CASE Session-Incremental/Agile Methods: Fit for Demands of Complex Aerospace Systems?

Roundtable Summary: 08 June 2017

AIAA AV2017, Complex Aerospace Systems Exchange (CASE)

Denver, CO

Bill Schindel, ICTT System Sciences

schindel@ictt.com

V1.4.1

Questions Posed

- Are the methods compatible or incompatible with aerospace?
- What relation do the methods have to systems complexity?
- Are the methods needed by aerospace? What problems are we solving? Has something changed?
- Are the methods already practiced by aerospace? Old hat or new?
- What is the method? Examples? Successes, Problems?
- When is the method a good fit? When is it not a good fit?
- How are these methods different from agile software approaches, if at all?
- Other related questions that need increased exposure?

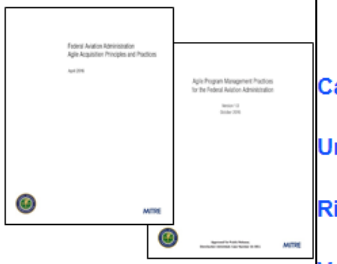
Avinash Pinto, MITRE/FAA Proj.
avinash@mitre.org

Rick Dove, Paradigm Shift, Int'l
dove@parshift.com

Jimmie McEver, JHU APL
jimmie.mcever@jhuapl.edu

Context

- Collaborated with Federal Aviation Administration (FAA) to enhance acquisition outcomes
  - Accelerate capability deployment
  - Ensure greater user acceptance
  - Improve operational value
- Provided guidance to acquisition specialists and program managers on executing Agile Methodologies



Available on http://fast.faa.gov -> AMS Building Blocks -> FAA Agile Acquisition

MITRE

Is This Your Problem Space?

CURVE

Internal and external environmental forces that impact project/process/product as systems

Caprice: Unknowable situations.
Unanticipated system-environment change.

Uncertainty: Randomness with unknowable probabilities.
Kinetic and potential forces present in the system.

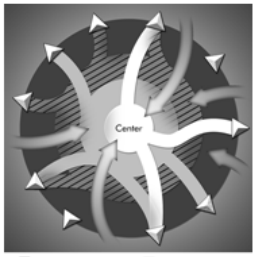
Risk: Randomness with knowable probabilities.
Relevance of current system-dynamics understanding.

Variation: Knowable variables and associated variance ranges.
Temporal excursions on existing behavior attractor.

Evolution: Gradual successive developments.
Experimentation and natural selection at work.

Engineering Itself is a Complex Endeavor

APL



Source: Hayes, Richard E., "It's an Endeavor, Not a Force," International C2 Journal, Vol. 1, No. 1 (2007).

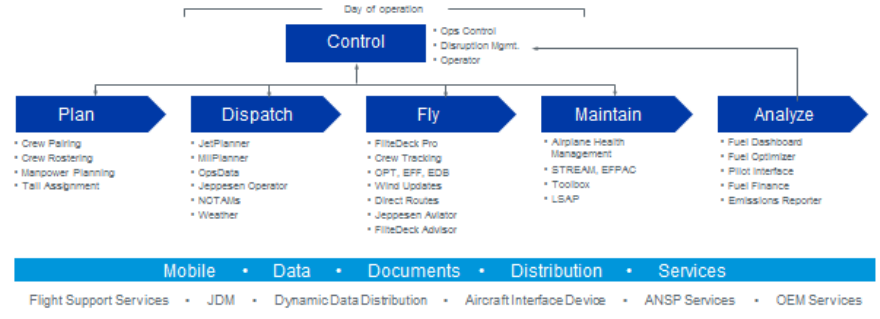
- Increasingly, we conduct systems development via Complex Endeavors*
- Hallmarks of a complex endeavor
  - Has a purpose or set of related purposes
  - Large number of disparate entities whose activities are related to a broad range of effects
  - No single "leader" or commander
  - Individual participants may be working toward different purposes
  - No subset of participants is capable of achieving its relevant goals absent contributions of others
  - Participants may have a variety of relationships with one another
- Boundaries may vary over the life of the endeavor

*Complex endeavors are introduced in Hayes and Alberts, Planning: Complex Endeavors, CCRP Publication Series, 2007.

APL

Boeing Digital Aviation

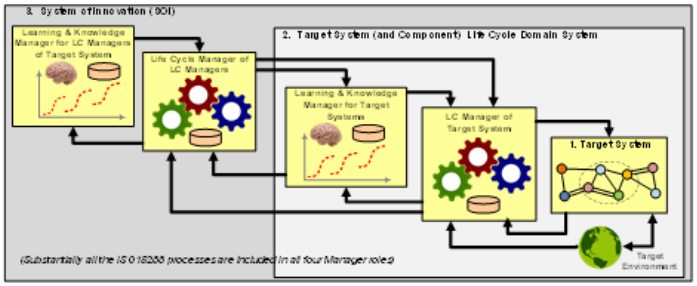
Broadest portfolio of operational efficiency and navigation solutions for aviation



BOEING JEPPESEN

CAS: System of Innovation (SOI) MBSE Pattern

Used in INCOSE Agile Systems Engineering LC Discovery Project



System 1: Target system of interest, to be engineered or improved.

System 2: The environment of (interacting with) S1, *including all the life cycle management systems of S1*, and including *learning about S1*.

System 3: The life cycle management systems for S2, including *learning about S2*.

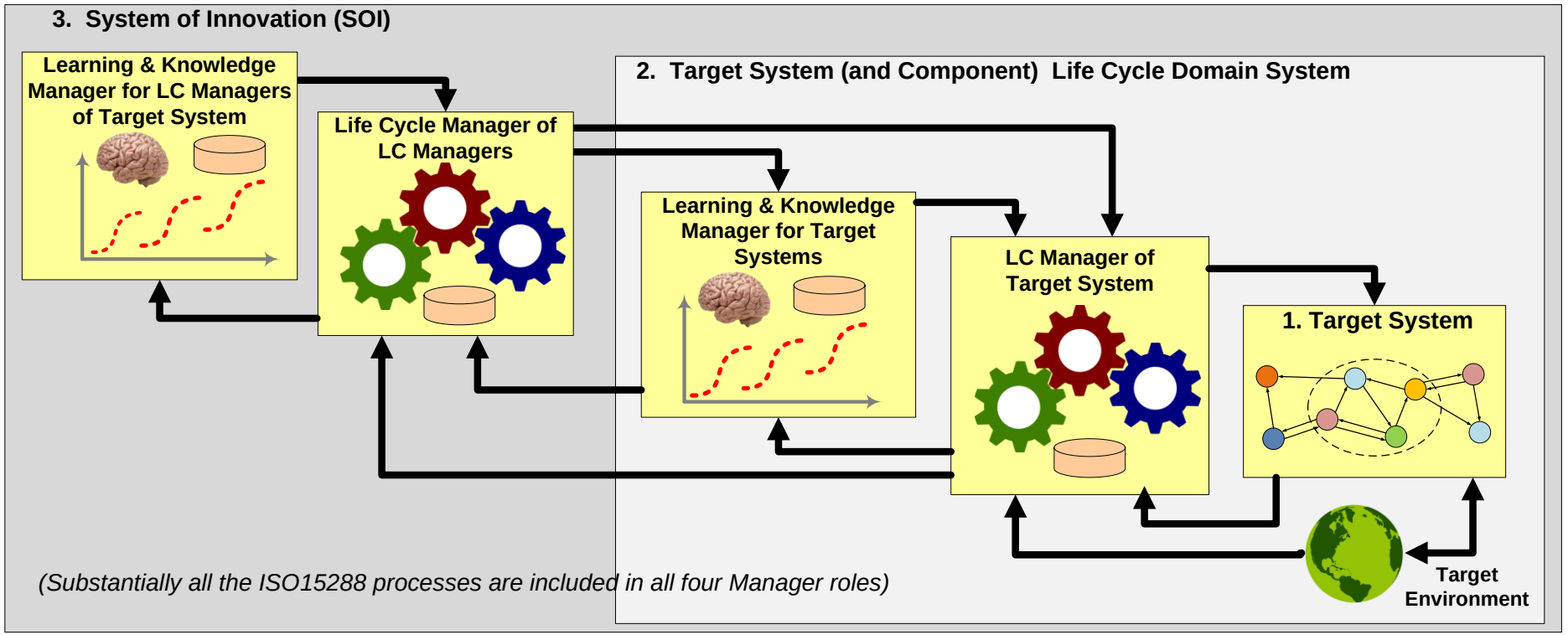
Joe Davis, Boeing/Jeppesen
joe.davis@jeppesen.com

Bill Schindel, ICTT System Sciences
schindel@ictt.com

Incremental / Agile Methods Session Chair: Sophia Bright, Boeing

CAS: System of Innovation (SOI) MBSE Pattern

Used in INCOSE Agile Systems Engineering LC Discovery Project



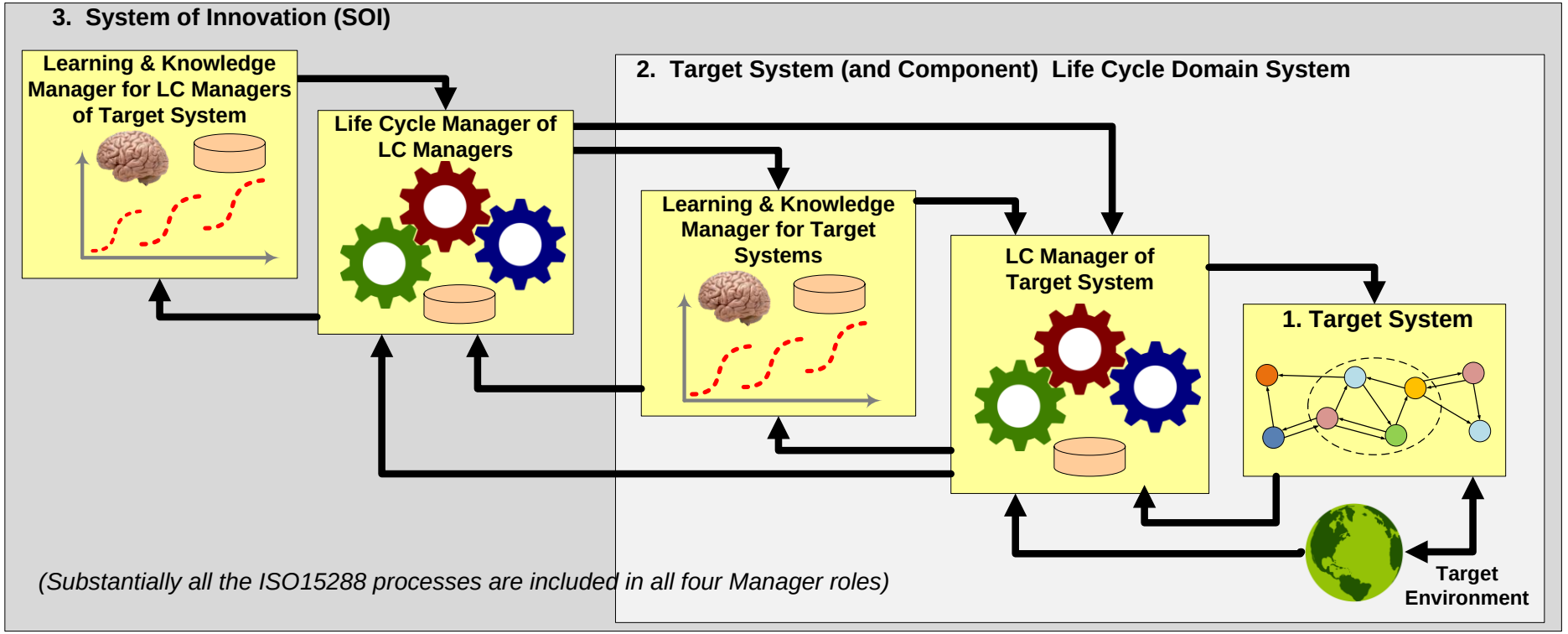
System 1: Target system of interest, to be engineered or improved.

System 2: The environment of (interacting with) S1, *including all the life cycle management systems of S1*, and including *learning about S1*.

System 3: The life cycle management systems for S2, including *learning about S2*.

CAS: System of Innovation (SOI) MBSE Pattern

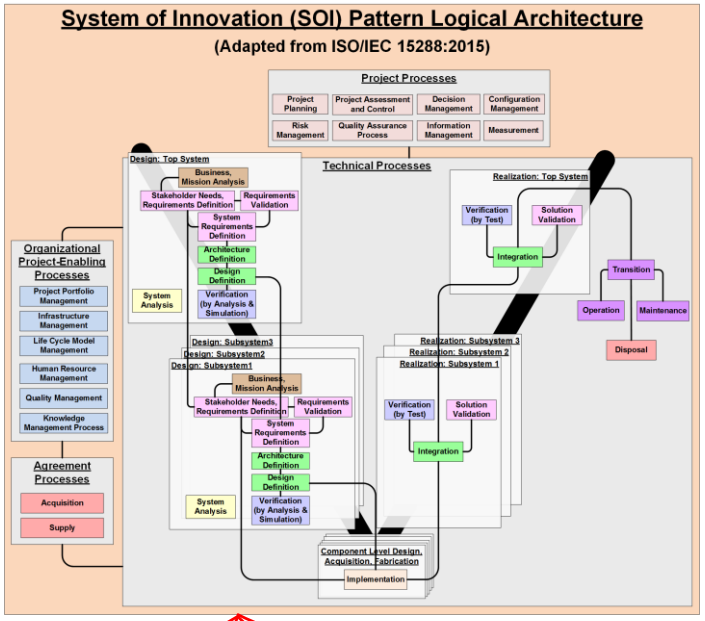
Used in INCOSE Agile Systems Engineering LC Discovery Project

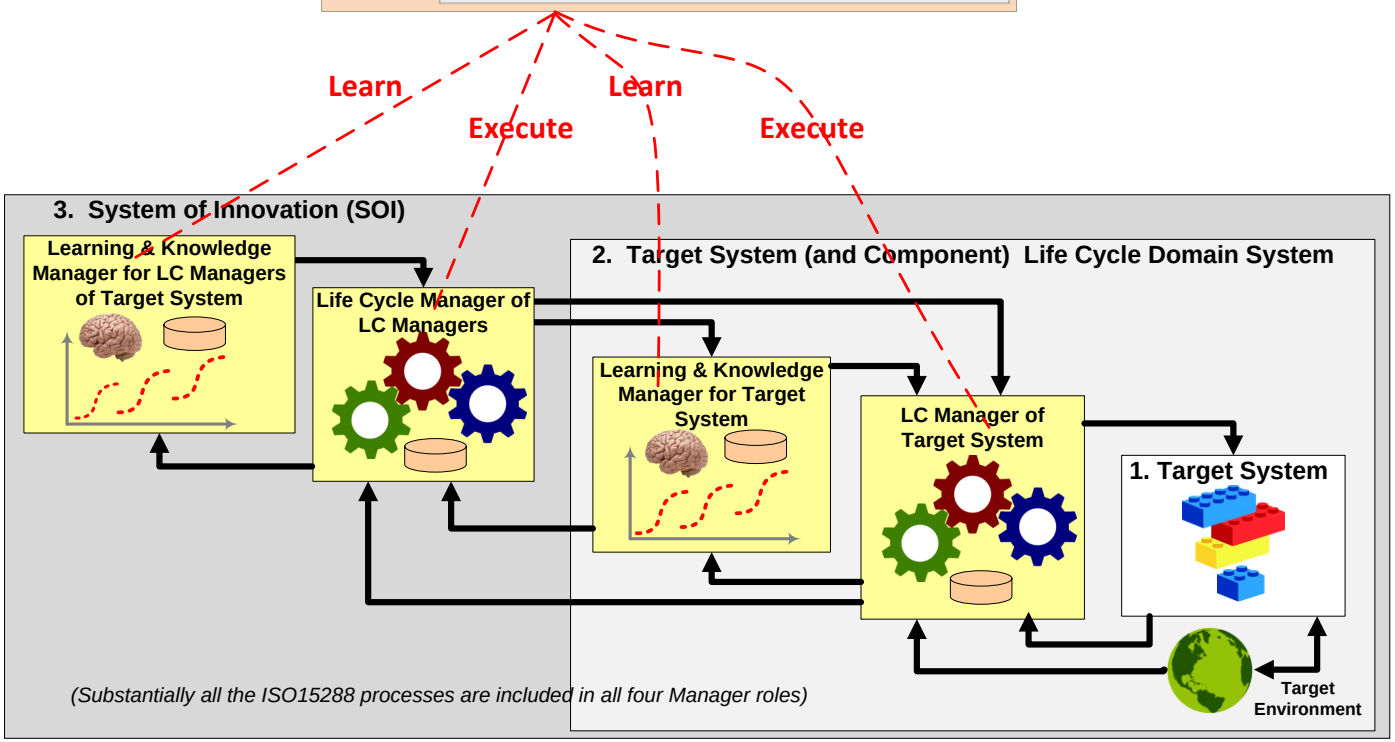


Challenges which encourage Learning about Systems 1 and 2:

Uncertainty about unfolding of future state or configuration of System 1, System 2, and risk of stakeholder impacts caused by that progression.

ISO 15288 life cycle processes appear 4 times (even if not recognized)

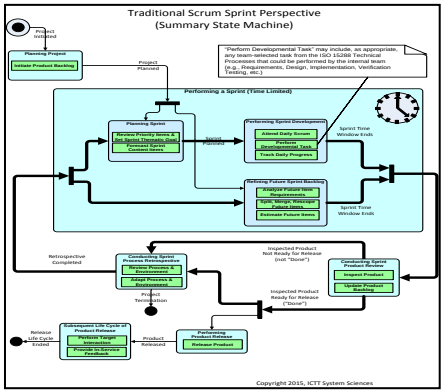




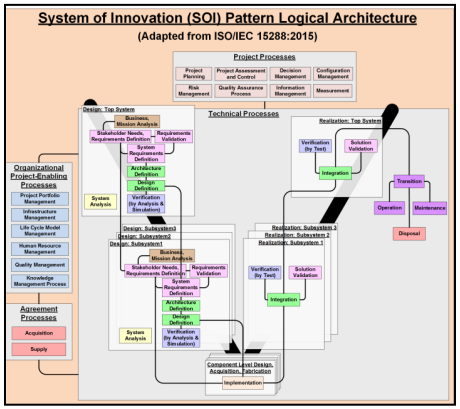
Four different representations of the same underlying reality:

1. The Scrum Pattern: Emphasizes time-bound outputs and feedback, focusing on processes for *learning from produced outputs and managing uncertainty*
2. The ISO15288 Pattern: Emphasizes types of processes, focusing on *management of processes*
3. The Agile Systems Engineering Life Cycle Model 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)

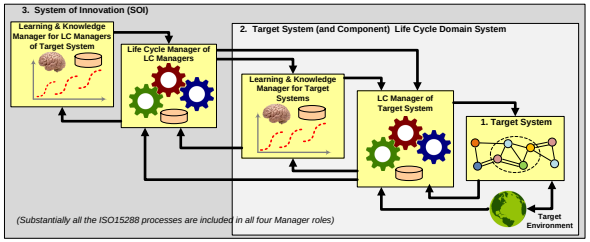
Scrum Pattern



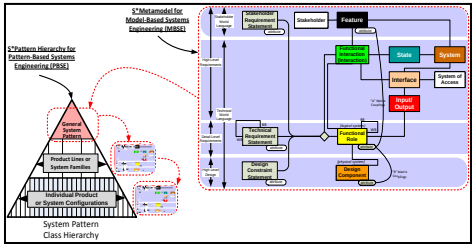
ISO15288 Life Cycle Pattern



ASELCM Pattern



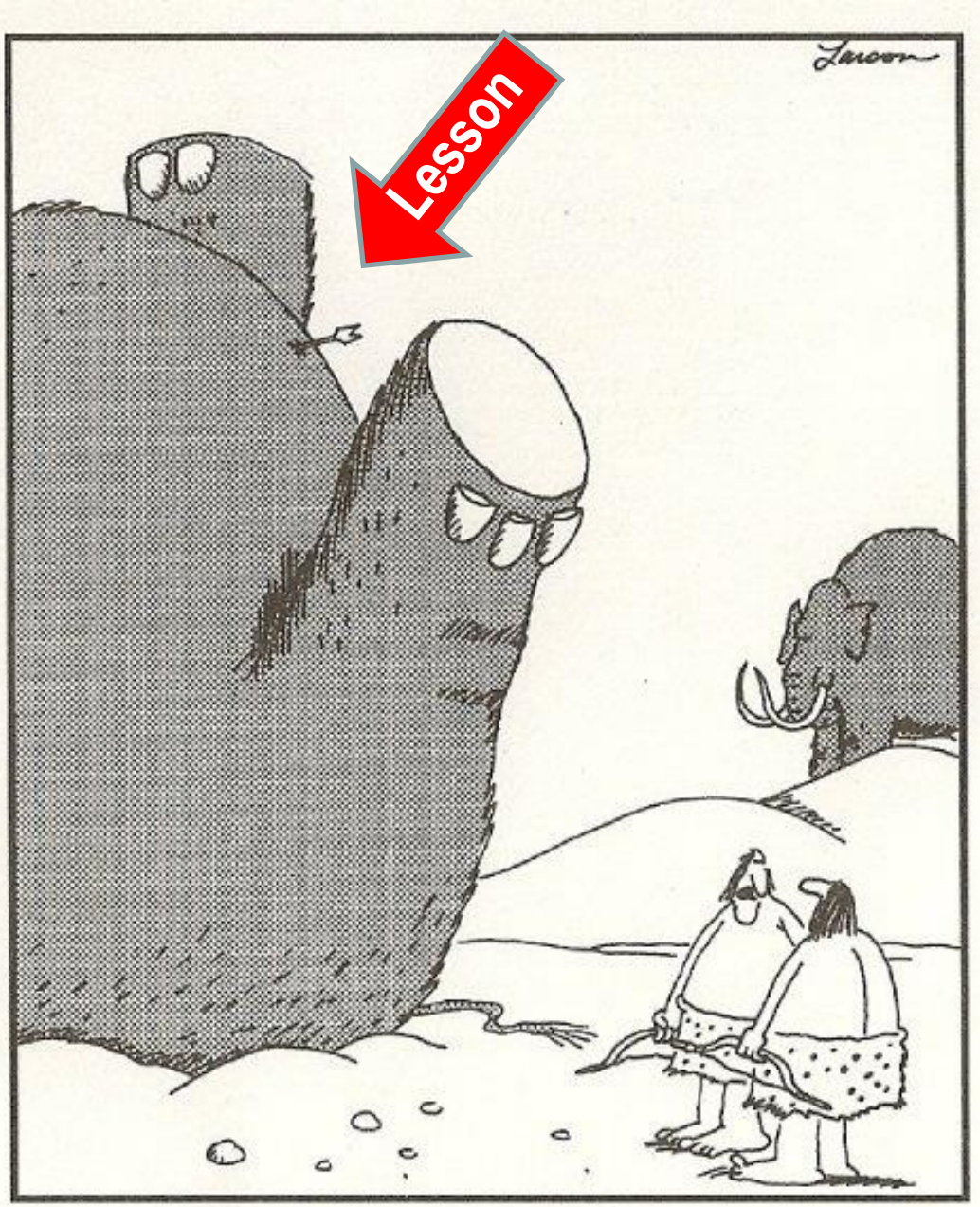
S*Patterns Metamodel



Lessons Learned?

Lessons Learned Report

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

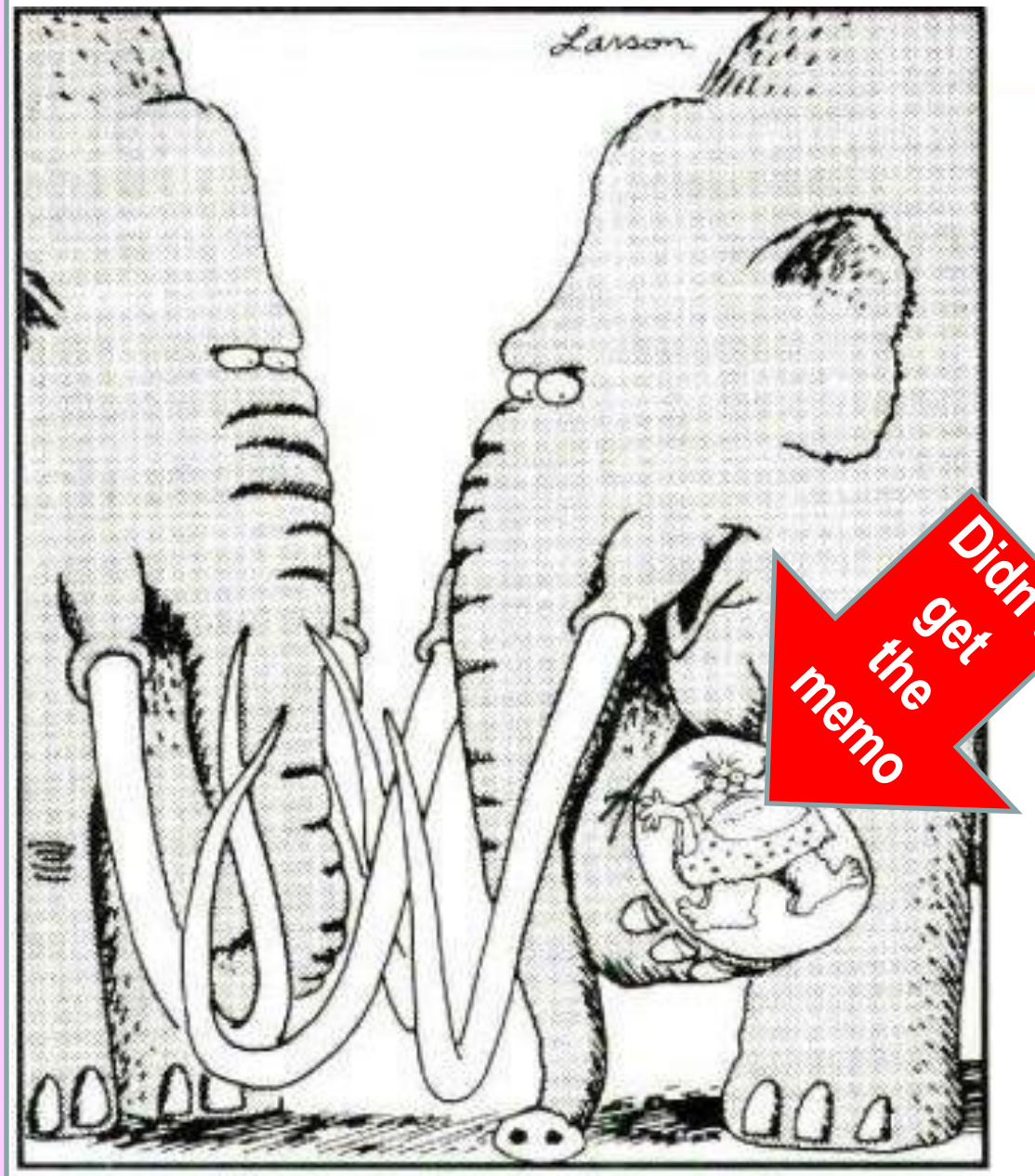


Lessons Effectively Learned?

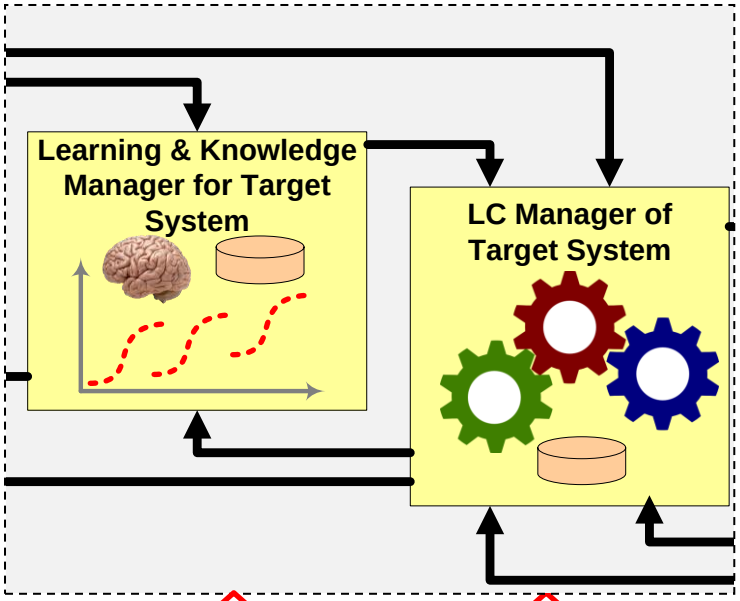
Lessons Learned Report

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

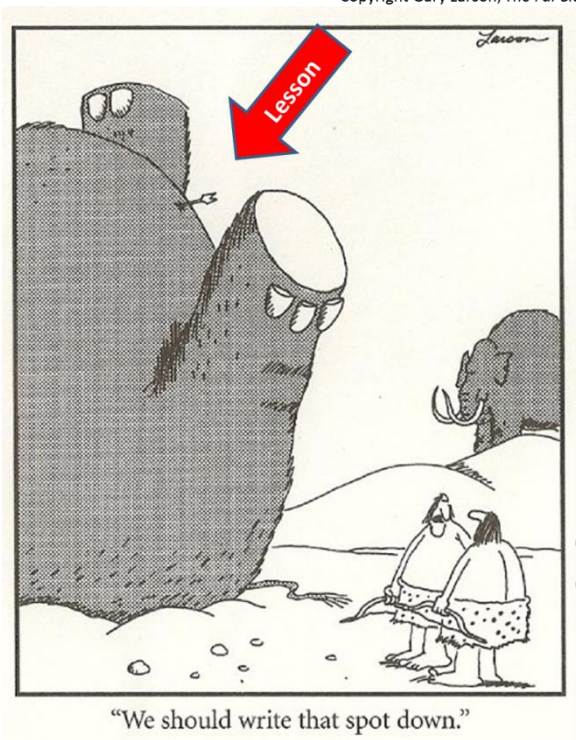
Building learned patterns into “muscle memory” . . .



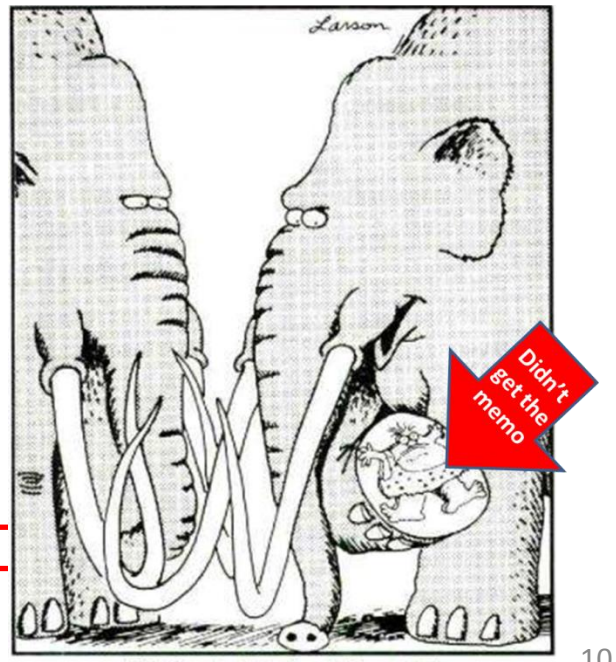
Learning



Executing



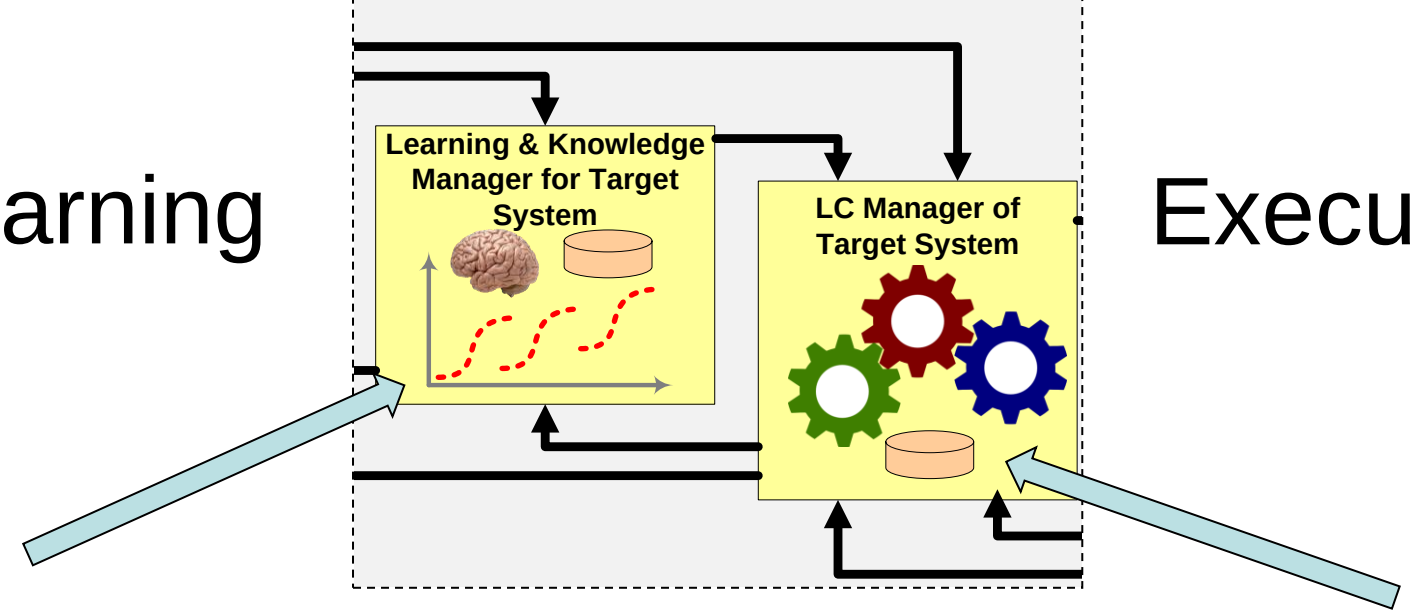
"We should write that spot down."



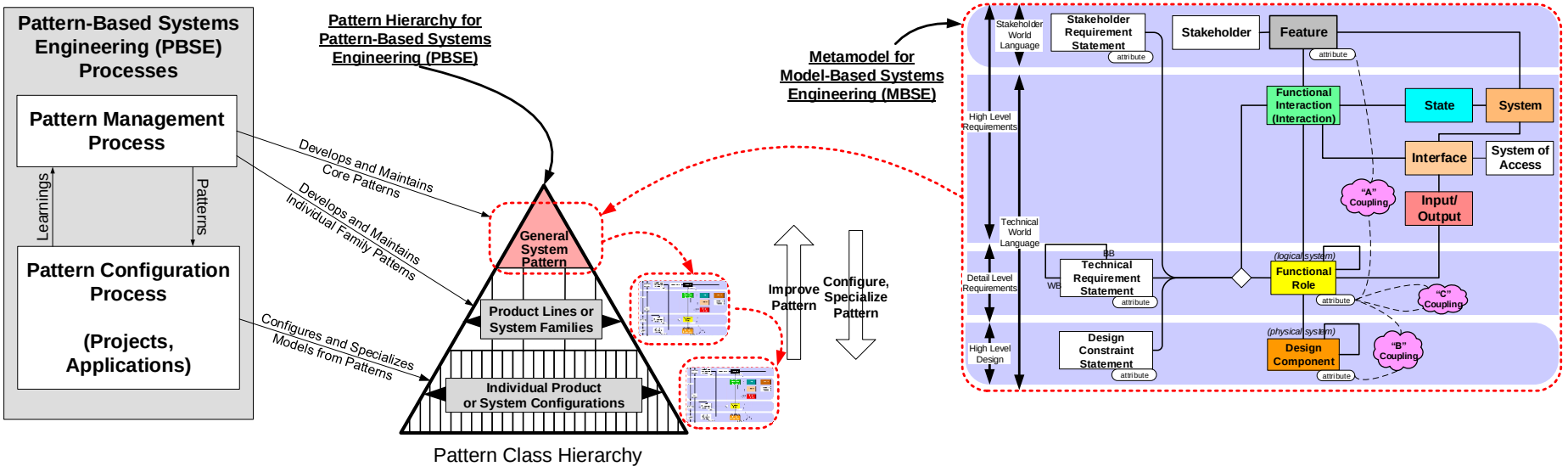
"Well, what the? ... I *thought* I smelled something."

Agility, Information Debt, Learning, Models

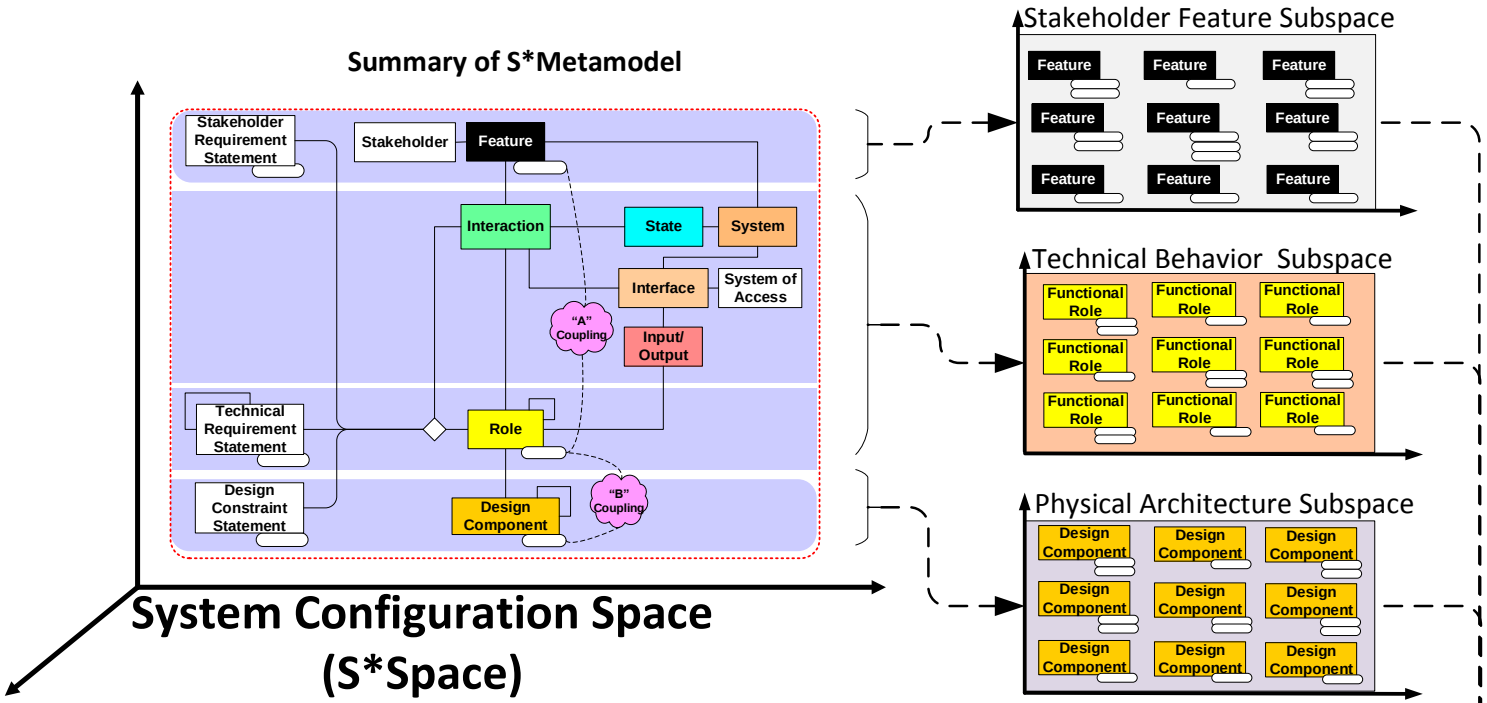
- Where are the "lessons learned" encoded? What would cause them to be accessed during execution?
- Compare to biology:
  - "Muscle Memory" builds "motor" learning directly *in line with a future task*, for future unconscious use, *vs*. syllogistic reasoning that may not be remembered fast enough, or at all
  - This is about "effective learning" for future agile use
  - Just having a growing file of "lessons learned", even if text searchable, is not the same as building what we learn directly in line with the path of future related work that will have to access it in order to be executed.
- Just because we label a report "lessons learned" does not mean that those who will need this information in the future will have *effective* access to it.

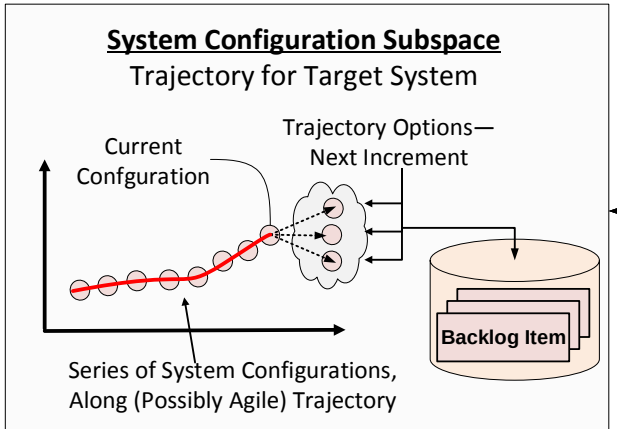


Emergence, Recognition, Extraction, in-line Application of Patterns



Agility as Optimal Trajectory Control in S*Space: Finding the Best Next "Direction" & Increments





References

1. Sheard, S., et al, “A Complexity Primer for Systems Engineers”, INCOSE Complex Systems Working Group, International Council on Systems Engineering, 2015.
2. Schindel, W., “Innovation, Risk, Agility, and Learning, Viewed as Optimal Control & Estimation”, Proc. of the INCOSE 2017 International Symposium, International Council on Systems Engineering, Adelaide, Australia, 2017.
3. Schindel, W., and Dove, R., “Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern”, in Proc. of INCOSE 2016 International Symposium, Edinburgh, UK, 2016.
4. Schindel, W., “Got Phenomena? Science-Based Disciplines for Emerging Systems Challenges”, Proc. of INCOSE IS2017 Symposium, Edinburgh, UK, 2016.
5. Schindel, W., “What Is the Smallest Model of a System?”, Proc. of the INCOSE 2011 International Symposium, International Council on Systems Engineering (2011).
6. INCOSE MBSE Initiative Patterns Working Group web site, at http://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns