System of Systems Software Assurance

Introduction

Under DoD sponsorship, the Software Engineering Institute has initiated a research project on system of systems (SoS) software assurance. The project's overall goal is to provide appropriate methods and tools to meet the assurance challenges of systems of systems. To focus the research, the project is specifically addressing SoS assurance issues that arise in large-scale adaptive information management and command/control systems of systems. The key research question is determining what types of evidence (and associated argumentation) are needed to achieve justified confidence that SoS behavior (with respect to quality attributes such as reliability, availability, performance, or[1] security) will be acceptable when the SoS is used in its actual and evolving usage environments.

This paper describes the premises underlying our research approach as well as the specific research activities we are undertaking.

Background

The project's focus is on systems of systems, whose salient characteristics, for the purpose of understanding why SoS assurance presents particular difficulties, are a subset[2] of those defined by [Maier 1998]:

- Managerial independence: A system of systems' constituent systems are separately acquired and integrated.
- Operational independence: The constituent systems are independently useful; they maintain a continuing operational existence independent of the system of systems.
- Emergent behavior: The principle behaviors of a SoS *emerge* from potentially dynamic combinations of its constituent systems to create a SoS capability or property (quality attribute).
- Evolutionary development: A system of systems comes into existence gradually and continues to change, "with functions and purposes added, removed, and modified with experience." [Maier 1998]

We next discuss how these characteristics make it especially difficult to establish confidence in overall (i.e., emergent[3]) SoS quality attributes such as security, performance, reliability, and availability.

- Managerial independence: Because a constituent system maintains its existence independent of the SoS, its features and capabilities are typically modified on schedules that may not mesh well with expectations of other constituent systems. For example, a new SoS capability that requires coordination among constituent systems may be difficult to implement in an assured way since various constituent capabilities may come online at different times. While the design of the new SoS capability might be sound once all the constituent systems are upgraded, the interim states in which some upgrades are available and others are not may lead to unexpected failures. It follows therefore that a SoS assurance approach must take into account a potential inability to control the SoS configuration.

[1] The use of "or" is intended to imply that we do not expect to address all of these quality attributes simultaneously, but instead, will focus on those that allow research progress to be made.
[2] The other characteristic defined by Maier is "Geographic Distribution." This characteristic implies that the relevant properties of the system are defined by the transfer of information rather than the transfer of mass or energy. Since we are already dealing with information systems, this is not a characteristic that presents additional SoS assurance difficulties.
[3] We call these *emergent* SoS attributes because they arise from interactions of constituent system behaviors.

- Operational independence: Because constituent systems are independently useful, each constituent system meets a potentially different mix of user expectations and demands, e.g., different demands for reliability, different requirements for speed in updating vs. retrieving data, etc. Systems of systems are created to meet user demands that cannot be met by constituent systems acting alone; by definition, constituent system interactions are needed to meet these demands, and these interactions may place new expectations on some constituents. Hence, when a constituent system is part of a system of systems, it will likely need to respond to a different spectrum of user demands than was initially considered in the constituent's design. A SoS assurance approach therefore needs to consider how to evaluate the effect of new populations of users and user demands on constituent systems and their interactions.
- Emergent behavior: A SoS capability, by definition, makes use of the capabilities of more than one constituent system to meet a demand. Similarly, SoS quality attributes emerge from constituent system interactions. For example, SoS reliability is typically independent of constituent system reliabilities—the SoS may be more reliable than its constituent systems because the SoS has more backup capabilities than any individual constituent, or it might be less reliable because of problems that arise in passing information among constituents. It is not sufficient to evaluate just constituent system requirements and behaviors relevant to reliability, security, etc.; interaction effects in the SoS as a whole must also be considered.[4] This evaluation of emergent properties is made even more difficult when considering that different configurations of constituent systems may emerge dynamically in response to user demands or needs.
- Evolutionary development: The gradual evolution of a SoS means that assurance evaluations are not done once and for all but instead must evolve as the SoS evolves. The addition of new constituent capabilities in support of new SoS capabilities may lead to new SoS failure modes; SoS assurance methods and conclusions must be robust in face of these changes.

Maier also categorizes systems of systems according to the degree of managerial control that is exercised over the SoS and its constituents. He proposes three categories of SoS, and the DoD "Systems Engineering Guide for Systems of Systems" [OUSD 2008] proposes a fourth:

- A *directed* SoS is "built and managed to fulfill specific purposes. It is centrally managed during long term operation to continue to fulfill those purposes and any new ones the system owners may wish to address. The component systems maintain an ability to operate independently, but their normal operational mode is subordinated to the central managed purpose. For example, an integrated air defense network is usually centrally managed to defend a region against enemy systems, although its component systems may operate independently." [Maier 1998]
- An *acknowledged* SoS has "recognized objectives, a designated manager, and resources for the SoS; however, the constituent systems retain their independent ownership, objectives, funding, and development and sustainment approaches. Changes in the systems are based on collaboration between the SoS and the system." [OUSD 2008]
- A *collaborative* SoS is one in which "the central management organization does not have coercive power to run the system. The component systems must, more or less, voluntarily collaborate to fulfill the agreed upon central purposes." [Maier 1998] The Internet is considered an example of this type of SoS, since standards are proposed but there is no enforcement mechanism *per se*.

[4] As another example, a constituent system's performance may be considered adequate when the constituent system and its usage is considered alone, but in a SoS context, the constituent system's use of shared resources may lead to unsatisfactory SoS performance for some sets of SoS users.

- A *virtual* SoS lacks "a central management authority and a centrally agreed upon purpose. … Large-scale behavior emerges—and may be desirable—but this type of SoS must rely upon relatively invisible mechanisms to maintain it." [OUSD 2008]

Because our research project is focused on meeting DoD's SoS assurance needs, we seek to address the SoS assurance issues that arise in DoD systems of systems, and these are primarily of the directed and acknowledged types.

This research project was initiated in October 2007. During 2007-2008 we interviewed test and evaluation (T&E) personnel in DoD to see what SoS problems are currently being experienced and what future problems may be of concern. The results of this survey were presented at the 2008 Systems and Software Technology Conference [Goodenough 2008]. The overall finding was that once today's systems are deployed, usually within a SoS environment, there are not very many critical failures, i.e., failures with critical effects and *no workarounds*. However, the increasing size and complexity of command and control systems is straining DoD's T&E process: in particular, it is taking too long to get constituent systems released, much less systems of systems. Moreover, T&E personnel are also concerned about their ability to properly evaluate systems of systems composed of an increasing number of constituent systems. They have difficulty in doing evaluation today in a timely manner. There is great concern that without improvements in assurance approaches, there will be increasing delays in fielding systems of systems and, if the T&E process is inadequate in evaluating SoS interactions, systems of systems will experience an increasing number of serious problems after release.

As a result of this survey and other inputs, we developed a SoS software assurance research rationale and agenda, which are presented in this note.

Assurance Definition and Focus

A DoD definition for *system assurance* is the following:

> System assurance is *the justified confidence that the system functions as intended and is free of exploitable vulnerabilities, either intentionally or unintentionally designed or inserted as part of the system at any time during the life cycle* [NDIA 2008]

For our purposes, the most relevant parts of the definition are "justified confidence" and "functions as intended." Although the definition is applicable to either a system or a system of systems, our focus is on *software's* contribution to *system of system* assurance, i.e., on obtaining justified confidence that software's contribution to overall SoS behavior is assessed adequately. In this project, SoS software assurance focuses on how well software supports a SoS mission capability or makes the SoS more robust against various threats.

Most of the words in this definition concentrate on information assurance and security concerns, but we interpret the definition broadly. The definition uses the criterion that a system (in our case, a SoS) "functions as intended." Although the phrase "functions as intended" suggests a focus on just the functions provided by the SoS, "as intended" necessarily includes such quality attributes as performance, availability, reliability, security, etc. No one intends that a SoS function with *poor* availability, reliability, security, etc. Our approach to SoS assurance pays special attention to developing justified confidence in these so-called "non-functional" system of systems properties.[5] We also interpret "functions as intended" to include not just the intent as specified in the original SoS requirements and as perceived by the SoS designers, but also the evolving intent of actual SoS users. In particular, because of their scale, SoS user intent as well as the SoS environment will almost certainly differ from what was considered in the original SoS requirements. Moreover, SoS user intent and the SoS environment will change over time. Despite these changes, we want to have justified confidence

[5] Quality attributes and non-functional properties are different terms for the same notion.

that actual SoS users do not experience unacceptable SoS behavior, even when the usage of constituent systems changes in unexpected ways or a constituent system is required to operate under unusual or unplanned environmental conditions.

Finally, the definition states that the goal is to have *justified* confidence that a SoS functions as intended. With respect to systems of systems, this implies developing:

- An improved understanding of what constitutes *justification* for having confidence that a system of systems will behave in acceptable ways. In particular, we must justify conclusions about expected SoS behavior when it is used operationally, under conditions that were not part of any test regimen.
- An improved understanding of what needs to be *measured* to justify confidence, how to measure it, and how to understand the significance of what is measured.

Principles and Assumptions

This project is based on a key premise, namely, that for systems of systems, testing is inadequate to develop justified confidence in a SoS's acceptable behavior under every condition of actual use. Evidence and analysis in addition to test results is needed.[6]

Much on-going work in software assurance is focused on defect prevention and detection. Such work is premised on the idea that software is perfectible. Eliminating defects is obviously important and valuable but is not sufficient as an approach to providing justified confidence in SoS behavior because in practice, not all defects can be eliminated, particularly considering the scale of systems of systems and the dynamic evolution of their usage patterns, constituents, and environments. Additional effort is needed to evaluate and ensure that a system of systems will minimize the impact of potential failure modes due to the presence of undetected defects and vulnerabilities and continual usage and environmental changes. Our focus is on this more neglected aspect of information system assurance, namely, assuring the acceptable operation of systems of systems despite the presence of defects, vulnerabilities, unanticipated usage, and environmental conditions.

This view of the problem leads us to focus on what might be called *robustness assurance*, i.e., assuring the acceptable operation of systems of systems despite the failure of certain assumptions underlying their design and operation. Lest this seem an impossible problem, note that safety engineers evaluate systems routinely to determine their sensitivity to certain assumptions or failure modes. They design such systems to detect impending failures (from potentially unknown causes) and to respond in ways that preserve essential system functions. This requires an architectural design approach as well as analysis of key hardware and software failure modes relevant to acceptable system operation. In particular, a safety evaluation depends on more than just testing. Aspects of the safety engineering approach need to be adapted to the SoS context.

Robustness analysis has not been a priority for complex information management systems such as battlefield information systems;[7] for example, architecting, designing, and testing for security (including recovery from breaches) is often funded only late in development (in directed systems of systems), but architectural security analysis is critical in the system of systems context, where it is clearly impossible to test for all possible combinations of system configurations and all combinations of user demands.

In summary, complex adaptable systems of systems are increasingly important to DoD. Such systems include

[6] Many of these ideas are based on the SEI's report on ultra-large-scale systems [Northrop 2006].
[7] Robustness analysis *has* been a priority for safety critical systems and for systems that cannot easily be fixed when flaws are discovered, e.g., space and satellite systems. Fairly standard analysis and fault tolerance approaches are used for these types of systems, and the approaches are applied at great expense. The issue is how such approaches can be applied economically at the scale relevant to today's emerging SoS-based information management systems.

battlefield information systems, systems using a SOA approach, and in general, large-scale systems of systems in which flexible operation and continuing evolution are important. Such systems of systems present increasingly difficult technical problems when it comes to gaining *justified* confidence that they will operate acceptably in the field. This program of research addresses these problems.

Research Thrusts

The project has four research thrusts covering a range of near-term to long-range technical and transition goals.

Thrust 1: Interoperability Assurance

The official DoD definition of interoperability is: "The condition achieved among communications-electronics systems or items of communications-electronics equipment when information or services can be exchanged directly and satisfactorily between them and/or their users." [JP1-02 2008] Another DoD definition in common use is "The ability of systems, units, or forces to provide services to (and accept services from) other systems, units, or forces and to use the services so exchanged to enable them to operate effectively together." [DACS 2008] According to either definition, since SoS constituents must be able to exchange information to support a desired SoS capability, interoperability assurance is a key aspect of SoS assurance.

The objective of this thrust is to determine how to make systems of systems more robust against the consequences of failure to exchange information properly among SoS constituents. Our approach is to first examine examples of such failures to determine what architecture robustness approaches could make constituents of systems of systems more robust against such interoperability failures. Because the overall project is concerned with developing justified confidence in SoS properties, this effort is also examining interoperability assurance practices to determine what types of evidence are needed to provide greater confidence in constituent system interoperability, given the fact that it is difficult (or impossible) and time-consuming to test every interoperable combination. We will be producing reports analyzing interoperability failures and possible robustness solutions as well as analyzing effective interoperability evaluation practices. We view this thrust as having near-term impact since we are studying current problems and practices, but the study will provide a basis for a better understanding of key underlying technical issues that must be addressed. We will also be noting acquisition and procedural barriers to putting good interoperability assurance practices in place.

Thrust 2: Release Decision Analysis

The objective of this thrust is to improve the soundness and timeliness of SoS release decisions by combining assurance information collected from various sources and phases of development. This thrust is focused on determining how a variety of design, analysis, *and* test information can be combined to develop increased and justified confidence in SoS behavior.

Thrust 3: Compositional Robustness Evaluation

This is a speculative research thrust aimed at identifying novel ways of predicting and bounding unacceptable SoS behavior, e.g., by use of non-monotonic logic frameworks or the development of robustness guards that limit the scope of unacceptable end-to-end system interactions. This work is in its formative stages.

Thrust 4: Transition and Collaboration Analysis

The objective of this thrust is to better understand barriers and incentives for transitioning new assurance

technology into practice. Our initial approach is to identify funders, developers, and users of new assurance technology and their motivations (value exchanges) for supporting the creation and adoption of assurance technology. Our initial work focuses on identifying organizational interactions supporting the development and adoption of assurance technology in the security assurance domain. This overview of assurance technology developers and users will be helpful in determining the most effective transition paths for technology developed by the other research thrusts.

Contact

For more information, contact John Goodenough (jbg@sei.cmu.edu) at the Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA.

Acknowledgements

This paper was greatly improved due to comments by Phil Boxer, Mark Klein, Carol Sledge, Chuck Weinstock, and Carol Woody.

References

[DACS 2008]
The Data and Analysis Center for Software (DACS), "Software Acquisition Gold Practice: Ensure Interoperability." https://www.goldpractices.com/practices/ei/index.php

[OUSD 2008]
Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering. "Systems Engineering Guide for Systems of Systems," Version 1.0. Washington, DC: ODUSD(A&T)SSE, 2008, http://www.acq.osd.mil/sse/docs/SE-Guide-for-SoS.pdf

[Goodenough 2008]
John B. Goodenough. "Reports from the Field: Evaluating the Readiness of Complex Software-Intensive Systems," **Systems and Software Technology Conference,** May 2008.

[JP1-02 2008]
Department of Defense, "Department of Defense Dictionary of Military and Associated Terms," Joint Publication 1-02, 12 April 2001 (as amended through 17 October 2008), http://www.dtic.mil/doctrine/jel/new_pubs/jp1_02.pdf

[Maier 1998]
Mark W. Maier. "Architecting Principles for Systems of Systems," **Systems Engineering 1**, 4 (1998): 267–284.

[NDIA 2008]
National Defense Industrial Association System Assurance Committee. "Engineering for System Assurance," Version 1.0, 2008, http://www.acq.osd.mil/sse/ssa/docs/SA-Guidebook-v1-Oct2008.pdf

[Northrop 2006]
Northrop, L., et al., "Ultra-Large-Scale Systems: The Software Challenge of the Future," Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006, http://www.sei.cmu.edu/uls/files/ULS_Book2006.pdf