Evolvability-Related Options in Military Systems of Systems

Nicola Ricci, Donna H. Rhodes, and Adam M Ross
Massachusetts Institute of Technology

March 22, 2014
Overview

- Intro
  - Motivation
  - Military SoS
- Framework
- Evolvability-related Options Examples
- Discussion & Next Steps
Overview

- Intro
  - Motivation
  - Military SoS
- Framework
- Evolvability-related Options Examples
- Discussion & Next Steps
Context – MIT SAI Method

Context – MIT SAI Method
Motivation

SoS Engineering
• Ongoing analysis
• Periodic **re-architecting** activity
• Evolution increments

What are ways in which evolvability can be embedded in the SoS?

The research aims to:
1) further **validate the usefulness of design principles** for architecting systems that possess desirable lifecycle properties such as **evolvability**.
2) contribute **real-world examples** that architects may use to inspire specific design options for their system of interest.

Designing for evolvability enables systems architects to avoid architecture alternatives that are well-suited to present needs, yet unable to sustain value delivery over the lifespan of the SoS.
Military Systems of Systems Mission SoS

BMDS - THE BALLISTIC MISSILE DEFENSE SYSTEM

BMDS
Ballistic Missile Defense System

BCT
Brigade Combat Team Modernization

JIE
DISA - Joint Information Enterprise

C2BMC
Command, Control, Battle Management and Communications

NMCC
USSTRATCOM
USNORTHCOM
USPACOM
EUROCOM
CENTCOM
Military Systems of Systems
IT SoS

BMDS - THE BALLISTIC MISSILE DEFENSE SYSTEM

Joint Information Enterprise – End State

- Defensibility/Redundancy/Resiliency
- Federation/Shared Infrastructure
- Enterprise Services
- Identity Access Management
- Cost: ????

“Enterprise Information Environment”

Access at the Point of Need


Unclassified

© 2014 Massachusetts Institute of Technology
Overview

• Intro
  – Motivation
  – Military SoS

• Framework

• Evolvability-related Options Examples

• Discussion & Next Steps
Evolutionary changes that SoS have undergone (or are undergoing) have been examined through the lenses of two constructs: **design principles** and **change options**.

- **Integrability**
- **Hard Point**
- **Change Payload**

**Design Principle** → **Path Enabler** → **Change Mechanism**
Design Principles

"Guiding thoughts [for design] based on empirical deduction of observed behaviour or practices that prove to be true under most conditions over time"

(Wasson – 2006)

<table>
<thead>
<tr>
<th>Desired Ility</th>
<th>Design Principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survivability</td>
<td>Concealment</td>
</tr>
<tr>
<td>Survivability</td>
<td>Hardness</td>
</tr>
<tr>
<td>Evolvability</td>
<td>Targeted Modularity</td>
</tr>
<tr>
<td>Evolvability</td>
<td>Decentralization</td>
</tr>
<tr>
<td>Survivability</td>
<td>Heterogeneity</td>
</tr>
</tbody>
</table>

→ Option
→ Option
→ Option
→ Option

(Richards, Ross, Hastings and Rhodes – 2009)
### Strategies

#### Design Principles

<table>
<thead>
<tr>
<th>Strategies</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Leverage Ancestry</strong></td>
<td>Employing successful design choices of assets, capabilities and/or operations from all prior generations of the system</td>
</tr>
<tr>
<td><strong>Disruptive Architectural Overhaul</strong></td>
<td>Re-architecting significant portions of the existing system or program at the same time in order to reduce the negative impact that making many smaller changes would have</td>
</tr>
<tr>
<td><strong>Mimicry</strong></td>
<td>Imitating or duplicating successful design choices of assets, capabilities and/or operations from other systems/domains for a similar purpose</td>
</tr>
<tr>
<td><strong>Resourceful Exaptation</strong></td>
<td>Repurposing assets or design choices from prior generations or other systems/domain in order to provide capabilities for which they were not originally selected</td>
</tr>
</tbody>
</table>
## Structual Design Principles

<table>
<thead>
<tr>
<th>Design Principle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Leverage Ancestry</strong></td>
<td>Employing successful design choices of assets, capabilities and/or operations from all prior generations of the system.</td>
</tr>
<tr>
<td><strong>Disruptive Architectural Overhaul</strong></td>
<td>Re-architecting significant portions of the existing system or program at the same time in order to reduce the negative impact that making many smaller changes would have.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Design Principle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decentralization</strong></td>
<td>Distributing assets, capabilities and/or operations to appropriate multiple locations, rather than having them located in a single location.</td>
</tr>
<tr>
<td><strong>Targeted Modularity</strong></td>
<td>Isolating parts of the system to reduce interdependencies in order to limit undesirable effects caused by either uncertainties or intentional changes.</td>
</tr>
<tr>
<td><strong>Integrability</strong></td>
<td>Designing interfaces for compatibility and commonality to enable effective and efficient integration of upgraded/new system components and constituents.</td>
</tr>
<tr>
<td><strong>Reconfigurability</strong></td>
<td>Creating intentional similarities in form and/or function of various system assets, capabilities, and/or operations to facilitate reuse or reallocation.</td>
</tr>
<tr>
<td><strong>Redundancy</strong></td>
<td>Intentional duplication of selected assets, capabilities and/or operations to enable their future redistribution without compromising existing requirements.</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>Making design choices that allow scaling of resources and/or assets up or down in order to accommodate uncertainties and emergent needs.</td>
</tr>
<tr>
<td><strong>Margin</strong></td>
<td>Architecting for intentional excess capacity in specific capabilities and/or operations to meet emergent needs without compromising existing requirements (i.e. meet or exceed future requirements).</td>
</tr>
<tr>
<td><strong>Slack</strong></td>
<td>Intentionally under-allocating or over-allocating specific available assets and/or resources in order to reserve excess capacity for accommodating uncertainties (i.e. prevent violation of constraints).</td>
</tr>
</tbody>
</table>
Leverage Ancestry

Employing successful design choices of assets, capabilities and/or operations from all prior generations of the system

Disruptive Architectural Overhaul

Re-architecting significant portions of the existing system or program at the same time in order to reduce the negative impact that making many smaller changes would have

Mimicry

Imitating or duplicating successful design choices of assets, capabilities and/or operations from other systems/domains for a similar purpose

Resourceful Exaptation

Repurposing assets or design choices from prior generations or other systems/domain in order to provide capabilities for which they were not originally selected

Strategies

Design Principles

Decentralization

Distributing assets, capabilities and/or operations to appropriate multiple locations, rather than having them located in a single location

Targeted Modularity

Isolating parts of the system to reduce interdependencies in order to limit undesirable effects caused by either uncertainties or intentional changes

Integrability

Designing interfaces for compatibility and commonality to enable effective and efficient integration of upgraded/new system components and constituents

Reconfigurability

Creating intentional similarities in form and/or function of various system assets, capabilities, and/or operations to facilitate reuse or reallocation

Redundancy

Intentional duplication of selected assets, capabilities and/or operations to enable their future redistribution without compromising existing requirements

Scalability

Making design choices that allow scaling of resources and/or assets up or down in order to accommodate uncertainties and emergent needs

Margin

Architecting for intentional excess capacity in specific capabilities and/or operations to meet emergent needs without compromising existing requirements (i.e. meet or exceed future requirements)

Slack

Intentionally under-allocating or over-allocating specific available assets and/or resources in order to reserve excess capacity for accommodating uncertainties (i.e. prevent violation of constraints)

Structural Design Principles

Architect’s Intent

- Desire to Add New Constituent System
- Desire to Add More of Existing Constituent System
- Desire to Replace or Upgrade Capabilities of Existing Constituent System
- Desire to Add More of Existing Function
- Desire to Change the Way in Which a Function is Performed
- Desire to Physically Relocate Resources/Capabilities
The **change mechanism** is the **method** through which a system goes from state A to state B (e.g., *swapping payload on UAV*).

The **path enabler** (i.e., a physical object, an action or a decision) is **what** gives the “option” of executing the change mechanism (e.g., *modular payload bay in original design*).
Overview

• Intro
  – Motivation
  – Military SoS

• Framework

• Evolvability-related Options Examples

• Discussion & Next Steps
By 2004, BMDS used **PAC-3** and (when available) **SM-3** (sea-based) as short-term ballistic missile defense for the terminal segment.

In 2008, **THAAD** (land-based) was tested and successfully added to the SoS for terminal defense segment.

---

**INTEGRABILITY**

**C2BMC (comms and data-management backbone)**

allows

integrating the new THAAD interceptor

---

**C2BMC** → Purposely designed **path enabler**

- Tactical Datalink 16
- Extremely High Frequency (EHF)
- Global Engagement Manager (GEM)
- access control lists – ACLs
As of 2009, BMDS had its interceptors (PAC-3, THAAD, SM-3 and GMD) on U.S. land (AK and CA). Communication and tracking capabilities were located in space, U.S. territory and oceans (SBX radar).

U.S. was going to seek a Phased Adaptive Approach (PAA) to missile defense in Europe. The SoS was planned to be extended to multiple bases in the Mediterranean Sea, as well as some European countries.

In September 2009, President Barack Obama announced that the U.S. was going to seek a Phased Adaptive Approach (PAA) to missile defense in Europe, also thanks to U.S. long-standing relationship with some of the European countries.
BMDS – Phase Adaptive Approach

“Desire to add more of existing function”

INTRODUCED

- **SLACK** inspires Agreement with European countries (Poland and Romania) allows expanding the SoS from Mediterranean bases to existing land facilities

Option

Intentionally under-allocating or over-allocating specific available assets and/or resources in order to reserve excess capacity for accommodating uncertainties (i.e. prevent violation of constraints)

- **Phase 1** – deployed
- **Phase 2** – planned deployment date: 2015
  - Phase 2 will see interceptors taken onto land in the first "Aegis-Ashore" deployment in Romania.
- **Phase 3** – planned deployment date: 2018
  - Phase 3 will see the introduction of the second “Aegis-Ashore” site in Poland with another SPY-1 radar and 24 SM-3 missiles

In September 2009, President Barack Obama announced that the U.S. was going to seek a Phased Adaptive Approach (PAA) to missile defense in Europe, also thanks to U.S. long-standing relationship with some of the European countries.
Overview

• Intro
  – Motivation
  – Military SoS

• Framework

• Evolvability-related Options Examples

• Discussion & Next Steps
Research in early stage, but some preliminary patterns are starting to be observed:

- Every design principle was encountered across all SoS types and evolution increments
- Some design principles appear to be less frequent than others

<table>
<thead>
<tr>
<th>Design Principles</th>
<th>Options Identified</th>
<th>Mission SoS - BMDS</th>
<th>Platform SoS - BCT</th>
<th>IT SoSs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>THAAD PAA Brilliant Pebbles Aegis BMD SBX Radar Moderniz. Program GCV MRAP CS 13 WIN-T TBMCS DISA UC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Decentralization</td>
<td></td>
<td>1 2 3 4 1 2 3 4 1 2 3 4 1 2 3 4 1 2 3</td>
<td>1 2 3 4 5 1 2 3 4 5 1 2 3 1 2 3</td>
<td>1 2 3 1 2 3 4 5 1 2 3</td>
</tr>
<tr>
<td>Leverage New Technology</td>
<td></td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Integrability</td>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Leverage Ancestors</td>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Margin</td>
<td></td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Mimicry</td>
<td></td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Reconfigurability</td>
<td></td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Redundancy</td>
<td>0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Resourceful Exaptation</td>
<td></td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Scalability</td>
<td></td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Slack</td>
<td></td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Targeted Modularity</td>
<td></td>
<td>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0 0 0 0 0 0 0 0 0 0 0 0</td>
<td>0</td>
</tr>
</tbody>
</table>

More info: seari.mit.edu

Presented to the Conference on Systems Engineering Research (CSER) 2014

© 2014 Massachusetts Institute of Technology
Next Steps

• **Additional case examples** to further validate *evolvability* design principles
  – Long-term goal is to have enough data points to be able to discern *differences in use of design principles*

• **SoS types discussed** (mission, platform and IT) are **not collectively exhaustive**, but they are mutually exclusive
  – Gather evidence of usefulness of evolvability design principles for different SoS types
  – Future work will *expand the current set.*

• **Ongoing research also looking at SoS evolvability with respect to what is specifically evolving**
  – **SoS program** (e.g., FCS evolving to BCT Modernization)
  – The **SoS architecture** as a whole
  – **Constituent system** that impacts the evolvability of the SoS as a whole
QUESTIONS?

Contact info:

Nicola Ricci – nricci@mit.edu
Donna H Rhodes – rhodes@mit.edu
Adam M Ross – adamross@mit.edu

MIT SEArI – seari.mit.edu

EVOLVABILITY-RELATED OPTIONS IN MILITARY SYSTEMS OF SYSTEMS
Back-up Slides
Motivation

The world in which Systems Engineers practice has undergone a significant metamorphosis over the past twenty years

- **Advent of the internet** → great increase in amount of resources available
- **Information travels at the speed of light** → instantaneous communication
- **High-speed computation** → Performance of very complex analyses

Systems are subject to highly dynamic operational environments

- A multitude of exogenous uncertainties can impact a system
  - Geo-political shifts (e.g., policy/regulation changes)
  - Disruptive technologies (e.g., advent of GPS)
  - Market variations (e.g., price &demand variations)

- Unanticipated shifts in stakeholder needs
  - Change of preferences
  - Change of mission objectives

- Systems of Systems
  - Managerial and operational independence
  - Continually evolving

If focused solely on the present state of the world, engineers may encounter the problem of designing systems that may not be well positioned in the future, forced to operate in contexts for which they were not conceived, and delivering capabilities no longer of interest to stakeholders.
• Modern *ilities* (e.g. flexibility) are one response to mitigate the impact of dynamic complexities on system value over time.

*ilities* are “*properties of engineering systems that often manifest and determine value after a system is put into initial use. Rather than being primary functional requirements, these properties concern wider impacts with respect to time and stakeholders.***”

(de Weck, Ross, and Rhodes – 2012)

• The purpose of any system is to provide some level of value to its stakeholders, whose interest is that system *keep providing value* throughout the expected lifecycle

• System designers attempt to make systems that can provide value in spite of exogenous perturbations → i.e., devise value sustainment strategies

(Ricci, Ross, and Rhodes – 2013)
Evolvability

**Evolvability:** The ability of an architecture to be inherited and changed across generations [over time]

- Collected different concepts and ideas regarding evolvability from various fields (from biology, to technology innovation, to systems engineering)
- Definition of evolvability rests upon a key distinction among systems’ architecture, design and instance

**BIOLOGY**
“the ability of a population to both generate and use genetic variation to respond to natural selection”
(Colagrace N, Collins S. – 2008)

**SYSTEMS ENGINEERING**
“evolvable systems as being able to accommodate adaptive variations in some locales without changing the behaviour of subsystems in other locales”
(Sussman GJ – 2007)

**SoS ENGINEERING**
“ability of the architecture to handle future upgrades “
(Butterfield ML, Pearlman JS, Vickroy SC – 2008)
The mobility (subscription-based) services used – unclassified or classified – are **scalable** to accommodate increasing (or decreasing) numbers of users.

- As such they represent a **path enabler**
- The SoS architecture uses commercial carriers for providing subscription-based services. Commercial carriers and other unclassified access networks provide **controlled connectivity** between users and mobile enterprise.

- Operational capability is expected to **grow** with subscription-based services up to **100,000 devices** – starting from 1,500 initial devices.