LOCKHEED MARTIN
ENGINEERING

The Challenge of Managing Enterprise-Scale Models in MBSE

Steven W. Mitchell
Lockheed Martin Fellow
Lockheed Martin Master Systems Architect

25 Jan 2015

Agenda

- Example: Using MBSE to Integrate the Submarine Combat System Enterprise
- Issues with the Current Model Management Environment
- Challenges for Tool Vendors

LOCKHEED MARTIN
ENGINEERING

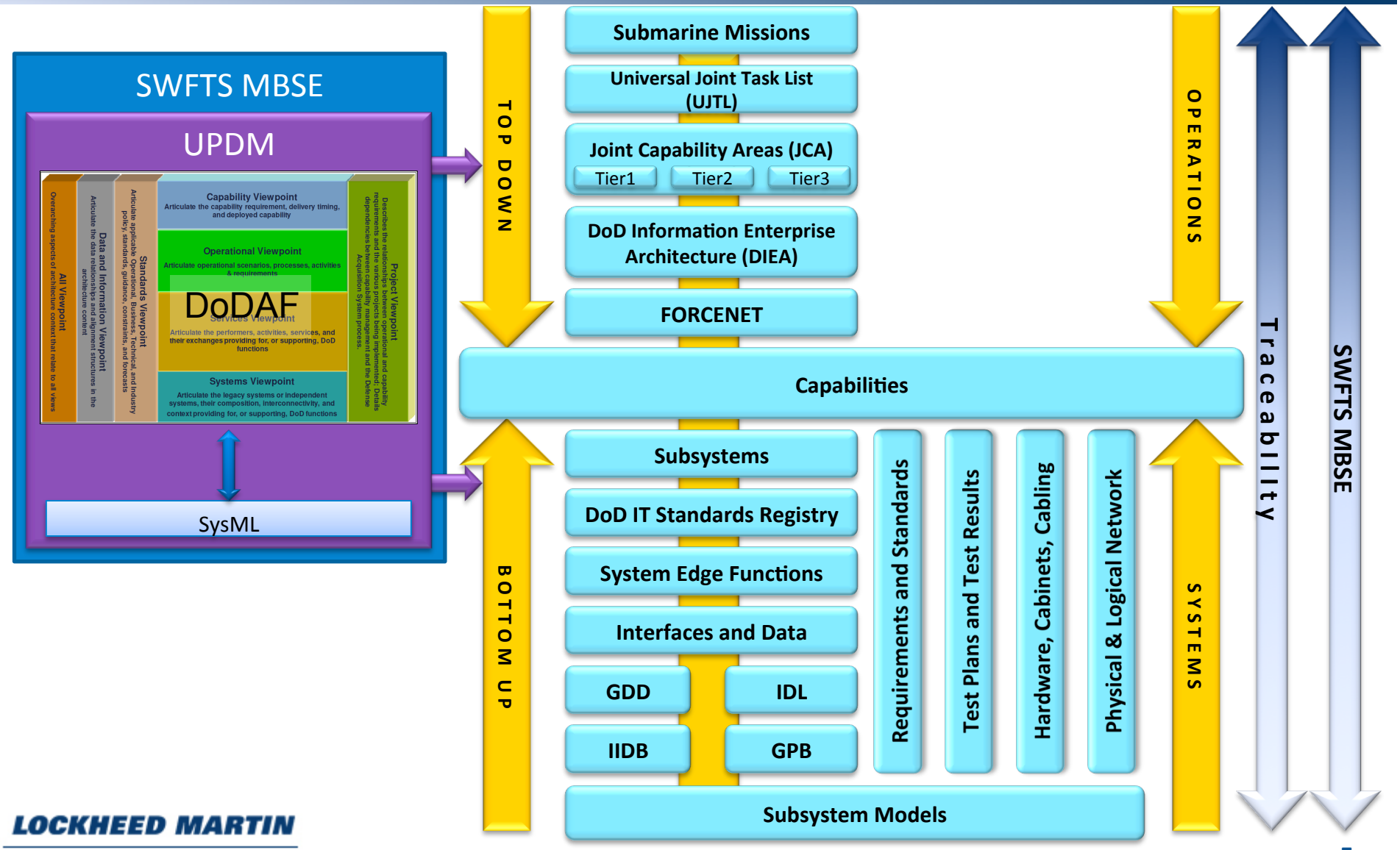
Submarine Warfare Federated Tactical Systems (SWFTS)

A Modern Naval Combat System Managed using MBSE

Submarine Warfare Federated Tactical Systems

- A common Combat System product family deployed across multiple submarine classes
  - USN: Los Angeles (SSN 688), Ohio (SSGN 726, SSBN 730), Seawolf (SSN 21), Virginia (SSN 774), Ohio Replacement (SSBN)
  - RAN: Collins (SSG 73)
- Federates multiple subsystems from multiple program offices and contractors
  - Sonar, ESM, Imaging, Tactical Control, Weapons Control, Communications, etc.
- SWFTS SE&I manages inter-subsystem interfaces and system I&T

Team Submarine MBSE Long Term Vision



LOCKHEED MARTIN
ENGINEERING

Scale of SWFTS Interface Baseline Model

- ~35 subsystems from ~20 program offices
- ~3,500 interface requirements
- ~500 services
- ~5,000 model elements for interfaces
  - Interfaces, methods, data structures
- >15,000 relationships between model elements
- >400,000 model elements

LOCKHEED MARTIN
ENGINEERING

Issues with Enterprise Model Management

Current Tools are Challenged by Industrial Scale Problems

- Supporting large, distributed teams is a challenge
  - Tool tailoring (code engineering sets, standard modules or profiles, diagram customizations, tool preferences, etc.) typically done on a per-installation basis, rather than centrally managed
  - Makes managing project standards unnecessarily labor-intensive
- Most stakeholders will never use a *modeling* tool
  - Must *efficiently* support multiple access modalities (web interfaces, documents, etc.)
  - Modeling tools must facilitate multi-modal access, including commenting, with minimal user training and effort
- Models do not lend themselves to a cumulative merging process
  - An average SWFTS BCR modifies 10 of the ~500K elements
  - MagicDraw® Merge process checks them all
- Moving from MBSE to MBE is an even bigger challenge

MBSE Long-Term User Expectations

- Model users who have a background in software expect to configuration manage a model similar to how GIT or SVN manages code
  - Model Integrity and Validation Occurs as the model is saved
- Systems Engineers require integration of system modeling tools with requirements database and analysis tools
- Enterprise-scale modeling should keep content focused on the shared repository and user views whether shared (public/protected) or localized (private)

Change Management (CM) Integration

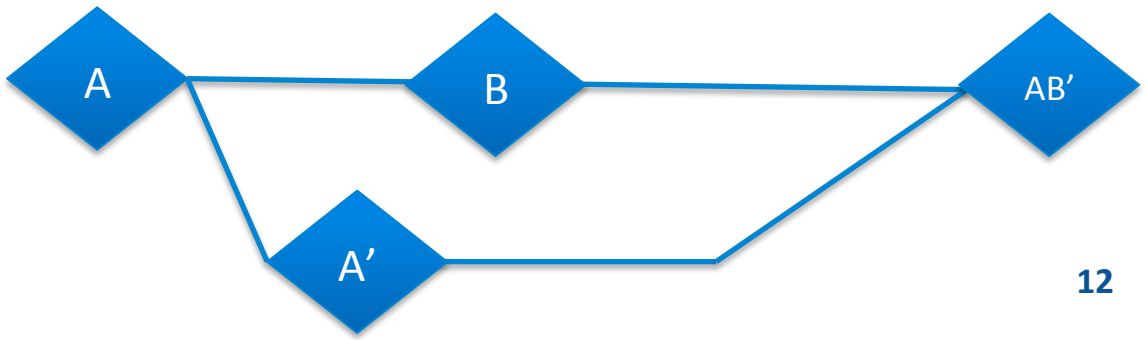
- CM is a ***significant*** challenge in the corporate environment
- For an enterprise scale model every change has to be documented, approved, resolved, verified and integrated. Each phase of a change needs to be managed and controlled.
  1. Need to create "tickets" or issues (similar to JIRA) for each proposed change
  2. Each proposed change need to be reviewed and approved
  3. An engineer/modeler needs to implement each approved change
  4. The modified model incorporating the change needs to be reviewed and approved
  5. The set of approved and implemented changes need to be integrated into the main model
  6. All changes must automatically be logged with associations to change requests
- This overall workflow is missing from most SysML modeling tools

Managing Changing Requirements

- Requirements often exist in 'punctuated equilibrium' across life cycle of large programs
- Typical MBSE workflow assumes they are static
  1. Receive enterprise architecture and requirements from customer
  2. Model objective system from requirements, maintaining traceability
  3. Implement system from model
  4. Maintain system and model
- When revised requirements arrive, need efficient tool support to merge changes into an on-going model, while maintaining traceability to each revision, with minimum re-work

Model Merging

- Current implementations typically use a three-way merge process for model merges
  - This requires two complete element-by-element comparisons of the model
  - In practice, too many tertiary model changes are rolled into a merge
    - AB' requires AB and AA' Compare and then the delta applied to B



LOCKHEED MARTIN
ENGINEERING

Problems With Current Model Merge Support

- Extremely slow
  - 100's of thousands of elements are compared for a single model change
- Memory Intensive
  - Three copies of the model are loaded into memory and compared
- Non-optimized
  - A single simple model change can affect multiple elements
- Inefficient
  - Most modeling tools store large text-files to represent the model
  - These files are inconsistently changed or "re-authored" with every save

LOCKHEED MARTIN
ENGINEERING

Model Management Challenges to Tool Vendors

More Efficient Model Repository

- Model storage must be efficient for enterprise-scale models
  - Managing 100s or 1000s of copies of large monolithic files for model versioning is a backup nightmare
- Model elements should be stored in a per-element structure
  - Element-level history to improve access efficiency
  - Element-level locking without impacting automated backups
  - Scanning or updating the whole model when setting/freeing a lock is very slow and highly disruptive of modeling teams
- Association between model deltas and issue-tracking database
- "Instant" deltas between subsequent model versions
- Indirect links between model elements
  - A Linking Structure vs. Embedded Links
- The ability to "rewind" or "fast-forward" single model elements
  - Similar to a GIT concept of taking a previous version of a file and replaying local changes on top to merge

Monolithic file-based repositories do not effectively support enterprise-scale MBSE

Change Management Challenges

- Expose an API to support integration with popular open source issue tracking system: Redmine, Bugzilla, Trac, etc…
- Communicate with the issue tracking system to track what state an issue is in, and enforce changes made to the model to be associated with the relevant issue
  - Requires many-to-many association between issue URLs and modified model elements
- Support promoting approved changes in the model from a working/change branch to a production branch.
  - Everyone could share one working branch, as long the tool tracks which changes are associated with which issues, or
  - Individual changes could be in separate working branches

Additional Change Management Challenges

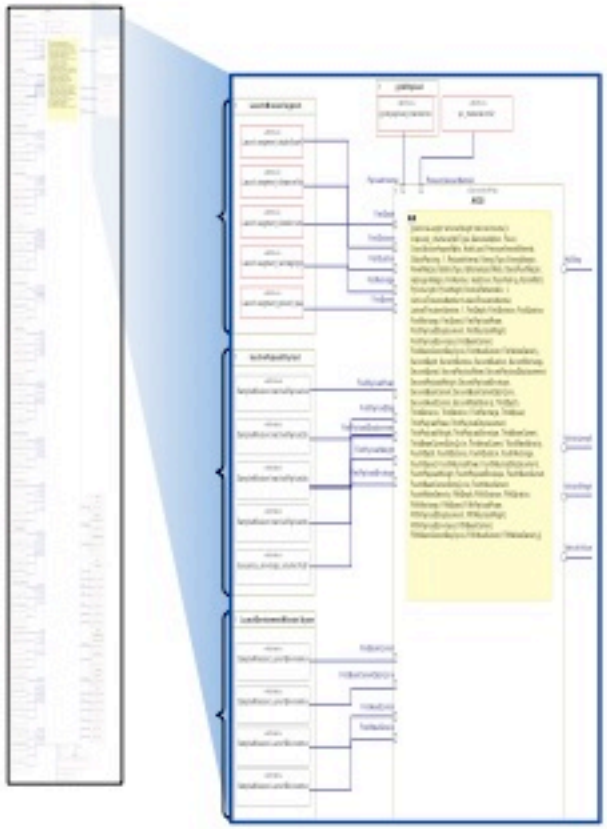
- Simplify peer review of changes
  - Change detection / review scope should be user-selectable (at element or package level) to minimize full 'model compares'
  - Provide easy, controllable API access to changes
- Enhance visibility into model merge
  - Need visibility into Project Option / Profile / Tool Tailoring / Diagram Customization / etc. changes, as well as changes in model
  - Where tool provides automatic model or diagram merging, need insight into decision process

Simplified User / Project Management

- Should integrate with enterprise single sign-on
  - Leverage existing Active Directory or LDAP user access control
  - Provide Individual, Group, and Privileged access
- Provision tool tailoring on basis of Group
  - Code engineering sets, standard modules or profiles, diagram customizations, tool preferences, etc.
- Model configuration control should also be controllable by Group
  - e.g., only members of CM group can commit changes to model trunk, UserGroup1 only has access to branches A & B, …

Efficient and Reliable Integration

- Most legacy programs use databases for requirements engineering
  - Traceable synchronization between modeling tools and requirements management tools is often fragile and very slow
  - Multi-user support is poor
- Current interface approach (specialized Parametric diagrams) between SysML models and analysis tools is clumsy and hard to maintain for real world models



Example Parametric Diagram interface between IBM Rhapsody and Phoenix ModelCenter

References and Acknowledgement

- **References**
  - *System Engineering and Integration for Submarine Combat Systems in the COTS Environment*, Dennis J. Cooper, Joan Sienkiewicz, and Mike Oliver, **NDIA Systems Engineering Conference**, San Diego, CA, 23-26 October 2006
    - Briefing Available Here
  - *Model-Based System Development for Managing the Evolution of a Common Submarine Combat System***,** Steven W. Mitchell, **AFCEA-GMU C4I Center Symposium on Critical Issues in C4I**, 18-19 May 2010
    - Paper and Briefing Available Here
  - *SWFTS - System Engineering Applied to Submarine Combat Systems***,** Mark A Zingarelli, Steven R. Wright, Robert J. Pallack, and Kevin C. Matto, **Engineering the Total Ship**, Falls Church, VA, July 2010
    - Briefing Available Here
  - *Complex Product Family Modeling for Common Submarine Combat System MBSE,* Steven W. Mitchell, **Third International Conference on Model Based Systems Engineering**, Fairfax, VA, Sept 2010
  - *Bridging the Gap: Modeling Federated Combat Systems,* B. Gibson, S. W. Mitchell, and D. Robinson, **Third International Conference on Model Based Systems Engineering**, Fairfax, VA, Sept 2010
  - *Efficient Management of Configurations in the Model-Based System Development of a Common Submarine Combat System,* Steven W. Mitchell, **AFCEA-GMU C4I Center Symposium on Critical Issues in C4I**, 24-25 May 2011
    - Paper and Briefing Available Here
  - *Efficiently Managing Product Baseline Configurations in the Model-Based System Development of a Combat System Product Family,* Steven W. Mitchell**, INCOSE International Symposium**, Rome, Italy, July 2012 Paper Available Here
  - *Transitioning the SWFTS Program Combat System Product Family from Traditional Document-Centric to Model-Based Systems Engineering*, Steven W. Mitchell, **Journal of Systems Engineering,** Vol. 17, No. 3, Autumn 2014. p. 313-329 Paper Available Here

- **Acknowledgement**
  - This research was supported by NAVSEA contract N00024-06-C-6272

LOCKHEED MARTIN
ENGINEERING