

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

Based on "System of Systems Engineering and Family of Systems Engineering from a Standards Perspective," by John O. Clark which appeared in the IEEE International Conference on System of Systems Engineering, 2008. SoSE '08. Copyright © 2009 by IEEE.

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 aids this understanding, including the relationship between SE, SoSE, and FoSE. This knowledge helps us affect Systems and Software by fine tuning technology.

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 traditional 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 SoSE is "different" from traditional SE, the traditional SE processes just don't work for SoSE, and additional processes are needed. Others, like me, believe the traditional SE processes as documented in the SE standards and guides: IEEE 1220, EIA/IS-632, EIA-632, ISO 15288, and ISO TR 19760, are a necessary and

sufficient set of processes for SoSE, and no additional processes are needed.

The above standards and guides 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 article or in softcopies of the presentation. Refer to these documents 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.

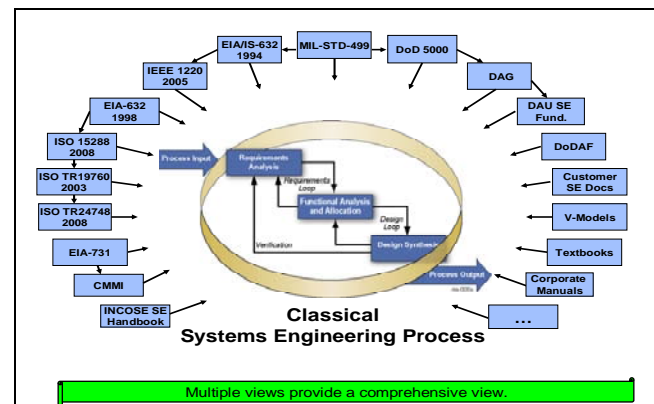


Figure 1. Systems Engineering Views

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. 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.

2. What is Different about SoSE and FoSE from Traditional SE?

SoSE and FoSE are an acquisition management problem, not a technical problem. The technical problem is solvable, but the acquisition management problem has not been solved. A few key 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 (INCOSE, 2007). To mitigate the risks inherent in these challenges, focus is placed on developing and controlling the interfaces between system elements and external systems. Developing and controlling interfaces correctly is what integration and interoperability are all about.

3. The Building Block

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

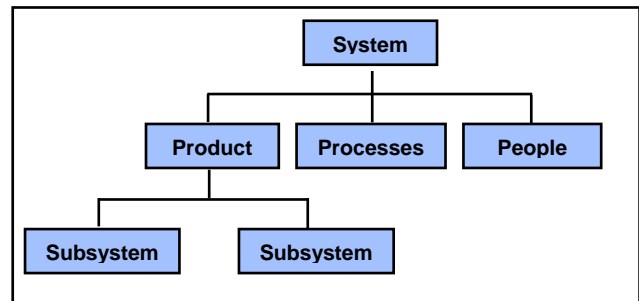


Figure 2. System Building Block

Next, the SE standards construct a system or SoS using these Building Blocks as shown in Figure 3.

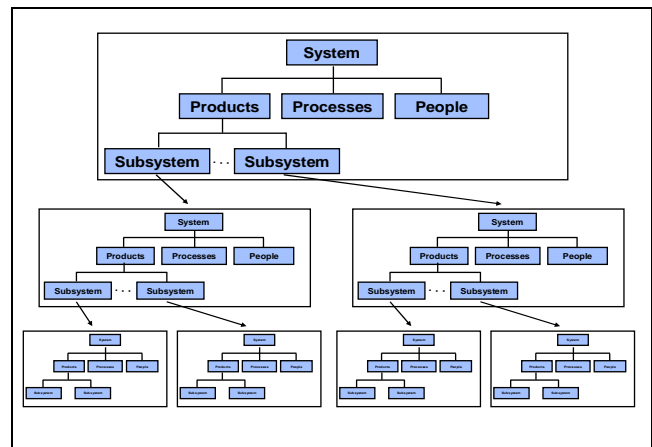


Figure 3. System of Systems Building Blocks

Each subsystem of the system or SoS is treated as a system in its own right. The Building Block structure continues on down the System Breakdown Structure (SBS) to the leaf-level that is needed to describe the SoS.

Other structures parallel the SBS such as the Work Breakdown Structure (WBS), the Specification Tree, and the Integrated Product Team (IPT) structure.

4. My Definition of SoS and FoS

Following are my simple definitions:

- 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

(Integrating systems can 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 Definition of SoS and FoS

Per the Defense Acquisition Guidebook [3], 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, 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.

6. The V-Model

Although not a standard, the V-Model is a very popular model of the SE process. The original V-Model [2] is shown in Figure 4.

The application of the V-Model to a 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.

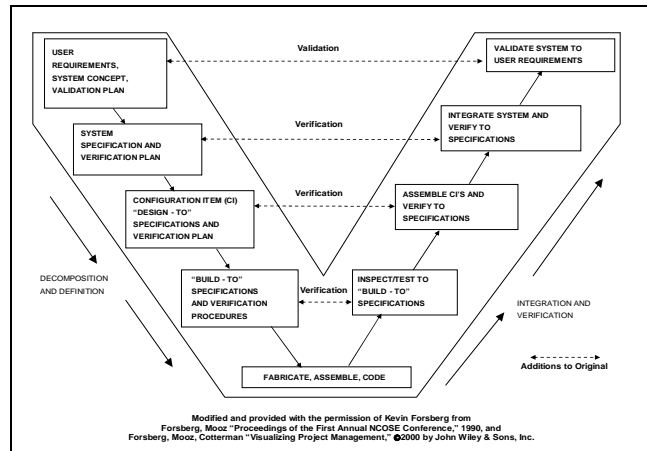


Figure 4. Original V-Model

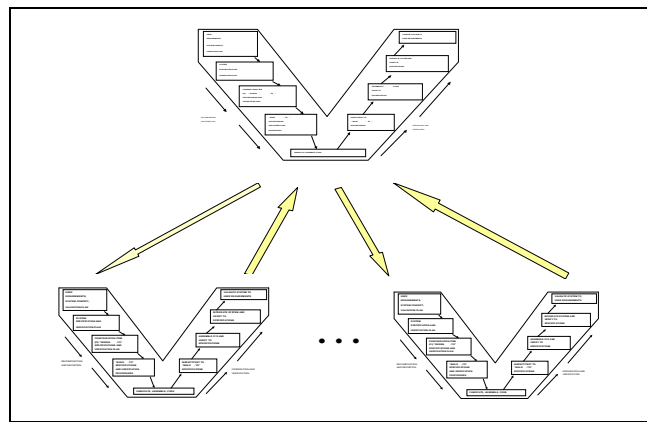


Figure 5. SoS V-Model

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 [4]. In this example 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 would be depicted in the backplane above the system.

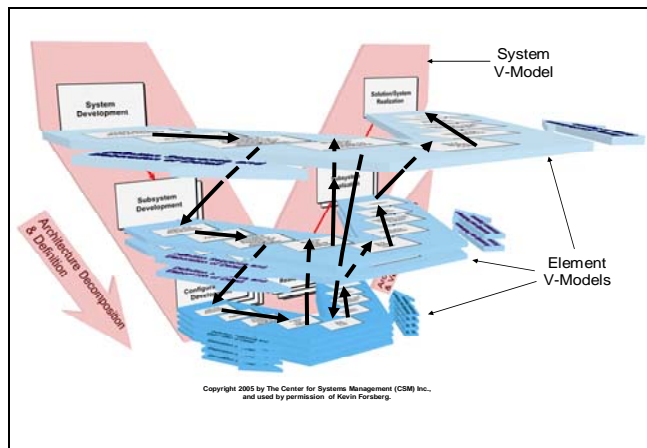


Figure 6. Dual V-Model

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, 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, 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 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.

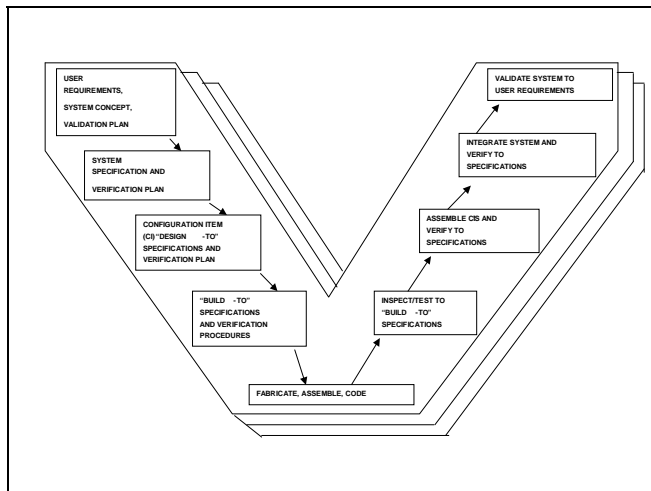


Figure 7. FoS V-Model

7. SoS Technical Baselines, Documents, and Reviews

System Baselines, Documents and Technical Reviews are shown in Figure 8 (the acronyms should be self-explanatory).

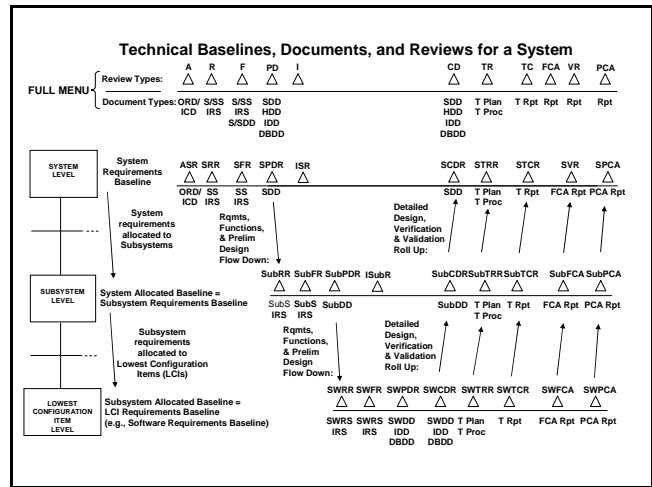


Figure 8. Technical Baselines, Documents, and Reviews for a System

The top group shows the full menu of Technical Baselines, Documents, and Reviews 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. 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.

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

Critical design, verification, and validation are shown on the right side. 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.

Extending Figure 8 to a SoS results in Figure 9. The same sequence of technical baselines, documents, and reviews applies. A SoS is just another system, albeit more complex, and should be treated as a system in its own right.

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, 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, apply good ole SE as evidenced in the SE standards, and "Stand on the standards."

10. References

- [1] Sheard, S. 2006. Is Systems Engineering for "Systems of Systems" Really Any Different? *INCOSE Insight, Volume 9 Issue 1*, October 2006.
- [2] 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.
- [3] *Defense Acquisition Guidebook (DAG)*. 2006. U.S. Department of Defense (DoD).
- [4] The Center for Systems Management (CSM), Inc., www.csm.com, provided with permission.

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

IEEE 1220. 1998. *IEEE Standard for Application and Management of the Systems Engineering Process*. <http://shop.ieee.org/ieeestore>

IEEE 1220. 2005. *IEEE Standard for Application and Management of the Systems Engineering Process*. <http://shop.ieee.org/ieeestore>

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

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>

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>.

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>.

International Council of Systems Engineering (INCOSE) Systems Engineering Handbook, Version 3.1, Section 2.4. August 2007, <http://www.incose.org>.

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>.

11. Biography

JOHN O. CLARK is a Chief Engineer in the Mission Systems Sector of Northrop Grumman. He is located at the Warfare Systems Engineering Department in Virginia Beach, VA. John currently supports the Mission 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 both an INCOSE CSEP and a Certification Application Reviewer (CAR). John has over 41 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; and a member of the review teams for ISO/IEC 15288, ISO/IEC 12207, ISO/IEC TR 19760, ISO/IEC 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 received a BS in Electrical Engineering from the Pennsylvania State University and an MS in Electrical Engineering from the State University of New York. He will become an adjunct professor in the MSSE curriculum at Old Dominion University in 2009.