Challenges and Approaches to Architectural Analysis in the Evolution of SW Intensive System Architectures

March 2006

Phillip Schmidt
Software Engineering Subdivision
The Aerospace Corporation
Phillip.P.Schmidt@aero.org
Outline

- Lifecycle Challenges
- Experiences
- Model Driven Engineering Approaches
- UML® 2.0 based approach in an evolving lifecycle
- Open Issues

® UML, Unified Modeling Language, is a registered trademark of Object Management Group, Inc. in the United States and other countries
Lifecycle Challenges

- Restore confidence to deliver space systems (processes and products)

- Understand how legacy designs constrain newer innovations

- Scalability of current practices in managing complexity
  - Synthesizing solutions

- Early insight while managing lifecycle change
  - Evolving conceptual models
  - Design optimization from alternative trade-offs
Experiences

- Conceptual and implementation-oriented models not well managed
  - Goal tradeoffs affecting operational concepts not well supported (e.g. safety vs. security)
  - Better methodologies to represent/analyze evolving conceptual architectures needed
- Evolving requirements refinement and constraint impacts
- Solutions-oriented vs. problem domain-oriented approaches
  - Construct by correction vs. correct by construction
  - Domain abstractions frequently revised: Platform Independent Models not unique!
  - Test driven design is costly
- Complexity of multi-phase builds
  - Split baseline designs complicates re-baselining and testing
  - High risk refactoring/rework from poor problem understanding
- Crosscutting concerns do not match problem ownership decomposition
Idea

Expanded Early Architectural Focus

- Needs Stmt, Opr env, Alternative tradeoffs
- System Purpose, Operational Concepts
- Request for Proposals
- Reqt Spec
- Use Cases
- Models

Common Architectural Focus

- Test Plans
- ICDs
- Code ECP
- Unit Tests
- Test Procedures

- Pre-contract planning
- Contract award
- Early analysis
- Early design
- Early dev
- Early testing
- Early integration

Architecture Trade off Analyses

Code Inspection

Independent test verification

Model Driven Engineering Approaches

- Model Integrated Computing (MIC)
  - Domain Specific Modeling Languages (DSML)

- Model Driven Architecture (MDA®)
  - UML 2 with domain-specific profiles
Current UML-related Work

- Dec 2004 – OMG UML Profile for STP v1.1
- Jan 2005 – OMG SPEM v1.1 Software Process Engineering Model
  - Metamodel or UML profile
- Feb 2005 – OMG UML Profile for MARTE RFP
  - STP profile replacement
- Jun 2005 Object Constraint Language 2.0 Spec (working)
- Jul 2005 UML 2 Specs
- Sep 2005 – OMG released UML Profile RFP for DoDAF/MODAF
  - System, technical, operational, strategic views
- Nov 2005 – MOF Query View Transformation Spec
- Other work:
  - Life Sciences
  - Knowledge Discovery
  - Space System Monitor and Control
  - Wireless
  - SW defined radio
  - Object DBs
  - Space Operations
  - Robotics
Model Driven Engineering Approach

Reference SW Process Models
- Reference Architecture Models
- Target Architecture Models

Platform-independent models
- Program-Specific Model Marking
- Program-Specific profiles
- DoDAF profile
- SPERM profile

Platform-specific models
- STP profile
- Program-Specific profiles
- External Information

Analysis Support Tools
- Analysis Support
- Contract Options
- Cost/Schedule Estimates
- Resources/Staffing
- Product workflows
- Multi-phase deliveries

COTS/GFE Dependencies
- Operational concept
- Mission Goals
- Requirements
- Feasibility analysis
- System constraints
- Optional Features
- Legacy Support

Platform-specific Traceability
- Requirement traceability
- Performance Data
- Legacy Dependencies
- COTS selection

Other Program Deliverables
- Equip Specs
- Design Specs
- Code
- Test Plan/proc
- DoDAF profile
- Program-Specific profiles
- External Information

UML Profile Approach

Periodic Deliveries of Evolving Artifacts

UML Model
  - Use Cases
  - Reqts
  - Sizing, Timing, Constraints
  - Test Procedures
  - Code
  - Other

As-designed UML (multi-viewpoint Metamodel)

Profiles
  - Domain-specific
  - Program-specific

XML

Augmentation Support

Re-augmentation Aspects

Concern Management/Extraction Aspects

Other Assessment Tools

COTS translation
  - Project-specific and reusable translations

Architectural markers (augmented/native)

UML Profile Approach

- Use UML2 models and domain-specific profiles as an architectural analysis framework
  - Aspects of interest defined as stereotypes, constraints, tags in domain-specific/project-specific profiles
    - Applicable across entire lifecycle
      - Requirements, constraints, non-functional goals, other model info (legacy)
      - Capture early architectural information prior to contract award
    - Permits augmentation
      - To identify conceptual vs. implementation models
      - Of reverse-engineered legacy assets
      - Reusable across evolving models
    - Crosscutting concerns managed over metamodel space
  - Capture early conceptual reference architectural information in an implementation-independent model
- Model query/transform capabilities
  - Model checking: aspect-oriented architectural analysis
  - Simulation generation for analysis of alternative designs
- Map conceptual model to implementation model
  - Permits project-specific schemas for model augmentation
Applying MDA approach in the System Lifecycle

- Pre Concept
- Concept Development
- Design Development
- Acquisition & Operation Support

Control Gates
Concept Development Rdy
Design Development Rdy
Production Rdy

Acquisition Organization
- Platform Independent (Reference) Architectural Models
- SW Process Engineering Models
- Mappings

Development Organization
Platform Specific (Target) Architectural Models
Program plans, schedules, development products

Operational Organization
Arch maint/evolution

Open Questions/Issues

- What challenges to analyzing evolving architectures have you found?
- What evaluation techniques have you found useful?
- How should systems acquisition business model change?
- How should architecture artifacts be maintained during life-cycle?
- Ownership, maintenance of conceptual models
- Managing mappings between conceptual and implementation models
- Processes to resolve crosscutting concerns effectively
- Effective interchange of models for analysis
Backup
All trademarks, service marks, and trade names are the property of their respective owners.
UML Model Evolution Example

Program Specific profiles

QVT

Data

Automated traceability between models

Other Program Deliverables

Program Specific profiles

Development

Can be small, modular, multiple models

Analysis Support Tools

Assessments

Other Program Deliverables

Program Specific profiles

Other Program Deliverables

Program Specific profiles

Other Program Deliverables

Program Specific profiles

QVT?
Implementation Approach

Eclipse

UML 2.0 Model

RT Profile

REACT Profiles

Project-Specific Profile

XML Schemas

UML Profiles

RT Support Profile

OMG Based:
- Schedulability
- Performance Timing
- Resources

REACT Profiles

Requirements Environment

Dynamic Assessment

Other Augmentation

Project-Specific Profiles

Domain-Specific

Sequence Diagram

State Diagrams

Other Schemas

Model Execution and Dynamic Assessment Tools

Analysis Results

UML 1.x Models

Config Files

Augmented UML 2.0 Model with Augmented Profiles

QVT transformations

Augmented Profiles

OMG Based:
- Schedulability, Performance Timing, Resources, etc.

Project-Specific Profile Augmentation Aspects

Contractor-provided

REACT Data

Eclipse Environment

Eclipse Support

REACT-Implementation

REACT-generated

COTS

Analysis Results

Model Extractor

UML 1.x, 2.0 Metamodels

UML 1.x & 2.0 Metamodels

Model Generator

Sequence-State Diagram Support Aspects

Requirement Visualization

QVT Transformation Plugins

OCL, QVT Support over EMF

QVT transformations

UML 2.0 Model

Ecore XML schemas

Requirements

Use cases

Project Data

Code

Other

Parser-provided

REACT Data

Eclipse Environment

REACT Support

REACT-Implementation

REACT-generated

COTS
Model Driven Development (Ideal)

Reference Model

Platform Independent Model

Constraints, Requirements, Ops Concepts

Target Model Refinement

Model Translator

Platform Specific Model
Model Driven Development (Evolving)

Reference Model Evolution

- Platform Independent Model
  - Constraints, Requirements, Ops Concepts
  - Model Translator(A₁)
  - Model Translator(B₁)
  - Model Translator(Z₁)

Platform Specific Models:
- Model A₁
- Model B₁
- Model Z₁

Target Model Refinement

- Platform Specific Models:
  - A₁A₂...
  - B₁B₂...
  - AiB₁...

Design Alternatives