Adoption Practices for Model Driven Engineering (MDE)

presented at:
Lockheed Martin Architects Workshop
August 17, 2011
by
Mark R. Blackburn, Ph.D.
(Mark.Blackburn@stevens.edu)
561.637.3452
Overview of Presentation

• Context setting
  – Stakeholder perspectives on model adoption
  – Current situation
  – Vision for the future
  – Strategy for getting to the future state

• Part I: challenges and why architecting matters

• Part II: model adoption practices

• Part III: research topics that are emerging as necessary practices for System of Systems (SoS) MDE
What are the Impacts of Model Adoption on the wide array of Stakeholders?

1) Skills, lifecycle timing, practices
2) Structures, virtual to physical, complexity
3) Operational requirements
4) Communication, timing of deliverables
5) Skills - formalizes communication of requirements and interfaces

“It is not the strongest of a species that survives, nor is it the fittest; it is those that can adapt the fastest.” (Darwin)
Today: Customers need continuous integration of capabilities spanning wide range of mission and system domains.
Future Vision:
Managing complexity in all dimensions through Systems of System Engineering

An Integrated Framework for NextGen Operations

Cross-Cutting Factors
- Environmental
- Safety
- Information Security
- Economic
- International
- Regulation

Enablers
- People
- Procedures
- Technology
- Data/Information
- Policy

Legend:
- Private Sector
- FAA (USG)
- Local entities
How do we get there (one strategy)?
Engineer Resilient Systems: Architecting to rapidly adapt to user needs in uncertain futures

Maximizing flexibility and innovation for uncertain futures: keeping more options open

Trustworthy Systems Design
Conceptual Engineering

Model Based Engineering
Platform Based Engineering

Trustworthy Systems Design

Replaceable product families

Design/Engineering via models, reuse of fielded (and modeled) systems
Systems of systems are emerging in many domains enabling unimagined complexity.
We must produce systems of the same complexity as hardware with similar costs and schedules.

**What’s Different?**
Software behavior often relies on floating point variables with nonlinear relationships and constraints.

*Note (*): Not a great metric. But that’s what we have today. META will come up with better metrics.*
Augustine’s Law – Growth of Software: Order of Magnitude Every 10 Years
Complexity Results in Diseconomy of Scale In Software often impacting size, scope and cost estimates

Optimal Architecting Matters: sequential path of least resistance often leads to delivery of poorly performing systems
New capabilities need to be continuously developed, deployed, and evolved.
Functional analysis across SoS is required to understand tradeoff of capabilities (CONOPS) and impact analysis.
Architecting is required for asynchronous integration and test across multiple SoS layers.
Transitioning into operations must accommodate users with mixed operational capabilities and maintain trusted system properties.
“We model to reason about the problem... And to communicate with others.”
Model Driven Engineering will revolutionize concurrent engineering
MDE approaches and tools must address gaps

- **Mission**: Understand evolving needs
  - **Customer**: Supports
  - **Mission**: Perform
  - **Users**: Enhance
  - **Skill**: Required
  - **Companies**: Built into
  - **Products**: Need

**How to use capability**
- **Standards**: DoDAF / UPDM
- **Methods**: SysML
- **Tools**: SystemC, VHDL, Simulink, UML

**What to build (architecture)**
- **Systems**: Family of Systems (FoS)
- **Subsystems**: System of Systems (SoS)
- **Capabilities**
  - Capability-based technology evaluation
  - **M & S**: Domain Specific Modeling

**GAP**
- **MDE**
- **Domain Specific Modeling**
From traditional SE ➞ Model-Based SE – standard, structured, rigorous, & linked

IEEE 1220 SE Process

- Requirements Analysis
  - Requirements Baseline
- Requirements Validation
  - Validated Requirements Baseline
- Functional Analysis
  - Functional Architecture
- Functional Verification
  - Verified Functional Architecture
- Synthesis
  - Design Architecture
- Verification/Validation
  - Verified Design Architecture

**PROCESS INPUTS**

IEEE 1220 SE Process

- Requirements Analysis
- Requirements Validation
- Functional Analysis
- Functional Verification
- Synthesis
- Verification/Validation

**PROCESS OUTPUTS**

- SysML Diagram
  - Structure Diagrams
    - Block Definition Diagram
    - Internal Block Diagram
    - Parametric Diagram
  - Behavior Diagrams
    - Use Case Diagram
    - Activity Diagram
    - Sequence Diagram
    - Statechart Diagram
- Requirement Diagram
Many tools support a typical SysML usage scenario

(Build me to see how it works)
Model Driven Engineering covers concepts, practices, tools, and future ideas – this is a core process for MBSA/MBSE
Structural views should include system domain, context, and interfaces.
Model topology often mirrors architecture of system
Establish common package elements to organize and structure model

- Change log
- Recursively applied
  - Analysis – could have subpackages
    - Context
    - Mission concept
    - Operational concept
    - Stakeholders
    - Scenarios
    - User cases
  - Behavior
  - Requirements
  - Structure
    - Architecture
- Simulations
- Variants
- Verification and test
- Documents (controlled)
- Domain context
- Environment
Model artifacts trace requirements through views and map derived requirements to software / hardware subsystems.
For continuous integration and test, we must be able to understand all of the interfaces and allocated requirements.

See next slide for example pattern for mapping HW/SW.

Context
Models depict interfaces and derived requirements allocated to SW & HW

Example use of MARTE Stereotype for HW Computing Resource

Create package with hyperlink to another model that refines the software
Behavioral views provide inputs for continuous integration and test planning and execution.
Architect for testability to support automation and leverage simulation and legacy data.
We must engage stakeholders in new ways to adapt faster and to determine what works, what doesn’t, and how it should be used.
A key focus is on developing the CONOPS for capabilities that need rapid deployment.
Operational views are critical as they represent how the system is to be used.
CONOPS: Then and Now
We have not progressed far - no meaning behind graphics, no human roles represented, takes too long, and customer often not involved

First Airborne Early Warning System to defend against aircraft (1945)

CONOPS from any current Naval program
New modalities and engineering capabilities are required to manage exponential complexity.
Concept engineering through graphical storytelling builds capability scenarios that are executable to understand dynamics and tradeoffs.
Graphical CONOPS can be leveraged for virtual training addressing challenge of evolving operational capabilities.
Successful model adoption often uses pilot projects to reduce risk

#1 – Understand how to relate traditional process activities to modeling practices and modeling artifacts

#2, 3 – Structure modeling context, domain, actors, target system, interfaces to test, simulation, environmental models, and external systems

#1, 3, 4 & 5 – Address changes to lifecycle schedule and deliverables that can impact proposals, reviews and stakeholders

#1 & 2 - Need to incorporate modeling methods, structure, practices and standards
Skills – Don’t “jump” into projects without knowing how to use MDE tools; have the right balance of modeling and domain expertise

Domain Experts

Modelers with Programming Skills

- Analysis/Model Checking/Proof of Properties
- Structure/Interfaces
- Behavior (functions)
- Concurrency
- Resources/Environment

Configuration Management (Based completely on the Models)

Complete Traceability

Test Automation

Transformation/Generation

Host

Simulation

Target

Verification Evidence

Model Validation & Verification Evidence
Talk with customer about technology, process, and deliverable changes

Incomplete or inconsistent models are obvious and difficult to review

Model-based artifacts contribute to multiple phases of reviews and downstream needs (e.g., V&V)
Topics discussed today provide coverage over some emerging issues and gaps

<table>
<thead>
<tr>
<th>Emerging Issues and Gaps</th>
<th>Concept Engineering &amp; Graphical CONOPS</th>
<th>Modeling Architectures for I&amp;T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecting for resilience</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Capability mapping bi-directionally</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Capability impact analysis for systems of systems</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Tradeoff analysis</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Continuous asynchronous integration and test</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Transition into operations</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Transforming the systems engineering workforce</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Involving disparate stakeholders</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Methods and standards</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>MDE adoption practices</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
There are many important ideas that we did not explore
Summary:

Optimal architecting will be critical in engineering resilient systems that can rapidly adapt to user needs in uncertain futures.

MDE can better formalize the architecture to support adaptable, evolvable systems important in complex systems of systems.

MDE can provide early insights into V&V and better support impact analysis needed for continuous integration of capabilities.

Adoption practices and method guidance should be considered and refined in pilot projects to manage risk.

Our research in Graphical Concept Engineering can help address operational needs while formalizing capabilities that span the SoS and can be leveraged for virtual training addressing challenges of continuous operational changes.
About Systems Engineering @ Stevens

• Largest Graduate Program in Systems Engineering in the United States
  – Broad engagement with Industry and Government
  – International Outreach
• Relevant and Flexible Curriculum Architecture
  – Developed and continually refined in collaboration with Industry and Government partners and sponsors
  – Individual Courses, Graduate Certificates and Degree Programs
  – Convenient Delivery Formats
• Experienced Faculty
• Leadership within the Systems Engineering community (US and Globally)
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AADL</td>
<td>Architecture Analysis &amp; Design Language</td>
</tr>
<tr>
<td>AP233</td>
<td>Application Protocol 233</td>
</tr>
<tr>
<td>ATL</td>
<td>ATLAS Transformation Language</td>
</tr>
<tr>
<td>BPML</td>
<td>Business Process Modeling Language</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-Aided Design</td>
</tr>
<tr>
<td>CASE</td>
<td>Computer-Aided Software Engineering</td>
</tr>
<tr>
<td>CATIA</td>
<td>Computer Aided Three-dimensional Interactive Application</td>
</tr>
<tr>
<td>CDR</td>
<td>Critical Design Review</td>
</tr>
<tr>
<td>CMM</td>
<td>Capability Maturity Model</td>
</tr>
<tr>
<td>CMMI</td>
<td>Capability Maturity Model Integration</td>
</tr>
<tr>
<td>CWM</td>
<td>Common Warehouse Metamodel</td>
</tr>
<tr>
<td>DBMS</td>
<td>Database Management System</td>
</tr>
<tr>
<td>DoDAF</td>
<td>Depart of Defense Architectural Framework</td>
</tr>
<tr>
<td>DSL</td>
<td>Domain Specific Languages</td>
</tr>
<tr>
<td>HW</td>
<td>Hardware</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines</td>
</tr>
<tr>
<td>ICD</td>
<td>Interface Control Document</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>INCOSE</td>
<td>International Council on Systems Engineering</td>
</tr>
<tr>
<td>IO</td>
<td>Input / Output</td>
</tr>
<tr>
<td>IPR</td>
<td>Integration Problem Report</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>Linux</td>
<td>An operating system created by Linus Torvalds</td>
</tr>
<tr>
<td>MAP</td>
<td>Modeling Adoption Practices</td>
</tr>
<tr>
<td>MARTE</td>
<td>Modeling and Analysis of Real Time Embedded systems</td>
</tr>
<tr>
<td>MATRIXx</td>
<td>Product family for model-based control system design produced by National Instruments</td>
</tr>
<tr>
<td>MBT</td>
<td>Model Based Testing</td>
</tr>
<tr>
<td>MBSA</td>
<td>Model Based System Architecture</td>
</tr>
<tr>
<td>MBSE</td>
<td>Model Based System Engineering</td>
</tr>
<tr>
<td>MDA*</td>
<td>Model Driven Architecture*</td>
</tr>
<tr>
<td>MDD**</td>
<td>Model Driven Development</td>
</tr>
<tr>
<td>MDE</td>
<td>Model Driven Engineering</td>
</tr>
<tr>
<td>MDSD</td>
<td>Model Driven Software Development</td>
</tr>
<tr>
<td>MDSE</td>
<td>Model Driven Software Engineering</td>
</tr>
<tr>
<td>MIC</td>
<td>Model Integrated Computing</td>
</tr>
<tr>
<td>MMM</td>
<td>Modeling Maturity Model</td>
</tr>
<tr>
<td>MoDAF</td>
<td>United Kingdom Ministry of Defence Architectural Framework</td>
</tr>
<tr>
<td>MOF</td>
<td>Meta Object Facility</td>
</tr>
<tr>
<td>MVS</td>
<td>Multiple Virtual Storage</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>OCL</td>
<td>Object Constraint Language</td>
</tr>
<tr>
<td>OMG</td>
<td>Object Management Group</td>
</tr>
<tr>
<td>OO</td>
<td>Object oriented</td>
</tr>
<tr>
<td>PDR</td>
<td>Preliminary Design Review</td>
</tr>
<tr>
<td>PIM</td>
<td>Platform Independent Model</td>
</tr>
<tr>
<td>Pro/EPro/ENGINEER</td>
<td></td>
</tr>
<tr>
<td>PSM</td>
<td>Platform Specific Model</td>
</tr>
<tr>
<td>RFP</td>
<td>Request for Proposal</td>
</tr>
<tr>
<td>ROI</td>
<td>Return On Investment</td>
</tr>
<tr>
<td>RTW</td>
<td>Mathworks Real Time Workshop</td>
</tr>
<tr>
<td>SSCI</td>
<td>Systems and Software Consortium</td>
</tr>
<tr>
<td>SCR</td>
<td>Software Cost Reduction</td>
</tr>
<tr>
<td>SDD</td>
<td>Software Design Document</td>
</tr>
<tr>
<td>SE</td>
<td>System Engineer</td>
</tr>
<tr>
<td>Simulink/Stateflow</td>
<td>Product family for model-based control system produced by The Mathworks</td>
</tr>
<tr>
<td>SOAPA</td>
<td>protocol for exchanging XML-based messages – originally stood for Simple Object Access Protocol</td>
</tr>
<tr>
<td>Software Factory</td>
<td>Term used by Microsoft</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SRS</td>
<td>Software Requirement Specification</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>SysML</td>
<td>System Modeling Language</td>
</tr>
<tr>
<td>SystemC</td>
<td>IEEE Standard 1666</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>XMI</td>
<td>XML Metadata Interchange</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
<tr>
<td>xUML</td>
<td>Executable UML</td>
</tr>
<tr>
<td>Unix</td>
<td>An operating system with trademark held by Open Group</td>
</tr>
<tr>
<td>VHDLVerilog</td>
<td>Hardware Description Language</td>
</tr>
<tr>
<td>VGS</td>
<td>T-VEC Vector Generation System</td>
</tr>
<tr>
<td>VxWorks</td>
<td>Operating system owned by WindRiver</td>
</tr>
</tbody>
</table>
Trademarks

- OMG®, MDA®, UML®, MOF®, XMI®, SysML™, BPML™ are registered trademarks or trademarks of the Object Management Group.
- IBM™ is a trademark of the IBM Corporation
- Java™ and J2EE™ are trademark of SUN Microsystems
- XML™ is a trademark of W3C
- BridgePoint is a registered trademark of Mentor Graphics.
- Java is trademarked by Sun Microsystems, Inc.
- Linux is a registered trademark of The Linux Mark Institute.
- MagicDraw is a trademark of No Magic, Inc.
- MATRIXx is a registered trademark of National Instruments.
- MVS is a trademark of IBM.
- Real-time Studio Professional is a registered trademark of ARTiSAN Software Tools, Inc.
- Rhapsody is a registered trademark of Telelogic/IBM.
- Rose XDE is a registered trademark of IBM.
- SCADE is copyrighted to Esterel Technologies.
- Simulink is a registered trademark of The MathWorks.
- Stateflow is a registered trademark of The MathWorks.
- Statemate is a registered trademark of Telelogic/IBM.
- TAU/Developer is registered to Telelogic/IBM.
- T-VEC is a registered trademark of T-VEC Technologies, Inc.
- UNIX is a registered trademark of The Open Group.
- VAPS is registered at eNGENUITY Technologies.
- VxWorks is a registered trademark of Wind River Systems, Inc.
- VectorCAST is a trademark of Vector Software.
- Windows is a registered trademark of Microsoft Corporation in the United States and other countries.
- All other trademarks belong to their respective organizations.