Integrating UPDM with SysML and UML on a DoD Acquisition Program

OMG UAF & MBSE Information Day
March 23, 2015

Paul Vaughan
NGIS Tech Fellow
Chief Engineer/Architect
DoD and ISR Acquisition Programs

- DoD and ISR programs typically require extensive and detailed traceability
  - Requirements flowdown and derivation down to software item/hardware item (SI/HI)
  - Functional flow block diagrams (activity diagrams/sequence diagrams/internal block diagrams) down to unit level
  - Requirements allocated to architectural elements (physical baseline) and system functions (functional baseline), including performance requirements
  - Contractor’s functional baseline, verification environments, and operations concepts are traceable to and consistent with the Government’s CONOPS
  - Test & evaluation planning is traceable to critical design, correlating test objectives, test environments, and test resources to allocated requirements

- Most programs require diagrams compliant with the DoD Architecture Framework (DoDAF)
  - Generally the Government doesn’t specify to what level this applies: usually interpreted as at the top level of the contract scope (system or segment level)

- The Systems Engineering team is responsible for the development of requirements, definition of the functions, and creation of the architectural design that provide the content for these diagrams

- The Software/Hardware teams are responsible for detailed designs consistent with higher level designs and that are fully traceable to Government requirements and CONOPS

- An objective of Model-Based methods is to do all that (and more) in a single integrated model
Stakeholder Requirements
Definition
• Identify users and stakeholders
• Define needs
• Capture source requirements
• Initialize requirements database
• Establish the CONOPS
• Generate System Requirements Document

Requirements Analysis
• Define systems capabilities and performance objectives
• Define, derive, and refine functional/performance requirements
• Define other non-functional requirements
• Develop specification trees and specifications
• Allocate requirements and establish traceability
• Generate a system specification

Architectural Design
• Define selection criteria
• Define/refine system element alternatives
• Synthesize multiple system architectures
• Analyze and select preferred system architecture/element solution
• Model, simulate, and prototype system architectures
• Define, refine and integrate system physical configuration
• Implement requirements and design feedback loops

Operational Context is provided by System Spec and CONOPS from Government

Operational Context in Model (User requirements and OV-5b)
Systems Engineering Provides the System Design Model, Including Traceability to OpsCon

**Stakeholder Requirements Definition**
- Identify users and stakeholders
- Define needs
- Capture source requirements
- Initialize requirements database
- Establish the CONOPS
- Generate System Requirements Document

**Requirements Analysis**
- Define systems capabilities and performance objectives
- Define, derive, and refine functional/performance requirements
- Define other non-functional requirements
- Develop specification trees and specifications
- Allocate requirements and establish traceability
- Generate a system specification

**Architectural Design**
- Define selection criteria
- Define/refine system element alternatives
- Synthesize multiple system architectures
- Analyze and select preferred system architecture/element solution
- Model, simulate, and prototype system architectures
- Define, refine and integrate system physical configuration
- Implement requirements and design feedback loops
Software Traces the System Engineering Design into the Software

System Engineering Design

Software/Hardware Design
System and Software Design With Full Traceability in an Integrated Model

- System requirements are decomposed into lower level specs and are traceable to the software and hardware designs.

- CONOPS (operational activities) are traceable to software and hardware behavior in use case, activity, state, and sequence diagrams.

- Requirements at each architecture level are allocated to SW/HW components and functions (activities).

- So have we achieved the objective?

- Along the way there was some creative modeling to make this happen:
  - System level items modeled in DoDAF (UPDM or SysML) per customer requirement.
  - Software items modeled in UML due to SW team familiarity with UML and (in some programs) desire to generate code from model.
  - DoDAF model items aren’t typically allowed on SysML diagrams.
  - Had to transition from DoDAF to UML by switching tools or building a transition layer.

One Model, Multiple Viewpoints
Typical Architecture Synthesis Overview

SEIT and Software/Hardware IPTs Share Ownership of Architecture and Design

Decomposition of requirements and structural items not shown but are similar to activity decomposition.
Architecture Synthesis Overview

Transition Layer Bridges Between SEIT and Software/Hardware Parts of Model

Transition Layer in one of these levels

Decomposition of requirements and structural items not shown but are similar to activity decomposition.
Conclusions

- Objective of full traceability from system to software is achievable but faces hurdles
- A single model can support both SEIT and SW/HW objectives
- Current architecture modeling tools don’t consistently provide a good solution
- UPDM and/or SysML may need revision to facilitate better traceability, or is it a tool implementation issue?

One Model, Multiple Viewpoints