System of Systems Engineering and Family of Systems Engineering From a Standards, V-Model, and Dual-V Model Perspective

John O. Clark
Chief Engineer
Northrop Grumman Corporation
713 Wolfsnare Crescent
Virginia Beach, VA 23454
757-481-1504
john.clark@ngc.com

Abstract - *System of Systems Engineering (SoSE) and Family of Systems Engineering (FoSE) continue to be two of the least well-understood SE disciplines. Knowledge of the SE standards, the V-Model, and particularly the 3-dimensional Dual-V Model, significantly aid this understanding, including the relationship between SE, SoSE, and FoSE.*

The goals of this paper are to: 1) define SoS, SoSE, and FoSE from an SE standards perspective; 2) describe the original V-Model and the Dual-V Model; 3) show how to apply these SE standards and V-Models to a system, to SoSs, and to FoSs; and 4) encourage and challenge the participants to understand, select, tailor, and apply these SE standards and V-Models to complex SoSs and FoSs. Individuals may have an understanding of portions of SE, SoSE, and FoSE based on other sources. The SE standards, V-Model, and Dual-V Model provide a more complete and common understanding.

Keywords: System of Systems (SoS), System of Systems Engineering (SoSE), Family of Systems (FoS), Family of Systems Engineering (FoSE), Complex Systems, Complex Systems Engineering, V-Model, Dual V-Model.

1 Introduction

The subject of SoSE versus SE is currently debated in the literature and at conferences. The question is asked: "Is engineering a system of systems really any different from engineering an ordinary system?" [1]. Some believe that SoSE is "different" from SE, the SE processes are inadequate or insufficient for SoSE, and additional processes are needed. Others, like me, believe the SE processes as documented in the SE standards: IEEE 1220, EIA/IS-632, EIA-632, ISO 15288 [2-8], and the guide: ISO TR 19760 [9], are a necessary and sufficient set of processes for SoSE, and no additional processes are needed.

The above SE standards and guide are referenced in this paper and used in the presentation that accompanies this paper. However, because they are copyrighted by the publishing organizations, material from them cannot be reproduced in this paper or in softcopies of the presentation. Refer to these SE standards and guide for this information.

In my opinion (based on reading, comparing, understanding, teaching, revising, tailoring, and applying the SE standards), there is only one classical SE process as shown in Figure 1 [5].

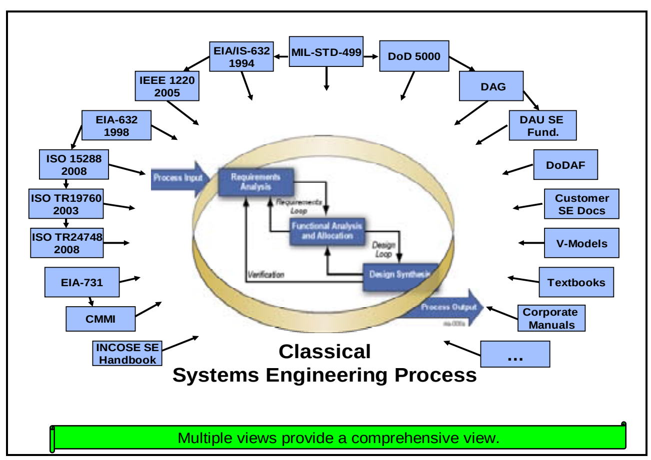

Figure 1. Systems Engineering Views

Each SE standard presents a slightly different view of this one classical SE process. By understanding each SE standard, and looking at each standard's view, a systems engineer can get a comprehensive view of this one classical SE process and apply it to SoSE and FoSE. This principle also applies to the guides, manuals, handbooks, etc, shown in Figure 1.

Systems engineers may struggle with applying SE to FoSE. However, FoSE is simply SE applied to a FoS. By family, we mean a product-line or domain, wherein some assets are re-used un-modified; some assets are modified, used, and re-used later; and some assets are developed new, used, and re-used later. Product-lines are the result.

This paper addresses SoSE and FoSE from the SE standards, V-Model, and Dual V-Model perspective. In my opinion, this information is sorely needed to meet the challenges of complex SoSE and FoSE.

2 What is Different about SoSE and FoSE from SE?

In my opinion, SoSE and FoSE are an acquisition management problem, not a technical problem. The technical problem is solvable using the SE Standards and V-Models, but the acquisition management problem has not been solved. A few key management issues are:

- There is no god (no overall Program Manager) of a SoS or FoS
- Acquisitions are stovepipes (single systems, not SoS or FoS)
- Systems are directed to "integrate" with other systems, often after fielding
- Suppliers don't cooperate with each other in FoSE (they believe it's not in their best interest)
- Acquirers don't cooperate with each other for the same reason
- FoSE costs more up-front to develop for re-use (but saves much more later)

There are several key challenges to SoSE [10]. For example, SoS and FoS may consist of new, modified, or unmodified systems; and some systems may be evolving and their future unpredictable. To mitigate the risks inherent in these challenges, focus should be placed on developing and controlling the interfaces between system elements and external systems. Developing and controlling interfaces correctly is what integration and interoperability are about.

3 The Building Block

For a system or a SoS, the SE standards apply the Building Block concept. A system or SoS Building Block consists of Products, Processes, and People (some standards call these three items Elements) as shown in Figure 2.

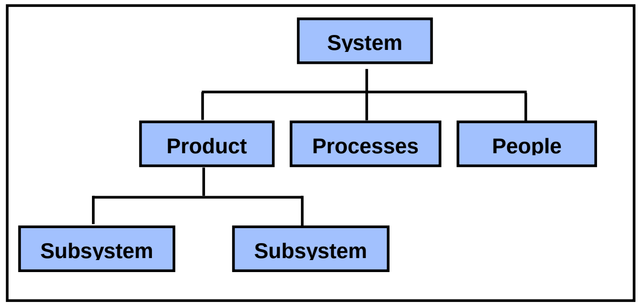

Figure 2. System Building Block

Next, the SE standards construct a system or a SoS using these Building Blocks as shown in Figure 3. A system or SoS can be decomposed from the top down, composed from the bottom up, or a combination of the two.

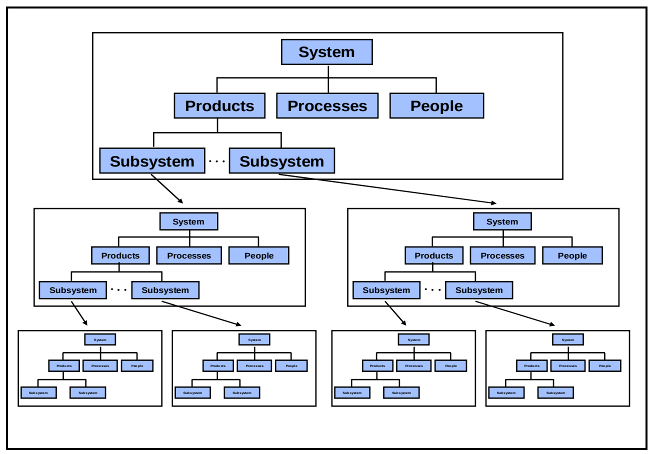

Figure 3. System of Systems Building Blocks

Each subsystem of the system or the SoS is treated as a system in its own right. For top-down SoSE, the Building Block structure continues on down the System Breakdown Structure (SBS) to the leaf-level that is needed to describe the SoS. For bottom-up SoSE, the opposite occurs.

Other structures are determined from the SBS such as the Work Breakdown Structure (WBS), the Specification Tree, and the Integrated Product Team (IPT) organization.

4 Simple Definitions of SoS and FoS

Following are simple definitions of SoS and FoS:

- SoS: The sum of the whole is greater than the sum of the individual parts:
  - The parts are integrated (i.e., have interfaces)
  - The parts may or may not be members of a common domain (such as a product line, for example: surface ship radars)

- FoS: The sum of the whole is equal to the sum of the individual parts:
  - The parts are not integrated
  - The parts are members of a common domain (such as a product line)

Integrating systems could result in the whole being less than the sum of the individual parts, but I assume that's not the case if they are integrated correctly!

5 The U.S. Department of Defense's Definitions of SoS and FoS

Per the DoD Defense Acquisition Guidebook (DAG) 2006 version [11], SoSE:

- Deals with planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into a SoS capability greater than the sum of the capabilities of the constituent parts.
- SoSs should be treated and managed as a system in their own right, and should therefore be subject to the same systems engineering processes and best practices as applied to individual systems.
- Differs from the engineering of a single system. The considerations should include the following factors or attributes:
  - Larger scope and greater complexity of integration efforts;
  - Collaborative and dynamic engineering;
  - Engineering under the condition of uncertainty;
  - Emphasis on design optimization;
  - Continuing architectural reconfiguration;
  - Simultaneous modeling and simulation of emergent system of systems behavior; and
  - Rigorous interface design and management.

Per the DAG 2006 version, a FoS:

- Is not considered to be a system per se.
- Does not create capability beyond the additive sum of the individual capabilities of its member systems.
- Basically a grouping of systems having some common characteristic(s). For example, each system in a FoS may belong to a domain or product lines (e.g., a family of missiles or aircraft).
- Lacks the synergy of a SoS.
- Does not acquire qualitatively new properties as a result of the grouping. In fact, the member systems may not be connected into a whole.

Per the DoD SE Guide to Systems of Systems [12]:

- As defined in the Defense Acquisition Guidebook (DAG) 2008 version, a SoS is "a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities." (Note by J. Clark: The DAG 2008 version has not been published as yet, but was anticipated to contain this definition.)

- An SoS is defined as a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities [DAG 2004]. Both individual systems and SoS conform to the accepted definition of a system in that each consists of parts, relationships, and a whole that is greater than the sum of the parts; however, although an SoS is a system, not all systems are SoS. (Note by J. Clark: The DAG 2004 version was superseded by the DAG 2006 version referenced above, and the DAG 2008 definition of SoS was anticipated to revert back to this DAG 2004 definition.)

- A family of systems (FoS) is defined as a set of systems that provide similar capabilities through different approaches to achieve similar or complementary effects [CJCS 2007(1)]. For instance, the war fighter may need the capability to track moving targets. The FoS that provides this capability could include unmanned or manned aerial vehicles with appropriate sensors, a space-based sensor platform, or a special operations capability. Each can provide the ability to track moving targets but with differing characteristics of persistence, accuracy, timeliness, etc." This definition is included for completeness. FoS are fundamentally different from SoS because, as CJCSI goes on to say, a family of systems lacks the synergy of a system of systems. The family of systems does not acquire qualitatively new properties as a result of the grouping. In fact, the member systems may not be connected into a whole. This guide specifically addresses SoS, but some of its contents may apply to FoS.

- SoS systems engineering deals with planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into an SoS capability greater than the sum of the capabilities of the constituent parts [DAG 2004]. Consistent with the DoD transformation vision and enabling net-centric operations (NCO), SoS may deliver capabilities by combining multiple collaborative and autonomous-yet-interacting systems. The mix of systems may include existing, partially developed, and yet-to be designed independent systems. (Note by J. Clark: As noted above, the DAG 2004 version was superseded by the DAG 2006 version.)

The SE Guide to SoS identifies 3 new SoS SE "roles":

- Translating Capability Objectives

- Understanding Systems & Relationships
- Monitoring & Assessing Changes

It is unclear why these three SoS SE roles are really "new." In my opinion they are included in the 16 technical and technical management processes defined in the DAG chapter 4, and are included in the SE Standards, V-Model, and Dual-V Model on which the DAG chapter 4 is based.

6 INCOSE's Definitions of System and SoS

Per the INCOSE SE Handbook [10]:

- A system is a combination of interacting elements organized to achieve one or more stated purposes.

- System of systems applies to a system-of-interest whose system elements are themselves systems; typically these entail large scale inter-disciplinary problems with multiple, heterogeneous, distributed systems.

Personally, I like this simple definition of SoS and would simplify it even further as follows:

- System of systems applies to a system whose system elements are themselves systems.

I like the above definition because it does not distinguish between dependent or independent systems, or any other characteristics of systems or SoS, and it supports my ideas about systems thinking discussed later in this paper.

7 The V-Model

Although not a SE standard, the V-Model is a very popular model of the SE process. The original V-Model [13] is shown in Figure 4. For top-down SE (i.e., forward engineering), the process starts on the upper left and goes to the upper right. For bottom-up (i.e., reverse engineering), it starts on the upper right and goes to the upper left.

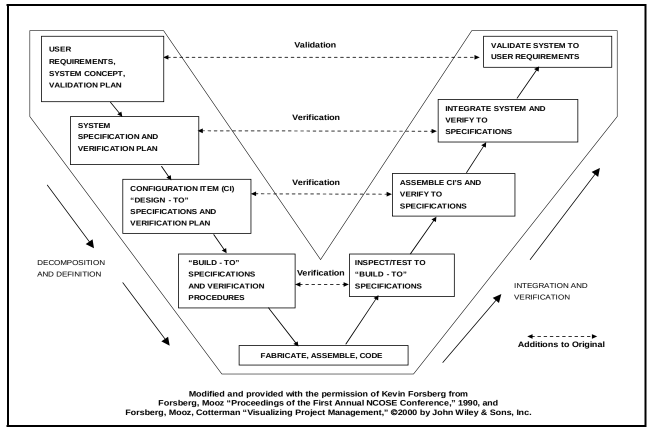

Figure 4. Original V-Model

The application of the original V-Model to a system or SoS is shown in Figure 5. This application is similar to the Building Block in which the Vs are repeated at each level of the SBS.

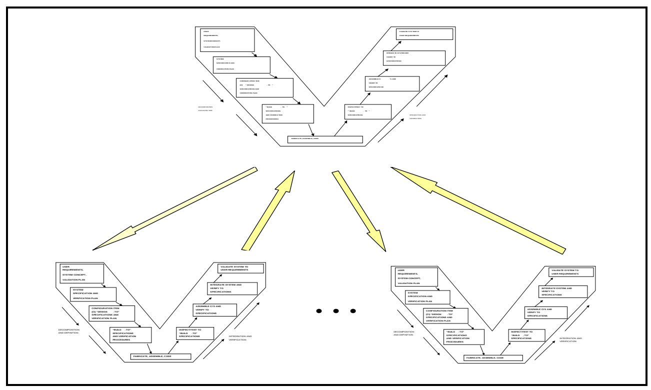

Figure 5. System or SoS V-Model

An example of the detailed application of the V-Model to a system or a SoS is presented in Figure 6, the Dual-V Model [14].

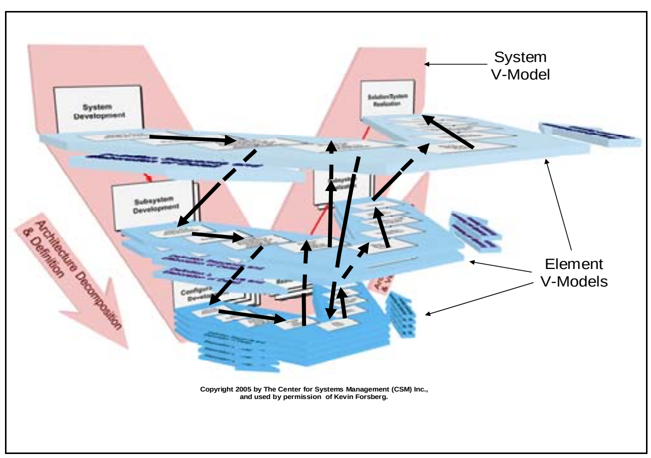

Figure 6. Dual V-Model

In this example in Figure 6 there are 1 system, 2 subsystems, and 4 Lowest Configuration Items (LCIs). The vertical backplane is the System-V and the horizontal planes are the Element-Vs. Each Element-V is the same as Figure 4 and is applied at each level of the System-V. A SoS-V would be depicted by adding the SoS in the backplane above multiple systems. Parts of an LCI would be depicted by adding the parts in the backplane below the LCI.

For top-down SE, in Figure 6, system requirements are allocated down to subsystems from the system "design-to" (i.e., requirements) specification on the left side of the System Element V. Each Subsystem Element V begins at its requirements process, passes its "build-to" (i.e., design) spec up to the system "build-to" spec process of the System Element V, ends at its validation process, and returns the result to the "fabricate, assemble, code" process at the bottom of the System Element V.

Similarly, subsystem requirements are allocated down to LCIs from the subsystem "design-to" specifications on the left side of the Subsystem Element V. Each LCI Element V begins at its requirements process, passes its "build-to" spec up to the subsystem "build-to" spec process of the Subsystem V, commences its "fabricate, assemble, code" process at the bottom of the LCI Element V, ends at its validation process, and returns the result to the "fabricate, assemble, code" process at the bottom of the Subsystem Element V.

The application of the original V-Model to a FoS is shown in Figure 7. Here, the Vs are sequentially nested into the page, signifying that for subsequent systems, some prior-system V assets are re-used un-modified; some assets are modified, used, and re-used later; and some assets are developed new, used, and re-used later. If a member of the FoS is a SoS, then the Vs continue down to the next lower-level as was shown in Figures 5 and 6.

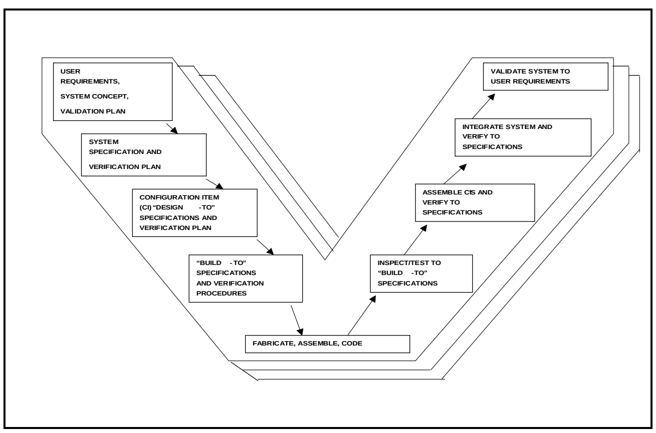

Figure 7. FoS V-Model

8 Technical Baselines, Documents, Reviews, and Audits

An example of Technical Baselines, Documents, Reviews, and Audits for a system is shown in Figure 8 (the acronyms should be self-explanatory).

The top group shows the full menu of Technical Baselines, Documents, Reviews, and Audits from which the systems engineer selects (tailors) the appropriate ones for the system, subsystem, and LCI levels.

Requirements, functions, and preliminary design are shown on the left side. For top-down SE, these flow down from the system- level. System requirements allocated to a subsystem (system allocated baseline) become the requirements baseline for that subsystem. Subsystem requirements allocated to a LCI (subsystem allocated baseline) become the requirements baseline for that LCI. From these requirements baselines come the functional, allocated, and product baselines for that level of the SBS.

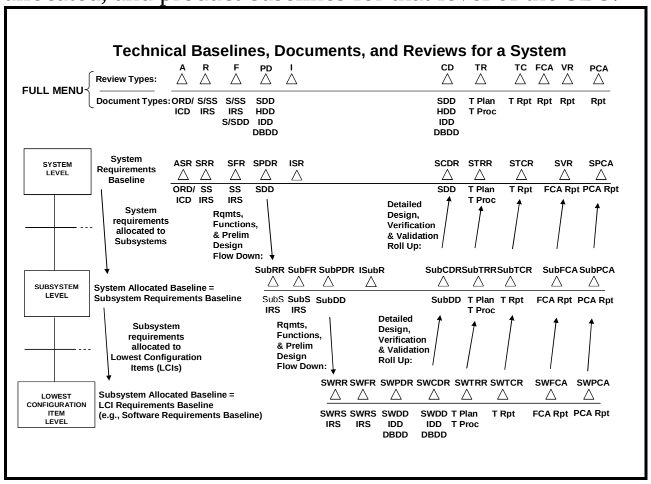

Figure 8. Technical Baselines, Documents, Reviews, and Audits for a System

System requirements reviews precede subsystem requirements reviews which precede LCI requirements reviews. The same sequence applies to functional and preliminary design reviews.

Critical design, verification (e.g., test), validation, and audits are shown on the right side of Figure 8. These flow up to the system-level. LCI critical design reviews precede subsystem critical design reviews which precede system critical design reviews. The same sequence applies to verification and validation reviews and audits.

Extending Figure 8 to a SoS results in Figure 9. The same sequence of technical baselines, documents, reviews, and audits applies. A SoS is just another system, albeit more complex. Per the Defense Acquisition Guidebook: "SoSs should be treated and managed as a system in their own right, and should therefore be subject to the same systems engineering processes and best practices as applied to individual systems."

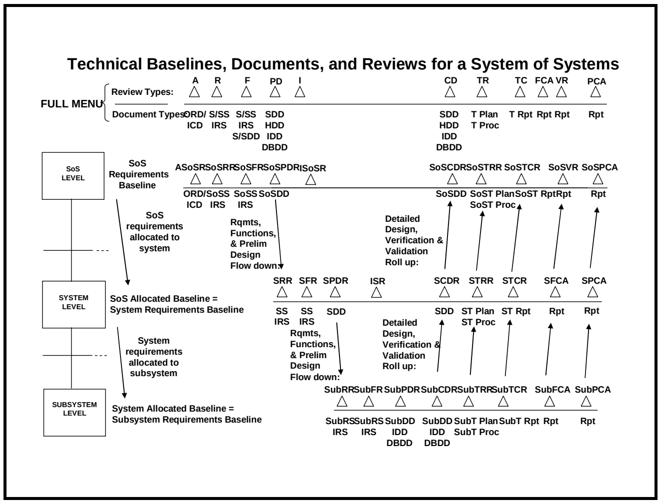

Figure 9. Technical Baselines, Documents, Reviews, and Audits for a System of Systems

9 Systems Thinking

In my opinion, systems thinking involves the following:

- Everything and everyone (from the universe to the nucleus of an atom) is a system, a SoS, and a subsystem of a higher-order system
- Everything and everyone that exists/existed (things, people, thoughts, sayings, writings, actions, etc.) uses/used the systems engineering process
- You see everything and everyone as a system, a SoS, and a subsystem of a higher-order system
- You “Stand on the standards”
- You have “The Knack”

Every SoS is a system, every system is a SoS, and every system is a subsystem of a higher-order system, from the universe to the nucleus of an atom. It’s a matter of degree of complexity. The universe is a system, a SoS, and a subsystem of a higher-order system (ponder which higher-order system); the milky way galaxy is a subsystem of the universe; the solar system is a subsystem of the milky way galaxy; the earth is a subsystem of the solar system; the weather system, the ecosystem, and the human system are subsystems of the earth; the digestive system is a subsystem of the human system; the stomach is a subsystem of the digestive system; the cell is a subsystem of the stomach; the molecule is a subsystem of the cell; the atom is a subsystem of the molecule; and the nucleus is a subsystem of the atom.

To me, the best example of a SoS is our human body. We are a SoS. We have a digestive system, a circulatory system, a respiratory system, a lymph system, a muscular system, a skeletal system, a reproductive system, a nervous system (sometimes I wish mine weren’t so active!), etc. These systems have an interface, are integrated, and are interoperable. How fearfully and wonderfully made we are!

To me, the best example of a FoS is brothers and sisters. When brothers and sisters are separated (not interfaced), they are a FoS (a product line of their parents!). However, when they come together, have an interface, and are integrated, they become a SoS … hopefully they are interoperable!

Knowledge of the SE standards, the V-Model, and particularly the 3-dimensional Dual-V Model, will significantly aid systems thinking and its application to a system, a SoS, a subsystem, and a FoS.

Systems thinking is really weird. Your thinking is transformed and your mind is renewed. You find yourself being transformed by the renewing of your mind! Enjoy systems thinking !

10 Complex SoSE and FoSE

So, how does all this apply to complex SoSE and FoSE? SoSE and FoSE address:

- Bounding and defining problem context
- Solution methods and techniques
- Solution tools
- Strategies for efficient solutions
- Various applications

This paper aims at providing and communicating focused solutions and propositions to the problems being encountered in complex SoS and FoS situations. Situations that form around complex SoS and FoS are characterized by lack of clarity, uncertainty, ambiguity, and limited understanding - stretching capabilities to manage and engineer effective responses.

SoSs and FoSs can be very complex. Now days, in evolutionary acquisition, many systems are using spiral development and thus their future behavior is unknown. (Not incremental development wherein their future behavior is known, and is just parceled-out into increments.) How do we integrate these evolutionary systems that are using spiral development into a SoS? Everyone is spiraling!

This is a complex situation. The approach to complex SoSE and FoSE is to surround and conquer. Apply a combination of top-down SE (forward engineering) and bottom-up SE (reverse engineering) using the Building Blocks and/or Vs. Apply the good ‘ole SE standards to the Building Blocks and/or Vs. Treat each Building Block and/or V as a system, and vice versa. Focus on developing and controlling the interfaces between system elements and external systems. This is what integration and interoperability are all about.

Put the interfaces under formal Configuration Management (CM). Form Interface Control Working Groups (ICWGs). Consider assigning the ICWG the overall responsibility for CM of the system elements and the external systems, not just their interfaces. Document, baseline, and CM what you currently know about the interfaces. Exchange that information with all sides of the interfaces. Control what you don't know by exception. For example, if unpredictable behavior occurs, and all sides of the interfaces agree, either alert the operator, record the data, apply artificial intelligence, or ignore the behavior.

Eventually, in time, the evolutionary behavior will settle down, become predictable, and quit spiraling. Until then, read and understand the SE standards and V-Models, apply good 'ole SE as evidenced in the SE standards and V-Models, and "Stand on the standards."

11 Conclusion

Is engineering a SoS really any different from engineering an ordinary system? Some believe that SoSE is "different" from SE, the SE processes are inadequate or insufficient for SoSE, and additional processes are needed. Others, like me, believe the SE processes as documented in the SE standards, and as illustrated in the V-Model and Dual-V Model, are a necessary and sufficient set of processes for SoSE, and no additional processes are needed. If you disagree, please get involved in the SE standards working groups and help us fix them.

12 References

[1] Sheard, S. 2006. Is Systems Engineering for "Systems of Systems" Really Any Different? *INCOSE Insight, Volume 9 Issue 1*, October 2006.

[2] IEEE 1220. 1994. *IEEE Trial-Use Standard for Application and Management of the Systems Engineering Process*. Copyright IEEE. http://shop.ieee.org/ieeestore

[3] IEEE 1220. 1998 and 2005. *IEEE Standard for Application and Management of the Systems Engineering Process*. Copyright IEEE. http://shop.ieee.org/ieeestore

[5] EIA/IS-632. 1994. *Systems Engineering*. Copyright © 1994, Government Electronics and Information Technology Association a Sector of the Electronic Industries Alliance. http://geia.org

[6] EIA-632. 1998. *Processes for Engineering a System.* Copyright © 1999, Government Electronics and Information Technology Association a Sector of the Electronic Industries Alliance. http://geia.org

[7] ISO/IEC 15288. 2002. *Systems engineering – System life cycle processes.* Copyright International Organization for Standardization (ISO), American National Standards Institute, 25 West 43rd Street, New York, NY 10036. (212) 642-4900, http://webstore.ansi.org.

[8] ISO/IEC 15288. 2008. *Systems and software engineering – System life cycle processes.* Copyright International Organization for Standardization (ISO), American National Standards Institute, 25 West 43rd Street, New York, NY 10036. (212) 642-4900, http://webstore.ansi.org.

[9] ISO/IEC TR 19760. 2003. *Systems engineering – A guide for the application of ISO/IEC 15288 (System life cycle processes).* Copyright International Organization for Standardization (ISO), American National Standards Institute, 25 West 43rd Street, New York, NY 10036. (212) 642-4900, http://webstore.ansi.org.

[10] *International Council of Systems Engineering (INCOSE) Systems Engineering Handbook*, Version 3.1, Section 2.4. August 2007, http://www.incose.org.

[11] *Defense Acquisition Guidebook (DAG).* 2006. U.S. Department of Defense (DoD).

[12] *Systems Engineering Guide for Systems of Systems*. 2008, U.S. Department of Defense (DoD)

[13] Forsberg, K. and Mooz, H. 1990. "*Proceedings of the First Annual NCOSE Conference;*" and Forsberg, K. Mooz, H. and Cotterman, H. 2000 "*Visualizing Project Management,*" John Wiley & Sons, Inc., provided with permission.

[14] The Center for Systems Management (CSM), Inc., www.csm.com, provided with permission.

13 Biography

JOHN O. CLARK is a Chief Engineer in the Information Systems Sector of Northrop Grumman. He is located at the Warfare Systems Engineering Department in Virginia Beach, Virginia. John currently supports the Information Systems Sector Directors of Process Management (SE Process) and Human Resources (SE Training). He led the development of and is the lead instructor for the INCOSE Certified Systems Engineering Professional (CSEP) course, and is an INCOSE CSEP. John has over 42 years experience applying SE and software engineering to the acquisition, development, verification/testing, operations, and support/maintenance of military command, control, communications, computer, intelligence, radar, sonar, electronic warfare, identification, weapon, network, scientific, and information systems. He is an active member of several Northrop Grumman Corporate Systems Engineering Advisory Group (SEAG) Working Groups and Communities of Practice; the Director of Education and Training of the INCOSE Hampton Roads Area Chapter; a member of the IEEE 1220 working group; a member of the EIA-632A GEIA G-47 SE committee; a co-lead of the GEIA G-47 Technical Reviews and Audits committee; and a member of the review teams for ISO/IEC 15288, 12207, TR 19760, TR 24748, and the INCOSE SE Handbook. He is an internationally recognized speaker and Subject Matter Expert in SE and teaches SE tutorials at major SE symposia. John earned a BS in Electrical Engineering from the Pennsylvania State University and an MS in Electrical Engineering from the State University of New York. He is an adjunct instructor in the MSSE curriculum at Old Dominion University in Norfolk, Virginia, and will be an instructor at Rutgers University.