An Innovative Strategy for System Sustainability

NDIA Systems Engineering Conference
30 October 2013
Arlington, VA
Outline

- Problem Statement
- Model-Based Knowledge Capture
- Observed Benefits
Enhanced Retention of System Knowledge Required

- **Large scale, legacy system sustainment context**
  - Systems experience can diminish in later lifecycle phases
  - Static infrastructure for information and processes

- **Modernization and upgrades often subsystem focused**
  - Subsystem changes yield risk of emergent systems behavior
  - Unable to accomplish systems-level re-optimization

- **System sustainability needs**
  - Prevent erosion of system level knowledge & capability
  - Understand impact of requirement/ design changes
  - Quickly obtain confidence of design’s ability to meet requirements
General Product Datatypes

SYSTEM DESIGN

- Impact Assessment
- Requirement Verification

SUBSYSTEM DESIGN

INTEGRATION & TEST

Design Verification
Traditional Data Sharing

- CONOPS Documents
- Requirement Documents
- ICDs

SYSTEM DESIGN
- Impact Assessment
- Requirement Verification

- Subsystem Designs
- DOORS Repositories

SUBSYSTEM DESIGN
- Models & Simulations

INTEGRATION & TEST
- HWIL / TACTICAL

VIRTUAL SIMULATIONS
- HWIL Simulations

VIRTUAL SIMULATIONS
- HWIL Simulations
Proposed Design

States: Sight

issuing Reset signal

GPS - ISA

Design Verification Requirement

Fully Integrated Data Sharing

Operational Use Cases

Acquire

mitigate Jammers

All Use Cases

Provide Navigation Data

Power Source

mitigate Jammers

provide Navigation Solution

Mitigate Jamming

«include»

Inspect Segments

Impact

Verifying

FAIL

Impact Assessment

Sighting Orientation (Mode 6)

Sighting Orientation (from Guidance System (Mode 6))

wait to terminate reset

«extend»

Mission Computer

provide navigation data

Power source

mitigate Jammers

TST

FLS

provide navigation data

Power source

mitigate Jammers

provide navigation solution

mitigate jelly marmalade

TST

FLS

provide navigation solution
Traditional Knowledge Capture

**Systems Knowledge**
- Federated Documents
  - Manual generation
- Manual traceability of requirements
- Reliance on domain experts to assess impacts

**Verification through Simulation**
- Now using auto-code generation
- Dynamic parameters setting
- Subsystem requirement verification
  - Often manually performed

---

Subject Matter Expert Experience

System Level Description

Subsystem Modeling & Simulation

HWIL
Integrated Knowledge Capture

**Systems Knowledge**
- Dynamically decompose to subsystem models
- Capture systems requirements, designs, tradeoffs
- Automatic specification generation

**Integration with Simulation**
- Shared data
- Automatic requirement verification
- Store results for future reference

**System Level Description**

**View of Descriptions**

**Subsystem Modeling & Simulation**

**HWIL**
Model provides a central location for access to data where updates are automatically distributed
Insure Consistency through Data Reuse

Capture the Data Once and Reuse as Needed

<table>
<thead>
<tr>
<th>Common Source of Information</th>
<th>System Decomposition</th>
<th>Artifact Generation</th>
<th>Implement Traceability</th>
<th>Subsystem Simulation</th>
<th>Requirement Verification</th>
</tr>
</thead>
</table>

Model-based approach enforce consistent relationships - prevent errors and miscommunication
Generating Requirements from Higher Level Design is a Standard Practice - Now Tools Can Do This for Us
Integration with Subsystem Definition

Link to Requirements or other Design Tools

- Common Source of Information
- System Decomposition
- Artifact Generation
- Implement Traceability
- Subsystem Simulation
- Requirement Verification

**Requirements**

**Subsystem Parameters**

MBSE Specification

3D CAD Implementation

Inheritance of Inalterable Parameters Insures that Consistent Values are Used in Design and Verification
Provide Early Design Confidence

Simulated flights provides confidence that the design meets requirements

- Common Source of Information
- System Decomposition
- Artifact Generation
- Implement Traceability
- Subsystem Simulation
- Requirement Verification

Closed-loop Simulation running externally-specified parameters
Simulations and Flight SW operate on the same data used to generate ICD
Automatic Assessment of Ability to Meet Requirements

Common Source of Information → System Decomposition → Artifact Generation → Implement Traceability → Subsystem Simulation → Requirement Verification

Post results back to repository for review

Requirements satisfaction status is automatically updated

Simple visualization allows teams to quickly identify and react to problems
Conclusions

- Systems model can greatly increase systems knowledge retention
  - A central data source greatly improves knowledge dissemination
  - Dynamically decomposable models improve data retrieval
  - Models enforce consistency and remove ambiguity
- Linked elements, models, and text facilitate change impact assessment
  - Automatic requirement generation reduces rework
  - Auto requirements saves time
  - Shared data reduces errors and redesign times
- Integrated simulations enable continuous verification
  - Provides design confidence earlier
  - Can reduce the extent of hardware testing
  - Automatic requirement verification reduces “overhead”
Brian London
The Charles Stark Draper Laboratory
Office: (617)258-3059
blondon@draper.com