LOCKHEED MARTIN
ENGINEERING

Efficiently Managing Product Baseline Configurations in the Model-Based System Development of a Combat System Product Family

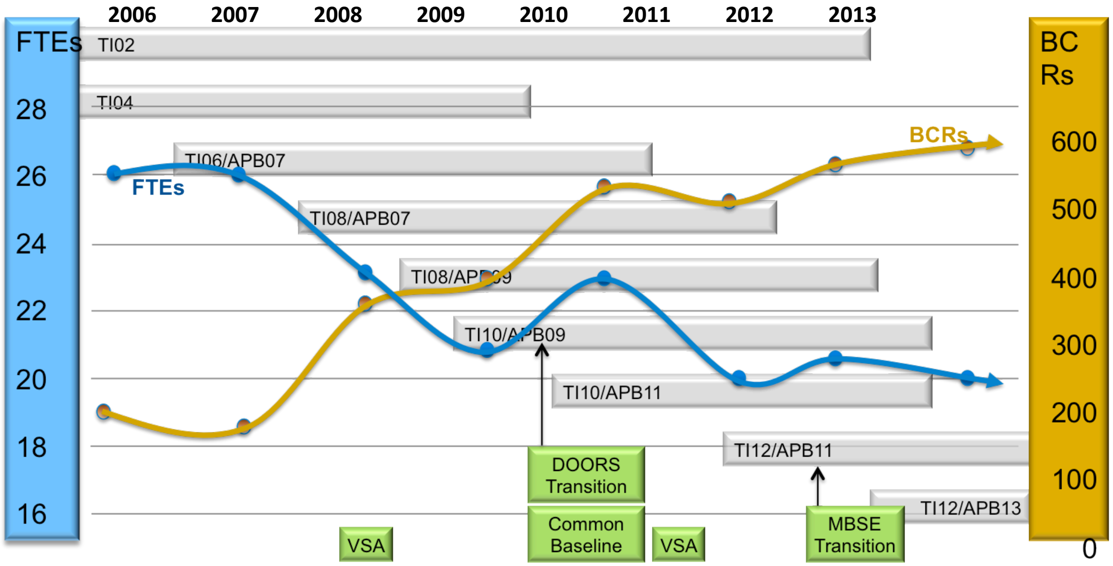
INCOSE IS Roma

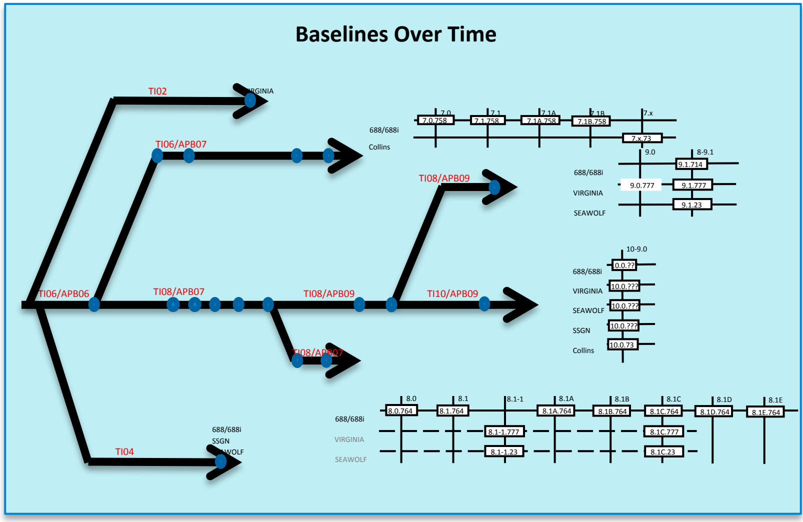
10 July 2012

Steven W. Mitchell
Lockheed Martin Master Architect

Submarine Warfare Federated Tactical System

- A Common Combat System Deployed Across Multiple Fleets
  - USN: Los Angeles (SSN 688), 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 Manages Subsystem Interfaces and System I&T





Continuous improvements in process and tools have reduced the cost of change and of SWFTS System of Systems Engineering

Model Based Systems Engineering in SWFTS

What SWFTS-MBSE is and is not

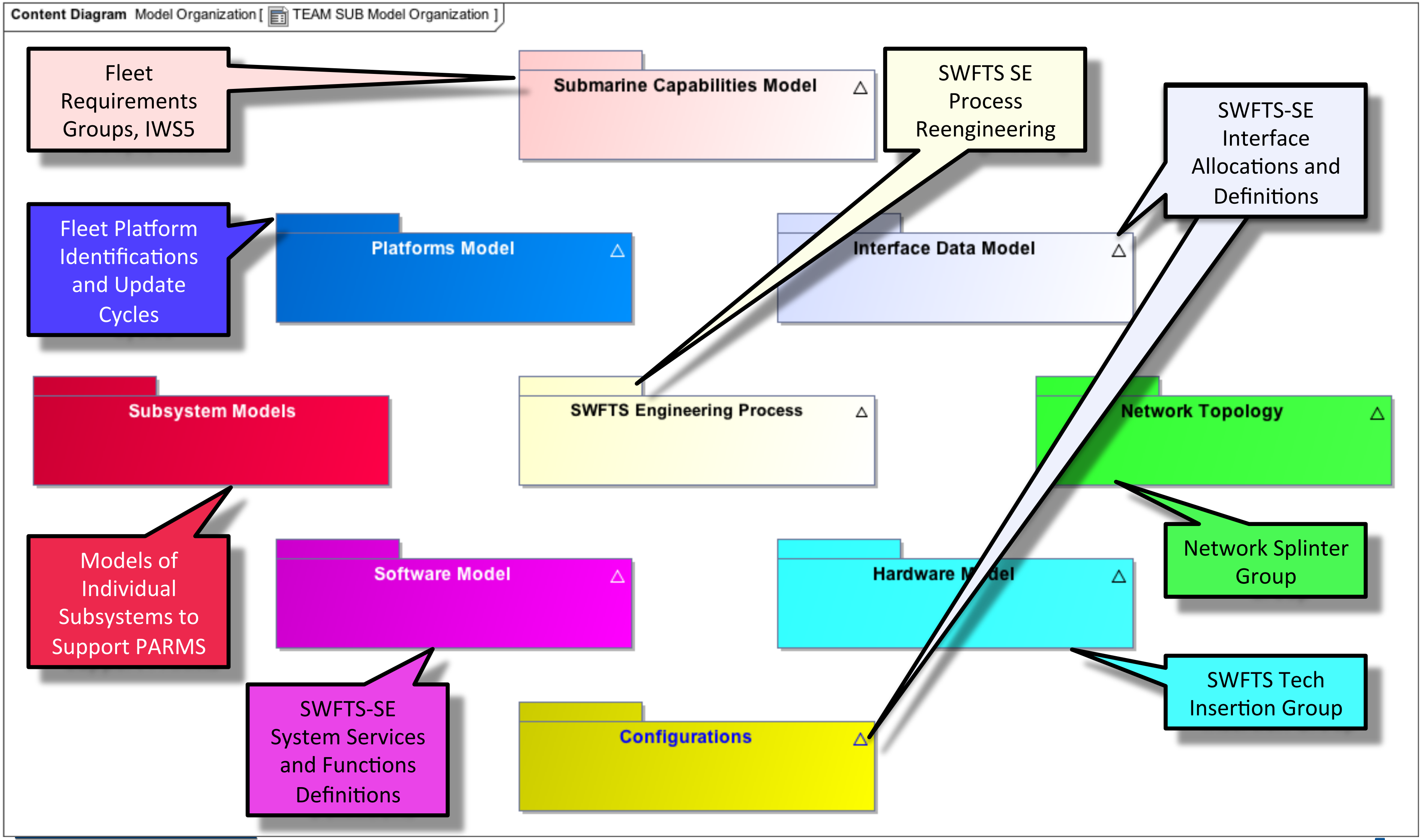
What SWFTS-MBSE is:

- **Federated Integration Tool**
  - Documentation management
  - Task Automation
    - MBSE reduces effort for SWFTS SE tasks
  - History, Decisions, Tribal Knowledge, Tribal Belief, etc. captured
- **Documentation Generation**
  - Reuse among communities
- **Communication**
  - Allows a Federated community to communicate more effectively

What SWFTS-MBSE is not:

- **Simulation**
  - Will not simulate parts of the system
- **Code Generation or Code Repository**
  - Will not generate or store code except subsystem interfaces (IDL/GPB)
- **Artificial Intelligence**
  - Will not replace Good Engineering
    - Will not find Engineering errors/issues
    - will find inconsistencies in documentation making good engineering easier

Physical Model Organization



Content Diagram Model Organization [TEAM SUB Model Organization]

Fleet Requirements Groups, IWS5

Submarine Capabilities Model

SWFTS SE Process Reengineering

SWFTS-SE Interface Allocations and Definitions

Fleet Platform Identifications and Update Cycles

Platforms Model

Interface Data Model

Subsystem Models

SWFTS Engineering Process

Network Topology

Models of Individual Subsystems to Support PARMS

Software Model

Hardware M del

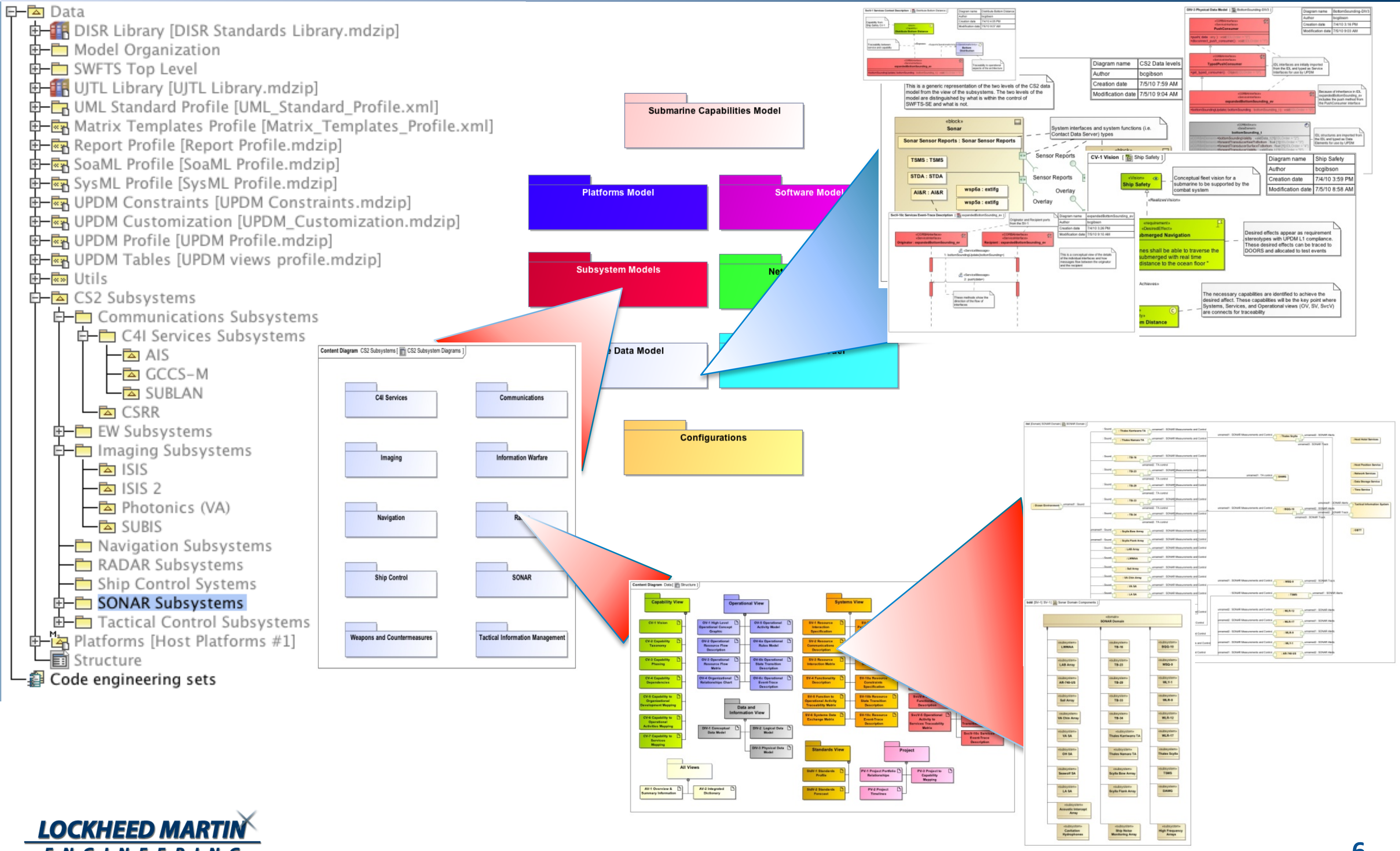
Network Splinter Group

SWFTS-SE System Services and Functions Definitions

Configurations

SWFTS Tech Insertion Group

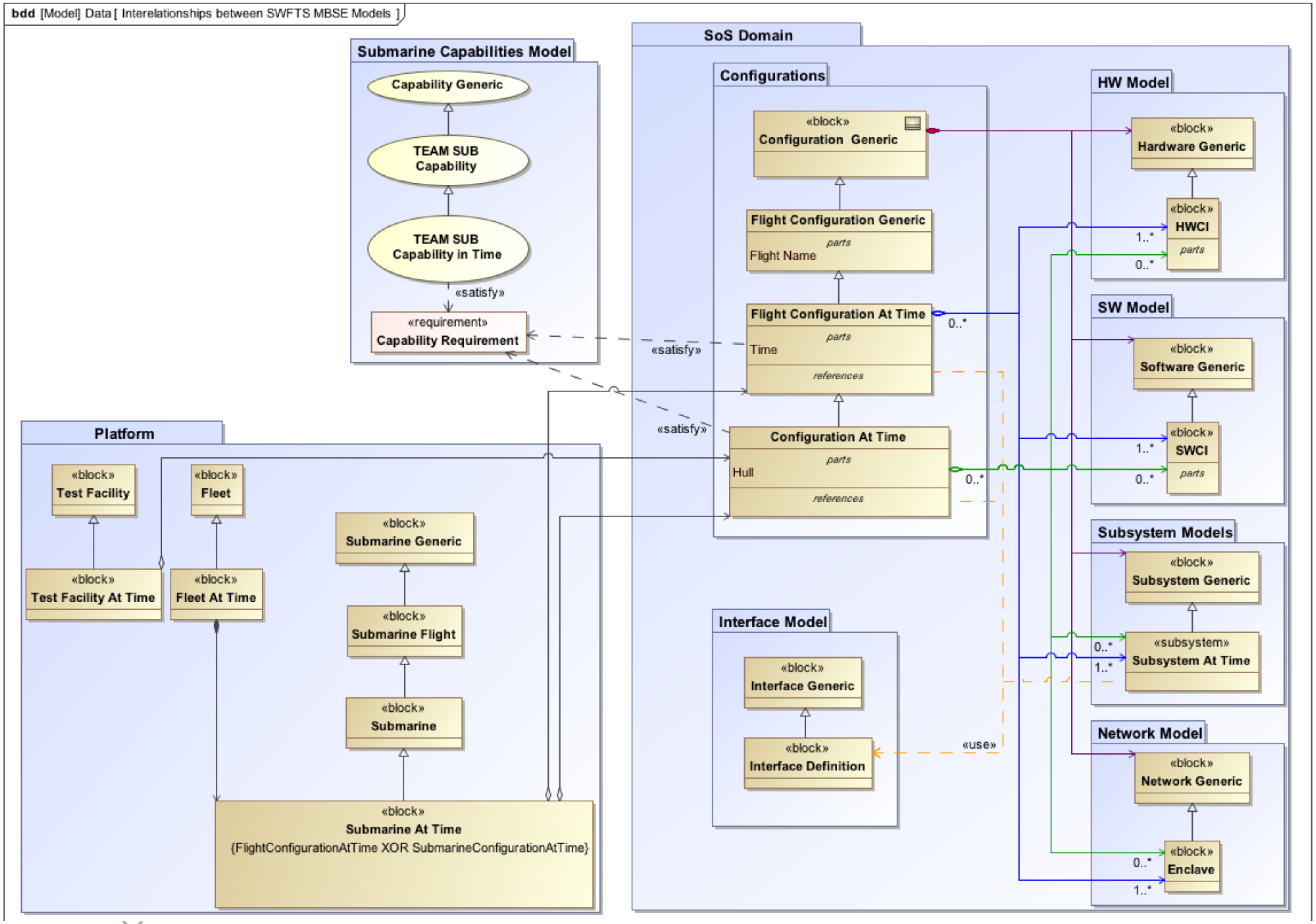
Hierarchy of Models Supporting TEAM SUBMARINE Engineering



- Data
  - DISR Library [DISR Standards Library.mdzip]
  - Model Organization
  - SWFTS Top Level
  - UJTL Library [UJTL Library.mdzip]
  - UML Standard Profile [UML_Standard_Profile.xml]
  - Matrix Templates Profile [Matrix_Templates_Profile.xml]
  - Report Profile [Report Profile.mdzip]
  - SoaML Profile [SoaML Profile.mdzip]
  - SysML Profile [SysML Profile.mdzip]
  - UPDM Constraints [UPDM Constraints.mdzip]
  - UPDM Customization [UPDM_Customization.mdzip]
  - UPDM Profile [UPDM Profile.mdzip]
  - UPDM Tables [UPDM views profile.mdzip]
  - Utils
  - CS2 Subsystems
    - Communications Subsystems
      - C4I Services Subsystems
        - AIS
        - GCCS-M
        - SUBLAN
      - CSRR
    - EW Subsystems
    - Imaging Subsystems
      - ISIS
      - ISIS 2
      - Photonics (VA)
      - SUBIS
    - Navigation Subsystems
    - RADAR Subsystems
    - Ship Control Systems
    - SONAR Subsystems
    - Tactical Control Subsystems
  - Platforms [Host Platforms #1]
  - Structure
- Code engineering sets

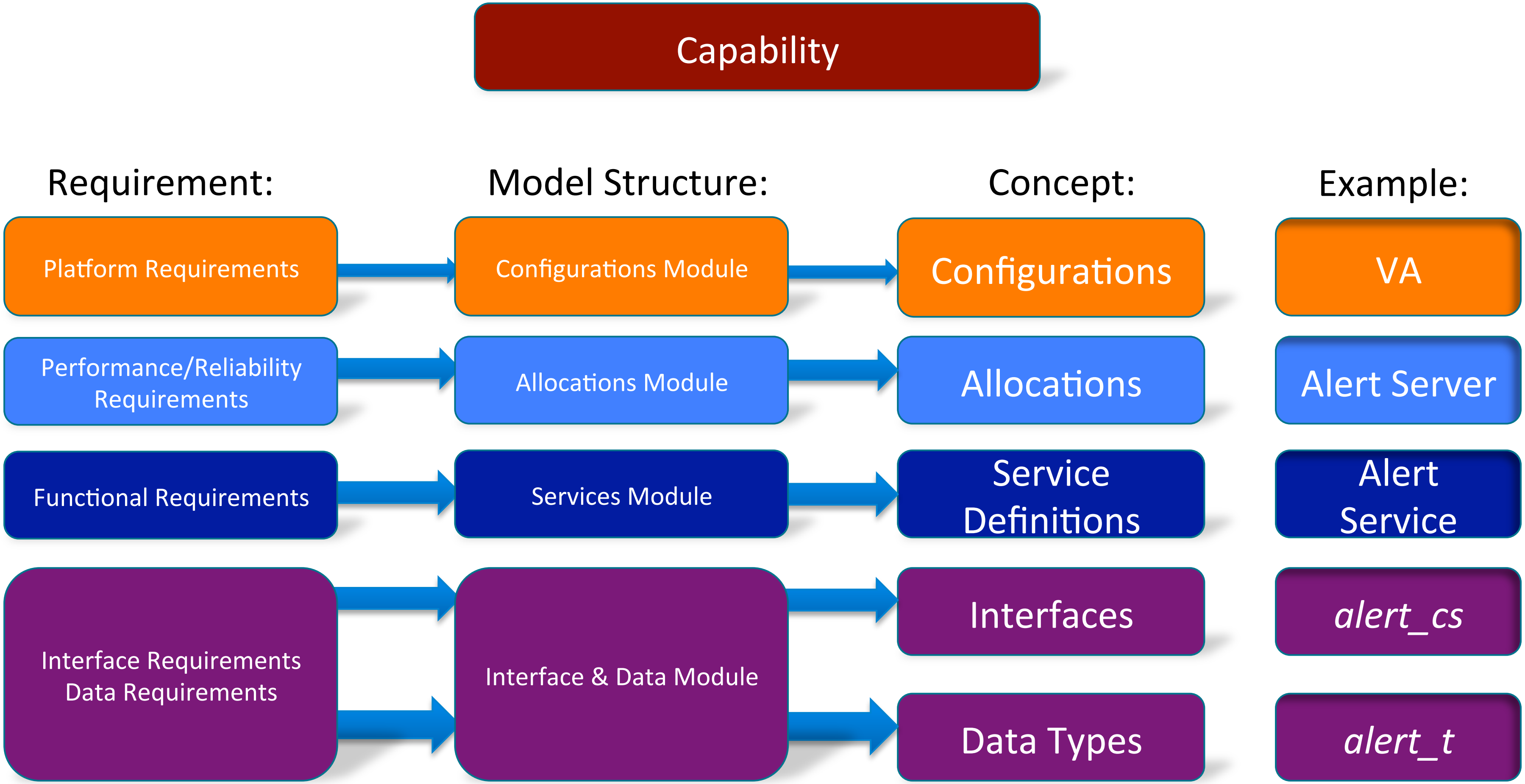
Submarine Capabilities Model
Platforms Model
Software Model
Subsystem Models
Net
Data Model
Configurations
Content Diagram CS2 Subsystems [CS2 Subsystem Diagrams]
C4I Services
Communications
Imaging
Information Warfare
Navigation
Ra
Ship Control
SONAR
Weapons and Countermeasures
Tactical Information Management
Content Diagram Data [Structure]
Capability View
Operational View
Systems View
Data and Information View
Standards View
Project
All Views
Diagram name
CS2 Data levels
Author
bcgibson
Creation date
7/5/10 7:59 AM
Modification date
7/5/10 9:04 AM
This is a generic representation of the two levels of the CS2 data model from the view of the subsystems. The two levels of the model are distinguished by what is within the control of SWFTS-SE and what is not.
«block»
Sonar
Sonar Sensor Reports : Sonar Sensor Reports
TSMS : TSMS
STDA : STDA
AI&R : AI&R
wsp6a : extifg
wsp5a : extifg
Sensor Reports
Overlay
System interfaces and system functions (i.e. Contact Data Server) types
CV-1 Vision [Ship Safety]
Ship Safety
Conceptual fleet vision for a submarine to be supported by the combat system
Diagram name
Ship Safety
Author
bcgibson
Creation date
7/4/10 3:59 PM
Modification date
7/5/10 8:58 AM
«RealizesVision»
«requirement»
«DesiredEffect»
Submerged Navigation
nes shall be able to traverse the submerged with real time distance to the ocean floor
Desired effects appear as requirement stereotypes with UPDM L1 compliance. These desired effects can be traced to DOORS and allocated to test events
Achieves
The necessary capabilities are identified to achieve the desired affect. These capabilities will be the key point where Systems, Services, and Operational views (OV, SV, SvcV) are connects for traceability
m Distance

SWFTS' Multiple Interdependent Models



LOCKHEED MARTIN
ENGINEERING

Logical Model Layout



Capability

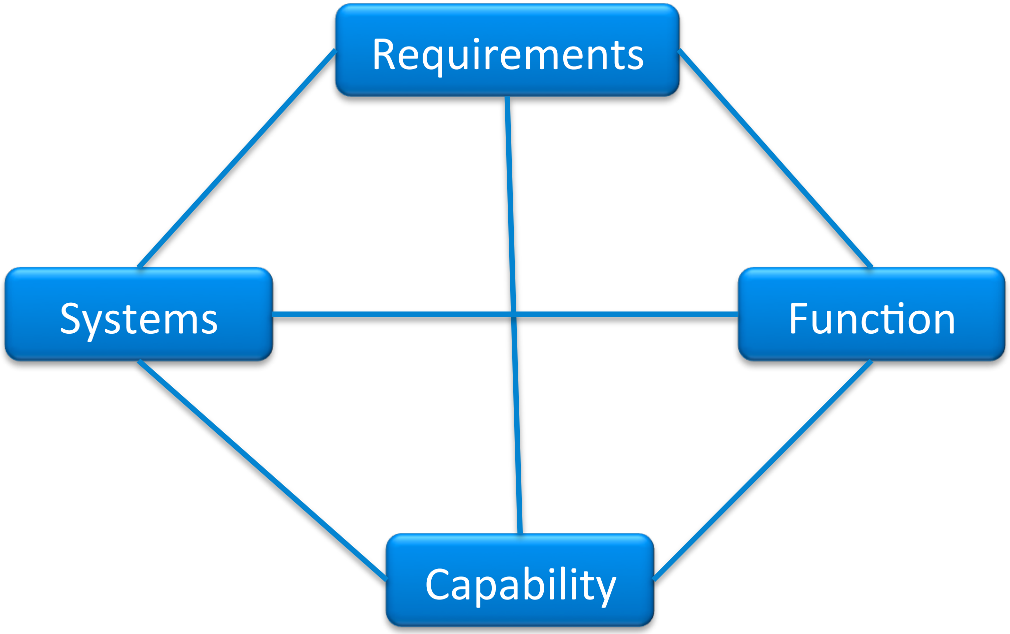
Requirement:	Model Structure:	Concept:	Example:
Platform Requirements	Configurations Module	Configurations	VA
Performance/Reliability Requirements	Allocations Module	Allocations	Alert Server
Functional Requirements	Services Module	Service Definitions	Alert Service
Interface Requirements Data Requirements	Interface & Data Module	Interfaces	*alert_cs*
		Data Types	*alert_t*

SWFTS Interface Baseline Model Size

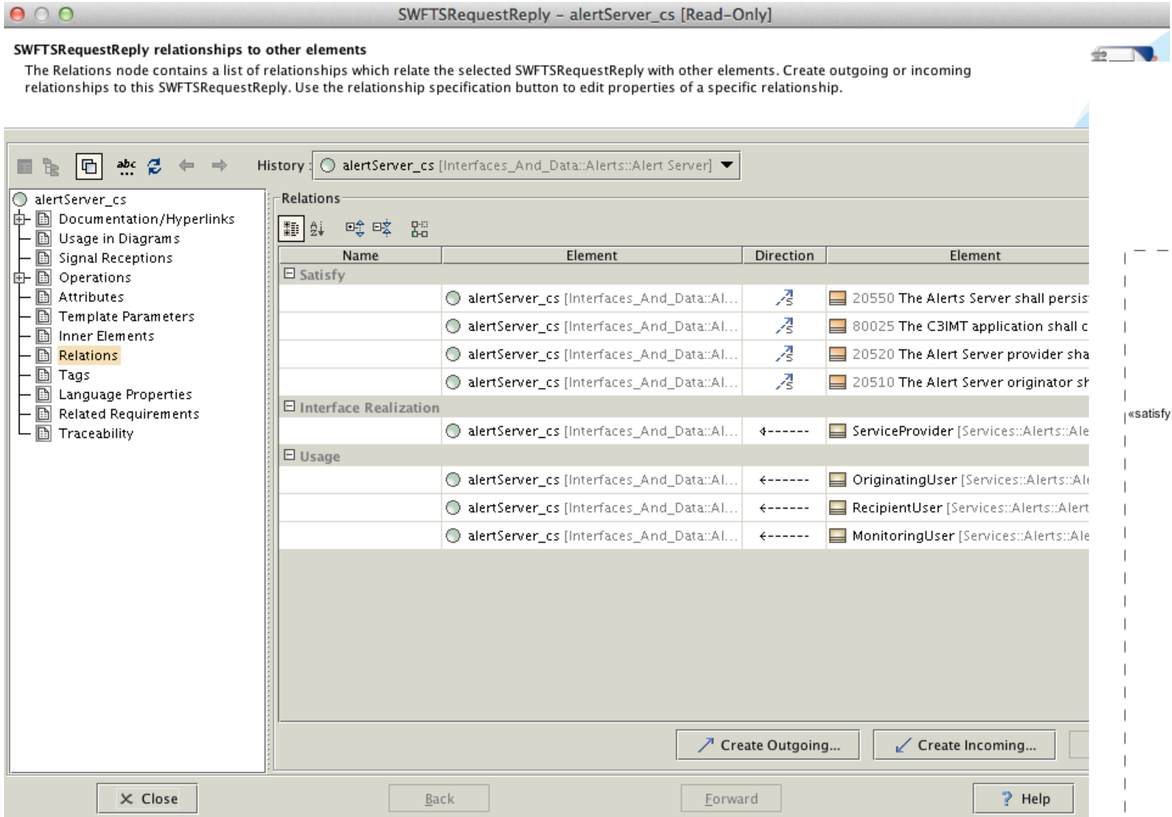
- ~35 subsystems from ~20 program offices
- ~3,500 interface requirements
- ~150 services
- ~3,700 model elements for interfaces
  - Interfaces, methods, data structures
- >15,000 relationships between model elements
- 517563 model elements

Traceability

- Document based SE approaches require expensive manual processes to maintain full traceability
- Model based SE provides a fully-related view of the federated system of systems



Full Requirements Traceability



SWFTSRequestReply – alertServer_cs [Read-Only]

SWFTSRequestReply relationships to other elements

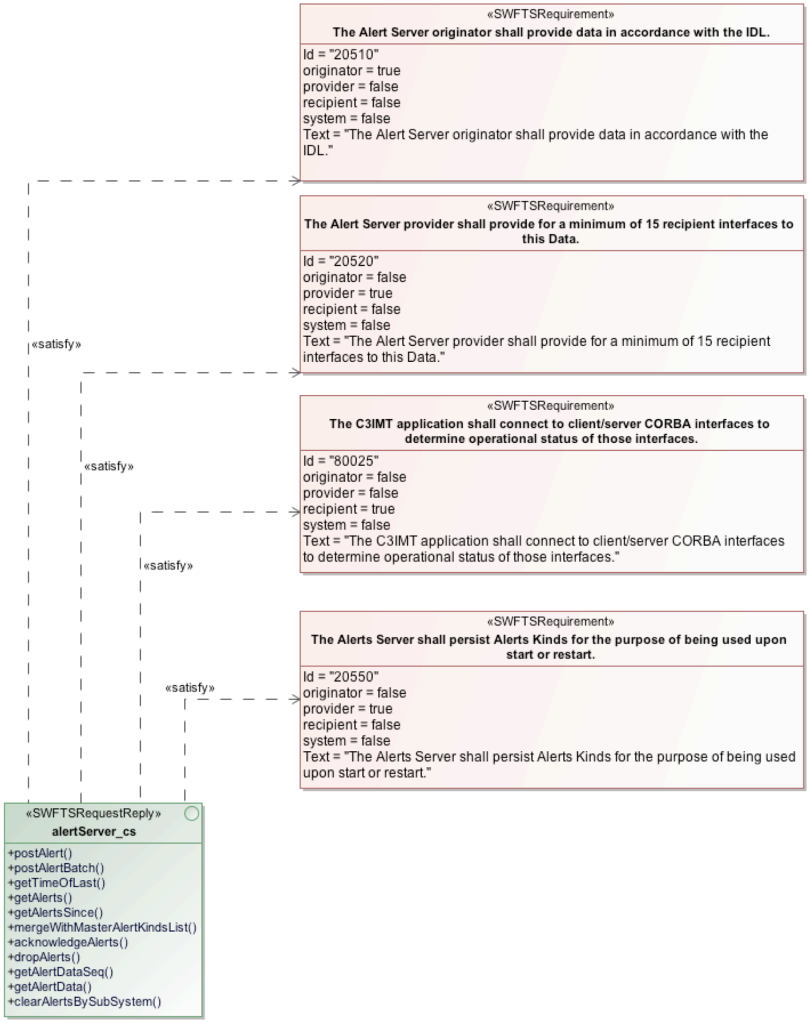
The Relations node contains a list of relationships which relate the selected SWFTSRequestReply with other elements. Create outgoing or incoming relationships to this SWFTSRequestReply. Use the relationship specification button to edit properties of a specific relationship.

History : ○ alertServer_cs [Interfaces_And_Data::Alerts::Alert Server]

- ○ alertServer_cs
  - Documentation/Hyperlinks
  - Usage in Diagrams
  - Signal Receptions
  - Operations
  - Attributes
  - Template Parameters
  - Inner Elements
  - Relations
  - Tags
  - Language Properties
  - Related Requirements
  - Traceability

Relations

Name	Element	Direction	Element
⊟ Satisfy			
	○ alertServer_cs [Interfaces_And_Data::Al...	↗	20550 The Alerts Server shall persis
	○ alertServer_cs [Interfaces_And_Data::Al...	↗	80025 The C3IMT application shall c
	○ alertServer_cs [Interfaces_And_Data::Al...	↗	20520 The Alert Server provider sha
	○ alertServer_cs [Interfaces_And_Data::Al...	↗	20510 The Alert Server originator sh
⊟ Interface Realization			
	○ alertServer_cs [Interfaces_And_Data::Al...	←------	ServiceProvider [Services::Alerts::Ale
⊟ Usage			
	○ alertServer_cs [Interfaces_And_Data::Al...	←------	OriginatingUser [Services::Alerts::Ale
	○ alertServer_cs [Interfaces_And_Data::Al...	←------	RecipientUser [Services::Alerts::Alert
	○ alertServer_cs [Interfaces_And_Data::Al...	←------	MonitoringUser [Services::Alerts::Ale



«SWFTSRequirement»
The Alert Server originator shall provide data in accordance with the IDL.
Id = "20510"
originator = true
provider = false
recipient = false
system = false
Text = "The Alert Server originator shall provide data in accordance with the IDL."

«SWFTSRequirement»
The Alert Server provider shall provide for a minimum of 15 recipient interfaces to this Data.
Id = "20520"
originator = false
provider = true
recipient = false
system = false
Text = "The Alert Server provider shall provide for a minimum of 15 recipient interfaces to this Data."

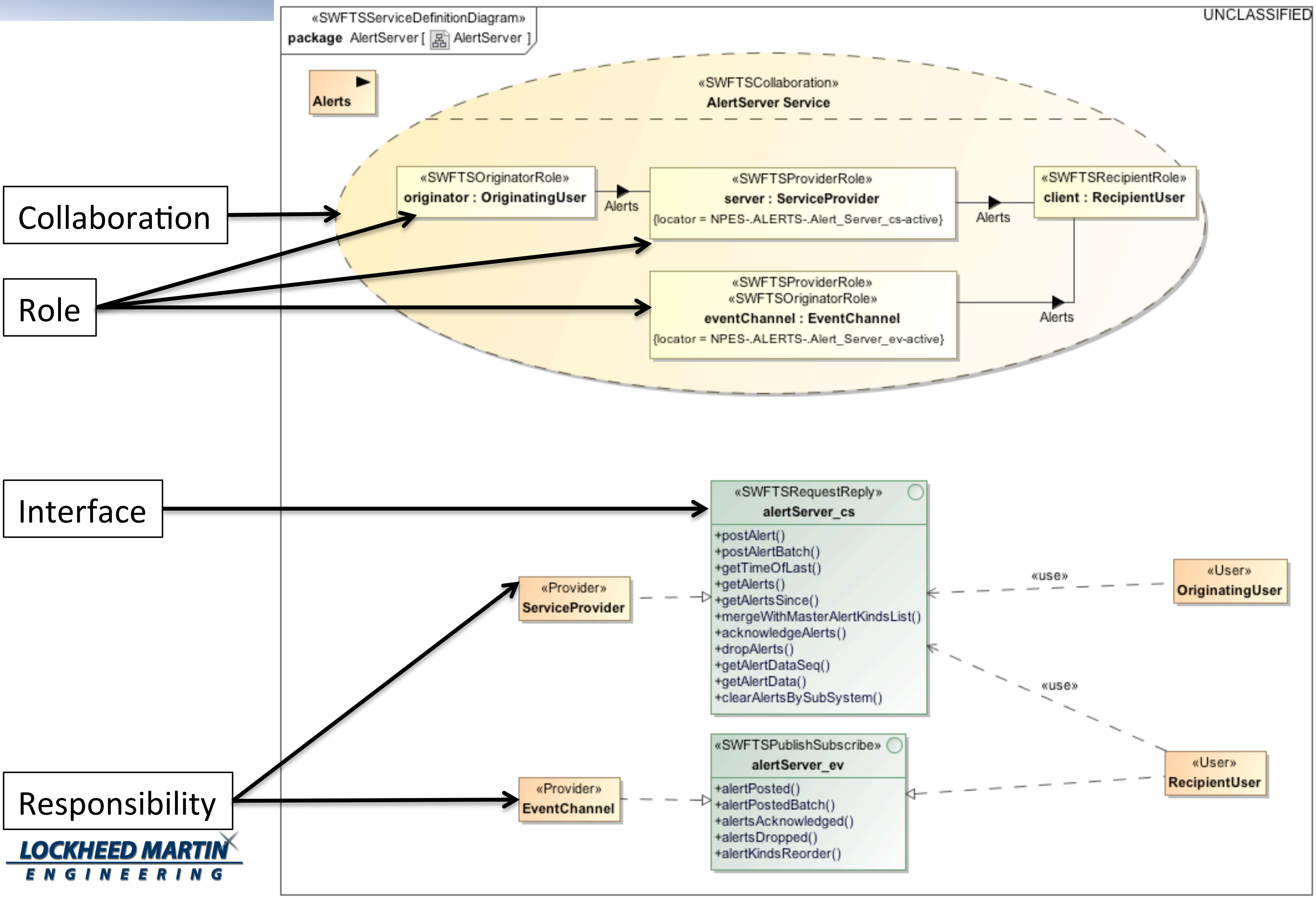
«SWFTSRequirement»
The C3IMT application shall connect to client/server CORBA interfaces to determine operational status of those interfaces.
Id = "80025"
originator = false
provider = false
recipient = true
system = false
Text = "The C3IMT application shall connect to client/server CORBA interfaces to determine operational status of those interfaces."

«SWFTSRequirement»
The Alerts Server shall persist Alerts Kinds for the purpose of being used upon start or restart.
Id = "20550"
originator = false
provider = true
recipient = false
system = false
Text = "The Alerts Server shall persist Alerts Kinds for the purpose of being used upon start or restart."

«SWFTSRequestReply»
alertServer_cs
+postAlert()
+postAlertBatch()
+getTimeOfLast()
+getAlerts()
+getAlertsSince()
+mergeWithMasterAlertKindsList()
+acknowledgeAlerts()
+dropAlerts()
+getAlertDataSeq()
+getAlertData()
+clearAlertsBySubSystem()

Model-Based Configuration of Services

UML Collaborations Define Service Roles



Collaboration

Role

Interface

Responsibility

«SWFTSServiceDefinitionDiagram»
package AlertServer [AlertServer]

UNCLASSIFIED

Alerts

«SWFTSCollaboration»
AlertServer Service

«SWFTSOriginatorRole»
originator : OriginatingUser

Alerts

«SWFTSProviderRole»
server : ServiceProvider
{locator = NPES-.ALERTS-.Alert_Server_cs-active}

Alerts

«SWFTSRecipientRole»
client : RecipientUser

«SWFTSProviderRole»
«SWFTSOriginatorRole»
eventChannel : EventChannel
{locator = NPES-.ALERTS-.Alert_Server_ev-active}

Alerts

«SWFTSRequestReply»
alertServer_cs

+postAlert()
+postAlertBatch()
+getTimeOfLast()
+getAlerts()
+getAlertsSince()
+mergeWithMasterAlertKindsList()
+acknowledgeAlerts()
+dropAlerts()
+getAlertDataSeq()
+getAlertData()
+clearAlertsBySubSystem()

«Provider»
ServiceProvider

«use»

«User»
OriginatingUser

«use»

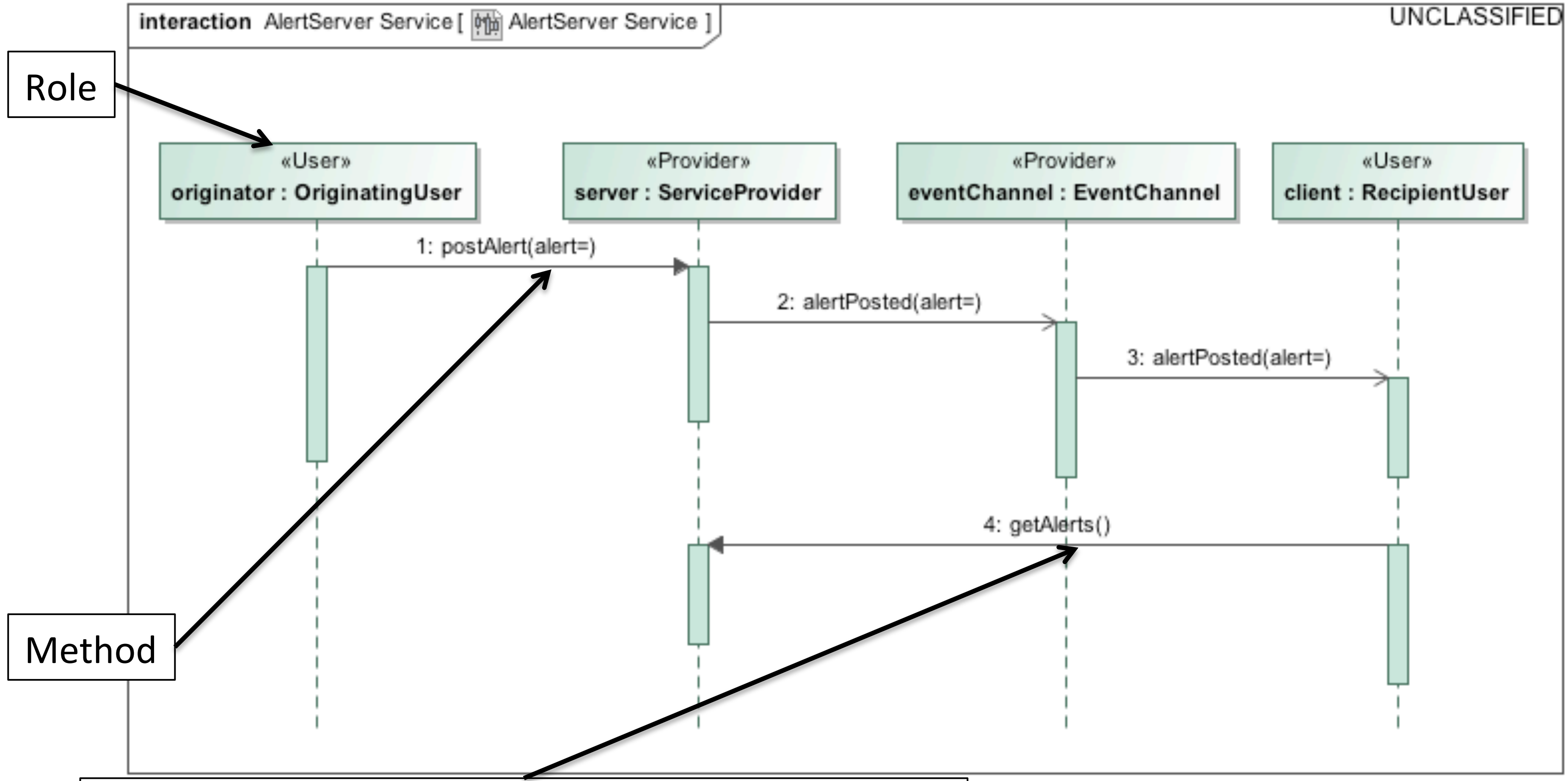
«SWFTSPublishSubscribe»
alertServer_ev

+alertPosted()
+alertPostedBatch()
+alertsAcknowledged()
+alertsDropped()
+alertKindsReorder()

«Provider»
EventChannel

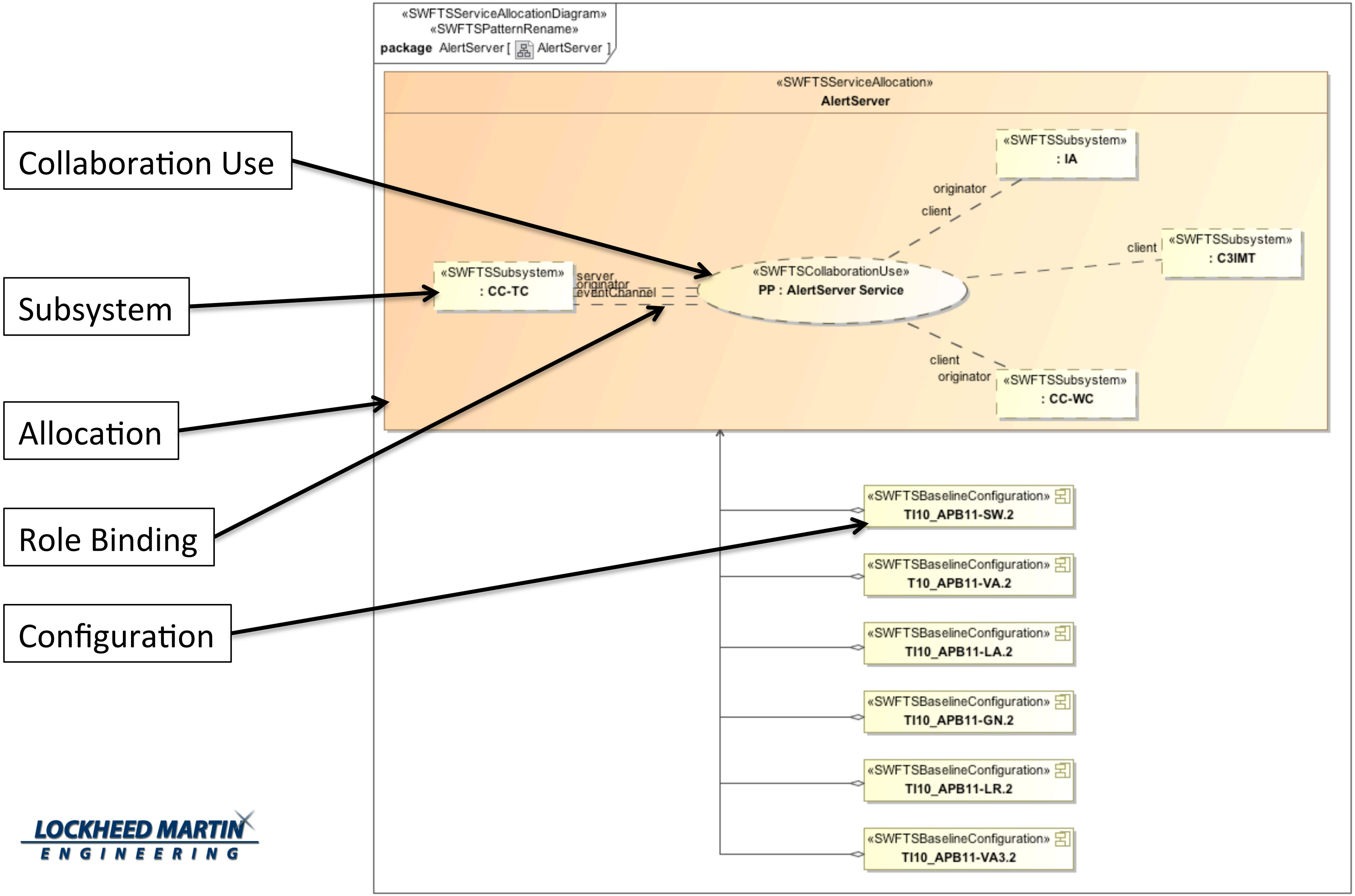
«User»
RecipientUser

Sequences Define Interface Methods



Return information is part of the method definition

Defining Services Configuration Baselines



Collaboration Use

Subsystem

Allocation

Role Binding

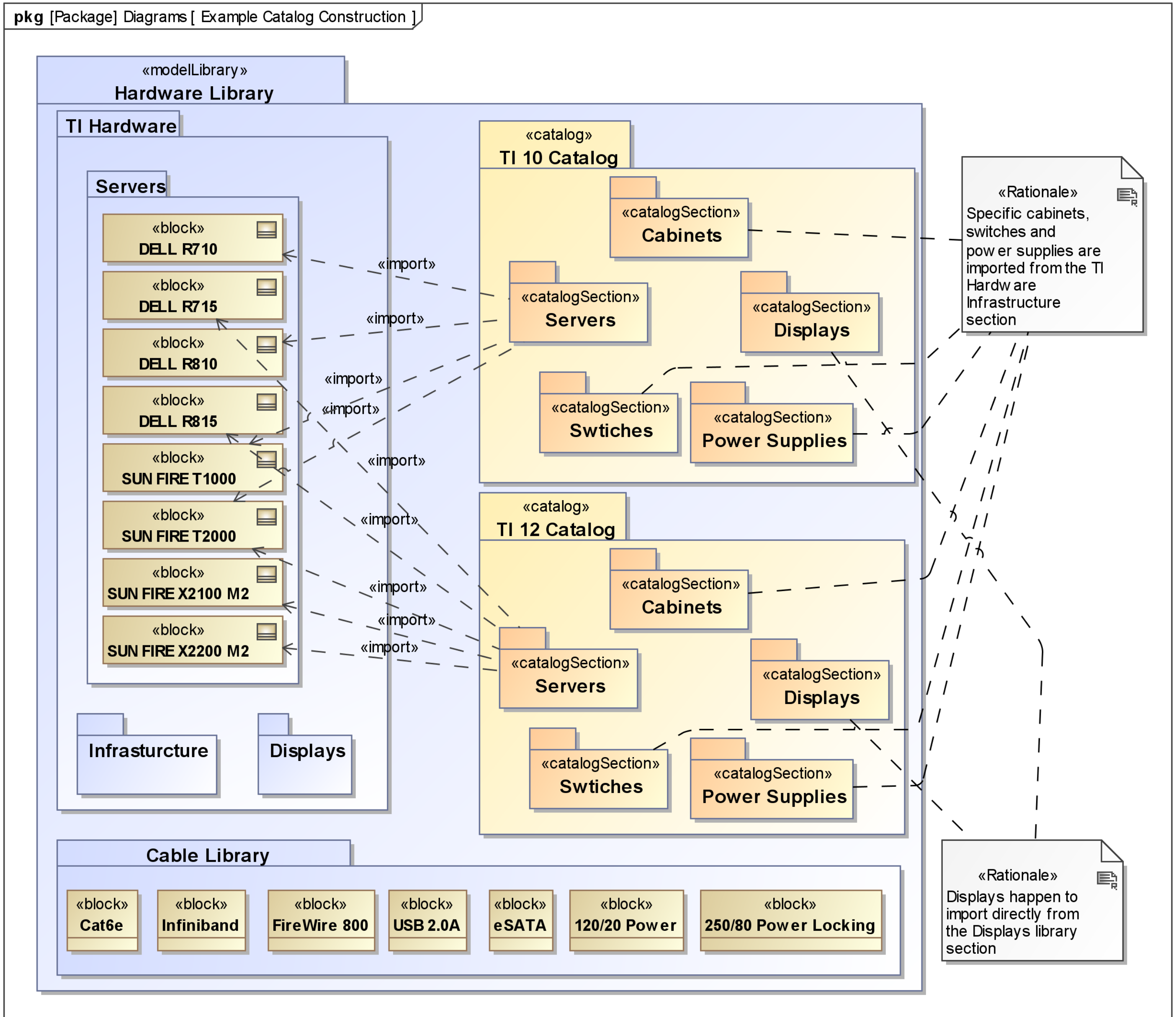
Configuration

Catalogs and Libraries

Managing Complexity

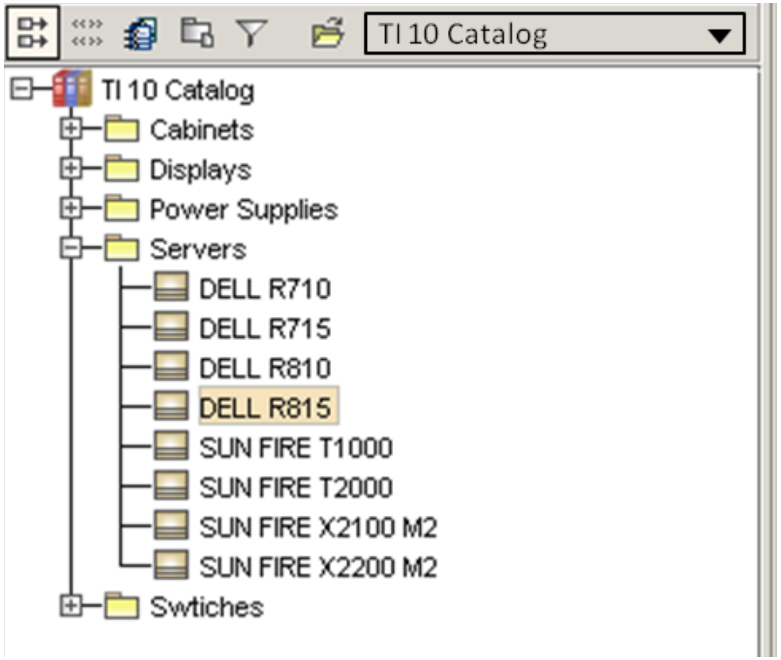
- Multiple Concurrent and Overlapping Baseline Configurations
  - Several concurrent iterations of the Systems Engineering "V"
  - Overlapping Advanced Development, Production Development, Deployment, Maintenance Cycles
    - Baselines deployed in 2002 are being actively maintained
- Maximizing Reuse Between Baselines
  - Defining ever-growing library data sets

Constructing Catalogs from Libraries



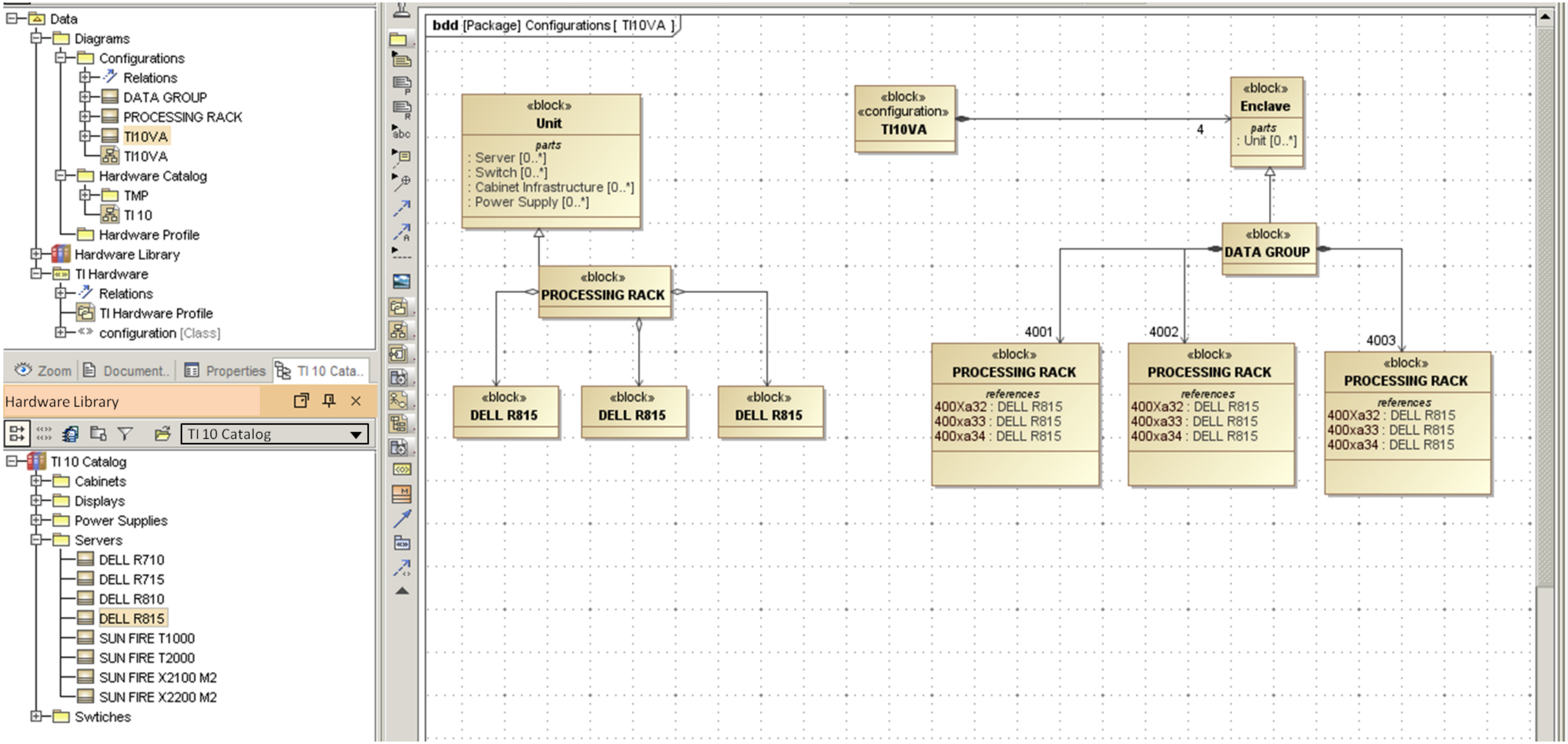
Catalogs Simplify the Configuration Task

- Catalogs frame alternative views of the model for the engineer
- Utilize the catalog as an active filter of the model
  - – Reduces the scope of the library without duplicating the elements
  - – Provides utilization assessments for elements across multiple baselines and baseline configurations



TI 10 Catalog
- Cabinets
- Displays
- Power Supplies
- Servers
  - DELL R710
  - DELL R715
  - DELL R810
  - DELL R815
  - SUN FIRE T1000
  - SUN FIRE T2000
  - SUN FIRE X2100 M2
  - SUN FIRE X2200 M2
- Swtiches

Assembling Baselines from Catalogs



Utility of Libraries and Catalogs Method

- Libraries and catalogs improve the quality and efficiency of the baseline configuration process
- Reduce duplication and inconsistency of element definitions
- Manage the complexity faced by systems engineers
- Provide intuitive tools for engineers to develop complex systems with maximum reuse

Observations and Conclusions

Preliminary Observations

- MBSE is improving efficiencies in how SWFTS-SE artifacts are managed and created
  - Reduced labor to identify potential impacts
  - Reduced time to modify subsystem allocations
  - More consistent, higher quality products
- Subsystems are able to utilize new information captured through SWFTS service-focused model based approach

Conclusions

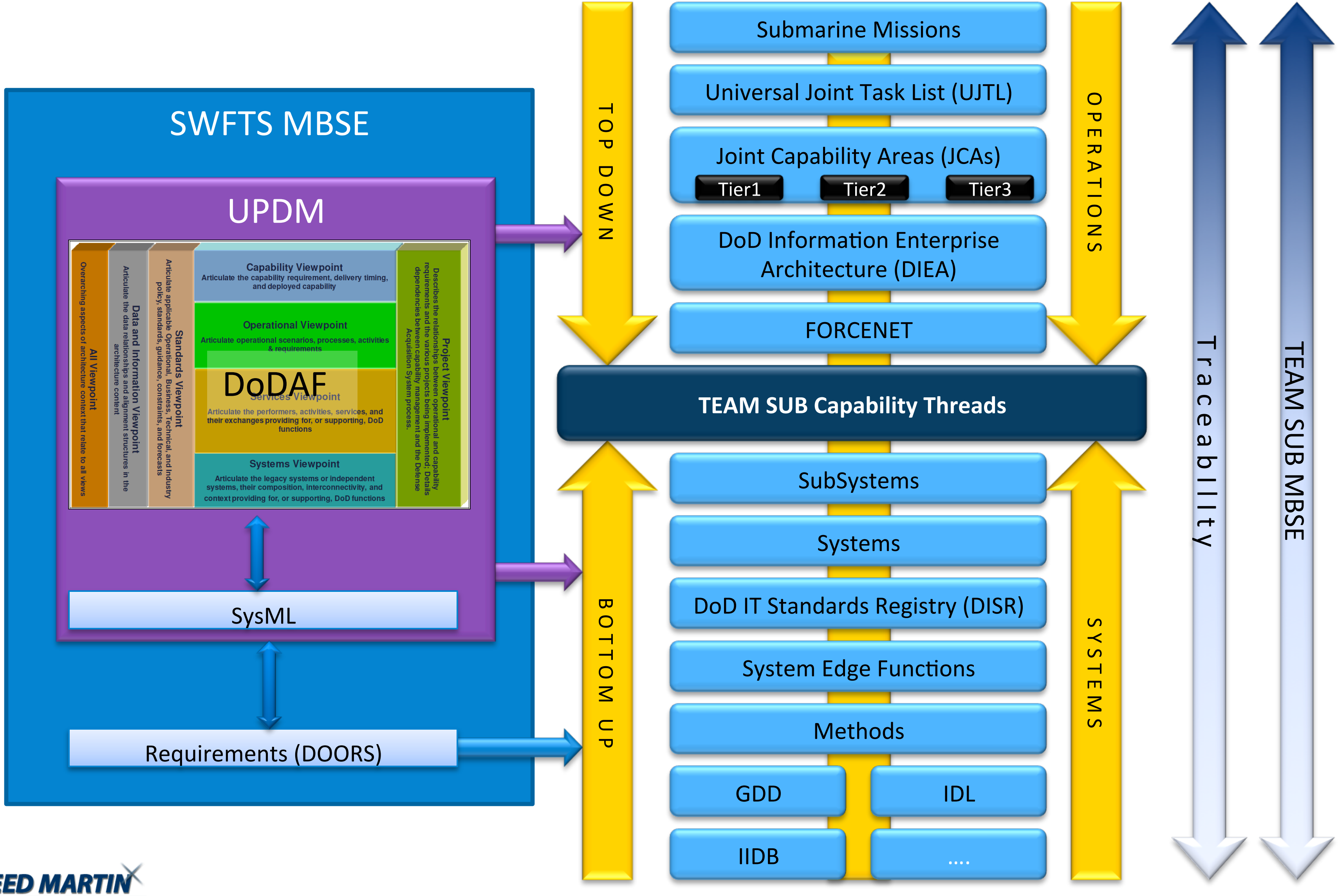
- SWFTS Model Based Systems Engineering improves SE efficiencies
- Subsystem impact are minimized by generating legacy-style artifacts
- MBSE enables SWFTS to maintain the engineering quality needed to continue evolving a federated System of Systems
- MBSE has potential to evolve into a full-lifecycle Team Submarine Enterprise model

Future Research

Evolve SWFTS Product Family Model

- Extend Current SWFTS MBSE Prototype to Full Information Model
  - May Discover Additional Limitations in UML, SysML, and UPDM Standards. If so, Work With Standards Bodies as Needed to Update
- Extend SWFTS MBSE Towards TEAM SUBMARINE Enterprise Architecture
  - Assess Interactions of Proposed Baseline Change Requests
  - Directly Support Selected Subsystem Providers
  - Interface with Shipyard Models to Enable Integrated Ship Performance Impact Assessments
  - Support Full Life Cycle

Extension of SWFTS SoS Model to TEAM SUBMARINE Enterprise Model



SWFTS MBSE

UPDM

DoDAF

All Viewpoint

Overarching aspects of architecture context that relate to all views

Data and Information Viewpoint

Articulate the data relationships and alignment structures in the architecture content

Standards Viewpoint

Articulate applicable Operational, Business, Technical, and Industry policy, standards, guidance, constraints, and forecasts

Capability Viewpoint

Articulate the capability requirement, delivery timing, and deployed capability

Operational Viewpoint

Articulate operational scenarios, processes, activities & requirements

Services Viewpoint

Articulate the performers, activities, services, and their exchanges providing for, or supporting, DoD functions

Systems Viewpoint

Articulate the legacy systems or independent systems, their composition, interconnectivity, and context providing for, or supporting, DoD functions

Project Viewpoint

Describes the relationships between operational and capability requirements and the various projects being implemented; Details dependencies between capability management and the Defense Acquisition System process.

SysML

Requirements (DOORS)

TOP DOWN

BOTTOM UP

OPERATIONS

SYSTEMS

Submarine Missions

Universal Joint Task List (UJTL)

Joint Capability Areas (JCAs)

Tier1 Tier2 Tier3

DoD Information Enterprise Architecture (DIEA)

FORCENET

TEAM SUB Capability Threads

SubSystems

Systems

DoD IT Standards Registry (DISR)

System Edge Functions

Methods

GDD

IDL

IIDB

....

Traceability

TEAM SUB MBSE

Projected 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

References and Acknowledgement

- **References**

  - ***Model-Based System Development for Managing the Evolution of a Common Submarine Combat System**,* Steven W. Mitchell, Lockheed Martin, AFCEA-GMU C4I Center 2010 Symposium on Critical Issues in C4I, 18–19 May 2010
    - [Paper and Briefing Available Here]
  - ***Complex Product Family Modeling for Common Submarine Combat System MBSE**,* Steven W. Mitchell, Lockheed Martin, Third International Conference on Model Based Systems Engineering, Fairfax, VA, Sept 2010
  - ***Bridging the Gap: Modeling Federated Combat Systems,*** B. Gibson, S. Mitchell, and D. Robinson, Lockheed Martin, Third International Conference on Model Based Systems Engineering, Fairfax, VA, Sept 2010
  - ***Efficient Management of Configurations in the Model-Based System Development of a Common Submarine Combat System,*** Steven W. Mitchell, Lockheed Martin, AFCEA-GMU C4I Center 2011 Symposium on Critical Issues in C4I, 24–25 May 2011
  - ***Efficiently Managing Product Baseline Configurations in the Model-Based System Development of a Combat System Product Family,*** Steven W. Mitchell, Lockheed Martin, INCOSE IS, Rome, Italy, 7–12 July 2012

- **Acknowledgement**

  - This research was supported by NAVSEA contract N00024-06-C-6272