Model-Based System Engineering and Software System Safety Workshop

by Barry Hendrix, Saralyn Dwyer and Dave West
Huntsville, Alabama

The G-48 System Safety Committee sponsored a Model-Based System Engineering (MBSE) and Software System Safety (SSS) workshop, hosted by A-P-T Research, Inc. (APT) in Huntsville, Alabama, on May 2-3, 2017. The idea of this workshop evolved at the 34th International System Safety Conference (ISSC) in Orlando, Florida, during presentations and a paper by Barry Hendrix, which noted that the MBSE needs to include more system safety and software system safety processes. An action recorded under urgent-need topics by International System Safety Society (ISSS) Fellow Dave West at the G-48 meeting in Orlando resulted in volunteers to host and coordinate the workshop.

The MBSE SSS workshop consisted of a panel of seven subject matter experts. Approximately 40 attendees were present and more than 70 people viewed the workshop via a NASA live video streaming feed. The MBSE SSS panel consisted of Barry Hendrix, APT; Dr. Fayssal Safie, APT; Dr. Donna Havrisik, Government Agency System Engineering; Josh McNeil, AMRDEC Software Engineering Directorate (SED); David Arterburn, University of Alabama Huntsville; Joe Hale, NASA; and Paul Gill, NASA. Many attendees were from local Redstone Arsenal agencies, such as AMCOM, PEO Missiles & Space, and the Missile Defense Agency. Several contractors from companies within Cummings Research Park also attended. Special out-of-town guests included Peggy Rogers, U.S. Navy Software System Safety Technical Review Panel (SSTRP); Bob McAlister, U.S. Air Force; and Lynece Pfledderer, Lockheed Martin (LM), along with five other LM attendees from Texas, Florida and Connecticut.

Background

The need for a MBSE SSS workshop became increasingly apparent in recent years, with the proliferation of Department of Defense (DoD) programs with various forms of MBSE tools and processes to add value and benefits to product lines. While many larger agencies, contractors and programs know how to plan and execute model-based tools and processes to influence complex system development and design, the system safety, software safety, reliability, mission assurance and human systems integration disciplines and processes were not always integrated proactively to achieve goals and objectives. Because of misconceptions and lack of understanding of model-based concepts — particularly at the acquisition management and program office levels — many programs did not sufficiently address or include needed disciplines and processes during MBSE planning and implementation. However, other programs recognized that MBSE is simply a modern toolset with better methods of systems integration. These programs clearly realized the value and benefits — from a business case — of capturing and influencing requirements that drive system architecture, design and engineering documentation in graphical models. These well-run programs also embraced, included and *required* multiple disciplines to participate in model-driven processes with enormous benefits. The only management concern expressed about model-based approaches are the perceived additional up-front cost and how to capture metrics for earned value (EV).

There is strong agreement throughout industry that large, complex systems with safety-critical functions — particularly the software-intensive systems of systems — are well suited for MBSE with software system safety as a vital part of the model. Some may ask, "Why is that so?" The most apparent answer is that the more complex a software system gets, the more difficult it is to perform any accurate functional hazard assessment or software safety analyses without the help of tools and models. Typical models in software yield many outputs in an easy-to-understand format, instead of in language that's often hard to interpret. By contrast, MBSE and model-based software aspects and software safety subsets allow numerous ways to express exactly how the software behaves against a set of expectations, requirements, goals and structured notation.

Typically, the model can yield use cases, sequence diagrams, functional flow diagrams, behavioral diagrams, safety significant functions, functional threads, safety attributes, safety features, traceability of safety requirements, and many other inputs and outputs when thoughtfully and purposely planned. Models have no value in the safety domain without system safety and software safety engineers having access to and understanding the model as a tool to help specify safety test cases and to verify safety requirements. This is important for mitigating hazards and mishap risks, and for yielding better, more objective safety evidence for safety assessment reports (SAR), safety cases (SC) and system safety assessments (SSA).

Workshop attendees who have used various MBSE tools and methods gave positive feedback that they helped reduce risk, particularly when transitioning from system design to software design. The requirements' decomposition and implementation of code are where the majority of expensive re-work frequently occurs. Aside from the more obvious benefits of a model helping to depict the big picture and value of influencing a safer, more reliable design, cost reduction and better integrity are the strong points of MBSE in general. Many engineers consider models of various forms a more eloquent and confident way of developing complex systems with fewer risks. This is assuming that individuals, smaller teams and larger collaboration groups use carefully chosen tools for the intended engineering environment and integrate the models effectively and efficiently along the critical path.

Without endorsing or evaluating any tools, models or modeling languages — or linking to any specific complex programs — DoD, NASA and contractors have a wealth of trademarked systems, such as Rhapsody, IMB Rational DOORS, UML, SysML, SIMULINK, MatLab, SCADE and many more. Readers are encouraged to research these online for familiarity, as this paper is not intended to have any opinion on these successful, market-driven tools. However, should a software system safety engineer be assigned to programs that use these and other tools, it wise to learn how they work and can benefit from standard software safety tasks, activities and work products. Some may feel they can run an adequate software safety program without them. Nothing could be further from the truth if the assigned program uses the tools and models to develop software use cases and a requirements repository. In those cases, system safety/software safety and other engineers must be part of the process. The issue for some is, "How do we do that?"

The short answer is to recall previous evolutions and adaptations of system safety, from a 1967 "hazards and risk" — based MIL-STD-882A environment to the current needed paradigms of requirements-based, functional-based, highly complex and software-intensive, model-based environments. Systems have become more complex, highly integrated and fused with multiple systems of systems, all with various forms of interoperability. MBSE is becoming the norm to help integrate these systems. System safety must acknowledge, accept and adapt (i.e., plug into) the model to be successful. It is a tool and a process, much like past models. For example, a fault tree is a model with which system safety has been comfortable since the 1980s. Why would we not accept and use other proven tool sets and models — especially those clearly mapping and depicting safety functions, logic, states/modes, failure/fault conditions and system behavior — to help safety aspects beyond standard hazard identification, safety analysis, safety assessment process and risk mitigation process expressed in words and matrix worksheets?

Goals and Objectives

The MBSE SSS workshop's goals and objectives were to present how models can be used to integrate certain parts of the system engineering process, software engineering process, and safety engineering process for system capability, requirements and functional domains. Software system safety was the central focus, since many models are in the software domain. These models can help break down and diagram vital system functions, behaviors and sequences, yielding safer, higher-integrity input and output in the safety-significant (safety-critical and safety-related) domain. This output data yielded from models incorporating safety inputs, attributes and facts can be used as objective safety evidence to present a viable safety case or refuting argument.

During the first day of the workshop, the following topics were presented, discussed, debated and captured. On the second day, the panel and attendees collaborated, exchanged ideas, asked questions and gave testimonies. Key points were summarized.

Model-Based System Engineering and Software System Safety Concepts, Goals and Objectives

Barry Hendrix presented the concepts, goals and objectives of a model-based system. MBSE development has emerged over the past decade as one solid solution proven to dovetail with software engineering/software system safety goals and objectives. Various agencies, such as the International Council on System Engineering (INCOSE), DoD and NASA, are implementing various forms of MBSE. Since traditional system safety may not be adequate for emerging and evolving system of systems and paradigm shifts, collaboration on model-based development and software system safety policy and best practices for complex and critical systems needs to finalize into recognized guidance. There are many system safety and software system safety advantages of moving toward MBSE, to including:

- MBSE can show the "big safety picture" and explicit safety functions, safeguards, safety features with easy-to-interpret sequence flow diagrams, and behavioral flow diagrams of safety-critical functions.
- MBSE improves engineering collaboration, teaming and communications across domains — same core representation — for safety documentation.
- With MBSE, system engineering, software engineering and safety engineering processes and actual FUNCTIONS and normal/failure CONDITIONS

can be visualized versus word interpretations that can be vague and ambiguous.
- MBSE allows proposed changes (safety changes) to be evaluated.
- MBSE provides more consistent safety documentation, and traceability improves technical integrity.
- With MBSE, already validated auto-code generation using the tools to perform them can be better analyzed in a model-based setting — a plus for safety.
- MBSE, MB SwEng and software system safety must be integrated into a "Golden Triangle" for success.
- DoD, with the help of INCOSE and large prime contractors, is in transition to current and emerging engineering methods to keep from falling behind.
- Software safety involvement and contributions require open-mindedness and transitioning from older traditional methods. Times, technology and environments are changing, and system safety and software safety must adapt to help make these changes work.
- Cultural changes and management buy-in are needed. This involves convincing ourselves that better, evidence-based safety is needed and emerging methods will really work.
- In any safety-critical program with MBSE, Model-Based Development (MBD) must ensure an adequate System Safety Program Plan (SSPP) and Software Safety Program Plan (SwSPP) (WHAT) with subordinate processes (HOW) are developed.
- Flexible policies, best practices and processes are needed for integrating model-based aspects of systems engineering, software engineering and system safety/software system safety.

Model-Based System Engineering Trends

Model-based system engineering trends were presented by Dr. Donna Havisik and Lisa Laurendine. Model-based engineering is the formalized application of modeling (both static and dynamic) to support systems design and analysis throughout all phases of the system lifecycle, through the collection of modeling languages, structure, model-based processes, and presentation frameworks used to support the discipline of systems engineering in a "model-based" or "model-driven" context. To date, there are many educational curriculums being established, with new projects continuing to evolve in the new environments. To meet stakeholder needs, the Lifecycle Steering Committee is defining, and continues to refine, the Lifecycle Modeling Language (LML). To meet future needs, research is ongoing on what a major milestone review will look like in a MBSE environment, and stakeholders are endorsing concepts to explore model-centric engineering.

MBSE Programs at the University of Alabama Huntsville and Other Thoughts

MBSE programs offered by the University of Alabama Huntsville (UAH) were outlined by David Arterburn. Current changes in technology are occurring at a much faster pace than changes to standards. As systems become more complex, traditional systems engineering, contracting methods and airworthiness processes and standards may not be sufficient to ensure the safety of the platform while supporting the acquisition process throughout the lifecycle. Traditional methods can also drive weight into the design, as well as cost, without significantly improving the safety and mission effectiveness of systems. This may drive developers to virtual prototyping of technology while assessing system-level performance. The acquisition process must clearly articulate the buyer's intent in terms of mission effectiveness and capability to create affordable systems, understand the trade space, and better assess cost and schedule risk starting at source selection through airworthiness determination and fielding. This, in turn, will provide early problem identification in the materiel development process.

UAH is leading the field with its complex systems integration lab. This lab provides the necessary environment for (1) providing the needed synchronization required between trade studies (technology push) and systematic operations analysis (technology pull); (2) executing trade-off methodology leveraging existing tools available; (3) integrating new tools and methods as they become available; (4) integrating a broad range of tools into a singular environment and (5) providing the necessary methodology and stakeholder environment for successful execution. The lab also provides a local and low-cost location for the government and industry collaboration necessary to support decision making throughout the acquisition process.

Planned UAH focus areas include program management dashboards that will use one model and allow entities to get updates on the status, cost capability analysis evolving from requirements, and integration of safety products with requirements development.

Software Engineering Directorate (SED) Model-Based Development Software Safety Guidelines

Josh McNeil, chief engineer, Aviation Division, U.S. Army Aviation and Missile Research Development and Engineering Center, described current SED MBSD guidelines. The SED Aviation Division recognizes the effectiveness and utility of model-based software development techniques for reducing system errors. A majority of software safety, as well as programmatic development, risk issues are caused by inadequate software requirements and design. SED has developed a

Model-Based Software Development Safety Guidelines handbook. Its purpose is to provide guidance to the developer and Army assessor for safety-related software systems on the underlying issues and concerns associated with meeting software safety requirements, such as DO-178C objectives. The guidelines are focused to provide guidance to the Army developer and Software Airworthiness and Software Safety approvers on MBD process steps and artifacts. They support compliance with requirements and guidance, including the Army Aviation and Mission Command (AMCOM) Software System Safety Policy, AMCOM Reg 385-17, SED Software Engineering Evaluation System (SEES) Program Managers Handbook for Aviation Software Airworthiness (PM-HASA), and RTCA DO-178C and DO-331. The guide covers model-based software requirements, design, code, verification and tools. The guide also addresses generic issues related to model-based processes in alignment with FAA/EASA/RTCA guidance material and Army Aviation Software Airworthiness. SED selected two case studies to support the understanding of the role of the auditor for a SCADE project and a Simulink project.

> "Research literature for MBMA (Model-Based Mission Assurance) shows that there is a tremendous amount of work to link MBMA and MBSE (Model-Based System Engineering). Linking designs to reliability analysis, reliability analysis to safety analysis, and safety analysis to MBSE designs can provide the framework to support model-based mission assurance activities."

DO-331 Model-Based Development and Verification Supplement to DO-178C and DO-278A

Barry Hendrix and Josh McNeil presented a briefing prepared by APT's Software Designated Engineering Representative (DER) Leslie Alford. The objectives for the DO-178C suite of documents and supplements include promoting safe implementation of aeronautical software, providing clear and consistent ties with the systems and safety processes, addressing emerging software trends and technologies, and implementing an approach that can change with the technology. The purpose of the supplement is to provide industry-accepted guidance for satisfying airworthiness requirements for avionics equipment. It presents compliance guidelines for software and criteria consistent with civil certification authorities. By treaty agreement, this applies to NATO nations and any other countries recognizing this set of guidelines for aviation software. The results provide agreed-upon criteria for airworthiness certification requirements for software that do not differ from one person or certification authority to another. It allows for recognition of aircraft model capability by air traffic control for airspace access and interoperability.

The RTCA SA-12 Safety Committee and other committees associated with the re-write of DO-178C and DO 278C determined that DO-330 Tools and DO-331 Model-Based Development and Verification add much value for software design and technical integrity (going much farther than safety alone). The guide deals with identifying the "safe-subset" use of MBD technology to be used in safety-related applications and using suitable graphical engineering methods to design a software system. Clear distinctions are made between two types of graphical models: specification models and design models. Determining which artifacts will be in a model drives the determination of applicable objectives and activities. The MBD data items (beyond the normal items) expected in a program include model planning, model standards and techniques, model element libraries, model coverage and model simulation.

Moving NASA/MSFC Toward a More Model-Centric Organization

Joe Hale and Paul Gill of the NASA Marshall Space Flight Center (MSFC) presented on how NASA is moving toward a model-centric organization. MBSE is much broader than a single modeling approach or tool. There are many tools needed to develop a full system model. MSFC is investigating model-based engineering (MBE) architectures and the set of system modeling approaches to fully represent the system. Proposed activities include cataloging on-going efforts, gathering stakeholder needs/expectations, sharing knowledge, holding tutorials and developing case studies. MSFC has developed a MBSE/MBE maturity matrix. The approach used in development included the identification and decomposition characteristics and factors that describe or comprise a fully model-centric organization's capability. Each row of the matrix reflects increasing levels of capability for that specific factor or attribute. Key features include providing a strategic vision to

guide tactical planning; the ability to track progress cell by cell; visualization of the factors or attributes being worked; the ability to track the status of work underway, completed and planned; and the ability to plan and prioritize future efforts cell by cell.

Model-Based Mission Assurance (MBMA

Dr. Fayssal M. Safie of APT presented a briefing developed with Dr. John Evans of OSMA, NASA Headquarters. The briefing focused on the MBMA concept in a MBSE environment and addressed what safety and mission assurance organizations need to do to participate and integrate into the MBSE environment. There are several major MBSE/MBMA benefits, including information consistency; propagation of changes; ease of communication and maintaining current project baselines; cross-training and experience for engineers; enhanced stakeholder communication; visibility into information gaps and system design integrity; rigorous traceability from needs through solution; and reduction in the number of requirements, early/ongoing requirements validation, and design verification.

NASA OSMA has developed an approach to provide flexibility while focusing on a vision that is rooted in technical objectives, rather than specifying products and processes. This approach uses the development of objectives hierarchies with supporting strategies for implementation. The results promise the potential of improved effectiveness, flexibility and compatibility with MBSE. The objectives-driven approach starts with a single, top-level objective of a successful project. This is then broken down into sub-objectives, much like the development of any systems engineering hierarchy. Integral to this structure, however, is the use of *strategies* to convey information about satisfying objectives. The strategy or strategies that couple with it identify non-process-specific methodologies for satisfying the objective.

Research literature for MBMA shows that there is a tremendous amount of work to link MBMA and MBSE. Linking designs to reliability analysis, reliability analysis to safety analysis, and safety analysis to MBSE designs can provide the framework to support model-based mission assurance activities.

The mission assurance community must get engaged and integrate with the MBSE communities. Assurance organizations may need to define new roles and develop new skills, and their products may need to be different in a model-based environment.

Summary of Workshop Key Points

After much professional debate and dialogue before, during and after the workshop presentations and collaboration discussions, the group agreed upon the following key points:

- MBSE is necessary for program consistency and understanding in complex systems
  - Consistent documentation
  - MBSE pulls together system of systems — integration
  - Leverages greater efficiencies
- Acquisition process/strategies need to leverage current technologies
  - Have to validate strategy at acquisition level
  - MBSE provides more than a set of requirements does
- Safety product must be integrated with requirements development
- Selection of tools and how they interface is important
- Early buy-in from all stakeholders must be achieved
  - Program offices may not understand MBSE yet
  - Applicable Boards — Joint Services
- The MBSE common definitions must be matured
  - Partner with INCOSE and contractors who have already "plowed ground"
  - Need good guidance/guidelines — best practices
- Assurance organizations need to define new roles, develop new skills and define products for model-based environment
  - Need to bring all the disciplines into this (system safety, reliability, human systems integration, etc.) and build a business case for value and benefits
- Consider benchmarking — RTCA DO-331 is an industry standard. INCOSE has the lead and guidelines published.
- Develop metrics for evaluating the fidelity of the model over the lifecycle
  - Model selection
  - Model parameters
- Base best practices on
  - Inclusiveness in the model
  - Lessons learned
  - Industry-proven best practices
- Real-time impact
- First task in MBSE is to have the same or accurate set of data
- Put existing requirements into the model
- Informal team of volunteers to exchange ideas and develop guidelines
- Interoperability of functions, systems, system of systems
- Link the disciplines (safety, software safety, etc.) to the system

The group concluded that the MBSE way forward should include policy recommendations and the development of best practices and process improvements.

Workshop Conclusions and Recommendations

In summary, the G-48-sponsored MBSE workshop, under the leadership of Dave West, drew many presenters, participants and others willing to learn. It was well received with positive feedback from most attendees. However, where do system safety professionals go from here?

While many participants were experts and practitioners of model-based concepts (and true believers), others were in various stages of learning about the concept and were not necessarily receptive of paradigms and modern methods, feeling comfortable with existing ways. The natural reluctance and resistance to more modern methods, techniques and processes — especially adding model-based tools with the absence of standards, and few guidelines — seemed to bother some. Others looked at it as "just one more thing we have to learn on our own" — on-the-job training — in a computerized and digital thread world, like learning how to use FTAs to quantify risk and DOORS to document and trace safety requirements.

The workshop concluded on a positive note, as many saw that the model can be helpful in safety tasks such as Off Nominal Safety Testing from a Model, conducting Failure Modes Effects Testing (FMET) from a model, as well as use cases, behavioral diagrams, functional flow of safety-critical functions, and threads and other areas well proven. However, some common concerns will not be fully answered until those in industry and INCOSE meet with the G-48 Committee, Fellows of the International System Safety Society, and others who know the technology to weigh in more on developing best practices, guidelines and standards with precise language for MBSE.

It was concluded that some big contractors on large, complex programs know how to do model-based system safety, but had to learn the hard way by plowing new ground. Many agencies are requiring MBSE, or contractors are doing it without being required, as the government can't dictate "how" a contractor meets requirements.

On May 4, the day following the workshop, the G-48 System Safety Committee received a briefing and is considering a future course of action in several areas. One area is to form a small subcommittee to meet with INCOSE and others who currently have established guidelines on MBSE standards, but lack any details on system safety or software safety aspects. The only action was the recommendation that this article be written and submitted to *Journal of System Safety* (*JSS*).

In conclusion, a few concerns and questions are certainly valid ones:

1. How do the technical experts who know the value and benefits of MBSE and system safety integration convince others from a business case?
2. How does system safety (or any discipline) communicate in a convincing way to those program office leaders and planners, including DoD acquisition authorities and certification authorities, that MBSE funds and budgets need to be allocated?
3. How do we address model-based safety, as well as additional burdens and tasks, short of having an affirmative statement in a DoD standard (or equivalent commercial standard, such as GEIA-STD-0010 *Standard Best Practice for System Safety Program Development and Execution*)?

Acknowledgements

We thank the participants in the MBSE SSS workshop, as well as their sponsored companies and agencies, who came together to help resolve pressing issues in system safety. Their input in the form of presentation documentation and verbal presentations to a large audience contributed as the primary source data used in this article.

A copy of all the workshop presentations and findings can be found at https://www.apt-research.com/MBSESSS/Agenda.pdf.

About the Authors

Barry Hendrix is a Fellow with the International System Safety Society. He is a retired System Safety Technical Fellow from Lockheed Martin who came to work for A-P-T Research, Inc., in 2015. His current assignments as the Sr. Principal for Software Safety include supporting several government agencies and program offices to influence software system safety on complex software-intensive programs.

Saralyn Dwyer is a Fellow with the International System Safety Society, and Vice President and Director of the Safety Engineering and Analysis Center (SEAC) at A-P-T Research, Inc.

Dave West is a Fellow with the International System Safety Society and has more than 20 years in system safety leadership with Science Applications International Corporation (SAIC).