

**2017 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY
SYMPOSIUM
SYSTEMS ENGINEERING (SE) TECHNICAL SESSION
AUGUST 8-10, 2017 – NOVI, MICHIGAN**

TRANSITIONING PROJECTS TO A MODEL-BASED APPROACH

Eric Alexander
Technical Planning & Management (TPM)
US Army TARDEC
Warren, MI

ABSTRACT

This paper will discuss the recent efforts of the Squad Centric Mounted Maneuver (SCMM) project to transition from document based to Model Based Systems Engineering (MBSE) in an effort to synchronize disparate sources of information into a unified model. This effort began with the transition of the system One-Wire diagram into a Systems Modeling Language (SysML) Internal Block Diagram (IBD) to generate the interfaces for the entire system. With the help of various stakeholders to format the Magic Draw diagram to look similar to the previous Visio diagram we were able to get buy-in from project leadership and Subject Matter Experts (SMEs) who have relied on Visio to generate these One-Wires. After the success of demonstrating the power of a modeling tool to rapidly update content and query data into usable reports we began using Magic Draw to document behavioral analysis through use case diagrams, sequence diagrams, state machine diagrams and activity diagrams. The project has continued to understand how to read these SysML diagrams with the help of TARDEC Architecture support and have been able to transition these architecture artifacts into more concise system requirements. The project is now gaining momentum by using a common language (SysML) to document system interfaces and behaviors once present only on conference room whiteboards and power point charts. From this effort came several techniques, reusable model content, cross-organizational collaboration within TARDEC, and additional best practices for system architecture development.

INTRODUCTION

Using models to represent complex systems is nothing new – ancient civilizations used models to describe complex systems such as the solar system and other geologic processes for example. What is new, in some domains, is the transition of project deliverables from volumes of disjoint artifacts (ICDs, Requirements, Drawings, etc.) to artifacts of the same kind from a single, integrated source of data – the system model. The system model in this context refers to the representation of the System of Interest (SoI) using SysML in a commercial grade modeling tool – in this case MagicDraw. “The OMG Systems Modeling Language™ (OMG SysML™) is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities” [1]. The representation of the SCMM system uses SysML constructs and TARDEC Technical Planning and Management (TPM) Architecture best practices to describe the relative maturity of the project at a given time. TPM Architecture developments that both assisted and matured from the SCMM project include the Integrated System Architecture Modeling Guide (ISAMG) – a collection of best practices in using SysML to model complex systems. The deployment of MBSE on a project, program, or organization must be executed

strategically in order to be considered useful to the key stakeholders. This paper will focus on some advantages of transitioning a project to a model based approach using SCMM as a case study, methods for successful transitions, as well as lessons learned along the way.

WHY MODEL-BASED?

Using models to represent complex systems is far superior to the efforts to manage disparate sources of technical data and the relationships between them. A well-defined system model is built upon an unambiguous modeling language, such as SysML, and implemented with best practices of Model Based Systems Engineering. TARDEC TPM Architecture is continuously developing a collection of best practices in the form of an Integrated System Architecture Modeling Guide (ISAMG) to inform future projects and speed up initial architecture constructs. Using a set of best practices, a highly capable modeling tool (Magic Draw), and a well-defined modeling language tailored for systems engineering (SysML) we can successfully describe any complex system with far greater clarity than ever before. Ultimately, a system model can further trace down to a software model to show implementation concepts. SCMM is in the process of doing just that with the SysML system model and UML software model referencing each other to support system modularity.

The SysML Learning Curve

Engineers love diagrams and are inherently good at deciphering them. What SysML provides, however, is much more than the ability to create well-formed diagrams. Placing too much emphasis on diagrams steers the development of the system model away from the true utility it can provide. The value of SysML is realized when the architecture is well-defined, scalable, and executable. In order to have a well-defined architecture one must know much more about SysML than simply how to create comprehensible diagrams. The system model is alive whether or not any diagram is created. Diagrams are often an efficient way of modeling relationships in bulk but should not be mistaken as the Model itself. The system architect must consistently educate stakeholders about the true nature of the model if they are to realize its actual worth. As models become increasingly complex and interconnected between projects, the underlying data structure will be put to the test.

MAKING THE TRANSITION

People are naturally resistant to change and therefore need to be sufficiently persuaded to get onboard with an effort to do so. Transitioning a project to a model based approach involves change in a variety of aspects including technical, organizational, and interpersonal.

Technical change involves the collection of disparate sources of technical information to be owned in the system model. Migrating technical information involves a period of data collection and collaboration with SMEs to populate the system model with the necessary content in sufficient detail. While engaging SMEs during this data collection phase, the system modeler must effectively capture the needs of the SMEs for describing the content and inform them of how the data is being managed within the SysML model. During these engagements the system modeler must take advantage of the opportunity to explain the benefits of using SysML as an unambiguous method of describing the structure and behaviors of the system or subsystem of interest.

Organizational change is sometimes required when transitioning larger programs to a model based approach as there it requires many organizational agreements. First, there must be an understanding of who administers the system model. Model administration should be documented in the project management plan or systems engineering administration plan. Administration includes setting access rights for different groups of users (read-only for reviewers, read-write for model contributors etc.). The effects of transitioning a project to a

model based approach without forethought on model administration can quickly derail the validity of the effort. Other agreements that occur during the transition is the relative accuracy of data in the model. It is essential that the model be developed with constant review from SMEs and (less frequently) project leadership to communicate the validity of the model at a given point.

Interpersonal change is a tricky thing to navigate and is dependent on a variety of factors. The most significant interpersonal change occurs when the system modeler (or modelers) begin interacting with the SMEs. The system modeler should consider the time constraints and priorities of the SME when asking them to contribute to the modeling effort. Modeling in real time can be more efficient given the right audience and complexity of material but can also severely disengage SMEs if they are not familiar with the format.

A technique that TPM Architecture has used successfully to efficiently absorb content during engagements with SMEs is what I like to call “Covert Modeling”. Covert Modeling means letting the SME’s present the material in their chosen media (often PowerPoint, Visio, CAD etc.) and absorbing the content in the model during the engagement and refining it later for review. This technique has worked well on SCMM as SMEs feel like they own the content and can present it in a way that suits them before TPM Architecture converts it into SysML form. This is especially helpful when dealing in a complex behavioral analysis discussion where the system model requires advanced SysML constructs to describe the behavior accurately (this takes some time offline to get it right).

Physical Architecture Definition

The late addition of TPM architecture support on the SCMM project resulting in the starting point of the transition at the physical architecture level. SCMM had matured subsystems down to the component level and the overall system view was maintained in a “One-Wire” Diagram generated in Microsoft Visio. Visio lacks the underlying data structure and language constructs of a SysML model and therefore this system One-Wire required much manual effort to maintain. After a few days of TPM Architecture support, the Visio One-Wire was successfully captured in a Magic Draw IBD. This system IBD was now dynamically linked to all other usages of the blocks in other subsystem IBDs. Using the SysML Model now allowed hands-off synchronization of ports between IBDs, and everywhere else for that matter since the changes made to elements of model definition are propagated throughout the entire model.

Modeling Behaviors

Once the project got onboard with the SysML model as a sufficient repository for system definition data, TPM Architecture was able to begin transitioning behavioral artifacts from various documents (PowerPoint, word documents, napkins etc.) into the model in the form of Use Cases, Activities, and Interactions that trace to the logical elements of the system. In order to do this properly, it was necessary to define a logical architecture separate from the physical/implementation architecture derived from the One-Wire diagram. Even though the logical and physical structures were separate, they still used common elements of definition - so that both structures would maintain synchronization of interfaces for example. Separating the logical and physical structures allows for different views to be managed for the appropriate stakeholders. In the SCMM model the physical view is placed in the context of the vehicle itself and is used to generate the One-Wire diagram for CSI cable procurement and vehicle integration purposes. The logical structure in SCMM is used to develop the behavioral analysis that supports the bulk of the systems engineering deliverables such as requirements, traceability reports, CONOPS, and software allocations. Using the validation capability of Magic Draw for SysML & UML correctness allows the system modeler to properly describe complex behaviors unambiguously.

Analysis & Traceability

Due to the rapid development of behavioral elements in the SCMM system model there was a need to go back and understand the relative maturity of these behaviors. Magic Draw provides the user the ability to create tables, matrices, and define queries to help understand the current state of the model. For example, a list of use cases and their owned behaviors is helpful to understand how many use cases have been decomposed. It is important to review model content in this way as there should be maturity from a starting point, not new developments in disjoint fashion. One would hope to have the system model look more like a growing tree rather than a field of weeds.

Shortly after the implementation of a System Model using SysML in Magic Draw, SCMM started developing a Software Model using UML in Magic Draw. Since the two models are compatible via the UML Metamodel, the model elements could be related to each other and reused for modularity. An example of this is the operation to method relationship that can be defined across models. The method is defined as the behavior owned by a block that executes in response to a particular stimulus, specifically when a request is made via a provided behavioral feature such as an operation [2]. For example, in the SCMM system model there is a block that represents a touchscreen display that owns an operation of “select gear”. “Select Gear” is an operation that satisfies a requirement of being able to change the vehicle gear from the crew station display and is implemented via software. Therefore, the “Select Gear” operation owns a method defined as a sequence diagram in the SCMM Software Model that describes what happens after the user invokes the behavior via the touchscreen display operation. Using this method relationship between system operations and software implementation, SCMM system requirements can be traced down all the way to the software level for verification & validation purposes.

Model Organization & Navigation

There are many techniques to generate reports from Magic Draw for a given set of desired artifacts. TPM Architecture has utilized the automated report generation capabilities of Magic Draw in order to rapidly generate the latest content from the system model for stakeholders. With a few simple steps, the system model can generate an entire report based on a predefined template with all the latest model content placed in the desired location within the document. Of course, when we do this document generation what we’re really doing is contradicting our model based approach.

It is possible to use the modeling tool itself to review architecture artifacts in a way that is understandable to the untrained user. It is important for the system modeler to perform due diligence on maintaining a user friendly navigation through the desired model views. One way this can be accomplished is with the usage of the Content Diagram in Magic Draw. The Content Diagram allows the modeler to create a webpage-like user interface on which diagrams (including tables, matrices, etc.) can be displayed as hyperlinks. In the SCMM system model there is a high level content diagram for each high level capability for the system that essentially serves as a table of contents that can be navigated forward and back from high level use case diagrams to sequence diagrams within the software model.

EFFECTS & LESSONS LEARNED

During the course of the transitioning the SCMM project towards a model based approach, TPM Architecture absorbed an enormous amount of reusable model elements, interpersonal relationships, and best practices. With the SCMM project team spanning various organizations within TARDEC (TPM, VEA, SEC, CSI, GVR) the system architecture development was gathered and reviewed from many different stakeholders.

One of the most valuable effects from the SCMM system model was the volume of reusable signals, interface blocks, and terminology that was promoted to the TARDEC Library and accessible to any project on the server. The ISAMG is continuing to evolve with more support across organizations within TARDEC that have contributed to common model elements as a result of the efforts of the SCMM project. An important lesson learned from this collaborative effort is that modeling in a vacuum is a recipe for disaster. The technique of covert modeling should be deployed during engagements with SMEs but the model content should always be presented to the SME for concurrence. With this in mind the system model should be sufficiently organized in a way that promotes ease of navigation for the model reviewer. If we can effectively create a user friendly environment within the model that all stakeholders can navigate, then we can truly begin to demonstrate the utility of the model based approach and stop generating vast amounts of redundant information in the form of standalone documents.

REFERENCES

- [1] Object Management Group (OMG), “What is SysML?”, <http://www.omg.sysml.org/what-is-sysml.htm>, 2017.
- [2] S. Friedenthal, A. Moore, and R. Steiner, “A Practical Guide to SysML – The Systems Modeling Language”, Elsevier Inc., Waltham, MA, 2012.