Progress Towards an Analytic Workbench for System of Systems Architectures

2014 SERC Sponsor Research Review

Dan DeLaurentis and Karen Marais

Center for Integrated Systems in Aerospace
http://www.purdue.edu/research/vpr/idi/cisa/

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.
Agenda

• Problem Statement and Vision

• SERC RT-108 SoS Analytic Workbench

• Methods in Workbench
  – System Importance Measures (SIMs)
  – Functional Dependency Analysis (FDNA/DDNA)
  – Robust Portfolio Optimization (RPO)
  – Decentralized Development Planning (DDP)

• Pilot Studies

• Future Work
Problem Statement

• SoS Architectures are highly complex, with many interdependencies across diverse constituent systems

• Difficult to know how and when to add/remove/integrate systems or connections
  • Too big for one analyst
  • Too many contingencies and choices for simple tools
  • Too many stakeholders for top-down management

<table>
<thead>
<tr>
<th>Pain Points</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>SoS Authority</td>
<td>What are effective collaboration patterns in systems of systems?</td>
</tr>
<tr>
<td>Leadership</td>
<td>What are the roles and characteristics of effective SoS leadership?</td>
</tr>
<tr>
<td>Constituent Systems</td>
<td>What are effective approaches to integrating constituent systems into a SoS?</td>
</tr>
<tr>
<td>Autonomy, Interdependencies &amp; Emergence</td>
<td>How can SE provide methods and tools for addressing the complexities of SoS interdependencies and emergent behaviors?</td>
</tr>
<tr>
<td>Capabilities &amp; Requirements</td>
<td>How can SE address SoS capabilities and requirements?</td>
</tr>
<tr>
<td>Testing, Validation &amp; Learning</td>
<td>How can SE approach the challenges of SoS testing, including incremental validation and continuous learning in SoS?</td>
</tr>
<tr>
<td>SoS Principles</td>
<td>What are the key SoS thinking principles, skills and supporting examples?</td>
</tr>
</tbody>
</table>

Survey identified seven ‘pain points’ raising a set of SoS SE questions


Can an organized set of Methods, Processes and Tools (MPTs), presented in a user-friendly way, solve these problems?
Vision: A *Useful* SoS Analytic Workbench

- **Rational**
  - Relegate complexities to methods
  - Delegate decision-making to users
- **Open**
  - Accommodates insertion of new SoS analytic methods (from Purdue or others)
- **Interoperable**
  - Outcomes produced in form suitable for additional SoSE phases
  - ‘Domain agnostic’, cross platform operations
  - Address uncertainty in data/simulation outcomes
- **Useable**
  - (Scalability) \(\Rightarrow\) reasonable scaling of computational need to problem sizes
  - (Ease of Use) \(\Rightarrow\) Users can translate problem to inputs required by relevant methods and tools
SERC RT-108: SoS Analytic Workbench

Methods in Toolset:
- Robust Portfolio Optimization
- Dynamic Programming
- System Importance Measures
- Functional/Developmental Dependency Networks

Input Data (e.g. DoDAF OV, SV, PV declarations)
A Network Representation of SoS

- Translate SoS problem into network problem (nodes, links, inputs, outputs)
- Map data and DoDAF description to equivalent network representation

Inputs (e.g. req.)
Outputs e.g. capab.

OV – Operation Flow
SV – Service Flow
PV – Project Flow
…
Simulation/Actual data

Mapping
Archetypal Questions in SoS development and operation

• **Design**
  1. What combination of systems gives the desired aggregate SoS capabilities?
  2. What changes to which systems offer the most (performance, resilience, etc.) leverage?
  3. Which systems are critical to SoS performance? SoS risks?
  4. Which parts of the SoS have excess or inadequate resilience?
  5. Which design principles can improve SoS robustness and resilience?

• **Development**
  6. How do/should partial capabilities evolve over time?
  7. How do we optimize multi-stage acquisitions in SoS development?
  8. How do we coordinate planning between local and SoS-level stakeholders?
  9. How do (desired and undesired) changes in system properties affect SoS development?

• **Failures and Delays**
  10. What is the impact of partial/total system failures during operations?
  11. What is the impact of partial/total failure of a system during development?
  12. What are the most critical systems in a given operational (or developmental) network?
  13. What is the impact of development delays in an interdependent network?
Addressing the Archetypal Questions

**Design**
1. What combination of systems gives the desired aggregate SoS capabilities?
2. What changes to which systems offer the most (performance, resilience, etc.) leverage?
3. Which systems are critical to SoS performance? SoS risks?
4. Which parts of the SoS have excess or inadequate resilience?
5. Which design principles can improve SoS robustness and resilience?

**Development**
6. How do/should partial capabilities evolve over time?
7. How do we optimize multi-stage acquisitions in SoS development?
8. How do we coordinate planning between local and SoS-level stakeholders?
9. How do changes in system properties affect SoS development?

**Failures and Delays**
10. What is the impact of partial/total system failures during operations?
11. What is the impact of partial/total failure of a system during development?
12. What are the most critical systems in a given operational (or developmental) network?
13. What is the impact of development delays in an interdependent network?
Inputs for SoS Analysis

Examining Current SoS AWB Methods

Translate user input into parameters of SoS AWB and data requirement

<table>
<thead>
<tr>
<th>FDNA/DDNA</th>
<th>User Input</th>
<th>AWB Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time to detect enemy / % of enemies detected</td>
<td>Operability</td>
</tr>
<tr>
<td></td>
<td>Probability of radar node detecting an enemy</td>
<td>Self Effectiveness (SE)</td>
</tr>
<tr>
<td></td>
<td>Scaled loss of operability when input missing</td>
<td>Strength of Dependency (SOD)</td>
</tr>
<tr>
<td></td>
<td>Effects of total loss of input</td>
<td>Criticality of Dependency (COD)</td>
</tr>
<tr>
<td></td>
<td>Effective range of radar</td>
<td>System Capabilities</td>
</tr>
<tr>
<td></td>
<td>Power req. of radar</td>
<td>System Requirements</td>
</tr>
<tr>
<td></td>
<td>Types of compatible power supplies</td>
<td>System Compatibilities</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Robust Portfolio Optimization/ADP</th>
<th>System Importance Measures</th>
<th>System Disruption Importance (SDI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Probability of radar loss</td>
<td>System Recoverability Importance (SRI)</td>
<td></td>
</tr>
</tbody>
</table>
Current and future input formats

- DoDAG Descriptions
- SySML
- UDPM
- Other user input formats

SoS – Analytic Workbench

Examples of “where they live”
Analysis & Verification

Initial Architecture + Candidates

Map Questions & Data to Methods

Evaluate Solutions

Verify

Data

Simulation (e.g. Agent Model)

SoS Analysis

Generate Architecture(s)

Iterative process to improve architecture

Use of simulation as a “truth model” and/or as data generator
SoS Analytic Workbench Demo. Software

User Manual for Analytic Workbench GUI

This manual aims at helping users learn the basic steps to conduct the analytic workbench using the GUI. It covers five methods, including Functional Dependency Network Analysis (FDNA), Development Dependency Network Analysis (DDNA), System Importance Measures (SIM), Robust Portfolio Optimization, and Decentralized Development Planning. Development and refinement of the workbench and tools within it are being continued under SERC RT-134.

Contents

Functional Dependency Network Analysis (FDNA) .................................................. 2
  Option 1: Demo Scenario ............................................................................. 2
  Option 2: User-defined scenario ................................................................. 8
  Option 3: Import data from spreadsheet ...................................................... 13
Development Dependency Network Analysis (DDNA) ................................. 16
  Option 1: Demo Scenario ............................................................................. 22
  Option 2: User-defined scenario ................................................................. 27
System Importance Measures (SIM) ................................................................. 27

System Importance Measures (SIMs)

1. Build SoS
   Enter Number of Systems: [ ] Apply

2. Upload from file: [ ] Load

3. Naval Warfare SoS (Demo)
   Load pre-determined inputs for naval warfare SoS: [ ] Load

Functional Dependency Network Analysis (FDNA)

Development Dependency Network Analysis (DDNA)

Decentralized Development Planning Framework

Input
- Input Pcs for Demonstration
- Se-SEIR Stakeholders
- Decomposed Case
- Participant Interactions
- Navy
- Air Force
- Add Systems

Experiments
- Centralized
- Hierarchical Case
- Standards Case
- Participative Interactions
- Navy
- Air Force
- Add Systems
- Centralized Case
- Hierarchical Case
- Standards Case
- Participative Interactions
- Navy
- Air Force
- Add Systems

Problem Setup Example

SoS-O-Matic System-Rel-DoS

Soft Technologies
- System Reliability
- System Cost
SoS Analytic Workbench Demo. Software

System Importance Measures (SIMs)  |  Robust Portfolio Optimization  |  FDNA/DDNA  |  Decentralized Planning (using ADP)
Systems Importance Measures (SIMs)
Resilience in Systems-of-Systems: Motivation

- Fire at Chicago ARTCC (ZAU)
  - Aurora, IL
  - ARTCCs control aircraft in en-route phase of flight

- Impacts:
  - Many cancellations (49%-ORD and 78%-MDW)
  - 2,077 cancellations of flights to, from and within the U.S.

- Lack of flexibility
  - Difficulties in shifting controller workload and information between ATC facilities
System Importance Measures

- Resilience metrics must be able to guide answers to the following:
  - Is the SoS resilient enough?
  - Which part of the SoS is lacking in resilience?
  - What SoS features balance the desired resilience with costs?

- System Importance Measures (SIMs):
  - Rank constituent systems based on
    a) impact of their disruption on the SoS
    b) their contribution to recovery when disruptions occur
  - Capture different aspects of SoS resilience
  - Inspired by Component Importance Measures (CIMs) in reliability theory
    - Birnbaum importance, Risk achievement worth (RAW), Risk reduction worth (RRW), Improvement potential
The Resilience Curve

Resilience is the ability of a system to **survive** and **recover** from disruptions.
How can we design resilient SoSs?

• Key questions related to designing resilience:
  – When is resilience needed?
  – How can resilience be designed?
  – How can resilience be maintained?

When is resilience needed?

How can resilience be designed?

How can resilience be maintained?

• Under what conditions is resilience appropriate?
• When is resilience suitable compared to other system-level attributes?
• How can resilience be measured?
• When is the SoS resilient enough?
• How can resilience be created?
• When are specific resilience improvement strategies suitable?
• Where (in the SoS) should these strategies be incorporated?
• What are the tradeoffs associated with these strategies?
• When does resilience change?
• How can changing resilience be tracked?
• How can resilience be stopped from declining?
System Disruption Importance (SDI)

- What is the impact of a system disruption on the overall SoS?
  - Worst-case scenario

- For example, what is the impact on the combat mission if a satellite is disrupted?

\[
\text{Impact unit} = \text{time}_{\text{disruption}} \times \text{performance}_{\text{loss}}
\]

When users have different preferences:

\[
\text{Impact unit} = \alpha_i \text{time}_{\text{disruption}} \times \alpha_p \text{performance}_{\text{loss}}
\]
System Disruption Importance (SDI)

- What is the impact of a system disruption on the overall SoS given that recovery measures are available?

- For example, if a satellite is disrupted, what is the impact on the combat mission if a UAV provides target identification capability?

\[
SDI_{i,j} = \frac{\int_{T_i}^{T_f} [f(t) - g(t)]}{\int_{T_i}^{T_f} [f(t) - P_{LOSS}]}
\]

- Low value: Disruption of System \(i\) has low adverse impact on SoS
  - LOW IMPORTANCE RANKING

- High value: Disruption of System \(i\) has high adverse impact on SoS
  - HIGH IMPORTANCE RANKING
System Recoverability Importance (SRI)

- How important is a system to SoS recovery when a disruption occurs?

- For example, if a satellite is disrupted, how important is the UAV to mission recovery?

\[
SRI_{i,j} = \frac{\int_{T_i}^{T_f} g(t) - P_{LOSS} dt}{\int_{T_i}^{T_f} f(t) - P_{LOSS} dt}
\]

- **Low value**: System \( j \) has little importance to SoS recovery

  **LOW IMPORTANCE RANKING**

- **High value**: System \( j \) is highly important to SoS recovery

  **HIGH IMPORTANCE RANKING**
Using SIMs to Guide Resilience Design

Expected System Importance (ranking)

SoS Architecture → System Importance Measures (SIM) analysis → Design Principles

Disruptions
Recovery options

Design changes

1. Physical redundancy
2. Stand-in redundancy
3. Inter-node interaction
4. Localized capacity
5. Drift correction
6. Repairability
7. Human-in-the-loop
8. System-level properties
9. Improved communication
10. Layered defense
Illustrative example

• Simple 4-node Naval Warfare SoS
  – SoS-level mission: **Detect and eliminate littoral threats**
  – Systems:
    • *Littoral Combat Ship (LCS)***: Radar and weapons
    • *Unmanned Aerial Vehicle (UAV)***: Radar and surface imaging
    • *Helicopter (MH-60)***: Radar, surface imaging, and weapons
    • *Unmanned Surface Vehicle (USV)***: Radar
Expected System Importance

- Expected System Disruption Importance (**SDI**): Worst-case scenario (no dedicated/active recovery measures)

<table>
<thead>
<tr>
<th>System</th>
<th>E(SDI)</th>
<th>Disruption Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCS</td>
<td>0.80</td>
<td>High</td>
</tr>
<tr>
<td>MH-60</td>
<td>0.75</td>
<td>High</td>
</tr>
<tr>
<td>UAV</td>
<td>0.40</td>
<td>Medium</td>
</tr>
<tr>
<td>USV</td>
<td>0.30</td>
<td>Medium</td>
</tr>
</tbody>
</table>

Now evaluate two resilience design principles:

- **Physical** (stand-by) redundancy:
  - MH-60 is “backed-up” by its identical secondary system

- **Stand-in** redundancy:
  - When a system fails, one or more systems have the capability to “stand-in” to compensate for this performance loss
  - Improved features on UAV and USV

Example:
Resilience curve when MH-60 is disrupted
Resilience curves: With recovery (physical redundancy)

Disrupted system: MH-60

Resilience improvement through physical redundancy:

Disrupted MH-60 (no dedicated recovery measure)
Resilience curves: With recovery (stand-in redundancy)

- Improved endurance
- High resolution camera
- Improved radar coverage

Disrupted system: MH-60

Resilience improvement through stand-in redundancy:

Disrupted MH-60 (no dedicated recovery measure)
Calculating SIMs: With recovery

**Improve resilience using physical redundancy:**

<table>
<thead>
<tr>
<th>System</th>
<th>$E(SDI)$</th>
<th>Disruption Impact</th>
<th>System</th>
<th>$E(SRI)$</th>
<th>Recovery Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCS</td>
<td>0.80</td>
<td>High</td>
<td>MH-60&lt;sub&gt;backup&lt;/sub&gt;</td>
<td>0.50</td>
<td>High</td>
</tr>
<tr>
<td>UAV</td>
<td>0.40</td>
<td>Medium</td>
<td>LCS</td>
<td>0</td>
<td>Low</td>
</tr>
<tr>
<td>USV</td>
<td>0.30</td>
<td>Medium</td>
<td>MH-60</td>
<td>0</td>
<td>Low</td>
</tr>
<tr>
<td>MH-60</td>
<td>0.15</td>
<td>Low</td>
<td>UAV</td>
<td>0</td>
<td>Low</td>
</tr>
<tr>
<td>MH-60&lt;sub&gt;backup&lt;/sub&gt;</td>
<td>0.10</td>
<td>Low</td>
<td>USV</td>
<td>0</td>
<td>Low</td>
</tr>
</tbody>
</table>

**Improve resilience using stand-in redundancy:**

<table>
<thead>
<tr>
<th>System</th>
<th>$E(SDI)$</th>
<th>Disruption Impact</th>
<th>System</th>
<th>$E(SRI)$</th>
<th>Recovery Importance</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCS</td>
<td>0.50</td>
<td>Medium</td>
<td>UAV</td>
<td>0.60</td>
<td>High</td>
</tr>
<tr>
<td>UAV</td>
<td>0.40</td>
<td>Medium</td>
<td>USV</td>
<td>0.45</td>
<td>Medium</td>
</tr>
<tr>
<td>USV</td>
<td>0.30</td>
<td>Medium</td>
<td>MH-60</td>
<td>0</td>
<td>Low</td>
</tr>
<tr>
<td>MH-60</td>
<td>0.15</td>
<td>Low</td>
<td>LCS</td>
<td>0</td>
<td>Low</td>
</tr>
</tbody>
</table>
Theoretical Refinements

• Cost-benefit analysis using SIMs

• Uncertainty analysis through *sensitivity studies*:
  – How do SIM rankings change when uncertainties are taken into account?
    • Probability that a system is disrupted
    • Probability that recovery systems are available
    • Variations in recovery curves

• *Collaborative recovery*:
  – In many SoSs, post-disruption recovery involves multiple systems
  – SIM analysis can be used to determine combinations of recovery measures that minimize disruptive impacts on SoS performance
Applications in Air Transportation

• Air Traffic Control
  – Assess importance of air traffic control facilities
    • What is the impact on the ATS if Chicago center is disrupted?
  – Decision-support for resilience improvement strategies
    • How important are Indianapolis and Minneapolis to recovery?
    • How can improvements to these centers improve resilience of the ATS?

• Airlines
  – Assess schedule impact
    • Which airports when disrupted and flights when cancelled would have the largest impact on airline operations?

• Airports
  – Evaluate importance of airports
    • What is the impact on the ATS if BOS suffers some hostile attack?
  – Decision-support for resilience improvement strategies
    • How much should airside and landside capacity be increased at other regional airports (such as MHT and ORH) in order to be resilient this disruption?
Applications in Multi-Modal Transportation Networks

- Northeast Corridor (NEC) Transportation SoS
  - NEC is densely populated region
  - Boston, New York City, Philadelphia, and Washington D.C.
  - Road, rail, and air transportation facilities
  - Interest in improving high-speed rail capability
  - SIMs will be used to determine impact of changes to these modes have on overall SoS resilience
Pilot Studies
Pilot Studies

Goal- Advance the AWB maturation from a user-centric perspective

- **Naval Surface Warfare Center Dahlgren Division (NSWCDD):** CRADA being signed for collaborative work on development of AWB tools towards in analyzing interstitial spaces of SoSE engineering environments and assessing Navy’s Integration & Interoperability initiatives.

- **MITRE Systems Engineering Technical Center:** 2-month activity to test usability of AWB on customer-inspired problems in the SoS space. Provide feedback to Purdue team on AWB and recommendations for enhancement.

- **Army Always-On / On-Demand (AO/OD):** Initial problem set-up and on-site use of AWB to explore tailoring to support Army AO/OD initiatives for 2014-2015 period.
Army Always-On / On-Demand
Army Always-On On Demand (AO-OD)

• “Always On-On Demand” is the integration of existing live, virtual, and constructive systems to create a persistent realistic and relevant Technical / Operational Synthetic Environment to address technology and process issues and gaps associated with Operational Networks and Net-Centric Systems that is available on demand.

SoS-AWB: Explore tradeoffs (performance-cost) from different combination of live and simulated elements.
On-Site Visit to AO-OD Orlando

• Reiterate understanding of Always On/On Demand primary decision problem in providing value for the Army, both "as-is and to-be"; intent to help improve the to-be.

• Translate a portion or abstracted version of this decision-problem into a SoS AWB compatible formulation

• Generate initial SoS AWB results for concept AO-OD relevant problem.

• Establish forward steps towards a deeper understanding of AO-OD problem space and refinement of tools for real world support of AO-OD FY-15/16 event.
On-Site Visit to AO-OD Orlando

<table>
<thead>
<tr>
<th></th>
<th>DIS</th>
<th>HLA</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAT-Missile</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Missile-RADAR</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>RADAR-MAOC</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>F-16-RADAR</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>F-16 MAOC</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>MAOC-F-16</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>MAOC-PAT</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>PAT-MAOC</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>PAT-RADAR</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>AME</th>
<th>ONESAF</th>
<th>FMS</th>
<th>LIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Missile</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Patriot</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>F-16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>MAOC</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>RADAR</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Current Effort with AO-OD

- Translating DoDAF inputs of ODENN EY-14 event scenario to SoS AWB Inputs
- Extend concept IAMD problem started in visit to include more detailed architecture information
- Show to-be capability of SoS AWB in providing value to Army in reducing costs and exploring tradeoffs with various architectures
MITRE SoS analysis experimentation
Background

• Purdue is developing a SoS Analysis Workbench

• Workbench is ready for initial 3rd party use
  – Help solve real problems
  – Provide feedback to Purdue for potential improvements and enhancements

• MITRE has funding thru December 2014 to provide initial assessment

• Expectation is this Purdue/MITRE collaboration will extend beyond December
## Summary Plan

<table>
<thead>
<tr>
<th>Tool</th>
<th>Problem</th>
<th>Starting Point</th>
<th>MITRE POC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDNA</td>
<td>Sequencing Tasks to Minimize Schedule</td>
<td>OpenBMC2 Blue Team Schedule</td>
<td>Javier</td>
</tr>
<tr>
<td>RPO, FDNA, SIM</td>
<td>Force Mix for Ground Kill Chain</td>
<td></td>
<td>Javier</td>
</tr>
<tr>
<td></td>
<td>- Vary sensor types, weapon types, targets to prosecute, red force capability, battlespace conditions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Measures of Performance: prosecution time of TSTs, range of surveillance, cost</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Minimize SoS cost</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Resilience and performance analysis in the event of failures of systems and/or communication links between systems (FDNA and SIM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Minimize loss of SoS capability during evolution to lower cost SoS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADP</td>
<td>Claims Processing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Work Plan

• Purdue provides initial tools and training (12 – 13 Nov)
• MITRE develops overall CY 2014 assessment approach (19 Nov)
  – Use each of the 4 available Workbench tools
    • Fairly simple customer inspired SoS problems (but won’t solve real problem)
    • Aim is to get some feedback to Purdue on all tools by end of December
      – Don’t get bogged down on one specific tool/problem
    – MITRE engage BCS-F to define post-Dec task that uses Workbench
      • Include Dan DeLaurentis
    – Begin working real customer problem as time/funding allows
• Collaboration
  – MITRE provide 12 – 13 Nov trip report to Purdue with initial feedback
  – Telephone call for MITRE to present planned Nov/Dec simple problem set
  – Purdue available to directly work some of the solution problem
    • Minimize else reduced opportunity to provide feedback on the tools
  – MITRE provides at least one midterm and final feedback assessments
  – Phone calls/e-mails as required
Other Related Efforts: SysML-MATLAB

- Currently developing/building translation between SysML logical/physical architectures to Discrete Agent Framework (DAF)
- Many architectures defined using SysML (MagicDraw)
- Translate to MATLAB (DAF) to simulate defined architecture as agent model
- Interface with SoS AWB toolsets in MATLAB environment
RT-108 AWB Near and Far Term Plans

- Mature AWB through collaborative exchange (AO-OD, etc)

- Planned deployment for broader DoD-SE community using HubZero technology → tighter integration with data input definitions (e.g. DoDAF) & SysML

- Development of visual analytics driven interface for toolset (tradespace visualization)
Demonstration Video

- Set of demonstration videos currently being updated
- MATLAB based GUI demonstration for each method
- Naval Warfare Scenario (NWS) example application
Backup Slides
FDNA/DDNA
Functional Dependency Network Analysis (FDNA) and Development Dependency Network Analysis (DDNA)

Methods to analyze dependencies and their effect (FDNA, DDNA)

Support decision in design and architecture. Facilitate trade-off between competing metrics

Compare architectures, answer "what if" questions, facilitate trade-off

Global metrics

Integrated methods (FDNA, DDNA)

Lifecycle metrics
FDNA/DDNA steps

Accordingly define suitable Functional Dependency Networks and Development Dependency Network.

State the problem and the desired results.

Perform the required analysis (FDNA only, DDNA only, or combined).

Define the proper meaning of Operability, Self-Effectiveness. Evaluate FDNA/DDNA input parameters. Define the metrics of interest.

Perform the required "what if" analysis, or stochastic analysis. Determine criticalities and reliability.

Analyze expected schedule, critical paths, possible delay absorption.

Analyze global features, assess partial operability, perform trade-off between competing metrics.

New architectures.
Background:

Functional Dependency Network Analysis (FDNA)

Functional Dependency Network Analysis (FDNA): Method to analyze and quantify interdependencies and cascading effects of disruptions on operability through networks of systems. **Operability is a normalized Measure of Effectiveness (MoE).**

FDNA representation: operability depends on **internal status** (Self-Effectiveness SE), and **Strength, Criticality, and Impact of Dependency (SOD, COD, IOD)**

Effects of failures in 2 systems: probability of the operability of interest

- C. Guariniello, and D. DeLaurentis, "Dependency Analysis of System-of-Systems Operational and Development Networks", CSER 2013
- NASA notional architectural elements for SLS Mission to Mars
Background:
Functional Dependency Network Analysis (FDNA)

- FDNA computes the operability of nodes, based on:
  - **Self-Effectiveness (SE)**
    the internal status
  - **Strength of Dependency (SOD)**
    how much of a system's operability depends on the feeder systems
  - **Criticality of Dependency (COD)**
    loss in operability when the feeders fail completely or almost completely
  - **Impact of Dependency (IOD)**
    how wide is the "COD zone"

- Values can be assessed through ABM simulation, historical/experimental data, expert opinion, etc.

- C. Guariniello, and D. DeLaurentis, "Dependency Analysis of System-of-Systems Operational and Development Networks", CSER 2013
When parameters of dependency are unknown, simulate all alternative architectures

Observed behavior (when the probability distribution of the Self-Effectiveness of systems $i$ and $j$ is uniform between 0 and 100, and system $k$ is working at maximum Self-Effectiveness):

- Probability density function of the Self-Effectiveness of systems $i$ and $j$
- Histogram of instances having a given operability of system $k$ (10000 total instances)
  - $E(O_k)=60.1$
  - $\sigma(O_k)=19.1$
Alternative architecture 1

System $i$ is supported by a redundant system (with uniform probability distribution of the Self-Effectiveness)

Histogram of instances having a given operability of system $k$ (10000 total instances)

- $E(O_k) = 59.9$
- $\sigma(O_k) = 18.8$

No substantial change
Alternative architecture 2

System $j$ is supported by a redundant system (with uniform probability distribution of the Self-Effectiveness)

Histogram of instances having a given operability of system $k$ (10000 total instances)

$E(O_k)=67.9$  \hspace{1cm} $\sigma(O_k)=13.7$
Alternative architecture 3

System $j$ is supported by a redundant system (with uniform probability distribution of the Self-Effectiveness)

Histogram of instances having a given operability of system $k$ (10000 total instances)

$E(O_k)=69.2$  $\sigma(O_k)=11.5$

Slight further improvement
When FDNA parameters are known: informed decision

- Dependency of $j$ from $i$ is weak
- Node $j$ is weak
- Dependency of $j$ from $i$ is weak
- Impact of $j$ on $k$ is strong

Improving the robustness of node $i$ won't have much impact

Node $j$ is critical

One possible improvement is to make this node redundant

Other improvements could involve increasing the robustness of node $j$, or decreasing the dependency of node $k$ from node $j$, if possible.
Background: Development Dependency Network Analysis (DDNA)

Development Dependency Network Analysis (DDNA): Method to analyze and quantify the effect of partial dependencies on the development of Systems-of-Systems. **Operability** here quantifies how well the system is following its expected best development time.

Partial dependency allows for partial delay absorption even on the critical path (PERT does not allow to do this)

Parameters:
**Strength of Dependency (SOD):** how much is the amount of early development of the receiver system \( j \) that can be executed before a feeder system \( i \) is fully developed (the less the SOD, the less the development of system \( j \) depend on that of system \( i \)).

**Criticality of Dependency (COD):** what is the minimum operability of feeder system \( i \) (system \( i \) must have been being developed with not too much delays) that allows for early start in development of system \( j \).

C. Guariniello, and D. DeLaurentis, "Dependency Analysis of System-of-Systems Operational and Development Networks", CSER 2013
Background:
Development Dependency Network Analysis (DDNA)

- DDNA results in a schedule of development (here reported as a Gantt chart)
- Final development, which accounts for partial dependencies, requires a shorter time than that resulting from PERT
- Delays, as well as stakeholders decisions, may cause change in the schedule, according to the development dependencies. This in turn impacts the capabilities, according to the functional dependencies
FDNA/DDNA

Examples of metrics and dependency analysis
Partial operability is achieved over time, based on the development and deployment of the systems (or events), and on their functional dependencies.
Partial parallel development and delay absorption

- DDNA is used to schedule partial parallel development that can be achieved, when partial development dependencies are considered.

- Since development dependencies are partial, delays may be absorbed, at least in part, even on the critical path.
Partial capabilities

- Combining FDNA and DDNA allows us to quantify the partial capabilities that can be achieved during the development of the SoS

- When other considerations are added, such as delays and critical impact of nodes, architectures can be compared based on the trade-off between the evolution of their features over time and partial capabilities.
FDNA/DDNA

Examples of FDNA applications
# Naval Warfare Scenario

<table>
<thead>
<tr>
<th>SoS</th>
<th>Systems</th>
<th>Self-effectiveness</th>
<th>Operability</th>
<th>SOD, COD, IOD</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naval</td>
<td>Ship's Recon Systems</td>
<td>Probability of the radar to detect enemy within range</td>
<td>Time to detect the enemy (or % of enemies detected within given time)</td>
<td>SOD: quantifies the loss in operability when part of the required input is missing</td>
<td>Performance, robustness, resilience (flexibility). Trade-off vs. cost</td>
</tr>
<tr>
<td>Warfare</td>
<td>Helicopter's Recon Systems</td>
<td>Probability of the radar to detect enemy within range</td>
<td>Time to detect the enemy (or % of enemies detected within given time)</td>
<td>COD: quantifies the effect of total or almost total loss of input</td>
<td></td>
</tr>
<tr>
<td>Scenario</td>
<td>Ship's Weapon Systems</td>
<td>Probability to engage the enemy, when within range</td>
<td>Time to engage the enemy (or % of enemies engaged within given time)</td>
<td>IOD: quantifies the range of &quot;almost total loss of input&quot; that results in critical loss of operability</td>
<td></td>
</tr>
</tbody>
</table>

**SOD**: quantifies the loss in operability when part of the required input is missing.

**COD**: quantifies the effect of total or almost total loss of input.

**IOD**: quantifies the range of "almost total loss of input" that results in critical loss of operability.
# On-orbit satellite servicing SoS

<table>
<thead>
<tr>
<th>SoS</th>
<th>Systems</th>
<th>Self-effectiveness</th>
<th>Operability</th>
<th>SOD, COD, IOD</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-orbit satellite servicing</td>
<td>Operational satellite component</td>
<td>Internal operational status</td>
<td>Overall status due to internal status and inputs</td>
<td>SOD: quantifies the loss in operability when part of the required input is missing</td>
<td>Dynamic performance over time, robustness to aging and failures, resilience due to servicing. Trade-off vs. cost</td>
</tr>
<tr>
<td>Operational satellite</td>
<td>Internal operational status due to component status and dependencies</td>
<td>Overall status due to internal status and inputs</td>
<td>COD: quantifies the effect of total or almost total loss of input</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Servicing satellite</td>
<td>-</td>
<td>(servicing satellites modify the self-effectiveness of serviced components)</td>
<td>IOD: quantifies the range of &quot;almost total loss of input&quot; that results in critical loss of operability</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Cybersecurity

<table>
<thead>
<tr>
<th>SoS</th>
<th>Systems</th>
<th>Self-effectiveness</th>
<th>Operability</th>
<th>SOD, COD, IOD</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>IT SoS</td>
<td>IT Systems</td>
<td>Internal status: generation and transmission of correct information</td>
<td>Generation and transmission / retransmission of correct information, given the internal status, the input, and possible cyberattacks</td>
<td>SOD: quantifies the loss in operability when part of the required input is missing</td>
<td>Performance, robustness, resilience (flexibility).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>COD: quantifies the effect of total or almost total loss of input</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>IOD: quantifies the range of &quot;almost total loss of input&quot; that results in critical loss of operability</td>
<td></td>
</tr>
</tbody>
</table>
Robust Portfolio Optimization (RPO)
Robust Portfolio Optimization

- Treat SoS as ‘portfolio’ of systems
- Model individual systems as ‘nodes’
  - Functional & Physical representation
- Rules for node connectivity
  - Compatibility between nodes
  - Bandwidth of linkages
  - Supply (Capability)
  - Demand (Requirements)
  - Relay capability
- Represent as mathematical programming problem
Robust Portfolio Optimization

Decision support approach from financial engineering/operations research to identify ‘portfolios’ of systems by leveraging performance against risk under uncertainties

- Represent behaviors as connectivity constraints
- Employ robust optimization techniques to deal with data uncertainty
- Computationally efficient tools to solve even for very large number of nodes

Robustification to include data uncertainties
General Optimization Problem

Objective
Maximize Performance Index

s.t.

- Requirement Satisfaction
  \[ \sum_q S_{qc} X_q^B \geq \sum_q S_{qB} X_q^B \] (Satisfy Requirements)

- Big-M Formulation (number of connections)
  \[ \sum_c X_{cij} - X_{ij} M \leq 0 \]
  \[ M \sum_c X_{cij} - X_{ij} \geq 0 \]

- Flow Balance Constraint
  \[ \sum_i X_{cij} - \sum_j X_{cij} = 0 \]

- Bandwidth Limit
  \[ X_{cij} \leq \text{Limit}_{cij} \]

- Node Connection Compatibility
  \[ X_1^B + \ldots + X_n^B = D \] (System Compatibility)
  \[ (e.g. X_1^B + X_1^B + X_1^B = 1) \]
  \[ X_{cij} = 0 \quad i,j \in \{\text{incompatible set}\} \]
  \[ \in \{0,1\} \text{(binary)} \]
Dealing with Uncertainty

- **Entities**
  - **System Capability**: Actual performance of system individually and as a whole SoS entity
  - **System Interdependence**: Interdependencies between systems and effects on translation of capability uncertainties

- **Addressing data uncertainty in portfolio selection**
  - Uncertainties in node (system) performance and connections (links)
  - Capture variation in performance at each node as **uncertainty sets**.
  - Variations/uncertainty bounds from ABM simulation or design choice.
Robust Operational Constraints

- Use Bertsimas-Sim approach to uncertain (data uncertainty) constraints
- Benefits: Linear Programming approach, constraint violation control with probabilistic guarantees, extends to discrete optimization

Adjust conservatism $\Gamma_i$ term to control probability of constraint violation

Conservatism Added
(This can be converted to an LP == easy to solve even for large problems)
NWS Communications Layer Analysis

- Build in robustness for communications layer subject to uncertainties in performance
- Robustness of ‘requirements for communications capability being met’

<table>
<thead>
<tr>
<th>Systems</th>
<th>Available System Packages</th>
<th>Gamma (Level of Conservatism)</th>
<th>0.01</th>
<th>0.21</th>
<th>0.41</th>
<th>0.61</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASW</td>
<td>Variable Depth</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Multi Fcn Tow</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>lightweight Tow</td>
<td>Lightweight Tow</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MCN</td>
<td>RAMCS II</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ALMDS (MH-60)</td>
<td>ALMDS (MH-60)</td>
<td></td>
<td>x</td>
<td>x</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SUW</td>
<td>N-LOS Missiles</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Griffin Missiles</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td>Seaframe</td>
<td>Package 1</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Package 2</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Package 3</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Comm.</td>
<td>System 1</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>System 2</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>System 3</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>System 4</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>System 5</td>
<td></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>System 6</td>
<td></td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>-</td>
</tr>
</tbody>
</table>

Trade SoS Performance for Comm. Conservatism (e.g. against cyber-attack)

Portfolios of systems at prescribed conservatism
Multi-Metrics: Power & Comm. Layer Analysis

Build in robustness for communications and power layer simultaneously

Robustness to constraint violation of ‘requirements for communications and power generation capability being met’ → Tradespace analysis

Each point is a collection of systems

Probabilistic guarantees on constraint violation for multiple dimensions

Trade Comm. Conservatism Against other metrics (e.g. Power Layer)
Approximate Dynamic Programming
Problem Description

Decision Makers

- DoD
  - Army
  - Navy
  - Air Force

Funding Interdependency

- Army
- Navy
- Air Force

Technological Interdependency

Army Planning

- Develop Systems?
  - t1
  - t2
  - t3

Exogenous Information (Uncertainty)
Original Concept of Transfer (Price) Contract

- **Business Unit A** (produce product w)
- **Business Unit B** (consume product w; produce product p)
- **Transfer price**
- **External Market**
Our Interpretation of Transfer Contract

• Original meaning of transfer contract
  – Revenue transferred from one **business unit** to another
  – Similar to shadow price: marginal utility of relaxing the constraint

• Meaning in acquisition field
  – Information transferred from one **service** to another
  – Partial capability (utility) or equivalent monetary value transferred from one service to another
Numerical Experiment Setup

Decision Makers

Stakeholder A (Army) (S1a, S2a)
Stakeholder B (Navy) (S1b, S2b)
Stakeholder C (Air Force) (S1c, S2c)

SoS-level (DoD)

Maximize the capability of terrestrial / maritime surface surveillance

Maximize the capability of terrestrial surface surveillance

Planning of A

Observe change of system capability contribution level

Stakeholder A (Army)

S1a (radar)

S2a (aircraft)
Numerical Experiments – Deterministic and Centralized

- Capability contribution objective (first stage value) obtained from ADP is 0.28% lower than the exact objective; it converges within 20 iterations.
- Change the basis function to the vector of resource number and a constant value, the accuracy of the ADP results (first stage value) is 10% lower than the exact optimum.

Values differ because the basis functions cannot capture the future values over only ten stages.
Numerical Experiments – Stochastic and Centralized

- Uncertainty of the individual system capability contribution level is modeled as a uniform distribution.
- The objective from ADP fluctuates within 10%.
Numerical Experiments – Decentralized / No Coordination

- **Left**: The capability contribution of the global and the individual stakeholders in the decentralized case without any coordination.

- **Right**: The budget constraint is violated because of the mis-specified model and lack of coordination.
Numerical Experiments – Decentralized / Coordination

- The figures show the capability value and budget violation with an arbitrary transfer contract coordination.
- There is a visible change in the capability contribution of different stakeholders and corresponding budget violation.
Numerical Experiments – Decentralized / Coordination

- Conjecture
  - Decentralized optimum should match the centralized optimum
  - The compensation received by an individual stakeholder equals the future value for this stakeholder losing the resources consumed by another stakeholder

- The global optimum reaches 7000, which is the same as the centralized optimum.
- Current work is exploring the effect of relaxing the assumptions
  - E.g., generalize the model specification of other stakeholders