Guidelines for
Model-Based Architecting and
Software (System) Engineering
(MBASE)

Marwan Abi-Antoun, USC-CSE

marwan@sunet.usc.edu

MBASE System Definition
Elements

- Operational Concept Description (OCD)
- System and Software Requirements Definition (SSRD)
- System and Software Architecture Description (SSAD)
- Life Cycle Plan (LCP)
- Feasibility Rationale Description (FRD)
- Risk-driven prototypes
Objectives

- No confusion on the intent and the semantics of the documentation
- Tight integration
- Little redundancy
- Strong traceability and conceptual Integrity
- Compatible with the WinWin Spiral Model
- Compatible with the Unified Modeling Language (UML)

Challenges

- Adopt a Model-based approach as opposed to a document-centric one
- Guidelines applicable for:
  - General Practice, across a wide variety of projects
  - CS 577a/b
    - enough guidance: a student body consisting of a mix of industry professionals and new graduate students with little industrial experience
    - avoid overkill on small-to-medium projects
Tight Integration

• Each system definition element presents one (or part of) "view", depending on intended audience
• No particular sequential order
• Architecture Description split in OCD, SSRD, SSAD
  - OCD {Domain Description, System Analysis}
  - SSRD {System Design}
  - SSAD {System Analysis, System Design}
Strong Traceability and Conceptual Integrity

- As a result of tight integration
- Start with "previous results and refine
- As opposed to try to retro-fit traceability after the fact, using traceability matrices, to determine what “came” from where

Dependency Chart (Example)
Immediate Predecessors and Successors

Solution to unwieldy dependency graph: show the immediate predecessors and successors; helpful when having multiple team members working concurrently on separate parts.

Use of WinWin Spiral Model

- Specific anchor points
- Specific completion criteria at each anchor point
- Documents correspond to parts of the Spiral
  - Establish Next Level Objectives
  - Evaluate Alternatives
  - Resolve risks,
  - ...
Refinement/Iteration Approach

- Within a given phase
  - e.g., Statement of Purpose → System Definition
  - e.g., Context Diagram → System Block Diagram → System Design View

- Across consecutive phases
  - e.g., SSAD → Detailed Design
  - e.g., LCP → Detailed Construction Plan
  - e.g., Scenarios → Test Cases

Common Pitfalls

- Spiraling out of control
- System Definition Elements "one iteration ahead" of the remaining ones
  - System definition elements are developed concurrently by different team members
Compatibility with the UML Language

• Provided UML guidelines where applicable
• Generated some MBASE-specific stereotypes for use with Rational Rose to generate various models, e.g., Behavior Classification

Guidelines for the
Life Cycle Objectives (LCO)

and the
Life Cycle Architecture (LCA)

deliverables for
Model-Based Architecting and Software Engineering (MBASE)

Inception and Elaboration
  • Operational Concept Description (OCD)
  • System and Software Requirements Definition (SSRD)
  • System and Software Architecture Description (SSAD)
  • Life Cycle Plan (LCP)
  • Feasibility Rationale Description (FRD)
MBASE Guidelines

- General Guidelines
- System Definition Element Guidelines

General Guidelines

- Process Guidelines
- Tool Guidelines
- Formatting Guidelines
- Overall Completion Criteria
  - Life Cycle Objectives (LCO)
  - Life Cycle Architecture (LCA)
- ...
System Definition Element Guidelines

• Purpose
• Completion Criteria
  – Life Cycle Objectives (LCO)
  – Life Cycle Architecture (LCA)
• Intended Audience
• Participants
• High-Level Dependencies
• Tool Support

Operational Concept Description (OCD)

Purpose

Completion Criteria

Intended Audience

Participants

High-Level Dependencies

Tool Support
Section-Level Guidelines

- Description and Rationale
  - LCO/LCA Differences (if applicable)
- Dependencies with other sections
  - [Consistent with DDD x.x.x]
- Detailed Guidelines
  - Best Practices (if applicable)
  - Techniques (if applicable)
  - Tool Support (if applicable)
  - Examples (if applicable)

Section-Level Guidelines (continued)

- Common Pitfalls
  - Counter-examples taken from actual mistakes
  - Use as Checklist items for determining overall quality
Guidelines Revision Process

- Yearly release (Corresponds with yearly cycle of Digital Library projects)
- Major update this year (Version 3)
  - Version 3.1: Students used up till LCO ARB
  - Version 3.2: Revision Following the LCO ARB that students used for LCA
  - Version 3.3: Revision Following LCA ARB that students use for rebaselining the LCA packages before Construction
Results

- A noticeably higher quality of the LCO and LCA packages, compared to previous years
- No significant increase in the average number of pages of each package
- Satisfaction of the students as demonstrated in their individual critiques:
  - "The documents almost write themselves"
  - "Wish we had updated version earlier"

LCO/LCA Spec Pages

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture Teams</td>
<td>15</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>LCO Spec Pages</td>
<td>160</td>
<td>110</td>
<td>114</td>
</tr>
<tr>
<td>LCA Spec Pages</td>
<td>230</td>
<td>154</td>
<td>167</td>
</tr>
</tbody>
</table>
Sources and Resources

- Commercial/DoD standards
  - EIA/IEEE J-STD-016
  - IEEE/EIA 12207.1-1997
  - IEEE Software Engineering Standards Collection
  - MIL-STD-498
- Commercial Best Practices
  - Lucent/AT&T, Software Architecture Validation
  - Rational Corp., Rational Unified Process

Questions