Surface Navy Combat Systems Engineering Strategy

Kathy Emery, Chief Architect
PEO Integrated Warfare Systems
4 March 2010
Open Architecture (OA) is a key enabler for meeting the CNO’s objectives

Naval OA is a multi-faceted business and technical strategy for acquiring and maintaining National Security Systems (NSS) as interoperable systems that adopt and exploit open-system design principles and architectures.

NAVAL OA CORE PRINCIPLES

- Modular design and design disclosure
- Reusable application software
- Interoperable joint warfighting applications and secure information exchange
- Life cycle affordability
- Increased competition and collaboration
Implementation of OA across the enterprise will yield many benefits

| Reduction in Time to Field | Decreased development and acquisition cycle times to field new warfighting capabilities  
|                           | Faster integration of open standards based systems |
| Increased Performance     | Improved operator performance thru delivery of cutting edge technologies and increased bandwidth capabilities from spiral developments and technology insertions |
| Improved Interoperability | Use of common services (e.g. common time reference)  
|                           | Use of common warfighting applications (e.g. track mgr)  
|                           | Use of published interfaces to standardize collaboration |
| Reduction in Risk         | Leverage proven reusable components  
|                           | Test early and often in the developmental cycle to minimize risk of delivering non-interoperable products |
| Cost Avoidance           | Cost avoidance from software re-use and use commodity COTS products at optimum prices  
|                           | Reduced training and streamlined lifecycle support |
Top Level Acquisition Process View

Product Line Engineering
- PEO IWS Level
  - Requirements Analysis
  - Product Line Architecture/Interface Mgmt
  - Modeling and Simulation
  - Science and Technology
  - FMS Management

Directorate Level
- Component Requirements Definition
- Component Development
- Component Test
- Capability Integration
- Capability Test
- System Test & Evaluation
- Certification
- Installation
- ILS & Training
- Readiness Assessment
- Combat/Warfare System Integration
- CVN
- Aegis DDG
- Aegis CG
- DDG 1000
- LCS
- AMPHIB
- FMS
- Integrated Warfare Systems & Capabilities
- Fleet

Integrated Customers
- JFCOM
- NNWC
- OPNAV
- DASN
- ONR
- International

Requirements
- Mission Archs
- Funding
- Regulations
- Policies
Implementing Open Architecture: Strategy, Interfaces and Open Standards

- Treat computing environment as a commodity
  - Select commercial mainstream COTS products that conform to well-established open system interface standards
  - Bundle specific COTS products for a given timeframe and revisit selections on a regular basis

- Isolate applications from high rate-of-change COTS through selection of standard APIs
  - Upgrade H/W and S/W Independently and on different refresh intervals

- Transform application development from single-platform development to multi-platform portfolio
  - Objective architecture defines key interfaces that support extensibility and reuse goals based on common data model
  - Eliminate redundant software development efforts
Information-Oriented Architecture Is Key to Defining Reusable, Extensible Components

- Define a common data model and information standard
- Component-to-network interfaces, not component-to-component
- Component interfaces are **coordinated*** and **authenticated****
- Expose information and post for any authorized subscriber to access
- Producers of information don’t have to be aware of consumers

*Coordinated = fully-specified IDD, Gov’t CM via ICWG
**Authenticated = interface compliance test before acceptance
Surface Combat System Network-Based Architecture

Common Core Software
- Sensor Management (local)
- Track Management (Common Tactical Picture)
- Combat & Control (incl. Tactical Mission planning)
- Weapon Management
- Vehicle Control (air, surface, underwater)
- Display / User Interface

Computing Environment*
- Infrastructure Services
- Computing Equipment (processors, displays, networks, storage, physical interfaces)

Support
- C2 Trainers
- Aviation Trainers
- Logistics Systems
- Total Ship Training System

Controlled Vehicles
(Air, surface, underwater)

Sensors
- Radar
- IFF
- ES
- EO/IR
- NCTR
- Acoustic

External Comms
- DDS
- CDL-N
- JTT
- ADNS
- TDL
- METOC
- C2/ISR
- Planning
- JPALS

Air Traffic Control
- Launchers
- Missiles
- Decoys
- Torpedoes
- Guns
- EA
- TTTCS

Weapons
- Nav Data Fusion System
- ECDIS-N
- AIS
- Nav Trainer
- Nav Sensors
- Time
- GPS
- Wind

Navigation

* Computing environment could be combat system specific or total ship
Transitioning to Objective Architecture Based Combat System

2008
- Aegis designed as an integrated combat system
- Aegis ACB 08 / TI 08 decoupled hardware from software
- SSDS designed with federated combat system network and hardware decoupled from software
- SSDS ACB 08 adds open standard middleware
- Future capability improvements planned for both programs through Advanced Capability Build (ACB)

2012
- Aegis modernization (ACB 12) component level interfaces delivered at CDR (10qtr FY10) and with each delivered computer program build
- SSDS interfaces already documented at component level
- Small number of common components integrated both Aegis & SSDS ACB 12

2014-2022
- Number of common components will increase with each ACB moving to a common software core for all Surface Navy Combat System

Objective Architecture

Required warfighting capabilities determine components modified
Increased computing power and network-based performance will enable significant combat system warfighting improvements.
## Evolution of Open Architecture

### Characteristics:
- Separation of Application/Infrastructure
- Commercial Standards
- Commodity Products
- COTS Performance Characterization
- Prototypes / EDMs
- Planned Refresh Cycles
- Component-Based Designs
- Networked Applications
- Configurable Test Environments
- Multi-Level Test and Evaluation
- KPP Validation
- Increased Reuse
- 3rd Party Developers
- Peer Reviews and Independent Assess
- Mentoring
- Fleet Involvement

### COTS Infrastructure
- • Increased Performance / Bandwidth
- • Reduced Cost

### Component-Based Software
- • Decreased Dev Time
- • Improved Testability
- • Reduced Cost (Reuse)
- • Scalability, Extensibility, Testability, ...

### Open Business Model
- • Open Business Practices
- • Rapid Transition of New Capabilities to Systems
- • Open Disclosure / Data Rights
- • Increased Number of Vendors/Opportunities
- • Improved Transition of S&T to Fleet

### Common Core Architecture
- • Common Objective Architecture / Interfaces
- • Common Components, Frameworks, Services
- • Common Precepts/Patterns/Standards
- • Improved Interoperability
- • Cost Avoidance
- • Reduced Training/Support

We are now focused here
PEO IWS Responsible for Achieving OA Objectives for Combat Systems

- Coordinate architecture and overarching interface principles for developing combat systems
- Oversee design, construction and maintenance of all ship combat systems
- Coordinate combat system acquisition programs across PEOs
- Leverage combat system software components across programs

“We must change from an approach that is optimized by program and platform to one that can solve the challenges of integrated systems that cross many platforms and functions…”

– ASN(RDA) MSG DTG 112123Z OCT 02

We are transforming to a product line acquisition approach
Navy Technical Reference Model

User Groups

Application Services
- Support Systems
- Aviation Systems
- ISR
- Command & Control
- Ship Control
- Combat Systems
- Platform Training

Common Services
- Basic Services
- SOA Core Services (Tactical Edge)
- Enterprise Services

Common Computing Environment
- Hosting Environment
- Computing Hardware

Communications & Networks
- WANs
- LANs
- Network/Circuit Mgmt
- SATCOM
- Wireless
- Security

Quality of Service (QOS)
- Information Assurance (IA)
- Data Architecture

User Groups: COI, COI, COI, COI
Notional Joint UAS Control Segment Software Framework

---

**Combat System Focus**

- Auto T-O/L
- Vehicle Status
- Vehicle Control
- Sense & Avoid
- Handover
- Comm Link Mgt
- Navigation
- ATC

**Payload Control Services**

- PLD Control
- PLD Status
- Comm Link Mgt

**Weapons Control Services**

- Weapon Status
- Weapon Control
- Find, Fix, Target

**Mission Control Services Bus**

- GIG
- Local
- Distributed
- HCI

---

**Supporting Services**

- Threat
- Route Planning
- Collection Planning
- Task Planning
- Navigation
- Weather

- Media Recording
- Track Storage
- Access, Info Ass
- Product Output

- Chat
- Tactical Mgmt
- Collaboration
- Audio Comm

- Task Processing
- Track Correlation
- Exploit Product
- Find, Fix, Target

- Link 16
- Blue Force Tracker
- AIS
- Training

- Geodetic
- Display Mgt
- Environment Feedback
- Network Mgt

---

**Content Management Services**

- System
- Product

**Planning Services**

- Training
- Network Mgt
- Sys Health, Maint.

---

**C4I Focus**

- Integrated App
- Integrated App
- Integrated App

---

**Integrated App**

- Display
- Local
- HCI

---

**Integrated App**

- Display
- Local
- HCI

---

**Integrated App**

- Display
- Local
- HCI

---
Today’s Shipboard Environment
(Direct interfaces, unique solutions, weak cross-domain integration)
Desired Shipboard Environment
(Networked interfaces, common/interoperable solutions, significant cross-domain integration)

External Comms Systems

Nav and HM&E Systems

Tactical Systems

Aviation Systems

Platform

C4I Systems

Functional Enclave

Interface Control

Network Management

(UNClassified)
CANES Services supports CS data exchange with C2 Applications (whether onboard and offboard)

Information Assurance is a Significant Hurtle to Resolve: PEO IWS and C4I will coordinate inputs to consolidated C&A activity
Common Component Requirements Flow from Combat System Requirements – PSEAs Involved

SYSTEM DEFINITION AND DESIGN

SOFTWARE ELEMENT REQUIREMENTS DEFINITION (B5 SPEC / IDS)

SUBSYSTEM REQUIREMENTS DEFINITION (B1 / B2 SPEC)

SYSTEM REQUIREMENTS DEFINITION (A SPEC / SSS / TPMs)

OPERATIONAL / CAPABILITY REQUIREMENTS DEFINITION (TLR / CDD / NCD / KPPs)

PSEAs

FUNCTIONAL BASELINE

ALLOCATED BASELINE

PRODUCT BASELINE

OPERATIONAL TESTS (DT&E / OT&E)

NAVY COMBAT SYSTEM TEST (SIT / DT) (Level 5)

SYSTEM TEST & EVALUATION, MULTI-ELEMENT INTEGRATION & TEST (MEIT) (Level 4/5)

ENGINEERING TEST & EVALUATION (ET&E) (Level 3)

ELEMENT INTEGRATION & TEST (Level 2)

SOFTWARE COMPONENT UNIT AND COMPONENT TEST (Level 1)

SOFTWARE DETAIL DEIGN (Design Model)

CODE

SOFTWARE COMPONENT REQUIREMENTS DEFINITION (B5 SPEC / SRS / IDS)

COMPONENT SOFTWARE REQUIREMENTS DEFINITION (B5 SPEC / IDS)

SUBSYSTEM REQUIREMENTS DEFINITION (B1 / B2 SPEC)

SYSTEM REQUIREMENTS DEFINITION (A SPEC / SSS / TPMs)

OPERATIONAL / CAPABILITY REQUIREMENTS DEFINITION (TLR / CDD / NCD / KPPs)

ALLOCATION BASELINE

PSEAs Involved

Software Component Developers
Process Definitions Needed for Each Phase from Strategic Planning to Delivery / Sustainment

**Strategic Planning**
- Mission Analysis
- Int. Product Roadmaps
- ACB Definitions
- Integrated POM Planning
- S&T Management
- Enterprise CCP

**ACB Execution / Combat System Integration**

**Delivery & Sustainment**
- Ship Integration
- Integ. Fielding Schedules
- Fleet Outreach
- Readiness/Perf. Asmnt

**Combat System T&E / Cert**

**Feeder Program / Common Asset Development**

**SE CONOPS**
A component is a bounded module with:
- Associated requirements (in DOORS)
- Design description
- Defined Interface (modeled in UML)

- Defined superset component requirements
- Establish interfaces IAW objective architecture
- Develop/deliver authenticated components

Available for fielding in CG(X) and backfit in 2014 & beyond
PEO IWS Product Line Approach for Surface Combat Systems

Based on a common architecture, with
- Common data standards and APIs
- Common services APIs
- Reusable software components (core assets)

Artifacts include:
- Requirements (functional, performance, interface, design)
- Design models, engineering studies and papers
- Source code, compile and build scripts, configuration files
- Test plans, procedures, harnesses, execution results, metrics
Benefits of Componentized Objective Architecture

- Common allocations and interfaces allow components to be reused across Combat Systems
  - Reuse reduces integration and test costs for new development
  - Improves interoperability and eases operator cross-decking

- Componentization localizes changes
  - Reduces Test / Cert costs for subsequent upgrades for component level changes

- S&T and new developers know how and where their products can fit in
  - Improved transition of new technology into Programs-of-Record

- Extensible to accommodate upcoming new warfighting capabilities:
  - Threat-D
  - MH-60R Integration
  - Netted Surface Tracking
  - Ship Protection Systems
  - Improved Surface and Underwater Pictures
  - Net-centric Services
  - Joint IFC / DWC
  - Hardkill / Softkill Coordination
  - Common Air Control
  - Fleet Synthetic Training
  - Distance Support
  - Maintenance Free Operating Periods
  - Optimized Manning Initiatives
**Product Line Approach**

**Way Ahead Perspectives**

### Technical Precepts
- Architecture
  - Component Based
  - Common Data Model
  - Network-based interfaces
  - Common framework for reqmts allocation
- Design for reuse / extensibility
- Extensive use of M&S and automated code, documentation and test tools & techniques

### Programmatic Precepts
- Spiral evolution process
  - Bi-annual Advanced Capability Builds (ACBs)
- Decouple new capability development from ACB dates
- Rapid transformation to common core software
- Rapid Capability Insertion Process (RCIP) development
- Integrated POM Inputs and program roadmaps
- Product Line Tasking & Funding to field activities

### Technical Precepts
- Architecture Precepts
- Objective Architecture
- Component Boundaries
- Key Interfaces
- Open Standards

### Programmatic Precepts
- Funding
- Schedule/Milestones
- People
- Requirements
- Platform Obligations
- Warfighting Commitments

### Acquirers (Formalized)
- IWS Systems Engineering Board
  - Cross-Program Coordination
  - Approval and Decision Process
- Architecture and Interface Control
- Enterprise Configuration Management
- Open Peer Review Process
- Common Asset Mgmt & Reuse Library
- Facilitate Information Transfer
- S&T Roadmaps and Transition Plans
- Org. Roles and Responsibilities

**Acq Mgmt Plan (AMP) V1.0**
Dec 2009

**Acquisition Precepts**
- Open Business Model
- Data Rights and OA incentives
- 3rd party development of components and capabilities
- Platform System Engineering Agents (PSEAs) for end-to-end C.S. engineering
- Competition at all levels

**Arch. Description Doc. (ADD) V1.0**
July 2009

**Sys Engr Mgmt Concept of Ops**
April 2010
## Business Characteristics of OA

<table>
<thead>
<tr>
<th>OPEN BUSINESS MODEL CHARACTERISTICS</th>
<th>OPEN SYSTEM MODEL CHARACTERISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ OA language in contracts</td>
<td>✓ Modular architecture</td>
</tr>
<tr>
<td>✓ Appropriate Data Rights</td>
<td>✓ Widely accepted/supported</td>
</tr>
<tr>
<td>✓ Design artifacts disclosed</td>
<td>standards</td>
</tr>
<tr>
<td>✓ Design artifacts published in repositories</td>
<td>✓ Use of commodity COTS</td>
</tr>
<tr>
<td>✓ Collaboration / Peer Reviews</td>
<td>✓ Published Interfaces</td>
</tr>
<tr>
<td>✓ Continuous competition</td>
<td>✓ Isolated proprietary components</td>
</tr>
<tr>
<td>✓ Rapid capability insertion process (RCIP)</td>
<td></td>
</tr>
<tr>
<td>✓ Fleet involvement</td>
<td></td>
</tr>
</tbody>
</table>
Executive plan to build and maintain Open Architecture Combat Systems
Provides objectives for stake holder alignment

Process for planning and specifying combat system upgrades for bi-annual Advanced Capability Builds

Describes software architecture for combat system product line with focus on Combat Management software

Integrated configuration management of overall portfolio of CS product line