Intricacies of System of Systems Operational Availability and Logistics Modeling

18th Annual Systems Engineering Conference

October 26 – 29, 2015

Charles M. Carter*
Dennis J. Anderson
Tamara J. Brown

Sandia National Laboratories
Albuquerque, NM

*cmcarte@sandia.gov, 505-284-4498
Overview

• Background and definition of operational availability \((A_o)\) and system of systems (SoS)

• SoS \(A_o\) equations

• SoS \(A_o\) complexities

• Example SoS \(A_o\) simulation results

This presentation is based on a paper (Anderson, D.J., Carter, C.M., and Brown, T., “Complexities of System of Systems Operational Availability Modeling”, Proceedings of the 2015 Reliability and Maintainability Symposium, Palm Harbor FL, Jan 26-29, 2015) with additional focus on system of systems logistics management.
System $A_o$

- $A_o$ 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

- Any reported $A_o$ reflects and depends upon all the above assumptions

- Comparison of $A_o$ to requirements, other system $A_o$, or across tradeoffs requires “synching up” above definitions to make comparisons valid

**Definitions and assumptions must be addressed for each intended use of $A_o$ models**
Well-understood, high-level $A_o$ definition

$$A_o = \frac{Uptime}{Uptime + Downtime}$$

$$= \frac{Total \ Time - Downtime}{Total \ Time}$$

$$= \frac{Operating \ Time + Operable \ Time}{Operating \ Time + Operable \ Time + Downtime}$$

Operable time (or standby time) usually defined as part of $A_o$

Common equation estimator of $A_o$

$$A_o = \frac{MTBF}{MTBF + MDT}$$

where MTBF is Mean time Between Failures and MDT is Mean Down Time

Results in static, steady-state estimation of $A_o$ if steady state MTBF and MDT are used

Not valid for short durations that do not reach steady state

MTBF can be scaled by utilization for intermittent use systems
System of Systems

• SoS definition
  – A SoS is comprised of a set of systems, each performing a defined task or mission, in which at least one system can be dependent on one or more other systems
  • SoS level performance is emerging and cannot be assessed by assessing individual system performances separately, except for the case where the systems operate (and are maintained) independently of each other
  • System dependencies can be of varying complexity and include:
    – Required sequential or parallel system tasks
    – System functional redundancies
    – K of N systems operating
    – Combinations of these

• SoS are more prevalent than ever
  – More autonomous systems functioning with other systems
  – Increased network-centric functionality
  – Effectiveness and requirements established for increased system synergies accomplishing increasingly complex missions
**SoS $A_o$ Equations**

- **Case 1:** If all systems in the SoS operate and are maintained independently, the SoS $A_o$ is equal to the product of the individual system $A_o$s

\[
SoS A_o = \prod_{i=1}^{n} A_{oi} \quad \text{where } A_{oi} \text{ is the } i^{th} \text{ independent system } A_o
\]

- **Case 2:** If all systems are dependent, such that if any one system fails the remaining systems stop operating, the SoS $A_o$ can be solved as a function of the individual system $A_o$s

\[
SoS A_o = \frac{1}{1 - n + \sum_{i=1}^{n} \frac{1}{A_{oi}}} \quad \text{where } A_{oi} \text{ is the } i^{th} \text{ independent system } A_o
\]

*Rarely does a SoS operate as Case 1 or Case 2*
SoS Ao Calculations

- SoS Ao equation calculations have limited application
- Complicating factors that negate the use of estimating equations for SoS Ao
  - Systems within the SoS not all in series
  - Systems not all independent or not all dependent
  - Systems not all operating all of the time
  - Complicated logistics strategies
  - Sustainment limitations, including spares, maintainers, special equipment
  - Short duration of operations, making the steady-state result not relevant
- Simulation modeling is usually required to capture complex operating, operable, and down time hours of a SoS
- System of Systems Analysis Toolset (SoSAT) simulation used to analyze and provide SoS Ao for an example problem
SoSAT Background and Applications

- **SoSAT (System of Systems Analysis Toolset)** is a suite of software tools:
  - State modeling tool
  - Stochastic simulation tool
  - Advanced data visualization tools
  - Reliability, consumables, and supply chain optimization tools

- Initially designed to provide DoD and military services capability to analyze large systems of systems and their various platforms across multiple mission scenarios to assess multiple key performance parameters
  - Supported multiple US Army Future Combat Systems (FCS) trade studies
  - Supported US Army PEO Integration with modeling and analysis of Logistics, Sustainment, Reliability Key Performance Parameters for Capability Packages
  - US Army PEO Ground Combat Systems (PEO GCS) using SoSAT for Fleet Management and Modernization Planning initiative
  - JPO MRAP using SoSAT for MATV assessments and analyses
  - Participating in formal Verification, Validation & Accreditation effort with Army Organizations (AMSAA and ATEC)
  - Navy Littoral Combat Ship and Littoral Mine Warfare using SoSAT for fleet modernization planning

- **SoSAT v1.0** released October 2007
- **SoSAT v1.5** released October 2008
- **SoSAT v2.0** released January 2010
  - SoSAT v2.0 simulation verified and validated (V&V’d) by the US Army
  - Availability calculations and algorithms officially accredited by the US Army
- **SoSAT v3.0** in development
SoSAT Capabilities

• SoSAT provides analysts the capability to
  – Simulate *any or all* of a system of systems (SoS) organizational structure
  – Simulate multiple mission segments for a SoS
  – Provide data to assess SoS performance objectives
  – Support business decisions and trade-offs

• Basic Modeling Features
  – System element reliability failures
  – Consumable usage and depletion
  – Maintenance activities including any required spares or services
  – Supply reorder for consumables and spare inventories

• Advanced Modeling Features
  – Damage effects modeling
  – Network modeling
  – Prognostics and Health Management
  – Time-based changes to model attributes
  – System Referencing (interdependencies)
  – Human performance

• Active Model Development
  – Advanced network modeling
  – Enterprise modeling

Tracks individual system components and functional availability over time

Track SoS functional availability and statistics over time
SoSAT Platform Modeling Concepts

- **Platform as a System State Model**
  - Multiple user defined functions/operations
  - Multiple States (not just functional or failed)
  - Models interdependencies
  - Can include external factors (weather, terrain, combat, etc.) that affect the overall system or just the system elements

- **Model system behavior by defining:**
  - States for all subsystems/components/functions
  - How transitions are made between states

- **States can change through:**
  - Normal processes (failure, repair, etc.)
  - External conditions (weather, terrain, combat, etc.)
  - Changes in functional states of other systems

The System State Model is an intuitive way to capture system behavior and is the building block for systems in the simulation.
• **Seemingly simple SoS with 4 systems**
  – System A and System B are each used with System C
    • First part of the mission, System A and System C operate together
    • Second part of the mission, System B and System C operate together
  – System D runs independently and continuously

<table>
<thead>
<tr>
<th>Day&gt;</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>System A</td>
<td>□</td>
<td>□</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System B</td>
<td>□</td>
<td></td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>System C</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
<tr>
<td>System D</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

• Even for this simple four-system SoS, numerous complexities arise from the use of multiple systems to perform a mission
Mission Profile Complexities

• Is the mission profile executed as planned even if System A experiences failures?
  – Depending on the scenario being modeled, failures of system A could cause a shift of the profile and extend the mission beyond 7 days
  – In this example scenario, the scheduled operating periods are missed and the schedule would proceed as planned

• What systems are required to be up for the SoS to be up?
  – This reliability block diagram is accurate for representing the SoS reliability, from a reliability perspective, at time of failure
    • If any system fails, the SoS fails
  – However, this RBD does not apply for A₀ over the entire mission
  – Systems required to be up can vary over the mission profile for a SoS
• **What happens if System A fails on Day 3 and is down for 24 hours?**

<table>
<thead>
<tr>
<th>Day&gt;</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>System A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>System D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- The affect on $A_o$ depends on specifics of the scenario being modeled.
- The SoS represented could be a ship transiting through an area that requires System A in one region and System B in another.
  - In this case, the series RBD is not valid for $A_o$ because the RBD is dynamic.
System $A_o$ Complexities

- **$A_o$ can be reported for the individual systems operating as part of the SoS, but for what time period?**
  - $A_o$ can be reported for:
    - The entire 7 day mission period
    - The portion of the mission that system has an active role
    - The specific segments when the system was scheduled to be operating
  - In many cases, all three periods are of interest, requiring additional statistics to be gathered during a test or simulation

- **If System C fails, does System A go to an operable state?**
  - The actual SoS this example was based on was a helicopter (System C) with payloads (Systems A and B)
  - If System A failed, System C would not operate
  - Similarly, if System C fails, System A would not operate
  - SoS assessment must account for these periods of dependency that result in additional operable time for non-failed systems
Dependency Effects on System $A_o$

• $A_o$ measures the availability in the operational environment
  – Up time includes operating time and operable time

• When a system operates outside the context of a SoS, operable time is a function of system utilization
  – A system that is scheduled to run for 18 hours/day has a utilization of 0.75

• Within a SoS, there can be additional operable time due to system dependencies
  – Direct dependencies, such as System A not being used because System C is down
  – Indirect dependencies, such as unrelated systems occasionally causing the SoS to move to an area for repairs, limiting the operating time of other systems
System dependencies can cause unexpected results

- An improvement to one system’s reliability can cause lower performance measures of other systems
  - A system experiences operable time if it is waiting for another system it depends on to be repaired
    - Operable time is counted as up time in the $A_o$ calculation for the dependent system
  - A system could operate more if another system it depends on is improved
    - Reliability and maintainability improvements could cause all systems within the SoS to operate more

- Time that was previously categorized as operable time will be replaced by additional operating time and down time

$A_o$ for a dependent system can decrease after an improvement to a system it depends on
Mean Down Time Complexities

- MDT, in isolation, can be a misleading statistic
- A reliability improvement to a system within a SoS can cause an increase in the SoS MDT
  - Suppose a SoS meets the SoS MDT requirement but does not meet the SoS MTBF requirement
  - A reliability improvement is made to a system that fails frequently
  - SoS MTBF now meets the requirement, but SoS MDT has increased and no longer meets the requirement
  - This unexpected result can occur if the repair times of the improved system are less than the SoS MDT
    - The total downtime is less as a result of the reliability improvement
- The effect of MTBF changes on MDT can be seen within an individual system, but SoS assessment is necessary to see the SoS MDT impact

MDT is most useful when combined with MTBF in an $A_o$ calculation
Logistics

• Logistics delays are an important part of MDT
  – Large delays can be incurred when waiting on parts, consumables, and personnel if correct planning has not taken place

• Critical to plan for spares, consumables, and personnel for systems operating within a SoS
  – Spare parts quantities, consumables, and personnel requirements for a system may differ when the system is considered operating in a SoS versus independent of the SoS
    • For example, within the SoS, a system may not operate as much as it would on an independent mission requiring less resources
    • If multiple systems within the SoS require the same repair skill type, additional delays can occur if the resource is limited and already in use

Space, weight, and personnel constraints can be traded off amongst individual systems to maximize SoS metrics
SoS $A_o$ Simulation Calculation

- SoS $A_o$ is calculated by this equation through simulation

$$A_o = \frac{Uptime}{Uptime + Downtime}$$

- Simulation allows for complex mission and system definitions and dependencies
- The sample SoS was simulated assuming the following attributes of Systems A - D

<table>
<thead>
<tr>
<th>System</th>
<th>MTBF</th>
<th>MDT</th>
<th>Scheduled Operating Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>20</td>
<td>18</td>
<td>10</td>
</tr>
<tr>
<td>B</td>
<td>50</td>
<td>12</td>
<td>18</td>
</tr>
<tr>
<td>C</td>
<td>25</td>
<td>2</td>
<td>28</td>
</tr>
<tr>
<td>D</td>
<td>500</td>
<td>4</td>
<td>168</td>
</tr>
</tbody>
</table>
SoS Ao Simulation Results

- The lines 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.
- Not all System A downtime results in SoS downtime.
Scheduled vs Actual Operating Time

- Each system operates less time than scheduled because of failures and dependencies
- This mission could be repeated many times by this system or a fleet of systems
- Fuel Usage and sparing estimates should be based on performance within the SoS.
SoS Instantaneous Availability vs. Time

- $A_0$ for the SoS was 0.929 for the entire mission
- SoS performs much better once operations shift from System A to System B
  - System B has a higher MTBCF and a lower MDT than System A

**Instantaneous $A_0$ can vary throughout the mission depending on the current activity**
• Systems of systems are becoming more prevalent than ever with increased use of autonomous systems and network-centric functionality

• System interdependencies and complex interrelated sustainment operations that exist in a SoS present complexities in calculating or estimating $A_o$
  – In most cases, simulation modeling is required

• Analyzing SoSs with accurate accounting for interactions is complex, but is crucial to obtaining meaningful and accurate SoS results