Model-Based System Development for Managing the Evolution of a Common Submarine Combat System

18 May 2010

Steven W. Mitchell
Lockheed Martin Fellow
LM Certified Systems Architect

Agenda

- Complex Product Families (CPF) with High Variability
- Systems Engineering for CPF: the Traditional Approach
- Model-Based Systems Engineering for the SWFTS Product Family
- Anticipated Benefits
- Future Research

Product Families

Product Families

- Definition: A Product Family is a group of products derived from a common product platform.
  - Chrysler K-cars, Boeing 747
- Product Families are commonly developed across Lockheed Martin to tailor base product platforms to varying applications
  - Atlas, F-35, Aegis
- Product Families also emerge through technology evolution
  - F-16, AN/UYQ-70

Product Families Enable Effective Reuse of Engineering Investment

Complex Product Families (CPF) Combine

- Major Versions for Different Applications
  - Aegis for Cruisers, Destroyers, Frigates, Corvettes
  - F-35 Versions for Conventional Take-Off, STOVL, Aircraft Carrier
- Variants within Versions
  - Atlas V 401, 402, …, 552, 5H1, 5H2
- Multiple Generations/Baselines
  - Technical Refresh, Capability Enhancements
- Export Considerations

Product Families May Have Many Dimensions of Complexity

Traditional Systems Engineering Approach to Product Families

Traditional Systems Engineering Approach

1. Document Initial Product Platform
2. Branch Documentation Package for Each Subsequent Family Member
3. Manually Maintain Correlation Between Document Tree Forks – Or Not
4. Manually Trace Bugs, Obsolete Components, etc. Between Family Members – Or Not

Maintaining Complex Product Families Is Labor Intensive and Error Prone

Submarine Warfare Federated Tactical System (SWFTS)

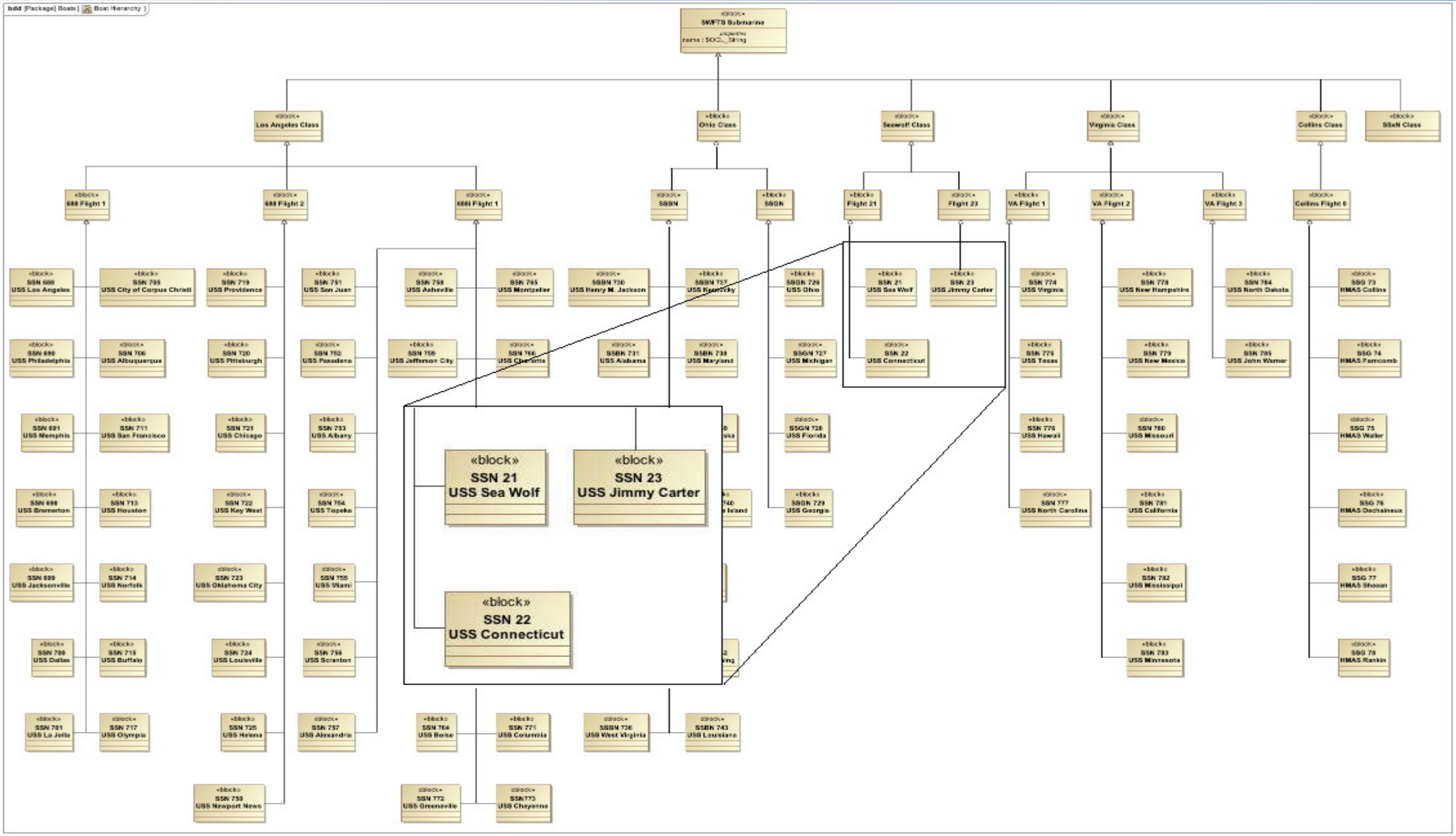
Product Family Example: SWFTS

Submarine Warfare Federated Tactical System

- A Common Combat System Deployed Across Multiple Fleets
  - USN: Los Angeles (SSN 688 and SSN 688i), Ohio (SSGN 726), Seawolf (SSN 21), Virginia (SSN 774), Ohio Replacement (SSBN)
  - RAN: Collins (SSG 73)
- Federates Multiple Subsystems from Multiple Program Offices and Vendors
  - Sonar, ESM, Imaging, Tactical Control, Weapons Control, Communications, etc.

SWFTS Reduces Maintenance and Training Costs while Maximizing Capabilities

SWFTS Target Platforms

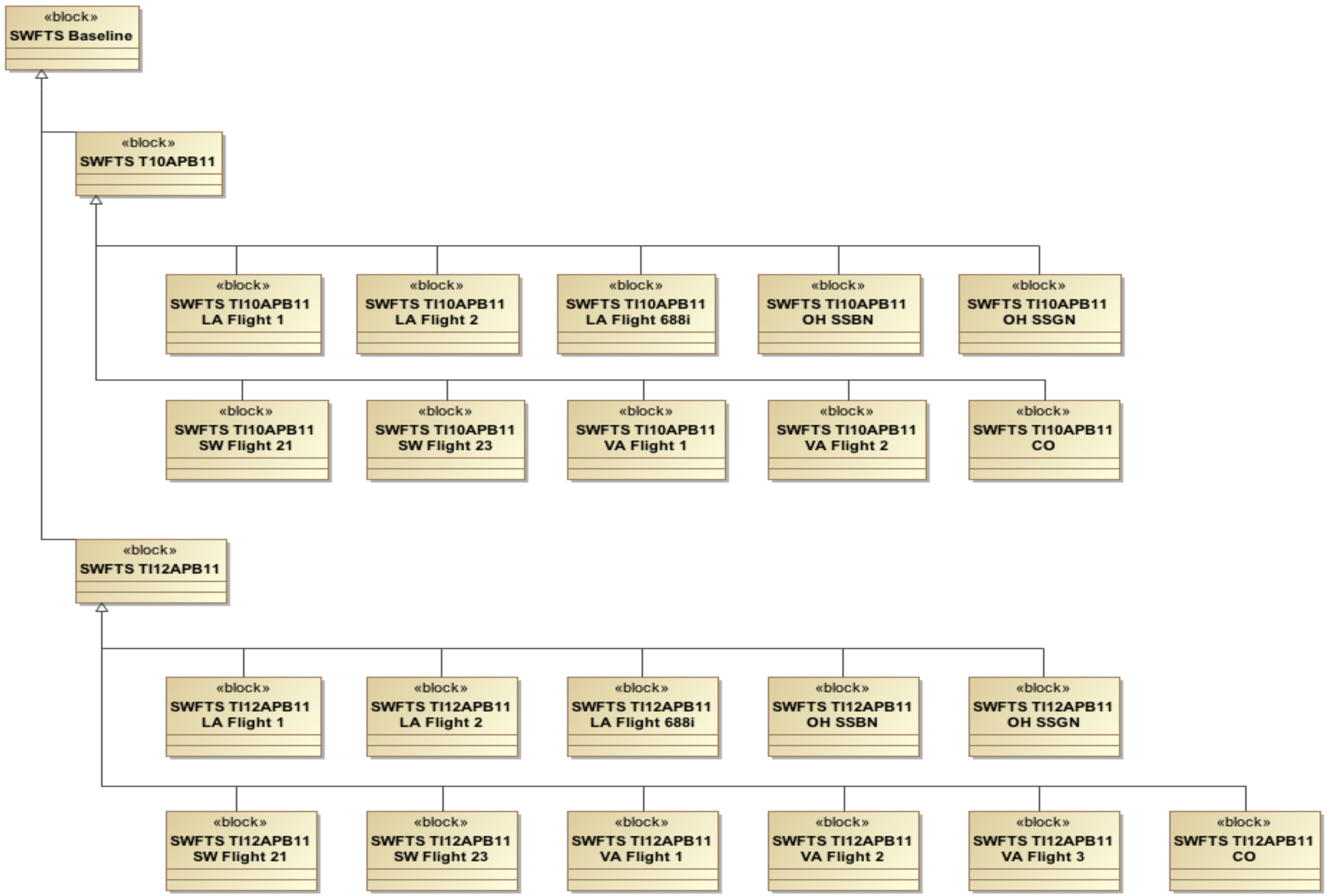


High Variability Between Product Family Members

- Each Class May Have Multiple Flights
  - Los Angeles Flights 1, 2, Improved
  - Ohio SSBN (Trident I, Trident II) and SSGN
  - Virginia Flights 1, 2, 3, …
- Within Flights, Ships May Vary Significantly
  - SSN 23 is 100 feet longer than sister ships
  - TEMPALTs and SHIPALTs for New Sensors, Weapons, etc for At-Sea Testing or Special Operations are Common
- Multiple Deployed and/or Managed Baselines within Flight

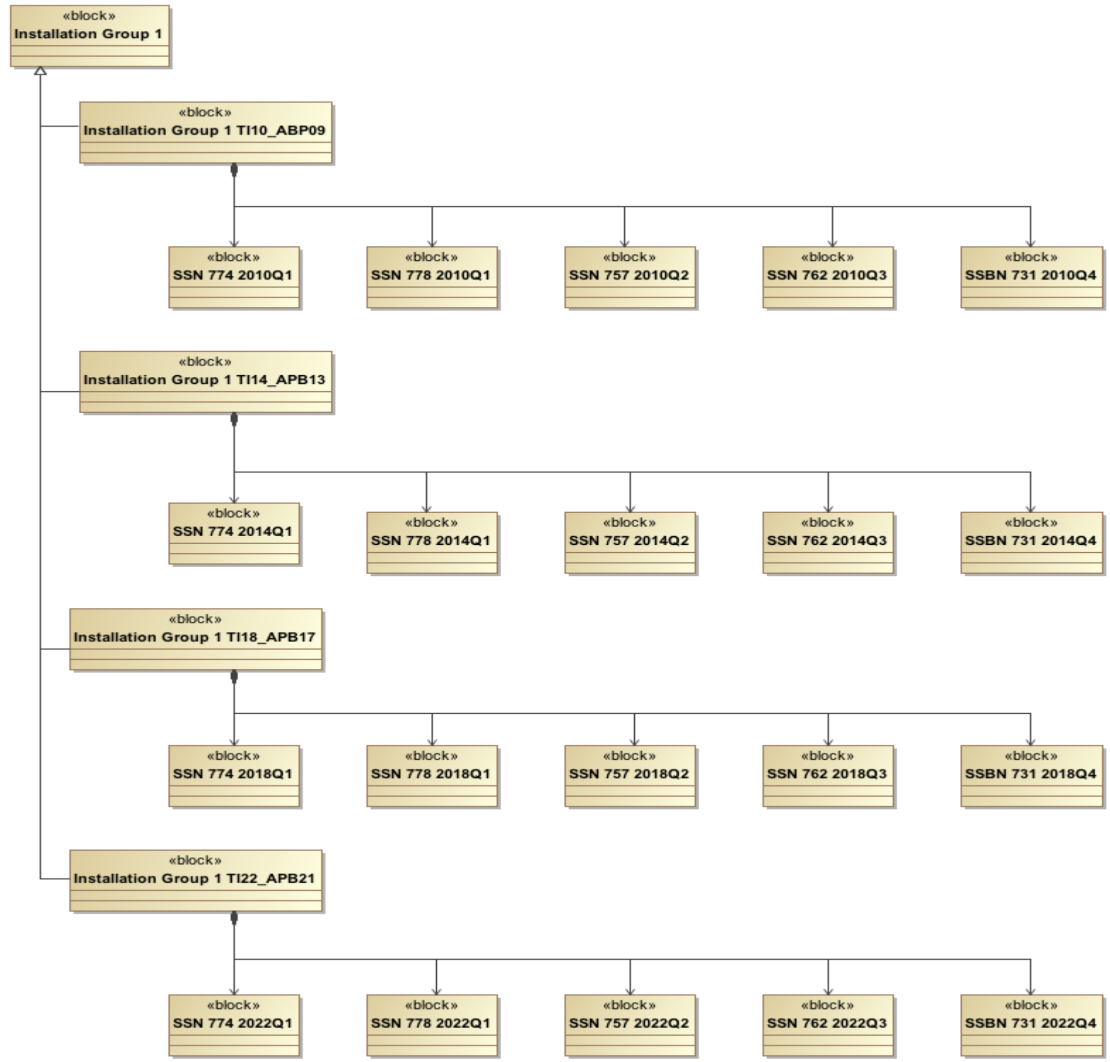
Over 21 Dimensions of Variation Must Be Managed in SWFTS

Baseline Variation within Flights



Managing Multiple Baselines Per Flight

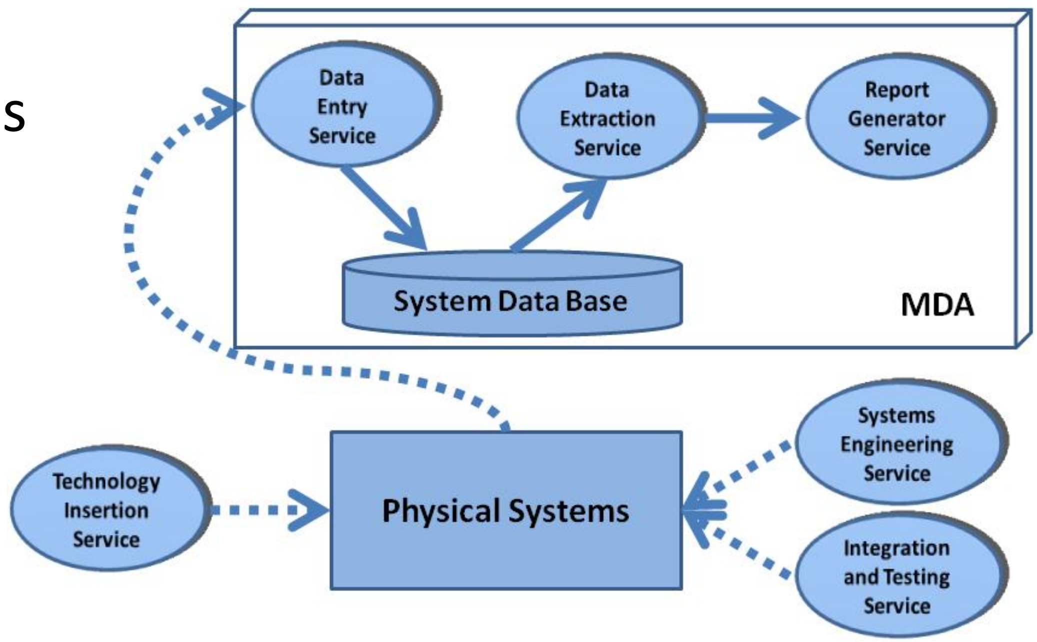
- Support of Deployed Baseline
- Rolling Installation of Next Baseline
- Development of Upcoming Baseline
- Planning for Future Baseline Evolution



Model Based Systems Engineering for SWFTS

First Step: Re-Engineer SWFTS SE Process

- Apply the MBSE Process to the SWFTS Program
  - Model the Existing Process Well Enough to Identify Duplicative Activities and Redundant Information Repositories
  - Define New Streamlined Processes Around Normalized Repository

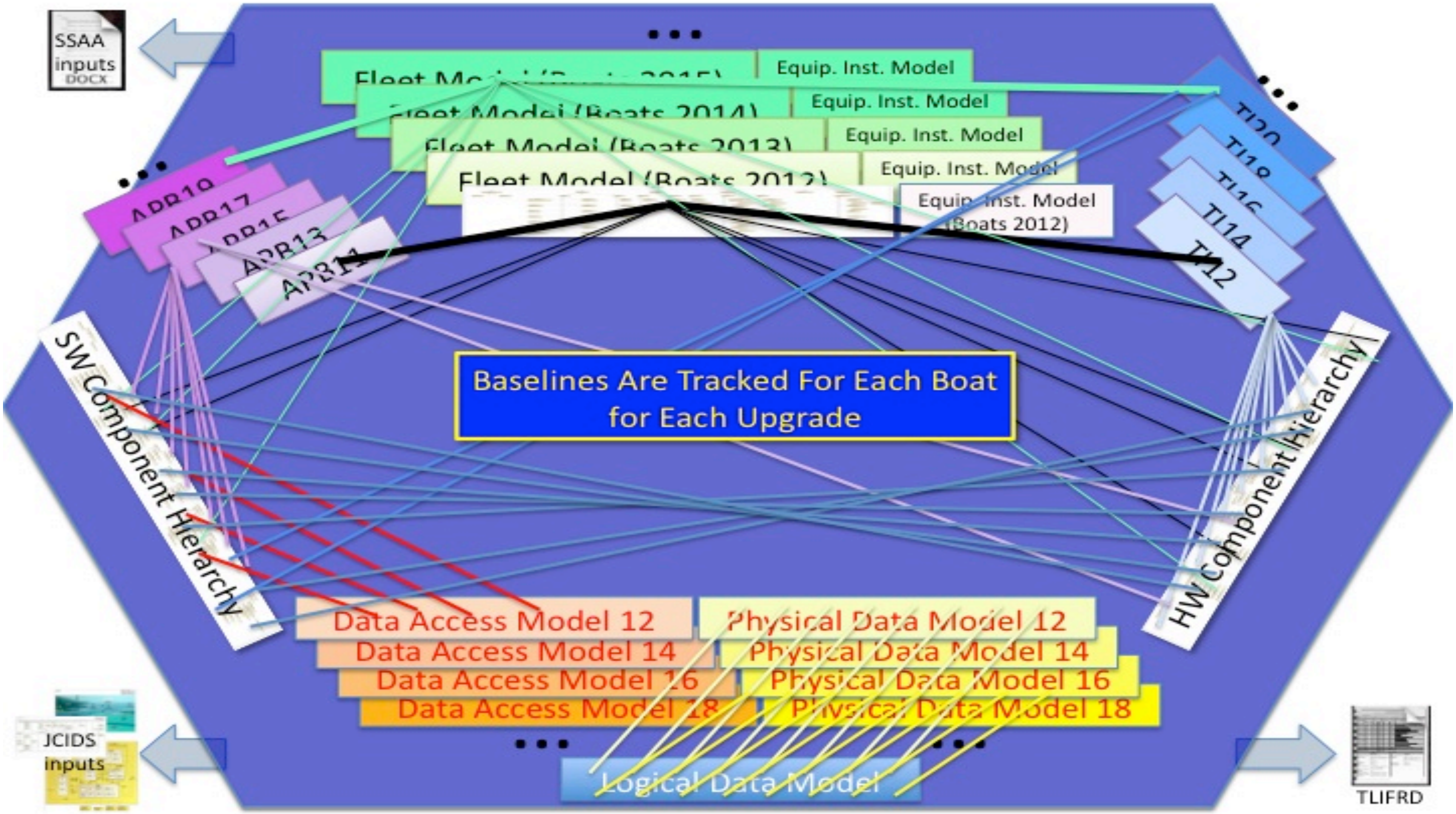


Conceptual Model of SWFTS SE Process

Submarine C5I Engineering Data Model

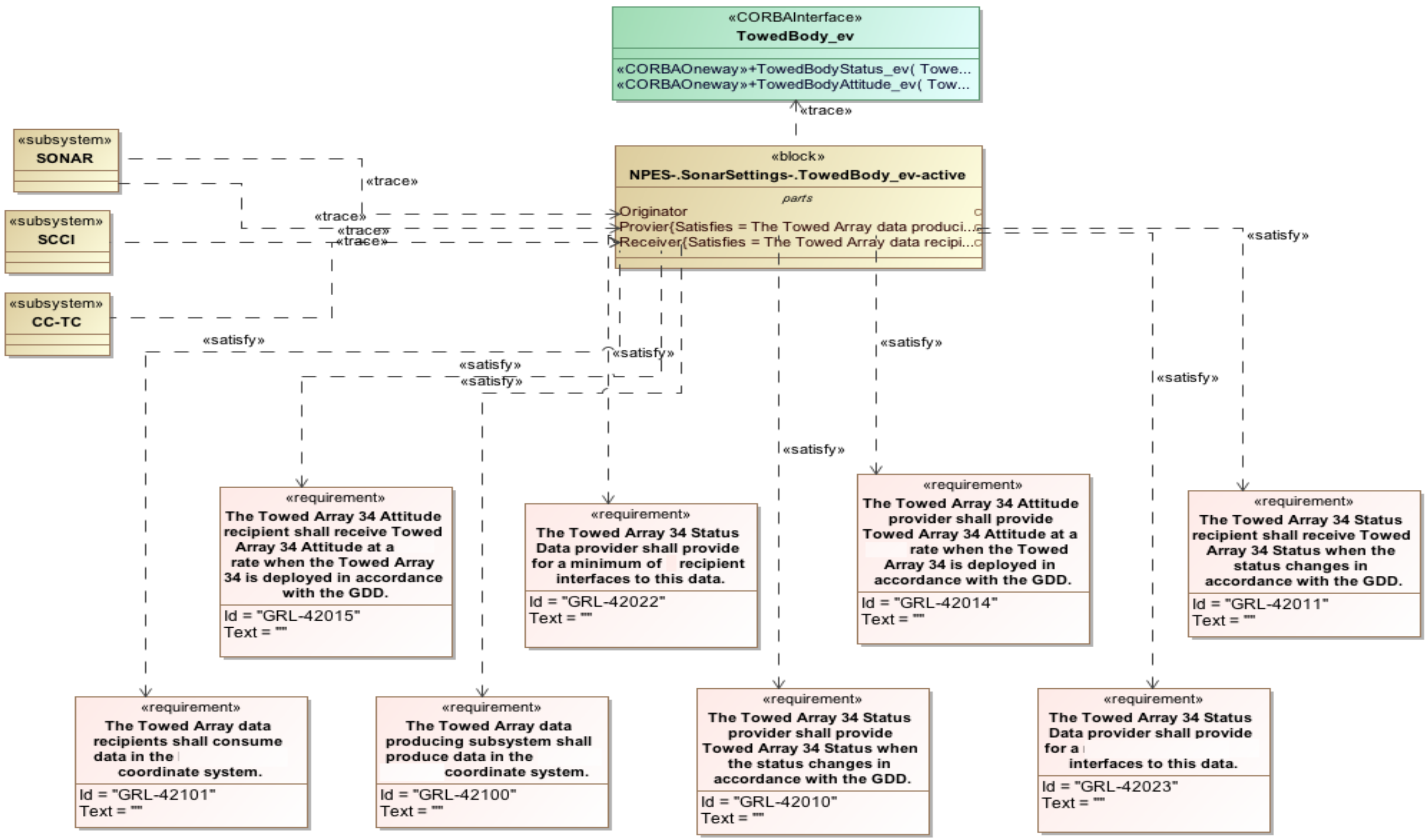
- Common Submarine Combat System is Modeled in UML & SysML
  - XML Representation Stored in Relational Database
- Supplemental Information Also Stored in Database
- Web Services Built Around Model Repository
  - Enable Engineers in Various IPTs to Access Models Through Task-Specific Interfaces (Spreadsheets, Database Queries, etc)
  - Produce Required Systems Engineering Artifacts as Reports from Model Repository

Many Informational Dimensions of Complexity



LOCKHEED MARTIN ENGINEERING

Requirements Tractability in SWFTS Model



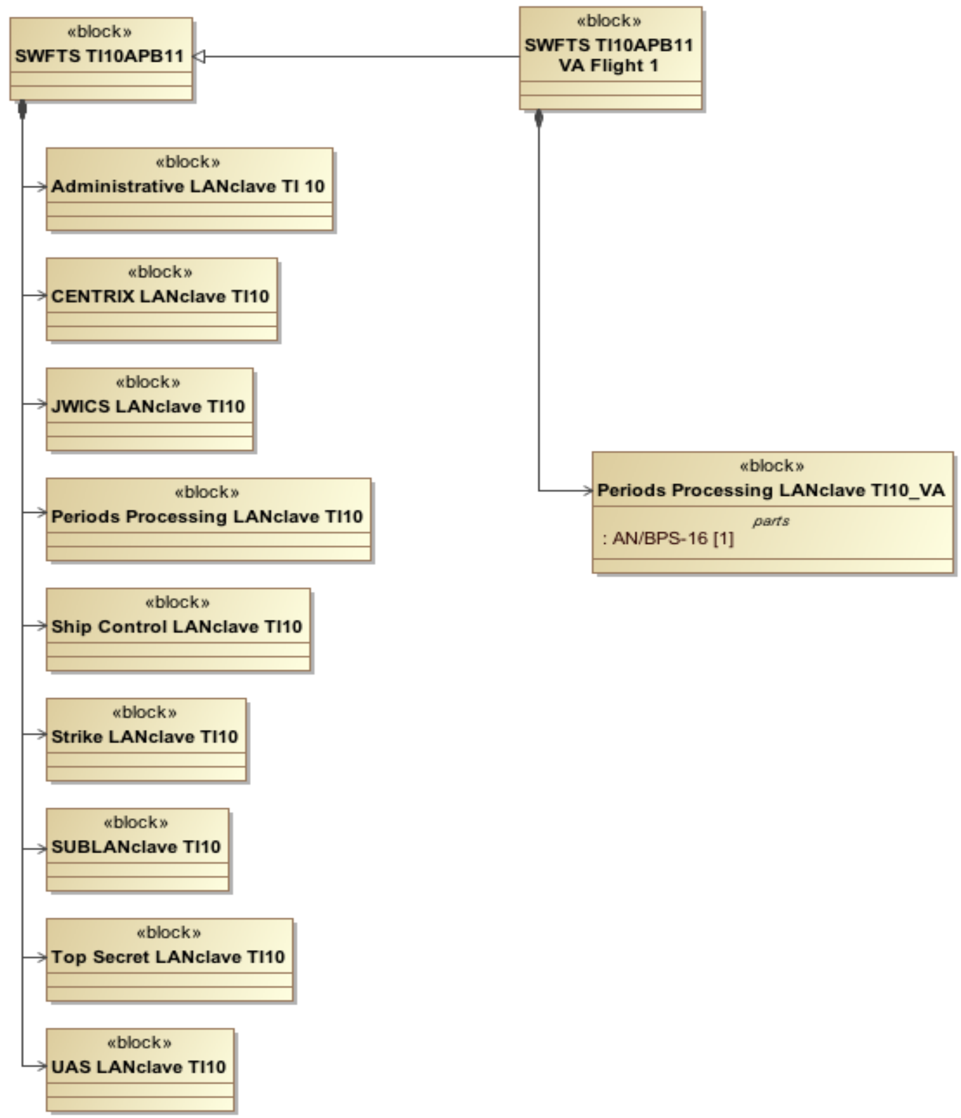
LOCKHEED MARTIN
ENGINEERING

Efficiently Managing Variations in SWFTS

- Ship-to-Ship and Baseline-to-Baseline Variations are Captured by Inheritance
  - Only the Variations are Added to the Model, Eliminating Duplication
  - Maintains Traceability of Requirements
  - Maximizing Reuse
  - Minimizes Manual Reconciliation and Configuration Management Across Information Repositories

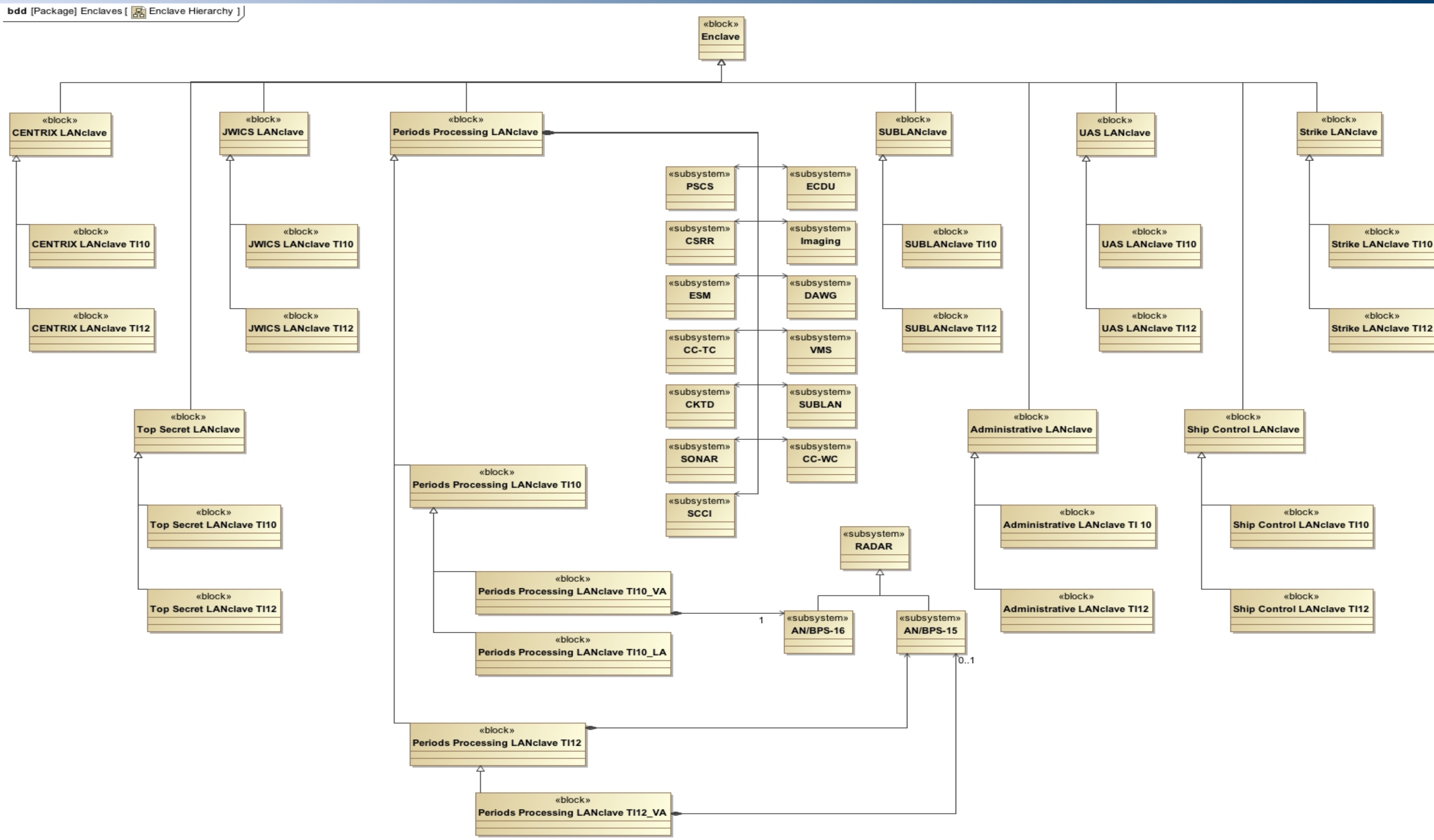
Class-to-Class Variations

Inheritance Efficiently Captures Class-to Class Variations



LOCKHEED MARTIN
ENGINEERING

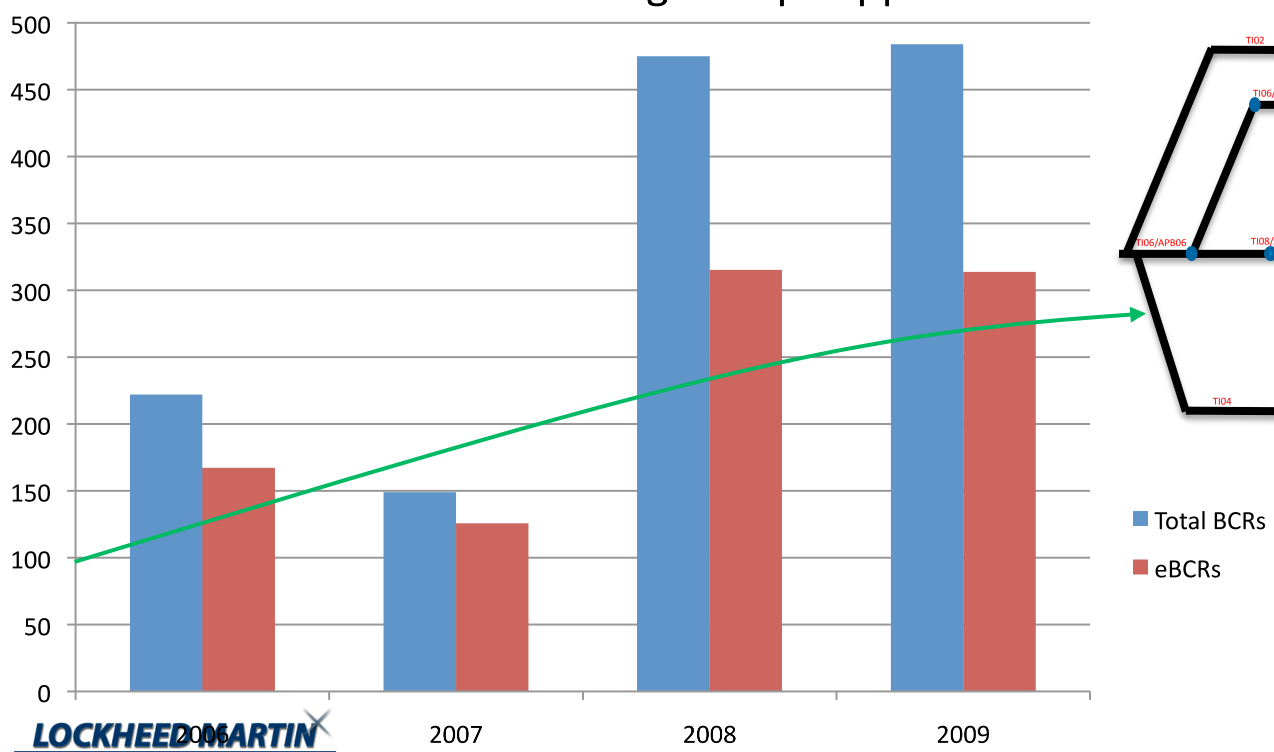
Variations in Applications of Subsystems

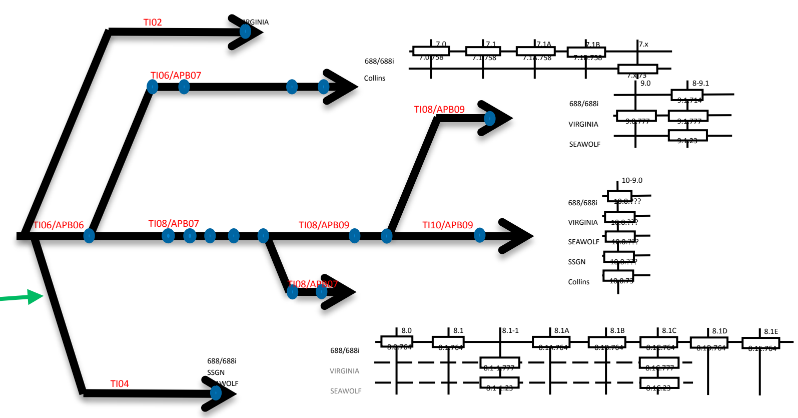


Anticipated Benefits of MBSE for SWFTS

Need for Change

- SWFTS Systems Engineering efforts have increased steadily in the past 3 years and Systems Engineering resources are stretched
  - Increasing number of Baseline Change Requests (BCR) processed per year
  - More concurrent baselines (baselines not retired) and more subsystems being added
  - Expanding System of Systems role, and Principle Systems Engineer IPT and Architecture Working Group support





eBCR = Equivalent BCR
(deprecates replications)

Decreasing Funding

- At the same time, the funding required to perform the Systems Engineering to define the interfaces has gone down

CY	Total SE FTE	TI FTE	SE Core FTE
2007	28.7	0	28.7
2008	28	2	26
2009	26.2	4	22.2
2010	24.8	3	21.8

Systems Engineering Core Trend

- The efficiency to perform the SWFTS System Engineering has been improved through continuous process improvements, but we're reaching the limits.
  - DOORS automated Requirements Baseline Management, but Interfaces (IP addressing, IDL, etc.) are still managed manually
  - MBSE affects larger scope
    - SWFTS (Capabilities Definition & Baseline Management), JCIDS/ISP, and Subsystems
    - implies larger potential savings
- The future growth in SE will drive the need for more SWFTS SE funding
  - In order to meet the continually increasing needs of the fleet without increased funding, the SWFTS SE process must become more efficient

MBSE Will Enable a More Efficient System Engineering Process

ROI for SWFTS

- Like DOORS, MBSE is necessary to keep up
  - Ability to do more with fewer resources
  - Answer SE & architecture questions
  - Enable faster response times
- MBSE provides flexibility and scalability
  - To see effects at platform level
  - To support add'l classes (e.g. OCRP)
  - To adopt new interface types (CORBA replacement)
  - To automate generation of interface code
- Expect MBSE savings
  - More automation like DOORS; includes IDL generation
  - Data integrity and concordance reduces QA effort
  - MBSE affects much more than is measurable, e.g. ability to answer what-if questions that take considerable time today

ROI for SE&I

- Expect 13% additional savings to SE from MBSE
  - 25% in Capability Definition
  - Another 10% over DOORS in Baseline Management
- Savings won't be seen until 4th year
  - 2 years to implement model
  - 1 year transition overlap with current process

SE Phase/steps	Current FTE/ yr	Future FTE/ yr	
SE Management (Risk, IMS, Metrics)	1.0	1.0	0%
Capability definition	8.4	6.3	-25%
Technology Insertion	4.0	4.0	0%
Baseline Development	12.8	11.5	-10%
TOTAL FTE	26.2	22.8	

13%

Future Research

MBSE of SWFTS Product Family

- Extend Current SWFTS MBSE Prototype to Full Information Model
  - May Discover Additional Limitations in UML and SysML Standards
  - Work With Standards Bodies as Needed
- Develop Tool Ecology Around SWFTS Model to Support Process Re-Engineering
  - Validate in Parallel with Current Process Using TI12/APB11
- Extend MBSE
  - Assess Interactions of Proposed BCRs
  - Directly Support Selected Subsystem Providers
  - Interface with Shipyard Models to Enable Integrated Ship Performance Impact Assessments
  - Support Full Life Cycle