System Engineering and Integration for Submarine Combat Systems in the COTS Environment

Dennis J. Cooper
SWFTS Program Manager
Lockheed Martin Integrated Systems

Joan Sienkiewicz
SWFTS Program Manager
General Dynamics Electric Boat

Mike Oliver
SWFTS Program Manager
Lockheed Martin Maritime Systems and Sensors
Submarine Combat System Paradigm Shift

- Submarine combat systems employ Commercial-Off-The-Shelf technology and open system standards to provide:
  - An affordable path to leverage commercial technology in parallel with the build-test-build development cycle
  - Asynchronous rapid transition of advanced functional improvements through modular software builds in a networked environment

- Harnessing this capability required a change to the system engineering, interface, integration, and configuration management processes

This SE&I Model Was Proven Successful on the Virginia Class Combat System
Today’s Dynamic Environment

- Varying acquisition “cycle times” between subsystems
- Fleet driven rapid introduction of warfighting capability
- Technology transition from custom to COTS
- Radical reductions and gaps in funding
- Spiral Development
- Increasing system complexity
- Software Intensive Systems
- Desire for commonality across platforms
- Network Centric / Interoperability
- Information Security and Assurance

*Business Models have evolved to adapt to the changing environment*
Architecture Contrast

Closed System

• Use of closely held, private interfaces, languages, data formats and protocols
• Critical importance given to unique design and implementation
• Use of individual company preferences to set and maintain interface specifications
• Vendor and technology dependence
• Difficult and more costly integration
• Use of sole-source vendor
• Expansion and upgrading usually requires considerable time, money and effort
• Components, interfaces, standards, and implementations are selected sequentially

Open System

• Use of publicly available and widely used interfaces, languages, data formats and protocols
• Critical importance is given to interface management and widely used conventions
• Use of group consensus process to maintain interface specifications
• Vendor and technology independence
• Easier and more cost effective to integrate
• Use of multiple vendors
• Easier, quicker and less expensive to upgrade
• Components, interfaces, standards, and implementations are selected interactively

Interface Management is the Key to Success

Source: ITEA Journal: September/October 2001 (Volume 22, Number 3)
Open System Model

- Focus on the network and subsystem to subsystem domains to define and manage:
  - Open Architecture / Open System Standards
  - Interface Design and Definition
    - Functional not physical
  - Network Design, Services and system performance
  - Network software solutions
  - Technology Evolution
  - Standardized processes and products

This Open System Model has been implemented on the Submarine Functional Interface Baselines by a disciplined and measured approach to interface development and data flow management
Virginia Class Submarine Combat System

Need to manage and control the interfaces for the 23 subsystems of the Virginia Class Submarine Combat System
SE&I provides Top Level Combat System and Interface requirements. All other requirements are managed by the Federated subsystems.

A Streamlined System Engineering Approach
Evolution of Interface Standardization

Traditional Approach

- SUBSYSTEM A
  - INTERPRETS IRS
  - GENERATES IDD
  - DESIGNS SUBSYSTEM

- IRS PROVIDED BY NAVY

- SUBSYSTEM B
  - INTERPRETS IRS
  - GENERATES IDD
  - DESIGNS SUBSYSTEM

- SUBSYSTEM Y
  - INVOLVED IN IDL WG
  - USE COMMON AGREED
  - TO IRS & IDD FOR DESIGN
  - USE COMMON IDL

- INTERFACE PROBLEMS FOUND DURING PLATFORM INTEGRATION
  - SIGNIFICANT $’s AND SCHEDULE

Network / Group Approach

- INTERFACE WORKING GROUPS
  - COMMON AGREED TO REQUIREMENTS
  - GROUP PRODUCTS (GDD, IDL)

- COMMON IDL

- EARLY IDENTIFICATION OF INTERFACE PROBLEMS/EARLY TESTING VIA EIT/WAN

- MINIMUM PLATFORM INTEGRATION
System Engineering
Network Architecture, Interface Design

Yesterday
Traditional Point to point Design Process

Today
Network / Group Product Design Process

Subsystem A
Application Software
CORBA

Subsystem B
Application Software
CORBA

Subsystem C
Application Software
CORBA

Subsystem D
Application Software
CORBA

Group Products
- Open System Architecture Requirements and Standards (OSA)
- Group Requirements List (GRL)
- Interface Definition Language (IDL)
- Group Data Dictionary (GDD)
- System Network Design

Interface Working Groups

Network
- OBTT
- HME
- CONTACTS
- COMMS
- NAV
- SONAR
- CORBA

Sonar Data
- Sonar Settings
- Streams

Infrastructure
- Architecture
- Networking
- X-Windows
- System Mgmt

Nav Data
- Own Ship Nav
- Charts

Contacts
- Misc / CC Settings

COMMS
- ECS

HME
- HM&E
- Environmental Alert
SE&I Interface Product Definitions

- **SE&I Interface Products under Configuration Management:**
  - **Group Requirements List (GRL)**
    - Database that contains all interface requirements for all subsystems.
  - **Group Data Dictionary (GDD)**
    - Contains definitions for all Interface Data Elements (IDEs) and IDE Assemblies (IDEAs) produced and consumed by subsystems within a Data Group.
  - **Interface Definition Language (IDL)**
    - IDL provides the standard interface between objects, and is the base mechanism for object interaction between subsystems.
  - **Interface Integration Data Base (IIDB)**
    - Provides details of what interfaces are required for each subsystem on a per hull basis. Grouped by interface types and contains the Interface / method / Name Space Name (CORBA), Service Name (Services), Subsystems, GRL linkages, GDD linkages and indications of Provider, Recipient, Originator.
SE&I Product Relationships

Request For Subsystem Requirement Change/Enhancement
Facilitators develop requirements and products based on request

Facilitators work with subsystems to generate IDL per requirement

IDL Embedded In Subsystem Application

GDD Auto generated from IDL

Subsystem Allocations

IDL

Subsystem Allocations

GDD

Auto generated from IDL

Method:
programmed procedure within an interface

Sequence No.

Subsystem Allocation

IIDL (CORBA)
- Used to verify subsystem I/F

Linked to a particular interface data rate

Method: programmed procedure within an interface
Interface Working Group Participants

- **Infrastructure:**
  - Lockheed Martin
  - Progeny
  - Electric Boat
  - GD-AIS
  - NUWC

- **Navigation:**
  - Lockheed Martin
  - Northrop Grumman

- **Contact Data:**
  - Raytheon
  - GD-AIS
  - Lockheed Martin
  - Kollmorgan
  - NUWC

- **On-Board Training:**
  - Electric Boat
  - Lockheed Martin

- **Hull/Mechanical Data:**
  - Electric Boat
  - NUWC

- **Communications:**
  - Electric Boat
  - Lockheed Martin
  - NUWC
  - SPAWAR

- **Environmental Data:**
  - Lockheed Martin
  - GD-AIS
  - Electric Boat

- Integrated Product Team structure to develop and maintain interface specifications and products (GRL, IDL, GDD, Network Design, Information Assurance)
- Adopts widely used standards, data formats, and protocols
- Defines key attributes of the network and services
- Consensus-based interface definition and design

“The key is in designing an architecture that is going to take advantage of commercial standards, the ability to pull pieces out and reuse them in other systems and platforms, and that allows third parties access”

*Delores Etter, Assistant Secretary of the Navy for RDA*
Interface Management designed in to streamline integration by reducing/eliminating at each of the layers of the Interface
SE&I Toolset

Fully integrated and web-enabled toolset for SE&I
What’s next?

• It doesn’t end with Interface Management!
  – Systems engineering processes and products must be balanced with integration products and processes
  – There is an inherent relationship between standards, interface engineering and system integration
    • System definition
    • Interface design and development
    • System test and integration
  – The confluence of an open systems architecture, standards and system integration leads to:
    • Easier technology insertion
    • Fielding capability faster
    • Streamlined system integration

**SE&I is a balanced approach between interface management, standards and system integration**
SE&I Process and Products

Fully integrated and web-enabled toolset for SE&I

Products:
- GRL
- GDD
- IDL
- IIDB

Products:
- SRVM
- Test Procedures
- Test Results
- PTRs
Test and Integration (The “I” in SE&I)

**Goals:**
- Achieve a single integrated, cohesive working Combat System
- Risk Mitigation
- Cost containment
- Schedule Control
- Optimize Government and Industry Lab resources
- Early integration and test of interfaces
- Pull integration and Lifetime Support issues from the fleet and solve in labs

**Approach:**
- Virtual Integration Facilities to optimize capital investments
- Structured interface testing
- Allow collaborative debug of problems from development lab
- Distribution of software builds/patches
- Remote access to development environments
- Responsiveness to Fleet needs
- Support Certification

*Networked Labs and Collaboration Provide Mechanisms for Early Integration Testing to Minimize Risk*
Automated SE&I Toolkit

Web-based Integration and Test Toolkit (WITT) Benefits
- Based on OSI Model: Expandable Environment
- Automated Test Procedure, SRVM, and ROA: Built-in Test Reuse
- Early System T&I Planning Suite with Auto-Lab-Utilization
- Universal Interface Debug Tools
- Online Test Pass/Fail Recording / Automated V&V
- Dashboard Style, Drill-Down Technical and Programmatic Metrics

Web-Enabled SE&I Toolkit Connects all Stakeholders
System Engineering & Integration Approach: Test What has Changed; When it is ready

Subsystem Schedule & Conduct

- Prioritization
- Debug
- Problem Isolation
- Resolution
- Verification

Facilities Plan

![Diagram showing System Engineering & Integration Approach]

System Integration approach has evolved through several iterations and provides complete management of Requirements, Tests, Facilities, Drops, and PTRs
Lessons Applied

• Be prepared for a major cultural change and manage it!
• Adopt standards widely developed/recognized by industry
• Focus on interface management not subsystem development
• Strive for consensus-based interface definition on all key interfaces early
• New business models and processes will be required
• Link functionality updates (APBs) to bundled interface baseline updates
• Be prepared to migrate with technology (standards lag the technology curve)
• All interfaces are not created equal
• Consider and develop a strategy for information assurance, fault management and fail-over early. Most standards ignore these areas!
• Plan for change!

Interface management is the key!