS**
- **In non-defense contexts**
  - Same issues can prevail but without the larger organizational constraints
  - Focus is on creation of the incentives and development environment which allow the systems to proceed to meet their own objectives while working cooperatively to support broader objectives

What are effective collaboration patterns in SoS?

Leadership

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

- **Leadership issues implied in the SoS authority pain point**
  - Lack of structured control assumed by SE for systems faces a void, calling for alternatives to provide coherence and direction, including influence and incentives
  - Without the type of traditional top down control, there are clear challenges for application of SE at the SoS level
  - An issue in both defense and non-defense

- **Increased discussion about organizational leadership skills as a key element in SE effectiveness**
  - Especially as systems have become more complex as has the SE environment
- **SoS organizational and technical complexity -- multiple independent stakeholders with their own interests and independence -- makes the role of leadership in SoS even more important**

What are the roles and characteristics of effective SoS leaders?

Constituent Systems (1 of 2)

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

- **Coordination and management of multiple independent constituent systems in SoS**
  - Legacy systems which "… not configured or managed to allow insertion into the over-all system of systems. This creates interoperability concerns between the older and newer systems."
  - Managerial and evolutionary independence can mean that "Constituent systems change in response to the perceived goals for that system, usually with little regard for the impact on SoS goals or behaviors."
- **Risks of coordinated constituent system SoS support beyond data exchange**
  - "In the cases where systems are owned/operated by different organizations … the systems may transfer data and information reliably between systems (if you're lucky), but different processes, cultures, working practices between different participating organizations can lead to problems."

What are effective approaches to integrating constituent systems?

Constituent Systems (2 of 2)

- **Poses core technical issues for SoS**
  - Systems identified for the SoS be limited in the degree to which they can support the SoS initially and their commitments to other users may mean that they may not be compatible with the SoS over time
  - Risk of mismatch in understanding the action or data provided by one system to the SoS if the systems context differs from that of the SoS
  - Impact on the architecture for the SoS which is essentially an overlay to these systems providing the framework for their cooperative activity and evolution over time (Ref SEBOK 1.0 SoS)
  - Implications may be felt in unpredictable SoS behavior as discuss below in technical area of autonomy and emergence.

What are effective approaches to integrating constituent systems?

Capabilities & Requirements (1 of 2)

- **The issue**
  - Traditionally, SE process begins with a clear, complete set of user **requirements** and SE provides a disciplined approach to develop a system to meet these requirements.
  - Typically, SoS are comprised of **multiple independent** systems with their own requirements working towards **broader capability objectives.**
  - In the best case the SoS capability needs are met by the systems as they meet their own local requirements, but in many cases the SoS needs may **not be consistent** with the needs of the constituent systems.

- **In these cases, the SoS SE needs to identify alternative approaches to meeting those needs through changes to the constituent systems or additions of other systems to the SoS**.
  - This is in effect asking the systems to take on new requirements with the **SoS acting in a way as the 'user'.**

How can SE address SoS capabilities and requirements?

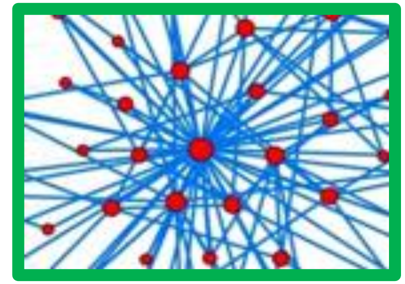
Capabilities & Requirements (2 of 2)

- **In SoS SE, it is not useful to develop detailed requirements at the SoS level, but rather to look at user capability needs at a higher level of abstraction**
  - Identify a multiple alternatives to adapt systems to meet the higher level SoS needs since the systems will each have their own constraints (both technical and non-technical)
  - Important for the SoS to have a wider range of options available since the preferred approach may not be feasible.
  - SoS capabilities may draw on a wider variety of non-material aspects of organizations which means that addressing SoS capability needs may go beyond adapting systems specific functionality and interfaces.

How can SE address SoS capabilities and requirements?

Autonomy, Interdependencies & Emergence

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



- **Combining component systems into SoS produce unexpected behavior.**

  "Well-structured approaches for 'design for emergence' are not available."

- **Complex relationships among systems in an SoS are often poorly understood and difficult to analyze**

  "Systems often have interdependencies that are either unknown or unacknowledged. This is exacerbated by interdependencies between systems in development, a system in development and fielded systems, and fielded systems; further, this is compounded by multiple combinations of all of these."

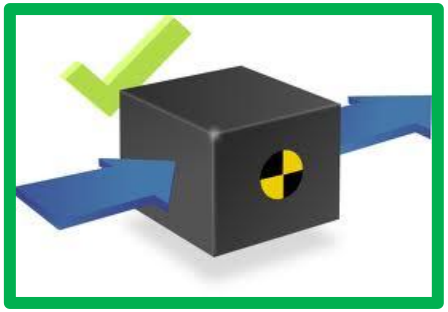
  "We lack methods for representing the SoS analytically so these interdependencies can be understood, and the SE of the SoS could examine impacts of different SoS changes."

- **Need for methods and tools to support the modeling and prediction of complex SoS behaviors including analysis and architecting methods**

How can SE address the complexities of SoS interdependencies and emergent behaviors?

Testing, Validation & Learning (1 of 2)

- **Most SoS face issues of conducting end- to-end testing**
  - Need a clear understanding of the SoS objectives and metrics
  - Depending on the SoS context there may be not funding or authority for SoS testing.
  - With multiple constituent systems on asynchronous development cycles, finding ways to conduct tradition testing across the SoS can be difficult is not impossible.
  - Many SoS are large and diverse making tradition full end-to-end testing with every change in a constituent prohibitively costly.
  - Often the only way to get a good measure of an SoS performance is from data collected from actual operations.



- **Nonetheless the SoS SE team needs to ensure continuity of operation and performance of the SoS despite these challenges**

How can SE approach SoS validation, testing, and continuous learning in SoS?

Testing, Validation & Learning (2 of 2)

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

- **These problems have been recognized and addressed in several ways**
  - Modeling and simulation environments for addressing effects of changes on SoS performance and providing test tools for augmenting system testing to assess SoS impacts
  - Architectures which minimize impacts of changes in one part of the SoS on other parts and the SoS performance as a whole.
  - Methods to identify the areas which may pose greatest risk and focus attention on these using data from a variety of sources as well as from more traditional testing.
  - Built-in ongoing validation throughout SoS evolution
- **Focus on approaches like incremental validation, reflecting a perspective that looks at significant learning going on over the life of an SoS**

How can SE approach SoS validation, testing, and continuous learning in SoS?

SoS Principles

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

- **Indicated were either [missing] or (needed) items for successful SoS, including:**
  - ["Lack of] formalized processes"
  - ["Lack of] examples of SoS Success
  - ["SoS requires] better trust to the work flow
  - ("Keep a SoS together) - It is very important to plan, design, purchase and maintain a SoS entity based on the SoS idea."
- **Cross cutting area – basic principles underlying other areas**
  - This area is one where progress in identifying and articulating SoS principles ('SoS Thinking') and examples, could have benefit to the discipline

What are the key SoS thinking principles?

Pain Points & Patterns?

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

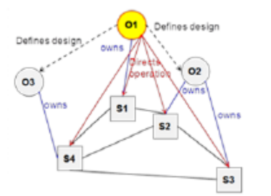
SoS Authority

What are effective collaboration patterns in SoS?

SoS Types – T-Area-SoS Presentation

Directed SoS

- Integrated SoS built and management to fulfill specific purposes
- Centrally managed and evolved
- Component systems maintain ability to operate independently
- Normal op mode is subordinated to central purpose



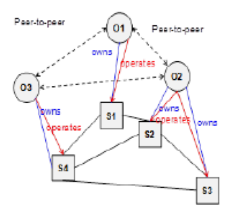
In a Directed SoS: operators O2 and O3 accept direction from O1 in terms of the specification and operation of the systems they own (O2 owns systems S2 and S3; O3 owns S4) This type of SoS is highly controlled by the central managing entity (O1).

From Draft 15288 SoS Annex

From T-Area-SoS Strategic Research Agenda

Collaborative SoS

- Component systems interact voluntarily to fulfill agreed upon purposes
- Collectively decide how to interoperate, enforcing and maintaining standards



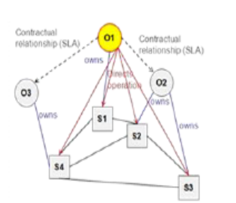
In a Collaborative SoS: there is mutual agreement to collaborate; usually covered by agreements of some form, but there is no overall managing entity; systems owners (O1, O2, O3) operate their own systems and collaborate with others to realize some shared benefit.

From Draft 15288 SoS Annex

From T-Area-SoS Strategic Research Agenda

Acknowledged SoS

- Recognized objectives, a designated manager, and resources for the SoS
- Constituent systems retain their independent ownership, management and resources



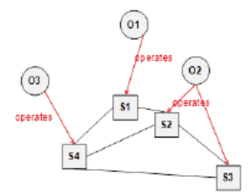
In an Acknowledged SoS: O1 directs choice of systems and operation; O2 and O3 have a contractual relationship (e.g. Service Level Agreement) with O1. In this case, the central managing entity (O1) has less control over the systems owned by O2 and O3 (S2, S3, S4) and must rely more on influence.

From Draft 15288 SoS Annex

From T-Area-SoS Strategic Research Agenda

Virtual SoS

- Lack a central management authority
- Lack of agreed upon purpose
- Emerging behaviors that relay upon relativity invisible mechanisms to maintain it



In a Virtual SoS: Owners (O1, O2, O3) access other systems through their own systems in order to realize individually sought benefits, though high level emergent behaviour may still occur. There is no overall goal, no central management and interoperation is achieved by recognized protocols, or standards, not through individual agreements between pairs of systems.

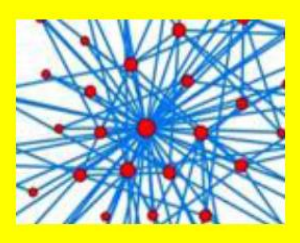
From T-Area-SoS Strategic Research Agenda

Pain Points & Patterns

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

Autonomy, Interdependencies & Emergence

How can SE address the complexities of interdependencies and emergent behaviors?



COMPASS Architecture Patterns

Pipe & Filter Architecture Pattern

Newcastle University UK | Malaysia | Singapore

- Data or materials processed from input form to output form
- Filters represent the processing steps
- Pipes represent connections between Filters
- Filters are independent, do not share state or know each other's identities

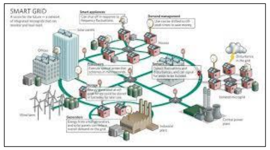
Garlan & Shaw 1996, Buschmann et al 1996

17/01/2016 21

Infrastructure Grid Architecture Pattern

Newcastle University UK | Malaysia | Singapore

- CSs exchange data in addition to a flow of energy/material with their 'neighbours'
- Typical high-level goal to ensure control of large-scale distribution of material (e.g., electric power, water, traffic access) to large area
- Range of centralisation

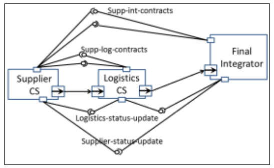


17/01/2016 24

Supply Chain Architecture Pattern

Newcastle University UK | Malaysia | Singapore

- Similar to pipe and filter pattern
- Overall SoS goal is the ultimate delivery or manufacture of a final product or service
- CSs have different roles (suppliers/integrators/…)
- Supply chains can be reconfigured if CSs conform to correct contracts

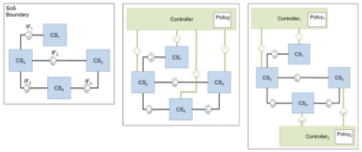


17/01/2016 25

Reconfigurable Control Architecture Pattern

Newcastle University UK | Malaysia | Singapore

- Dynamic architectural reconfiguration addresses SoS challenges:
  - continuous evolution
  - challenging environment
  - autonomous and independent CSs
- Use *metadata* to describe the functions CS offer
- A *policy* details *when* and *how* to reconfigure SoS



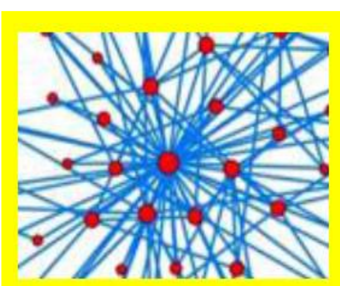
17/01/2016

SoS Pain Points

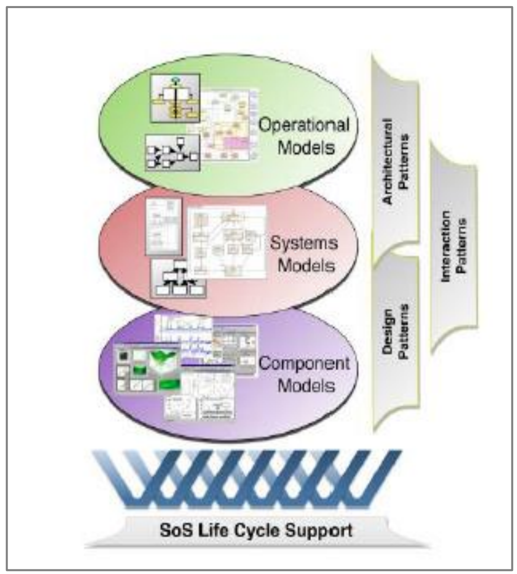
INCOSE 2016 International workshop
Los Angeles, CA, USA
January 30 - February 2, 2016

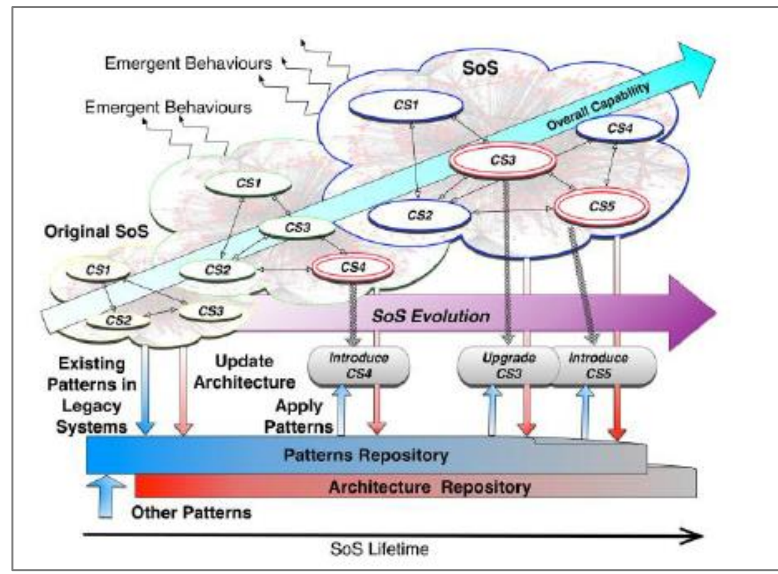
Autonomy, Interdependencies & Emergence

How can SE address the complexities of interdependencies and emergent behaviors?



DANSE
Patterns to Understand SoS Evolution



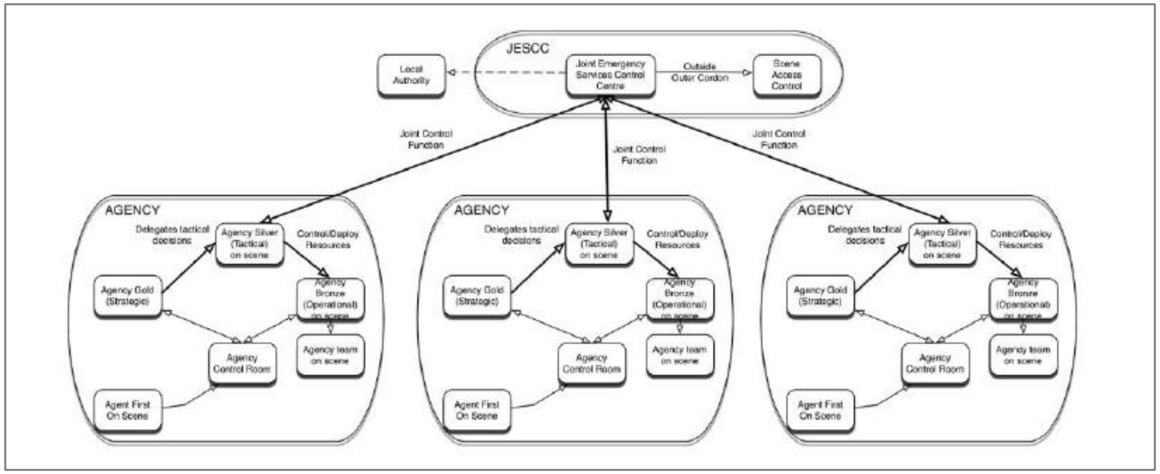


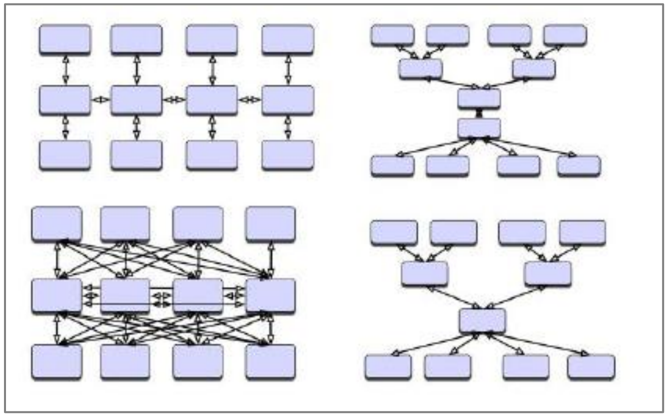
Pain Points & Patterns

DANSE SoS Patterns

Constituent Systems

What are effective approaches to integrating constituent systems?





Pain Points & Patterns

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

Leadership

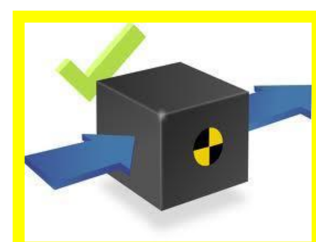
What are the roles and characteristics of effective SoS leaders?

Capabilities & Requirements

How can SE address SoS capabilities and requirements?

Testing, Validation & Learning

How can SE approach SoS validation, testing, and continuous learning in SoS?



SoS Principles

What are the key SoS thinking principles?

Opportunities for Patterns?

Introduction to

S*Patterns-- As Applicable to Systems of Systems

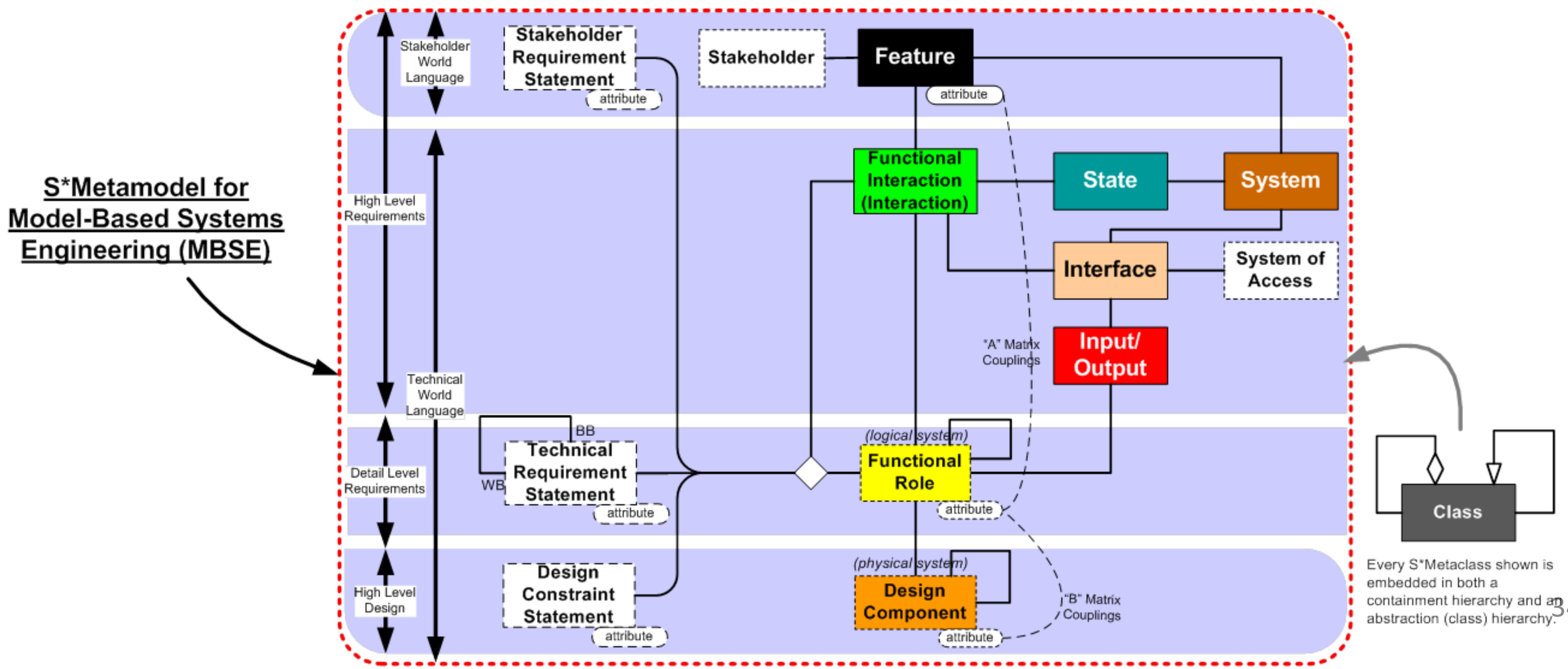
MBSE Patterns Working Group

Introduction to S*Patterns--
As Applicable to Systems of Systems

- What are S*Models?
- What are S*Patterns?
- What do S*Patterns bring to MBSE representations of Systems of Systems?
- Excerpts from some S*Patterns describing Systems of Systems

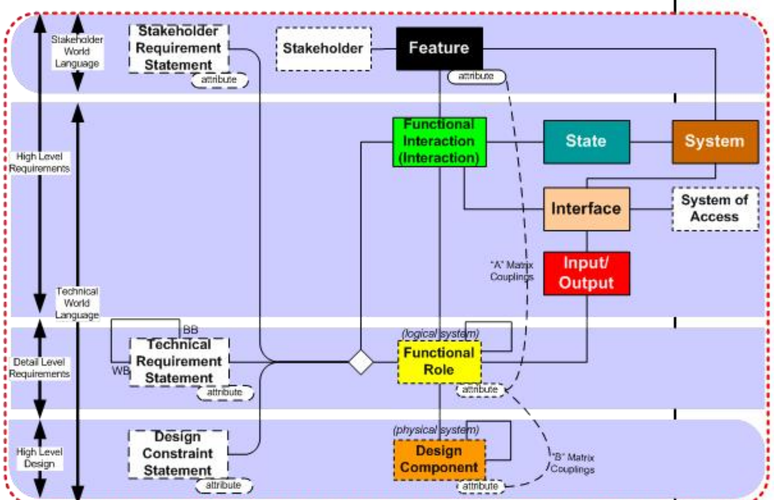
What are S*Models?

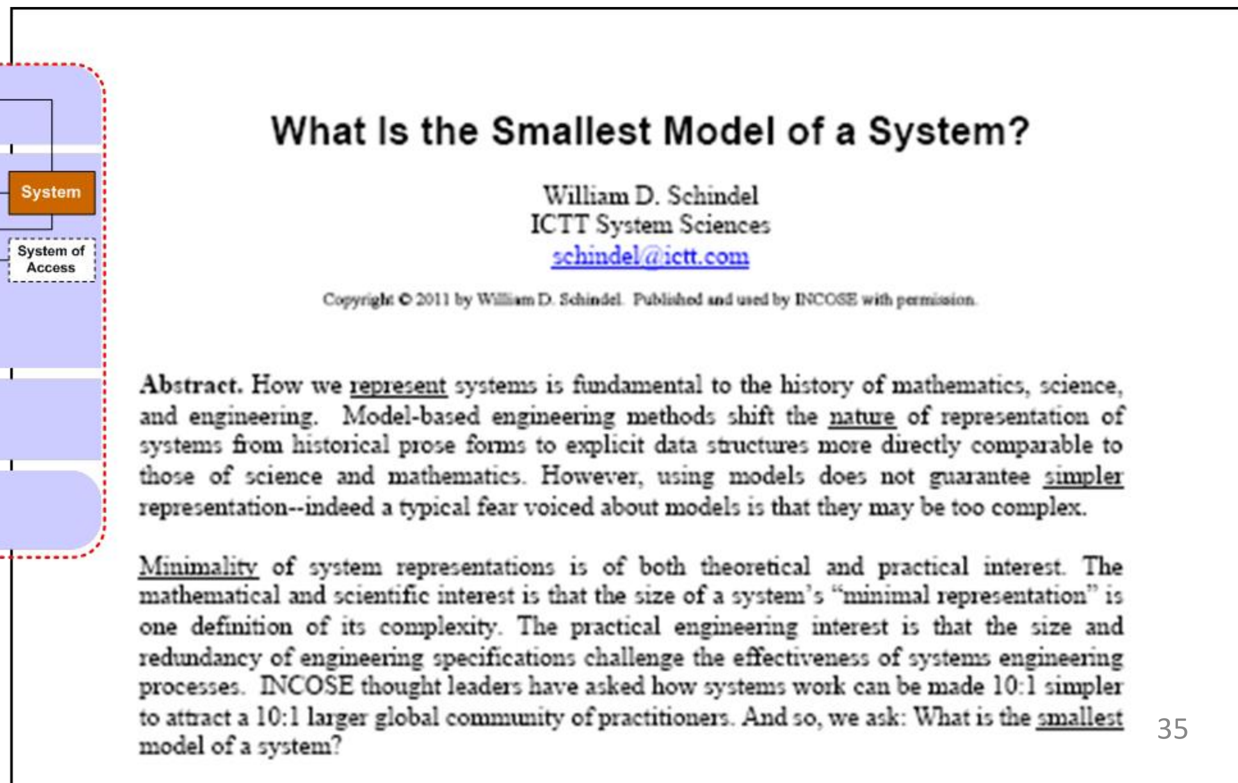
- <u>S*Models</u> are MBSE models that are based on the S*Metamodel:
  - Independent of specific modeling language.
  - S*Metamodel maps into any contemporary modeling language, including OMG SysML®, third party COTS tools.



What are S*Models?

- S*Models are MBSE models that are based on the S*Metamodel:
  - The smallest amount of modeled information necessary for purposes of science or engineering.
  - What parts of S*Models might be helpful for SoS models?





What Is the Smallest Model of a System?

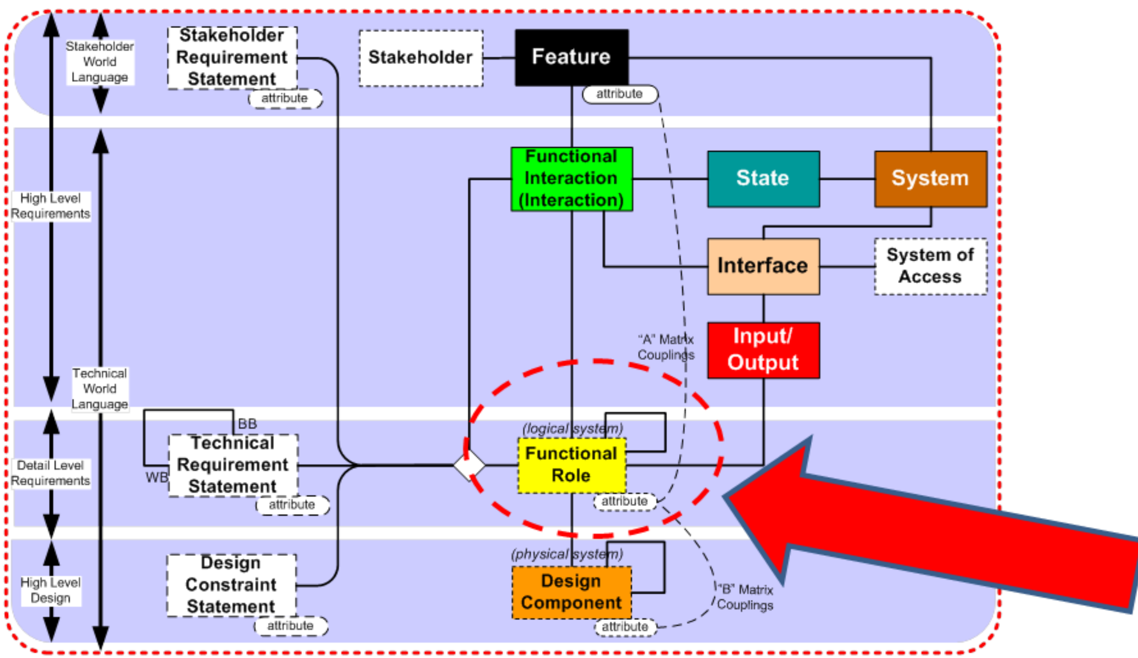
William D. Schindel
ICTT System Sciences
schindel@ictt.com

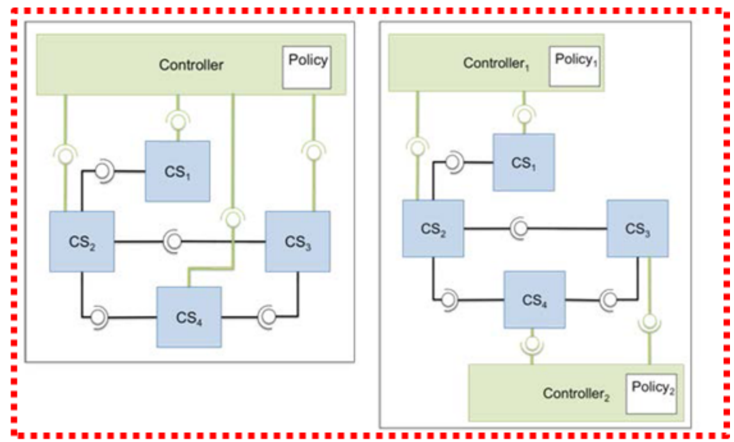
Abstract. How we represent systems is fundamental to the history of mathematics, science, and engineering. Model-based engineering methods shift the nature of representation of systems from historical prose forms to explicit data structures more directly comparable to those of science and mathematics. 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?

Some Elements of S*Models

- **Functional Roles**: Describe chunks of behavior, independent of the physical things that perform it, parameterized by role Attributes.
- **Architectural Relationships**: These connect Functional Roles, to describe Logical Architecture
- Both of these can be seen in SoS patterns published by the SoS WG

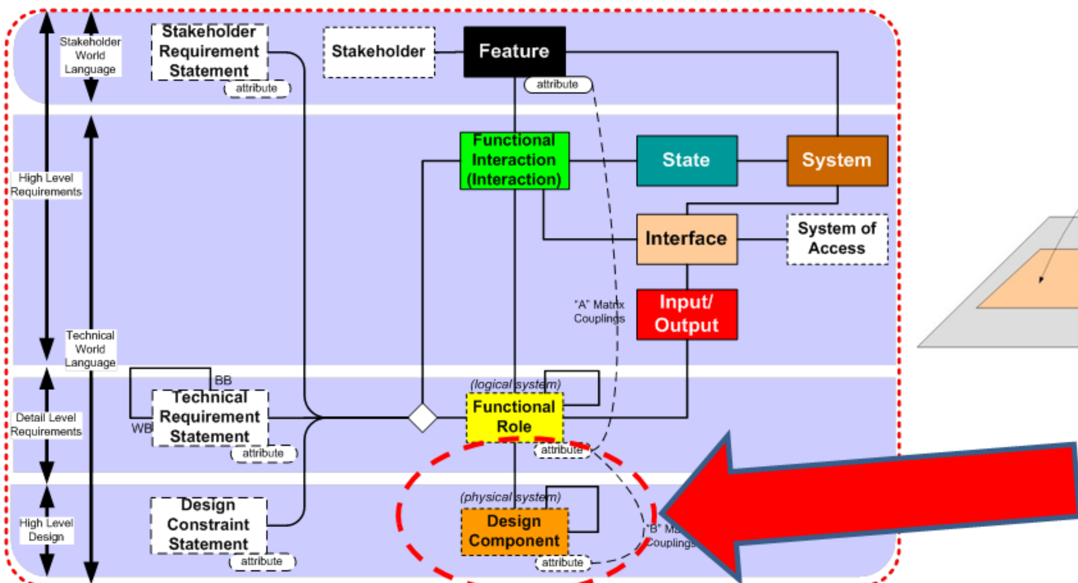


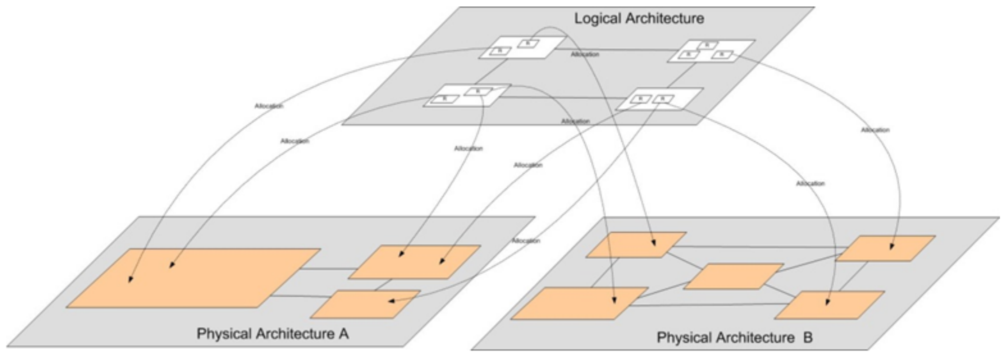


Extract from an SoS Pattern

Some Elements of S*Models

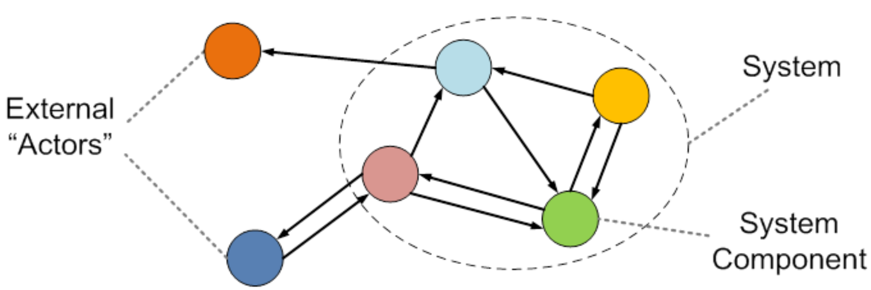
- **<u>Design Components</u>**: Model component identities, without behavior, and parameterized by Attributes.
- **<u>Architectural Relationships</u>**: Connect Design Components, to describe Physical Architecture.
- **<u>Allocation Relationships</u>**: Describe allocations of Functional Roles to Design Components.
- Show patterns of allocations to different physical architectures.

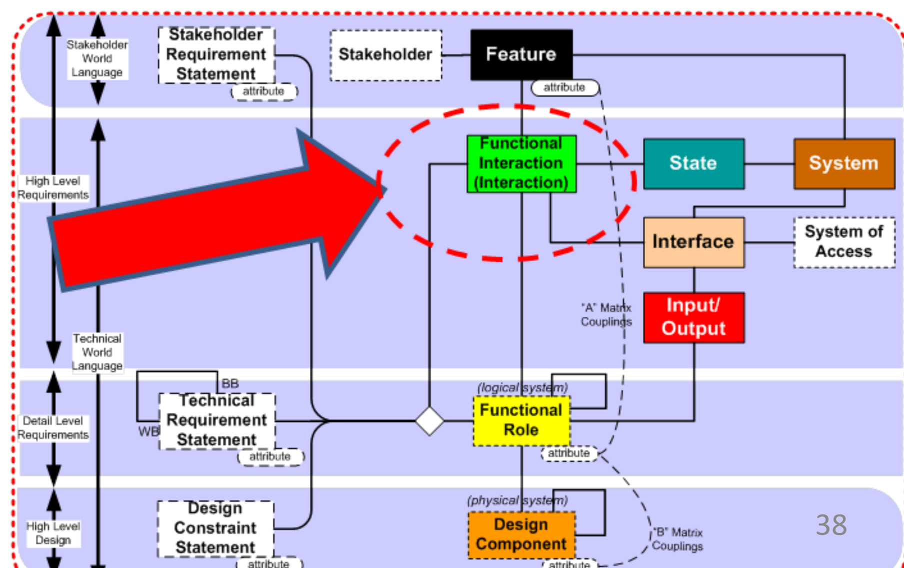




Some Elements of S*Models

- **<u>Interactions</u>** are at the heart of the S*Metamodel, and SE.
- This approach defines a <u>System</u> as a collection of interacting components:
  - By "interact" we mean exchanges of force, energy, mass, or information, resulting in changes of state.
  - Virtually all the laws of the physical sciences uncovered during the last 300 years are expressed in terms of such Interactions.
  - All behavior occurs as interacting Functional Roles.
  - Interactions are central to SE.

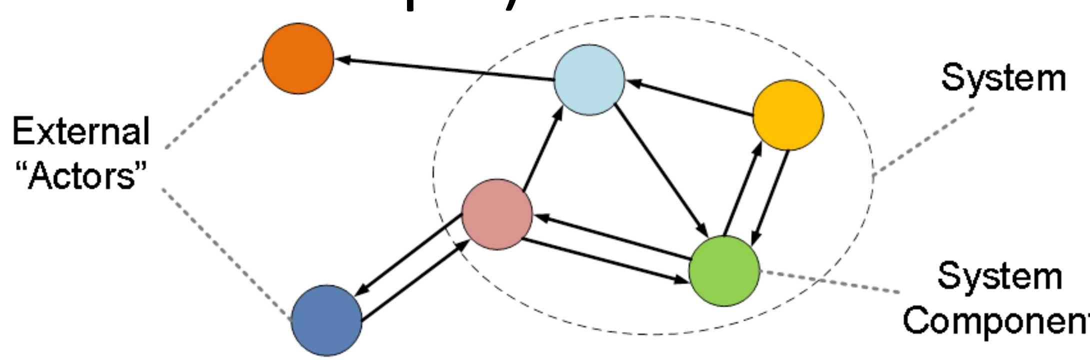




The System Phenomenon

- All "phenomena" of the hard sciences are instances of the **<u>System Phenomenon</u>**:
  - behavior emergent from the interaction of behaviors (phenomena themselves) at a level lower
- In each such case, the emergent interaction-based behavior of the larger system is a stationary path of the action integral (Hamilton's Principle):

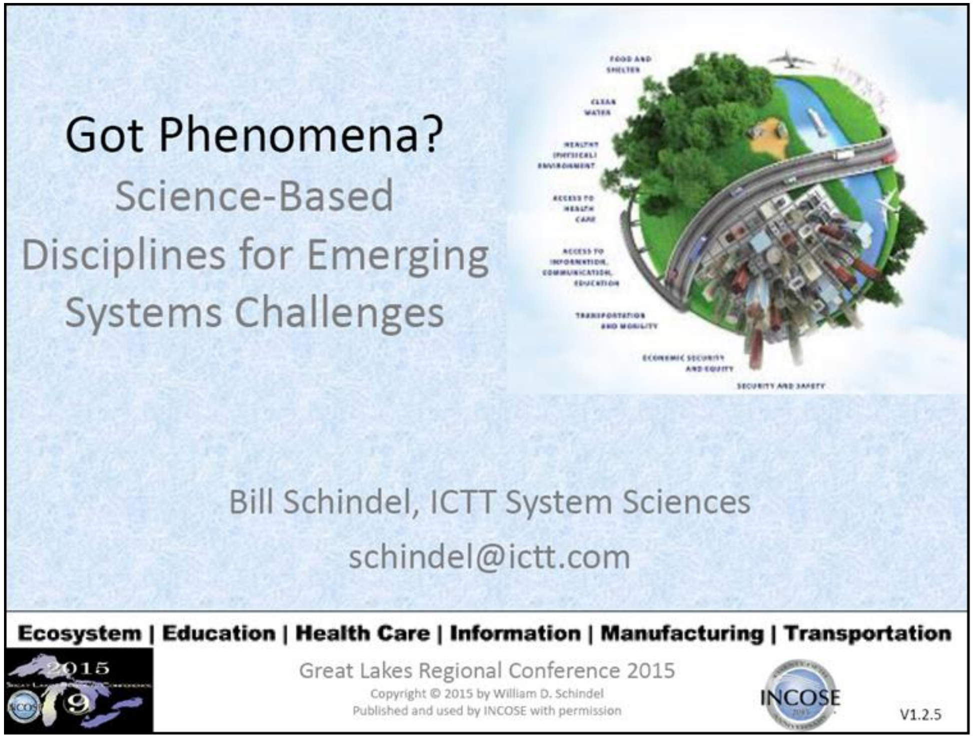
$$\mathcal{S} = \int_{t_1}^{t_2} L(x, \dot{x}, t)\, dt$$

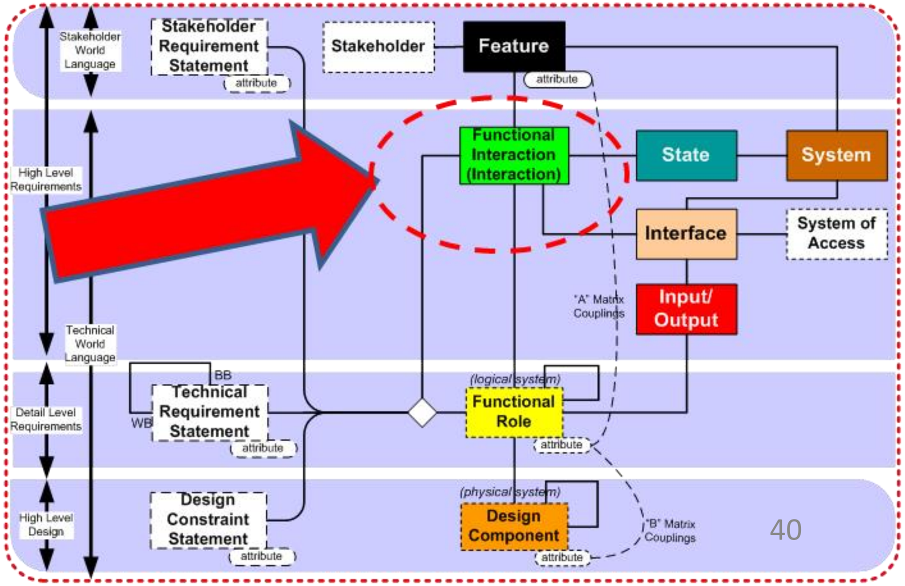


- The resulting equations of motion (or if not solvable, empirically observed paths) provide "physical laws" subject to scientific verification.

Some Elements of S*Models

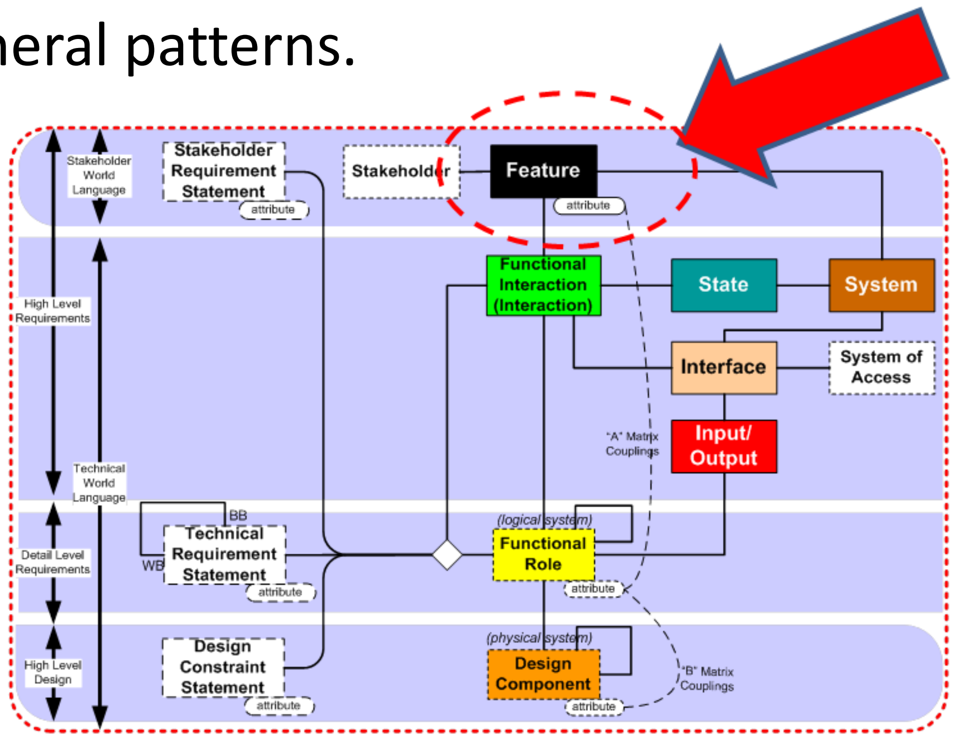
- We will be interested in representing what can learn about patterns of emergent behavior of Systems of Systems.





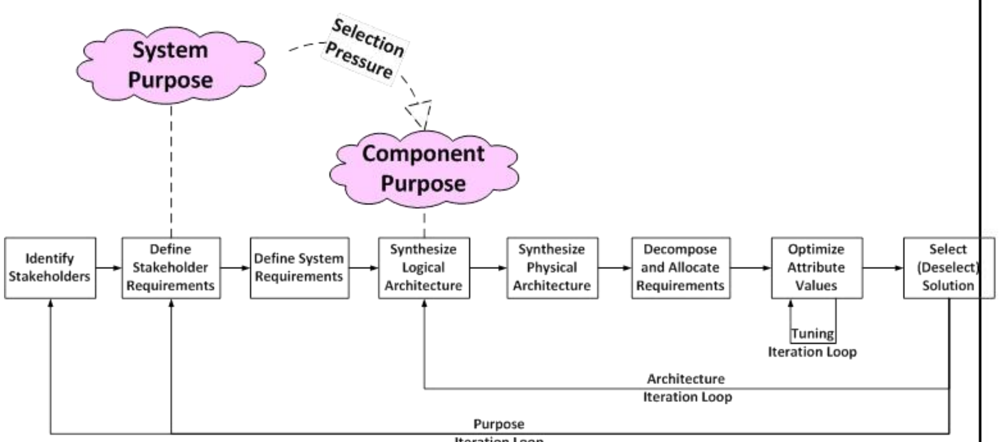
Some Elements of S*Models

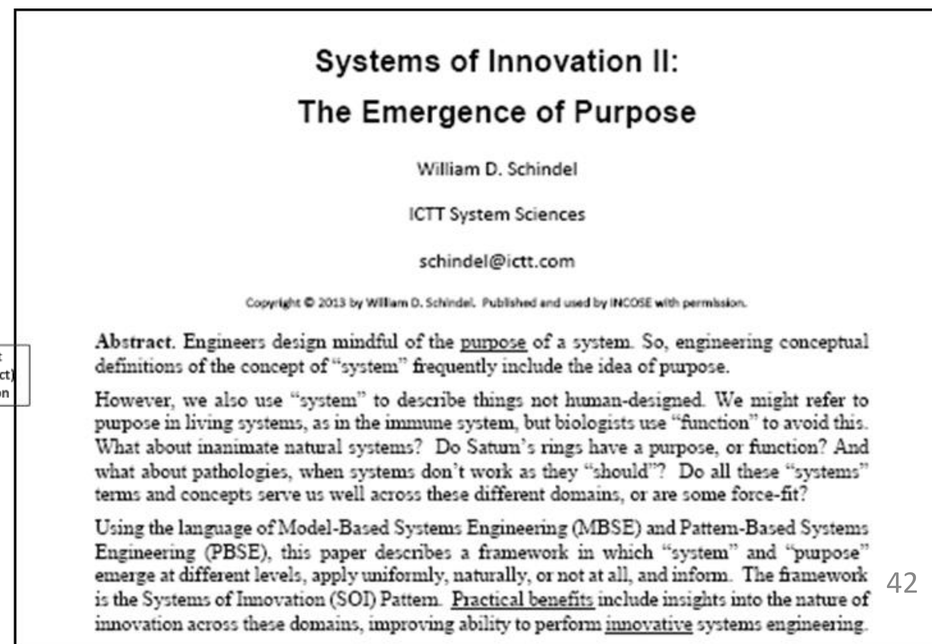
- **<u>Features</u>** express emergent, selectable value (fitness) as expressed by selection mechanisms (market, cognitive, biological, other):
  - When we want to represent fitness, goodness of performance, or other expressions of value in SoS, then Features, parameterized by Feature Attributes, model that fitness space.
  - Features later provide a natural basis for configuration of specific cases of general patterns.



Some Elements of S*Models

- **<u>Features</u>** express emergent, selectable value (fitness) as expressed by selection mechanisms (market, cognitive, biological, other):
  - The <u>purpose</u> of a system is the functional role for which it is selected, or the role it performs in a (larger) selected system.
  - System purpose (function in biology) emerges over time, even in human-designed systems.
  - SoS Patterns will need a way to express modeled fitness or value.





Systems of Innovation II: The Emergence of Purpose

William D. Schindel

ICTT System Sciences

schindel@ictt.com

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

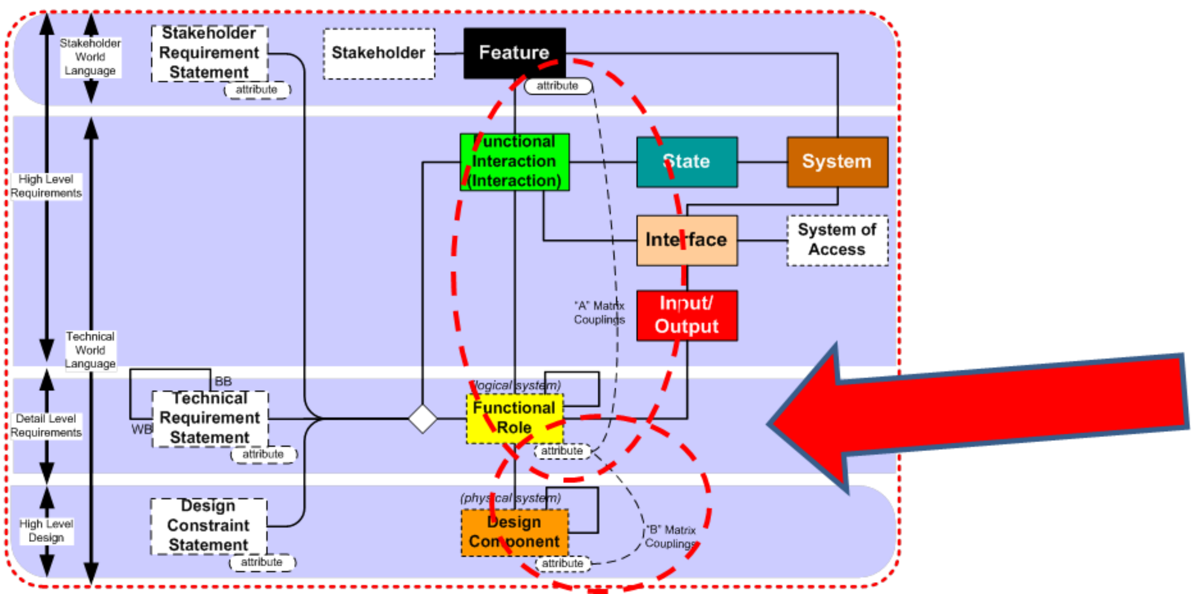
Abstract. Engineers design mindful of the <u>purpose</u> of a system. So, engineering conceptual definitions of the concept of "system" frequently include the idea of purpose.

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. <u>Practical benefits</u> include insights into the nature of innovation across these domains, improving ability to perform <u>innovative</u> systems engineering.

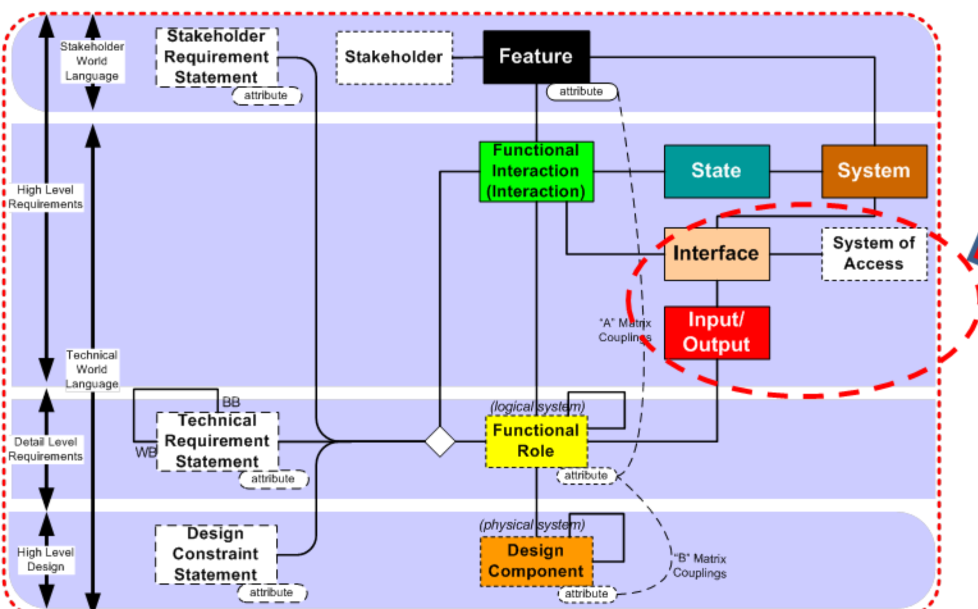
Some Elements of S*Models

- **<u>Attribute Couplings</u>** identify quantitative relationships between quantitative attributes (parametric relationships):
  - <u>A Couplings</u>: Express how fitness or value is coupled to technical behavior.
  - <u>B Couplings</u>: Express how technical behavior is coupled to chosen components.
- We are interested in representing what we can learn about these couplings for Systems of Systems—including emergent attributes.

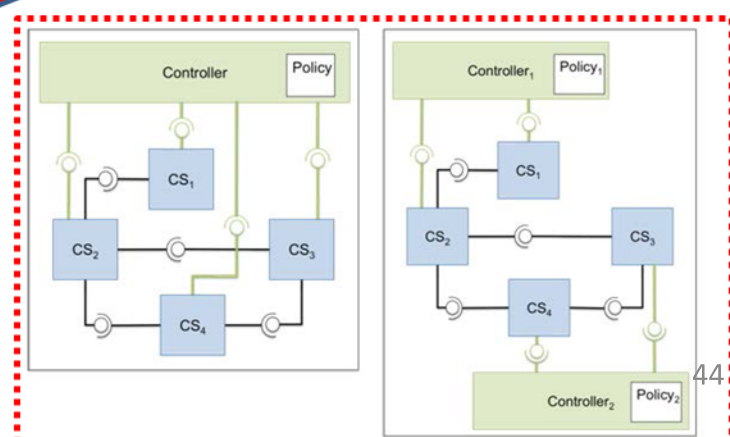


Some Elements of S*Models

- **<u>Interfaces</u>** describe behavior at system boundaries or between components:
  - What Interfaces exist?
  - What **<u>Input-Outputs</u>** are exchanged at an Interface?
  - What is the behavior at an Interface? (Interactions)
  - What is the **<u>System of Access</u>** at an Interface?
- We are interested in modeling interfaces for Systems of Systems.

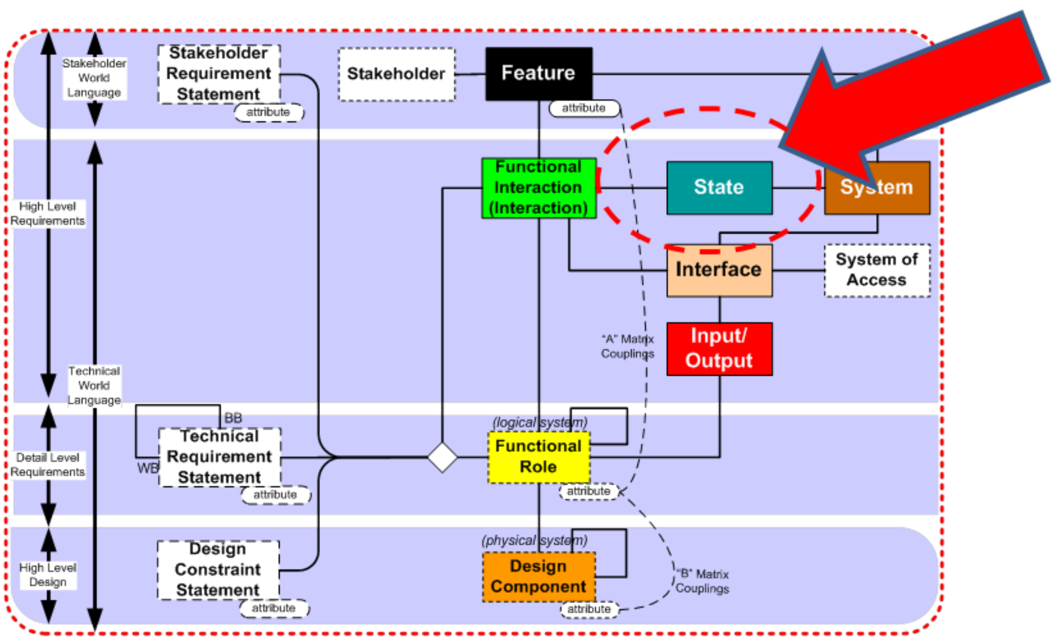


Extract from an SoS Pattern



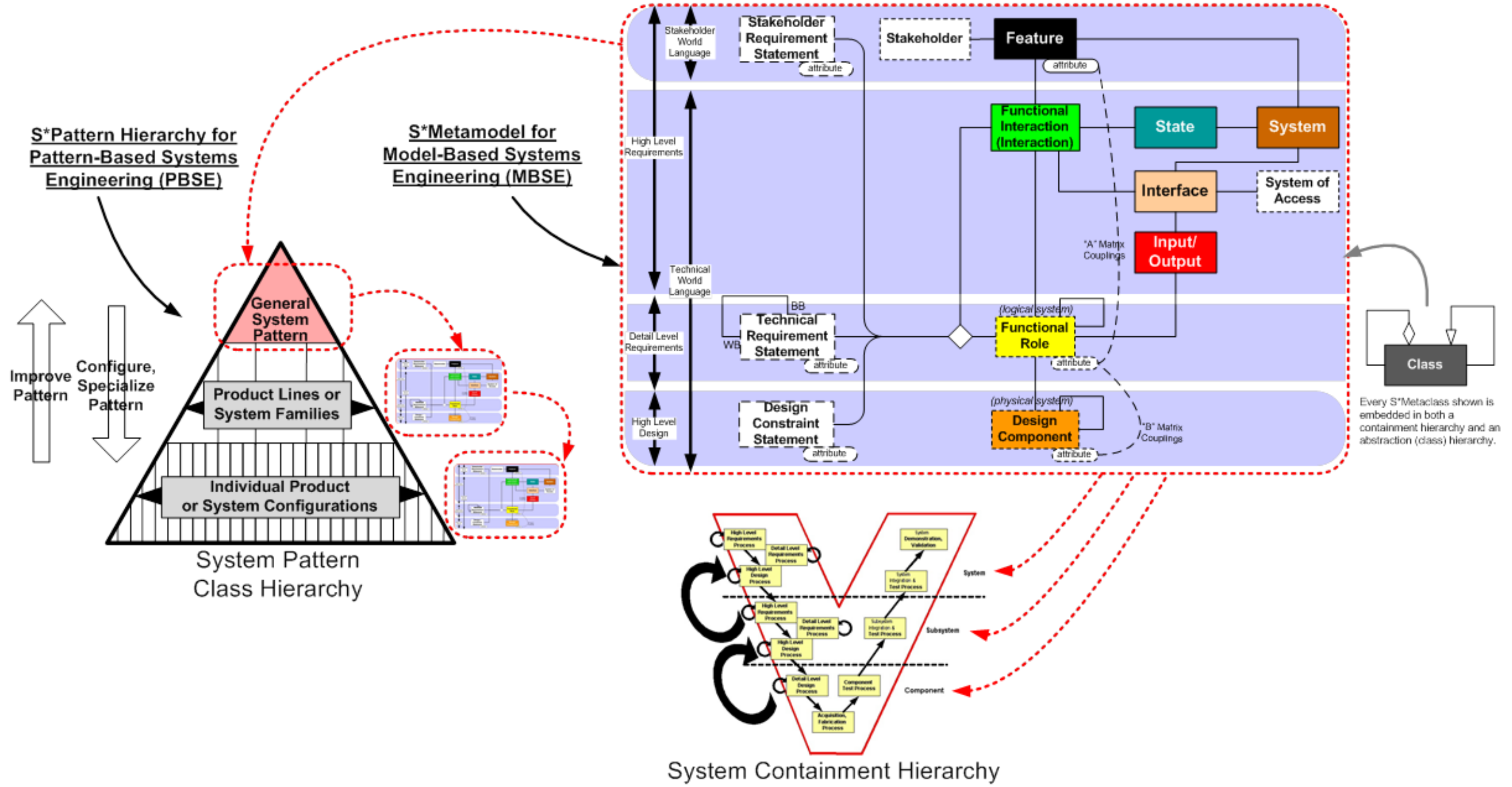
Some Elements of S*Models

- **<u>States, Modes, and Phases</u>** describe conditions or situations of systems:
  - In different system states, system behavior may be different, by intent or nature, or need to be different.
  - We are interested in the states of a System of Systems, its environment, or constituent systems.



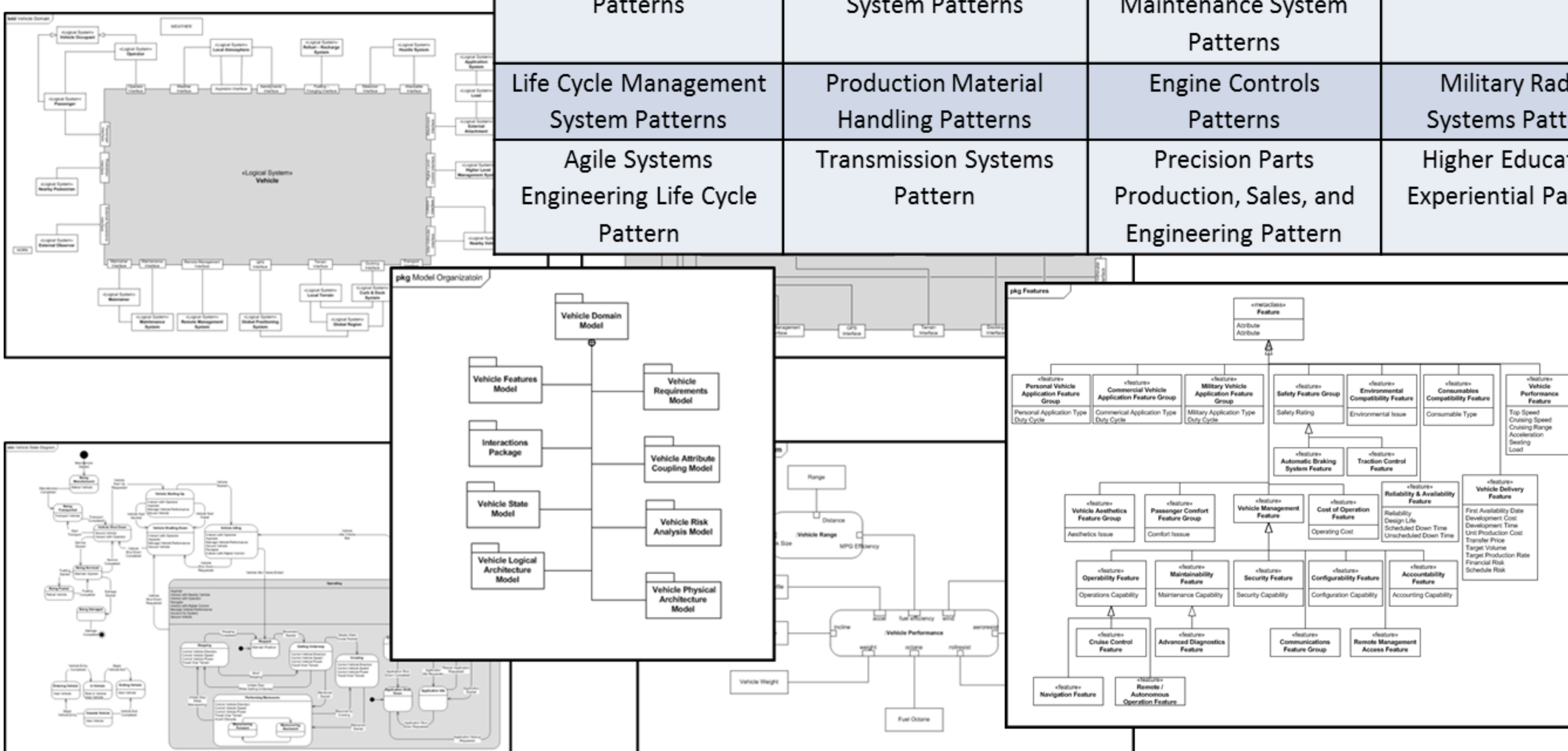
What are S*Patterns?

- S*Patterns are configurable, re-usable S*Models of families of systems:
  - Architectural Frameworks, Product Lines, Platforms, etc.
  - A form of model compression.
  - Using the elements of the S*Metamodel.



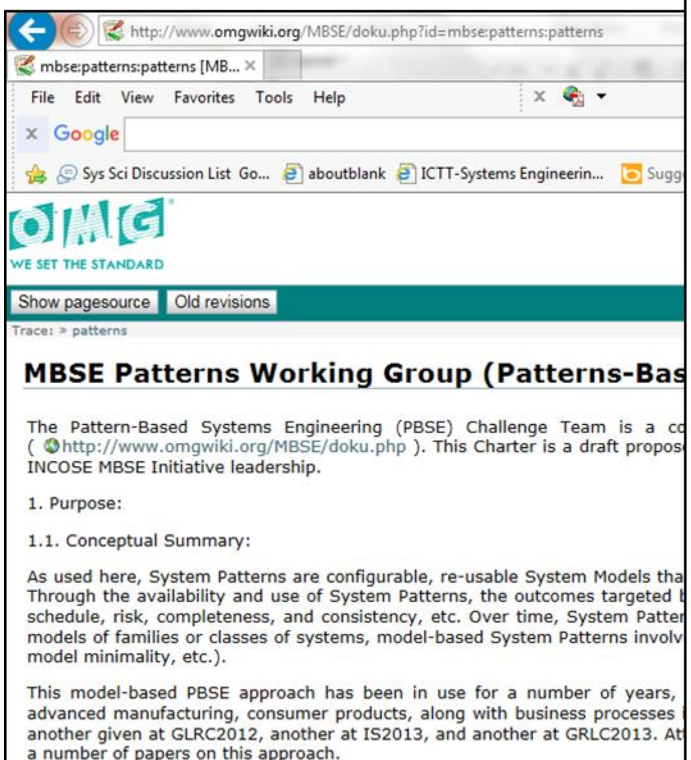
S*Patterns have been applied across many domains, over several decades

Medical Devices Patterns	Construction Equipment Patterns	Commercial Vehicle Patterns	Space Tourism Pattern
Manufacturing Process Patterns	Vision System Patterns	Packaging Systems Patterns	Lawnmower Product Line Pattern
Embedded Intelligence Patterns	Systems of Innovation (SOI) Pattern	Consumer Packaged Goods Patterns (Multiple)	Orbital Satellite Pattern
Product Service System Patterns	Product Distribution System Patterns	Plant Operations & Maintenance System Patterns	Oil Filter Pattern
Life Cycle Management System Patterns	Production Material Handling Patterns	Engine Controls Patterns	Military Radio Systems Pattern
Agile Systems Engineering Life Cycle Pattern	Transmission Systems Pattern	Precision Parts Production, Sales, and Engineering Pattern	Higher Education Experiential Pattern



What are S*Patterns?

- The basis of Pattern-Based Systems Engineering (PBSE), an extension of MBSE:
  - The focus of the INCOSE Patterns Working Group, an INCOSE/OMG MBSE Initiative Challenge Team



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

mbse:patterns:patterns [MB...

File Edit View Favorites Tools Help

Google

Sys Sci Discussion List Go... aboutblank ICTT-Systems Engineerin... Sugg

WE SET THE STANDARD

Show pagesource Old revisions

Trace: » patterns

MBSE Patterns Working Group (Patterns-Bas

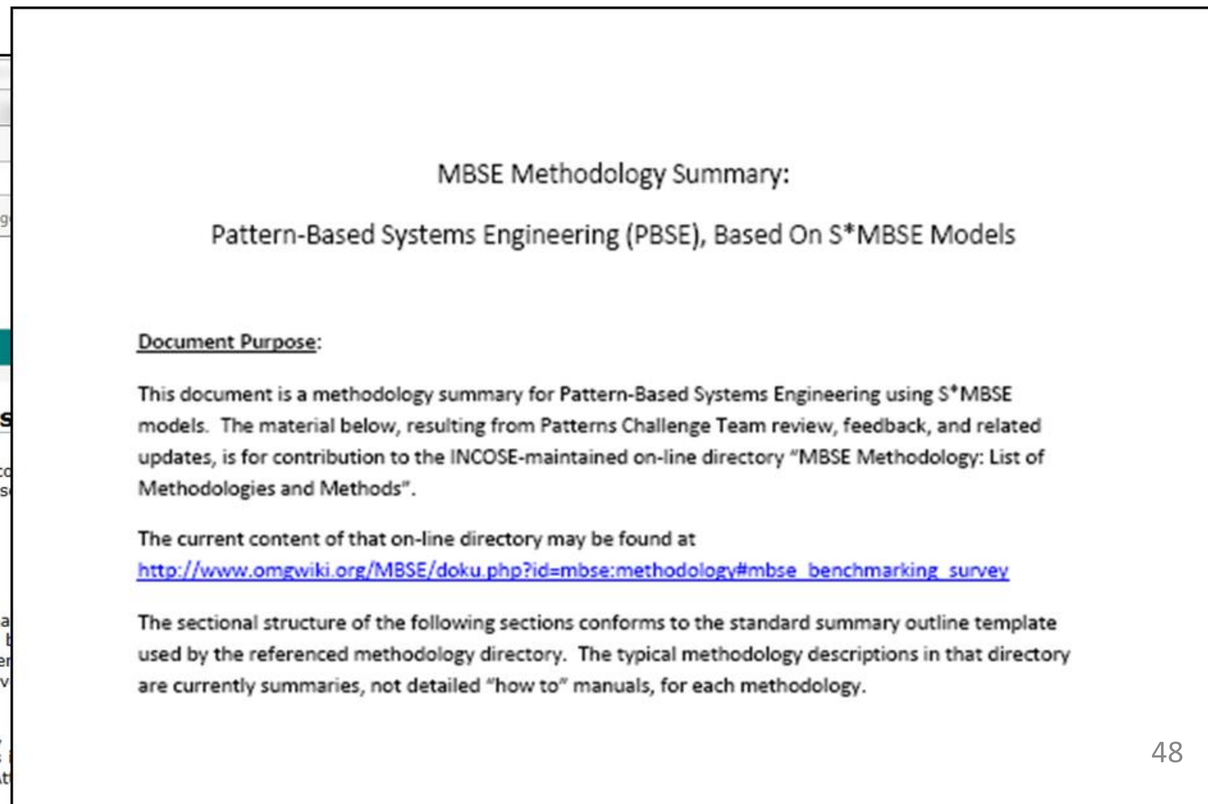
The Pattern-Based Systems Engineering (PBSE) Challenge Team is a co (http://www.omgwiki.org/MBSE/doku.php). This Charter is a draft propos INCOSE MBSE Initiative leadership.

1. Purpose:

1.1. Conceptual Summary:

As used here, System Patterns are configurable, re-usable System Models tha Through the availability and use of System Patterns, the outcomes targeted b schedule, risk, completeness, and consistency, etc. Over time, System Patter models of families or classes of systems, model-based System Patterns involv model minimality, etc.).

This model-based PBSE approach has been in use for a number of years, advanced manufacturing, consumer products, along with business processes i another given at GLRC2012, another at IS2013, and another at GRLC2013. At a number of papers on this approach.



MBSE Methodology Summary:

Pattern-Based Systems Engineering (PBSE), Based On S*MBSE Models

Document Purpose:

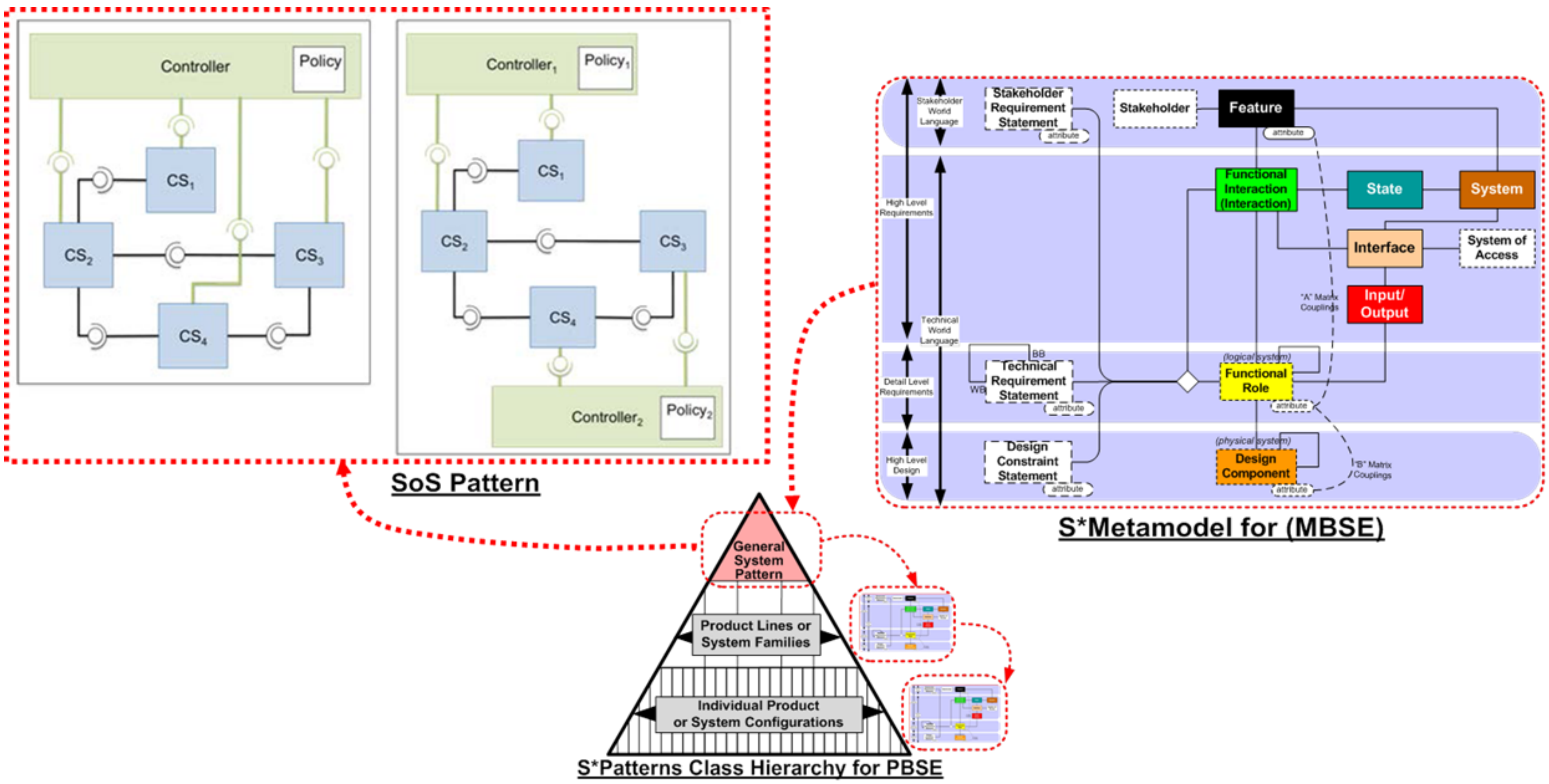
This document is a methodology summary for Pattern-Based Systems Engineering using S*MBSE models. The material below, resulting from Patterns Challenge Team review, feedback, and related updates, is for contribution to the INCOSE-maintained on-line directory "MBSE Methodology: List of Methodologies and Methods".

The current content of that on-line directory may be found at http://www.omgwiki.org/MBSE/doku.php?id=mbse:methodology#mbse_benchmarking_survey

The sectional structure of the following sections conforms to the standard summary outline template used by the referenced methodology directory. The typical methodology descriptions in that directory are currently summaries, not detailed "how to" manuals, for each methodology.

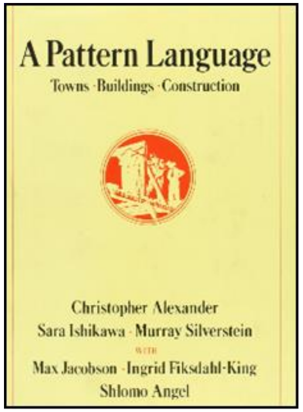
What are S*Patterns?

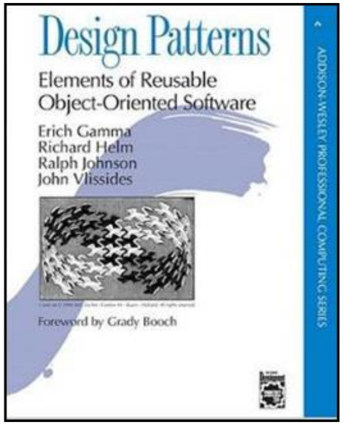
- Connecting our two Working Groups:
  - We will show that SoS Patterns can be expressed as S*Patterns, and . . .
  - They can add valuable insights about Systems of Systems.

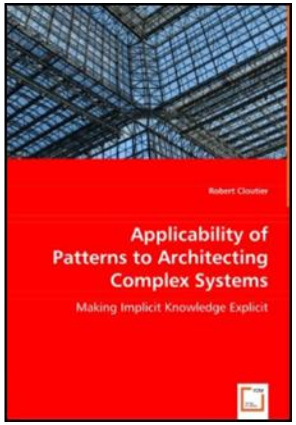


What are S*Patterns?

- The term "pattern" appears repeatedly in the history of design, such as civil architecture, software design, and systems engineering:





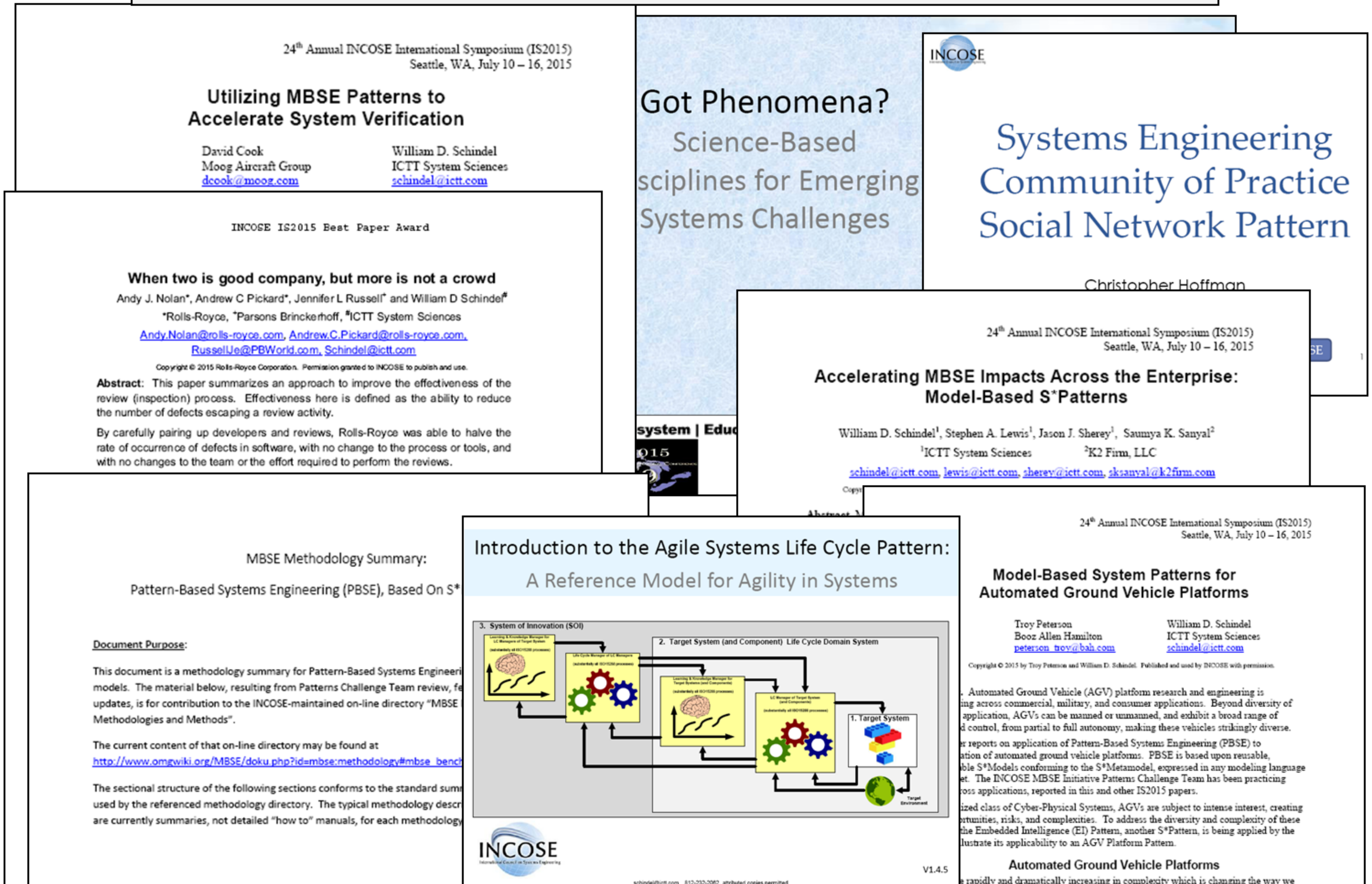


- Those "patterns" represent regularities that repeat, modulo some variable aspects, across different instances in space, time, and other dimensions.
- However, when we refer to "patterns" in the Patterns WG, we mean the use of <u>S*Patterns</u>.

What are S*Patterns?

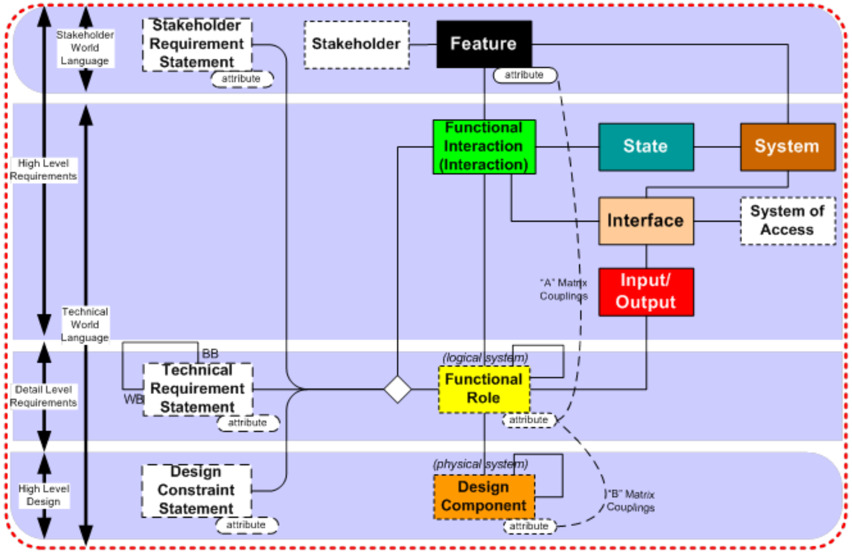
- S*Patterns are model-based (not all historical "patterns" are expressed as MBSE models).
- S*Patterns conform to the S*Metamodel—as a minimal reference model of essential engineering information.
- S*Patterns are embedded in modeled concepts about physical interactions that are the basis of physical laws of the hard sciences emerging over the last 300 years.
- S*Patterns are about "whole systems" (historical "patterns" were sometimes about parts of systems).

Some Pattern-Related Publications by Patterns Working Group Members



What do S*Patterns bring to MBSE representations of Systems of Systems?

- Recurring patterns in Systems of Systems are about more than architectural patterns alone.
- Even though we might not need to model a lot of detail about an SoS, additional insights may be learned about SoS:
  - Interactions
  - Interfaces
  - Fitness space (Features)
  - States (modes, phases)
  - Attribute (parametric) Couplings
  - Simulatable behavior
  - Selection and Evolution



- Gestalt Rules for Patterns
- Including model capture of observations about SoS's that may already be in prose discussion form.

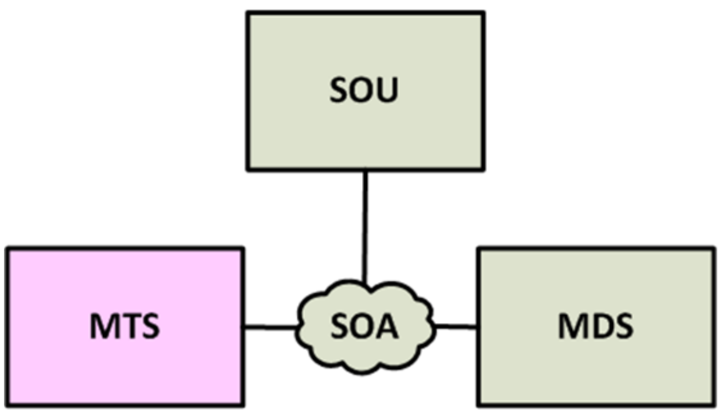
Excerpts from some S*Patterns describing Systems of Systems

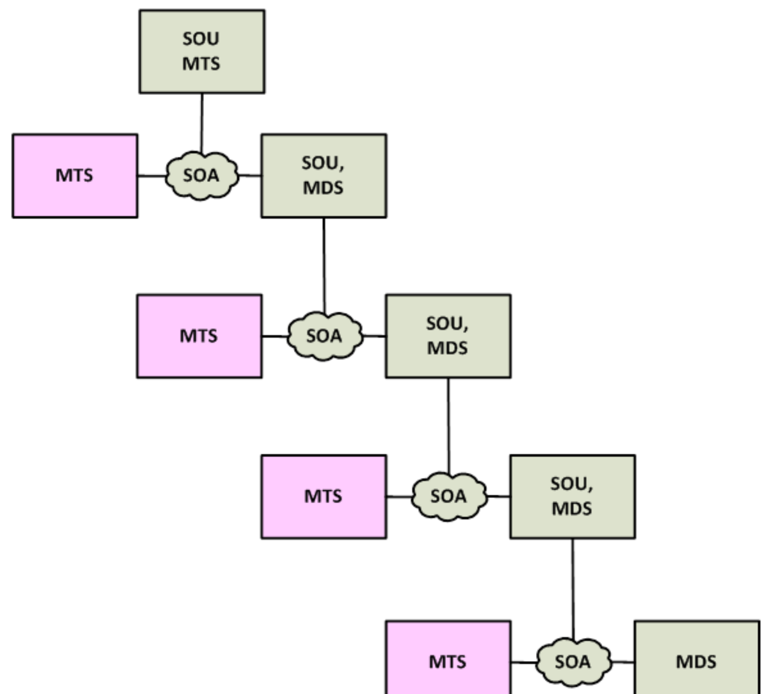
- The Embedded Intelligence (EI) S*Pattern
- The Agile Systems Engineering Life Cycle Management (ASELCM) S*Pattern
- S*Patterns for Progressively Emergent System of Systems

The Embedded Intelligence (EI) S*Pattern

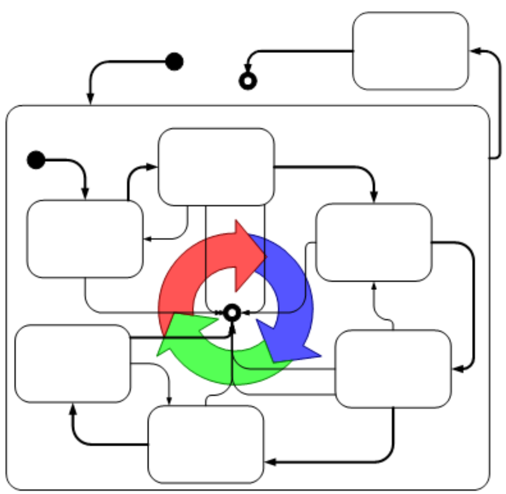
- The **Embedded Intelligence Pattern** (sometimes called the **Systems Management Pattern**) describes an emergent "fractal" pattern of system structure and behavior concerned with "intelligence" in systems of all types—engineered, natural, etc.
- It is found in large enterprises, across multiple connected enterprises (markets, supply chains, etc.), lower-level embedded networked cyber-physical systems, and the Internet of Things -- Factories, supply networks, interacting vehicles, interacting military platforms – and is typically found distributed across systems.
- The "intelligence" embedded may be human, automated, or (frequently) hybrid.
- No matter what the purpose of the intelligence, it can be projected into the EI Pattern's framework of "management", meaning ISO System Management Functional Areas (SMFAs):
  - Performance Management (classical regulatory control)
  - Configuration Management
  - Fault Management
  - Security Management
  - Accounting Management

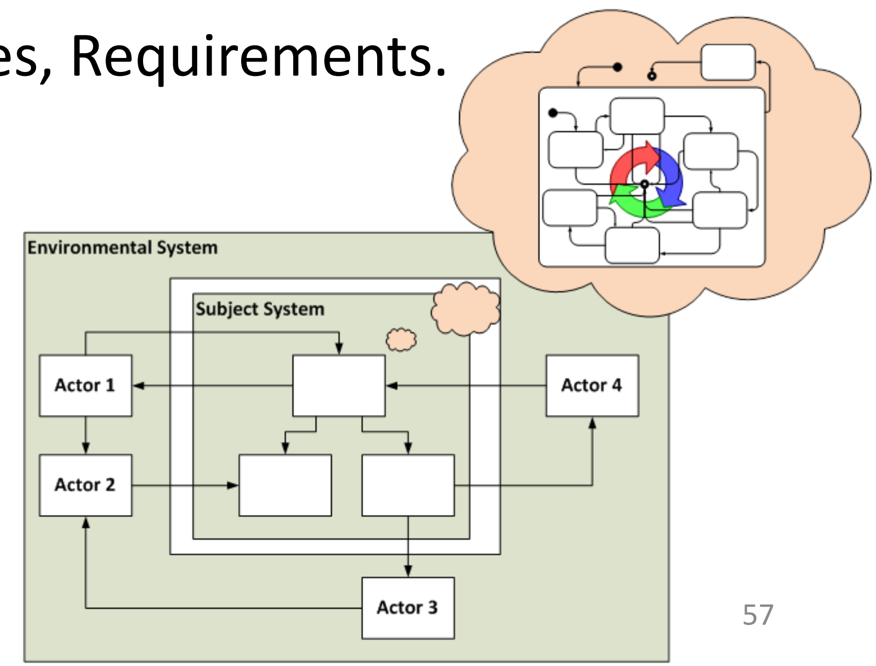
- Embedded Intelligence Pattern Functional Roles:
  - **Managed System (MDS):** The system whose performance, configuration, faults, security, or accounting is to be managed.
  - **Management System (MTS):** The system providing management.
  - **System of Users (SOU):** Consumers of managed results.
  - **System of Access (SOA):** Provides instrumentation, actuation, networking, connectivity, or other interaction enabling media.
- A distributed, hybrid, fractal hierarchy typically emerges:





- EI Pattern States, Interactions, Interfaces, Requirements:
  - Cyclic situation resolution (regulatory) state models appear in each of the five SMFAs, reflecting the general regulatory role of intelligence in all cases.
  - This includes a framework for Attention Management, reflecting the application of limited system resources to varying (sometimes over capacity) external demand situations.
  - It provides a Situation Awareness framework, including pattern structure for Situationally Aware Systems.
  - Patterns of Interactions, Interfaces, Requirements.



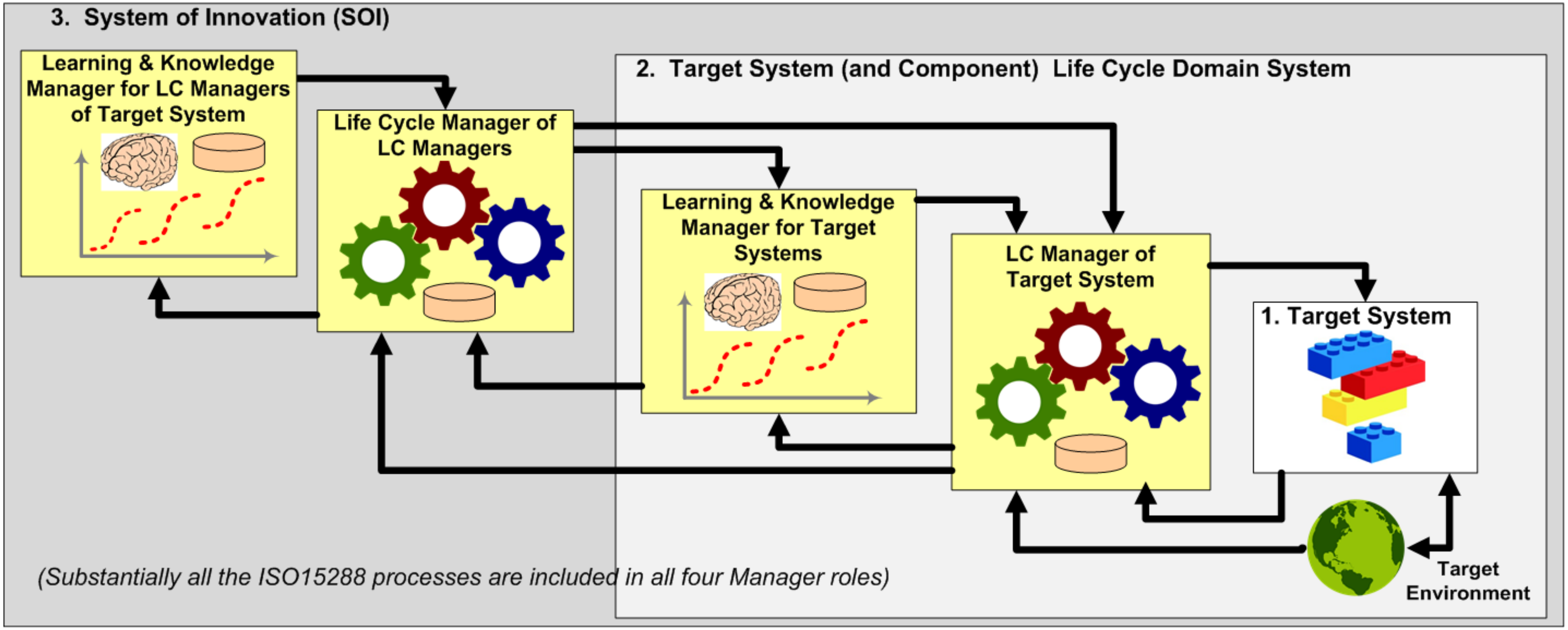


The Agile Systems Engineering Life Cycle Management (ASELCM) S*Pattern

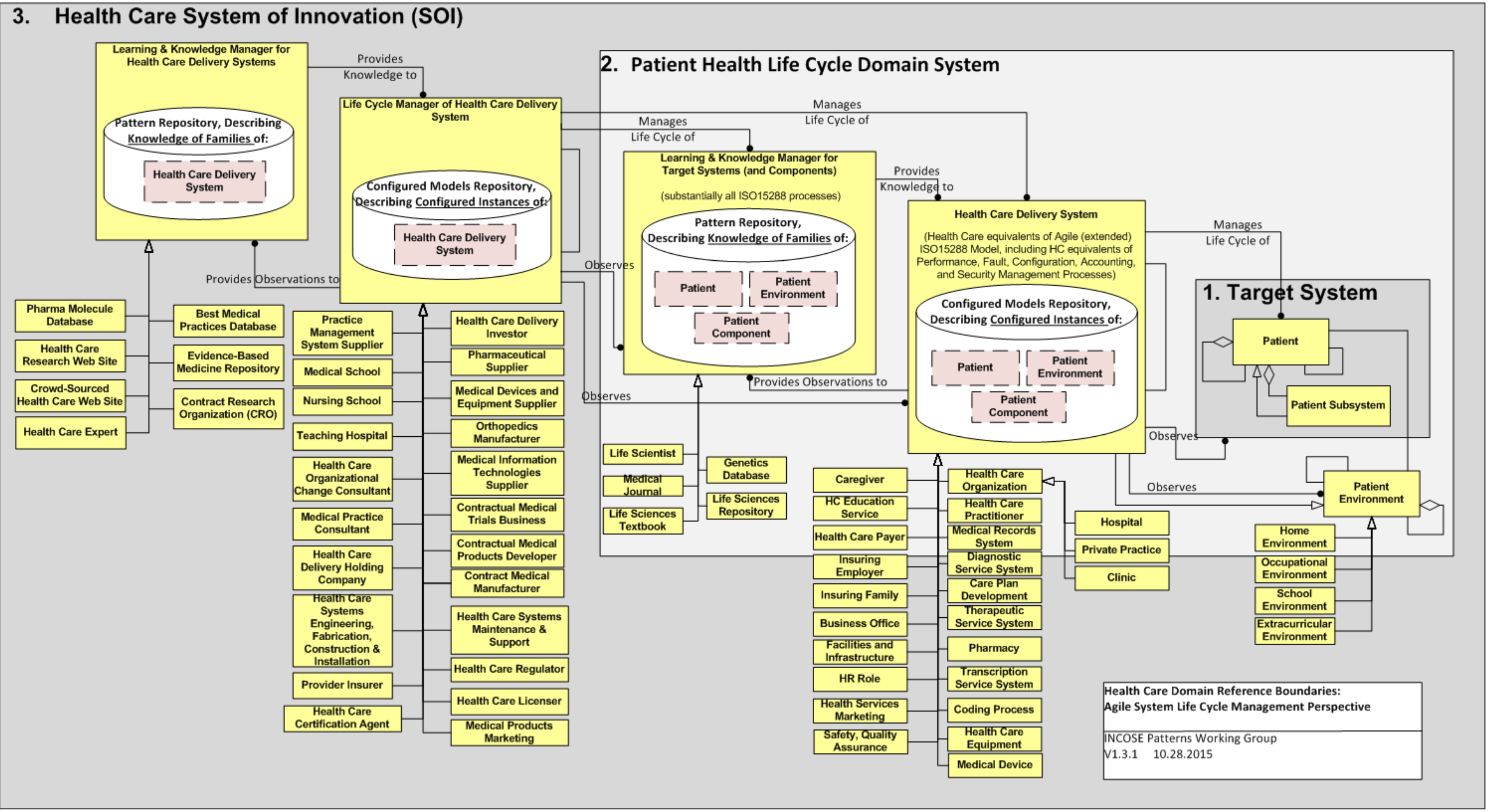
- The ASELCM Pattern describes recurring patterns of structure and behavior in systems of agile innovation and life cycle management—including the environment or ecology in which they operate.
  - This pattern is being constructed by the INCOSE ASELCM Discovery Project, based on 2015-2016 host site workshops investigating current practices in Agile Systems Engineering.
  - This INCOSE Agile SE WG Project is supported by the Patterns WG.
- The ASELCM Pattern uses content from both the EI Pattern and the Systems of Innovation (SOI) Pattern.
  - The Systems of Innovation (SOI) Pattern describes recurring structure and behavior seen in innovation processes, and the ecology in which they operate--whether natural or human-performed.
  - Includes elements from both the living world and ISO 15288.

ASELCM Pattern: Logical Architecture

- System 1: The Target System being innovated or otherwise supported over its life cycle.
- System 2: The Life Cycle Systems (development, manufacturing, support, etc., including operational environment)
- System 3: The System of Innovation for managing and improving System 2.



ASELCM Pattern: Health Care Ecosystem Example

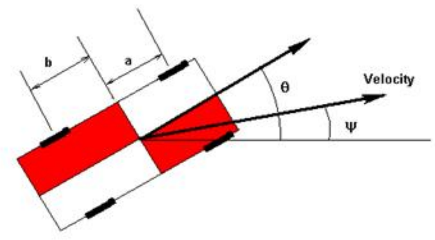


S*Patterns for Progressively Emergent Systems of Systems

- Recent historical examples
- Future examples

Recent Historical Examples

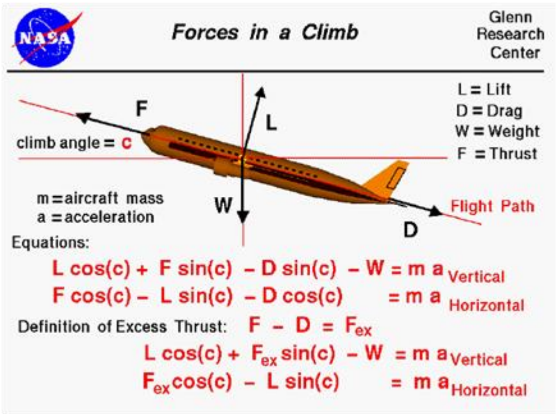
- Ground Vehicles
- Aircraft
- Marine Vessels
- Biological Regulatory Networks

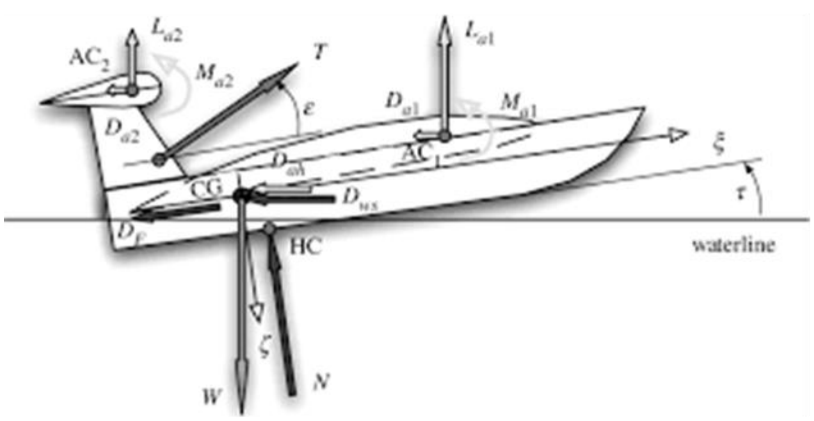


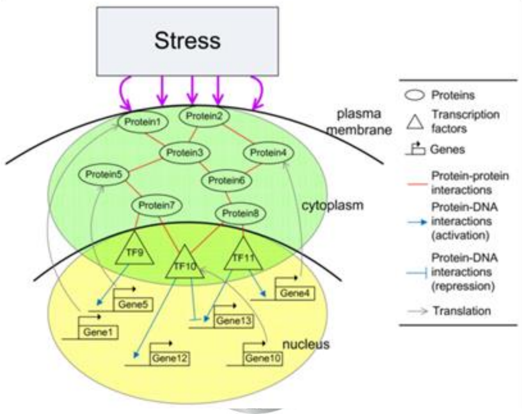
Dynamics of Road Vehicle

Denoting the angular velocity ω, the equations of motion are:

$$\frac{d\omega}{dt} = 2k\frac{(a-b)}{I}(\theta-\psi) - 2k\frac{(a^2+b^2)}{VI}\omega$$
$$\frac{d\theta}{dt} = \omega$$
$$\frac{d\psi}{dt} = \frac{4k}{MV}(\theta-\psi) + 2k\frac{(b-a)}{MV^2}\omega$$

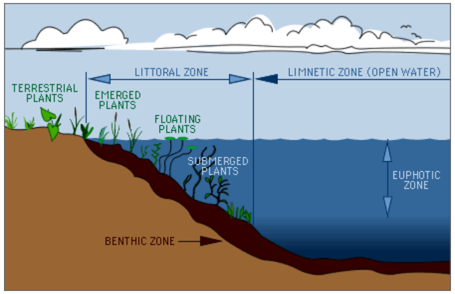


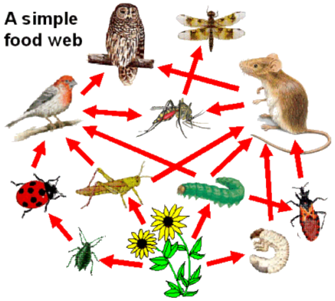


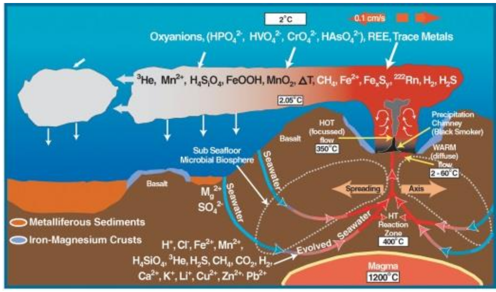


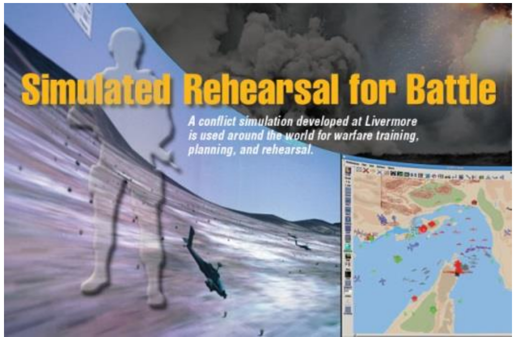
Future Applications

- Utility and other distribution networks
- Biological organisms and ecologies
- Market systems and economies
- Health care delivery, other societal services
- Systems of conflict
- Agile innovation









Teams Break-Out Session

Architectural Modelling Patterns for Systems of Systems

Claire Ingram, Richard Payne, John Fitzgerald

Newcastle University, UK

These patterns were presented as part of a paper on SoS Architectural Patterns at the INCOSE 25th Anniversary International Symposium 2015, Seattle, WA

Plenary Discussion

-
-
-
-
-
-
-

Systems of Systems References:

1. SoS Working Group web site: http://www.incose.org/ChaptersGroups/WorkingGroups/processes/system-of-systems
2. Maier, M. “Architecting Principles for Systems of Systems.” *Systems Engineering*. 1998. Vol. 1, No. 4: 267-284.
3. Dahmann, J. and Baldwin, K., 2011. “Implications of Systems of Systems on System Design and Engineering”. In *International Conference on Systems of Systems Engineering* (SoSE), 2011.
4. Claire Ingram, Richard Payne, John Fitzgerald, “Architectural Modelling Patterns for Systems of Systems”, *Proc of INCOSE International Symposium*, Seattle, 2015.
5. Dahmann, J, 2014. “Systems of Systems Pain Points”. In *Proceedings of the 24th INCOSE International Symposium.* Henderson, NV, USA. INCOSE, June 2014.

S*Patterns and PBSE References:

1. Patterns Working Group web site: http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns
2. Schindel, W., et al, “MBSE Methodology Summary: Pattern-Based Systems Engineering (PBSE), Based On S*MBSE Models”, V1.5.5A, INCOSE Patterns Working Group, retrieved 2016 from: http://www.omgwiki.org/MBSE/doku.php?id=mbse:pbse
3. Schindel, W., “What Is the Smallest Model of a System?”, *Proc. of the INCOSE 2011 International Symposium*, 2011.
4. Schindel, W., "Got Phenomena? Science-Based Disciplines for Emerging Systems Challenges”, *Proc of INCOSE Great Lakes Regional Conference*, Cleveland, OH, 2015.
5. Peterson, T., and Schindel, W., “Pattern Based Systems Engineering – Leveraging Model Based Systems Engineering for Cyber-Physical Systems”, *Proc. of NDIA GVSETS*, 2014.
6. Schindel, W., “Introduction to the Agile Systems Engineering Life Cycle Pattern”, *Proc of INCOSE Great Lakes Regional Conference*, Cleveland, OH, 2015.