The Use of Model-Driven Methodologies and Processes in Aegis Development

Jamie.Durbin@lmco.com
Christopher.M.Thompson@lmco.com

May 17, 2011
**The Use of Model-Driven Methodologies and Processes in Aegis Development**

**Summary**

Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License

<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>2. REPORT TYPE</th>
<th>3. DATES COVERED</th>
</tr>
</thead>
<tbody>
<tr>
<td>17 MAY 2011</td>
<td></td>
<td>00-00-2011 to 00-00-2011</td>
</tr>
</tbody>
</table>

**Authors**

Lockheed Martin Corporation, Chery Hill, NJ, 08002

**Supplementary Notes**

Approved for public release; distribution unlimited

**Subject Terms**

- Model-Driven Methodologies
- Aegis Development

**Security Classification**

- a. REPORT: unclassified
- b. ABSTRACT: unclassified
- c. THIS PAGE: unclassified

**Limitation of ABSTRACT**

Same as Report (SAR)

**Number of Pages**

- 30

**Distribution/Availability Statement**

Approved for public release; distribution unlimited

**Sponsoring/Monitoring Agency**

Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License

**Abstract**

[Abstract text]

**Security Classification**

- a. REPORT: unclassified
- b. ABSTRACT: unclassified
- c. THIS PAGE: unclassified

**Limitation of ABSTRACT**

Same as Report (SAR)

**Number of Pages**

- 30

**Name of Responsible Person**

[Name]

Approved for public release; distribution unlimited

**Field Reports**

- a. CONTRACT NUMBER
- b. GRANT NUMBER
- c. PROGRAM ELEMENT NUMBER
- d. PROJECT NUMBER
- e. TASK NUMBER
- f. WORK UNIT NUMBER

**Permit and Distribution**

Approved for public release; distribution unlimited

**Date of Report**

17 MAY 2011
Open Architecture Evolution
Phase 1: Commercial Technology…

- Commercial Standards
- Commodity Products
- Separation of Application/Infrastructure
- Planned Refresh Cycles

Leverage the Commercial Marketplace – Exploit Continuous Increases in Performance
Open Architecture Evolution
Phase 2: Componentized Software…

- Component-Based Designs
- Distributed Processing
- Message-Passing Architectures
- Modern Development Technologies

Decreased Development Time – Reduced Cost
Open Architecture Evolution
Phase 3: Open Business Model…

• **Open Disclosure / Data Rights**
• **Peer Reviews and Independent Assessments**
• **OAET – Contract Guidebook**
• **SHARE Repository**
• **Industry Days**
• **Technology Collaboration Centers**

*Increased Number of Players/Opportunities – Improved Transition of S&T Into Fleet*
Open Architecture Evolution
Step-by-Step…

- Commercial Standards
- Commodity Products
- Separation of Application/Infrastructure
- Planned Refresh Cycles
- Component-Based Designs
- Distributed Processing
- Message-Passing Architectures
- Modern Development Technologies
- Open Disclosure/Data Rights
- Peer Reviews and Independent Assessments
- OAET – Contract Guidebook
- SHARE Repository

Maximizing Affordability
Open Architecture Evolution

Combat System Alignment…

- **Product-Line Approach**
- **Decouple Capability Development from Platform**
- **Government Owned Architecture and Authenticated Interfaces**
- **High-Level Acquisition Strategy**
- **Objective Architecture**

**Reuse Capability Improvements – “Build it Once”**
Objective Architecture
Roles and Responsibilities...

- Objective Architecture
- Common Data Model and Messages
- Common Component Interfaces

- System Engineering and Integration of GFE Capabilities
- Support Peer Reviews
- End-to-End System Performance

- Technology Capability and Components

Navy Taking a Lead Role
Objective Architecture
Architectural Context…

• Consistent Domains / Boundaries
• Common Precepts and Methods
• Common Functional Allocation
• Common Data Model
• Core Common Components

Architecture Definition Document Provides “Guidance” for Product/Capability Development
Architecture Description Document

1. Intro
2. Reference Documents
3. Governance
4. Architecture Characteristics
5. Architecture Patterns
6. Architecture Views
7. Domains (11)

- Purpose
- Stakeholders
- Applicable Documents / Efforts
- Architectural Control at Component-Level

- Product-Line Approach
- Component-Based Architecture
- Pub/Sub Messaging / Common Data Model
- Fault Tolerance
- Quality of Service

- Precepts
- Domain-Specific Data Model
- Components
Top Level Objective Architecture

“Functional View…”

- External Communications Domain
- Display / User Interface Domain
- Vehicle Control Domain
- Weapon Mgmt Domain
- Sensor Mgmt Domain
- Track Mgmt Domain
- Command and Control Domain
- Support Domain
- Ship Control Domain
- Infrastructure Domain
- Training Domain
Top Level Objective Architecture

“Functional View…”

<table>
<thead>
<tr>
<th>External Communications Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Provide secure and non-secure voice/data communications</td>
</tr>
<tr>
<td>• Perform link management and data forwarding</td>
</tr>
<tr>
<td>• Control/monitor communications devices and route data among them</td>
</tr>
<tr>
<td>• Report communications status and effective throughput</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Display/User Interface Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Display Data (e.g., tracks, tactical situation, operational readiness, …)</td>
</tr>
<tr>
<td>• Provide set of common display services and frameworks for applications</td>
</tr>
<tr>
<td>• Support variety of display devices</td>
</tr>
<tr>
<td>• Display alerts and status</td>
</tr>
<tr>
<td>• Process operator actions</td>
</tr>
<tr>
<td>• Provide common/consistent GUIs across platforms</td>
</tr>
<tr>
<td>• Provide visualization tools</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vehicle Control Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Launch, recover, and control manned air vehicles (fixed wing and helo)</td>
</tr>
<tr>
<td>• Deconflict airspace/maintain safety of flight</td>
</tr>
<tr>
<td>• Control manned small boats</td>
</tr>
<tr>
<td>• Control unmanned vehicles (air, surface, undersea)</td>
</tr>
<tr>
<td>• Manage links to controlled vehicles</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Weapon Mgmt Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Process engagement orders</td>
</tr>
<tr>
<td>• Compute threat engageability</td>
</tr>
<tr>
<td>• Select weapons and schedule engagements</td>
</tr>
<tr>
<td>• Initialize/control weapons</td>
</tr>
<tr>
<td>• Monitor engagement status</td>
</tr>
<tr>
<td>• Monitor/control missiles in flight</td>
</tr>
<tr>
<td>• Assess engagement effectiveness/kill</td>
</tr>
<tr>
<td>• Report weapon and engagement status</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sensor Mgmt Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Manage and control sensors</td>
</tr>
<tr>
<td>• Search/detect objects in background environment</td>
</tr>
<tr>
<td>• Transition to track targets</td>
</tr>
<tr>
<td>• Report target measurement, track, and attribute data</td>
</tr>
<tr>
<td>• Assess and correct measurement errors/biases</td>
</tr>
<tr>
<td>• Report sensor status and capabilities</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Track Mgmt Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Create System Track File</td>
</tr>
<tr>
<td>• Contribute to operational picture</td>
</tr>
<tr>
<td>• Determine track ID</td>
</tr>
<tr>
<td>• Resolve ID conflicts</td>
</tr>
<tr>
<td>• Integrate multiple track data sources</td>
</tr>
<tr>
<td>• Perform geodetic registration</td>
</tr>
<tr>
<td>• Distribute track data according to need</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command and Control Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Control tactical doctrine</td>
</tr>
<tr>
<td>• Evaluate threats</td>
</tr>
<tr>
<td>• Evaluate tactical options</td>
</tr>
<tr>
<td>• Determine COA</td>
</tr>
<tr>
<td>• Plan missions</td>
</tr>
<tr>
<td>• Assess mission effectiveness</td>
</tr>
<tr>
<td>• Manage resources</td>
</tr>
<tr>
<td>• Provide decision aids for operators</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Support Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Evaluate overall ship readiness</td>
</tr>
<tr>
<td>• Provide data to fleet logistics systems and distance support</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Infrastructure Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Provide computing and network services for applications</td>
</tr>
<tr>
<td>• Monitor and control computing environment resources and report status</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Provide simulated training environment</td>
</tr>
<tr>
<td>• Script and execute training scenarios</td>
</tr>
<tr>
<td>• Simulate system inputs IAW scenario</td>
</tr>
<tr>
<td>• Control training configuration and events</td>
</tr>
<tr>
<td>• Capture training assessment data</td>
</tr>
<tr>
<td>• Playback training results and debrief crew</td>
</tr>
<tr>
<td>• Provide simulated ship position/movement</td>
</tr>
<tr>
<td>• Maintain training record</td>
</tr>
<tr>
<td>• Evaluate overall ship readiness</td>
</tr>
<tr>
<td>• Provide data to fleet logistics systems and distance support</td>
</tr>
</tbody>
</table>
Engineering Approach
Engineering Approach

Integrating Product-Line Components into Aegis...

• Establish Architecture Context:
  – Adopt Consistent Terminology, Allocation and Boundaries
  – Define Common Core Components – Key Interactions
  – Provide Overall Guidance – Architecture Definition Document (ADD)

• Normalize Requirements:
  – Utilize Government-led Working Groups
  – Compare/Contrast System Requirements and Design Approaches
  – Resolve Differences:
    • Establish Common Requirements
    • Establish System-Dependent Behavior
  – Verify Using System Threads

• Utilize Model-Based Tools/Methodologies:
  – Define Data Representation (e.g., Structure, Information-Flow)
  – Auto-Generate Code and Specifications/Documents
  – Provide Configurable and Flexible Performance

Data Model is Key
Model Driven Architecture
“Defining Components…”

System Threads

Comp A  Comp B  Comp C  Comp D

message

function

Data Model

Data at Rest
Data in Motion

- Map legacy data
- Normalize it
- Bin into messages

IPO per component

component
- resp data
- functions

- Captured in ADD
- Appropriate for including in SADs and SSDDs

SRS per common component

Applicable legacy requirements
New flowed down requirements

Common reqmts

DOORS

Functional Performance Interface Verification

Copyright © 2011 by Lockheed Martin Corporation
Data Model Overview

Top Level View

Plan

Doctrine

COA Intent

Service/Effect

Task

Controller

Coordinator

Commander

Planner

Plan Creates 1:* Planner

Doctrine Defines 1:* Planner

Plan Evaluates 1:* Commander

Task Directs 1:* Coordinator

Service/Effect Manages 1:* Coordinator

Task Directs 1:* Planner

Doctrine Constrains 1:* Service/Effect

Doctrine Initiates 1:* Service/Effect

Service/Effect Generates 1:* Task

Service/Effect Observes 1:* Environment

Task Directs 1:* Resource Schedule

Controller Schedules 1:* Resource Schedule

Task Directs 1:* Resource

Task Directs 1:* System

Task Directs 1:* Network

Resource Schedule Directs 1:* Resource Schedule

Resource Schedules 1:* Task

Resource Schedule Directs 1:* Resource

Resource Schedule Directs 1:* System Track

Resource Schedule Directs 1:* Network

Resource Schedules 1:* Resource Schedule

Resource Directs 1:* Resource Schedule

Resource Directs 1:* Resource

Resource Directs 1:* System

Resource Directs 1:* Network

Resource Uses 1:* Weapon

Resource Uses 1:* Vehicle

Resource Uses 1:* Sensor

Weapon Carries 1:* Resource

Weapon Carries 1:* Vehicle

Vehicle Carries 1:* Resource

Vehicle Carries 1:* Sensor

Sensor Produces 1:* Object

Sensor Detects 1:* Target

Sensor Detects 1:* Environment

Object Represents An 1:* Target

Object Contributes To 1:* System

Object Contributes To 1:* Network

Object Contributes To 1:* System Track

Object Contributes To 1:* Network

Object Participates In 1:* Network

Object Participates In 1:* System Track

Object Observes 1:* Environment

Environment Observes 1:* Object

Environment Observes 1:* Sensor

Target Engages 1:* Weapon

Target Engages 1:* Vehicle

Target Engages 1:* Sensor

Object Associated With 1:* Target

Object Represented An 1:* Target

Object Participates In 1:* System Track

Object Participates In 1:* Network

Object Contributes To 1:* System

Object Contributes To 1:* Network

Resource Schedule Schedules 1:* Task

Resource Schedule Schedules 1:* Task

Resource Schedule Schedules 1:* Task

Resource Schedule Schedules 1:* Task

Resource Schedule Schedules 1:* Task

Resource Schedule Schedules 1:* Task

Weapon Carries 1:* Resource

Weapon Carries 1:* Vehicle

Vehicle Carries 1:* Resource

Vehicle Carries 1:* Sensor

Sensor Produces 1:* Object

Sensor Detects 1:* Target

Sensor Detects 1:* Environment

Object Represents An 1:* Target

Object Contributes To 1:* System

Object Contributes To 1:* Network

Object Contributes To 1:* System Track

Object Contributes To 1:* Network

Object Participates In 1:* Network

Object Participates In 1:* System Track

Object Observes 1:* Environment

Environment Observes 1:* Object

Environment Observes 1:* Sensor

Target Engages 1:* Weapon

Target Engages 1:* Vehicle

Target Engages 1:* Sensor

Object Associated With 1:* Target

Object Represented An 1:* Target

Object Participates In 1:* System Track

Object Participates In 1:* Network

Object Contributes To 1:* System

Object Contributes To 1:* Network

Object Contributes To 1:* System Track

Object Contributes To 1:* Network

Object Observes 1:* Environment

Environment Observes 1:* Object

Environment Observes 1:* Sensor

Target Engages 1:* Weapon

Target Engages 1:* Vehicle

Target Engages 1:* Sensor

Object Associated With 1:* Target

Object Represented An 1:* Target

Object Participates In 1:* System Track

Object Participates In 1:* Network

Object Contributes To 1:* System

Object Contributes To 1:* Network

Object Contributes To 1:* System Track

Object Contributes To 1:* Network
Data Model Overview (cont.)

System Track Data Model

- Object
  - Represents An
  - Derived From
- Supporting Source Track
- System Track
  - Contains
    - Identity
    - Classification
    - Attribute State
    - Kinematic State
    - Ownership Attributes
    - Self-Reporting Unit Attributes
  - Contains
    - Intel Report
    - ES Parameters
    - Behavior
  - Contains
    - NCID Report
    - Space IFF Report
    - IFF Mode Code

- Vehicular Track
  - Land Track
  - Surface Track
  - Operation Area
  - Emergency Point
  - Impost Point
  - DLRP
  - Acoustic Bearing
  - Jammer

- Space Track
  - Area of Probability
  - Threat Reference Point
  - Launch Point
  - Sonoboy
  - EIW Bearing

- Subsurface Track
  - Ambiguous
  - Range Only
  - Range-Bearing
  - Bearing Only

Copyright © 2011 by Lockheed Martin Corporation
Model Driven Architecture
“Data Model Definition…”

- Provides Lexicon
- Describes Precepts and Desired Qualities
- Defines Boundaries/Components/interfaces
- Describes Optional Techniques and Methodologies

Conceptual Level
- **Generalized Data Model(s):**
  - Blocks of Information and Relationships
  - Major Attributes and/or Groups
  - Major Interaction of Components:
    - Groups of “Flows” – not actual messages
    - Significant Library Services / Plug-ins

Logical Level
- **Actual Data Representation:**
  - Specific Classes, Attributes and Types (ie “data at rest”)
  - Specific Messages/Fields (ie, “data in motion”)
  - Actual Data and Software Method API’s

- Use Cases/Sequence/Activity Diagrams
- Requirements reference messages, notifications, APIs
Sample Track Attributes

Extracted from track data model...
Sample Track Server APIs

Extracted from track data model...

TrackAccessService
- getSystemTrack
- getKinematicState
- getIdentification
- getIdentity
- getClassification
- getEngagedTrackStatus
- getBehavior
- getSourceTrackList
- getSourceTrack
- getNextTrack
- getOwnership
- getClientData
- getTrackKey
- getSourceTrackKey
- getTrackAssociations
- getTrackCount
- getTrackCounts

TrackMaintenanceService
- newTrack
- updateTrack
- dropTrack
- purgeStaleTracks
- purgeTrainingTracks
- purgeAllTracks
- updateSystemTrackList
- trackTypesAllowed
- maintainVariablePurgePeriodOrder
- getNextEmptySlot

TrackExtrapolationService
- extrapolateTrack
- extrapolateTrack

NotificationService
- subscribe
- unsubscribe
- subscribe
- unsubscribe
- unsubscribe
- unsubscribeAll

NotificationTypes
- NEW_TRACK
- UPDATE_TRACK
- DROP_TRACK
- HEIGH_UPDATE
- CORRELATION_DECLARED
- DECORRELATION_DECLARED
- INTERCHANGE_DECLARED
- ADD_REPORTING_SOURCE
- REMOVE_REPORTING_SOURCE
- SENSOR_SOURCE_EVENT
- HEIGHT_SOURCE_EVENT
- FIRM_TRACK
- MAINTENANCE_UPDATE
- TRACK_PARAMETER
- ACTIVE_TBM_CONS_EVENT
- TN_ASSIGNMENT
- TAC_SIG_EVENT
- CORRELATION_EVENT
- COVARIANCE_EVENT
- HIGH_PRIORITY_NEW_VEHICULAR_TRACK
- HIGH_PRIORITY_UPDATE_VEHICULAR_TRACK
- HIGH_PRIORITY_LOST_TRACK
- IDENTIFICATION_CHANGE
- GEOFIX_UPDATE_EW_TRACK

SystemTrackTypes
- ACOUSTIC
- EW
- SPECIAL_POI...
- IFF
- VEHICULAR

SourceTrackTypes
- «Enum»
Component Framework
Operational Mode Transitions...

Combat System Operational Modes

Transition Modes

Safe

T2: Ordered Safe
T3: Transition to Tactical Complete

T11: Transition to Test Complete

Test

T12: Ordered Safe

Test_To_Safe

T13: Transition from Test to Safe complete

Safe

T4: Ordered Safe
T5: Transition from Tactical to Safe complete

T10: Ordered Test

Safe_To_Tactical

T6: Ordered Training

Safe_To_Training

T7: Transition to Training Complete

Training

T8: Ordered Safe

Training_To_Safe

T9: Transition from Training to Safe Complete

T14: Ordered Safe

T15: Ordered Safe

T16: Ordered Safe

Tactical

T1: Ordered Safe
T2: Ordered Safe
T3: Transition to Tactical Complete
T4: Ordered Safe
T5: Transition from Tactical to Safe complete
T6: Ordered Training
T7: Transition to Training Complete
T8: Ordered Safe
T9: Transition from Training to Safe Complete
T10: Ordered Test
T11: Transition to Test Complete
T12: Ordered Safe
T13: Transition from Test to Safe complete
T14: Ordered Safe
T15: Ordered Safe
T16: Ordered Safe

Copyright © 2011 by Lockheed Martin Corporation
Component Framework
Availability and Presence States…

- Deploy
- Uninstantiated
  - Ordered Into Execution
  - Instantiating
    - Instantiated
      - Active
      - Standby
      - Ordered to Remove Assignment
- Ordered to Remove Assignment
- Quiesced
  - Ordered Quiesced
  - Report Work Completed
- Ordered Active
  - Active
  - Ordered Standby
  - Standby
- Ordered to Remove Assignment
- Instantiation Failed
  - Instantiation Failed after n attempts
- Terminating
  - Normally Terminating
    - Cleaning Up
  - Termination Failed
  - Cleanup Failed
- Termination Failed
  - Administratively Repaired
- Success Reported
- Fatal Error Detected
Model Driven Architecture
“Code Generation…”

**Infrastructure Services**
- Availability Management
- Data Recording
- Logging
- Navigation
- Operational Mode
- Pub/Sub
- Sub Component
- Time

**Rhapsody**
- Class Diagrams
- States and Modes
- Interfaces
- Sequence Diagrams

**Component Framework UML Model**

**Track Data**
- Class Diagrams
- Message/Field Definition
- Interfaces

**Track UML Model**

**Code Generator / Scripts**

**Generated Files**
- .h
- .cpp
- .mak
- sh

**Platform Source Include Files**
- Exceptions
- Data Types / Enums
- Method API’s
- Classes/Structures
- Build Directives

**Generated Files**
- .h
- .cpp
- .IDL
- .mak
- sh

**Platform Source Include Files**
- Data Types / Enums
- Method API’s
- Classes/Structures
- Message IDL
- Build Directives

**Track Data**
- Track Types and Attributes
- Distribution / Access Methods
- Notifications
- Messages

**Platform-specific Parameters (C++ / Java)**
Model Driven Architecture
“Product-Line Component Generation…”

Rhapsody
- Class Diagrams
- State Diagrams
- Interfaces
- Sequence / Activity Diagrams
- Method Code “Container”

STM/TS UML Model
- Supplementary Graphics

Code Generator / Scripts
- .h
- .cpp
- .mak
- .sh

Reporting
- SDD (Generated)

Format Template

MS Word

GCC
- Compiler / Linker
- Executable
- XML Config Files

STM/TS UML Model

SM/TS Source

Supplementary Graphics

Code Generation Features:
- Object Execution Framework (Not Used)
- State Machine (Not Used)
- Class Relationships (USED)

Limited to Promote Portability of Modeling Tools
Variation Techniques
Providing flexibility for SW component behavior…

Software Generation
- Arch Models
- Software Library
- Generate / Update SW Model
- Generate / Update Code

Build-Time
- Config Files
- Selected Files
- Configure Build
- Compile / Link

Run-Time
- Config Files
- Doctrine & Ref DBs
- Instantiate / Initialize
- Execution

Software Architecture and Design
- Software Code
- Identification of Files
- Compilation Instructions & Dependencies
- Run-Time Parameters

Run-Time Parameters
- Run-Time Parameters
- Instantiate / Initialize
- Execution

Variation Techniques
- Core / Adapt/ Library Services / Plug-in Functional Allocation
- Extensible Frameworks
- PIM / PSM Separation
- Applicability and Version Tags
- Model Compile Scripts and Configuration Files
- Source Code Compiler Tags
- Static Plug-in Binding
- Library Services Binding
- File Applicability and Version Tags
- Compiler Directives & Scripts
- Makefiles
- Build Scripts
- Site-specific Configuration Files
- Platform Adaptation Data
- Dynamic Plug-ins
- Install Scripts
- Dynamic Registration of Services
- Untriggered Code Left In Place
- Doctrine and Settings / Controls
Model Driven Architecture
“Putting It All Together…”

AMOD ACB 12

A-Level
System Spec

B1-Level
Element PIDS

B5-Level
Software Specs

C5-Level
Software Design

System Component Model

SAD (Generated)

DOORS Database

Component Framework UML Model

Infrastructure Services APIs (Generated)

TS APIs (Generated)

Message Definitions (Generated)

Component UML Models

SRS (Generated)

DOORS Database

Component Code (Generated)

IDD (Generated)

ADD

Configuration Control

Copyright © 2011 by Lockheed Martin Corporation

Durbin – May 17, 2011 - 27
Summary

• **Navy Has Followed a Comprehensive Plan Towards Open Architecture**
  – Utilization of COTS Technologies and Products
  – Incorporation of Component-Based Designs
  – Increased Number of Players/Opportunities

• **Product Line Approach Established to Facilitate Combat System Alignment**
  – Government Owned Architecture and Authenticated Interfaces
  – High-Level Acquisition Strategy
  – Robust Reference (aka Objective) Architecture

• **Model-Driven Tools/Methodologies Have Been Used Effectively**
  – Data Model Representation of Information Structure and Flow
  – Auto-Generation of Code and Specifications/Documents
  – Initial Common Components Have Been Successfully Integrated into Aegis Baseline 9
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACB08</td>
<td>Advanced Capability Baseline 2008</td>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>ACB12</td>
<td>Advanced Capability Baseline 2012</td>
<td>LM</td>
<td>Lockheed Martin</td>
</tr>
<tr>
<td>ACS</td>
<td>Aegis Combat System</td>
<td>LOT</td>
<td>Launch on TADIL</td>
</tr>
<tr>
<td>ADD</td>
<td>Architecture Definition Document</td>
<td>MMSP</td>
<td>Multi-Mission Signal Processor</td>
</tr>
<tr>
<td>Aegis</td>
<td>(not an acronym) Greek Shield of Zeus</td>
<td>MS</td>
<td>Microsoft</td>
</tr>
<tr>
<td>ALIS</td>
<td>Aegis LAN Interconnect System</td>
<td>NIFC-CA</td>
<td>Naval Integrated Fire Control - Counter Air</td>
</tr>
<tr>
<td>AMOD</td>
<td>Aegis MODernization</td>
<td>OA</td>
<td>Open Architecture</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
<td>OAET</td>
<td>Open Architecture Enterprise Team</td>
</tr>
<tr>
<td>ASCM</td>
<td>Anti-Ship Cruise Missile</td>
<td>OASM</td>
<td>Open Architecture System Management</td>
</tr>
<tr>
<td>ASROC</td>
<td>Anti-Submarine ROcket</td>
<td>P/S</td>
<td>Publish/Subscribe</td>
</tr>
<tr>
<td>BL</td>
<td>Baseline</td>
<td>PIDS</td>
<td>Prime Item Development Specification</td>
</tr>
<tr>
<td>BMD</td>
<td>Ballistic Missile Defense</td>
<td>PIM</td>
<td>Platform Independent Model</td>
</tr>
<tr>
<td>C2</td>
<td>Command and Control</td>
<td>PSEA</td>
<td>Platform System Engineering Agent</td>
</tr>
<tr>
<td>CCB</td>
<td>Configuration Control Board</td>
<td>PSM</td>
<td>Platform Specific Model</td>
</tr>
<tr>
<td>CEC</td>
<td>Cooperative Engagement Capability</td>
<td>Pub/Sub</td>
<td>Publish/Subscribe</td>
</tr>
<tr>
<td>CG</td>
<td>Guided Missile Cruisers</td>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>CIWS</td>
<td>Close In Weapon System</td>
<td>SAN</td>
<td>Storage Area Network</td>
</tr>
<tr>
<td>CM</td>
<td>Configuration Management</td>
<td>SBT</td>
<td>Sea-Based Terminal</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off-the-Shelf</td>
<td>SI/DA</td>
<td>System Integrator / Design Agent</td>
</tr>
<tr>
<td>CPCR</td>
<td>Computer Program Change Request</td>
<td>SM</td>
<td>Standard Missile</td>
</tr>
<tr>
<td>CR</td>
<td>COTS Refresh</td>
<td>SMP</td>
<td>Symmetric MultiProcessor</td>
</tr>
<tr>
<td>CSEA</td>
<td>Combat System Engineering Agent</td>
<td>SQT</td>
<td>System Qualification Test</td>
</tr>
<tr>
<td>CVN</td>
<td>Carrier Vessel Nuclear</td>
<td>SRS</td>
<td>System Requirements Specification</td>
</tr>
<tr>
<td>DDG</td>
<td>Guided Missile Destroyer</td>
<td>SSDD</td>
<td>System/Segment Design Document</td>
</tr>
<tr>
<td>DDS</td>
<td>Data Distribution Service</td>
<td>SSDS</td>
<td>Ship Self Defense System</td>
</tr>
<tr>
<td>DM</td>
<td>Data Model</td>
<td>SSR</td>
<td>Software Specification Review</td>
</tr>
<tr>
<td>DOORS</td>
<td>Dynamic Object-Oriented Requirements System</td>
<td>STM</td>
<td>System Track Manager</td>
</tr>
<tr>
<td>ESSM</td>
<td>Evolved Sea Sparrow Missile</td>
<td>SVC</td>
<td>Service</td>
</tr>
<tr>
<td>ETR</td>
<td>Engineering Technical Review</td>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>GCC</td>
<td>GNU Compiler</td>
<td>SysML</td>
<td>Systems Modeling Language</td>
</tr>
<tr>
<td>GFE</td>
<td>Government Furnished Equipment</td>
<td>T&amp;E</td>
<td>Test and Evaluation</td>
</tr>
<tr>
<td>HM&amp;E</td>
<td>Hull, Mechanical and Electrical</td>
<td>TADIL</td>
<td>Tactical Digital Information Link</td>
</tr>
<tr>
<td>HW</td>
<td>Hardware</td>
<td>TI</td>
<td>Technology Insertion</td>
</tr>
<tr>
<td>IAW</td>
<td>In Accordance With</td>
<td>TLAM</td>
<td>Tomahawk Land-Attack Missile</td>
</tr>
<tr>
<td>IDD</td>
<td>Interface Definition Document</td>
<td>TOR</td>
<td>Test Observation Report</td>
</tr>
<tr>
<td>IDS</td>
<td>Interface Design Specification</td>
<td>TS</td>
<td>Track Server</td>
</tr>
<tr>
<td>IMS</td>
<td>Integrated Master Schedule</td>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>IPO</td>
<td>Input/Output/Process</td>
<td>VLA</td>
<td>Vertical Launch ASROC</td>
</tr>
<tr>
<td>IR</td>
<td>Infrared</td>
<td>VLS</td>
<td>Vertical Launch System</td>
</tr>
<tr>
<td>JTM</td>
<td>Joint Track Management</td>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
<tr>
<td>KA</td>
<td>Kill Assessment</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>