An Activity-Based Methodology* (ABM) for Development and Analysis of Integrated DoD Architectures: “The Art of Architecture”

C2 Assessment & Tools, #077
June 2004

Steve Ring, MITRE Corporation, sring@mitre.org
Dave Nicholson, MITRE Corporation, dnichols@mitre.org
Jim Thilenius, MITRE Corporation, jethilen@mitre.org
Stanley Harris, Lockheed-Martin, stanley.harris@lmco.com

MITRE Approved for Public Release
Distribution Unlimited, Case #04-0351
©2004 The MITRE Corporation. All rights reserved

* Activity-Based Methodology is a concept developed by The MITRE Corporation and Lockheed-Martin, Copyright © 2003
Vendor Participation

**Popkin Software**
...System Architect v9.1.40 based on ABM
publically announced 5/18/2004

**Proforma Corporation**
...initial discussions held with Chief Methodologist
Brian James in March
...webx demo provided in May – 3-way OV-5 in-place

**Computas/Metis**
...Preliminary discussions held with
Chief Consultant Don Hodge 12/2003
Agenda

- Define integrated architectures
- Present Activity-Based Methodology
- Present ABM Architecture Description Specification Model – “ADSM”
- Show numerous integrated architecture analysis techniques and strategies
- Present steps to integrated Operational and System Architecture Descriptions- the “Art of Architecting”

- Present “dynamic” architecture descriptions transitioned from integrated “static” architectures
Before you can use architecture descriptions for any type of analysis purposes you must first have an architecture that is **Integrated, unambiguous, and consistent**.

**What’s an Integrated Architecture?**
- Based on DOD Architecture Framework (DoDAF)
  1. Integrated Operational and System views (via SV-5) within **single architecture** - AV-1, AV-2, OV-2, OV-3, OV-5, SV-1, and TV-1
     + OV-4 the forgotten product, key to DOTLMPF
  2. Integrated architectures between and among **multiple architectures**
- Joint Capabilities Integration and Development System Process (JCIDS)

**DoD Architecture Repository System (DARS) provides source of authoritative architecture**
- Populated with DoD architecture information built in accordance with the DoDAF
- Store legacy, draft, and approved architecture information developed by the Commands, Services and Agencies
- Enable sharing, exchange and reuse of architecture data
What Is Activity-Based Methodology?

- Consists of a tool-independent approach to developing fully integrated, unambiguous, and consistent DODAF views.

- Enables both
  - “As-Is” (now) architectures - all details known
  - “To-Be” (future) architectures - based on unknowns and abstract elements where not all details known
    - “To-Be” architectures must support “gap-analysis” to discover future unknown rules, patterns, practices, relationships, and requirements.

- Uses data centric approach for architecture element and product rendering
  - Supports cross-product relationships based on core set of architecture elements
  - Simplified “architecture specification model” of architecture elements and their associations/relationships based on DoDAF and not CADM.

- Captures sufficient representations of architectures models to transition to “dynamic” executable process models.
Methodology Based on Six Principles

OA and SA objects symmetrically aligned to each other

4 OA and 4 SA elements provide core foundation building blocks of Intergrated Architecture

Architecture Data Specification Model defines associations between sets of core entities

Core architecture data entered from single DoDAF products

Manual

Automation

Several DoDAF elements automatically formed from core entities

Several DODAF products automatically rendered
Symmetrically Aligned DoDAF Architecture Objects

Operational

- Entities
  - Zachman
    - Why
    - Strategy
    - Design
    - Strategy

- Relationships
  - Activity
  - Info
  - Ops Node
  - Need Line

- Attributes
  - Org
  - Knowledge Skills & Abilities
  - Process

System

- Entities
  - System Function
  - System Node
  - Interface
  - Data Exchange

- Relationships
  - Link
  - Transfer
  - Structure
  - Characteristic

- Attributes
  - Process
  - When
  - Performance

Core Objects

Generated Objects
Associations between Core Entities Forms Foundation of an Integrated Architecture

**Triple 3-way Associations of Core Entities**
Triple 3-way Associations of Core Entities

- Op Nodes
- Sys Nodes
- System

Op Activities
Sys Functions
Role
Org Unit

Op Node → Activity
Activity
System Function
System Node

Role
System
Org

Role
System
Integrated Architecture Represented as Architecture Data Specification Model – “ADSM”
Integrated Architecture Data Analysis

- Nodal Analysis: "WHERE"
- Functional Analysis: "HOW"
- Product Analysis: "WHAT"
- People, System and Training Analysis: "WHO"

Integrating OV -to- SV
Integrating SV -to- OV

Op Node
Activity
Function
Sys Node
Info
Role
System
Org Unit
Mapping ADSM to DOTMLPF
“Gap-Analysis” for “To-Be” Architectures

<table>
<thead>
<tr>
<th>Activities</th>
<th>Op Nodes</th>
<th>Roles</th>
<th>Sys Nodes</th>
<th>Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>Not Valid Must know Activities</td>
</tr>
<tr>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>?</td>
<td>Most likely “gap-analysis”</td>
</tr>
<tr>
<td>✔️</td>
<td>?</td>
<td>✔️</td>
<td>?</td>
<td>Possible “gap-analysis”</td>
</tr>
<tr>
<td>✔️</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>Least likely</td>
</tr>
</tbody>
</table>
Analysis of Integrated Architectures Between Other Architectures - “Seam Analysis”

ONodeA – Act1 – SF1 – SNodeA
ONodeA – Act1 – SF2 – SNodeA
ONodeA – Act2 – SF3 – SNodeA
ONodeA – Act2 – SF4 – SNodeA
ONodeA – Act3 – SF5 – SNodeA
ONodeA – Act4 – SF6 – SNodeA
ONodeA – Act4 – SF7 – SNodeA
ONodeA – Act5 – SF8 – SNodeA
Steps to an Integrated Systems Architecture

- **System Function**
  - SV-1
  - SV-4

- **Data**
  - Nodes
  - Data Entry

- **System Node**
  - SV-1
  - Nodes

- **Render System Data Exchanges**
  - Interface
  - Complete SV-1

- **Manual 3-way Associations**
  - Sys Nodes

- **Automation**
  - Auto form 3-way associations

- **Sys X**
  - "NodeA~SysX"

- **Function**
  - "Func1~SysX"

- **Manual 3-way Associations**
  - Sys Nodes

- **SV-7 Systems**
  - System

- **SV-6**
  - Generate SV-6
Chained *Leaf Activities* Produce Candidate Activity Thread (Scenario) Models Of Sequenced Actions

- External Activities/ Nodes
- Lowest activities in node tree chain
- No further decomposition
- Leaf activities signified by *Blue Boxes*
- OV-6 generation
- Information Exchanges and Need Lines built only from Leaf Activities
- Use Cases/System Functions

**Ordering of Leaf Activities Follows OV-5 Information Flow**

External Input

- **Act 11** “Wash DC”
- **Act 13** “Chicago”
- **Act 14** “KC”
- **Act 121** “Boston”
- **Act 122** “LA”

Mission Request

- **Act 13** “Chicago”
- **Act 121** “Boston”

Mission Schedule

- **Act 14** “KC”
- **Act 122** “LA”

Mission Orders

- **Act 21** “Cleveland”
- **Act 221** “Miami”
- **Act 222** “New York”

Mission Data

- **Act 13** “Chicago”
- **Act 14** “KC”

Mission Plans

- **Act 21** “Cleveland”
- **Act 221** “Miami”
- **Act 222** “New York”

Task Request

- **Act 14** “KC”
- **Act 21** “Cleveland”

Task Orders

- **Act 221** “Miami”
- **Act 222** “New York”

External Output

- **Act 221** “Miami”
- **Act 222** “New York”
Data Cleaning: 1st Step in Architecture Development

Turning *Unfriendly, Dirty* Data into *Friendly, Clean* Data*

<table>
<thead>
<tr>
<th>Air Mission</th>
<th>Fort Hood</th>
<th>AT&amp;T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air Mission</td>
<td>Fort Hood</td>
<td>AT&amp;T</td>
</tr>
<tr>
<td>Air Mission</td>
<td>Fort Hood</td>
<td>AT&amp;T</td>
</tr>
<tr>
<td>Air Mission</td>
<td>Fort Hood</td>
<td>AT&amp;T</td>
</tr>
<tr>
<td>Air Mission</td>
<td>Fort Hood</td>
<td>AT&amp;T</td>
</tr>
<tr>
<td>Air Mission</td>
<td>Fort Hood</td>
<td>AT&amp;T</td>
</tr>
<tr>
<td>Air Mission</td>
<td>Fort Hood</td>
<td>AT&amp;T</td>
</tr>
</tbody>
</table>

**Synonyms**

different names mean same things
(‘location’ and ‘loc’, ‘Target’ and ‘tgt’)

**Homonyms**
same name means different things (‘mission’, ‘tank’, ‘mustang’)

*Cleaning a manual operation*
Architecture Data Mining with Extended OV-3 & SV-6

**What is Required**

- Role
- Act
- Info
- ActN
- ... (信息交换)

**What is Delivered**

- Org
- Sys
- SFnc
- Data
- SysN
- ... (系统数据交换)

**OV-3 Information Exchanges**

- SV-5
- Act:SFnc

**SV-6 System Data Exchanges**

- Automated information exchanges (from OV-3) implemented in Systems
OV-3 & SV-6 Relationship Analysis

What Role Uses To Performs an Operation

Systems Supporting Activities

Impact of losing System Node

Impact of losing System

OA-SA Requirements

Automated IEX implemented in Systems

What if? If What?
Transition to Executable Architectures

Reconfigure three DODAF Views into Process & Resource Views

Process View “PV”

Resource View “RV”

DoD Architecture Framework 1.0
Transformation to Dynamic Process Models

Integrated Architectures

Architecture Tools

OV-5 Leaf Activities
Rearranged to Match OV-2

OV-3 Info Exch

OV-4 Org/Roles

Mapping Static
-to-
Dynamic

Generated OV-6a

Executable Architectures

Executable Modeling Tools
What Are Executable Architectures?

* Static Operational Models only show that Activities "must be capable of" producing and consuming Information
  - No details on event sequencing
  - No details on how or what conditions information is produced/consumed
  - No details on producers/consumers themselves or other resources used

* Dynamic (over time) Executable Architecture Models go beyond "must be capable of" – "WHEN"
  - Defines precise sequential/concurrent event model
  - Defines precisely under what conditions Information is produced/consumed
  - Defines details on producers/consumers (number and process ordering) and other resources (when [not] available)

* Dynamic model of Activities and their event sequencing performed at Operational Nodes by Roles (within Organizations) using Resources (Systems) to produce and consume Information
Summary

DoDAF + Architecture Tools + Activity-Based Methodology

Guidance + Compliant Tools and Methodology Render Integrated Architectures

Integrated Architectures + Simulation Tools and Scenarios (context) Render Executable Architectures

Integrated Architectures + Executable Architectures + Analytical Tools and Methods Render Quantitative Actionable Information

Funding decisions, acquisitions, system engineering, investment strategy,....