Intricacies of System of Systems Operational Availability and Logistics Modeling

System of Systems Engineering Collaborators Information Exchange (SoSECIE) Webinar
March 29, 2016

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

Sandia National Laboratories, Albuquerque, NM

*cmcarte@sandia.gov

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. Distribution Statement: Approved for public release; distribution is unlimited. SAND2014-16349PE.

Sandia National Laboratories

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

System A_o

- Operational Availability (A_O) is the portion of time that a system is either operating or capable of operating
- 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

A_o Definitions

- 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_os

$$SoS\ A_o = \prod_{i=1}^{n} A_{o_i}$$

where A_{o_i} is the i^{th} 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_os

$$SoS\ A_o = \frac{1}{1 - n + \sum_{i=1}^{n} \frac{1}{A_{o_i}}}$$

where A_{o_i} is the i^{th} independent system A_o

Rarely does a SoS operate as Case 1 or Case 2

SoS A_o Calculations

- SoS A_o equation calculations have limited application
- Complicating factors that negate the use of estimating equations for SoS A_o
  - 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 A_o for an example problem

SoSAT Background and Applications

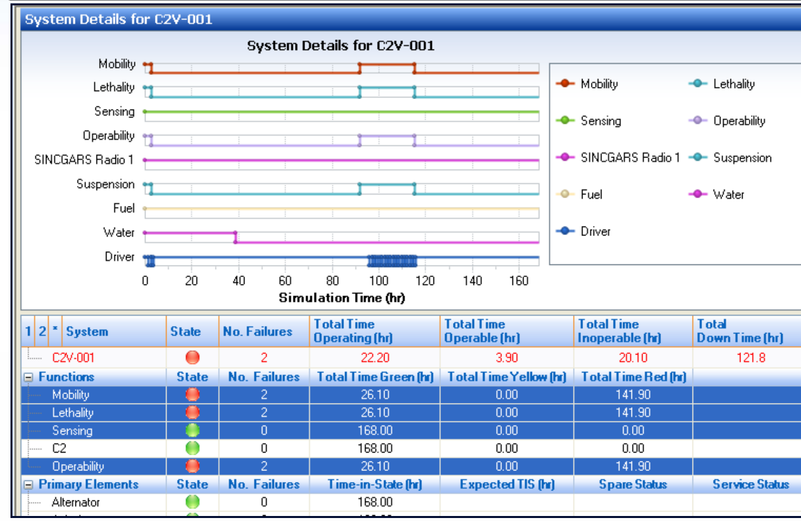
- SoSAT (System of Systems Analysis Toolset) is a suite of software tools:
  - State modeling tool
  - Stochastic simulation tool
  - Advanced data visualization tools
  - Reliability, consumables, and supply chain optimization tools
- Initially designed to provide DoD and military services capability 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 Mission Modules using SoSAT for modeling and analyssis

- **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**

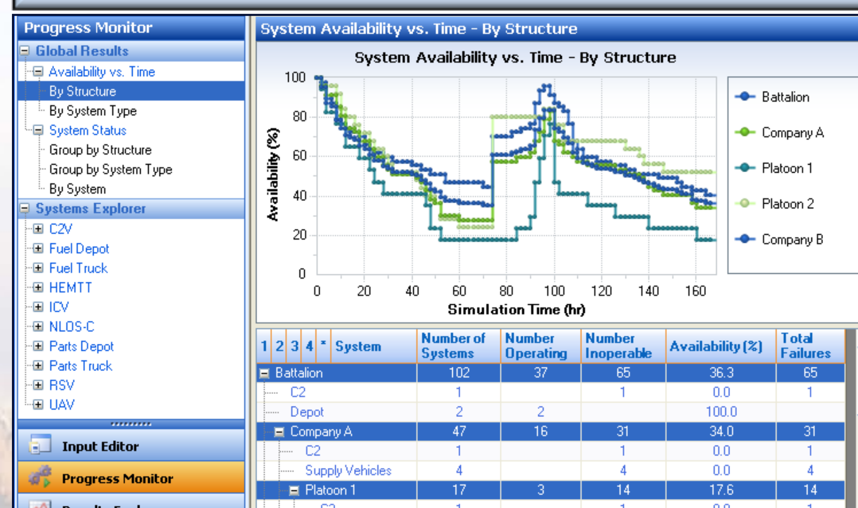
SoSAT Capabilities

- 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

Platform as a System State Model

System Model

Example Elements

- 105 mm Cannon
- M240 Machine Gun
- Sandstorm

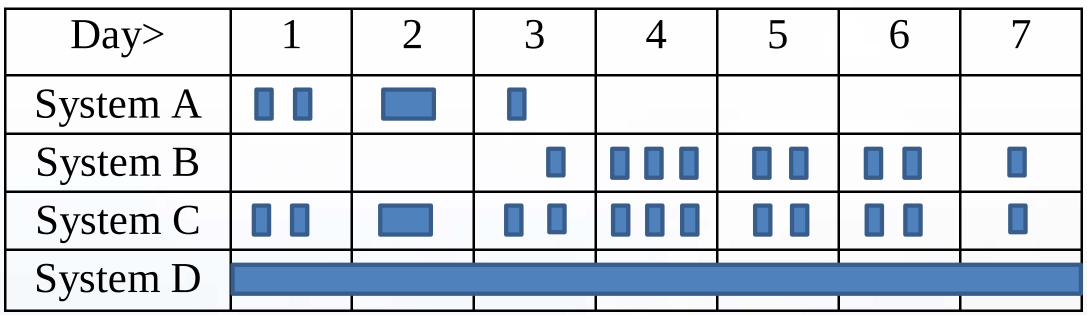
Example Operations

- Operability
- Lethality
- Mobility

The System State Model is an intuitive way to capture system behavior and is the building block for systems in the simulation

Example Problem

- *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



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

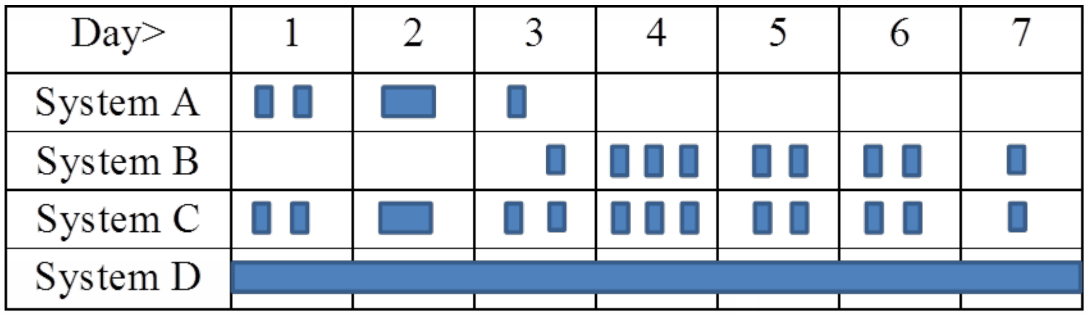
Mission Profile Complexities

- ***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

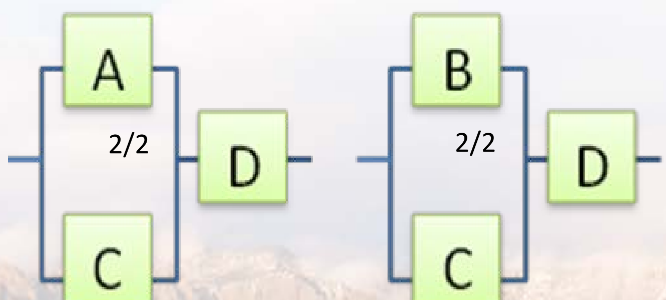
  - However, this RBD does not apply for A_o over the entire mission
  - ***Systems required to be up can vary over the mission profile for a SoS***

Mission Profile Complexities (cont.)

- ***What happens if System A fails on Day 3 and is down for 24 hours?***



  - 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



Days 1-3 **Days 3-7**

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

Dependency Effects on System A_o (cont.)

- 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

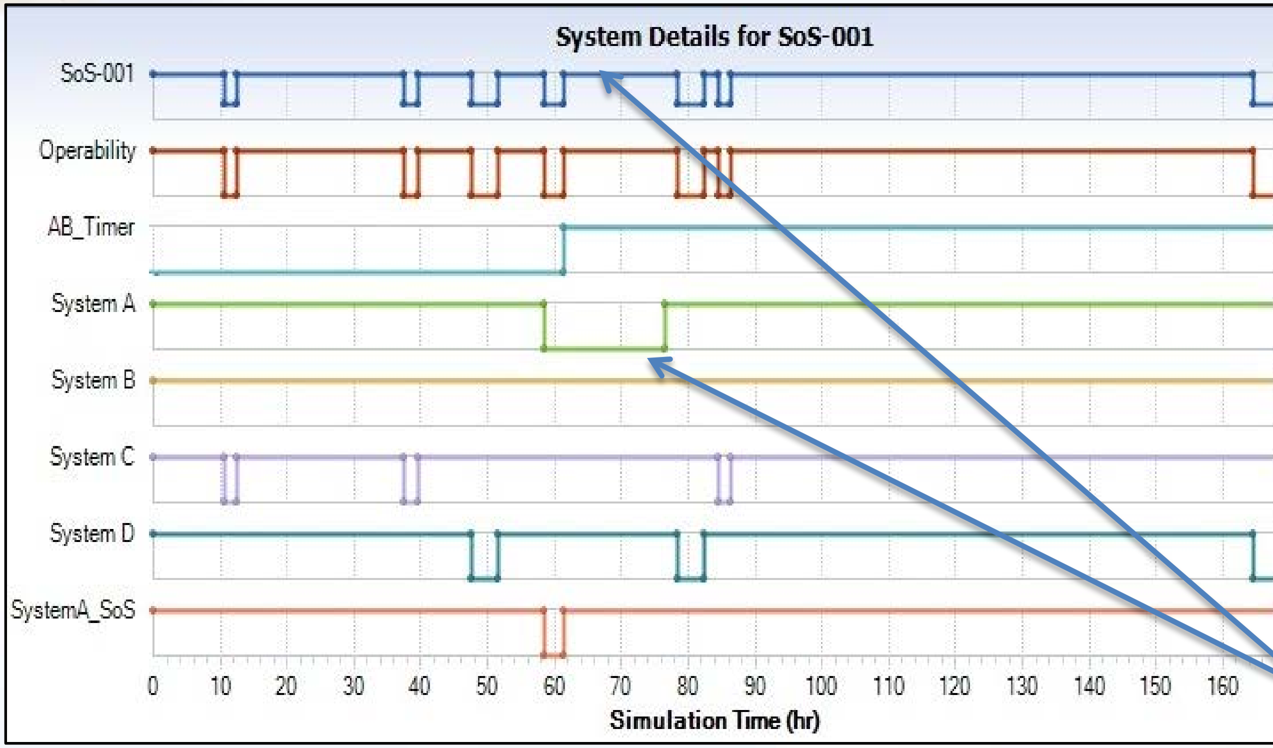
- SoS A_o is calculated by this equation through simulation

$$A_0 = \frac{Uptime}{Uptime + Downtime}$$

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

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

SoS A_o Simulation Results



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

SoS A_o

- 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

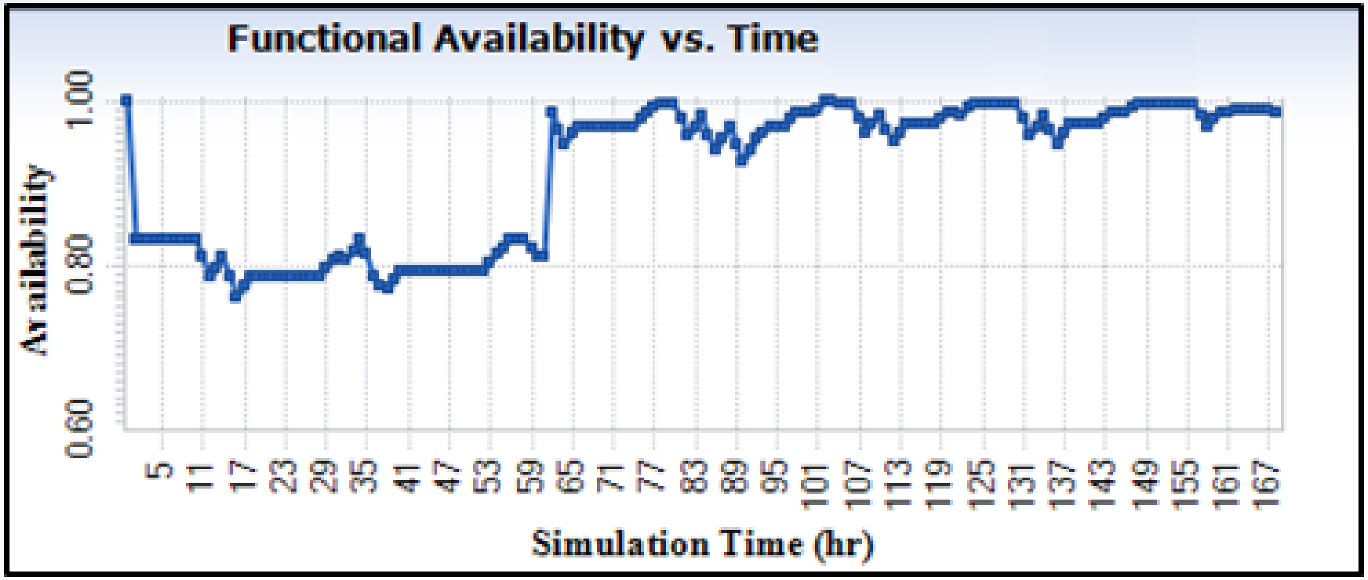
System	MTBF	MDT	Scheduled Operating Hours
A	20	18	10
B	50	12	18
C	25	2	28
D	500	4	168

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

- 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_o 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_O can vary throughout the mission depending on the current activity

Summary

- 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