Managing Requirements for a System of Systems

Ivy Hooks
Compliance Automation, Inc.

As we encounter more system of systems (SOS) and more complex SOS, we must consider the changes that will be required of our existing processes. For example, requirements management has been focused on a system or a product. This article looks at how the SOS has evolved, including what parts of requirements management apply to the SOS and where the process will need revision. It also discusses the need for dynamic scope for the SOS and more use of standards to interface the systems. Challenges definitely exist in SOS, not just in the Department of Defense but also in every aspect of the networked world. Our existing requirements management process is necessary, but not sufficient for the SOS.

There is much in literature depicting system of systems (SOS) [1]. I will use the characteristics Maier [2] has defined (in boldface below), followed by my summary of each.

1. **Operational Independence of the Individual Systems.** If you decompose the SOS, each component system can still perform independently of the others.
2. **Managerial Independence of the Systems.** Each individual system has its own purpose independent of the others and is managed separately for that purpose.
3. **Geographic Distribution.** Often individual systems are distributed over large geographic areas.
4. **Emergent Behavior.** The SOS performs functions not possible by any of the individual systems operating alone. The reason for developing the SOS is to obtain this unique behavior.
5. **Evolutionary Development.** An SOS is never finished; it continually evolves as needs change and newer technologies become available.

Maier defines an SOS as having all or a majority of these characteristics.

Evolution of SOS
The Past
In the first space systems, we built a system to do all of the functions simultaneously. The responsibility for the system fell under one organization, although the work may have been parceled out to many organizations. There was a central point of control. For example, when the National Aeronautics and Space Administration (NASA) built the Apollo space vehicle 40 years ago, NASA built all elements of the vehicle, its launch pad, and many other ground facilities.

When I toured the NASA Goddard Space Flight Center nearly 20 years ago, I questioned the need for dozens of different data processing systems – one for each satellite program. The person providing the tour had no idea why things were like they were, but I later talked to a NASA headquarters person who explained it very clearly. "Of course fewer systems would be better, but we can't take the risk. If Program A and Program B agree to share a ground data system and then one or the other gets cancelled, the remaining program will not have the funds for its data

> *"The SOS scope creates the vision and sets the bounds for what is to be accomplished. Scope includes the need, goals, and objectives for the SOS ... Additionally the SOS scope will need to address all the system-to-system interfaces within the SOS."*

processing system. To protect against this highly probable scenario, it's every man for himself." This can still happen.

The Present
Today, we have a number of existing systems that serve many other systems. These systems, e.g., Telemetry Data Relay Satellite System and the Global Positioning System (GPS), enable multiple new systems to accomplish their mission without reinventing the wheel or duplicating capabilities. Writing requirements to interface to these existing systems is generally straightforward, involving an understanding of what the new system must do to interface to the existing system.

Interfacing to a developing system where its design is evolving even as your design is evolving is much more difficult. In the automotive industry, with many computers under the hood of every vehicle, interfaces are a nightmare. One story I was told involved creating a new dashboard – an SOS comprised of entertainment, car information, temperature control, air bags, etc. The designer for the air bag system noticed that if anyone else sent a particular command on the bus, then the air bag would deploy. "But nobody would ever do that," he said. When the dashboard was assembled and an unsuspecting person moved the temperature control, the air bag deployed.

Managing Requirements
If you have not already, you will probably encounter an SOS in the near future. Although I wrote about managing requirements for single systems [3] without regard to the SOS, the basic principles apply. In fact, the basic activities shown in Table 1 are even more essential for managing an SOS than for a single system. In a single system, management is by a program or project manager. Requirements elicitation is the responsibility of system engineers or analysts who report to the program/project manager. In an SOS, these roles will need to be performed, but will be difficult organizationally. While using standards can benefit almost every system, standards may be essential for a successful SOS.

Strategic planning is essential for SOS development. The overall vision must be defined and embraced. Since an SOS does not have a limited life cycle but continues with the evolution of the SOS, its strategic plan must also evolve. The SOS capabili-

ties must evolve as needs change and new technologies become available.

SOS Scope

In product development, it is essential to identify the scope of the product before writing requirements. It is even more important to define the scope of the SOS before embarking on any aspect of requirements writing. The SOS scope creates the vision and sets the bounds for what is to be accomplished. Scope includes the need, goals, and objectives for the SOS. It also includes operational concepts for all life-cycle phases from the viewpoint of all stakeholders. Scope includes the external drivers, e.g., regulations and external interfaces. Additionally the SOS scope will need to address all the system-to-system interfaces within the SOS.

If we can identify the problem to be solved, then we can determine our need, goals, and objectives for the SOS. An example problem might be to obtain more accurate weather data using new technology in the following example:

- **Need.** Validate using the new technology to increase weather forecast accuracy.
- **Goals.** Fly instruments A and B to obtain information. Analyze information and make predictions. Compare predictions with and without the new technology.
- **Objectives.** Fly instruments using existing satellite and launch vehicle within 24 months. Obtain data for 24 months. Provide comparisons within two weeks of first obtaining data and biweekly thereafter.

Often we see processes that ask for requirements up front. Generally what is meant is that the need, goals, and objectives should be identified and operational concepts developed. It is premature to write requirements until all stakeholders have agreed on the operational concepts.

During the scope discovery process, there are a number of questions that must be answered (see Table 2). It can be beneficial to try to answer these questions initially to understand how much is known and what is unknown. If there are many unknowns, this will be a much more involved and drawn-out process. Significant engineering and analysis efforts, involving conceptual studies, trade studies, modeling, simulation, and prototyping, may be required to obtain the answers. The results of this effort may culminate in modified goals and objectives.

In the preceding example, the opera-

Requirements Basics	
Process	**Benefit**
• Define scope before requirements.	• Bound the problem/solution space.
• Develop operational concepts for the entire life cycle.	• Prevent requirements omissions.
• Identify stakeholders and involve them from the beginning.	• Prevent requirements omissions and misunderstandings.
• Identify external interfaces.	• Ensure the system works within the larger SOS.
• Educate all writers and reviewers on scope.	• Share the vision; prevent misinterpretations.
• Educate all writers and reviewers on what good requirements are.	• Get needed, clear, concise, and unambiguous requirements.
• Capture rationale for each requirement.	• Capture corporate knowledge and limitations imposed by existing systems.
• Capture verification method for each requirement.	• Think ahead to understand how to verify and to ensure verifiableness.
• Validate requirements as they are submitted.	• Reduce review time.
• Ensure each requirement is responsive to the scope.	• Avoid requirement and scope creep.
• Allocate each requirement to the next level.	• Ensure everything is allocated and required.

Table 1: *Requirements Basics*

tional concept might start with something like the following: We will launch the instruments using an *xyz* class launch vehicle out of the Kennedy Space Center. We will install the instruments in Company A's satellite using the satellite for power, pointing, and communications. We will spend three weeks doing instrument checkout on-orbit and this will include all communications to ground facilities. Following checkout, we will point the instruments per the data-gathering plan and begin taking data. This data will be downlinked daily by the satellite. In the analysis phase, weather forecasters will combine this with other data to make a forecast that will be compared to a forecast made without the instrument data. Both forecasts will be compared to the actual weather to determine the accuracy of the forecasts.

In this example, we have just barely started; however, once the questions in Table 2 are answered we have the next level of questions (see Table 3 on Page 6).

As difficult as gathering this information will be for a single system, it will be magnitudes harder for an SOS. In order for the SOS to succeed, these questions must be answered. The bounds for the SOS must be clearly defined, including operational concepts for each life-cycle phase and all transitions.

The list in Table 3 is a starting point. As you can see from our example operational concept, we have many other questions. Also, this SOS is planned for a four-year period at least: two years of development and two years of operation. We already know about ground data system changes that are going to need to be

accommodated over this time period. If it is successful, we may want to continue using these same instruments for a longer period.

It is not possible to overstress the need for full stakeholder participation and for full life-cycle coverage of operational concepts. Information is needed to feed planning, cost, and schedule estimating, and for developing complete requirements. This scope information must be provided to all stakeholders so they understand and agree to the scope before venturing forward, or the risks will be unmanageable.

Stakeholder buy-in to the SOS scope is essential for a successful SOS and for writing good requirements.

Since one of the characteristics of an SOS can be that of continual evolution, this implies that the scope will also evolve. This does not mean that we can just make it up as we go along. We need answers to the questions in Tables 2 and 3 to begin. We must understand the big picture,

Table 2: *Questions To Be Answered*

Questions To Be Answered
• What is the initial operational concept for a nominal operation of the SOS?
• How robust does the system need to be in response to off-nominal events?
• What is the high-level functional architecture of the SOS?
• Which systems of the SOS already exist?
• How might the existing systems evolve during the SOS life-cycle?
• How many new systems will need to be developed to meet the need?
• Who are all the stakeholders of the SOS across the entire life-cycle?
• When must the SOS be operational?
• How much money is available?

More Questions To Be Answered

- Who knows what about each existing system?
- What reliable documentation exists for each system?
- Who among the stakeholders can provide added information about their systems?
- How can we contact and work with these stakeholders?
- What are the interfaces that must exist between existing and planned systems?
- Who controls each interface?
- What standards have been used by existing systems that can enable the interfaces?
- What are the standards that may be available for new systems and future systems?
- What are the regulatory requirements that must be met in order to deploy the SOS?
- What changes can we anticipate over the life cycle?
- What evolution in technologies can we expect?

Table 3: *More Questions To Be Answered*

---

including the environment and the context in which the SOS must exist and operate. We need a plan from which to base our efforts – not just a random effort.

Management

It is not immediately clear after reading many SOS articles who actually manages an SOS effort for different endeavors. Success depends on someone being in charge. There must be an SOS manager, whether or not that manager has any direct control over the individual systems.

Likewise, it seems clear that there must be SOS architects and system engineers to facilitate the efforts for scope and requirements definition, to manage the interfaces, and to provide the sustaining effort essential to an evolving SOS. The organizational affiliations of these people may be many, and they may be selected for their experience with the systems that comprise the SOS.

SOS Requirements

With the scope clearly defined, we can now look toward writing requirements for the SOS. We are doing this with at least a conceptual architecture in mind and with operational concepts that incorporate existing, evolving, and new systems. The requirements for the SOS will reflect capabilities and performance implied by the scope results as well as the limitations and restrictions imposed by existing systems, standards, and regulations.

Allocation

Although requirements will be written to define the capability and performance of the SOS, there is really no such product.

Therefore, it is critical that each and every SOS requirement be allocated to an existing or new system or to an interface between two or more systems. The SOS manager is responsible for ensuring the acceptance of allocated requirements by each of the participating system managers (those managers of existing or new systems within the SOS).

There may be situations where we will use a system within the SOS but we will never work with the manager of that system; we will just use what is available as we do with many commercial products today. The SOS system engineers must anticipate and incorporate possible changes to such systems, e.g., Internet or GPS.

Rationale

For system managers to agree to the allocated requirements, they must completely understand each requirement allocated to them and its relationship to the SOS scope. By ensuring that rationale is provided with every requirement, this can be accomplished. In those cases where a requirement is allocated to more than one system, the affected managers must work with their counterparts to define the interfaces.

For example, the rationale might explain how the requirement is constrained by an existing system. The manager of that system can concur that the requirement is correct, or can state that he or she is planning changes that would invalidate the requirement. If the manager never sees the rationale, he or she may assume that the requirement is caused by someone or something else and never acknowledge the planned changes to his or her system.

The requirement will also provide information about how it relates to the scope so that as the scope evolves so will the requirement.

Standards

The number of standards is downright intimidating. Yet use of standards may hold the key to making systems work effectively together in an SOS. It is important to note the impact of standards on the development of cost-effective products. Having standards has enabled the hardware developers to grow and expand while reducing the price of goods to consumers. There are also standards for software, but fewer since software is a much younger field of engineering.

We rely on these standards when we buy a light bulb for our lamp, hook up our new computer to our existing printer cables, or turn on our laptop in airports

and hotels around the world. The Institute of Electrical and Electronic Engineers has thousands of standards to help engineers understand what other engineers are talking about [4]. The American National Standards Institute, the ISO [International Organization for Standardization], and the International Electrotechnical Commission are also providers of large numbers of international standards.

Standards are not static and unchanging; they are updated to account for changes in technology and needs. New standards are in development to fix perceived problems. In many areas, using standards is not a common practice, and the design approaches currently being taken will not effectively support an SOS. Thus more emphasis upon standards is essential to successful SOS development.

From the requirements arena, a standard is a requirement that is agreed upon and understood by a wide range of practitioners. Hardware and software designed to interface standards allow each side of the interface to build toward the middle without extensive negotiations and modifications as the two sides evolve. Using standards also enables us to unplug one system and plug in another that also meets the same standard interface.

Defensive and Self-Healing Requirements

One of the biggest examples of SOS complexity exists in our world of computer hardware and software. There are many brands of computers, operating systems, peripheral devices, device drivers, and applications. These all are expected to work together, but problems occur in getting all of these individual systems working together as a SOS.

I have had so many problems with my small personal world of creating a SOS from commercial off-the-shelf products that I cannot understand how anyone keeps large networked systems operational. With hundreds of workstations, networks, routers, etc., how does anyone dare to ever make any updates?

For a robust SOS, if my system and yours communicate or interact in any way, we need to both protect against what can happen at or across the interface. This begins with each of us defining requirements that will protect our integrity regardless of the other's system. We need requirements that defend against problems like those experienced by the car dashboard SOS. We need self-healing requirements for each system to enable its recovery from the impact of other systems.

There seems to be a desire or maybe even a mandate to merge new and existing systems from disparate organizations to achieve a new capability. This is often a long and tedious process and in many cases is simply abandoned in frustration. For example, the National Oceanic and Atmospheric Administration (NOAA), NASA, and the Navy agreed to do a joint project for developing new weather forecasting capability. NASA has had problems just having multiple NASA centers working together on a project, and this program interjected two other government agencies. NASA was responsible for developing the weather instrument. The Navy was responsible for a launch vehicle and a satellite to house the instrument. The NOAA, NASA, and the U.S. Navy all had a say in what the instrument was to measure and responsibility for ground stations to receive the data. A requirement that the instrument data had to be able to be processed by both NOAA and U.S. Air Force ground facilities only added to the program's complexity. NASA was named the project lead.

Attempts were made by the project team to develop operational concepts and write high-level requirements. While the team was able to document a lot of information, they had no formal training in developing operational concepts or writing requirements. This resulted in a system specification that contained very low-level requirements that constrained the instrument design. As the NASA team struggled to write requirements for their instrument, it was clear that they did not know the details for interfacing to the satellite or launch vehicle. These interfaces and the environments expected by and imposed by the other systems are critical to writing the instrument requirements; the Navy was unresponsive to providing the data. With unintentional constraints from above and lack of information, it was impossible to write good instrument requirements.

This is not an uncommon situation; it happens all too often on many government programs.

The end of the saga was that the Navy bailed out and withdrew its support; NASA continued instrument development to interface with an unknown launch vehicle and satellite, and hardware and software now exist. Will it ever be deployed? That is a good question.

Conclusion

Requirements management will be an important aspect of an SOS, as it is to all systems. It is critical to do the prerequire-ments work of scope definition, documentation, dissemination, and stakeholder buy-in before the SOS requirements are developed. Management of requirements allocation will be a major activity more than in single-system activities. Issues related to allocation, interfaces, and information transfer must be high on management's agenda and resolved swiftly. Extended use of standards and development of standards to empower an SOS may be key to successful endeavors. Each individual system must pay more attention to its defensive and self-healing requirements to participate in future SOS.◆

References

1. Sage, A., and C. Cuppan. "On the Systems Engineering and Management of Systems of Systems and Federations of Systems." Information, Knowledge, and Systems Management 2 (2001): 325-345.
2. Maier, M. Architecting Principles for Systems of Systems. Proc. of the Sixth Annual International Symposium, International Council on Systems Engineering, Boston, MA, 1996: 567-574.
3. Hooks, Ivy F., and Kristin A. Farry. Customer Centered Products – Creating Successful Products Through Smart Requirements Management. New York: Amacom, 2001.
4. Vonderheid, E. "Standards Hidden in Plain Sight." The Institute Mar. 2004.

About the Author

Ivy Hooks is president and chief executive officer of Compliance Automation, Inc., a company whose focus is requirements. Hooks is an internationally recognized expert, author, and speaker on the subject of requirements. She brings experience from a 20-year career at the NASA Johnson Space Center followed by 20 years experience consulting for government and industry.

Compliance Automation, Inc.
1221 South Main ST
STE 204
Boerne, TX 78006
Phone: (830) 249-0308
Fax: (830) 249-0309
E-mail: ivy@compliance
automation.com