Model-Based Systems Engineering for System Safety: An Introduction

by Patrick R. Oliver
Orlando, Florida

Model-based systems engineering (MBSE) has gained momentum as the predominant method of analyzing and deriving system requirements, as well as of verifying and validating system performance. Over the years, several frameworks have gained prominence as approved methods and formal techniques to model systems. MBSE technology continues to gain popularity within the systems engineering domain, especially in markets of complex systems. To remain relevant within the context of concurrent engineering, it is advantageous for system safety engineers to learn how these techniques are affecting system design so that safety is addressed within system development. This paper provides an overview of MBSE in theory and practice, and provides high-level details on how the system safety engineer can use these methods for optimum impact in affecting safety design.

Introduction

The practice of systems engineering has been enabling technology for the development of complex or mission-critical systems. As systems engineering techniques have evolved, methods have adapted from document-based systems to more efficient model-based approaches. The purpose of this paper is not to provide a complete familiarization with model-based systems engineering techniques; rather, it is intended to provide the system safety practitioner with an understanding of the different ways MBSE techniques may be applied to a development activity and how system safety integrates within a MBSE development activity. At this paper's conclusion, it is intended that the reader be able to integrate, use and capitalize on these techniques to enhance the effectiveness of the system safety program to reduce system safety risk.

Model-Based Systems Engineering Application

In the simplest sense, a "model" is an abstract representation of a real objective item [Ref. 1]. The concept of modeling is not new. Two-dimensional drawings and schematics, as well as three-dimensional models, have been used throughout history to represent the objective of the design. Models allow engineers to manage system complexity through abstraction, simplifying the system concept to allow analysis and the ability to communicate its attributes. Models may be formal or informal, ranging from sketches on the back of napkins to formally released engineering. Models may be simple or complex, depending on the system complexity or necessity of the developer. During the last several decades, we have seen modeling processes institutionalized through various frameworks and defined in increasing sophistication of modeling languages. The evolution of systems modeling has had a profound impact on how we perform engineering tasks and provision system safety to share in the benefits, if care is given to its integration.

What is Model-Based Systems Engineering?

Systems engineering is defined by the International Council on Systems Engineering (INCOSE) as follows:

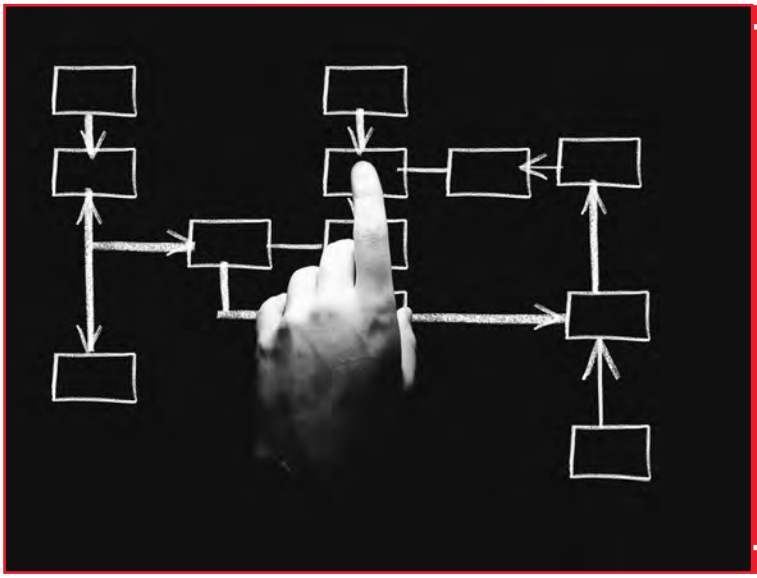
"Systems engineering (SE) is an interdisciplinary approach and means to enable to the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost and schedule, performance, training and support, test, manufacturing and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets user needs" [Ref. 2]

INCOSE goes on to define model-based systems engineering as:

"Model-based systems engineering (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" [Ref. 3].

MBSE Processes

MBSE processes are implemented to meet some standard of systems engineering. Such systems engineering



> "Models allow the analyst to more fully express the implications of a requirement as it translates to a lower level of abstraction representing the intended solution. The model representation reveals functional and physical relationships between elements that can be probed and tested for validity in meeting higher-level requirements. At the same time, relationships between model elements are defining lower-level operations and interfaces of the solution."

process standards include EIA 632, ISO 15288, IEEE 1220 and CMMI. Several frameworks decompose these process standards to formally define approaches to architecture development including conceptual elements, terminology and artifacts. Examples include the Federal Enterprise Architecture Framework (FEAF), the Department of Defense Architecture Framework (DoDAF) and the Zachman Framework. Modeling methods define the "how to" of the approach to developing model artifacts. Methods have evolved from behavioral methods such as IDEF0 to object-oriented (OO) methods (object-oriented methods are currently the most popular). Methods are applied to specific languages, Systems Modeling Language (SysML) and Unified Modeling language. [Ref. 1]

MBSE Objectives

Architecture Definition — INCOSE defines architecture as the "fundamental concepts or properties of a system in its environment embodied in its elements, relationships and in the principles of its design and evolution" [Ref. 2]. Architecture is the higher-level abstraction of the system that defines and amalgamates the detailed design of the system. One important aspect of MBSE is model-based architecture (MBA) development. MBA provides framework and language conventions to standardize architecture definition for the purposes of repeatability, portability, quality and efficiency.

Requirements Decomposition — Models may be the expression of higher-level requirements within the context of the system solution. A model may also be used as an analytical tool to derive requirements. In most cases, the model does both. Models allow the analyst to more fully express the implications of a requirement as it translates to a lower level of abstraction representing the intended solution. The model representation reveals functional and physical relationships between elements that can be probed and tested for validity in meeting higher-level requirements. At the same time, relationships between model elements are defining lower-level operations and interfaces of the solution. Capturing these in terms of requirements decomposes the higher-level requirements to the lower-level requirements of the system design.

Verification — The model representation of the solution also provides methodologies for verification. Models can aid in the development of test constructs that can be used to prove-out principles and assumptions in the design. Engineers are able to use models to test underlying assumptions of the model, or to test hypotheses of system behaviors or performance. In advanced executable models, developers can verify requirements in a model before developing hardware.

Benefits of MBSE

There are several benefits of an MBSE approach. Proponents claim improvements in efficiency, communications, quality and productivity to enhance system development. Such claims are readily supportable. Developing models instead of hardware limits the need to generate multiple iterations of prototype hardware, since designs are matured as models. This allows corrections and improvements to be made in the model rather than in hardware, resulting in fewer "design-fix-design" cycles. MBSE also supports productivity, since model-driven environments are predicated on rapid

collaboration and real-time review of design concepts. Because models can be scalable and portable means of design communication, they add visibility to other disciplines who contribute to the design. Enhanced system design visibility allows other disciplines to see aspects of the system being developed as they are developed. Development costs may be reduced through testing system assumptions in a model. It should be noted that a model is only as good as its underlying assumptions; therefore, a model's validity in representing its intended domain should be frequently challenged and assessed.

MBSE and System Safety

The objective of system safety within an MBSE environment is to apply the technical and management principles of system safety engineering so that the resulting system meets the high-level objectives of the safety plan. Just as system safety is part of a traditional systems engineering effort, system safety may also be applied as an MBSE effort to achieve safety objectives. In principle, the application of system safety principles is the same as those applied to traditional systems engineering but is augmented by the differences in MBSE techniques. A safety order of precedence [Ref. 4] prioritizes elimination and design mitigation of hazards as the preferred means of reducing system risk. Because a cohesive MBSE effort will affect both architecture and design, MBSE should be a focus area of influencing hazard mitigation through the order of precedence. If MBSE methods and techniques are being employed effectively, it is necessary for the system safety practitioner to learn how to integrate hazard analysis and assessment techniques to capitalize on the benefits these methods and techniques have to offer.

As a system develops, decisions are made to allocate lifecycle cost to the system. Those allocated costs compound over time. If the decisions to allocate those costs are made without regard to system safety objectives, late discovery of hazards will incur a level of technical debt that must be "paid down" to mitigate the hazards. This concept is analogous to the concept of requirements defects. In this case, the lack of safety requirements that results from unidentified hazards are the defects that incur costs. MBSE shares the common goal of reducing the quantity and severity of requirements defects by formalizing the processes of flow-down, allocation and derivation. If the development activity is leveraging MBSE techniques to reduce the impact of requirements defects, then system safety benefits from a mutually beneficial relationship sharing the common goal of reducing system technical debt. MBSE facilitates the rapid decomposition of stakeholder needs into a solution. Therefore, system safety must integrate with these techniques as soon as these methods are instantiated to be effective. The early identification of architectural-level safety requirements will define the necessity of safety requirements, hazard controls and major system safety features. These are implemented in a cost-effective manner when done early in the design process rather than later, thereby reducing the cost impact of the safety program and its efficiency in reducing safety and programmatic development risk.

System Safety Prerequisites

A preliminary hazards analysis [Ref. 4] is one of the most necessary prerequisites to any system safety effort. It provides the analyst with a basic understanding of how the system contributes to system mishaps and their relationships with causal factors. Because of the importance of the preliminary hazards analysis, it is recommended to be the first step, since having a basic understanding of hazards is key to making the decisions described here. Additionally, it is necessary to know what the end objective is prior to beginning. What does "done" look like? This requires the system safety practitioner to address several questions that shape how system safety techniques will map to MBSE methods.

Selecting an Accident Causality Model — To apply system safety principles to an MBSE approach, one must understand what accident causality model is assumed. Accident causality models can be differentiated into two basic types: cause and effect and systems-based. Cause and effect models [Ref. 5] generally ascribe accidents being the result of cause and effect chains. Such models are premised on the fact that for every effect, there is an antecedent cause. The safety argument is predicated on the analyst identifying all the causes and their associated mitigations. The safety assessment is based on the effectiveness of mitigations to prevent causal factors from contributing to mishaps. In the systems-based accident model [Ref. 6], system mishaps are emergent from uncontrolled system behaviors. The distinction between these causality model types becomes important when evaluating model-based architectural and design artifacts with respect to meeting high-level safety objectives. The techniques associated with cause and effect models will need some definition of design, and hence a higher level of maturity to be effective. Systems-based models

“Through experience, we know that a system safety program is most effective when implemented early in development. However, there are plenty of instances when this is not the case. For this reason, it is important for the safety practitioner to ask the question, ‘What stage of development are we in?’ The answer to that question determines how safety will be applied with respect to an MBSE environment.”

are more functionally oriented and may be employed earlier in the architectural definition when system concepts are more abstract.

Behavioral Contribution to System Mishaps — One important consideration in selecting an accident causality model is how system behaviors contribute to system mishaps. In general, there are two basic behavioral constructs to be discerned. The first is that in which the system must be able to fail to a finite and relatively static safe state. Cause and effect techniques such as FTAs and FMECAs can be effectively applied to fail-safe systems. The second behavior construct is that the system must be able to operate continuously to maintain high levels of availability to mitigate safety risk. Aircraft situational displays and fly-by-wire control systems are good examples of high-availability safety systems. Systems-based or functional techniques may be a better fit for such applications since they focus on the definition of constraints (in terms of requirements) that enforce safe system behaviors [Ref. 6].

Determining the Stage of Development — It is also important to understand the stage of development in which the safety program is being implemented. Through experience, we know that a system safety program is most effective when implemented early in development. However, there are plenty of instances when this is not the case. For this reason, it is important for the safety practitioner to ask the question, “What stage of development are we in?” The answer to that question determines how safety will be applied with respect to an MBSE environment. Generic lifecycle stages include concept, development, production, utilization and retirement. If the safety program is implemented in the early stages, such as concept or development, there is substantial opportunity to use the results of safety analysis to influence the resulting system being developed. When implemented early, the safety analysis may be integrated with MBSE activities to identify physical or functional hazards during the high-level architectural planning stages. This allows early definition of safety requirements and influence over the design prior to the allocation of development costs. The system design is the most malleable during these stages and making major design decisions at this stage causes the least amount of impact. However, starting early is not always possible. Many system development activities start out with major elements of the system already developed, if not developed completely. In such cases where the safety program is implemented in production, utilization or retirement, the safety program takes on more of a reactive role. In these cases, there may be less advantage to investing safety resources in MBSE activities, since most of the development costs have already been committed to the solution and there are fewer design options to consider.

Utilization of MBSE — To be effective, the safety analyst must understand to what extent the development effort will utilize MBSE techniques. There is a range of utilization between document-based and model-based development approaches by which MBSE techniques will be applied. The type of approach may not be ex-

plicitly called out in the systems engineering management plan, so it is up to the system safety practitioner to recognize the type. In a document-based approach, the focus is on generating plain-language requirements and art-based drawings. Such an approach is driven by a "specification tree" and a "drawing tree" to delineate how the system will be decomposed from the high-level performance needs down to the details of design implementation. In a pure model-based approach, the model will define system decomposition and will also act as the center of the engineering repository. The safety practitioner must also understand the intent of how MBSE is to be applied. MBSE may be used as a design technique or a documentation technique. When used primarily for documentation, the MBSE effort intends to describe the system with various modeling artifacts. In contrast, using MBSE as a design technique intends to define the system. Therefore, the level of system safety's integration with MBSE will need to be proportional with the centrality of the model-based approach within a development activity to be effective.

Describe the Lifecycle — Lifecycle describes the system solution from inception to final disposition. All development programs will subscribe to a life cycle of some type. These may be familiar as the "V" model, "waterfall" model or "spiral development" model. Newer development models may incorporate Agile or Scrum methods [Ref. 7]. This paper is not intended to evaluate the virtues or vices of these models. Nonetheless, it is important for the system safety practitioner to understand how the lifecycle is intended to execute to align system safety objectives with overall development objectives. The lifecycle will define important milestones for design approval, verification and validation, material release, sustainment and retirement. Each of these milestones should be evaluated to determine what safety goals must be accomplished for the system to achieve high-level safety objectives. In addition, the point at which system safety enters the lifecycle makes a difference whether the safety program is proactive or reactive. Later insertion of system safety into development will result in a generally more reactive, assessment-based approach. Earlier insertion will result in a more proactive, influential approach. For this reason, there is more return on investment in integrating system safety with MBSE if inserted earlier rather than later.

Define How System Safety Objectives Align with the MBSE Effort — The system safety practitioner must determine how safety objectives will align with overall MBSE efforts. Understanding the depth and extent of the MBSE methods being employed will aid the system safety practitioner in determining the extent to which safety objectives will integrate with MBSE efforts. As a general principle, the safety program will benefit most from integrating with MBSE methods when they are tightly coupled with development activity. Further, it is recommended that system safety align and not work on stand-alone MBSE efforts. Stand-alone models and analytical work will tend to become disassociated from the main development activity, getting less attention from engineering decision makers and the requirements flow-down. Safety techniques should be implemented in a way that maximizes influence on MBSE architecture and design, providing connectivity to tracked hazards, safety-significant requirements or other traceable safety attributes mapped to model elements.

MBSE techniques are fundamentally employed in one of two major objectives — descriptive or definitive. Descriptive techniques use the model to help describe the solution to stakeholders. The purpose of model artifacts is mostly to convey an understanding of implementation. When the MBSE techniques tend to be more descriptive, system safety uses model artifacts as source information to substantiate or bolster safety assessments and analyses. When the MBSE goals are mostly definitive, the artifacts will be used to drive design and implementation. With definitive MBSE goals, the safety artifacts are integrated in *support of the model*. Safety analysis and hazard analysis information supports the development of model artifacts to influence the solution to safety objectives.

If the MBSE effort within the development effort is not highly indoctrinated or integrated, then the safety practitioner may consider employing MBSE techniques in a small-scale stand-alone effort, capitalizing on smaller framework segments and using limited model elements to support the safety analysis process. For example, the analyst may consider using sequence diagrams (e.g., UML® or SysML®) to work out dynamic safety requirements in a logic-based system component. It is recommended that such techniques be employed at the discretion of the system safety leader in support of the needs of the safety program.

Evaluating Tools and Methods — There are a host of tools available, all with varying levels of capability. Once the system safety practitioner understands how system safety will be involved with the overall MBSE,

then the details of connecting artifacts and analyses will begin to focus on the tools being used. The more integrated the system safety approach will be with the development MBSE effort, the more commonality it will need in the selected toolset. During the safety planning effort, the tools being used by other engineering efforts should be understood and evaluated to determine to what extent system safety will be interacting with them. This requires the system safety practitioner to be familiar with the capabilities and limitations of the tool, as well as the extent to which safety methods can integrate with them. Some MBSE tools are not more than elaborate drawing tools with templates and features that support specific modeling languages. Such tools tend to be used for descriptive modeling objectives but may be used for definition if users are disciplined enough to manage interrelated artifacts and traceability. Many of the sophisticated modeling tools have structural capabilities that retain not only the viewable images, but relationships between model elements. Structural models may be executable based on a variety of model domains and independent variables. Structural models are more suited for system definition goals and would therefore benefit from safety data input to the model through traceability or mapping utilities. Model execution may be in the form of a simulation that replicates aspects of system behavior. In executable models, safety artifacts can be captured from verification output data from the simulation or the executed model. With all these points considered, it should be noted that different engineering disciplines may be using different tools for different purposes. It should be decided, as part of system safety program planning, how tools will benefit system safety and to what priority should they be employed.

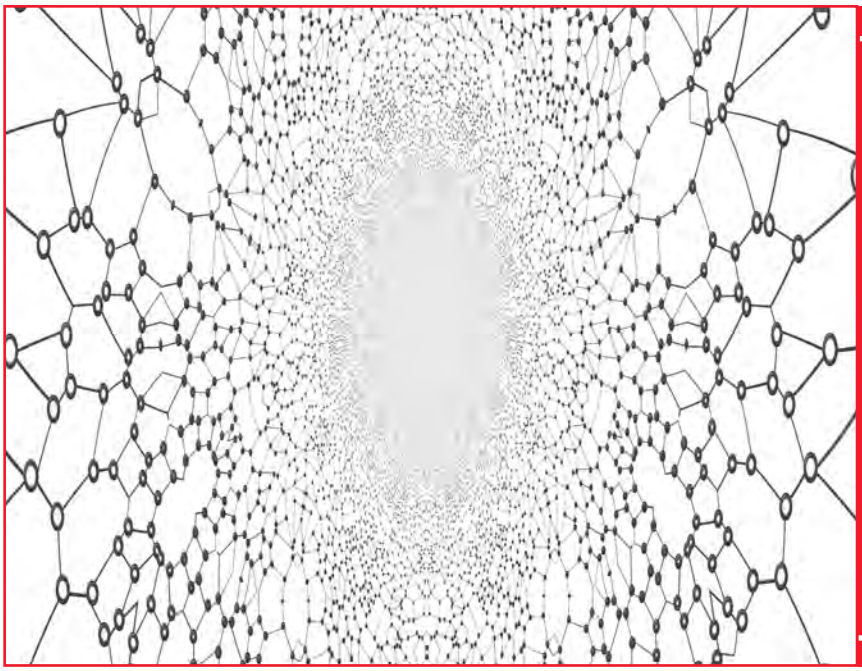
"When engineering a solution to meet stakeholder needs, it is imperative for the architect to ascribe what is within the system and what is outside it. Model representations will constrain the design problem space to what is under the authority of the design agent to determine the boundaries that constrain the solution. Once the system boundaries are defined, architecture provides a high-level organization and definition of constituent elements."

Integrating System Safety with MBSE

Once the previously discussed prerequisites have been accomplished, the system safety practitioner may begin the larger task of integrating system safety with the MBSE effort. The main areas of integration include architecture, design, requirements, verification and validation. Each of these areas of integration provides important means of influencing the safety of the resultant system. For each MBSE task or artifact, the system safety analyst must be able to answer three questions: 1) What are the hazard sources? 2) What are the vulnerabilities? and 3) What are the items to be protected? To be effective, the safety practitioner applies these questions to the MBSE objectives of architecture, design, requirements and verification.

System Safety Architecture in MBSE — MBSE adds formality to the architecture definition process through frameworks, modeling languages and constructs. Modeling representations will start with high levels of abstraction before committing resources to further define lower levels of abstraction. At the highest level, system architecture will provide definition of the system. Most formal MBSE methods will provide a high-level representation of the system to derive or document architectural decisions. Different languages (e.g., UML® & SysML®) will have taxonomies of specific views to capture such representations. Preliminary hazard analysis and functional hazard analysis can start concurrently with this part of the modeling effort. As soon as high-level diagrams are being constructed, they will start to point to physical and functional relationships that are capable of precipitating hazardous conditions. A look at these relationships from a system safety perspective will start to reveal how they can fail or incorrectly function in a hazardous way. Whether the system safety analyst uses a preliminary hazards list based on lessons learned or draws on experience experience, a list of hazards may be derived as soon as there is some organization of the system and definition of relationships.

When engineering a solution to meet stakeholder needs, it is imperative for the architect to ascribe what is within the system and what is outside it. Model representations will constrain the design problem space to what is under the authority of the design agent to determine the boundaries that constrain the solution. Once the system boundaries are defined, architecture provides a high-level organization and definition of



"Given a context for the system model will define boundaries, external actors will interface with the system. The system's capability to cause mishaps is generally at its interface boundary. From the viewpoint of system interfaces, the analyst must determine how unintended functional or physical effects impact elements outside the system boundary. Hazard analysis techniques should utilize such model representations to assess how system-level hazards will propagate outside the system."

constituent elements. Such elements are generally divided into two categories: physical and functional. Functional elements answer the question, "What does it do?" whereas physical elements answer the question, "How does it do it?" Functional architecture describes an action with an action verb statement (many functional descriptions are anthropomorphic). The physical architecture, on the other hand, normally emerges as a result of decisions to apply the functional architecture. For this reason, it is generally accepted systems engineering practice that system functions are *allocated* to physical elements. The process of functional allocation is highly important to implementation. If functional system hazards are understood, then the necessary controls and constraints can be translated to the physical architecture during their allocation.

System Interfaces — Given a context for the system model will define boundaries, external actors [Ref. 8] will interface with the system. The system's capability to cause mishaps is generally at its interface boundary. From the viewpoint of system interfaces, the analyst must determine how unintended functional or physical effects impact elements outside the system boundary. Hazard analysis techniques should utilize such model representations to assess how system-level hazards will propagate outside the system. Hazard tracking will need to communicate these effects to the system stakeholders, along with the limitations of the system to control them. In cases where the system is not effective in reducing the risk to an acceptable level, hazard tracking is the vehicle to communicate the amount of residual risk that will either need to be accepted or will need additional mitigations applied.

Architectural techniques — Risk-reduction techniques can be applied at the system level to begin steering design to achieve a principle-based system safety approach. High-level system architectural models provide an opportunity to apply these techniques to reduce system risks. Mitigating hazards requires isolating mishap causes from combining with initiating mechanisms that cause hazards. Understanding system hazards, along with how faults and errant behaviors can propagate through the system, will allow the architect to apply architectural safety techniques. Some of these architectural safety techniques include:

- **Partitioning and Isolation.** Isolation and partitioning techniques cause the hazard to be contained within a boundary, where it may be dealt with in a way that maintains acceptable levels of risk.
- **Multiple events and Independence.** For severe mishap consequences, the architecture may apply system elements that apply a multiplicity of events to reduce the probability of causing the associated mishap.
- **Incompatibility and Design Diversity.** Incompatibility simply refers to how incapable one event is to cause another. Examples of incompatibility include the use of different types of technology that have different failure modes and fault effects, or the use of strongly encoded variables to ensure they cannot be mistakenly used by other operations.

- **Determinism and Deterministic Boundaries.** Hazard controls need to be designed with a high level of certainty that the mitigation is effective and repeatable. Boundaries should be established to prevent non-deterministic operations from affecting deterministic safety functions.

System Safety Design in MBSE — Design is the low-level representation that implements a higher-level architecture. System design is characterized by the definition of specific details sufficient for component designers to implement elements of the higher-level architecture. MBSE provides a basis for cogent and coherent methodology for decomposing high-level system objectives to lower levels of architecture and design abstractions. MBSE design should adhere to the same safety themes set forth in the high-level architecture. Safety themes effectively mapped to system elements are more likely to be decomposed into safety features as the model design approaches lower levels of abstraction in defining the system. Architectural techniques of isolation/partitioning, independence/incompatibility and determinism are the same in design — except the level of detail is increased in how the model represents how such techniques will be implemented.

System Safety Requirements Analysis — As the system is decomposed from higher to lower levels of abstraction, the necessity of clear and unambiguous requirements becomes a priority. This is to ensure that the various engineering activities, specialties and suppliers are developing in accord to support the established high-level system objectives. MBSE methods are used to either represent how higher-level requirements are decomposed or as a source to derive and allocate lower-level requirements. The goal of the requirements analysis process is to define the minimum set of requirements that clearly communicates stakeholder needs and decompose how system elements align with those needs in a way that is unambiguous and verifiable. Requirements are either flowed, derived or allocated. System safety is a primary stakeholder in the requirements analysis process. Hazards and safety-significant functions should be capable of being mapped to requirements so that the design may incorporate safety features necessary for hazard mitigation. When MBSE methods are used to derive or allocate requirements, the originating model artifacts must be evaluated with respect to applicable hazard analysis techniques. From a functional standpoint, this is to define emergent behaviors that result from design decomposition. From a physical standpoint, failures and hardware interaction effects may become hazardous. As the requirements are derived from model constructs, the safety practitioner must identify gaps to be filled with safety requirements. Safety requirements hazard analysis (MIL-STD-882E) performed at each level of requirements decomposition will provide analytical coverage to determine if safety requirements are adequately associated with hazards.

Requirements may be "plain language" requirements communicated by a speaking language (e.g., English, French, etc.). Although plain language requirements may be widely interpretable, they are also characteristically prone to ambiguity. Formal MBSE languages provide highly defined lexicons that specify system performance, behaviors and physical constructs with a relatively low level of ambiguity. Therefore, some development activities elect to use MBSE views as the requirements themselves. If this is the case, then it must be understood by the system safety practitioner early in development, since it will be mandatory to map safety requirements analysis techniques directly to the model representations.

Requirements Traceability — In terms of requirements analysis, the concept of traceability maintains the context and origin of the individual requirement. Traceability provides a means of addressing the questions of where the requirement came from and why it is there. MBSE methods and tools provide the means of establishing traceability, if they are invoked by the development activity. Object-oriented modeling languages such as SysML® and UML® provide model element types that connect requirements to other model elements such as blocks, classes, activities, etc. Sophisticated modeling tools such as Simlulink® and Rhapsody® provide means of dynamically linking requirements to the requirements database or specific documents so that as the model changes, the traceability links are retained. Traceability of safety-significant requirements is highly important to the system safety practitioner. If hazards are traceable to requirements, then the traceability of model design to the requirement points the hazard tracking to the specific safety feature called out in the model. This allows the follow-on safety assessment to be able to identify the specific design elements that mitigate hazards so those mitigations can be properly evaluated in terms of system risk.

Verification and Validation — As mentioned earlier, model-based approaches provide a means of test-

ing assumptions of the model level of abstraction to the implementation or solution. Since the model is a representation, it is also an abstraction on how the system can be verified. A test architecture can be implemented as an overlying representation of how the solution will be verified. Test architectures use the model representations for the development of test vectors to be applied to the implementation unit under test. Test architectures are used to target key attributes, operations and performance parameters that are translated into specific test procedures and test cases. Many MBSE tools provide automated verification traceability so that test vectors may be associated with their target model elements and requirements. Sophisticated models are executable and allow the analysts to utilize the model itself as a representation to test assumptions, requirements and even environments. Executable models may simulate various domains of the system and thereby be used as a means of verification. External environments, sources and interfaces may be emulated to determine how the system will perform under various external conditions. Safety features represented in the model can be verified in an executable model.

Where verification is the assurance that the requirements are correct, validation is the determination that the solution meets stakeholder needs. Models are tools for validating operational and design concepts. Model representations allow the solution to be communicated early with stakeholders to determine if the solution meets their needs. Dynamic executable models can be assessed in various situations and conditions to test original statements of stakeholder needs in simulated environments. Validity includes the safety as a system attribute, as most stakeholders would not accept an unsafe system as valid.

Conclusions

Model-based safety engineering is growing as a systems engineering discipline to reduce development risks and costs. The practice of system safety engineering can capitalize on MBSE methods to enhance architecture and design to better achieve system safety objectives. System safety can integrate with MBSE in many different ways, but the end goals and desired results must be determined ahead of time. Coupling system safety methods with MBSE processes provides substantial advantages in reducing cost, influencing architecture and affecting design to minimize safety risk associated with the developed system. For such benefits to be realized, system safety must be integrated with the development MBSE framework as early as possible in the system's lifecycle.

About the Author

Patrick R. Oliver is a system safety and human factors engineering lead on various military weapons projects supporting all services of the Department of Defense. He is a U.S. Navy veteran and holds a MS in aeronautical science from Embry-Riddle Aeronautical University, specializing in both aerospace safety systems and human factors. He holds a BS in aeronautics, specializing in safety systems, also from Embry-Riddle Aeronautical University.

References

1. Friedenthal, S., A. Moore and R. Steiner. *A Practical Guide to SysML: The Systems Modeling Language*, Elsevier, New York, 2014.
2. *International Council on Systems Engineering. Systems Engineering Handbook*. Wiley, San Diego, California, 2015.
3. "International Council on Systems Engineering Technical Operations," *Systems Engineering Vision 2020*, International Counsel on Systems Engineering, http://www.icose.org/media/upload/SEVision2020_20071003_v2_03.pdf, September 2007.
4. U.S. Department of Defense. "Department of Defense Standard Practice: System Safety," Air Force Materiel Command/SES Headquarters, Wright-Patterson Air Force Base, Ohio, 2012.
5. Ericson, C. A. *Hazard Analysis Techniques for System Safety*, Wiley, Hoboken, New Jersey, 2015.
6. Leveson, N. G. *Engineering a Safer World: Systems Thinking Applied to Safety*. MIT Press, Cambridge, Massachusetts, 2016.
7. Larman, C. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, Prentice Hall PTR, Upper Saddle River, New Jersey, 2004.
8. Arlow, J., and I. Neustadt. *UML and the Unified Process: Practical Object Oriented Analysis and Design*, Pearson Education, Boston, Massachusetts, 2005.