Linking Systems Engineering Artifacts with Complex System Maturity Assessments

2009 NDIA Systems Engineering Conference
28 October 2008

Brian Sauser, Ph.D.
Stevens Institute of Technology

Richard Volkert
SSC-Pacific

Eric Forbes
Northrop Grumman Corporation

Lance Harper
Northrop Grumman Corporation
Overview

• Motivation

• System Acquisition Management Approach

• System Readiness Level Concept Overview

• System Maturity Assessment Process

• System Performance Level Monitoring

• System Availability

• System Capability Satisficing

• Future Work and Applications
Motivation

- Development and acquisition activities continue to be challenged by the formulation of larger and more complex systems.

- This is compounded by the emergence of *Acknowledged Systems of Systems* which are characterized as having multiple stakeholders with competing interests and priorities.

- Traditional management tools continue to be applied, but do not provide a holistic view of development.

- Failure to adequately consider all systems integration challenges has led an environment of cost overruns, schedule slips, and degraded performance.

System Level Program Management Tools

• New methods, processes, and tools are needed in order to effectively manage and optimize complex system development

• Significant management tools exist at the individual technology level, but are limited in application for systems development
  – Technology Readiness Levels:
    Do not consider integration of components into a system
  – Technical Performance Measures:
    Individual component performance does not translate to system level
  – Availability Analysis:
    Multiple system sub-capabilities present different availability options
  – Risk Management:
    Additional unanticipated risk areas are introduced through the linkage of formerly independent systems

• Emerging systems management resources have been few and far between

• DoD’s Systems Engineering Guide for Systems of Systems “acknowledges these issues, but does not make any recommendations for changes to existing management and control structures to resolve inter-system issues”.
The US Navy’s Littoral Combat Ship Mission Modules Program (PEO LMS) in collaboration with the Northrop Grumman Corporation and Stevens Institute of Technology is developing a holistic System Maturity Model for systems development management.
System Maturity Monitoring - TRL Shortcomings

- Application of TRL to systems of technologies is not sufficient to give a holistic picture of complex system of systems readiness
  - TRL is only a measure of an individual technology

- Assessments of several technologies rapidly becomes very complex without a systematic method of comparison

- Multiple TRLs do not provide insight into integrations between technologies nor the maturity of the resulting system
  - Yet most complex systems fail at the integration points

**Individual Technology**

**Can TRL be applied?**

Yes

**System of Technologies**

**Can TRL be applied?**

NO
**Goal:** Institute a robust, repeatable, and agile method to monitor / report system development and integration status

Create a **System Readiness Level (SRL)** that utilizes SME / developer input on technology and integration maturity to provide an objective indication of complex system development maturity

**APPROACH**

- **Technology Readiness Levels (TRL)**
  - Status of technologies making up the system

- **Integration Readiness Levels (IRL)**
  - Status of connections between the technologies

- **System Readiness Levels (SRL)**
  - Overall system maturity appraisal

- Provides a **system-level** view of development maturity with opportunities to drill down to element-level contributions
- Allows managers to evaluate system development in real-time and take **proactive** measures
- Highly **adaptive** to use on a wide array of system engineering development efforts
- Can be applied as a **predictive** tool for technology insertion trade studies and analysis
What is an IRL?

A systematic measurement reflecting the status of an integration connecting two particular technologies

<table>
<thead>
<tr>
<th>IRL</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Integration is <strong>Mission Proven</strong> through successful mission operations.</td>
</tr>
<tr>
<td>8</td>
<td>Actual integration completed and <strong>Mission Qualified</strong> through test and demonstration, in the system environment.</td>
</tr>
<tr>
<td>7</td>
<td>The integration of technologies has been <strong>Verified and Validated</strong> with sufficient detail to be actionable.</td>
</tr>
<tr>
<td>6</td>
<td>The integrating technologies can <strong>Accept, Translate, and Structure Information</strong> for its intended application.</td>
</tr>
<tr>
<td>5</td>
<td>There is sufficient <strong>Control</strong> between technologies necessary to establish, manage, and terminate the integration.</td>
</tr>
<tr>
<td>4</td>
<td>There is sufficient detail in the <strong>Quality and Assurance</strong> of the integration between technologies.</td>
</tr>
<tr>
<td>3</td>
<td>There is <strong>Compatibility</strong> (i.e. common language) between technologies to orderly and efficiently integrate and interact.</td>
</tr>
<tr>
<td>2</td>
<td>There is some level of specificity to characterize the <strong>Interaction</strong> (i.e. ability to influence) between technologies through their interface.</td>
</tr>
<tr>
<td>1</td>
<td>An <strong>Interface</strong> between technologies has been identified with sufficient detail to allow characterization of the relationship.</td>
</tr>
</tbody>
</table>

SRL Calculation Example

**TRL Matrix**

\[
\begin{bmatrix}
 TRL_1 \\
 TRL_2 \\
 TRL_3 \\
\end{bmatrix} =
\begin{bmatrix}
 9 \\
 6 \\
 6 \\
\end{bmatrix}
\]

**IRL Matrix**

\[
\begin{bmatrix}
 IRL_1 & IRL_{12} & IRL_{13} \\
 IRL_{12} & IRL_2 & IRL_{23} \\
 IRL_{13} & IRL_{23} & IRL_3 \\
\end{bmatrix} =
\begin{bmatrix}
 9 & 1 & 0 \\
 1 & 9 & 7 \\
 0 & 7 & 9 \\
\end{bmatrix}
\]

**SRL Matrix**

\[
SRL = IRL \times TRL
\]

(Normalized)

**Component SRL**

\[
\text{Component SRL}_x = \begin{bmatrix}
 SRL_1 \\
 SRL_2 \\
 SRL_3 \\
\end{bmatrix} = \begin{bmatrix}
 0.54 \\
 0.43 \\
 0.59 \\
\end{bmatrix}
\]

Component SRL\_x represents Technology “X” and its IRLs considered

**Composite SRL**

\[
\text{Composite SRL} = \frac{1}{3} \left( SRL_1 + SRL_2 + SRL_3 \right) = \frac{1}{3} \left( 0.54 + 0.43 + 0.59 \right) = 0.52
\]

The Composite SRL provides an overall assessment of the system readiness

For complex systems, the amount of information obtained from the SRL evaluation can be overwhelming.

To maximize applicability SRL outputs are tied to key, program-specific development milestones.

Progress against these milestones provide key insight to the user regarding current program status, risk and progress.
System Maturity Assessment Process

1. Develop System Architectures
   - System architecture provides the foundation for system maturity assessments
   - Functional Capability
     - Software/Hardware

2. Determine Criticality
   - Identification of critical elements and interfaces to be evaluated
   - Critical Elements

3. Build Assessment Process
   - Customize applicable TRL / IRL criteria
   - Build SRL advancement schedule
   - Tie criteria to program test events / milestones

Architectures and framework are locked after approval and will remain so unless the program is re-baselined

4. Conduct System Maturity Analysis w/ SRL
   - Evaluate and Justify TRLs / IRLs
   - Calculate SRL
   - Build Maturity Reports

5. Interpret and Apply Results
   - Identify Risks Against Schedule
     - SRL assessment and test events / milestone gates are at or in advance of schedule
     - SRL assessment is at or in advance of schedule, but test events / milestone gates remain to be closed
     - SRL assessment and test events / milestone gates are behind schedule
   - Outputs of the analysis are analyzed against projected cost and schedule data to determine current development status
   - Future planning can also be conducted through trade-off analyses and risk management activities

Iterate
## System Performance Level Monitoring (PLM)

### Goal:
Predict the ability of a complex systems to achieve required performance

1. Map the Systems to their impacts on key performance parameters

<table>
<thead>
<tr>
<th>Notional System of Systems</th>
<th>KPP Impacted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capability/MS</td>
<td>Search</td>
</tr>
<tr>
<td>Tech 1</td>
<td>X</td>
</tr>
<tr>
<td>Tech 2</td>
<td>x</td>
</tr>
<tr>
<td>Tech 3</td>
<td>X</td>
</tr>
<tr>
<td>Tech 4</td>
<td></td>
</tr>
<tr>
<td>Tech 5</td>
<td>X</td>
</tr>
</tbody>
</table>

2. Map the maturity development of the Systems to the SoS development schedule

<table>
<thead>
<tr>
<th>Notional Maturity</th>
<th>MP Impacted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capability/MS</td>
<td>MP1</td>
</tr>
<tr>
<td>Tech 1</td>
<td>EDM</td>
</tr>
<tr>
<td>Tech 2</td>
<td>ADM</td>
</tr>
<tr>
<td>Tech 3</td>
<td>EDM</td>
</tr>
<tr>
<td>Tech 4</td>
<td>PROD</td>
</tr>
<tr>
<td>Tech 5</td>
<td>PROD</td>
</tr>
</tbody>
</table>

3. Develop a relationship between system usage satisfying a KPP in a SoS and its maturity (in terms of a weighted value) against anticipated performance

\[
T_{1n} = \omega_n \times \alpha
\]

\[
T_{1n} = \omega_n \times \alpha
\]

\[
T_{1n} = \omega_n \times \alpha
\]

\[
T_{1n} = \omega_n \times \alpha
\]
Performance Level Monitoring (PLM)

4. Adjust for usage impact under various employment options

\[
\begin{align*}
\text{CONOPS}_A &= \beta T_{1n} + \gamma T_{5n} \\
\text{CONOPS}_B &= \delta T_{1n} + \varepsilon T_{3n} + \gamma T_{5n} \\
\text{CONOPS}_C &= \theta T_{3n} + \eta T_{5n}
\end{align*}
\]

5. Average the results from individual employment options to obtain insight into ability to achieve obtainment of the desired performance parameter

6. Use predictions of improved maturity (SRL) over time to derive a predicted growth path of performance for SoS

\[
\text{KPP}_{\text{SEARCH}} = [\text{CONOP}_A, \text{CONOP}_B, \text{CONOP}_C] = \text{AVG}(\text{CONOP}_A + \text{CONOP}_B + \text{CONOP}_C)
\]
Performance Level Monitoring (PLM)

7. Use estimates of performance and maturity to define predictions of performance

8. Use variances of the usage rates to establish bands of performance based on varying usage options of the individual systems/modules

9. As data is gathered, updated predictions/calculations to verify if development is proceeding as desired
System Availability

**Goal:** Adapt availability analysis to systems with multiple capabilities

- Defining a subset of system components that contribute to the mission will vary the Availability
  - Increased number of system components weighs heavily on mission function availability
  - Statistical combination of CONOPS and a blending of the contributions will identify the critical components and provide insight into which provide better availability

- Through mission string analysis we gain insight into system functional performance and availability insight linked to CONOPS

- Alternative System/Mission components or CONOPS can help achieve System availability
  - Plan Availability Evolution (Improved Technology Insertion or Obsolescence Removal)
  - Trade improvement options with Program Cost and Schedule, so that in the system roadmap availability increases over the program life cycle

- Modular concept components enable functional expansion across system

- Using Reliability Block Diagram's as a method for picking component insertion/replacement by looking at the available and functional impact across a mission
Goal: Optimize system resource allocation across multiple variables

“What technologies and integrations are important or critical to each architectural view to achieve a functionality or capability?”... “How will the systems maturity vary depending on the architectural variants?”

“What functionalities or capabilities are sufficient, critical, or important to achieving a level of system maturity that can satisfy a warfighter’s needs?”

“What impact does this have on system maturity and ultimately the acquisition of a deployable system?”

“Can we use multi-attribute decision making/techniques in systems maturity assessment; parametric sensitivity analysis on how various TRL/IRL combinations drive SRL; and sensitivity analysis to determine what the most critical technologies are?”

Builds upon the foundational approaches previously defined to maximize system capability for every dollar spent

---

Statement A: Approved for Public Release, Distribution is Unlimited
Analyzing Component Importance

- Analytical approach provides insight into which components and integrations provide greatest contribution to maturity.

- This can then be used to ensure some level of functionality can be attained while full system continues to develop.

- Factors can include performance, schedule, cost, etc…
SRL methodology can be used not only to assess current system maturity status, but also to roadmap and assess future development options along with cost and performance.

Future work will focus on the creation and integration of applications which continue to leverage the SRL foundation to provide a holistic management dashboard and decision environment.

Key Aspects:

- Development of a cost discretization across maturity increments using historical data
- Validation of an approach to monitor planned versus actual system maturity, cost, and schedule
- Linking of requirements and testing to performance and maturity

Applications:

- Future technology insertion, obsolescence, and evolution planning
QUESTIONS?
Back-up
Abstract

In a collaborative research effort that has involved Stevens Institute of Technology’s Systems Development & Maturity Laboratory, the Northrop Grumman Corporation, and the U.S. Navy (PMS 420 / SSC-P), a measure of complex system development maturity entitled System Readiness Level (SRL) has been created. This measurement methodology builds upon the pre-existing Technology Readiness Level (TRL) and incorporates an Integration Readiness Level (IRL) in its formulation and practice. Unfortunately, the use of TRL, and subsequently IRL, in the formulation of SRL means that all of the drawbacks associated with the inherent subjectivity of their evaluation and assessment are carried forward. To address this issue, work was previously done to grow the readiness level definitions from a somewhat ambiguous, single line per level to a series of program tailored guides delineating tasks to be completed to achieve each maturity increment. Though the guides have been a significant step forward, additional work remains to be done in linking these TRL and IRL attributes and SRL increments with system architectures, technical performance measures, and development milestones (i.e. systems engineering artifacts). This is a critical step for two reasons: 1) it enables the tracking of development performance via the number and degree to which the artifacts have been satisfied; 2) it provides the decision maker with insight into the current level of system performance achieved and an understanding of what employment of the system (or a subsystem) at its current level of maturity will provide in terms of overall performance against requirements. Furthermore, a more accurate linkage to program costs can be established by tracking projected versus actual expenditures required to meet each successive level of development maturity. This presentation will review the development, implementation, and verification and validation of this concept as it is being executed with the U.S. Navy’s PMS 420 Program Office.
From a System to an Acknowledged System of Systems

Table 2-1. Comparing Systems and Acknowledged Systems of Systems

<table>
<thead>
<tr>
<th>Aspect of Environment</th>
<th>System</th>
<th>Acknowledged System of Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management &amp; Oversight</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stakeholder Involvement</td>
<td>Clearer set of stakeholders</td>
<td>Stakeholders at both system level and SoS levels (including the system owners), with competing interests and priorities; in some cases, the system stakeholder has no vested interest in the SoS; all stakeholders may not be recognized</td>
</tr>
<tr>
<td>Governance</td>
<td>Aligned PM and funding</td>
<td>Added levels of complexity due to management and funding for both the SoS and individual systems; SoS does not have authority over all the systems</td>
</tr>
<tr>
<td>Operational Environment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operational Focus</td>
<td>Designed and developed to meet operational objectives</td>
<td>Called upon to meet a set of operational objectives using systems whose objectives may or may not align with the SoS objectives</td>
</tr>
<tr>
<td>Implementation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acquisition</td>
<td>Aligned to ACAT Milestones, documented requirements, SE with a Systems Engineering Plan (SEP)</td>
<td>Added complexity due to multiple system lifecycles across acquisition programs, involving legacy systems, systems under development, new developments, and technology insertion; Typically have stated capability objectives upfront which may need to be translated into formal requirements</td>
</tr>
<tr>
<td>Test &amp; Evaluation</td>
<td>Test and evaluation of the system is generally possible</td>
<td>Testing is more challenging due to the difficulty of synchronizing across multiple systems’ life cycles; given the complexity of all the moving parts and potential for unintended consequences</td>
</tr>
<tr>
<td>Engineering &amp; Design Considerations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Boundaries and Interfaces</td>
<td>Focuses on boundaries and interfaces for the single system</td>
<td>Focus on identifying the systems that contribute to the SoS objectives and enabling the flow of data, control and functionality across the SoS while balancing needs of the systems</td>
</tr>
<tr>
<td>Performance &amp; Behavior</td>
<td>Performance of the system to meet specified objectives</td>
<td>Performance across the SoS that satisfies SoS user capability needs while balancing needs of the systems</td>
</tr>
</tbody>
</table>

System of Systems Challenges

SoS increases the complexity, scope, and cost of both the planning process and systems engineering, and introduces the need to coordinate inter-program activities and manage agreements among multiple program managers (PMs) as stakeholders who may not have a vested interest in the SoS. The problems that need to be addressed are large and complex and are not amenable to solution by better systems engineering alone. Without a solid governance and management approach for an SoS, independent authorities who oversee the multiple governance processes of DOD are unlikely to accept guidance from a systems engineer they do not control, placing the systems engineer in an untenable position in attempting to support an SoS. An administrative/governance structure that addresses these realities will enable SoS SE to be more effective in all phases of the processes as outlined in this document. This document acknowledges these issues but does not make any recommendations for changes to existing management and control structures to resolve inter-system issues.
SRL Calculation

- The SRL is not user defined, but is instead based on the outcomes of the documented TRL and IRL evaluations.

- Through mathematically combining these two separate readiness levels, a better picture of overall complex system readiness is obtained by examining all technologies in concert with all of their required integrations.

\[
SRL = IRL \times TRL
\]

\[
\begin{pmatrix}
SRL_1 & SRL_2 & SRL_3 \\
\end{pmatrix}
= \begin{pmatrix}
IRL_{11} & IRL_{12} & IRL_{13} \\
IRL_{12} & IRL_{22} & IRL_{23} \\
IRL_{13} & IRL_{23} & IRL_{33}
\end{pmatrix}
\times
\begin{pmatrix}
TRL_1 \\
TRL_2 \\
TRL_3
\end{pmatrix}
\]

Composite SRL = \(\frac{1}{n}\left(\frac{SRL_1}{n} + \frac{SRL_2}{n} + \frac{SRL_3}{n}\right)\)

\[
= \frac{1}{n^2}\left(SRL_1 + SRL_2 + SRL_3\right)
\]

- These values serve as a decision-making tool as they provide a prioritization guide of the system's technologies and integrations and point out deficiencies in the maturation process.
“String” Analysis Incorporated

Complex systems often offer numerous options for conducting operations.

- Operational strings were created that identified the components required to utilize a single function of the system.
- Assessment of the SRL for each of these options allows for a better understanding of the maturity of each operating configuration.
- Understanding the true status of the system on an operational string level allows for the opportunity to field initial capability earlier and then add to it as other strings mature.
## Verification and Validation Activities

### IRL Criteria
- Created expanded list of IRL criteria for each readiness level
- Goal was to capture the key elements of the integration maturation process
- Presented to 30 integration SMEs from across government, academia, and industry
- Asked to assess importance of each criterion
- Results show solid buy-in among SMEs that identified criteria are key factors in successful integration

### SRL Evaluation Process
- Conducted a “blind trial” of SRL methodology and evaluation process
- User’s Guide and evaluation criteria were sent to key system SMEs
- From just these resources SMEs were asked to conduct the evaluation and report on the results
- Compiled results and iterated on lessons learned to improve the process
Trading Off Technology Options

Trade Between Advanced Capability or Increased Maturity
Taking Action to Mitigate Risk

System Maturity is Enhanced

LEGEND
- MP Technology
- Sea Frame System
- Current Mission Package SRL Status
- Previous Mission Package SRL Status
- Current Mission System SRL Status
- Technology Readiness Level
- Integration Maturity Level
- System Readiness Level Demarcation
- Scheduled Position

Risk to Cost and/or Schedule
- Low
- Medium
- High

Statement A: Approved for Public Release, Distribution is Unlimited
Effectively Channeling Resources

6 months later...

**Statement A:** Approved for Public Release, Distribution is Unlimited
Linking Cost to Maturity via Milestones

Total R&D Cost

![Graph showing the timeline and milestones for linking cost to maturity via milestones.]

- **MILESTONE A**
  - Jun 2006: Materiel Development Decision
  - Jan 2007: Initial Technical Review
  - Jul 2007: MILESTONE A
    - Alternative Systems Review
    - Systems Requirements Review
    - Mission Systems Testing
  - Nov 2007: MILESTONE B
    - System Functional Review
    - Technology Readiness Assessment 1
  - May 2008: Preliminary Design Review
  - Sep 2008: Critical Design Review

- **MILESTONE B**
  - Aug 2009: Test Readiness Review
  - Sep 2008: Mission Module Testing
  - Nov 2008: System Functional Review

- **MILESTONE C**
  - Apr 2010: Initial Operational Capability
    - System Verification Review
    - Functional Configuration Audit
    - Production Readiness Review
  - Jan 2011: Full Operational Capability
    - Physical Configuration Audit
    - Full Rate Production Decision Review
  - Apr 2012: In-Service Review

Current Mission Package SRL Status by View (Functional, Physical, Logical)

- **SRL assessment and test events / milestone gates are at or in advance of schedule**
- **SRL assessment is at or in advance of schedule, but test events / milestone gates remain to be closed**
- **SRL assessment and test events / milestone gates are behind schedule**
Lessons Learned

- Methodology is highly adaptable and can be quickly applied to a wide variety of development efforts
- Programs tend to minimize the importance of system and subsystem integration and thus overestimate the maturity of their development
- Widespread familiarity with TRL makes acceptance and utilization of TRL and IRL easier
- Formulating the system architecture early in development is a key step and leads to an enhancement of the overall systems engineering effort
- System architecture formulation also provides the opportunity to bring together SMEs from both the physical and logical realms and necessitates insightful discussions across the team
- The decision maker is afforded the ability to assess program status from a system of systems perspective

The SRL methodology delivers a holistic evaluation of complex system readiness that is robust, repeatable, and agile