LEARNING MADE EASY

2nd Siemens Special Edition

MBSE

for dummies®

A Wiley Brand

Elevate system modeling

Transform the design process

Reduce the impact of late changes

Compliments of

SIEMENS

Steve Kaelble

About Siemens

Siemens Digital Industries Software is driving transformation to enable a digital enterprise where engineering, manufacturing and electronics design meet tomorrow. The Xcelerator portfolio helps companies of all sizes create and leverage digital twins that provide organizations with new insights, opportunities and levels of automation to drive innovation. For more information on Siemens Digital Industries Software products and services, visit www.sw.siemens.com or follow us on LinkedIn, Twitter, Facebook and Instagram. Siemens Digital Industries Software – Where today meets tomorrow.

Note: A list of relevant Siemens trademarks can be found here. Other trademarks belong to their respective owners.

MBSE

2nd Siemens Special Edition

by Steve Kaelble

Technical Contributor: Hans-Juergen Mantsch,
Distinguished MBSE & SDV Engineer,
Siemens Digital Industries Software

for dummies®
A Wiley Brand

MBSE For Dummies®, 2nd Siemens Special Edition

Published by
John Wiley & Sons, Inc.
111 River St.
Hoboken, NJ 07030-5774
www.wiley.com

ISBN 978-1-394-28266-1 (pbk); ISBN 978-1-394-28267-8 (ebk); ISBN 978-1-394-28268-5 (ebk)

Publisher's Acknowledgments

Development Editor: Jen Bingham
Acquisitions Editor: Traci Martin
Senior Managing Editor: Rev Mengle
Client Account Manager: Matt Cox
Production Editor: SaiKarthick Kumarasamy

Foreword

Since the first edition of this book was published, the adoption of model based systems engineering (MBSE) has expanded and MBSE is an accepted reality in many engineering processes across diverse industries. However, demands for efficiency and safety in the aerospace and defense (A&D) industry drive increased complexity and increased numbers of systems, and have exponentially increased the number of intersections among systems. The risk of errors multiplies at these intersections. Traditional MBSE approaches exacerbate the issues because they can involve siloed and disconnected teams and solutions.

Moreover, in the automotive & transportation (A&T) industry, software defined vehicles (SDV) are transforming the ecosystem of carmakers and their suppliers, making software a vital focus for ongoing and future vehicle designs that benefit from adopting structured MBSE approaches, enabling better, and swifter adoption of the challenging automotive transformation driven from SDV.

Specifically, in the A&D industry, Siemens is focusing on a shift to a mission-driven approach to MBSE. A mission-driven systems engineering approach ensures end-user mission success by managing the technical baseline, beginning with the business understanding of the mission (the product) and extending through the end-user execution of the mission (the product).

This new version delivers more opportunities, processes, and methods that highlight changes, expansions, and migration guidelines to help engineers achieve a mission-driven systems engineering approach.

Introduction

It is an understatement to say that aircraft and aerospace and defense products are complicated. So, too, are the systems engineering development processes bringing them to life. But for too long, systems engineering has been its own niche function — one not really plugged into the rest of the development process. As products evolve, so must the processes for creating them. The build-and-then integrate approach for complex systems engineering design no longer works because optimizing a system in isolation brings the risk that it won't mesh well with other aspects of the end product. Despite this, for many organizations, changing the approach to systems engineering is hard. Customers often don't know where to start, or they haven't fully implemented new solutions because they require investment.

The truth is that a model-based approach to systems engineering can be the vital driver of any program. Because, despite the complexity of the products they create, engineering organizations face growing demands to innovate more quickly while reducing costs. That would seem to present competing priorities. After all, the more complex the product, the more likely you are to encounter integration issues, which can cause you to slam the brakes on a project. But new digital transformation processes are driving revolutionary ways to design components and entire systems. This transformation connects disciplines and allows teams early in the process to see how systems interact and evaluate the impact of decisions and revisions across the entire design.

Today's model-based systems engineering (MBSE) provides insights during conceptual brainstorming. It can drive requirements down into the detail level of the various domains and brings with it complete validation and traceability. Incorporating the mission-driven approach to systems engineering will help you achieve a digital transformation of systems engineering. A new approach demands that you start with a view of the product's end-use mission as well as consideration of intended variants. It also requires an integrate-and-*then*-build approach.

Icons Used in This Book

Throughout this book you'll spot several icons. Think of them as road signs, letting you know of something significant nearby.

REMEMBER

This book is short enough to zip through. Just don't miss the paragraph next to this icon.

TIP

This icon draws your attention to a helpful hint.

WARNING

No one needs to tell you things can go very wrong. Here's a pointer to avoid a bad situation.

Beyond the Book

This book, which is a few dozen pages long, only scratches the surface of how to expand your knowledge and use of MBSE. If you come away with an appetite for more, here are some resources:

- **Talking Aerospace Today:** An informative podcast from the Siemens Aerospace and Defense Industry team.
- **Siemens/IBM Partnership:** Siemens and IBM have partnered to enhance MBSE by integrating SysML v2, an open standard, into Siemens' digital ecosystem. This collaboration focuses on improving early product development, encouraging model reuse, and creating cross-domain integration.
- **The Aerospace Systems Engineering web page:** Siemens solutions help aerospace engineers adopt a mission-driven approach to aerospace systems engineering to connect systems, domains, and stakeholders.
- **The Aerospace Design web page:** Discover how Siemens brings an aircraft or rocket design to life faster with a multidisciplinary design and optimization approach.
- **The Aerospace Program Management web page:** Discover the power of cloud-based PLM with TeamcenterX.

IN THIS CHAPTER

- Accelerating product innovations
- Embracing the challenges inherent in change

Chapter 1
Transforming Aerospace & Defense

Few industries fuel the human imagination as intensely as the aerospace and defense (A&D) sector. Think about all the books, movies, toys, and video games that revolve around flying somewhere exciting, winning a battle, or soaring into space. Consider how people around the world have been captivated by everyone from Otto Lilienthal to the Wright Brothers to Neil Armstrong.

This chapter explores the fast-changing A&D sector and its embrace of innovations that will become tomorrow's books, movies, toys, and games. It outlines visions for the future, factors that drive change, and the difficulty of keeping up with the forces of the evolving digital landscape. In A&D, products must now rely heavily on software- and electronics-driven systems.

The chapter also introduces you to the groundbreaking changes that are happening in the automotive and transportation (A&T) industry. One of the key enablers of these changes is often referred to as the software-defined vehicle (SDV) paradigm.

Continuous Innovation

Innovation is the name of the game in A&D. In fact, innovation is creating new opportunities in propulsion, supersonic and hypersonic flight, and urban air mobility (UAM).

Consider, for example, the concept of advanced air mobility (AAM). An electric vertical takeoff and landing (eVTOL) aircraft is an alternative for transporting people and goods in a faster, more environmentally friendly way. Although designed for urban environments, eVTOL also helps to serve less-populated areas, delivering cargo or medical supplies with greater ease.

An all-electric taxi is an interesting example because it could involve technological innovation and whole new business models, such as Uber-like services for air mobility. All of this requires spectacular and disruptive innovation.

REMEMBER

And then, there are ongoing innovations in existing, more traditional A&D sectors. There's always a desire to get somewhere or deliver something faster, which means an ongoing push for such things as supersonic business jets or even hypersonic transport. Higher travel speed isn't the only consideration. There's more concern than ever about environmental impact, which causes A&D product developers to keep sustainability top of mind.

Expand your gaze further into space, and the innovation continues. For example, satellites are getting smaller and smaller, yet all the more powerful. Consider the Starlink constellation of small, low-orbit satellites developed by SpaceX to provide satellite-based internet access.

Many of these exciting ideas are happening in the private sector, driven by commercial interests. Think of the current "space race" between Virgin Galactic, SpaceX, and Blue Origin. Of course, for every innovator on the commercial side, you've got many who are forward-thinking in the defense-focused A&D sectors.

No matter where you look, great ideas seem to emerge more quickly than ever. Everyone is trying to get to market faster, and at less cost. For example, digitalization enables a mechanical system to move to a software-based capability that's both efficient and cost-effective to adopt.

Meanwhile, innovation on the military side of A&D has traditionally come with a skyrocketing price tag, as famously suggested by former Lockheed Martin CEO Norman Augustine. Augustine's "Law Number 16" states that while defense budgets may grow at a linear pace, the cost of new aircraft grows exponentially.

Rising costs are simply no longer sustainable. The emphasis now is on innovating, yes, but also on controlling costs and cutting the development schedule significantly.

Despite exciting signs of progress, the truth is that A&D is an industry that's not always accustomed to moving quickly. It's not a matter of wanting to move slowly in product development — it's just that products that fly through the air or shoot into space are incredibly complex.

REMEMBER

Many years can pass between that light-bulb moment and a product hitting the market. Yet, as the pace of progress accelerates, long cycle times become a problem. What if expectations change while your product is still on the way to market? How can you remain flexible enough to evolve the product while it's still in development?

The complexities that slow the wheels of progress aren't going away. There is more electrification, for example, and more efficient and reliable electromechanical options are supplanting hydraulic technologies.

Everything these days is software- and electronics-driven, not just in A&D but virtually everywhere. Again, this enables greater control and flexibility, improved efficiencies and futuristic capabilities. But it can make engineers' lives even more demanding.

Embracing Change

The thirst for innovation is far from the only factor driving change in the way companies operate. There are also ongoing pressures from governmental agencies and regulators, everything from the U.S. Department of Defense (DOD) to the Federal Aviation Administration (FAA) to the European Union Aviation Safety Agency (EUASA).

REMEMBER

Companies have urgent priorities, some not always in alignment with one another. The industry must adapt to serve its many masters — and that must include the way its products are designed, engineered, and manufactured.

Change is also being propelled by newcomers' efforts, including startups in space, air mobility, and defense. Disruptors such as drones are making noise across the industry, as well.

Ecological considerations are driving change, too, and in ways that might differ from one part of the world to another. A push toward alternative fuels in one society may be matched by increased interest in electric or hybrid solutions somewhere else.

Safety and meeting stringent regulatory requirements, of course, are ever-present issues that get attention from the industry, the public, and regulatory agencies. Safety is more important than everything else combined. To succeed, aircraft manufacturers must consider every conceivable hazard from turbulence and thunderstorms to bird strikes to human error across the design and manufacturing process.

Meanwhile, companies are subject to workforce issues that impact many industries. Baby boomers are moving toward or into retirement. As A&D employees, they've developed valuable institutional knowledge. Their employers must quickly capture that knowledge and ensure that it gets passed down to new workers. The need for greater collaboration is one more reason to update the way organizations manage engineering-related data and knowledge.

Longtime institutional knowledge and established processes can also work against the forces of change. That's an issue with organizational cultures that do things because, "We've always done it this way."

Perhaps the biggest challenge of embracing these multifaceted and sometimes competing forces of change is the fact that everything in this complex A&D world is so incredibly interrelated. Every little change can have a significant and sometimes hard-to-predict impact somewhere else in the design, engineering, and production processes.

The bottom line is, you're not just designing an airplane, helicopter, rocket, drone, or whatever the final product's end use may be.

REMEMBER

You're developing a system of interconnecting systems on the journey to digitally transform your products and processes. This system of systems isn't just limited to the aircraft. The system that is an airplane must perform its primary mission of getting safely off the ground, of course, but it also must work within the air-traffic control system and airport infrastructure.

Each system has teams of engineers bringing it all to life. Helping all of these equally important players collaborate effectively is the goal of MBSE. What is needed is digital transformation of systems engineering across the entire product and product lifecycle. A new approach demands that you start with a view of the product's end-use mission as well as consideration of intended variants. It leverages integrated, holistic processes to connect systems, domains, and stakeholders across the complete product lifecycle to achieve early optimization and to mitigate risk, and that's what this book is all about.

Discussing Software-Defined Vehicles

The A&T industry is on the cutting edge of technological development today. Autonomous driving, drive train electrification, and new business models for software-enabled functions continue to revolutionize the way automobiles are made and operated.

REMEMBER

SDVs are at the heart of this development and will have a huge impact on all aspects of the vehicle and its lifecycle. SDVs are driving the development of software architecture and network communication as well as demanding integration, verification, and visualization of embedded software systems.

The SDV paradigm will affect collaboration, design, development, and manufacturing. SDVs will affect sales as well — offering a way to implement a tailored after-sales upgrade process and tailored market platforms. Most germane to this book, SDVs will also influence how the software, electric, and electronic vehicle architecture is designed and implemented, moving from a highly distributed architecture of up to 150 electronic control units (ECUs) to a zonal approach, where general-purpose, high-performance compute units and local zone controllers managing electrical and communication interconnect are combined (see Figure 1-1).

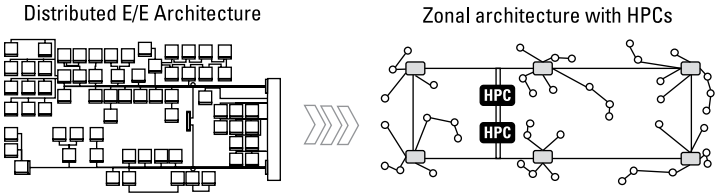

FIGURE 1-1: SDV impact on electrical/electronic architecture.

With all this complicated technology interacting, it may seem obvious that MBSE can offer significantly benefit to SDV manufacturers by allowing systems engineering best practices to expand into the software domain, driving development, testing, validation, and verification activities from a common systems engineering approach.

REMEMBER

The SDV paradigm can decouple software and hardware development into parallel instead of serial processes. This allows engineers to develop these interacting parts concurrently in order to deliver the benefits of speed and shift left.

IN THIS CHAPTER

- Understanding a mission-driven approach to systems engineering
- Taking on the complexities
- Innovating digitally
- Adopting or expanding on a model-based approach

Chapter 2
Rethinking Your Systems Engineering Approach

The aerospace and defense (A&D) and automotive and transportation (A&T) industries create all kinds of exciting things, all of them incredibly complex (for more on this, see Chapter 1). An airplane or a software-defined vehicle (SDV), for example, is a combination of countless interconnected systems that all must work in perfect harmony as they tackle hundreds of thousands of requirements. Ongoing advances in electrification mean more complex systems of wiring to link these systems. And the creation of the many systems on an aircraft or SDV must proceed as smoothly as a symphony, because any note out of place can impact the final score in unpleasant ways.

That symphony needs a new conductor these days. This chapter explains why by exploring how systems engineering has worked in the past and how ever-increasing complexities require a new, modern approach. It discusses how digital innovations drive new options and gives a basic understanding of the next-generation model-based systems engineering (MBSE) approach.

Defining Systems Engineering

Systems engineering is a multidisciplinary engineering approach that pulls together complex systems of systems across their lifecycles. It's a structured and robust approach guiding design, development, production, and ongoing maintenance of a complex product such as an airplane. Metaphorically speaking, systems engineering is kind of like the various musical scores of the symphony mentioned earlier, with a part for the first violin and a part for the tuba and a part for all of the other players who collectively make it sound amazing. Systems engineering is the composition of that beautiful music.

The International Council on Systems Engineering (INCOSE) defines MBSE as "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 lifecycle phases."

REMEMBER

Systems engineering pays close attention to requirements and how various requirements relate to one another. It gauges how adjustments to one set of requirements might impact the ability to meet others. It makes sure requirements are refined and adequately constrain the design process. And it assures that the design can be verified to ensure that requirements are fulfilled.

The scope of systems engineering continues past design modeling and simulation and optimization and verification into production, and, ultimately, into product operation, maintenance, retirement, and disposal. Systems engineering ensures that all requirements are satisfied while maintaining safety as a paramount priority throughout development.

To make a long and extremely complicated story short, systems engineering means a lot of different things to a lot of impacted people. It may be one of the most misunderstood disciplines involved in creating complex products, with common confusion between systems engineering and systems design.

Systems engineering starts with the big picture and works its way down into the details. It's very rigorous about managing requirements, and driving those requirements and functions down into systems, subsystems, and components. Through this methodical

process of driving requirements further down from one level to the next, it's possible to trace a requirement from the big-picture view of the aircraft down to an integrated circuit and back again.

Systems design, on the other hand, is an interdisciplinary engineering activity that enables the realization of a specified system. It's a type of process that defines components, interfaces, and other data to ensure that all specified requirements are met and that these systems effectively work together.


REMEMBER

It's also an important way to focus on safety. That's because the more safety-critical a particular system is, the more important it is for people to think about it earlier in the design process.

Systems engineers take into account the safety-critical nature of a particular system to ensure that the design will result in safe operation. It's on their minds as requirements are set, as the function of the design is considered, and as implementation unfolds. They write new requirements based on the safety analysis — creating a closed loop that begins with initial requirements, proceeds to safety analysis, and then initial requirements are revised to accommodate newly decomposed requirements. It's common to run multiple iterations to define product requirements.


REMEMBER

Product requirements aren't just about creating textual descriptions of functionality. They require capturing the intent, use cases, interactions, dependencies, structure, and nonfunctional demands of a product.

The V model of traditional systems engineering, shown in Figure 2-1, paints a picture of how it all works. Engineers start from the left side at the top with a concept of operations, the overall functional aim that drives downward through high-level and detailed requirements into high-level and detailed design.

That process of definition and decomposition of requirements reaches the bottom point of the V, where implementation unfolds. Then it's back up the right side where integration and verification take place. Exit the right side of the V, and you've reached operations and maintenance. See Figure 2-1.

No matter what level, systems engineering assigns a means of verification. For example, there may be checks of an electrical system box, bench testing or rig testing of a subsystem, or closing the loop by flight testing the whole product.

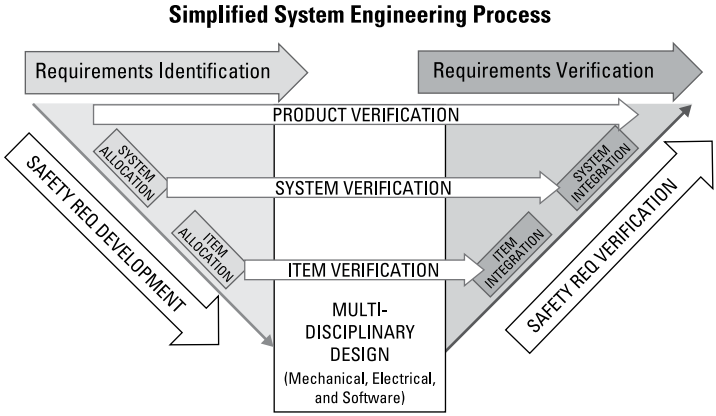

FIGURE 2-1: The systems engineering V process, simplified.

Systems engineering is very rigorous and fully interdisciplinary, even if it's sometimes not well understood. Many people talk about systems design, and when they do, they may be thinking about three-dimensional, computer-aided design (CAD), or modeling systems in CAD, or other aspects of simulation or verification. Some think of requirements, decomposition of requirements, diagrams, and the modeling of a product or the various systems within the product.

REMEMBER

But systems engineering is more than a beefed-up CAD system. Systems engineering is an interdisciplinary field that includes various domains and types of engineering, such as electrical, software, electronics, mechanical, and so on. The systems engineering process tends to orchestrate, while the rest of processes execute. A systems engineer provides the glue that holds the system development process together across the entire product lifecycle. Put another way, the systems engineer is a program manager's best friend.

The next-generation MBSE approach, the focus of this book, is far more than just the left side of the V, and the trip back up the other side of the V. It's adopting a digital transformation approach that's model driven and automated. It's leading the process through multiple conceptual iterations that define requirements and the product architecture. It's facilitating a clean hand-off from conceptual design into detail design, on through verification, and finally into manufacturing and product support. It's all about beginning with the business understanding of the mission

or product and extending through the end-user execution of the mission or product.

In any well-implemented systems engineering approach, the hand-off of the project should not just entail delivery of printouts of requirements and respective diagrams, but rather it should deliver models with their associated in-context data for direct downstream consumption. Transforming Systems Modeling Language (SysML) into Unified Modeling Language (UML), for example, can create a direct link and digital thread from abstract systems engineering into the software implementation stage. In this sense, systems engineering delivers a direct contribution into the downstream design flow by providing a model rather than text, avoiding potentially error-prone interpretations.

Additional examples of benefits from providing models for direct downstream consumption are electrical and network connectivity or parameters for validation and verification runs. All of these measures help to increase acceptance of systems engineering so that it's not seen as simply an abstract, upfront, must-do activity but rather an active contributor, increasing the efficiency of development teams.

REMEMBER

Going back to that symphony metaphor, MBSE is a bit like the conductor's score. It has one unified display of what all the various instruments are playing, giving the conductor visibility into how the various parts interact. The conductor knows that when the concertmaster plays this particular thing on the violin, the tuba needs to be playing that other thing. Ultimately, the conductor knows what will sound best from the audience's perspective and will know what to do if part of the symphony doesn't go over as well as hoped with the audience.

MBSE isn't an entirely different approach from traditional systems engineering, but it does artfully remove some of the laborious intensity of traditional systems engineering processes. A CAD-like, multidisciplinary model-based methodology and digital transformation toolset make that possible.

Keeping Up with the Complexities

It's worth discussing why an updated approach to systems engineering is necessary: The practices of the past few decades have continuously evolved to meet the demands of increasingly

complex products, the need for an advanced digital approach, and industry demands to do everything in less time with lower cost and fewer resources.

You may not have even been born yet when systems engineering came into the picture, but if you were around half a century or so ago, you know that technology was a lot simpler. Back in that day, the kind of computers we use today barely even existed in science-fiction imagination. There were computers built into our space program's vehicles but not into cars or typical airplanes. And the computers that flew astronauts to the moon couldn't do most of the things your mobile phone can do today.

Even going back 20 years, a lot of important design-related work and data storage took place in spreadsheets or word-processing documents, maybe diagram software, often even on paper. Everything was document-centric and difficult to effectively manage in a big-picture kind of way.

In A&D design, more and more functions became interconnected and consolidated and controlled with software. As systems became increasingly federated, it became more complicated to monitor system interactions. With integrated modular avionics, there are far too many interactions to track on a spreadsheet.

As the complexity of an aircraft increases, so too does the complexity of its software, electrical, electronic, and mechanical systems. That puts new challenges onto software development and its time frame, especially as you combine that complexity with the challenges of certifications. You need tools that can move quickly as you integrate software with hardware and other systems, model, and test — as you slowly begin to embrace the digital realm.

Think now about the ever-smarter weapons systems that military leaders seek or autonomous vehicles in either the defense or commercial sectors. As vehicle-level electronics become more sophisticated, along with the digital infrastructure in the environments where these vehicles operate, how can you create verification plans to ensure that complex systems will behave appropriately and safely?

What's required is a single source of truth, an integrated environment to ensure that all product verification events, whether

simulation modeling or analysis, are driven by requirements, planned and executed in the correct sequence, linked to the necessary resources, and conducted with full traceability.

REMEMBER

Just as tasks and technologies are becoming more complex, so too are regulations. That's true across many sectors, particularly in such areas as A&D and automotive. Safety standards? Sustainability targets? They're growing in number and complexity.

They're established in individual nations (and sometimes regions), in the EU, and in international forums. They're coming from regulatory bodies, standards institutes, insurance groups, ratings agencies, and the like.

WARNING

The many regulations from many places bear lots of similarities but also enough differences to cause extra headaches. And as new requirements emerge, designers take note, make changes, and adjust accordingly, and that creates ripples that can cause complications, delays, and cost overruns, especially when these changes occur later in the design process.

Exploring Digital Innovations

To clarify, the discussion of increasing complexities in the previous section is a high-level summary. The full story could go on page after page after page and leave you feeling hopeless. And that would be wrong, because this complicated situation is anything but hopeless.

Indeed, there is a need for a more disciplined, mission-driven digital approach to systems engineering. You really can't rely on documents to get a handle on the complexities. You need to understand the product's end-use mission and related requirements to effectively design its systems. There's just too much to maintain, with thousands of parts, hundreds of thousands of requirements, and millions of interactions. And any change you make in one place must be updated elsewhere.

You can't rely on humans to keep up with it all, either. We are, after all, only human. In earlier times, with a file-based system, there was a lot of manual cutting-and-pasting by people already feeling overwhelmed.

Fortunately, this is the next phase in the evolution of digital-transformation maturity, with ever-more-powerful digital capabilities. That's why this is not a hopeless situation. A good example of this is a digital thread. A *digital thread* refers to an interconnected set of technology solutions synergistically aligned to support a key area of a customer's business processes. For example, systems engineering, design, certification and verification, manufacturing, or sustainment. A digital thread brings a multiplying effect to teams using a digital twin by enabling interconnectivity of numerous digital technology solutions, domains, data, and processes across multiple systems. Each digital thread intersects the others to deliver a comprehensive, scalable, flexible approach on an open ecosystem.

To take it a step further, one can think of a digital thread as a digital fabric that aligns with the most commonly used functional programs: engineering, program management, supply chain, production, and product support. This arrangement isn't serial processes within a single function, nor are these functions intended to operate independently of one another. The exact opposite is true. With the digital thread, task automation is achieved and functions are interconnected, integrated, and linked so that connected digitalized teams can quickly access, share, and manage program details across the entire product lifecycle — at any time, from any location. A digital thread can automate formerly manual processes — not just automate them, but actually transform the entire process.

Meanwhile, digital innovations in the world of aerospace systems engineering continue to march forward. SysML is just one example. The open-source SySML grew out of UML, and it handles the business of specification, analysis, design, verification, and validation for systems. It was the first stab at creating a graphical language for systems engineering. Meanwhile, Arcadia is a systems engineering approach that taps into models in the architectural design of systems, hardware, and software.

Digital transformation has clearly been a good thing for those who live in the world of systems engineering, but in its original form, whether SysML, Arcadia, or any other modeling tool, there have been issues with collaboration and shortcomings with regard to how these Version 1 tools weave into the digital thread, which is a key component that brings MBSE to life. Figure 2-2 shows examples of the various modeling languages, methods, and tools currently available at a system engineer's disposal.

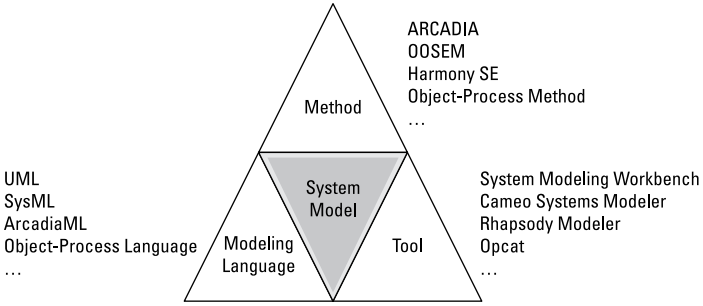

FIGURE 2-2: MBSE Essentials Pyramid.

SysML Version 2 (SysML v2) replaces that original release with an entirely new model, by focusing on data-level representation in a repository. The new version offers new syntax, new language, and a whole new way of interaction that bridges silos and takes advantage of today's digital thread.

The implementation and representation of SysML v2 presents itself in the form of almost a programming language that has major benefits for distributed collaboration and coordination across development teams. It adds a new level of abstraction into the development flow. Although managing code is a fundamentally well-established process — creating hierarchical libraries, syntax and semantic validation engines, continuous integration, and development frameworks that are state of the art in the software industry — interpretation of the abstract content is a much harder task.

SysML v2 created a graphical framework that will help visualize the model content on the fly, enabling well-defined application programming interfaces (APIs) to illustrate model intent rather than having to graphically design it. This will add speed, reusability, and interoperability to the systems engineering environments of the near future. A similar standards-based approach has been very successful in the 3D modeling domain. Jupiter Tessellation (JT), which is a Siemens-developed, openly-published ISO-standardized 3D CAD data exchange format used for product visualization, collaboration, digital mockups, and other purposes, has been established as the standard exchange format among the board of diverse solutions.

SysML v2 provides a more robust foundation for developing, designing, and managing complex systems. The partnership

between IBM and Siemens, through the integration of SysML v2 with IBM Rhapsody, equips engineers and project teams with advanced tools and frameworks to enhance efficiency and effectiveness in modern systems engineering.

Turning to MBSE

More and more companies have already adopted MBSE processes to manage product development, requirements flow down, and overall integration. These are all things that traditional systems engineering has attempted to tackle — now the question is, how to do it much more efficiently in a way that is not overwhelmed by the complexities of such things as A&D programs or SDV-enabled A&T platforms that are currently under development.

WARNING

A siloed approach doesn't cut it in today's A&D world. You can exchange all the PDFs you want, but you still end up with blind spots in the middle of the program. Various departments and domains all use their own tools, which work fine for looking at isolated performance in the silo but don't allow an integrated view.

You can have all players doing a fantastic job in their own little worlds, but that doesn't give insights into the integrated performance of a complete aircraft. Inside each silo, you may have the best expertise and the latest digital tools but you don't have a dynamic view of the program as a whole. This lack of interconnectivity also creates risk of late changes downstream, which drives significant cost increases and missed delivery targets.

The MBSE approach involves taking all elements that were file-based and making them model-based, connected by a digital thread, allowing for greater collaboration and reducing blind spots. Robust interface integration and collaboration capabilities, coupled with continuous verification, help ensure cost-saving risk mitigation early and upfront. When you manage the design at the interface boundaries, you get a robust exchange of detailed architectural information with internal teams and external suppliers. Customers can verify and optimize systems in the context of the product and mission, across the complete product rather than examining isolated systems of individual parts. The information moves back and forth smoothly and efficiently — and when compared with navigating countless separate documents, it's far easier to manage.

The mission-driven MBSE approach also applies to workflows, allowing systems engineers to fully manage all the technical data, rather than simply being caught up in the administration parts of the job. Most importantly, the next generation of MBSE is a recipe for far better control of and visibility into the entire technical program and product lifecycle, with downstream linkages and the impact of changes much clearer.

Don't forget that the systems engineering V really can't be simply a sequential approach anymore, especially when it comes to A&D programs. It is time to do more than shift left. The A&D industry needs to *leap* left. Today's MBSE supports a mission-driven process where different domains are working concurrently, collapsing that V.


TIP

For example, consider the notion of spelling out a requirement, then sending it on down the line to be dealt with. With the latest MBSE approach, simulation becomes more than just simulation — it can also help to repeatably make informed, fact- and number-based decisions about what architecture is the most appropriate for every product aspect of the software, electrical, electronical, and mechatronics subsystems, from cost, weight, performance, sustainability, and upgradeability. It can create alternatives and potential solutions for all to see.

Should your aircraft rely on hydraulic brakes or an electromechanical braking system? Writing down the requirement doesn't make that decision, but a simulation can. Now imagine making all kinds of other decisions through multiple simulations that also provide information about costs, maintenance, and reliability. Because you can quickly roll out new simulation models, you can get insights into high-level requirements early in the process and make better and more balanced decisions about technologies and architectures. You just can't do that with a digitized paper-based process, paper, or presentation software.

Analogies, though often imperfect, can also be helpful at times. Think in terms of the CAD space. You can draft a component and create 3D digital models. And you can then combine components until you've built a digital representation of an entire plane. Not a bad way to think about it, though your CAD model may not really take into account various functional aspects of what the software is up to, the electrical system, and so forth. And that's where MBSE comes in. It defines the brains of the aircraft and, analogously, the A&T vehicle platform.

REMEMBER

In short, by adopting MBSE, you're moving away from a niche modeling approach, with several very specialized individuals producing architectures and analyses. A new approach to systems engineering demands that you start with a view of the product's end-use mission as well as consideration of intended variants. It leverages an integrated, holistic processes for continuous integration, verification, and optimization of systems design across the complete product. It requires seamless collaboration across all domains across the complete lifecycle. Powerful, holistic multidomain simulation is needed to achieve robust interface management.

IN THIS CHAPTER

- Connecting with the digital thread approach
- Taking control of the technical program
- Deriving data from disparate tools
- Optimizing your products and processes

Chapter 3
Using MBSE to Orchestrate Your Technical Program

Model-based systems engineering (MBSE) unlocks the most effective and efficient way of orchestrating engineering operations for both the aerospace and defense (A&D) and automotive and transportation (A&T) industries. Document-centric methods simply require a whole lot more work. Using a model-based, mission-driven approach drives greater insights throughout the program, builds more effective connections and more scalable processes, ensures optimal interoperability between systems, and gets all players working together in new ways.

This chapter explores how a digital thread approach links domains and changes work routines. It outlines how MBSE orchestrates your technical program while allowing experts within their respective domains to keep using the tools they're already familiar with. And it describes ways your products, processes, and results can improve.

Leveraging the Digital Thread Approach

The basic problem with documents is that you have to put them somewhere. Paper documents are stored in file cabinets or stacks of folders on your desk. Electronic documents are on a hard drive, in a folder, perhaps in the cloud. They may be on a shared drive or on Bob's laptop that he left at home while on vacation.

Clearly, with a complex project, this unsophisticated, tedious approach doesn't cut it. Each document contains data that's disconnected from the rest of the development process, which will need to be extracted in one way or another. That's a major cause of late-in-the-game integration problems that end up eating half of the schedule and resources. A digital thread approach to MBSE solves this problem. It centralizes information, making it visible and accessible to those who need it.

REMEMBER

The idea of a digital thread starts at the front end of the program lifecycle. It's established by decisions made by engineers at that early stage of the process. It begins with a problem to solve and looks at alternative solutions. It connects downstream requirements to functions that are tested and simulated. And then tested and simulated over and over again.

A digital thread is, in short, a connection of all aspects of a specific discipline within the product development process. It allows you to drive your desired functionality to the system level, connecting with the logical architecture while ensuring interoperability and connectivity across the entire product lifecycle.

REMEMBER

MBSE maintains the continuity of behavioral, structural, and interface requirements throughout the program. It is this set of requirements that influences how the system is designed, tested, and delivered.

Physical testing alone is no longer sufficient. The digital thread approach creates an effective means to comprehend and prioritize verification. Virtual modeling and simulation have become a critical dependency of the total system verification. That said, physical testing is always part of the picture, so the digital thread must link virtual with physical. Linking the virtual with the physical helps create continuous connectivity that leverages the comprehensive digital twin to enable seamless collaboration across all disciplines and domains.

Ultimately, mission-driven systems engineering improves program execution by allowing you to:

- Define your system of systems using modeling to continuously refine product requirements and architecture.
- Optimize your design space by leveraging multidisciplinary design, ultimately leading you to a better design faster than ever.
- Use the comprehensive digital twin to test virtually before you build, reducing the amount of testing, risk, and cost.
- Promote reusability and/or modularity of system engineering artifacts across products, projects, and programs, improving program-specific execution.
- Manage integration across the value chain, with technical oversight of the design process and clearly defined interfaces.
- Trace all requirements and verification plans to ensure that you achieve compliance.
- Keep tabs on key performance indicators throughout the whole process.
- Access nearly any data source through the cloud, ensuring that authorized team members can securely view files and author changes.

Orchestrating Your Technical Program

How can you orchestrate the technical scope of a program more effectively throughout the entire product lifecycle?

REMEMBER

MBSE improves how everyone interacts with each other and with systems across the entire process. It means implementing functionality down through the various parts of the product development program and introducing the final product more easily.

It's about keeping things in sync. A major problem today is the unpredictable nature of product development in A&D and A&T. Changes caused by late-discovered issues and disconnected systems can create havoc during development. The key is to integrate early, which can mitigate risk by bringing issues to light

earlier in the process when they can be fixed with fewer impacts on schedules, program cost, and designs.

REMEMBER

In A&T development, and to an increasing degree in A&D development, the newly dominant role of software in the delivery and fulfilment of vehicle functions raises the necessity of decoupling software and hardware. Vehicle software now needs to be updateable after the vehicle is sold, when the owner is in possession and the vehicle is in-use.

New business value chains driven by enhanced functionality require parallel development of software and hardware to support demand growth. This allows software functions to be developed before physical hardware is available and then distributed through over-the-air (OTA) updates. There's more software in products today than ever before, which adds a new level of complexity to product development — and to systems engineering in particular.

Although classical V-cycle or waterfall-driven processes with dedicated handover points and quality- and release-gates are state of the art in mechatronic design, software development is more agile. Its development can be continuous in nature, allowing a quicker and more efficient turnaround time, down to multiple software builds a day. Integrating the two requires a new level of engineering design.

Synchronizing and reconciling hardware and software development should allow for function development before hardware platform decisions have been made or before preproduction hardware is available. This kind of concurrent planning demands new development, testing, and validation methods. This is where MBSE excels. See Figure 3-1 for a visual.

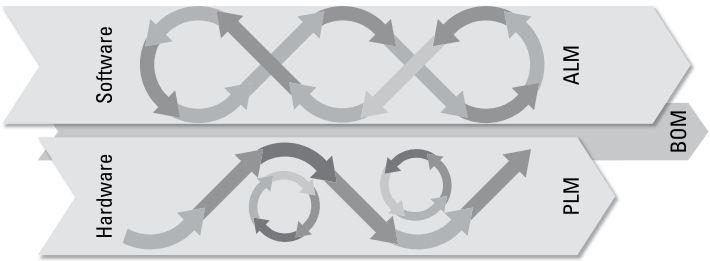

FIGURE 3-1: Concurrent software and hardware development flow, synchronizing ALM, PLM, and bill-of-materials.

With a digital thread approach in place and the resulting robust interface management, you're able to work more collaboratively with suppliers and partners, as well as with fellow engineers on your team. Because everyone is connected, it doesn't matter which specific toolset anyone is using — everything is shared across all the different domains and functional areas. Figure 3-2 shows how a systems engineering driven flow can help teams work together.

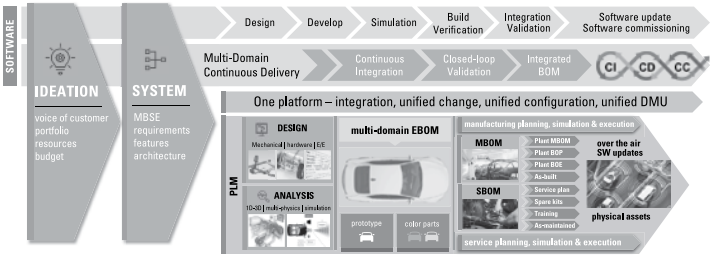

FIGURE 3-2: Systems engineering driven development flow across software and mechatronic development.

Establishing this connection requires a systems engineering-enabled digital thread that spans from early product definition and E/E architecture design through to domain-specific implementation. Achieving this also demands a sophisticated development flow for effective validation and verification. MBSE can help engage diverse development engineering teams through direct benefits in day-to-day work by boosting their efficiency. MBSE allows the team to automate tedious tasks and focus on creating the best possible product, which helps drive acceptance.

TIP

MBSE and systems engineering help to establish a link between the product lifecycle management (PLM)-driven mechatronic design and the application lifecycle management (ALM)-driven software design. Especially in the new software defined vehicles (SDV) paradigm, it's essential to connect the two design types into a coherent development flow. By doing so, you create a link between the work products and the teams responsible, enabling them to understand the common goals and requirements, including configurations, intended use cases, scenarios, test, and verification and validation (V&V) cases.

A digital thread approach drives the entire program. The digital thread builds tremendous synergy among players across various areas. Starting with product and system configurations and

requirements, once you begin to identify the solution, different engineers tackle the varying details. All have different tools, different modeling technologies, and their own simulation tools. But they're all linked together within a product lifecycle management system, such as Teamcenter, and an application lifecycle management system, such as Polarion, both part of Siemens Xcelerator platform.

As shown in Figure 3-3, parameters are objects that are used to manipulate the desired form, fit, and function to meet the targets and constraints for various system attributes. They are just one of the many MBSE solution artifacts that can be used when building the digital thread. Equally important are interfaces, requirements, and safety analyses — all are critically important components to an overall MBSE strategy.

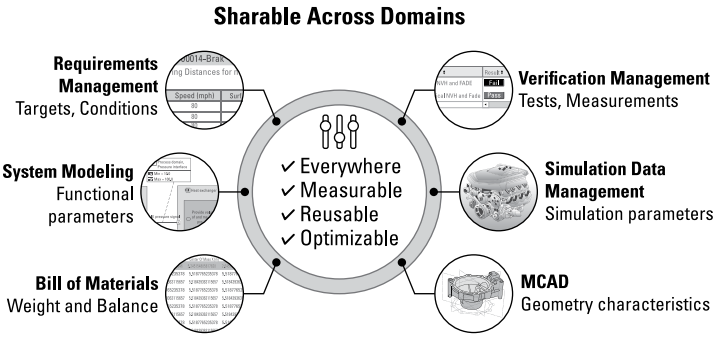

FIGURE 3-3: Managing parameters across multiple domains.

Here are more ways the MBSE approach helps when orchestrating the technical program:

- Eliminate the integration issues between complex systems and suppliers before moving into detail design.
- Easily trace all design decisions within the product from the concept phase to the final certified product.
- Establish continuous connectivity to achieve seamless collaboration across all domains across the complete product and lifecycle.
- Continuously monitor, with real-time data and parameters, key technical performance indicators, rather than taking snapshots at points in time.

REMEMBER

How does this concept work in practice? Take as an example an MBSE digital thread solution built on a PLM system, such as Teamcenter software from Siemens. With this particular MBSE digital thread, the focus is on orchestrating the technical program and driving the technical scope by moving from system modeling in isolation to a fully connected and interoperable approach realized through accelerating digital transformation of systems engineering

This comprehensive digital approach tracks all technical requirements and architecture from conceptual design to verification. It captures design decisions as the product matures, ensuring fewer late-discovered changes and mitigating risk, while creating efficiencies in cost and time. And all of this can be accomplished securely in the cloud.

WARNING

That last point is vital considering that development cycles can run a decade or longer, and you'll inevitably have product updates or changes along the way. Certification and compliance guidelines will also change, so relying on a few knowledgeable people (who may leave the company) isn't the best course of action.

The approach that leverages the most powerful digital transformation technology on the industry-leading digital backbone lets users integrate all the product design and supplier interfaces within a very flexible and open multitool solution.

Making It Comprehensive

MBSE doesn't work unless it works for everybody in the program. A digital thread approach must be feasible regardless of the stakeholders' preferred tools.

The digital thread approach is created with a database structure to organize the data and provide an authoritative source of truth. But in the interest of allowing comprehensive information and full compatibility, that data can come from multiple different tools.

The caveat, however, is that something that's truly comprehensive can quickly become overwhelming and complex. As you

connect across tools, stakeholders, and domains, you want only the right details to be shared and you need to ensure that the environment is secure.


TIP

On the other hand, there are cases when a greater level of detail is essential. If there's a new supplier in the loop, you may need more details available to closely manage the new relationship. Whatever is needed for managing suppliers and identifying integration challenges is the right level of detail.

Streamlining Design in Complex Systems

In the realm of A&D and A&T engineering, your team knows how to get the job done. The question is, can you find a way to get the job done better, particularly in terms of efficiency and cost?


REMEMBER

MBSE holds significant promise in that regard. Most companies going down this path find their development cycle greatly reduced. They're getting through programs with reduced risk, too.


TIP

MBSE continues to bring new insights into the early development stages. Multidisciplinary analyses can shed light on performance early on. There are timesavings further down the line, too, as hybrid testing modes introduce simulation with actual physical testing. That can reduce testing time in some cases.

Achieving certification and compliance depends on meeting the engineering challenges. Through MBSE, you can model the form and function of a particular item and its fluid, thermal, vibration, noise in the cabin, noise outside, and other dynamic characteristics. All of that together is what helps you understand behavior, from the component through the integrated system and product levels.

Integration issues are also a major headache for big original equipment manufacturers, often threatening to derail programs. MBSE and the digital thread have proven their ability to eliminate many major integration issues.

IN THIS CHAPTER

- Defining and understanding requirements
- Creating the models
- Putting safety at the forefront
- Writing software that makes it all work
- Keeping tabs on performance

Chapter 4
Driving Seamless Collaboration across All Domains

This chapter dives into the specifics of how leveraging model-based systems engineering (MBSE) helps you achieve seamless collaboration across the various domains that collectively comprise new aerospace and defense (A&D) or automotive and transportation (A&T) products. It offers more details on how the digital thread weaves the whole program together and how multidisciplinary teams can leverage integrated, holistic processes to connect systems, domains, and stakeholders across the complete product lifecycle to achieve early optimization and to mitigate risk. It details requirements management, system modeling, the key role of software, the importance of safety, the need to verify every requirement, the challenge of integrating many operations, and the need for up-to-date data.

Modernizing the Systems Engineering Process

Why does system engineering today seek to "leap left" with a mission-driven approach? To see the visual behind this question, refer to the figure in Chapter 2, which introduces the traditional V diagram that outlines the general flow of the systems engineering process, from the light-bulb moment at the top left through the complex series of activities that end up at the top right with your airplane safely in the air or your vehicle cruising the streets. Simply put, definition and distillation of requirements occur on the left, and the detailed work of integrating the many interconnected systems takes place on the right. As you climb the right side, you're verifying that those requirements on the left actually happened and ensuring that there are no problems.

That used to be the accepted approach, but it can take forever to get from one side of the V to the other. When forced to make decisions in sequence in this traditional, linear process, teams find increased rework due to late-discovered changes. This type of unpleasant surprise contributes to the increased cost of developing complex products. And these days, engineers don't have time to wait forever. Indeed, if work in one domain doesn't happen fast enough, downstream domains may get to work anyway, and you end up with late changes, rework, and cost overruns.

The reality is that you can't view the V as simply a sequential proposition. In one part of the V, there are changes made, problems addressed, and important realizations, which impact activities elsewhere. Differences resolved in one domain may result in solutions that can be applied in other domains, too. How can you get a handle on all that activity?

REMEMBER

MBSE fulfills this need by creating one connected digital thread across the entire process — from conceptual design, through preliminary design, through detailed design, to the test and build stage, to certifications and determination of airworthiness or roadworthiness. It's the mission-driven approach that guides all other threads that are supporting various functional teams and domains. Such areas may include product design, verification, manufacturing, and program and supplier management.

The goal of digital transformation technology providers like Siemens is to compose the entire product lifecycle process along a flexible, scalable, and open digital ecosystem, from design to manufacturing to service, and at all points in between. A mission-driven, digital-thread approach to MBSE takes systems optimization beyond an individual domain or part and offers visibility of the complete system of systems in context of the whole product and across the lifecycle.

This approach also crosses between the many different tools found across the individual domains. For example, a CAD engineer works in one mode, and a simulation engineer takes CAD data to see how it performs, using a different tool in a different mode. The simulation results created by the simulation engineer may be important to the designer who wants to know how the design is performing, but they're not in the design file. And in fact, the simulation file is many gigs of data that only a few people know how to interpret. By continuously integrating data throughout the lifecycle of a project, the digital thread ensures that all relevant information — design, manufacturing, and performance data — flows seamlessly into a unified project file, making it accessible and actionable for all stakeholders Using the digital thread and adopting this comprehensive, holistic approach weaves it all together in a way that can make sense for everyone.

Systems engineering processes have been around for decades. Still, today's A&D challenges necessitate a move from linear, document-centric processes to connected, agile, model-based methods to more of a comprehensive mission-driven systems engineering approach. That's the only way to effectively manage product development, requirements, and overall integration of design, analysis, validation, and verification activities.

In addition, with the software-defined vehicle (SDV) paradigm becoming an integral part of A&T development, reconciling the classical V-cycle or waterfall process for mechatronic component development with the agile, continuous software development flow is essential. The rest of this chapter examines how an integrated development flow might look, starting with systems engineering driven product design and its adoption though the software development flow.

Managing Requirements

Your project begins with determining the features and functions. If it's an aircraft, you begin with the general vision of which functions the plane must achieve and under what circumstances. Imagine that you're having dinner with the customer CEO and jotting it all down on a cocktail napkin. This aircraft can fly 2,000 miles, with a specific number of passengers, in particular conditions, and at a particular top speed. You're getting a general idea here, but you're running out of room on the napkin.

These features and functions — also referred to as the mission — lead to requirements that inform the architecture, and the bigger the project, the more requirements you're defining. Designing a new type of aircraft equipped with software? You're talking about hundreds of thousands of conditions and requirements criteria. Definitely, more detail than your cocktail napkin can handle.

REMEMBER

Thus is the challenge of managing requirements. As you work through the requirements, from big picture to smaller and more specific systems, you encounter more variables that can impact requirements up and down the line. As concepts mature, system-level requirements can change, as does the overall understanding of what the system looks like. There are many iterations before anything is ever finalized.

Perhaps for this customer whose request started on a cocktail napkin, you have three general concepts that might fit the bill. Within these concepts are different potential engines, each of which drives its own configuration because of its specific fuel consumption, the size of the wing where it will be attached, the drag, the fuel tanks (or battery power required, if it is an electric vertical take-off and landing aircraft), and so on. The product-level requirements work their way down into system-level requirements, on and on until you get to the component level.

Coordinating and making sense of these requirements involves connecting lots of dots and consistently checking that the product meets requirements and constraints. It's also a matter of ensuring the quality of those connections, being sure there aren't any missing connections, and ensuring that the requirements are specific enough to be actionable and verifiable.

TIP

Consider the family dynamics of requirements as you flow down. All requirements need to be traceable through what is essentially a family tree. For every requirement, you should be able to identify its parent, which is the requirement directly above it in the hierarchy. And the majority of requirements in a complex project have children that have been derived below them.

You know you've got a problem when you have an orphan requirement, which is just what it sounds like: a requirement that you can't trace up to the parent. It's also a sign of a problem if you have a top-level requirement that has no children. If no requirements are flowing down, you've likely missed something or it's a bad requirement. Suddenly, you have requirements creep that threatens to change the scope of the job.

An important point in requirements management is that requirements should be as specific and individual as possible. Are you specifying a part that needs to be round and blue and made of aluminum? That's not a requirement — that's three requirements. A specific function needs to respond in a well-defined time, with a given resolution — that's two requirements. As an A&T example, timing an airbag to deploy at the right time is essential: Engaging too early, during inflation, and the airbag explodes in the passenger's face, engaging too late, during deflation, and the passenger hits the steering wheel.

TIP

And quality requirements are "shall" statements rather than "should" statements. Wishy-washy needs are not quality requirements. That's not to say requirements are inflexible — a requirement can establish an acceptable range for whatever it's specifying. But the result must be within the range to meet the requirement.

How do you communicate these requirements in a way that informs the various domains that play a role in meeting them as they model their various systems? Your approach likely taps into many different authoring and validation tools. Parameter definition and management is the central part that builds the thread across these tools spanning multiple domains.

Your parameters will often come in sets of four.

- **The goal:** Specify the desired thrust, for example. Or the goal for fuel consumption that is coming from the requirement system.

- **The minimum:** This is the bottom level of the range that's considered to be acceptable.
- **The maximum:** The upper limit for the range viewed as acceptable.
- **The result:** This is the current measured value of the particular parameter, based on the model as it stands now.

Consider any activity that involves lots of numbers like this.

WARNING

Traditionally, the systems engineering requirements have ended up in a big requirements document, often a huge book sitting on the system engineer's desk. That approach may work if you're keeping track of your flooring installation and plumbing work, but it falters if you're tracking hundreds of thousands of requirements related to a new aircraft, to be designed and built by thousands of engineers.

That's why the MBSE approach links requirements in a database, where each requirement is an individual object that's part of an overall product configuration. Now you know that this particular requirement impacts these different part numbers, and, in turn, whatever requirements assigned to them.

But that, too, doesn't cut it in a complex situation. It's better to extract values in the requirement and create measurements on the geometry of a component using a 3D model in a CAD system, such as NX software. Also, users may choose to plug in the metrics of a performance-based requirement, such as stresses, strains, temperatures, or volume flows, into computer-aided engineering (CAE) systems such as Simcenter software.

REMEMBER

The digital thread approach to MBSE is designed to ensure that there is connectivity and flow through all of this. It can tie into the relationships between requirements, logical architecture, and physical components. Understanding and accounting for these relationships is the key to overcoming the complexities of both A&D and A&T engineering — so you can trim both costs and development cycle times. And everything is verified, so you don't run into a conflict of requirements.

Modeling the System

With requirements well established, your MBSE digital thread carries you into the process of modeling the system, which includes the all-important work of testing before you build. For example, functional modeling explores the many functions that you've built into requirements.

TIP

Through opportunities like the Siemens/IBM collaboration, which is aimed at designing a solution to enable a holistic and continuous integration of systems design across the lifecycle, you can better examine how different systems are doing the work. The system architecture model must be integrated with product lifecycle management (PLM) — and in A&T cases with software related application lifecycle management (ALM) — to carry the digital threads it produces through architectural decisions made during system modeling.

For example, you're modeling the control of the aircraft on the ground, and in doing so, breaking functions into subfunctions. You're doing the same thing when modeling control of the airplane in the air, considering such subfunctions as roll, pitch, and yaw control.

Your system safety process starts with functional hazard assessments. And that begins by examining and defining hazard levels.

To continue with the example regarding aircraft control, consider roll control. Lose that control, and the hazard is catastrophic. That determination, in turn, drives redundancy and integrity considerations — and drives requirements back up and into other models.

Your MBSE approach winds its way through parameters (system design attribute, performance metric, and so on) and requirements, functional modeling, as well as through logical modeling and the details of the architecture. A physical model follows — not the physical product itself, but a manifestation of the design. Consider the example of SMW for Teamcenter illustrated in Figure 4-1. MBSE takes various forms while winding through architecture modeling, starting from high-level mission concept through identifying the physical components and their behavior. This is done through cloud collaboration with PLM tools to plan, develop, and deliver better products faster.

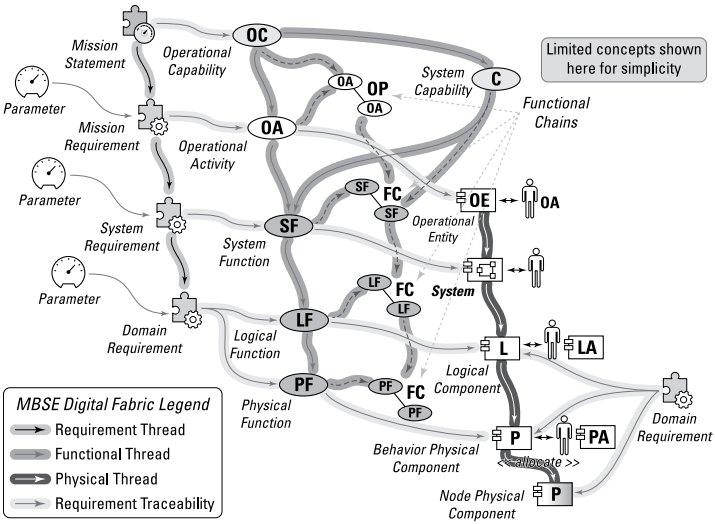

FIGURE 4-1: MBSE digital fabric within System Modeling Workbench (SMW).

Indeed, system modeling isn't the work of one person who develops the system model of the entire aircraft. A system model consists of numerous models, namely architectural, 1D/3D multiphysics, MCAD, ECAD, DFMEA, RAMS, and so on.

Ensuring Safety

In the A&D industry, safety is a critical priority. The safety process plays a key role in the overall systems engineering process.

This includes assessing functional hazards and integrating safety with the product lifecycle. As mentioned in the previous section, this includes determining and applying a level of severity to different events. For a particular situation, is safety affected? And if so, is it major or minor? These vital classifications determine integrity, reliability, and redundancy considerations.

For a system where failure would be considered catastrophic, you must have redundancy built in. The loss of electrical power on a plane is such an example. You can't just have one system, but instead you need to plan for parallel, redundant electrical systems. You must ensure that the loss of one system won't cause loss of the aircraft.

With features in the A&T industry such as autonomous driving, autonomous harvesting, and navigating in large open mines, safety has increasingly become a necessity there, too. Certification and compliance with national or international regulations are must-haves for any vehicle on the road, even in closed environments if the risk of environmental impact is present. Building a traceability framework along all artifacts constituting and describing the feature, from its ideation, the concepts, system interfaces, onto its implementation and interaction with components and other features, is a great way to create documentation for certification bodies.

The more a feature interacts with other components, the wider the search base for documentation would be. Making use of explicitly or implicitly created traceability features rather than walking through a paper trail is simply more efficient. Traceability also helps by creating what-if scenarios, failure mode assessments, and dependency trees that can be used to calculate failure probabilities or identify the perimeter of reuse.

REMEMBER

As you understand what a system looks like, safety alters the requirements picture. Checking product design requirements against safety regulations often generates an updated set of product requirements. As the product design matures, so do the requirements for software and hardware. That's why it's important to identify and analyze system performance, and safety and reliability issues upfront during the concept design phase and mitigate the technical risk in the design.

Driving Software Integration

As with so many other products today, using software in aircraft continues to ramp up dramatically. Software development starts with requirements definition and considers the level of integrity you need to maintain. Is it flight-critical software? That changes how you design the software and puts different requirements on the development process.

TIP

There must be independence between who's designing flight-critical software and who's testing it. It may be that a person in one place writes software, and a different person elsewhere writes testing requirements. Bringing disconnected users and

stakeholders together is a tremendous challenge, and if not done well, can result in massive delays or missed communications.

A solution such as Polarion has automated tools for defining, building, testing, and managing these complex systems. In the area of software, it's vital to bring it all into one place because software creation uses various open-source tools. You need to be able to talk to all of those different tools as you paint the big picture.

The role of software in the overall operation of the finished product can sometimes be deceiving. This is true in A&D and in other sectors, too. Take a drive in your new car, for example, and it is a very physical experience of getting from here to there with a certain level of comfort, efficiency, and style. It's easy to overlook how much of that experience is controlled by software.

Take that automotive example a step further. The newest model year may have some amazing new features driven entirely by a new piece of software code rather than a physical improvement.

Software in A&T has become crucial. It enables new business models, such as selling additional functionality after the vehicle has been purchased. The user can upgrade to more advanced features by enabling the use of sensors, actuators, and hardware already present in the vehicle. This can add comfort and keep the vehicle up to date, which can extend the vehicle's lifetime.

This underscores the importance of software within your modeling environment and as part of the MBSE digital thread approach. If you're going to model something, you must be able to model the software in parallel to look at the product from the perspective of a true digital thread. An integrated solution like Siemens integration with IBM Rhapsody can effectively link model-based systems engineering in Systems Modeling Language (SysML) and software architecture modeling in Unified Modeling Language (UML), supporting numerous process improvements. These improvements range from concept space exploration, requirements elicitation, specification, predicting early behaviors, and even automating test-case generation.


TIP

Integrating software with PLM and plugging it into a solution such as Polarion ALM helps ensure that you're fully addressing the impact that software has on the overall product and its capabilities, allowing the hardware and software design to closely

integrate and align without one blocking the progress and agility of the other. As you look at the product on a larger scale, as a finished aircraft, you don't see the software. It's not visible to the naked eye. It's not obvious. Still, its presence has every bit as much an impact on operations and capabilities as all the aluminum, composites, and every other physical element. This goes back to the idea of starting a project with the end in mind by integrating the mission with the product lifecycle. That's why it's important to drive software in parallel with systems by using product architecture that carries into software development, thus enabling continuous integration (versus expensive late-cycle integration).

In SDV, software development is happening in parallel with every other activity needed to produce the product, including creating hardware and integrating sensor and actuator components. As the software teams create their parts of the ecosystem, they must move at the right speed to integrate and reconcile with maturing hardware and design by managing integration releases at the right times. One method that helps in delivering the flexibility needed is to enable different levels of hardware virtualization for test and validation of software and use cases.

Figure 4-2 shows such a hardware and software separation in an integrated design flow enabled by cloud-based collaboration and hardware virtualization technology. This creates an end-to-end flow that is traceable and aligns with both the V-cycle-driven hardware flow and the agile software development flow. Doing things this way provides a digital twin to the development teams to design more quickly and efficiently, enhancing manufacturing capabilities. All of this together enables a quicker time to market and reduces overall spend.

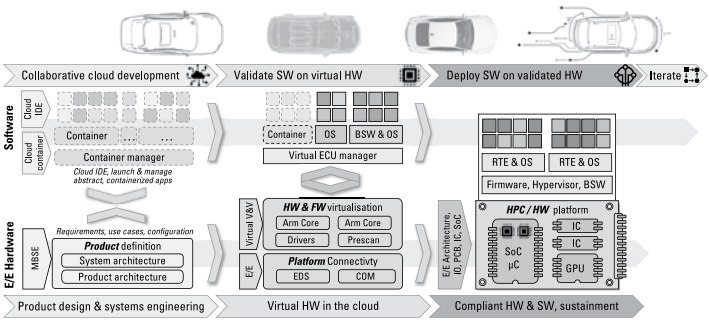

FIGURE 4-2: Cloud enabled SDV driven software and hardware development flow.

Establishing traceability from systems and electrics/electronics (E/E) into software enables a true end-to-end solution. This is not only for software component interface and network design, but also to aid E/E hardware circuit design, MBSE tooling, and AUTOSAR embedded software for the ECU or software partitions, virtual validation, n-the-loop simulation, and hardware testing.

This framework includes decisions based on metrics, facts, and quick iterations rather than guesses. Taking the systems engineering models and transforming them into communication matrices allows you to evaluate network load and CPU load as functionality is allocated into the architecture alternatives. This results in the ability to assess signal and processing delays across network and directly validate them against system requirements.

Another benefit of such an integrated development flow is to create a first-pass connectivity of the electrical systems. Integration to 3D feeds the resulting design models into the wiring and harness design processes and allows the data originating in the systems engineering models to be used for aftersales fault diagnosis and safety and compliance certification.

One key aspect of connecting systems engineering and software development in the context of SDV is to enable test and virtual validation against virtual hardware and software images, testing system level requirements against the implementation as you elaborate the design. As hardware matures, the virtual tests can move from an abstract, virtual representation of the targeted hardware components to the production hardware. The goal is to identify and fix critical design and implementation issues long before hardware samples are available.

It's a bit different for hardware development, which is more of a gated approach. Software development doesn't happen all at once but is agile, with sprints and pieces developed on the fly. Adopting a digital thread approach ensures that you're synchronized at key points within the engineering process.

Verifying the Result

Verification management is an overarching process that starts with requirements. It doesn't matter what the requirement is — it could be performance-related, a functional need, an attribute of the system, or a particular certification criterion. What matters is that some specific thing is required, and there needs to be a plan for verifying whether that requirement is satisfied.

REMEMBER

Indeed, for each requirement, you must develop a verification plan. How will you demonstrate that the requirement has been fulfilled and the intent has been met? Through some kind of analysis? By performing a physical inspection? By conducting an audit? Creating a physical test?

The bottom line is that verification management is the process of determining what the goal looks like. That verification plan explains exactly what you must do to be considered successful at meeting the requirement.

REMEMBER

As for managing the verifications of your A&D project, once you have successfully run through the verification plan for a particular requirement, you now have data showing compliance (or lack of compliance). Continuous verification from the start is key. Start compliant to stay compliant. Your MBSE digital thread includes all of this data for the entire program, collected automatically to help ensure compliance from the beginning. Every time a requirement is verified, it closes a loop.

Of course, nothing is simple. What happens if you find that you need to modify one system to fulfill a particular requirement? That may impact what you did earlier to fulfill a different requirement, and send a different team back to the drawing board. That's why you need a digital thread tracking the big picture and all the smaller images within.

Ultimately, your team's work isn't finished until you've closed all the loops and verified that all requirements have been met. Your systems engineering approach helps you ensure that each item is crossed off the list, and it does so quickly — notifying all others involved.

The digital thread gives a welcome assist to the highly complicated task of managing information. But it does even more than that.

Given the high level of regulation and oversight that is the way of life in industries such as aerospace, you need for the whole process to be transparent and traceable, from requirement through verification, including all applicable test data and analysis.

REMEMBER

Your team needs to confirm compliance to know when the job is done, and safety agencies also need proof that you've fulfilled critical requirements. The traceable chain of data made possible through the MBSE digital thread helps provide that proof.

Adopting Interface Management across the Complete Product Lifecycle

For every complex program, there are many different functional areas. There are likely multiple departments and locations within the company, a design center in one place and another department somewhere else.

You've also got multiple suppliers engaged (see Figure 4-3). Components of all kinds: avionics, hydraulic systems, and electrical systems, to name a few. Many different players from all over the world are involved.

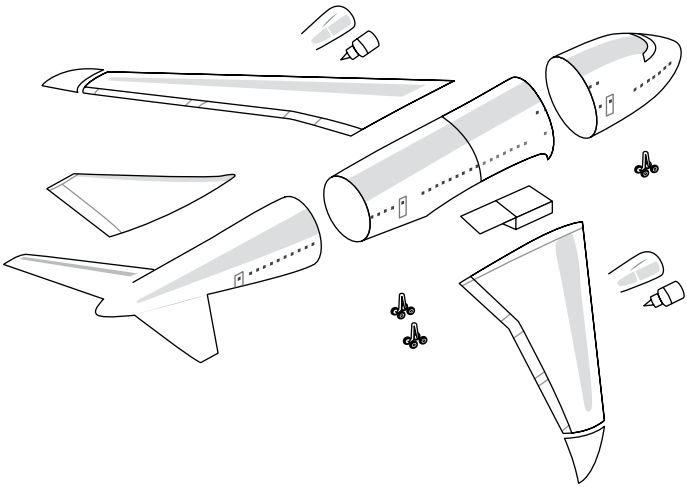

FIGURE 4-3: Increasing interface complexity of subsystems involving multiple suppliers.

Now add in the complexities of the interfaces between the work products of these many players. Most obvious are physical interfaces. Perhaps your horizontal tail comes from one supplier, and it must connect to a vertical tail from another supplier. All the more reason to have an integrated interface management system that's part of PLM on an open and flexible ecosystem such as Siemens Xcelerator, and — for software-related use cases — ALM. Interface definition happens during system modeling and carries throughout the lifecycle to enable the integration process. This interface has to be managed or nothing works.

Likewise, hydraulic systems must mate successfully with whatever they're controlling, with compatible connections and pressures. Electrical harnesses require functional connection points that match. Data networks must connect. And speaking of data, you're moving into digital interfaces that also must speak the same language to link one system to another successfully.

WARNING

Because A&D systems are so complex, there is ample room for error. Different sections of a plane don't fit together. Wiring harnesses are too short or too long, or there are two male connectors, or connectors with different numbers of pins. There are many ways to stumble, each of them potentially very expensive and consuming huge chunks of time from your program schedules. All of these are reasons why interface management is key.

Interface definition happens during system modeling and carries throughout the lifecycle to enable the integration process.

Automated control over this integration and all the various interfaces is vital. Think of interfaces almost like a contract between the departments or suppliers on each side of the connection. Your MBSE digital thread approach ensures that everyone is complying with the agreed-to interfaces.

As you develop your early designs and the necessary interfaces, those details can then be used to drive execution and guide the design and development program. You achieve much better visibility into changes and the impact of potential revisions. If you change a requirement, you'll have a much better assessment of what impact that change is going to cause.

You hear a lot of talk about systems modeling. This is one of the biggest values — using requirements to manage the interfaces. The interfaces, in turn, help manage integration more effectively.

Integration management is all about interface management and information management. You have lots of people and organizations within the supply chain. So, how do you exchange information across the supply chain? How will you communicate a specific design change from the original equipment manufacturer (OEM) to a supplier who must deliver a critical component?

A model-based design chain is the extension of the MBSE enterprise to the supply chain. By implementing such a design chain, product architectures drive the design through the entire supply chain, which supports a closed-loop product development process. These model-based design chains are becoming increasingly important as the digital transformation continues. OEMs recognize the need for a major overhaul of their product development approaches, including their supply chains. Today's systems are "systems of systems," and the level of complexity of these systems, along with increasing numbers of disciplines involved, almost demand that system architectures work in collaboration with their suppliers. Today's supply chains must be integrally connected to the overall MBSE environment.

Teamcenter has been mentioned as a key element with the MBSE digital thread. Here too, Teamcenter offers a digital backbone where systems architects within the OEM and supplier organization can digitally come together and access the authoritative source of truth.

Consider the fact that as you've built up a detailed view of functionality, you've divided your aircraft up into various systems. There are reasons why one player in the supply chain must have full visibility into a certain area but zero visibility into something else.

To address this issue, what's needed is powerful, holistic multi-domain simulation. These types of simulations can achieve robust interface management to mitigate the technical risk of both internal and outsourced portions of the design.

WARNING

Thus, you need to be able to carve content out of one model and give it to a supplier that is working on only that part. Your digital thread approach must support a multiple-domain view of any problem because if you had to build it all into one view of a very large and complicated system, you could end up with IP issues.

Monitoring Performance

Imagine taking a course in college in which you're doing important work every day, but the professor doesn't provide feedback very often on how you're doing. And you're working on a group project, but you don't really have any idea how the fellow students in your group are doing on their individual parts of the project.

That sounds like a recipe for anxiety at the least and failure at the worst. Frequent progress reports would help you know whether you're on track, and if you're not, a report of some kind might help you to course correct. That environment wasn't always the case in the past, but students are accustomed to an online portal displaying up-to-date indicators of how they're doing these days. And along those lines, parents of grade-schoolers often can use the same kind of portals to help ensure that their kids are doing okay.

A similar, although much more complex, reality applies in the world of systems engineering.

In the past, you got an update on various key performance indicators (KPIs) perhaps once a month. And then you sometimes had to analyze the information, and by the time that was done, weeks had passed. It would have been nice to have some actionable insights a lot earlier, but it just wasn't realistic or possible.

REMEMBER

Today's mission-driven systems engineering approach provides you real-time access into the status of how your project is performing — both from a technical perspective and in terms of cost, in small pieces, and throughout the whole program.

As the program advances, progress is visible through dashboards that display the many aspects of performance and the KPIs associated with various systems and requirements. How's fuel consumption looking? You have an up-to-date answer. Weight management? Aerodynamic efficiency? Your data is available in real time.

And it happens this way throughout the entire lifecycle, not through a pause to generate a static report card every few weeks, but in up-to-the-second information about how on-track you are as the plan matures.

The bottom line is: Performance must be checked and monitored in all phases, from concept, through 1D and 3D models and forward through physical tests. You must be continuously integrated — or "start integrated, stay integrated."

You're still using tools that are domain-relevant, but you're putting it all on the table in a collaborative engineering approach. MBSE brings together the various disciplines, including the all-important domains of safety and reliability.

IN THIS CHAPTER

- Determining where you are on the path to digital transformation
- Implementing tools and processes to achieve a mission-driven approach
- Mapping out your MBSE goals

Chapter 5
Achieving a Digital Transformation of Systems Engineering

This chapter focuses on the vision that materializes as you adopt this mission-driven approach to model-based systems engineering (MBSE) across your organization. It explores what it takes to bring stakeholders on board for a successful transformation and discusses things to look for as you choose products and vendors to achieve future MBSE goals.

Putting Your MBSE Plan into Practice

To create an aircraft, other aerospace and defense (A&D), or automotive & transportation (A&T) products, you'll be tapping into complex, multidisciplinary design teams. You can't do this work alone — you need to engage others and leverage all available knowledge. This mission-driven MBSE approach facilitates the kind of teamwork that must happen if you're going to succeed.

One of the main goals of system modeling is deciding on a list of capabilities and functions you want to be included in your product.

You also have to decide which discipline they fall under. By making these decisions, you're dividing problems to be more scalable.

Through the process, your team determines how you'll deliver different system functionalities. You'll identify what needs to be bought versus what can be developed. You're taking a journey through product logical and physical architecture, deciding whether a hardware unit will handle this particular function. And with what kind of software support?

REMEMBER

You must give portions of the concept to experts in that domain and bring their wisdom back into the program. It boils down to five basic patterns of architecture interoperability:

- Start with the end in mind to integrate the mission with the entire product lifecycle from the beginning.
- Capture architecture and use an integrated product architecture to align mission relationships, constraints, and requirements across the system lifecycle.
- Understand impact now in order to link architecture in Systems Modeling Language (SysML) to see the impact of changes immediately.
- Continuously utilize digital threads within SysML for cross-domain trade-offs.
- Drive software with systems to ensure continuous integration by carrying product architecture into software development to achieve closed-loop verification.

That last bullet, closed-loop verification, is a big deal. The system's expected behavior in a given context can be used for piloting verification.

In the case of autonomous driving and software-defined vehicles (SDVs), the verification must be executed on not-yet-defined test cases. There is simply not a single database of every possible scenario that may occur when driving on a public road that the vehicle can be tested against. Test cases may be created by applying machine learning onto recorded traffic situation or through means of artificial intelligence. Desirable results, parameters such as response time, and G-force exposure need to be created or derived out of requirements and descriptions based on the intended use cases or even from medical databases. The earlier in

the process that system models can be validated, the better. Earlier validation is also more cost- and time-efficient.

The vision of a digital thread that loops in all domains opens the door to the knowledge needed for your business process. For example, in many cases, you'll have suppliers who have content that you can incorporate (as discussed in Chapter 4).

If you think in terms of 3D CAD, people talk a lot about exchanging digital mock-ups (DMUs), which have been a powerful and helpful concept. The new vision takes that philosophy several steps forward — you're not just exchanging DMUs. You're exchanging models that include how systems interact with each other (signals, power, hydraulic pressure, control laws, and so on) and sharing that information. It's much more than pretty pictures.

Consider risk management. There are many processes for this, but it often boils down to a brainstorming process in which the program management team identifies risks, assigns severity, predicts cost, and gauges probability.

Inevitably, the group winds up with a risk category referred to as "unknown risks." Ultimately, "unknown risk" is a tacit admission that you don't quite have a complete understanding of your systems and processes to know all the risks.


TIP

With MBSE, you're able to take a more disciplined approach than a doomsaying brainstorming session. Having more information and control over technical workflows and more rigorous processes, you're able to identify potential risks more specifically and mitigate them before they happen.

Just as important, once you discover a risk, you can capture and document it in the connected, digital workflow and learn from the experience, rather than losing that knowledge when those who have worked on a project move on to a different role.

The MBSE vision is a story of building relationships between requirements, functions, geometry, and domains that otherwise would be in silos. That is key to harnessing intelligence and maintaining a handle on the constant change in a major program.

One functional area can perform an impact analysis, for example, and engineers in different domains can get messages of changes. You can explore the effects of hitting a bird or debris

on the runway, determine which functions are affected, explore what the backups might be, and make changes as needed to minimize the detrimental effects. With various kinds of system models (including behavioral models) linked through the digital thread, you have a vision of the total impact.

Altogether, it's a futuristic vision that can be done through the discipline of MBSE. But it's not sci-fi — it's reality, and many companies are well down this path already. These forward-thinking companies are more agile because they're better at managing and leveraging information. Once companies have fully adopted MBSE, the next step is to focus on capturing the mission, requirements of the mission, understanding the constraints of the mission, the inputs, outputs, and so on before developing a solution to ensure the mission is foremost in design consideration — enabling "mission accomplished" from the start.

This mission-driven approach is helping companies get to market faster, at reduced costs, and with fewer schedule glitches. They've cut risk, too. They're getting through testing and manufacturing with fewer failures.

In short, companies with a mission-driven approach are notching more wins. That's how the MBSE process helps companies be more effective and more competitive.

Optimizing Overall System Performance

If you want to implement a future state, the first step is assessing where you currently are on this path to digital transformation of systems engineering. Maybe your company has implemented MBSE and is looking to further streamline the process. Or perhaps your company is looking to adjust the kind of software they're using to achieve a mission-driven approach.

WARNING

Part of the challenge is checking your view of the world, to see how you might look at things differently. If your aim is simply doing the same things as before — but more electronically — you won't get anywhere near the benefit you're seeking.

Consider, for example, your relationship with suppliers. You need to transform how you interact with suppliers, deliver measurables, and sequence them in the product development process. That's more than making a better electronic connection.

In the case of A&T and SDVs, your relationship with suppliers may be completely different compared to how it used to be. If software is the function, it comes independently from the hardware. A high-performance-computing (HPC) unit may contain software from dozens of companies that will not want to share the source code or algorithms with their competitors. This demands a change in the collaboration model. Systems engineering models and MBSE can have a leading role in enabling the change and ensuring adequate security in the data.

As shown in Figure 4-2, sharing use cases, test cases, requirements, and interface descriptions generated from SysML models in a cloud-based collaboration platform is a great way to facilitate distributed validation. As a next step, providing virtual validation hardware enables virtual validation based on model-in-the-loop, software-in-the-loop, or hardware-in-the-loop simulation. Testing application software running on a virtual hardware image that's connected to mixed-fidelity virtual instances of sensors, actuators, and other control units allows you to validate software functions long before production hardware is available.

That said, the future state starts with the digital thread approach because the traceable chain of data is a critical part of the process. You're automating a massive matrix, bringing together the best tools for modeling, simulation, and design optimization and orchestrating them on a strong product lifecycle management (PLM) — or digital backbone — platform.


TIP

Perhaps a more logical place to begin, though, is tackling organizational culture. Employees need to be on board to adopt these new approaches and commit to making them work.

In many cases, the best way to initiate change is an incremental approach. Establish a vision, then pick out achievable targets to help people make positive changes. This approach fosters a culture of ownership within the organization and establishes momentum when taking the next step toward the final vision.


TIP

So, pick a pain point. For example, it's not uncommon to have multiple repositories of documents in different places. Move those to a PLM system and create new project parameters that are defined from the beginning of the process. Show how the pain is eased, prove the value, then move on to another pain point.

Aerospace engineers are clearly very smart individuals. Go to them with a proven solution and there's a good chance they'll accept it. Create those little victories, and bit by bit, you'll move people in the right direction.

Accessing a New Approach

Many companies are looking for the kind of transformation outlined in this book. The benefits are easy to see. So are the challenges inherent in pursuing change this drastic.


TIP

When thinking about the user, don't limit this approach to the systems engineering enthusiasts but also focus especially on the engineering domains. Using MBSE to deliver output that is relevant can directly offload tedious tasks. After all, additional workload without any perceived added value will not help create acceptance in your company.


TIP

Success depends upon choosing the right technology and partner to help implement or advance your MBSE process. It's far more than buying a software solution — the solution provider must bring a robust suite of technology, expertise, and partnering mindset needed to achieve successful adoption.

Begin with the technical requirements. A satisfactory solution needs to address areas such as:

- **PLM:** MBSE must fully integrate into a PLM solution. It has to bring together multidomain product development, including mechanical, electrical, electronic, and software domains. And it must fully address such considerations as configuration, baseline, workflow, cost, reliability, and manufacturability. The PLM solution must establish the digital backbone for what will be built, instruct players how to do it, and inform the downstream development processes.
- **Product requirements engineering:** Requirements should be about driving the product development process from the customer's perspective, so they should be part of your PLM. That way, they can be allocated to downstream functions, features, and product architectures, tracked by reports, documentation, and dashboards.

- **Product architecture and system modeling:** These capabilities within PLM capture product architecture across the entire product lifecycle, allowing a holistic view into design decisions and integrating the various domains across the product lifecycle.
- **System simulation management:** 1D models generate product data, so they must be part of your data management objectives in the same way as 2D/3D models.
- **Workgroup consistency:** An out-of-the-box (OOTB) standard methodology needs to be in place to guide the systems development process, ensuring that the entire team is working consistently.
- **Technical risk management:** Look for a product that includes a complete, model-based product safety and reliability approach. By adding reliability modeling to the product lifecycle, you can cut down on product recalls and really focus on creating safe and reliable products.
- **Change management:** Systems engineering components must be included in the overall PLM process. That means standard change management practices that keep tabs on requirements, functions, logical, physical, processes, interfaces, targets/parameters, and other details. Rather than manage changes separately, include them with global product change planning and management.
- **Program planning and systems engineering:** Resources and schedules should be committed by systems engineering architecture and requirement decisions. Requirements, functions, interface definitions, and the like should be tied directly to program milestones and project tasks.
- **Application lifecycle management:** Software and its requirements, function breakdown, interface definitions, test cases, build and release plans, agile development, continuous integration, delivery and commissioning of software are essential parts of the product.
- **Bill of materials:** The product will evolve — product updates, service, maintenance, software upgrades, and added features will change the product, as well as the individual instance, down to the tail-number or VIN number. Knowing which vehicle is equipped with which combination of hardware, software, and communication data isn't just nice to have but is also a legal necessity.

That is a lot to consider. But you'd be surprised at just how far the digital transformation has progressed and the types of solutions now available. The Siemens Xcelerator business platform is one such example. Siemens Xcelerator is a comprehensive, integrated portfolio of software, services, and an application development platform. It includes both a comprehensive digital twin and the MBSE digital thread (discussed throughout this book), along with an open, scalable, and flexible approach to helping companies become digital enterprises.

REMEMBER

That's a good shopping list for the tools needed. Equally important is having a methodological backbone in place to support these tools and solutions. The vendor you select should also be keenly aware of such methodologies, as these are critical to a successful mission-driven MBSE environment.

Ultimately, you need a technology vendor with deep engineering expertise and the technical know-how required to develop mission-critical systems. Look for a vendor that is ready to help you with the paradigm shift on your journey to achieving digital transformation of systems engineering.

Chapter 6
Eleven Ways to Win with MBSE

For many years, aerospace and defense (A&D) engineering has been on a hard-to-sustain path of ballooning development cycle times and skyrocketing costs. New aircraft designs are increasingly reliant on electrical, electronics, and software systems. As with just about everything else in the world, there are pressures to deliver more innovation, more sustainability, faster, and at a lower cost.

In the automotive and transportation (A&T) sphere, software-defined vehicles are at the heart of an automotive revolution. New cloud-based technologies, such as edge devices and containerization, are being adopted by the automotive ecosystem, not only for operation but especially for implementation of in-vehicle and off-vehicle functions. Future vehicles must be built based on integrated, collaborative development. They must use a model-driven approach incorporating systems engineering principles.

The model-based systems engineering (MBSE) approach promises gains in all of those areas. The following are some ways your organization can win by adopting or expanding on MBSE, and

some of the lessons to remember as you move forward in your digital transformation journey.

Understand the Mission

The mission-driven approach to systems engineering is all about enabling "mission accomplished" from the start. By adopting this mission-driven approach, you can ensure end-user mission success by managing the technical baseline, beginning with the business understanding of the mission or product and extending through the end-user execution of the mission or product.

The unique ability to take systems optimization beyond an individual domain or part — and optimize across the entire product and lifecycle — is the future of MBSE.

Transform the Process

To be clear on what's needed — transform, transform, and transform again. Adopting or expanding the MBSE approach certainly requires bringing on the right product lifecycle management system that builds on a digital thread and includes a wide array of integrated tools for modeling, product architecture, change management, and so forth. But in the end, it's not just a matter of adopting new software. Adopting a mission-driven MBSE approach involves a transformational mindset that must spread throughout the entire organization. You won't win if you don't transform.

Remain Patient

You need to think about the whole process, end-to-end, and be open to transformation. At the same time, remain aware that many players in this program already have tools in place working for them. You need an MBSE environment that's transformative but also open, flexible, and adaptable to processes that already exist or that key domains are hoping to put in place.

Redefine System Modeling

As you implement or mature your MBSE techniques and tools, a central part of the picture is system modeling. You'll use the system modeling way of thinking and as an architectural representation of your design, and that's a paradigm shift in the way you start building models. Take your architectural description and make it domain-specific — you can do that all over the design cycle.

Embrace Seamless Collaboration

Some processes take time. Working through an A&D program can move faster and more effectively when you have subprocesses running simultaneously in a collaborative way. The important thing is that you don't have separate engineers or supply chain partners in their silos, creating their own subprocesses without regard to the big picture.

A holistic, digitalized approach to MBSE removes the silos so that these collaborators can work more dynamically. They understand the various phases of development, as well as the impact of changes. And regarding the supply chain, MBSE aligns and integrates with supply chain partners, so all requirements are understood and defined, alleviating potential problems from happening in the first place. With real-time access from anywhere, MBSE users get best-in-class collaboration that is secure, scalable, and flexible.

Keep an Eye Out

Your MBSE solution should provide continuous performance monitoring. Seeing key performance indicators (KPIs) — *in real time* — is an essential component that helps everyone involved gain confidence in developing a product and ensures that all performance targets and program requirements are being met. This includes the level of maturity, the design itself, the relationships, the requirements, the verification of those requirements — just about everything that happens on both sides of the V diagram (see Chapter 2).

Understand the Importance of Traceability

Traceability is essential, and it's one of the key benefits of your MBSE digital thread. You're using MBSE to organize the whole program — the data, dependencies, activity, and all the interactions that happen along the way. Having a traceable process and traceability at your fingertips in the design solutions will pay significant dividends, and your tool choices can enable and automate traceability. For example, System Modeling Workbench (SMW) for Teamcenter, with its Arcadia method backbone, enables such traceability.

Make It Real

You can implement the best tools in the world, but you won't get anywhere if you don't have everyone on board. You need solid buy-in, but before you ever get to buy-in, you need people to understand what you're trying to do. On the other hand, many early adopters of Systems Modeling Language (SysML) believe that SysML is MBSE. But the reality is that SysML is merely an enabler of the overarching MBSE approach. Once you understand these distinctions, you'll be well on your way.

Close the Loop

Much has been said about the V diagram that underscores the systems engineering process. But it's also worth thinking in terms of an O-shaped diagram. Not actually the letter O, but a circle that illustrates the ongoing, back-and-forth connections among requirements engineering, system modeling, analyses, safety compliance, and managing technical content into interface, integration, and verification.

The MBSE paradigm is a continual path connecting these elements that all inform one another. That's true for systems engineering in the big picture and later in the process within the various domains, such as software, electrical, electronic mechanical/physics, and electronics/hardware.

Get Physical

Models and simulations are remarkable things. MBSE allows you to benefit early in the process from insights that, in the past, would only have become evident much later. Those insights drive better requirements and make for a much more efficient journey. It's sometimes easy to forget that you will have physical testing to do. That is simply a fact: The more you innovate, the more you will at some point need to test your innovations. But make no mistake, when you digitally transform your approach to systems engineering you significantly reduce the need to do physical testing. By remembering the importance of continuous connectivity, you can leverage the comprehensive digital twin to link between physical and virtual worlds.

Partner

Digitally transforming your MBSE process isn't a simple nor a singular task. It may require continuous organizational transformation. Look for partners, internally and externally — seek advice when needed and helpful to ensure that you continue to have the right tools for your MBSE journey.

Digitally transform aerospace engineering

Siemens' approach to aerospace systems engineering helps fix the unsustainable impact of late-discovered issues due to disconnected systems. It ensures end-user mission success by connecting systems, domains and stakeholders across the complete product. The power of Siemens Xcelerator portfolio, including multi-domain simulation, takes systems optimization beyond an individual domain or part and offers visibility of the complete system of systems in context of the whole product and across the lifecycle to keep program deliveries on schedule.

siemens.com/mbse-aerospace

SIEMENS

Adopt a mission-driven approach to systems engineering

Model-based systems engineering (MBSE) unlocks a whole new way of orchestrating your aerospace and defense (A&D) engineering operations — covering the entire product development lifecycle. Take MBSE to the next level by adopting a mission-driven approach to aerospace systems engineering to connect systems, domains, and stakeholders across the complete product and lifecycle, ensuring cost-saving risk mitigation up front.

Inside…

- Understand a mission-driven approach to systems engineering
- Leverage the comprehensive digital twin
- Integrate early, mitigate risk
- Enable robust interface integration
- Connect systems, domains, and all stakeholders across the lifecycle
- Achieve digital transformation of systems engineering

SIEMENS

Steve Kaelble is the author of many books in the For Dummies series, and his writing has also been published in magazines, newspapers, and corporate annual reports. When not immersed in the For Dummies world or writing articles, he engages in healthcare communications.

ISBN: 978-1-394-28266-1
Not For Resale

for dummies®
A Wiley Brand

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.