Improving Project Management Using Formal Models and Architectures

Ian Sturken
Ted Kahn
Project Management Challenge
February 9-10, 2011
Problem Statement

Today, project information is stored in unstructured documents

- Information that is inconsistent
- Information that has no traceability
Objectives

You will have a general understanding and knowledge of:

• A theoretical background of enterprise architecture, business architecture and formal modeling
• Actual projects using formal modeling and enterprise architectures
• The application of formal modeling to Project Management
• Specific do’s and don’ts for making formal modeling work for you today: short term ROI
Agenda

• Problem statement, objectives, agenda
• Theory of:
  – Enterprise and Business Architecture
  – Formal modeling
• Case studies:
  – Ares development
  – Ames process modeling
  – MODEAR
  – Flight Readiness System
• Applying Architecture Frameworks and Modeling Languages to Project Management
• **Making Modeling Work for You Today**
• Future Trends and Closing Remarks
• Q&A
Four Modeling Perspectives

1. Organization Perspective
   Enterprise Architecture

2. Line of Business Perspective
   Program
   Enterprise Business Architecture
   Project

3. System Perspective
   Technical Architecture

4. Engineering Perspective
   Chemical
   Economic
   Orbit
SE Standards, Languages, AFs

Information Expression
Provides semantic and grammatical guidance (having varying degrees of formality) for expressing entities, their attributes, relationships and behaviors.

Describing Architectures
- ISO/IEC 19793
- ISO/IEC 10746
- ISO/IEC 42010 (ANSI/IEEE 1471)
- E-Gov FEA

Enterprise Architecture Frameworks
- Zachman
- TOGAF 9
- CCSDS RASDS
- DoDAF V2.0
- FEAF

Information Organization
Provides guidance regarding what entities are important, how they should be described and the types of relationships and behaviors they might have.

Modeling Languages
- ISO/IEC 62264
- The Open Group ArchiMate
- OMG UML
- OMG BPMN
- ISO/IEC 19501
- OMG SysML
- SAE AADL
- USAF ICAM IDEFx
- ISO/IEC OPM (emerging)

Enterprise Architectures
- NASA
- Constellation
- DoDAF V2.0

Formal Enterprise Architectures
- OMG UPDM
ENTERPRISE ARCHITECTURE
What is an Enterprise Architecture?

An Abstraction of the Physical World Around Us

• 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 things.)

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

Are DoDAF, FEA, and Zachman Enterprise Architectures?
Enterprise Architectures....

- **Enterprise architectures** prescribe what entities or “things” in a system or organization are important, how the information about these entities should be characterized and the relations they should have among themselves.
- **Systems** can be an enterprise, organization or system of systems.
- **Entities** can be people, processes, products, resources, etc.
Benefits of Enterprise Architecture

Enterprise Architecture

«analytical»
Simulate the architecture and produce artifacts used to execute enterprise processes

«descriptive»
Promote a common understanding of the enterprise

«constraints»
Determine critical resources, processes and data

«objective»
Optimize resource consumption and production

«gaps»
Determine needs for resources, processes and data
# Zachman Framework

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

DoD Viewpoints
Which AF 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
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 SYSML, IDEF, UML, UPDM

• 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
MODELS, Formal Models and SysML

MODELING
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
Model 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 connected using precise structural rules.
SysML Semantics

DEA Tool Functional Use Cases

- **Query DEA Database**
  - Id = "2.1.1"
  - Text = "The tool shall provide users the ability to query the database subject to appropriate permissions."

- **Create, Update, Delete DEA’s**
  - Id = "2.1.2"
  - Text = "The tool shall provide users the ability to create, update and delete DEA’s."

- **DEA Lifecycle**
  - Id = "2.1.3"
  - Text = "The tool shall support a formal workflow lifecycle process as part of the update/change features."

- **DEA Electronic Signature**
  - Id = "2.1.4"
  - Text = "The tool shall provide an electronic signature."

- **Print DEA**
  - Id = "2.2"
  - Text = "The tool shall be able to print a DEA in RTF format."

- **Lists of DEA’s**
SysML Requirements Relationships

SysML Requirements Dependency Relationships

- Any behavioral diagram.
- <<refine>>
- <<refine>>
- <<verify>>
- <<allocate>>
- <<satisfy>>
- <<requirement>>
  - Id = ""
  - Text = ""
- <<trace>>
- <<block>>
- External Information

Any entity executing the activity: person, org, machine, software, etc.
SysML Diagram Taxonomy

Source: OMG Specification
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. This will allow us to:
  – Define, develop, validate and execute our missions with a common understanding of how our people, processes and systems will interact with one another
  – Run models and simulations of our business to validate our processes and systems and find areas where efficiencies can be achieved
  – Determine our net-readiness and interoperability capabilities
  – Find gaps and overlaps between our operations needs and system capabilities

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

### MODEAR

#### Browse Process Instances

Filter Number/OpNode/Title: [Clear filter]

<table>
<thead>
<tr>
<th>Op Node</th>
<th>Number</th>
<th>Process Instance Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA5</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 DOLILU Instance</td>
</tr>
<tr>
<td>DA6</td>
<td>2.4.3.1</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 I-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 I-Load Update Instance - Step 1
   -> Flight Design Flight Software Discrepancy Resolution Report (DM)
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 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
Integrated Architecture

Activity Diagram

Connectivity Diagram

Exchange Report

Information Element

Data Model
APPLYING EA & MODELING TO PM
Modeling and 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.
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>
An informal diagram from SE handbook – Convert to AF or SYSML?
What is the effect of changing this?

On this slide we can show that a change in one part of the model would allow you to determine what changed in other parts of the model.
Technical Decision Analysis
(Trade Analyses)
Portfolio Management

Portfolio Management

Connectivity Diagram (ov-2)

System Diagram (SV-1)

Activity Diagram (0v-5)
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>
Culture Issues

• People do not necessarily want to share their information
  – Job security
  – They don’t know the information, and perhaps are reluctant to say so.
  – Its time-consuming to get the information, what’s in it for them?

• People like to work independently
Modeling Summary

• Think small, think of specific questions your model should answer.
• Don’t use a formal architecture.
• Do learn the semantics of your modeling language.
• Do take the time to learn your tool.
• Do pro actively manage the modeling task.
Practical Information for achieving quick ROI

MAKING MODELING WORK FOR YOU TODAY
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 Enterprise Framework
- Need to know stakeholder questions
- Need to understand modeling semantics
- Need to know modeling tool
- Need to know project/system domain

Impediments to Modeling

- Need to understand modeling semantics
- Need to know Enterprise Framework
- Need to know modeling tool
- Need to know project/system domain

Need to understand stakeholder questions
Think Small, Think Focused  
(Get ROI in Weeks!)

• Think hard about stakeholders and the questions they need answered. (You are a stakeholder.)
• Determine whether you need to utilize and architecture framework to satisfy your stakeholders
• Do not get hung up on tools, useful comparisons can be very difficult to make
• Use SysML, unless you have a compelling reason not to. And learn the relevant parts.
Modeling Tools are Hard
(In fact, they are two tools in one.)

- Database program
- Drawing program

Always make sure you know which action you are taking: database or drawing.
Two Modeling Steps

1. Project knowledge
2. SysML knowledge
3. Tool knowledge

Translating or applying, never both at the same time.
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

- Constellation
  - Actors
    - Cap Comm
    - President
    - Program Director
    - Actor Taxonomy
  - Artifacts
    - Information Artifacts
      - Problem Report
      - Information Artifact Taxonomy
    - Physical Artifacts
      - Physical Artifacts Taxonomy
  - Behavior
    - Launch Activities
    - Launch Activities
  - Functional Relationships
    - Mate
    - Nested Flow

- Requirements
  - Business Requirements
    - 1 Go To Mars
    - 2 Go to Moon
    - 3 Go to ISS
  - Business Requirements Diagram
  - Functional Requirements
    - 4 ISS Rocket
    - Functional Requirements Diagram
  - Non Functional Requirements
    - 5 Cost < $10B
    - 6 Time < 10 years
    - Non Functional REquirements Diagram
  - Structural Relationships
    - First Stage
    - Second Stage
    - Rocket Product Brakedown Structure
  - Use Cases
    - Build Capsule
    - Build Rocket
    - But Capsule on Rocket
    - Launch Rocket
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, its 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.
  – Its time-consuming to get the information, what’s in it for them?

• Some people like to work independently
Modeling Summary

• Think small, think of specific questions your model should answer.
• Don’t use a formal architecture.
• Do learn the semantics of your modeling language.
• Do take the time to learn your tool.
• Do pro actively manage the modeling task.
FUTURE TRENDS AND CLOSING REMARKS
Future Trends

- Fully define semantics
- Prescriptive methodologies
- Improved tooling
- Analytical integration
Projects are defined and change mostly due to external events: Continuously.

1. Represented by...

2. ...which encompasses project data...

3. ...used for creating SysML model...

4. ...and also used for aligning project schedule, scope and resources...

5. ...producing strategic PM artifacts...

...leading to better estimates for schedule, scope and resources...

...maintaining a consistent, feasible project and a refined model...

Formal Model = Formal semantic relationships + consistent representations
⇒ improved common understandings & decisions.

...providing consistent operational information used by all stakeholders...

Text Docs  Spreadsheets  Diagrams