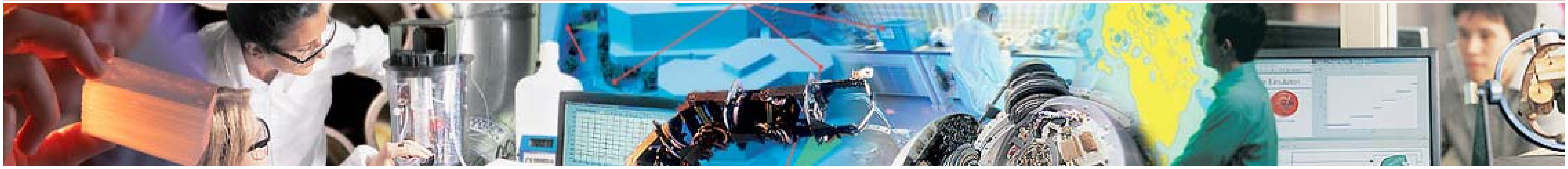


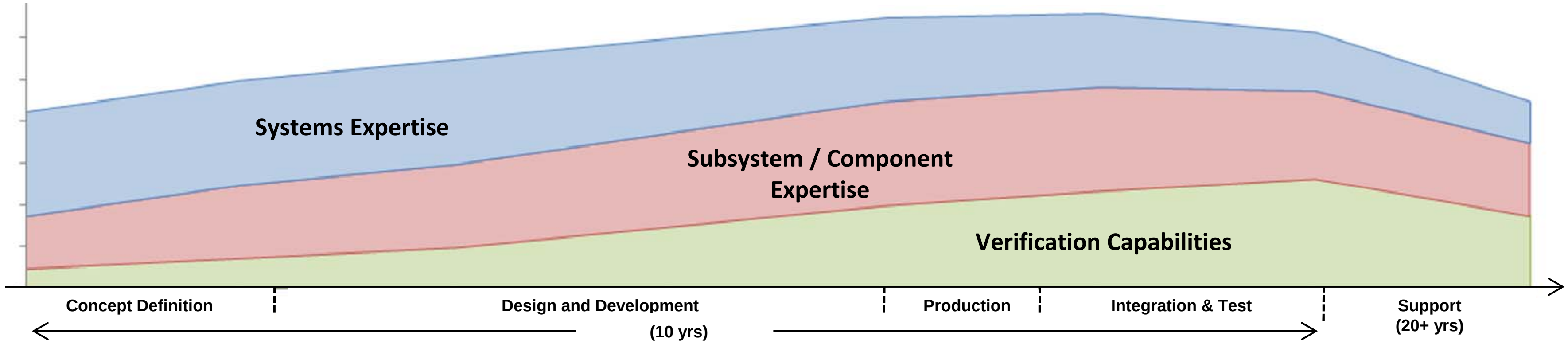
An Innovative Strategy for System Sustainability

NDIA Systems Engineering Conference
30 October 2013
Arlington, VA

Outline

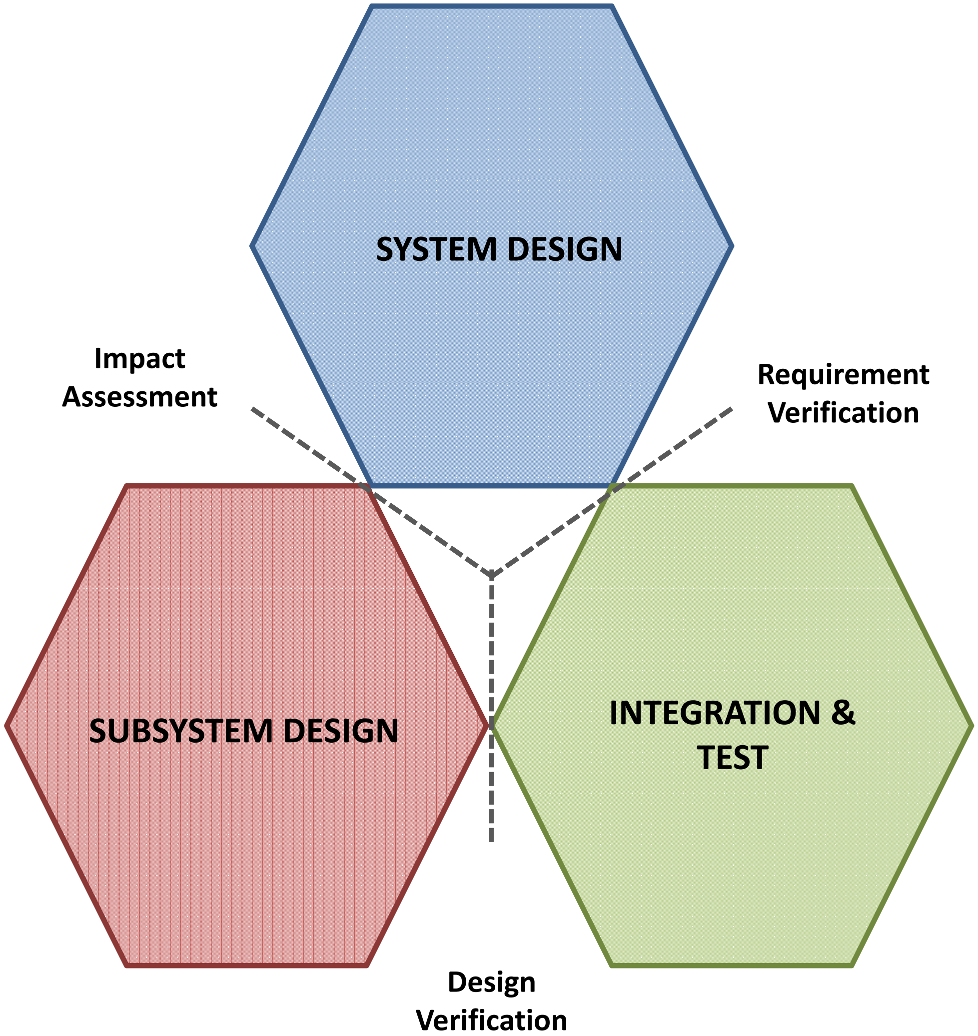
- Problem Statement
- Model-Based Knowledge Capture
- Observed Benefits

Enhanced Retention of System Knowledge Required

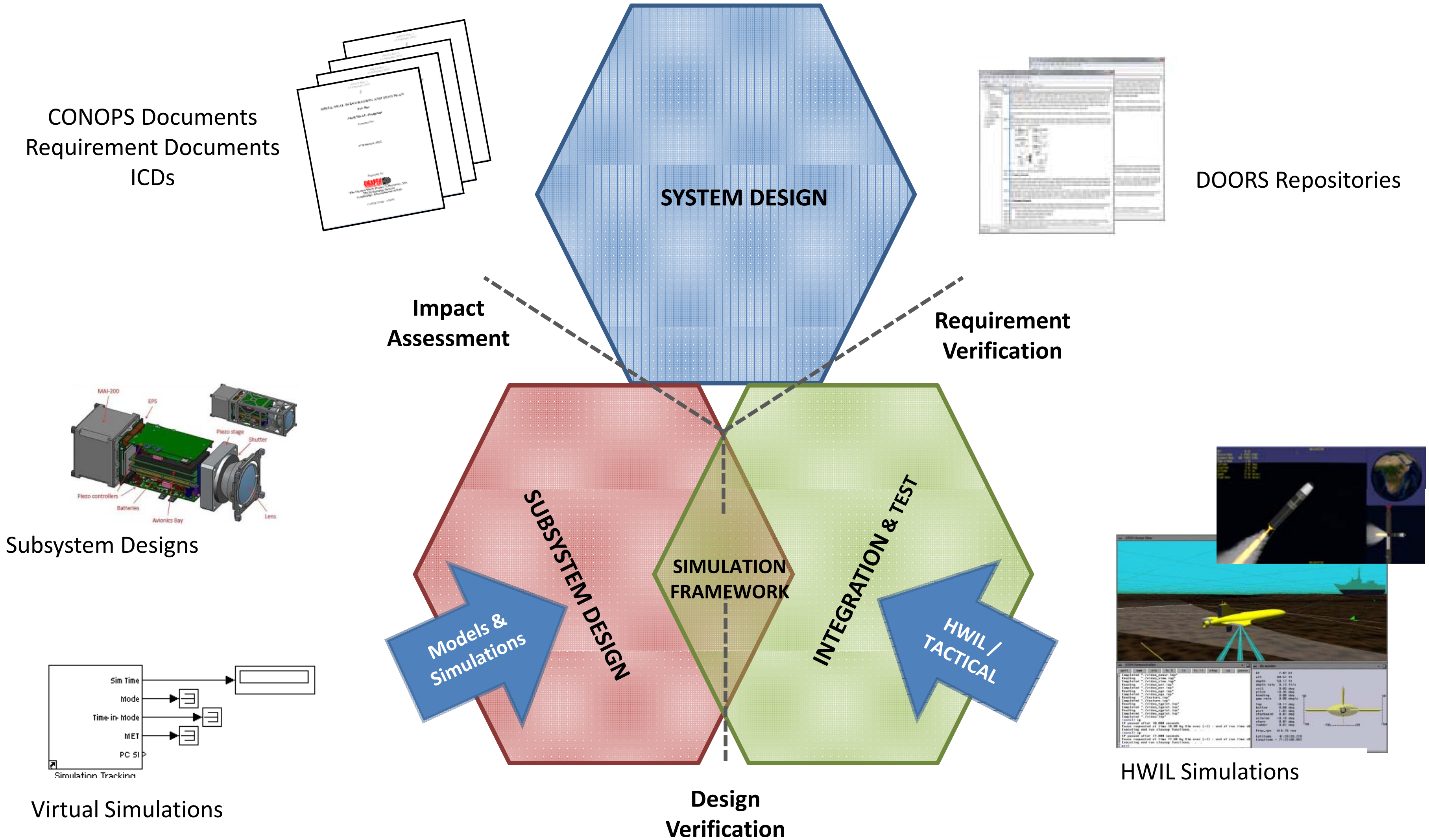


- **Large scale, legacy system sustainment context**
  - Systems experience can diminish in later lifecycle phases
  - Static infrastructure for information and processes
- **Modernization and upgrades often subsystem focused**
  - Subsystem changes yield risk of emergent systems behavior
  - Unable 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

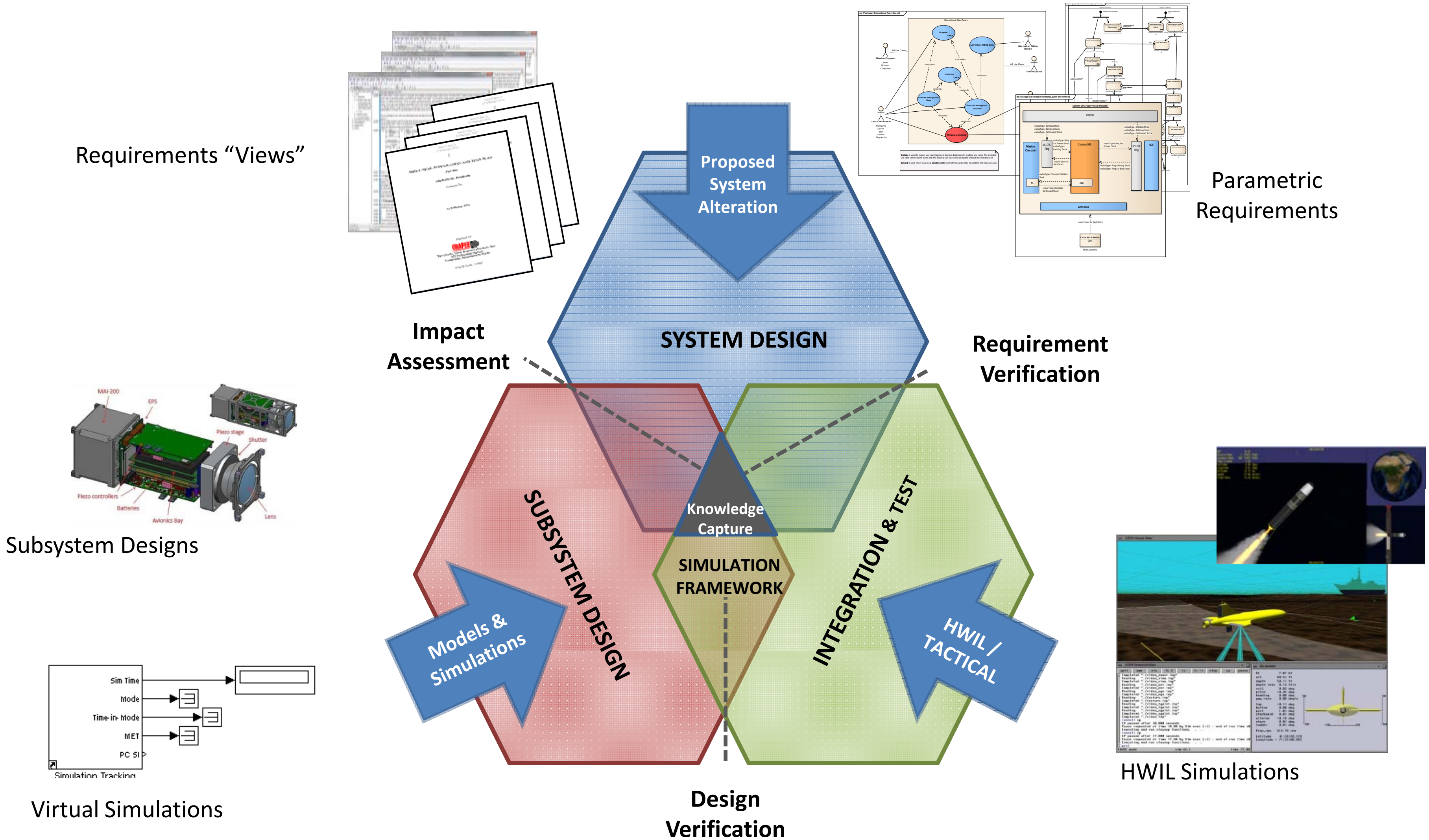
General Product Datatypes



Traditional Data Sharing



Fully Integrated Data Sharing



Traditional Knowledge Capture

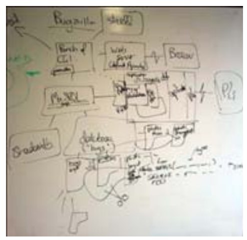
Systems Knowledge

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

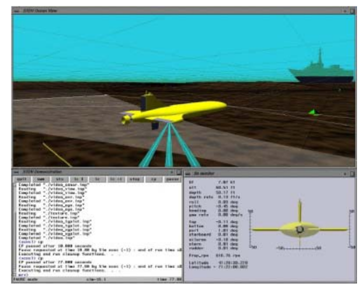
Verification through Simulation

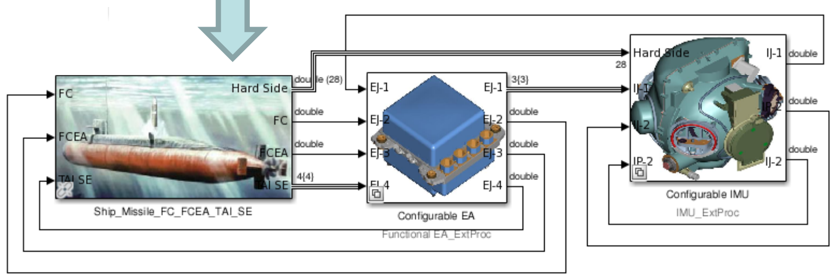
- Now using auto-code generation
- Dynamic parameters setting
- Subsystem requirement verification
  - Often manually performed

System Level Description



Subject Matter Expert Experience





Subsystem Modeling & Simulation

HWIL

DRAPER LABORATORY

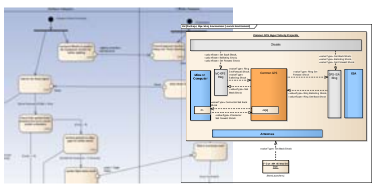
Integrated Knowledge Capture

Systems Knowledge

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

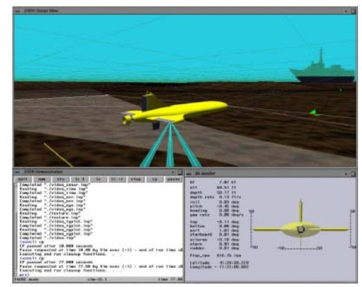
Integration with Simulation

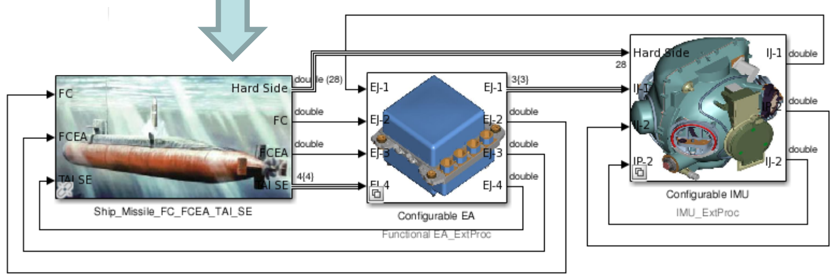
- Shared data
- Automatic requirement verification
- Store results for future reference



System Level Description

View of Descriptions



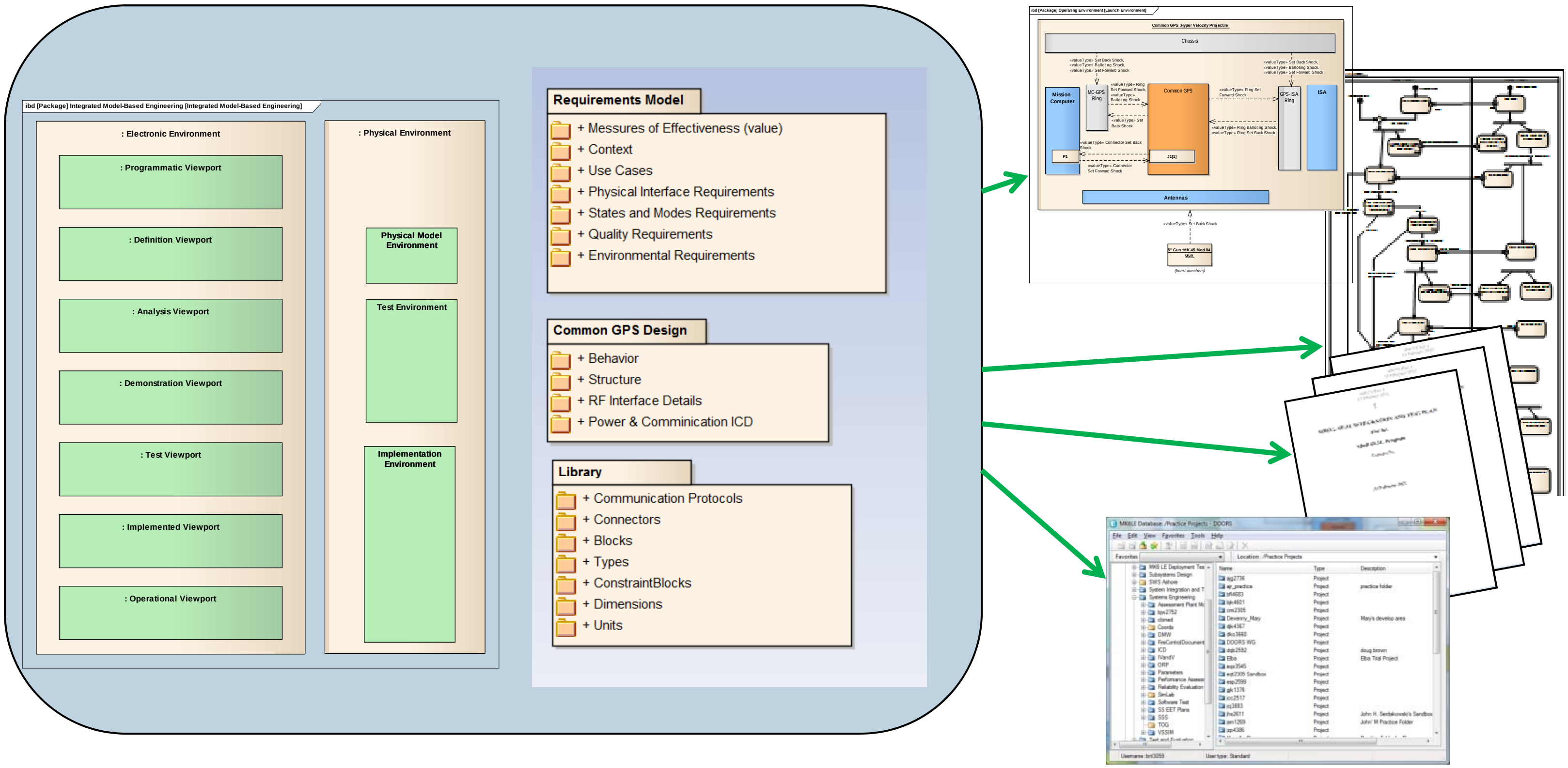


Subsystem Modeling & Simulation

HWIL

Getting the Right Information, Right Away

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

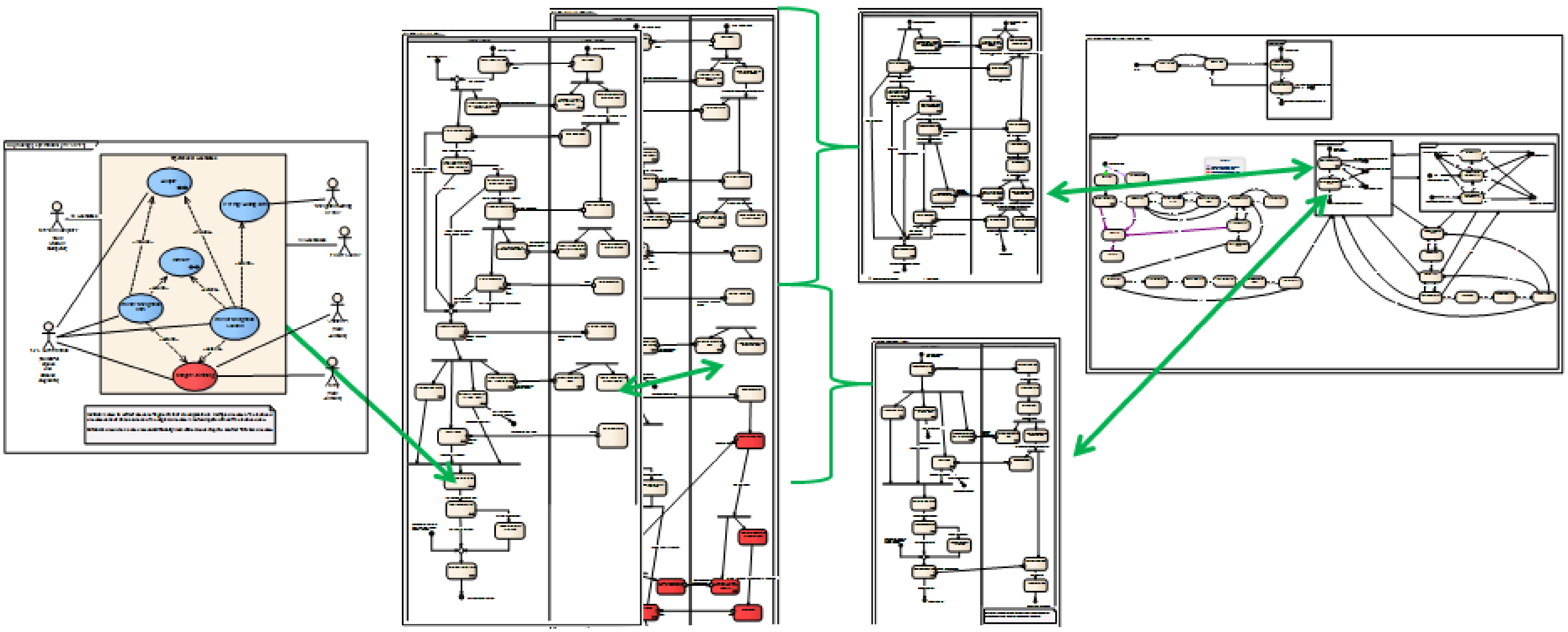


Model provides a central location for access to data where updates are automatically distributed

Insure Consistency through Data Reuse

Capture the Data Once and Reuse as Needed

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



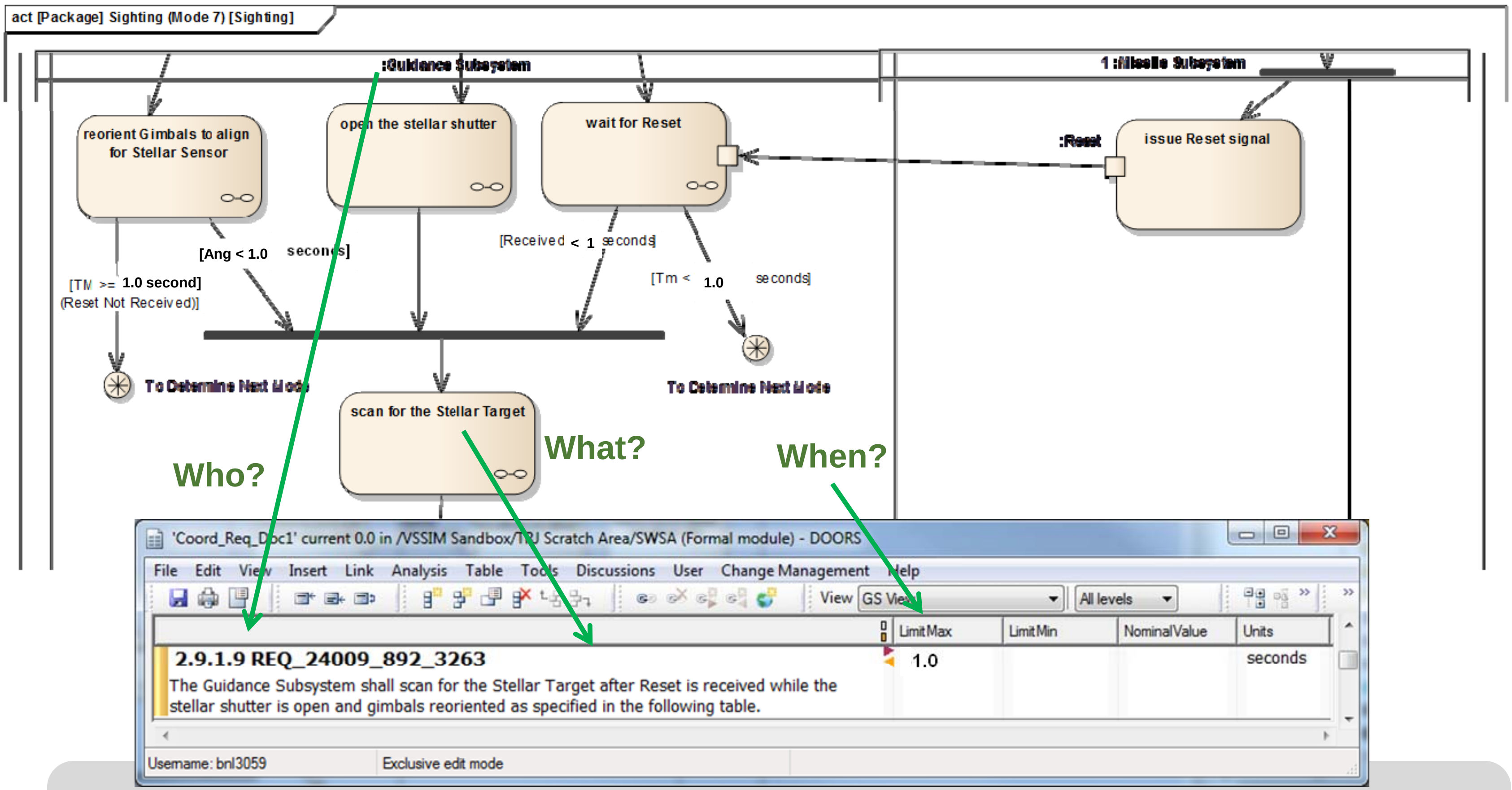
Model-based approach enforce consistent relationships - prevent errors and miscommunication

DRAPER LABORATORY

Automatic Requirement Generation

Reducing Manual Translation and Errors

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

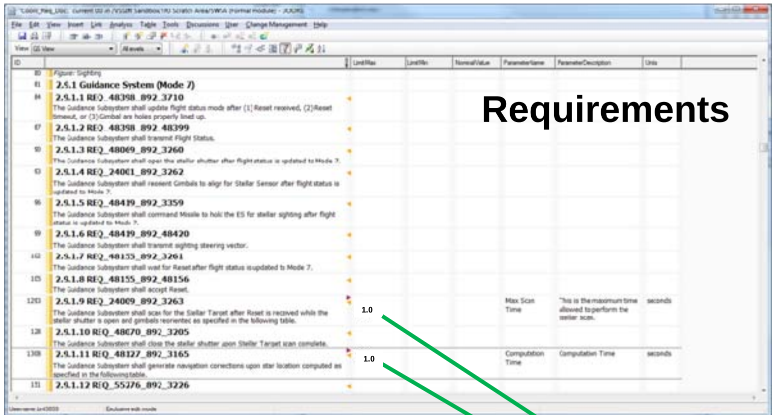


Generating Requirements from Higher Level Design is a Standard Practice - Now Tools Can Do This for Us

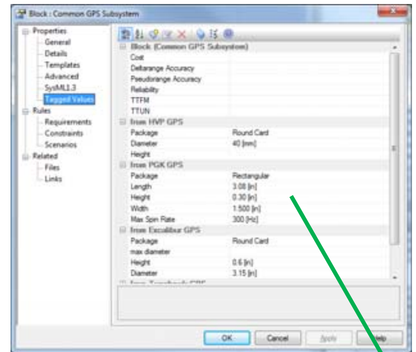
Integration with Subsystem Definition

Link to Requirements or other Design Tools

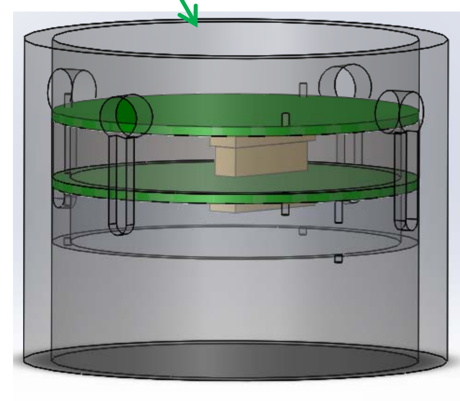
Common Source of Information → System Decomposition → Artifact Generation → Implement Traceability → Subsystem Simulation → Requirement Verification



Requirements

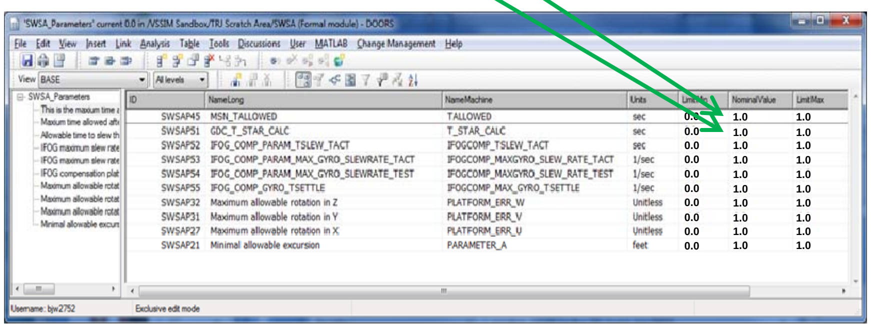


MBSE Specification



3D CAD Implementation

Subsystem Parameters



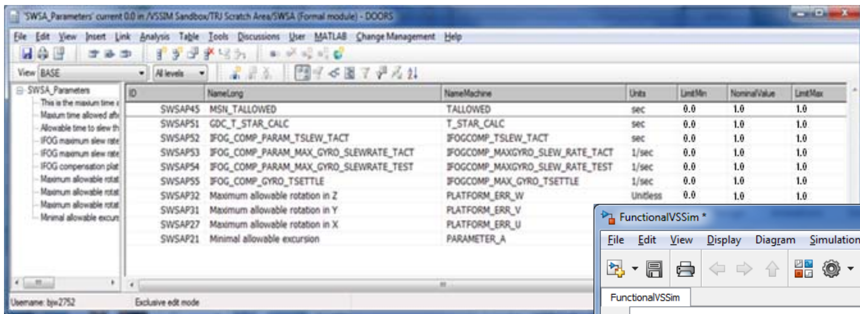
Inheritance of Inalterable Parameters Insures that Consistent Values are Used in Design and Verification

DRAPER LABORATORY

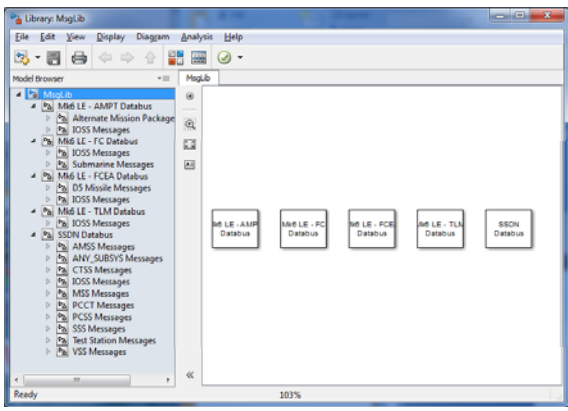
Provide Early Design Confidence

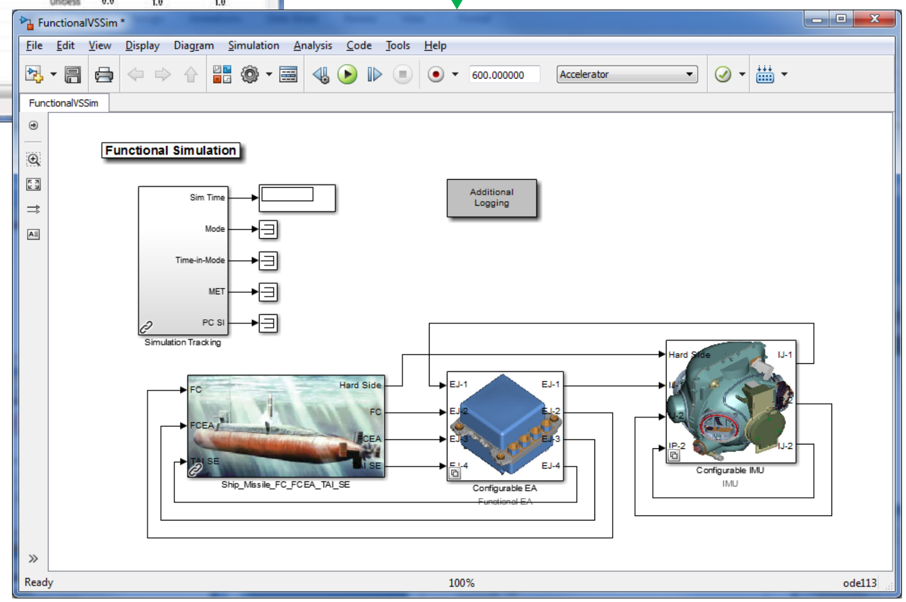
Simulated flights provides confidence that the design meets requirements

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



High Level Requirements and Implementation Parameters



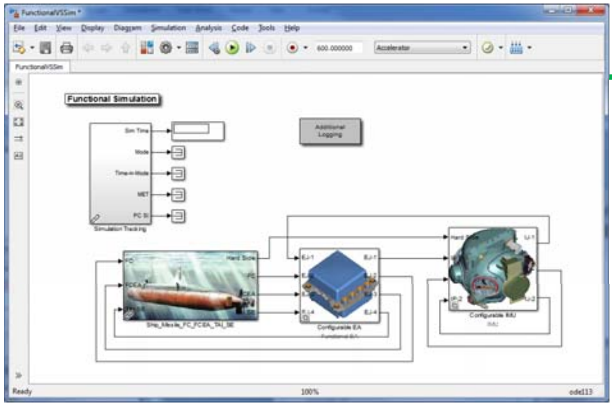


Closed-loop Simulation running externally-specified parameters

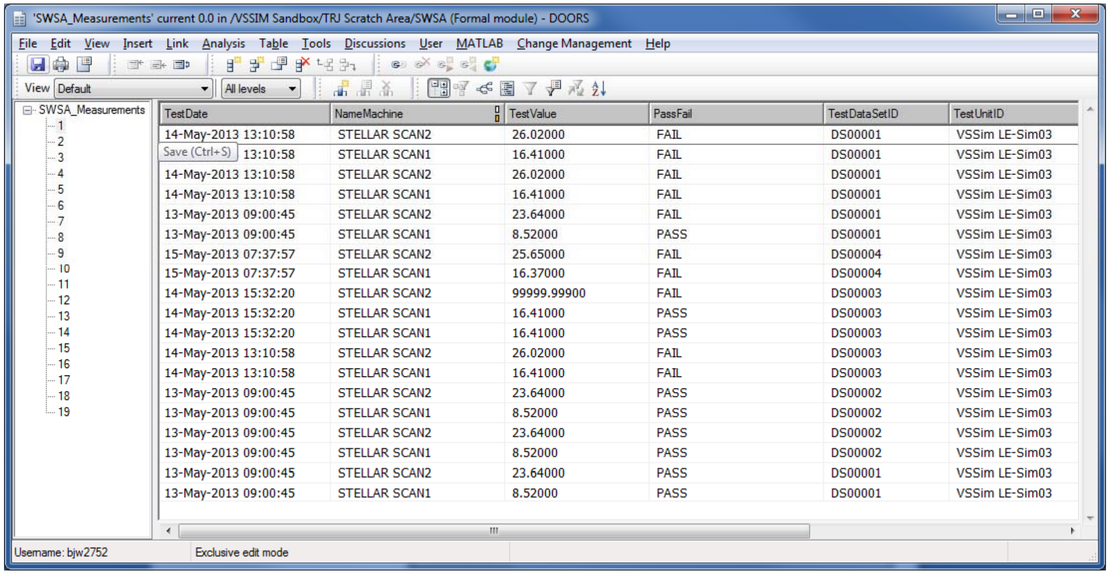
Simulations and Flight SW operate on the same data used to generate ICD

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



'SWSA_Measurements' current 0.0 in /VSSIM Sandbox/TRJ Scratch Area/SWSA (Formal module) - DOORS

File Edit View Insert Link Analysis Table Tools Discussions User MATLAB Change Management Help

View Default | All levels

SWSA_Measurements

	TestDate	NameMachine	TestValue	PassFail	TestDataSetID	TestUnitID
1	14-May-2013 13:10:58	STELLAR SCAN2	26.02000	FAIL	DS00001	VSSim LE-Sim03
2	Save (Ctrl+S) 13:10:58	STELLAR SCAN1	16.41000	FAIL	DS00001	VSSim LE-Sim03
3	14-May-2013 13:10:58	STELLAR SCAN2	26.02000	FAIL	DS00001	VSSim LE-Sim03
4	14-May-2013 13:10:58	STELLAR SCAN1	16.41000	FAIL	DS00001	VSSim LE-Sim03
5	13-May-2013 09:00:45	STELLAR SCAN2	23.64000	FAIL	DS00001	VSSim LE-Sim03
6	13-May-2013 09:00:45	STELLAR SCAN1	8.52000	PASS	DS00001	VSSim LE-Sim03
7	15-May-2013 07:37:57	STELLAR SCAN2	25.65000	FAIL	DS00004	VSSim LE-Sim03
8	15-May-2013 07:37:57	STELLAR SCAN1	16.37000	FAIL	DS00004	VSSim LE-Sim03
9	14-May-2013 15:32:20	STELLAR SCAN2	99999.99900	FAIL	DS00003	VSSim LE-Sim03
10	14-May-2013 15:32:20	STELLAR SCAN1	16.41000	PASS	DS00003	VSSim LE-Sim03
11	14-May-2013 15:32:20	STELLAR SCAN1	16.41000	PASS	DS00003	VSSim LE-Sim03
12	14-May-2013 13:10:58	STELLAR SCAN2	26.02000	FAIL	DS00003	VSSim LE-Sim03
13	14-May-2013 13:10:58	STELLAR SCAN1	16.41000	FAIL	DS00003	VSSim LE-Sim03
14	13-May-2013 09:00:45	STELLAR SCAN2	23.64000	PASS	DS00002	VSSim LE-Sim03
15	13-May-2013 09:00:45	STELLAR SCAN1	8.52000	PASS	DS00002	VSSim LE-Sim03
16	13-May-2013 09:00:45	STELLAR SCAN2	23.64000	PASS	DS00002	VSSim LE-Sim03
17	13-May-2013 09:00:45	STELLAR SCAN1	8.52000	PASS	DS00002	VSSim LE-Sim03
18	13-May-2013 09:00:45	STELLAR SCAN2	23.64000	PASS	DS00001	VSSim LE-Sim03
19	13-May-2013 09:00:45	STELLAR SCAN1	8.52000	PASS	DS00001	VSSim LE-Sim03

Username: bjw2752 Exclusive edit mode

Requirements satisfaction status is automatically updated

Simple visualization allows teams to quickly identify and react to problems

DRAPER LABORATORY

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”

Questions?

Brian London

The Charles Stark Draper Laboratory

Office: (617)258-3059

blondon@draper.com