Complexities of System of Systems Operational Availability Modeling

Dennis J. Anderson, Sandia National Laboratories

Charles M. Carter, Sandia National Laboratories

Tamara Brown, Sandia National Laboratories

Key Words: system of systems, operational availability, repairable systems, reliability modeling, modeling and simulation

SUMMARY & CONCLUSIONS

Calculating operational availability (A_o) for a system of systems (SoS) presents unique challenges to reliability, availability, and maintainability (RAM) assessment, modeling, and analysis. System interdependencies and complex interrelated sustainment operations that exist in a SoS present complexities that must be accounted for in calculating or estimating A_o for the SoS. These system interdependencies affect the operating, operable, and down times of the individual systems.

Both system-level and SoS-level A_o performance must be assessed within the SoS context for logistics and planning purposes. However, metrics calculated for the individual systems as part of the SoS may not be appropriate for assessing the individual system performance against their individual system requirements. In most cases, simulation modeling is required to capture the complex operating, operable, and down time hours of a SoS and the systems in the SoS, and to accurately aggregate the individual system availabilities to higher SoS levels. This paper explores some of the complexities involved in SoS A_o modeling and presents SoS simulation results from a modeled SoS application.

1 INTRODUCTION

In today's increasingly technological world, combinations of systems often function together as a SoS to accomplish an objective. A variety of definitions exist for SoS, many of which are not specific enough to present possible SoS metric definitions and calculations. For purposes here, a SoS is defined to be a set or arrangement of interdependent systems, each performing a defined task or mission, in which at least one system is dependent on one or more other systems. System-of-systems-level performance is emerging and cannot be assessed by assessing individual system performance separately, except for simple cases where the systems operate (and are maintained) independently of each other. System dependencies can be of varying complexity, including sequential or parallel system operations, system functional redundancies provided by other systems, R of N systems operating, and combinations of these. These system interdependencies can be further complicated by complex interdependencies with support systems and operations.

The mathematics and logic for analyzing a SoS may seem similar to analyzing the components of a single system, which may be true for some simple cases. More often, the system-to-system interdependencies are difficult to account for. Components in systems typically operate on the same schedule and the dependency effects are usually consistent. If the system fails, the remaining parts stop operating. For a SoS, the interactions among the systems are almost always more complicated than the interaction of parts within a system.

Systems of systems are becoming more prevalent than ever with increased use of autonomous systems and network-centric functionality. If operations, maintenance, and sustainment of the systems will be performed at the SoS level and for the intended SoS operational environment, assessing SoS A_o performance is imperative.

1.1 SoS A_o

For a system or a SoS, A_o is described as the percentage of time the system or SoS is operating or operable (up) over a specified time interval involving defined system and/or SoS operations. A_o is often viewed as being a straightforward metric, but even for a single system, its calculation can involve numerous complexities, subtleties, and assumptions. A_o is calculated, estimated, and modeled for a specifically defined collection of hardware performing specifically defined operations over a specifically defined timeframe, with specifically defined reliability and maintainability operational performance characteristics and specifically defined sustainment assumptions. Effects of system interdependencies in a SoS on the operating times and down times of the individual systems can be complex and may not be apparent during individual assessment and testing of the systems.

One complexity present in calculating and modeling A_o for a SoS is the definition of "up" for the SoS. Often, not all individual systems in a SoS are required for all segments of the SoS operational mission. Instead, the SoS operational mission involves a sequence of segments using various combinations of systems at various times. The SoS may be considered up, but individual systems may be operating, operable, or even down. In fact, the criteria that define the SoS as "up" may change throughout a scenario.

978-1-4799-6703-2/15/$31.00©2015 IEEE

The interaction of systems operating in a SoS also affects the operating, operable, and down times of individual systems. System of system interdependencies can result in failures of one system in the SoS producing additional operable time for other systems that depend on the failed system to operate. A poor performing system in a SoS may actually result in higher system A_o for other systems, considered in the context of the SoS, that are spending additional time in an unutilized condition. For this reason, A_o and operating, operable, and down times for individual system performance in the context of a SoS will differ from A_o calculated for the individual systems individually. Competition for limited resources, such as maintainers, parts, and special handling equipment, can also affect system down times that similarly increase operable times of the other systems in the SoS. Additionally, A_o is defined for specific scenarios or mission durations, and it is not uncommon that the scenarios/missions defined for the SoS A_o requirements are not in sync with the individual system A_o scenarios.

1.2 Equation-Based Approach to SoS Ao

Occasionally, a system of systems is simple enough that an equation can be used to determine steady-state SoS A_o. These equations have limited applicability that must be understood to properly apply them. The simplest case is one with multiple systems that are all required for the mission, are 100% utilized, and operate independently. In this case, availability of each system is independent of the status of the other systems. This situation can be expressed mathematically as:

$$P(SoS\ Up) = P(Sys_1\ Up) \bigcap P(Sys_2\ Up) \bigcap \ldots P(Sys_n\ Up) \quad (1)$$

Since the systems are independent,

$$P\big(SoS\ Up\big) = P\left(Sys_1\ Up\right) * P\left(Sys_1\ Up\right) * \ldots P\left(Sys_n\ Up\right) \quad (2)$$

If all systems operate independently and all systems are required at all times, then calculation of the SoS A_o is the product of the individual system operational availabilities.

$$A_{O_{SoS}} = \prod_{i=1}^{n} A_{O_i} \quad (3)$$

Another case for which SoS A_o can be determined using an equation is the case of series systems that are all dependent. In this case, the systems operate together with a combined failure rate that is the sum of the failure rates of the individual systems and the mean time between failures (MTBF) of the SoS is the inverse of the sum of the individual system failure rates, λ_i:

$$\mathrm{MTBF}_{SoS} = \frac{1}{\sum_{i=1}^{n} \lambda_i} \quad (4)$$

For this case, down times from system failures do not overlap and the mean down time (MDT) of the SoS is a failure rate weighted average of the individual system MDTs:

$$\mathrm{MDT}_{SoS} = \frac{1}{\sum_{i=1}^{n} \lambda_i} * \sum_{1=1}^{n} \lambda_i * MDT_i \quad (5)$$

An equation estimator of SoS A_o can be expressed as:

$$A_{O_{SoS}} = \frac{MTBF_{SoS}}{MTBF_{SoS} + MDT_{SoS}} \quad (6)$$

Substituting the above equations for MTBF and MDT and performing some simple algebra yields the following equation, where MDT_i and $MTBF_i$ are the MDT and MTBF of the individual systems:

$$A_{O_{SoS}} = \frac{1}{1 + \sum_{i=1}^{n} \frac{MDT_i}{MTBF_i}} \quad (7)$$

This equation can be further developed to make A_o of the SoS a function of the individual system A_o values. If A_{O_i} is defined as the A_o of system *i* operating independently with 100% utilization, A_{O_i} is expressed as:

$$A_{O_i} = \frac{MTBF_i}{MTBF_i + MDT_i} \quad (8)$$

Equivalently, after some algebra:

$$\frac{MDT_i}{MTBF_i} = \frac{1}{A_{O_i}} - 1 \quad (9)$$

If all systems in the SoS are in series in a dependent relationship so that all other systems stop operating when an individual system fails, A_o for the SoS can be expressed as

$$A_{O_{SoS}} = \frac{1}{1-n+\sum_{i=1}^{n} \frac{1}{A_{O_i}}} \quad (10)$$

Both cases described above are simpler than those typically encountered in a real SoS. Complicating factors that negate the use of these equations include:

- Systems within the SoS not all in series
- Systems not all independent or all dependent
- Systems not all operating all of the time
- Logistics delays associated with down time
- Limited spares
- Short duration operations where Ao has not reached steady-state

1.3 System and SoS A_o Modeling and Analysis Approach

In most cases, simulation modeling is required to capture the complex operating, operable, and down time hours of a

SoS and the systems in the SoS, and to accurately aggregate the individual system availabilities to higher SoS levels. The System of Systems Analysis Toolset (SoSAT) simulation, developed by Sandia National Laboratories, has been applied to a number of US Army, Navy, Air Force, and Marine Corps availability and logistics modeling efforts, including several SoS applications. The SoSAT v2.0 simulation has been verified and validated and the availability calculations and algorithms have been officially accredited by the Army. SoSAT was used to model a SoS application and generate the SoS results presented below.

The SoSAT is a set of tools centered on the simulation capability and includes relational databases, input and output interfaces, system reliability models, state modeling, and SoS functional dependencies and redundancies. The SoSAT simulation is a multi-system, discrete event, stochastic simulation capability being developed and applied to model and analyze complex SoS capabilities and performance. Modeling a SoS, with up to thousands of systems, each represented by up to thousands of components, for multiple metrics at up to brigade levels and higher, and for up to lifecycle durations, presents significant modeling and computational challenges. Sandia integrated reliability, availability, supply chain, and state modeling concepts into SoSAT based on research and applications experience with complex, high-consequence systems.

SoSAT provides the capability to model systems, components, collections of systems in organizational structures (such as an Army brigade), and multiple organizational structures operating over time. Missions to be modeled can range from small, high-utilization missions to longer-term missions up to peacetime training and lifecycle timeframes. SoSAT models system operational performance and system reliability and maintainability along with detailed repair, supply, and sustainment operations, including competition for resources. For SoS performance modeling, SoSAT models user-defined functional dependencies and redundancies that comprise SoS-level performance, where system performance can be dependent on the performance of other systems, subsystems, and conditions.

2 SoS COMPLEXITIES

The operating profile of an example SoS, with four systems that are designed to operate over a week-long period, is shown in Figure 1. For this example SoS, System A and System B are completely dependent on System C and vice versa, and System D runs independently and continuously for the mission duration. This SoS operational mission was based on an actual scenario where a helicopter (System C) was necessary for the operations of two other systems, which are payloads (Systems A and B).

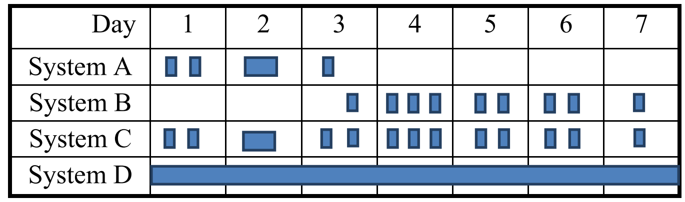

Figure 1. SoS Schedule

2.1 SoS Availability

This four-system SoS may be considered somewhat simple, but even this SoS presents questions to be addressed pertaining to A_o interpretation.

If System A experiences early failures, is the SoS mission profile still executed as planned? In some cases, the mission profile is shifted to the right due to System A failures, extending the schedule beyond the original seven days. For this example, it is assumed that the schedule will be executed as stated if the systems are capable of operating; if a system is scheduled to operate but cannot because it is down, that operating time will be missed and the schedule will continue.

What systems are required to be up for the SoS to be up? If failure of any individual system will cause a SoS-level failure, the corresponding reliability block diagram (RBD) is shown in Figure 2. However, this diagram is not necessarily accurate from an availability perspective.

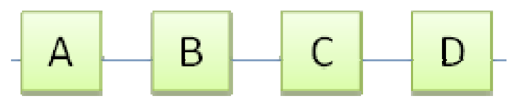

Figure 2. Four-System SoS Series RBD

What happens if a system fails near the end of its required segment? As an example, if System A fails on Day 3 and is down for 24 hours, what happens depends on the scenario being modeled. If the SoS is a ship that is transiting through a certain area with a threat that requires System A for the first portion of the mission, but because of changing conditions needs System B after a certain point, then having System A available matters for part of the mission but not all of the mission. The RBD is dynamic and changes over the course of the period of interest, as depicted in Figure 3. Once there is no longer a need for system A, the SoS could be defined as up even though System A is down.

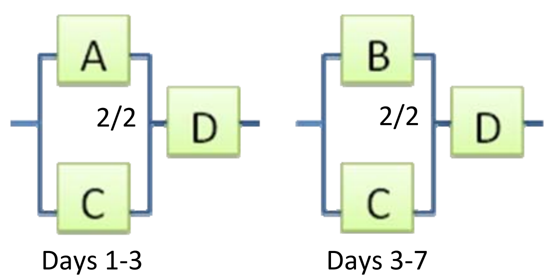

Figure 3. Four-System SoS Dynamic RBD

2.2 System Availability

For individual system A_o when the system is part of a SoS, some ambiguity exists about the time period of interest. The systems that make up the SoS often have individuals or organizations primarily responsible for that system performance. When reporting A_o for System A, the following questions need to be considered:

- Does the system availability pertain to the entire operational mission?
- Is the system A_o needed during only the portion of the mission that the system has an active role? (For System A in our example, the active period is from the start of the mission through midday on Day 3.)
- Is the system A_o needed specifically during the segments when the system is scheduled to operate?

In many cases, the answer to all these questions is “yes” and additional statistics need to be calculated during the simulation to address them.

2.3 System Dependencies

In the example SoS, Systems A and C are used together, as are systems B and C. For any SoS, it is necessary to ask which systems are dependent on other systems. For the SoS the example is based on, if the helicopter (System C) is in a failed state, the payloads (Systems A and B) cannot be used, and if a payload is down, the helicopter will not begin or continue operations. Equation- or simulation-based SoS models have to account for the period of non-usage due to dependent systems. For the example, System D operates independently and continuously. This mix of dependent and independent systems makes solving for system and SoS performance difficult.

In some cases, dependencies may occur for operational reasons rather than functional reasons. For example, if System D is completely unrelated to the other systems, it can operate without A, B, or C, and vice versa. However, when System D fails, the repair procedures may vary depending on the failure mode. If System D is repairable in the field, Systems A, B, and C can continue to operate as scheduled. However, sometimes System D cannot be repaired in the field and requires the SoS to return to an area where repairs for System D can occur, during which normal activities are suspended. This type of interaction is not uncommon and adds additional complexity to modeling a SoS. Analysis of a system with 10 parts, or even 100, can often be accomplished with a spreadsheet. A SoS comprised of 10 unique systems can be rife with interactions that require special handling.

3 DEPENDENCY EFFECTS ON RESULTS

In a SoS, system dependencies result in one system’s performance affecting other systems’ performance, often in ways that are not readily apparent.

3.1 System Availability Interactions

Since A_o is a measure of overall availability in the operational environment, up time includes time when the system is operating or operable. When the system operates outside of the context of the SoS, operable time is based on system utilization. A system that is used for 18 hours per day and completes a day without a failure will experience 6 hours of operable time before being used again and will have utilization of 75%. Within the context of the SoS, the system can experience additional operable time due to dependencies with other systems. A system that is deployed on a helicopter, for example, cannot go on a sortie by itself if the helicopter is down.

System dependencies have significant effects on calculated metrics. In a simulation of a SoS over a fixed period of time, or in an actual fixed length test, the operating, operable, and down times for each system can be recorded. The calculated A_o for a particular system must be interpreted with the understanding that it is the realized A_o for the system in the context of the SoS. A system could have a high value for A_o that is in large part due to poor performance of other systems.

The interdependence of systems produces situations where improvements made to certain systems affect other systems. If improvements are made to System A, and the simulation or test is repeated, System C’s availability may go down, even though System C was unchanged. System C will accrue less operable time and may have more failures over the course of the fixed duration mission.

This realized value for system A_o in the context of the SoS may not seem useful since it can vary based on other systems’ performance. However, in many ways it is the most informative value for A_o since it is calculated based on the system as it will be used in the SoS. Planning fuel and spares required for a system should be based on how much the system operates in the context of the SoS. The system A_o in the context of the SoS is likely a better metric for these purposes than A_o of the system in isolation.

3.2 Mean Down Time for a SoS

Assessing mean down time (MDT) in isolation can be misleading, which is even more apparent for a SoS. Consider the case of a SoS with a MDT estimate of 3.8 hours, meeting its requirement of 4 hours. The SoS has a system that fails frequently but can be repaired in 2 hours. Another system with higher reliability requires 24 hours for repair. To address SoS-level MTBF shortfalls, reliability improvements may be made to the system that fails most often, greatly improving its reliability. However, when the SoS model is re-run to determine new SoS metrics, the MDT for the SoS is now 4.2 hours and no longer meets the requirement. The only change was in improvement to a system’s reliability. Although MDT went up, the number of downing events decreased.

3.3 Logistics for a SoS

Logistics are an important part of calculating operational availability. Typically, logistics are considered for a single system and decisions about spare part quantities held at various spare part locations, as well as personnel requirements are made to increase the A_o of that single system. The problem with this approach in a SoS is that outside influences and dependencies are not taken into account. Considering the system within the context of a SoS will increase the A_o of the SoS while optimizing the spare part quantities, locations, and personnel requirements for the systems as part of the SoS.

Space and weight limitations necessitate evaluation of sparing at the SoS level. As an example, consider logistics for System C, for which independent analysis suggests having ten spare parts in a nearby location. When considered in a SoS, perhaps only six spare parts are needed in the nearby location because System A in the SoS fails more often and System C's operations depend on Systems A's operation. Thus, System C would be idle when System A is down and System C may not fail as frequently as it would if there were no dependencies. These considerations and results would not be apparent without simulating the SoS and various spare part scenarios.

Similarly, personnel requirements must be considered in a SoS context rather than on a single system basis. Personnel requirements will be different when operating a single system versus operating that system within the context of a SoS. Operators or maintainers of certain systems will be needed when certain systems are operating and if there are dependencies on the operations of other systems within a SoS, operators and maintainers could be needed more or less than when looked at for a single system. Further, contention for personnel could become a larger issue when the systems are studied within the context of a SoS.

4 SIMULATION TO ASSESS A_o

As previously stated, accurately assessing Ao for a SoS will usually require simulation. In some cases, commercially available software will have the features necessary to model the SoS. Modeling the dependencies, however, can be difficult and typically requires the user to have some formal training in the software application. For larger or more complex SoSs, it may be necessary to develop a customized, special-purpose model or utilize software specifically designed for SoS analysis, such as SoSAT.

4.1 Simulation Example

The SoS described in Section 2 was modeled using SoSAT. The specific scenario for each system was entered into the model along with each system's failure and repair characteristics. Input data is shown in Table 1. Additionally, higher level rules were implemented to enforce the system dependencies and define when the SoS is considered to be in an "Up" state.

Table 1. Input Data

System	MTBF	MDT	Scheduled Operating Hours
A	20	18	10
B	50	12	18
C	25	2	28
D	500	4	168

4.2 Simulation Progress Monitoring

While a simulation is in progress, insight into the SoS performance is critical to ensure the SoS logic and system dependencies are correctly implemented. Scenario animation capabilities are particularly critical during model development. Once the model is determined to correctly react to various changing conditions, final runs with a large number of statistical trials can be performed without using a graphical display. Figure 4 presents a SoSAT screen capture of one trial of a simulation of the sample SoS.

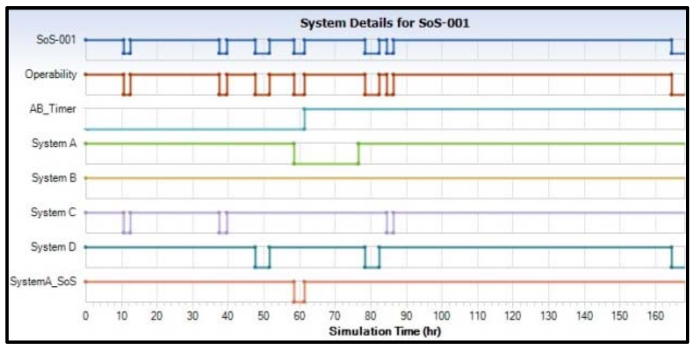

Figure 4. SoS Model Screen Capture

The lines in the diagram in Figure 4 represent the status of systems, user defined functions, and the SoS. When a line dips down, the item it represents is in a down state. The line labeled "AB_Timer" was added to indicate when the scenario changes from using System A to System B. This particular trial was captured because it highlights a unique situation regarding System A. On this trial, System A failed on day 3 and stayed down for 18 hours. A_o for the SoS over time is represented by the top line. The SoS experiences down time coinciding with system failures several times in this chart. The SoS goes to a down state when System A fails. Once System A is not required, as indicated by the AB_Timer, the SoS is back up. SoS A_o is the total uptime for the SoS divided by the total mission time, 168 hours in this case.

4.3 Simulation Results

Figure 5 shows the time spent in each state by each of the systems and the SoS, the SoS A_o, and the individual system A_o values, within the SoS context. A_o for the SoS and each system is determined as the sum of the operating and operable time divided by the total time (sum of all states). A_o for the SoS is 0.929, but this availability is not evenly distributed throughout the period of interest. Because System A has a relatively poor MTBF and the systems are used at different times, the SoS instantaneous A_o is low while System A is required and varies over time, as shown in Figure 6.

System Details

System	Time Operating	Time Operable	Time Inoperable	Time Down	Availability	Total Failures
SoS	373.754	548.280	10.874	75.092	0.915	4.796
SoS-001	156.098	0.000	0.000	11.902	0.929	1.850
System A-001	9.273	153.219	4.955	0.553	0.967	0.306
System B-001	17.022	146.593	3.567	0.818	0.974	0.368
System C-001	24.148	141.968	0.852	1.032	0.989	0.942
System D-001	166.713	0.000	0.000	1.287	0.992	0.330

Figure 5. Simulation Results

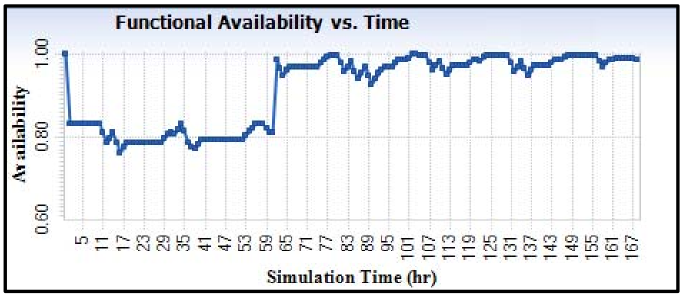

Figure 6 SoS Instantaneous Availability over Time

System A operated for 9.3 hours when operating within the SoS. In a separate simulation, in which all systems except System A were set to very high MTBFs so they would not fail, System A operated for 9.6 hours. In isolation, System A operated more and thus experienced more failures than when operating within the context of a SoS. Although the difference in operating hours may seem insignificant, if the mission is repeated multiple times, estimates for spares based on analysis could be inaccurate. The differences could become important if model results are projected to annual or population-wide projections.

5 CONCLUSIONS

Systems of systems are becoming more prevalent than ever with increased use of autonomous systems and network-centric functionality. For example, UAV systems often incorporate communication networks and command and control equipment. Energy systems are incorporating energy sources such as wind turbines and solar technologies that have intermittent usage. Defining A_o for these systems is not straightforward and will involve many of the same questions raised for the example in this paper.

For systems functioning as a SoS, assessing SoS A_o performance is imperative for logistics and planning purposes, since operations, maintenance, and sustainment of the systems will be performed at the SoS level and for the intended SoS operational environment. Analyzing SoSs with accurate accounting for interactions is complex, but is crucial to obtaining meaningful and accurate SoS results.

Approved for public release, distribution is unlimited, July 30, 2014, SAND2014-16359C.

BIOGRAPHIES

Dennis J. Anderson
Sandia National Laboratories
PO Box 5800 MS-1188
Albuquerque, New Mexico 87185 USA

e-mail: djander@sandia.gov

Dennis J. Anderson is Distinguished Member of the Technical Staff in the Military and Energy Systems Analysis Programs Department at Sandia National Laboratories in Albuquerque, New Mexico, where he has worked for 23 years. He is currently Principal Investigator for two large system-of-systems (SoS) logistics modeling and analysis projects for the US Army and is technical consultant for SoS projects with the US Marine Corps and the US Air Force. Mr. Anderson received an M.A. in Mathematics (Applied Statistics option) from Arizona State University, Tempe, Arizona, and a B.A. in Mathematics, St. John's University, Collegeville, Minnesota.

Charles M. Carter
Sandia National Laboratories
PO Box 5800 MS-1188
Albuquerque, New Mexico 87185 USA

e-mail: cmcarte@sandia.gov

Charles M. Carter is Senior Member of the Technical Staff in the System Readiness and Sustainment Technologies Department at Sandia National Laboratories, Albuquerque, New Mexico. He has worked in the reliability field for over 20 years. As an Air Force officer, he worked as a reliability analyst for the Air Force Operational Test and Evaluation Center. He worked for SAIC as a payload safety engineer at Johnson Space Center, performing safety analysis on Space Shuttle and International Space Station payloads. Mr. Carter developed the simulation engine for the Raptor™ reliability engineering tool and has extensive experience using simulation to evaluate reliability and availability. Mr. Carter received an M.S. in Systems Engineering from the Air Force Institute of Technology and a B.S. in Aeronautical and Astronautical Engineering from the University of Illinois.

Tamara Brown
Sandia National Laboratories
PO Box 5800 MS-1188
Albuquerque, New Mexico 87185 USA

e-mail: tambrow@sandia.gov

Tamara Brown is Senior Member of the Technical Staff in the System Readiness and Sustainment Technologies Department at Sandia National Laboratories, Albuquerque, New Mexico, where she has worked for 7 years. She is currently Principal Investigator for a complex U.S. Navy SoS acquisition program involving reliability and availability modeling, analysis, and optimization. Ms. Brown received an M.S. in Statistics from Iowa State University, Ames, Iowa, and a B.S. in Mathematics from University of Nevada, Reno, Nevada.