Day 1 Introduction to workshop topic

Abstract: Systems of Systems Evolutionary Integrity

Dr. Scott Workinger, Workinger Consulting. scottworkinger@gmail.com

Evolutionary integrity in System of Systems is concerned with upgrades to constituent systems during operation and mitigating disruptions that arise from asynchronous and unpredictable changes when independent constituent systems change without warning. Effective integrity management seeks seamless up-grades to constituent systems and the SoS as a whole. Yet, in practice, service-outage windows often don't accommodate major upgrades, and lengthening the outage can create an unacceptable disruption to SoS capabilities. Moreover, self-serving changes in constituent systems can interfere with total-SoS functionality. Even with well-meaning efforts to manage constituent systems, emergent behavior from constituent system interactions can arise unpredictably, creating serious disruptions.

In general, what are the barriers to integrity management in an SoS composed of independently-owned systems? What inhibits sustained integrity in a complex collection of interacting systems and how can we define integrity for an SoS with no central authority to approve changes? What are the characteristics of a workable integrity management approach? Are these characteristics of effective integrity management represented in examples that we can share? Are there general principles that we can identify and apply to achieve robust integrity management?

This workshop will explore these questions and others that participants have, with the objective of profiling the issues, converging on a set of general needs that an effective integrity management approach must satisfy, and sharing knowledge and experience on approaches that show some effectiveness.

Scott Workinger, Ph.D., Stanford Engineering has 35 years experience leading people who create innovative, practical solutions to business problems and field working systems in a broad spectrum of industries. He is a Past President of the Silicon Valley Chapter of INCOSE and leads the Systems Engineering Transformation Caucus. He teaches technical leadership, systems architecture, test engineering, problem analysis, systems engineering, design thinking, systems thinking, system of systems thinking and transformational thinking. The students who attend his courses come from a broad cross section of backgrounds and include experienced leaders and technologists from such diverse backgrounds as the US Navy, NASA, pharmaceutical companies, aircraft program management, and engineering consulting. Scott has a passion for empowering his students through research, application, and teaching. His teaching style emphasizes coaching students in practical problem solving exercises, dialog, and class discussion.

Workshop:

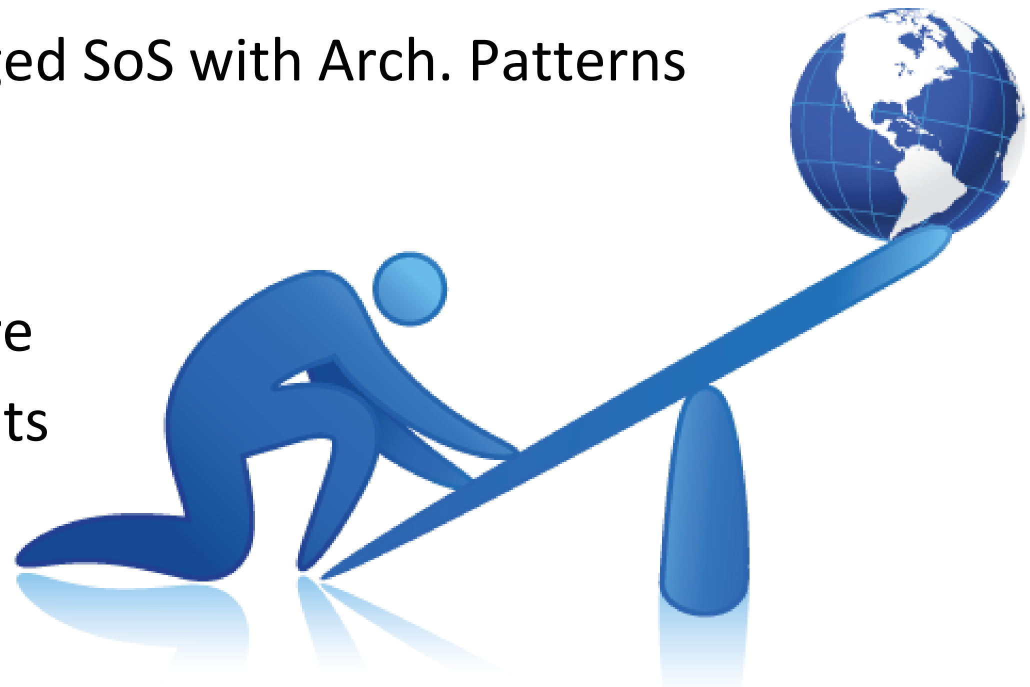
Evolutionary Integrity in Systems of Systems

Socorro, New Mexico

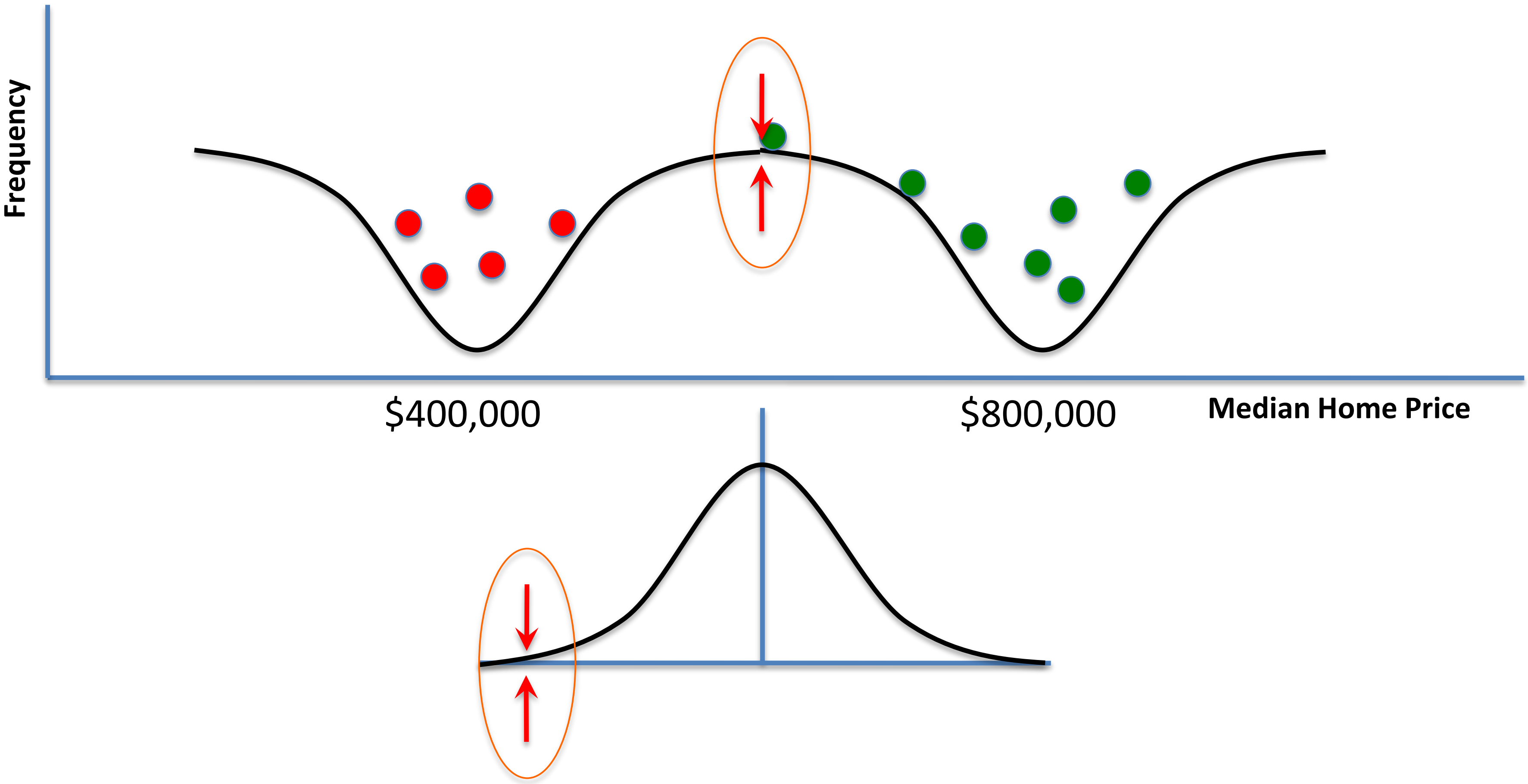
October 28 - 29, 2016

Overview

- Evolutionary Integrity: An example
- The Paradigm Shift in Systems
- SoS Defined
- Examples of Systems of Systems
- Four SoS Types
- Example of an Acknowledged SoS with Arch. Patterns
- Views of SoS Integrity
- What is SoS Integrity?
- Seven Pain Points and More
- Issues / Problem Statements
  - Identifying
  - Prioritizing
  - Tentative Objectives

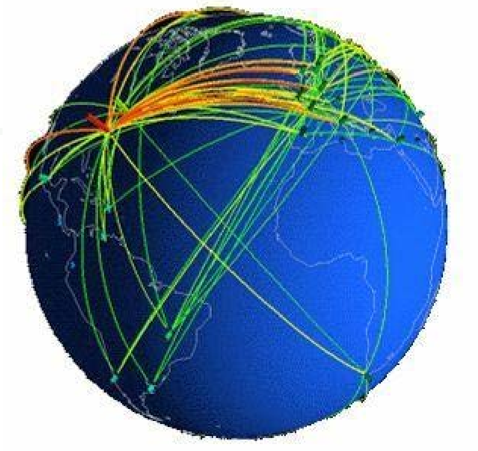


An SoS Integrity Example

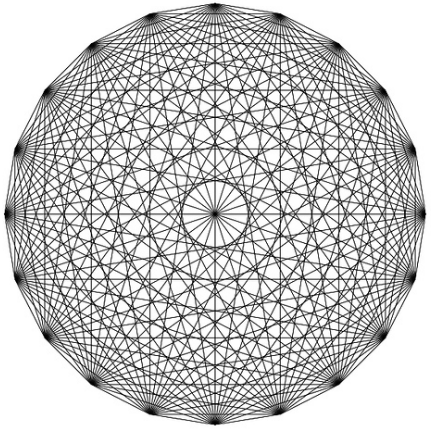


The Paradigm Shift in Systems

- Networked Computation affects:
  - The systems we develop, test, evaluate & manage:
    - Larger
    - More complex
  - The systems we use to test & support them:
    - Often Network Based
    - Test Environments are often Systems of Systems
- Integrated legacy systems
  - "Green field" designs → less frequent
  - Capability engineering → becoming common
- Organizational relationships are changing
  - Top-down policy directives → reaching limits of effectiveness.
  - Spiral development → often preferred
  - T&E → extending through lifecycle
  - Stakeholders →large, diverse groups
- Timing
  - Working on "Internet time" → development & testing schedules compressed
  - Capability development extends through lifecycle



Network Effect



Metcalfe's Law

$V = k * N^2$

ScottWorkinger@gmail.com (707) 632-5134

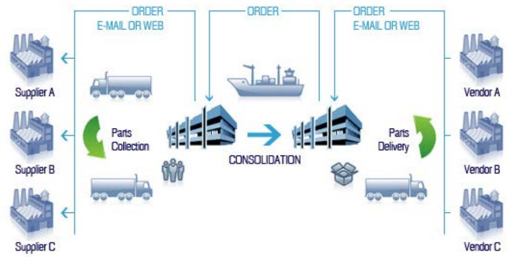
A Definition: System of Systems (SoS)

1. Operational Independence: Component systems must be able to provide value operating independently.
2. Managerial Independence: Although separately acquired and integrated, component systems maintain operational independence.
3. **Evolutionary Development: The development and existence of the system-of-systems is evolutionary.** Functions and purposes are added, removed, and modified over time.
4. Emergent Behavior: The principle purposes of the system of systems arise from functions that do not reside in any component system.
5. Geographic Distribution: The geographic span of the system is sufficiently large that although components can readily exchange information, they cannot exchange substantial quantities of mass or energy.

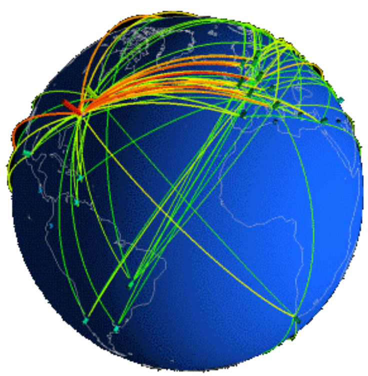

Valiant Shield 2006

\- Mark W. Maier

Examples of Systems of Systems



- A Supply Chain
- A Strike Force
- A Modern Airport
- The iPod Music Delivery System

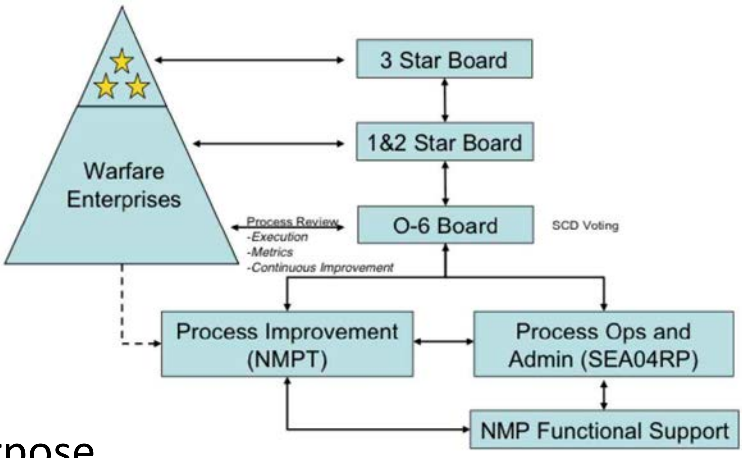


- The Internet
- Internet of Things

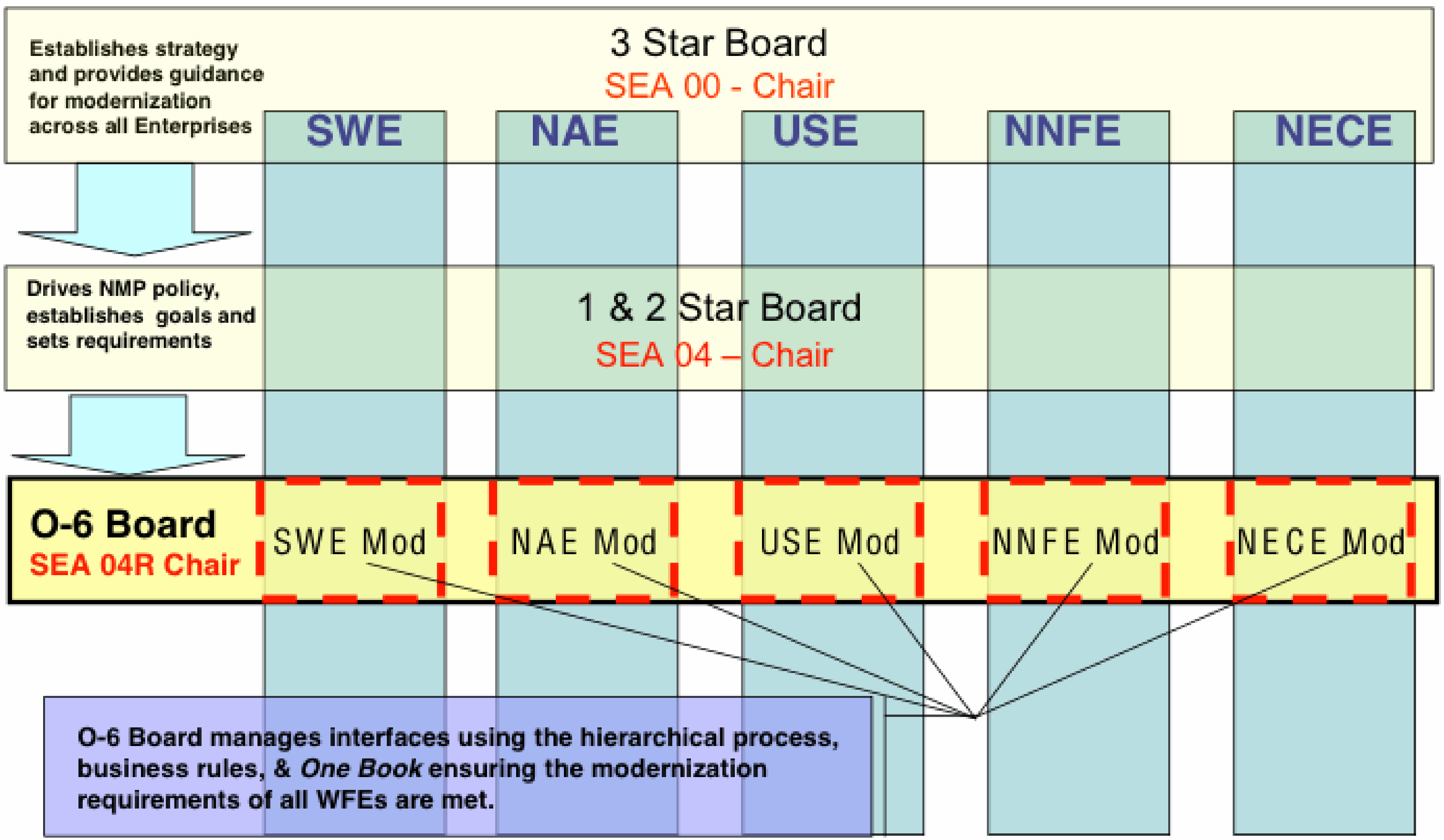
Virtually every system developed today will operate in an SoS context.

Types of Systems of Systems

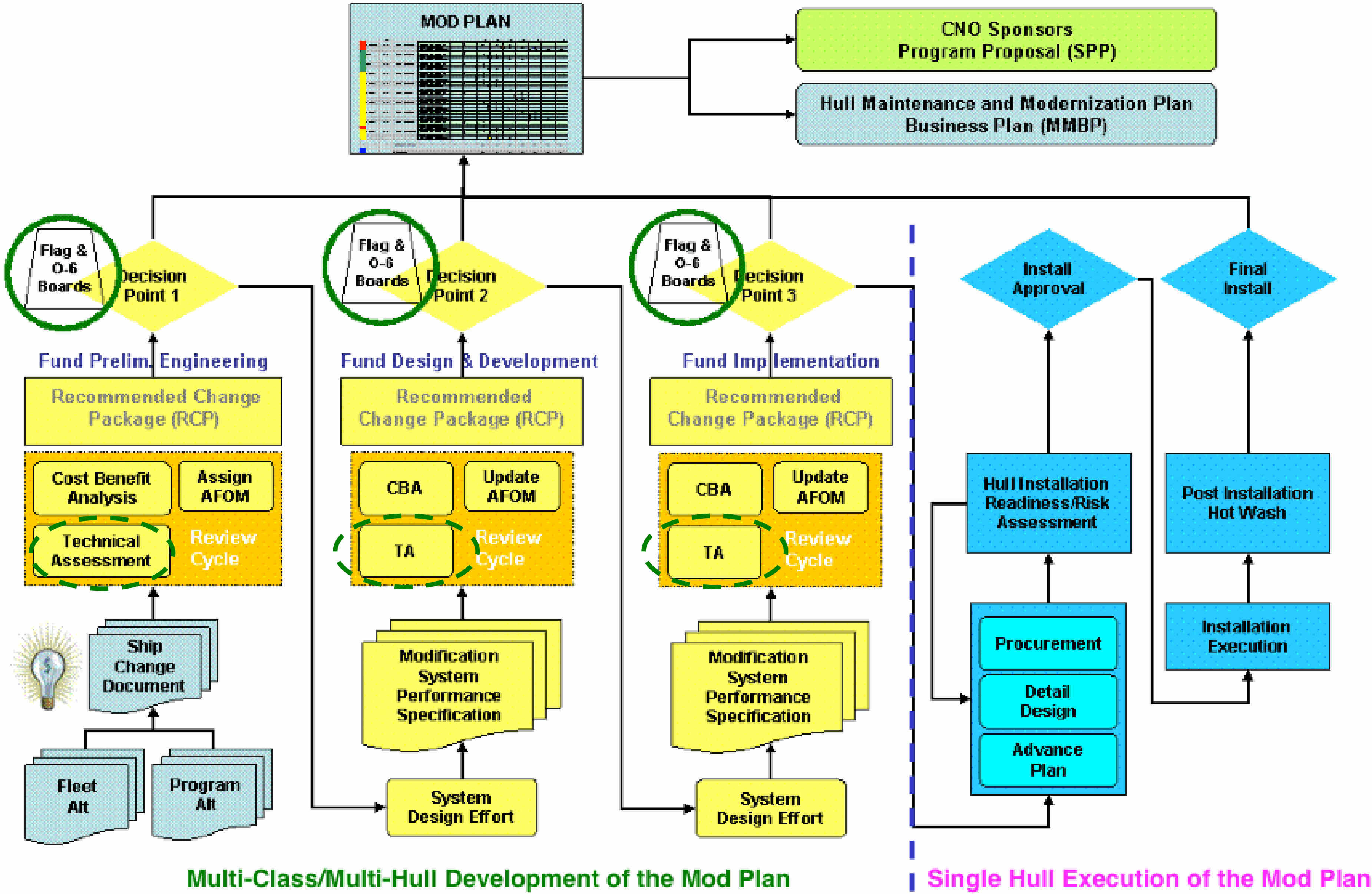
- Directed
  - Specific purpose and central management
  - Constituent systems are subordinates
- Acknowledged
  - Recognized objects, designated manager
  - Ex: Navy Modernization
- Collaborative
  - Voluntary agreement,
  - A coalition of the willing
  - Ex: Internet Engineering Task Force
- Virtual
  - Lacks central management & agreed purpose
  - Ex: Internet



Navy Modernization Enterprise Architecture



Ship Change Development Process



Change Complexity & Emergence

- Approximately 100 ECPs per year
- Multiple versions in service
- Deployed in multiple services
- Deployed with allies
- Temporary deployments
  - Drilling holes for temporary equipment
  - The equipment is temporary
  - The holes are not
- All deployed aircraft must be interoperable with fleet!
- Implementing changes must be approached with care
- Configuration management on older systems is rarely perfect…

F18G

F18F

F18E

F18D

F18C

Requirements: Systems vs. SoS

In *most* SoS:

- No single Owner
- No Chief Architect
- No one to approve requirements

- In SoSE what fulfills the role of classical SE requirements?

SoS Pain Point Survey Results (presented at IS 2012, Rome)

- **Lack of SoS Authorities and Funding**

  Question 1: What are effective collaboration patterns in systems of systems?

- **Constituent Systems**

  Question 2: What are effective approaches to integrating constituent systems into a high functioning SoS?

- **Leadership**

  Question 3: What are the roles and characteristics of effective SoS leadership?

- **Autonomy and Emergence**

  Question 4: How can SoSE provide methods and tools for addressing the complexities (e.g. analysis, modeling, prediction, and architecture) of SoS interdependencies and emergent behaviors?

- **Capabilities and Requirements**

  Question 5: How can SE address SoS capabilities and requirements?

- **Testing, Validation and Learning**

  Question 6: **How can SE approach the challenges of SoS testing, including incremental validation and continuous learning in SoS?**

- **SoS Principles**

  Question 7: What are the key SoS thinking principles, skills and supporting examples?

Tentative Agenda for Discussion

- Primary Objectives
  - Identification of Evolutionary Impact
  - Complexity Exceeds Cognition
  - Distinction Between Complex and Large Systems and SoS
- Secondary Objectives:
  - Dvelopment SoS vs. Production SoS,
  - Defining Systems as SoS
  - Managing SoS without Established Authorization Authority
- What are our objectives moving forward?
  - . Resolve Issues
  - .
  - .

We can modify the above. **The group owns the agenda**

SoS: Perspectives on

Issues, Problems and Opportunities

Suggestion: Use examples for grounding

Suggestion: Try multiple perspectives

SoS: Discussion Points

- In today's environment it is hard to find a system that is not an element of a system of systems. Nearly all systems are a part of a System of Systems (SoS). They interact/interoperate with other systems and are enabled by other systems (e.g. logistics system, training system, etc.)
- There have always been SoS, we (S.E.'s) have just recently opened up out aperture to see it and deliberately engineer them.
- Gödel's incompleteness theory applies to systems and SoS in terms of meta levels, i.e. at some point the only way to deal with SoS is at the meta level where it becomes a system with specific attributes.
- What is the difference between systems and SoS? Is there value in describing how to develop these "systems" differently?
- Hard to find a system that is not part of an SoS.
- From a S.E. perspective, is there a difference between a system and a SoS?
- Need to discuss the difference between a large, complex (complicated) system and a SoS.
- How do we deal with SoS complexity? Modularity? Limitation of a person or people to grasp it?
- What are alternate SoS models? Outside the hierarchical/military models? How else can we look at SoS?

SoS: Discussion Points

- What are we dealing with? The 7 characteristics of SoS and a better way of dealing with one of them? Or the lack of owner/architect? Or the application of SE techniques like requirements, development, and flow?

- How are change impacts identified across a SoS? Lots of discussion on definition of SoS. Is there a standard definition and how is SE community being educated?

- Complexity [of a SoS] exceeded any human cognition (Banking fall).

- Is there a set of known issues or problems already defined? What is prioritization scheme?

- What is evolutionary integrity? How they [SoS] decomposed and how they are formed. What is the original purpose?

- Can an SoS have an inherent "throttle" such that when a positive feedback starts, another part will start a negative feedback to balance and brings SoS back into homeostasis. What SoS have that inherent quality? How can you determine Yes/No? How/If can you add that stability?

- One means to address requirements/capabilities in a SoS is to implement a joint interface control WG to address them by consensus. Not always possible to do this.

- Production SoS versus Development SoS. Biologic Evolutionary parallels: episodic/gradual, ecosystem/niche maximal efficient use of resources.

- Ontogeny recapitulates phylogeny. Is the SoS more than the sum of its component systems? Does the component system drive the evolution of the SoS?

Supplemental

References for Further Reading:

ISO/IEC JTC1/SC7 SoSE Study Group Report

Edit: Of particular interest while searching for the above reference:

J. Dahmann and G. Roedler, "Moving towards standardization for system of systems engineering," *2016 11th System of Systems Engineering Conference (SoSE)*, Kongsberg, 2016, pp. 1-6.
doi: 10.1109/SYSOSE.2016.7542953

"*When Wizards Come Out to Play*" History of Internet Creation.

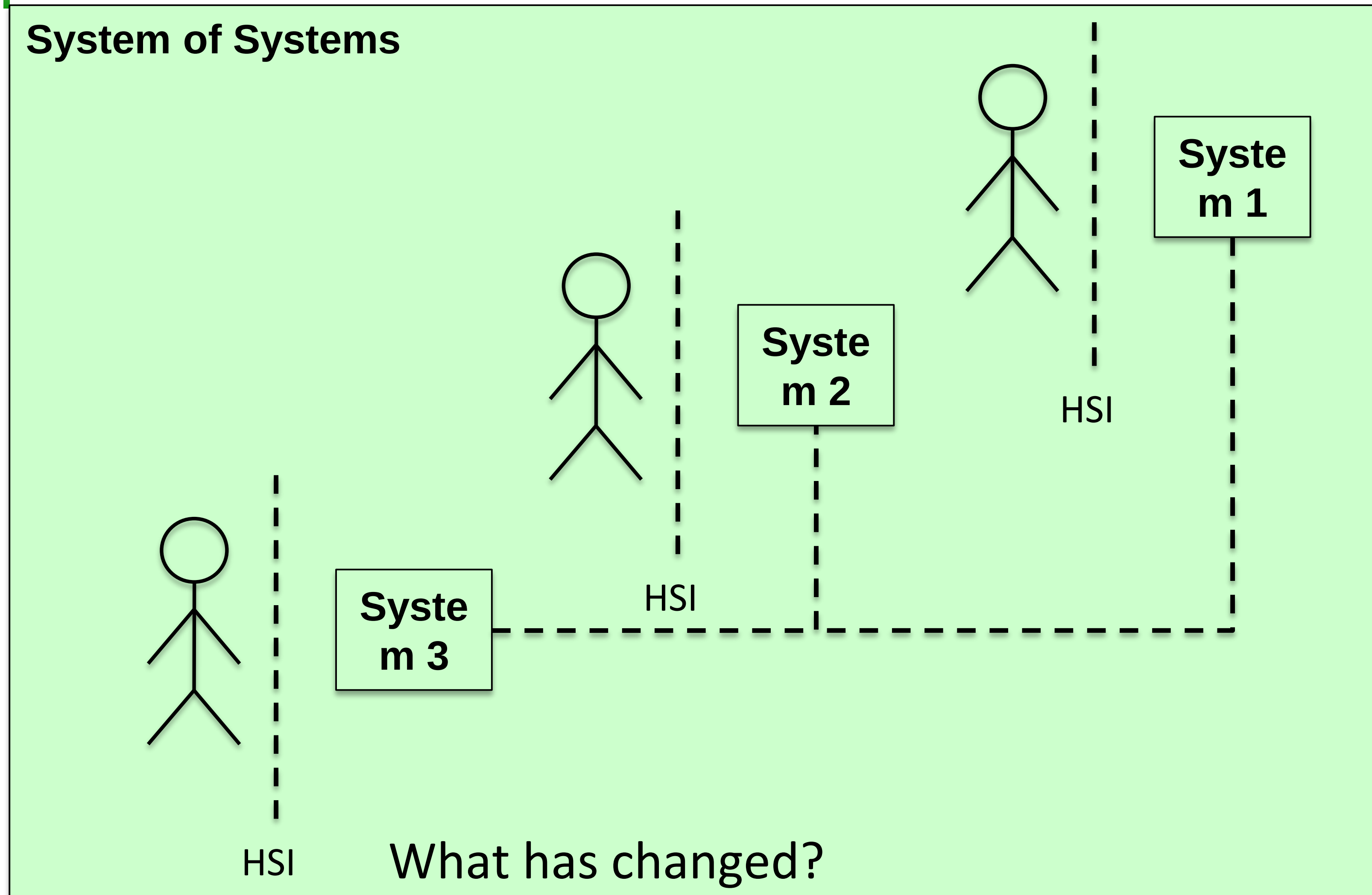
Edit: Reference not found.
Potentially "*Where Wizards Stay Up Late: The Origins of the Internet*" by Kate Hafner and Matthew Lyon

Problems with Viewing Large Systems

- Viewing <u>very</u> large systems
  - Shared awareness
  - Seeing emergent properties
  - Supporting nimble use of views
- Supporting large, diverse teams
  - Shared meaning
  - Shared view
  - Geographic dispersion
  - Displaying <u>all</u> the relevant connections
  - Supporting large, diverse teams
  - Viewing non-co-located collaborative work
- Displaying relevant affordances
- Creating Transparency

Milky Way (Detail)

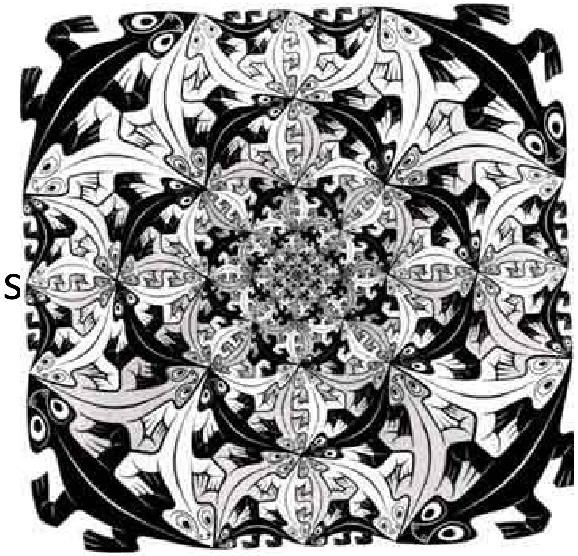
A Consequence of the Paradigm Shift…



Answer: ***The people are inside the system!***

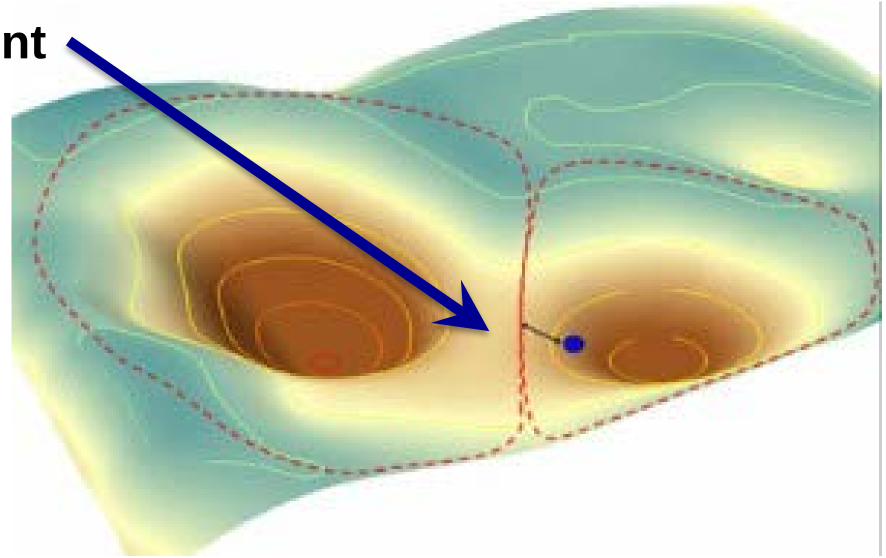
What Makes a System Complex?

- Some Practical Complexity Metrics:
  - Number of components
  - Number of connections
  - Number of types of connections / interface types
  - Number of Internal States
  - Complexity of functions
  - Dynamic complexity
  - Interacting dynamics with varying time lags
  - Strong interactions between components
  - Number of interactions between feedback loops

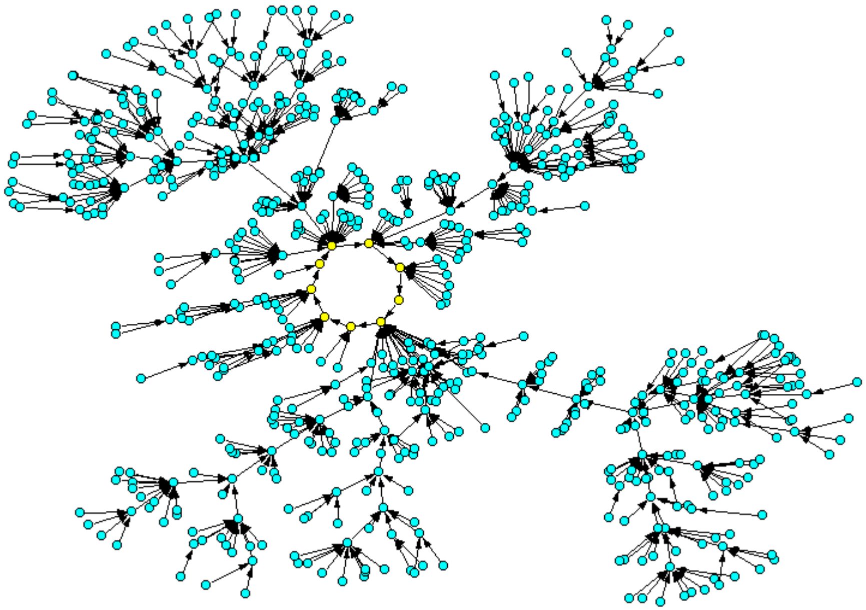


Basins of Attraction

Tipping Point



Multiple Basins of Attraction



Basin of Attraction

Complexity: Leaky Abstractions

Bail faster! Our abstractions are leaking…

Discussion Questions & Comments:

- In today's environment it is hard to find a system that is not an element of a system of systems. Nearly all systems are a part of a System of Systems (SoS). They interact/interoperate with other systems and are enabled by other systems (e.g. logistics system, training system, etc.)

- There have always been SoS, we (S.E.'s) have just recently opened up out aperture to see it and deliberately engineer them.

- Gödel's incompleteness theory applies to systems and SoS in terms of meta levels, i.e. at some point the only way to deal with SoS is at the meta level where it becomes a system with specific attributes.

- What is the difference between systems and SoS? Is there value in describing how to develop these "systems" differently?

- Hard to find a system that is not part of an SoS.

- From a S.E. perspective, is there a difference between a system and a SoS?

- Need to discuss the difference between a large, complex (complicated) system and a SoS.

- How do we deal with SoS complexity? Modularity? Limitation of a person or people to grasp it?

- What are alternate SoS models? Outside the hierarchical/military models? How else can we look at SoS?

- What are we dealing with? The 7 characteristics of SoS and a better way of dealing with one of them? Or the lack of owner/architect? Or the application of SE techniques like requirements, development, and flow?

- How are change impacts identified across a SoS? Lots of discussion on definition of SoS. Is there a standard definition and how is SE community being educated?

- Complexity [of a SoS] exceeded any human cognition (Banking fall).

- Is there a set of known issues or problems already defined? What is prioritization scheme?

- What is evolutionary integrity? How they [SoS] decomposed and how they are formed. What is the original purpose?

- Can an SoS have an inherent "throttle" such that when a positive feedback starts, another part will start a negative feedback to balance and brings SoS back into homeostasis. What SoS have that inherent quality? How can you determine Yes/No? How/If can you add that stability?

- One means to address requirements/capabilities in a SoS is to implement a joint interface control WG to address them by consensus. Not always possible to do this.

- Production SoS versus Development SoS. Biologic Evolutionary parallels: episodic/gradual, ecosystem/niche maximal efficient use of resources.

- Ontogeny recapitulates phylogeny. Is the SoS more than the sum of its component systems? Does the component system drive the evolution of the SoS?

References for Further Reading:

ISO/IEC JTC1/SC7 SoSE Study Group Report
Edit: Of particular interest while searching for the above reference:
J. Dahmann and G. Roedler, "Moving towards standardization for system of systems engineering," *2016 11th System of Systems Engineering Conference (SoSE)*, Kongsberg, 2016, pp. 1-6.
doi: 10.1109/SYSOSE.2016.7542953

"*When Wizards Come Out to Play*" History of Internet Creation.
Edit: Reference not found. Potentially "*Where Wizards Stay Up Late: The Origins of the Internet*" by Kate Hafner and Matthew Lyon

Fail-Fast Rapid Innovation Concepts

Moderator: Scott Workinger

Day-1 Brief Out (as decided Friday, subject to change during Saturday)

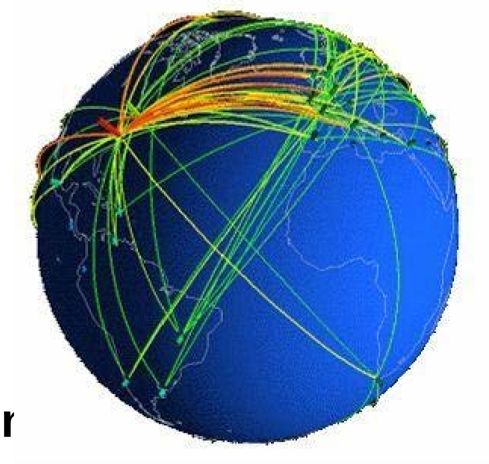
Tentative Agenda for Discussion

- Primary Objectives
  - Identification of Evolutionary Impact
  - Complexity Exceeds Cognition
  - Distinction Between Complex and Large Systems and SoS
- Secondary Objectives:
  - Dvelopment SoS vs. Production SoS,
  - Defining Systems as SoS
  - Managing SoS without Established Authorization Authority
- What are our objectives moving forward?
  - . Resolve Issues
  - .
  - .

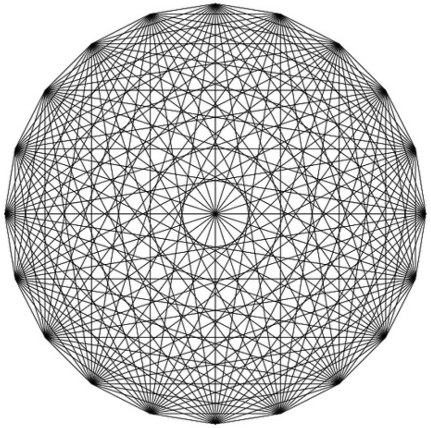
Evolutionary Integrity in Systems of Systems
Moderator: Scott Workinger
Day-2 Brief Out

Topics:

- Defining a System as an SoS
- Managing Evolutionary Impacts for an SoS
- Complexity is a serious challenge to Cognition
- Development SoS vs. Production SoS
- Distinction between SoS and Large & Complex Syster
- Managing an OS with minimal authority



Network Effect



Metcalfe's Law
$V = k * N^2$

Defining a System as an SoS

It may be that the name, "System of Systems" is unlovely

"A rose by any other name would smell as sweet."

We agreed that the main purpose of our use of names in this context is to:

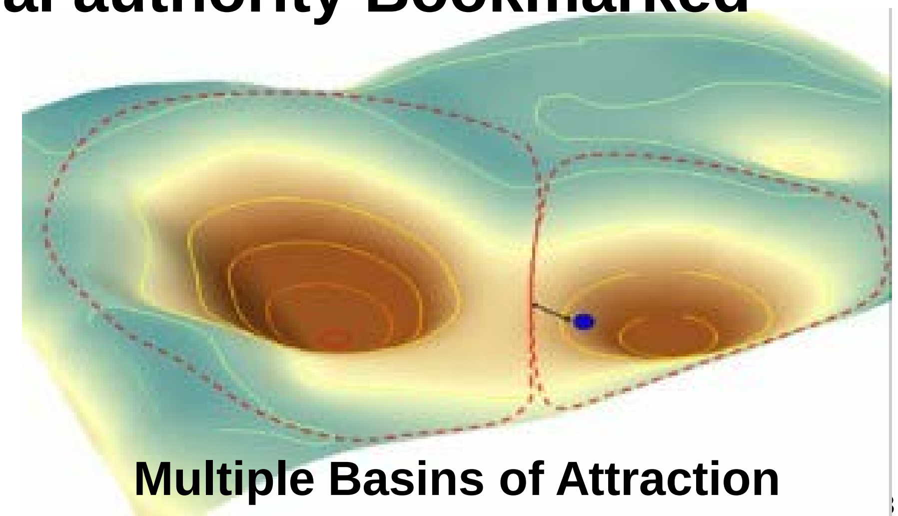
- **Label distinctions that help us make useful conclusions about what engineering techniques to apply in particular situations.**

Bail faster!

Our abstractions are leaking…

The Path Forward…

- Managing Evolutionary Impacts for an SoS – White paper [Team, Lead: Scott W.]
- Human engagement with SoS – Paper or presentation to SoS WG [Rick Kennedy, Bandit] (Reframed from "Complexity exceeds cognition."]
- Developmental and Operational SoS – Managing the differences Bookmarked
- Distinctions between SoS and large, complex systems Bookmarked
- Managing an SoS with minimal authority Bookmarked
- Deliverables planned:
  - Two papers
  - Collaboration with:
    - SOS WG
    - Complex Systems WG



Multiple Basins of Attraction