IMPROVING PROJECT MANAGEMENT USING FORMAL MODELS AND ARCHITECTURES

Theodore Kahn
theodore.e.kahn@nasa.gov

Ian Sturken
ian.sturken@nasa.gov

Project Management Challenge
February 9-10, 2011
Problem Statement

Project information is stored in various documents, spreadsheets and systems with little consistency and/or formal structure.

A lack of common understanding of a project’s organizations, roles, objectives, behaviors and constraints.
Agenda

- Problem statement, objectives, agenda
- Theory of:
  - Enterprise and Business Architecture
  - Formal modeling
- Applying EA and Modeling to Project Management
- Case studies:
  - Ares development
  - Ames process modeling
  - MODEAR
  - Flight Readiness System
- Making Modeling Work for You Today
- Future Trends and Closing Remarks
- Q&A
Four Modeling Perspectives
SE Standards, Languages, AFs
Enterprise Architecture
What is the Scope of an Enterprise Architecture?

Several ideas are in common use:

- An accounting of an organization’s IT artifacts and their application to lines of business. *(Lists of IT things.)*

- The relationships and behaviors of an organization’s IT artifacts and their application to lines of business. *(Lists and Life-cycle of IT things.)*

- An accounting of an organization’s meaningful artifacts and their application to lines of business. *(Lists of “all” things.)*

- The relationships and behaviors of an organization’s meaningful artifacts and their application to lines of business. *(Lists and Life-cycle of “all” things.)*
FEA and DoDAF EA Definition

- A strategic information asset base,
- which defines the mission,
- the information necessary to perform the mission,
- the technologies necessary to perform the mission, and
- the transitional processes for implementing new technologies in response to changing mission needs.
- EA includes a baseline architecture, a target architecture, and a sequencing plan.
How did we use an Enterprise Architecture?

- To organize the information about our processes, products, people and systems
- To relate these entities to one another
- To provide different diagrams and reports of the information suitable to each of the stakeholders in the project
- To export information to other tools for analysis and simulations
Benefits of Enterprise Architecture

- **Enterprise Architecture**
- **Descriptive**: Promote a common understanding of the enterprise
- **Analytical**: Simulate the architecture and produce artifacts used to execute enterprise processes
- **Gaps**: Determine needs for resources, processes and data
- **Constraints**: Determine critical resources, processes and data
- **Objective**: Optimize resource consumption and production
# Zachman Framework

<table>
<thead>
<tr>
<th>Scope</th>
<th>What (Data)</th>
<th>How (Function or Process)</th>
<th>Who (People)</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of things important to business</td>
<td>Function Hierarchy</td>
<td>List of Organizations</td>
<td></td>
</tr>
<tr>
<td>Business Model</td>
<td>Conceptual Data Model</td>
<td>Process Model</td>
<td>Org to Function mapping (roles)</td>
</tr>
<tr>
<td>System Model</td>
<td>Logical Data Model</td>
<td>Use Case</td>
<td>Process to Role Matrix</td>
</tr>
<tr>
<td>Technology Model</td>
<td>Physical Data Model</td>
<td>Activity Diagram</td>
<td>Roles/Access Matrix</td>
</tr>
<tr>
<td>Detailed Design</td>
<td>Technology Specific</td>
<td>Technology Specific</td>
<td>Technology Specific</td>
</tr>
</tbody>
</table>

## Technology Agnostic - Understand The Business

- ConOps Requirements
- System Requirements
- Design Specifications

## Technology Specific – Specify and Design the Systems to Support The Business
DoD 2.0 Framework

- **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

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.
Which EAF do I Use?

- **Zachman**
  - Easier to grasp and get started with. Can start with lists of “things” and start relating these to other parts of the business
  - Hierarchical in nature, provides good mechanism for abstracting levels of detail from executive to engineer
  - More IT centric

- **DoDAF**
  - More prescriptive in nature – specific products to fill different purposes
  - Separate different viewpoints – business processes from systems that support them
  - Supported by many tools
  - Has a modeling language specifically designed for it: UPDM
  - General purpose

- **Create your own**
  - If you don’t use a standard framework – you will create your own mechanisms for organizing and relating information in your models!
Use Standard Architecture Framework, Model or Both?

- **Architecture Frameworks:**
  - Can range from simple (lists) to complex
  - Useful for providing an outline of what information to gather and how to organize that information
  - Can customize this outline to fit your needs
  - Can be used to compare different systems from different vendors
  - Can be used to study “as-is” states to “to-be” states
  - Can leverage modeling languages such as UML, SysML, and Archimate

- **Modeling Standards**
  - Can range from simple to complex
  - Quick to build a few diagrams
  - For larger projects will need to organize model
  - Formal language/annotations used

- **Both**
  - Provides guidance on what modeling artifacts you will need and how to organize them according to a standard framework
What is a Model?

An Abstraction of the Physical World Around Us

- An electrical schematic of a radio
- An economic model
- A mathematical model
- A model student
- A non working model airplane
- A written description of a pencil
- A diagram
- A spreadsheet
- Music
- Art
- Natural languages
## Sample of Modeling Languages

<table>
<thead>
<tr>
<th>Language</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDEFx</td>
<td>Business</td>
</tr>
<tr>
<td>UML</td>
<td>Software</td>
</tr>
<tr>
<td>BPN</td>
<td>Software</td>
</tr>
<tr>
<td>AADL</td>
<td>Hardware, software, realtime (avionics, aerospace, automotive, and robotics).</td>
</tr>
<tr>
<td>Simulink</td>
<td>Simulation and analysis of multidomain dynamic systems</td>
</tr>
<tr>
<td>Archimate</td>
<td>Business</td>
</tr>
<tr>
<td>SysML</td>
<td>Systems of systems</td>
</tr>
</tbody>
</table>
Modeling Language Attributes

Formality

More formality = less ambiguity, more accuracy

Abstraction

Less abstraction = more detail, more precision
Abstraction Levels

La Joconde

Femme au Chapeau Orné
What is a Formal Model?

The degree to which the model adheres to:

• Well defined **semantics**: model components have precise interpretations.

• Well defined **grammar**: model components can only be related using precise structural rules.
SysML Semantics
SysML Requirements Relationships
SysML Diagram Taxonomy

Source: OMGSpecification
Applying EA & Modeling to PM
Projects are now modeled using spreadsheets, diagrams and documents to represent different parts (components) of the project.

A formal model does not change this. Instead, your project components must now be represented using formal grammar and semantics. And, if you are using a standardized framework, your project follows a well known architecture.
System Engineering and Project Management

## Review Entrance Criteria
(NASA Systems Engineering Handbook)

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Artifacts</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Concept Review</td>
<td>System Goals And Objectives</td>
</tr>
<tr>
<td></td>
<td>Concept of Operations</td>
</tr>
<tr>
<td>System Requirements Review</td>
<td>System Requirements</td>
</tr>
<tr>
<td></td>
<td>System Functionality Description</td>
</tr>
<tr>
<td></td>
<td>Concept of Operations</td>
</tr>
<tr>
<td></td>
<td>Preliminary System Requirements</td>
</tr>
<tr>
<td>Preliminary Design Review</td>
<td>Preliminary subsystem design Specs</td>
</tr>
<tr>
<td></td>
<td>Operational Concept</td>
</tr>
<tr>
<td></td>
<td>Interface Control Documents</td>
</tr>
<tr>
<td></td>
<td>Requirements Traceability Matrix</td>
</tr>
</tbody>
</table>

These can all be described in one model!
# Building ConOps from Model

<table>
<thead>
<tr>
<th>Conops Section</th>
<th>DoDAF product</th>
<th>SYSML Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenarios</td>
<td>OV-5 Activity Diagram</td>
<td>Use Case Diagram, Activity Diagram</td>
</tr>
<tr>
<td>Conceptual Overview</td>
<td>OV-1 High Level Concept</td>
<td>Block Definition Diagram</td>
</tr>
<tr>
<td>Event sequence</td>
<td>OV-6c</td>
<td>Sequence Diagram</td>
</tr>
<tr>
<td>Connectivity Architecture</td>
<td>OV-2 Node Connectivity Diagram, OV-3 Information Exchanges, SV-1 System Interface, SV-2 System Communication</td>
<td>Block Definition Diagram</td>
</tr>
<tr>
<td>Glossary</td>
<td>AV-2 Integrated Dictionary</td>
<td>Block Definition Diagram</td>
</tr>
</tbody>
</table>
Technical Decision Analysis
(Trade Analyses)
Formal Modeling and Six Sigma
(Complementary Technologies.)

<table>
<thead>
<tr>
<th></th>
<th>Six Sigma</th>
<th>Formal Models</th>
<th>Both Together</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methodology</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Formal Data Semantics &amp; Grammar</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Data Persistence</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Case Studies
MOD Flight Production Process Re-engineering

- **Goal:** MOD needs to transform into an agile organization to be able to quickly meet needs and opportunities that arise in the next decade.

- **Challenge:** Currently, most information about how we conduct business is housed in different documents, spreadsheets, systems and other repositories. It is difficult to gain a comprehensive, integrated, common view of the way we conduct business and what the impact of changes are on our people, processes and systems.

- **Approach:** An enterprise architecture provides a framework that will allow us to organize information about our people, processes and systems in an organized, structured and integrated manner.

- **Benefits:** An organization that can quickly assess the impact of external events saving $$$$$ and reducing risk.
MOD EA Repository

---

![MODEAR Interface](image)

**Browse Process Instances**

<table>
<thead>
<tr>
<th>Op Mode</th>
<th>Number</th>
<th>Process Instance Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA6</td>
<td>2.4.2.1</td>
<td>Integrate Flight Software Development and Issues Resolution - Flight Design Post MRD Instance</td>
</tr>
<tr>
<td>DA6</td>
<td>2.4.2.2</td>
<td>Integrate Flight Software Development and Issues Resolution - Flight Design Uplink Instance</td>
</tr>
<tr>
<td>DA6</td>
<td>2.4.2.3</td>
<td>Integrate Flight Software Development and Issues Resolution - Flight Design DDLIU Instance</td>
</tr>
<tr>
<td>DA6</td>
<td>2.4.2.1</td>
<td>Provide Flight Software Change Coordination - Instance 1</td>
</tr>
</tbody>
</table>

**Source Process Instances (and exchanged products)**

4.2.1.2_52_1 - Update Trajectory and 1-Load Design Instance - Step 2

--- Flight Design Flight Software Discrepancy Report (DM)

**Selected Process Instance**

2.4.2_2 - Integrate Flight Software Development and Issues Resolution - Flight Design Uplink Instance

Function: 2.4.2 - Integrate Flight Software Development and Issues Resolution

Operational Node: DA6 - Technical Integration and Production Control Office

Description:

**Recipient Process Instances (and exchanged products)**

4.2.1.3_51_1 - Generate Day-of-launch 1-Load Update Instance - Step 1

--- Flight Design Flight Software Discrepancy Resolution Report (DM)
Use Architecture Information for Several Purposes

Performers

Mission EA Repository ➔ System Architect ➔ DoDAF Diagrams

Mission EA Repository ➔ Discrete Event Simulator ➔ Operations timeline, critical path determination

Tabular reports
DoDAF Connectivity Diagram (OV-2)
Flight Readiness System

- Goal: Develop a new system to support certification of flight readiness for Cx
- Challenge: How do we specify the components of our system with varying levels of detail while maintaining consistency throughout
- Approach: Use DoDAF/UPDM to describe the ‘as-is’ process and systems for shuttle. Then design a new set of processes and supporting systems for Constellation as a ‘to-be’.
- Benefits: Information is organized and represented consistently with various levels of detail appropriate to different stakeholders
As-Is and To-Be Templates

As-Is

To-Be
Flight Readiness “As-Is” OV-5
Flight Readiness “To-Be” OV-5
Flight Readiness System Model
Modeling Ares Development

Problem Definition:
1. Large amount of data...
2. maintained in three separate artifacts: document, spreadsheet and diagram...
3. to meet different stakeholder needs.

Lead to the following concerns:
1. Time consuming and error prone to modify data as Ares program changes.
2. Not easy to meet new stakeholder needs.
Ares Model Architecture
(UPDM)
Ares Document Attributes
Ares Document Relations
UPDM OV7 Diagram
(Structure)
UPDM OV2 Diagram
(Behavior)
UPDM Exchange Types
Ames and EBA
EBA Architecture
Stakeholder Taxonomy
Use Cases
SysML Activity Diagram
Constraints

Documentation of Constraint Scan Items

HTML

(2 wks * (.5 person) * 27 inventories * 2 people) * 1.5 overhead = 300hrs

Basically, this says that 2 people are working half time for 2 weeks which = 2 person-weeks = 80 hrs.

1.5 (50%) overhead represents starting/stoping inefficiencies as well as working with property custodians having little experience.
Applying Constraints
Making Modeling Work for You Today

Practical Information for achieving quick ROI
Modeling is an Engineering Task

- Approach it systematically
- Know what resources you will need
- Define milestones, a roadmap
- Be pragmatic
What Makes a Good Formal Model?

- Model those aspects of the project required to answer stakeholder questions, and no more.

- Model the degree of precision required to answer stakeholder questions, and no more.

- Models must always be accurate.
Five Knowledge Domains
(Why modeling is hard)

- Need to know Architectural Framework
- Need to know modeling tool
- Need to know project/system domain
- Need to understand stakeholder questions
- Need to understand modeling semantics

Impediments to Modeling

58
Four Modeling Steps
(Do only one at a time.)

Questions & Project knowledge

translate to

Architectural Framework Knowledge

apply to

Modeling Language knowledge

apply

Tool knowledge
Modeling Tools Encompass Two Areas
(Do only one at a time.)

- Database program
- Drawing program
Think Small, Think Focused
(Get ROI in Weeks!)

- What questions should your model answer?
- Select a modeling language.
- Determine the architecture.
- Select a tool.
You’re in Front of your Computer
(Now what do you do?)

- Your tool is running
- You created a new SysML project
- And...
- You create packages to organize your project
A “Template” SysML Model
Extending SysML

- Use English to document each entity.
- Use diagram notes to highlight explain diagram elements.
- Use SysML Profiles to extend SysML semantics to meet your own domain specific needs.
Modeling Tips

- What if you don’t know something?
  - Make your best guess, it’s easy to change.
- What should go on a diagram?
  - It should tell a story, answer a question, address a specific stakeholder need.
- Look to see how a set of diagrams might meet a stakeholder’s need in some specific area.
- Model only those elements for which you know there is a value.
Culture Issues
(Modeling is about sharing information.)

- Some people do not necessarily want to share their information
  - Job security
  - They don’t know the information, and perhaps reluctant to say so.
  - It’s time-consuming to get the information, what’s in it for them?
- Some people like to work independently
Modeling Summary

- Think small, know what questions your model should answer.
- Keep the architecture simple.
- Learn your modeling language semantics.
- Pro actively manage the modeling task:
  - Engineering effort
  - Cultural issues
Future Trends and Closing Remarks
Future Trends

- Fully defined semantics
- Prescriptive methodologies
- Improved tooling
- Analytical integration
- EA Frameworks are adding behavior and project management representations
Backup
DODAF 2.0

Capability Viewpoint
- Vision
- Taxonomy
- Phasing
- Dependencies
- Organizational Mapping
- Activities Mapping
- Services Mapping

Project Viewpoint
- Portfolio
- Timelines
- Capabilities Mapping
Projects are defined and change mostly due to external events: Continuously.