Model Based Management of Configurations of a Complex Systems:
Common Submarine Combat System

Case Study

Saulius Pavalkis
System Analyst
Saulius.Pavalkis@nomagic.com
Contents

Introduction ..............................................................................................................................................3
Challenges Pushing the MBSE Adoption ...............................................................................................3
Configuration Management Solution ......................................................................................................4
Configuration Management Mechanism .................................................................................................5
No Magic response – Smart Packages Usage for System Configuration Catalogs .......................6
Detailed Solution for Creating Dynamic System Configuration Catalogs ...................................7
Conclusions ...........................................................................................................................................9
Introduction

In order to manage growth, complexity, and demand for resources of mission critical systems, Lockheed Martin Corporation (LMCO) has transitioned to using Model Based Systems Engineering (MBSE) (see the side bar) in large scale. The transition was very successful; but it also required adopting best practices along the way. The newest MagicDraw version provides real–life project capabilities (i.e. Smart packages) out of the box, which will provide further productivity and quality gains supporting configuration management approach.

Challenges Pushing the MBSE Adoption

MBSE has been adopted for US NAVY submarines combat systems software and hardware configurations management.

Submarine Warfare Federated Tactical Systems (SWFTS) program (see the side bar) provides parallel management of external interfaces to the combat system and internal interfaces between subsystems within the combat system.

In addition, to the complexity of configuration management, the SWFTS model is large. The combat system includes approximately:

- 35 subsystems from over 20 program offices
- 2,500 interface requirements
- 100 services
- 3,700 model elements for interfaces
- More than 15,000 relationships between model elements
- 500,000 model elements.

The scope of the SWFTS systems engineering efforts have increased over time with more parallel changes, more concurrent baselines, thus increasing the Engineering workload.

Lockheed Martin (LMCO)

Lockheed Martin is a global security and aerospace company that employs about 116,000 people worldwide. Majority of its business is with the U.S. Department of Defense and the U.S. federal government agencies. It is the largest provider of IT services, systems integration, and training to the U.S. Government.

Source (www.lockheedmartin.com/)

Model Based Systems Engineering (MBSE)

MBSE is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.

Source (INCOSE-TP-2004-004-02, Version 2.03, September 2007)

Submarine Warfare Federated Tactical System (SWFTS)

SWFTS manages combat subsystem Interfaces and combat system I&T. This includes the logical network configuration data (e.g. IP addresses, protocols, node characteristics) and the data exchanges including requirements levied on subsystems for data origination and receipt, data types, protocols and more. It combines multiple subsystems from Multiple Program Offices and Vendors.

A Common Combat System deployed across multiple fleets is comprised of 59 different variants installed across Royal Australian Navy Collins class submarines and 71 US Navy submarines from six distinct submarine classes including: USN: Los Angeles (SSN 688), Ohio (SSGN 726), Seawolf (SSN 21), Virginia (SSN 774).

To handle complexity, increase productivity, and save costs, MBSE was adopted to manage SWFTS configurations.

The key issue in applying MBSE was efficiently representing system variation to the systems engineering of product families. This is important both to minimize duplicative data to be maintained and synchronized within the system models, and to minimize the conceptual complexity of the system model.

**Configuration Management Solution**

To handle the task of dozens of product configurations managed in parallel, with many of those baselines being updated several times a year, LMCO developed a new SysML modeling technique.

It extends the concepts of libraries with SysML Catalogs to bound the complexity of the configuration task, improving the quality and efficiency of the systems engineering process.

Catalogs frame alternative views of the model for the engineer. Usage of catalogs gives ability to utilize the catalog as an active filter of the model:

- Reduces the scope of the library without duplicating the elements.
- Provides utilization assessments for elements across multiple baselines and baseline configurations.

As shown in Figure 1, the approved subset of servers from the list of all servers is imported into a catalog for a specific baseline (TI10 or TI12 in the example). Similarly, these catalogs are populated with other hardware components approved for those baselines. Each catalog restricts the scope of the configuration to those components approved for the specific baseline.

LMCO noticed that constructing the baseline system configurations is a technically challenging task. Given the large number of baselines that must be managed, the total number of software and hardware components, interface specifications, etc., used in one or more baselines at any given time is quite large.

For an engineer constructing a new baseline, hunting manually through dozens of server and switch models or tens of hundreds of versions of interface specifications would be so laborious and error-prone as to defeat the productivity and quality objectives of introducing Model Based Systems Engineering to the SWFTS program.
Efficient management of the product configuration process is a challenge in the evolution of any industrial scale product family. The standards themselves are not addressing this problem in a scalable fashion. In addition, existing UML/SysML modeling tool support for variation points appeared to be inadequate for an industrial problem of this magnitude.

**Configuration Management Mechanism**

It was necessary to create some mechanism or plugin for appropriately restricting the scope of objects available to the engineer constructing or modifying a given baseline. If the totality of servers, switches, displays, etc., included in the hardware model is considered as a library of candidate hardware components, what is needed is a catalog containing only those components, which are approved for baseline use in the configuration at hand.

The process of constructing a baseline from a set of catalogs is shown in Figure 2. In this case, a variant configuration from the TI10/APB09 baseline is being constructed for a specific class of submarines. The TI10 hardware catalog is open in the browser on the left side of the screen capture (1), and specific servers are being configured into processing racks that will be installed on the submarines (2).

LMCO identified that the tool support shown in Figure 2 is critical to the productivity and quality gains projected for the conversion of SWFTS from a document-based to a model-based systems engineering process. The unique solution was implemented by LMCO as a No Magic Inc. MagicDraw plugin.
LMCO predicted that a similar user-interface feature is likely to be imitated by other tool vendors as a natural side-effect of competition.

No Magic Inc. responded with a highly flexible capability to have a criteria dependent package → **Smart package**.

**No Magic response – Smart Packages Usage for System Configuration Catalogs**

A Smart package is a special collection of model elements. An element is included in the smart package automatically if the element meets the set of criteria defined by the user. For example the user can create a group “T14 Catalog” with the criteria “all components with import relations incoming from package T14 Catalog”.

**Note:** If you no longer need the contents of a smart package to be dynamic, you can simply freeze it.

Smart packages aggregate relevant elements so that you can:

- Browse, navigate, list, and discover these elements in the Containment tree.
- Narrow the scope in both the Find dialog and the Element Selection dialog.
- Define dynamic row and column scopes in dependency matrices. For example after tagging a component with T14, the component is automatically included into the group “T14 Catalog” and thus is added to the dependency matrix where this smart package is defined as scope.

Smart packages are query based. The newly enhanced query engine (Figure 4) is extremely flexible and is now the most powerful in the modeling tools industry. The criterion can be as simple as a UML relationship and as complex as an Object Constraint Language (OCL) expression. You can define the following flavors of criteria: Simple criteria, OCL expressions, Meta chains for navigation through chains of properties, and Java code.

The criterion can also be any combination of the items from the preceding list. In addition, the Query engine is parameters based so one query result can be the parameter of another query, scope, or type, without any limits.
Detailed Solution for Creating Dynamic System Configuration Catalogs

Figure 5 shows construction of the TI14 catalog of approved components from TI Hardware library of available components: (1) Smart package – the catalog Cabinet, Server, etc., use the query engine to aggregate required components dynamically. (2) Once the query is specified, the smart package –TI14 catalog content is created and updated automatically. (3) Some components, such as DELL R710, are imported into catalog TI14 manually by drag and drop. (4) Ownership of individual components, such as the DELL R710, is not changed and allows reuse in multiple catalogs.

Figure 5 Constructing catalogs of approved components from libraries of available components

Alternative solution is to construct catalogs based not on the import relations, but on properties values (1) Figure 6. Find query (1) is used to search for catalog TI14 components and show them in TI14 Smart package (3).
Figure 6 Labels based construction of catalogs
Conclusions

Challenges

- Manage the Complexity Faced by Systems Engineers
- Manage High Variability Between Platforms
- Maximize Reuse Between Baselines
- Improve the Quality and Efficiency of the Baseline Configuration Process

Benefits

- Usage of MBSE found bugs in previous baselines
- 13% Savings between SE and MBSE
  - 25% in Capability Definition
  - Another 10% over DOORS in Baseline Management
- Savings Seen in 4th Year
  - 2 Years to Implement Model
  - 1 Year Transition Overlap with Current Process

Solution

- Adopt MBSE to Enable a More Efficient System Engineering Process
- Provide Intuitive MBSE tools to enable Engineers to Develop Complex Systems with Maximum Reuse
- No Magic Inc. responded to LMCO and other customers working on complex systems configuration management with highly flexible capability to have criteria dependent package → Smart package
- Smart package capability with the most powerful in the modeling tools industry query engine enables efficient management of the product configuration process in any industrial scale product family. It is major mean to the productivity and quality gains for the conversion from a document-based to a model-based systems engineering process

Results

- MBSE applied to an existing system achieved greater productivity and improved quality of existing program.
- Hierarchy of Models Supporting TEAM SUBMARINE Engineering
- Reduced duplication and inconsistency of element definitions
- Developed Libraries and Catalogs to improve the quality and efficiency of the baseline configuration process
About the Author

Saulius Pavalkis
System Analyst, Global User Support Manager
Saulius.Pavalkis@nomagic.com

I'm Global User Support Manager and Analyst in MagicDraw R&D team for over 10 years with increasing responsibilities. My major expertise area is model-based requirements engineering. I'm the owner of a new Cameo Requirements Modeler product, which has been recently introduced in MagicDraw product line. I'm a Ph.D. candidate at Kaunas University of Technology (KTU). I'm also hold multiple professional certificates: OMG-Certified UML Professional, OMG-Certified Expert in BPM, ITIL V3, OMG-Certified Systems Modeling Professional. I have written multiple research and practical articles in model-based software design. I'm the founder and chief editor of modelling community blog (blog.nomagic.com) dedicated for sharing practical model-based engineering experience.

Contacts