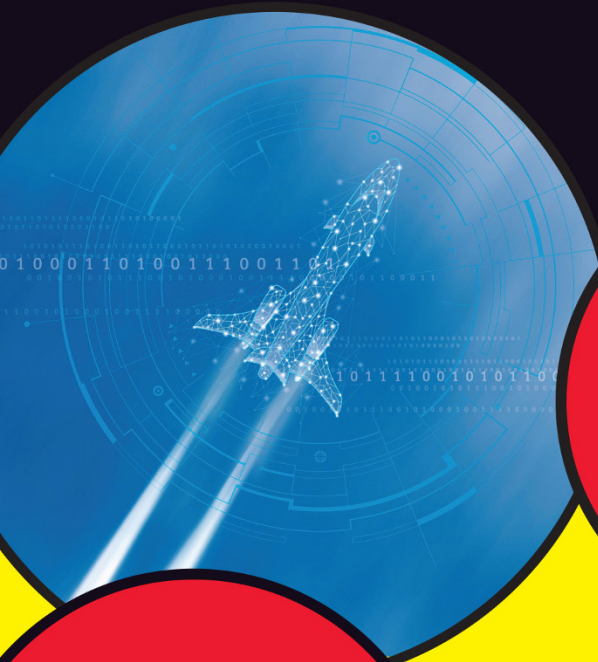


LEARNING MADE EASY

Siemens Special Edition

MBSE

for
dummies[®]
A Wiley Brand



Elevate system
modeling

Transform the
design process

Innovate like
never before

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

Siemens Special Edition

by Steve Kaelble

**for
dummies®**
A Wiley Brand

MBSE For Dummies®, Siemens Special Edition

Published by

John Wiley & Sons, Inc.

111 River St.

Hoboken, NJ 07030-5774

www.wiley.com

Copyright © 2022 by John Wiley & Sons, Inc., Hoboken, New Jersey

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Trademarks: Wiley, For Dummies, the Dummies Man logo, The Dummies Way, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and other countries, and may not be used without written permission. Siemens and the Siemens logo are trademarks or registered trademarks of Siemens Trademark GmbH & Co. A list of relevant Siemens trademarks can be found here. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc., is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, or how to create a custom *For Dummies* book for your business or organization, please contact our Business Development Department in the U.S. at 877-409-4177, contact info@dummies.biz, or visit www.wiley.com/go/custompub. For information about licensing the *For Dummies* brand for products or services, contact BrandedRights&Licenses@Wiley.com.

ISBN 978-1-119-79316-8 (pbk); ISBN 978-1-119- 79317-5 (ebk)

Publisher's Acknowledgments

Some of the people who helped bring this book to market include the following:

Project Manager:

Jennifer Bingham

Acquisitions Editor:

Ashley Coffey

Editorial Manager:

Rev Mengle

Business Development

Representative: Matt Cox

Content Refinement Specialist:

Vivek Lakshmikanth

Cover art: Image created by Oweb

www.oweb.es.

Special Help from Siemens:

Dale Tutt, Sonya Sauve,
Mark Malinoski, Vincent Braibant,
Mark Sampson, Thierry Olbrechts,
Stefan Dutre, Tom Behrens,
Tony Komar, Steven Vickers,
Shashank Alai, Scott Salzwedel,
Robert Lyons and
Indrakanti Chakravarthy

Introduction

Airplanes and other aerospace and defense (A&D) products are incredibly complicated. So are the systems engineering processes bringing them to life. Every innovation adds to the complexity.

At the same time, A&D organizations face growing demands to innovate quicker while reducing costs. That would seem to present competing priorities. Among other challenges, the more complex the product, the more likely you are to encounter integration issues, which can slam the brakes on a project.

As the product evolves, so must the processes for creating it. New digital processes are driving revolutionary ways to design components and whole systems. This transformation connects disciplines and allows teams to see how systems interact and evaluate the impact of revisions across the entire design. This approach is known as model-based systems engineering (MBSE), a methodology that uses models rather than documents to orchestrate product development, requirements flow down, and overall integration.

Traditional systems engineering is its own niche function, not really plugged into the rest of the development process. A model-based approach to systems engineering, however, can be the overarching driver of any program.

MBSE provides insights during conceptual brainstorming. It can drive requirements down into the detail level of the various domains, with complete validation and traceability. It can help predict, identify, and address problems proactively before they spiral into delays and cost overruns.

Foolish Assumptions

In preparing this book, we're making a few assumptions about you, the reader:

- » You play a vital role in the engineering process for A&D programs, maybe as a program manager, an executive leader, or a systems engineer.

- » You may know about systems engineering and perhaps MBSE, but it's not clear how you can make MBSE work in your organization.
- » Your time is valuable, so you'd appreciate a quick read that will inevitably make your work life easier.

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

We hope this book will provide actionable insights, and 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 is a few dozen pages, which only scratches the surface of MBSE. If you come away with an appetite for more, here are some resources:

- » Model-based Systems Engineering: Information from Siemens about solutions that can help you along the path
- » Accelerate Product Development with MBSE: A video on how MBSE can orchestrate the technical program
- » Talking Aerospace Today: An informative podcast from the Siemens Aerospace and Defense Industry team

- » Flying high with A&D innovations
- » Embracing the challenges inherent in change

Chapter 1

Soaring With Aerospace & Defense

Few industries fuel the human imagination as intensely as the aerospace and defense business. 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 the Wright Brothers to Neil Armstrong to Elon Musk.

This chapter explores the fast-changing aerospace and defense sector and its embrace of innovations that will become tomorrow's books, movies, toys, and games. It outlines visions for the future, factors driving change, and how difficult it is to keep up with the forces of evolution and revolution.

Taking Flight

Innovation is the name of the game in the aerospace and defense sector (often known as 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. 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, it also means serving 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, causing A&D product developers to keep greener alternatives on 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 may enable a mechanical system to move to a software-based capability that's both speedy 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 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. 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-driven, not just in A&D but virtually everywhere. Again, this enables greater control and flexibility, improved efficiencies, and futuristic capabilities. But it's making engineers' lives even more stressful.

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 shifting and 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 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, of course, is an ever-present issue that gets attention from the industry, the public, and regulators. The issues that the Boeing 737 MAX experienced are just one example of the way safety is more important than everything else. To succeed, aircraft manufacturers must consider every conceivable hazard from turbulence and thunderstorms to bird strikes to human error across the design 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 consider how to 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 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."



REMEMBER

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. There's more integration. 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 or helicopter or rocket, or whatever the final product may be. You're developing a system of interconnecting systems. They all must work in unison, where one system works affects many others.

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.



REMEMBER

Each system has teams of engineers bringing it all to life. Helping all of these equally important players collaborate effectively is the aim of model-based systems engineering, and that's what this book is all about.

- » Understanding systems engineering
- » Taking on the complexities
- » Innovating digitally
- » Adopting a model-based approach

Chapter 2

Rethinking Your Systems Engineering Approach

The aerospace and defense (A&D) industry creates all kinds of exciting things, all of them incredibly complex (for more on this, see Chapter 1). An airplane, 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 link these systems. And the creation of the many systems on an aircraft 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 provide these new options and give a basic understanding of the 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 life-cycles. It's a structured and robust approach guiding design, development, production, and ongoing maintenance of a super-complicated 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 composer of that beautiful music.

Meanwhile, the International Council on Systems Engineering defines model-based systems engineering 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 requirements are fulfilled.

The scope of systems engineering continues past design modeling and simulation, optimization and verification, into production and ultimately, into product operation, retirement, and disposal. Systems engineering ensures 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 the creation of 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 all specified requirements are met.



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 the design will be safe. It's on their mind as requirements are set, as the function of the design is considered, and implementation unfolds. They write new requirements based on the safety analysis — creating a 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.

The V model of systems engineering, illustrated 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.

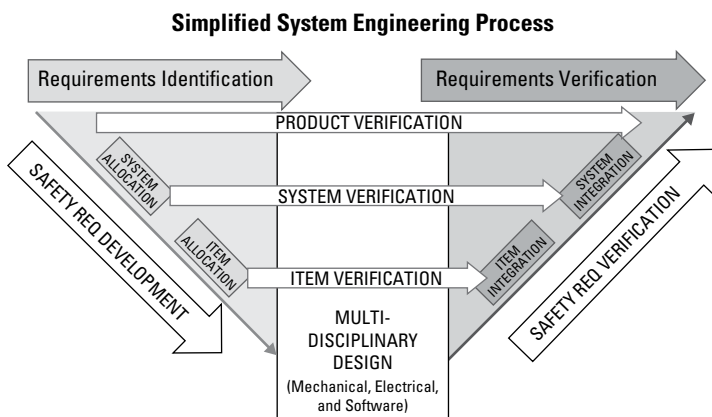


FIGURE 2-1: The systems engineering V process, simplified.

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.

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, 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. The systems engineering process tends to orchestrate, while the rest of processes execute. It's the glue that holds the system development process together. Put another way, the systems engineer is a program manager's best friend.

The 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 an approach that's model-aided and design-automated. It's leading the process through multiple conceptual iterations that define requirements and the product architecture. It's facilitating a clean handoff from conceptual design into detail design, on through verification, and finally into manufacturing and product support.



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 systems engineering. A CAD-like, model-based methodology and toolset make that possible.

Keeping Up with the Complexities

It's worth discussing a bit more about why we need an updated approach to systems engineering and why the practices of the past few decades are not enough for the needs of the 21st century.

You may not have even been born yet when systems engineering came into the picture, but you know that technology was a lot simpler if you were around half a century or so ago. Back in that day, the kind of computer we are using 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, sometimes even on paper. Everything was document-centric, hard to effectively manage in a big-picture kind of way.

In A&D design, more and more functions were interconnected and consolidated, 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.

From the software perspective, as the complexity of an aircraft increases, so too does the complexity of the software. That puts new challenges onto software development and its timeframe, especially as you combine that complexity with the challenges of certifications. You need tools that can move quickly as you integrate software with hardware, 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 complex systems will behave appropriately and safely?

What's required is a single, 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? They're growing in number and complexity.

They're established in individual nations (and sometimes regions), in the European Union, 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, especially 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 for page after page after page, and leave you as a reader feeling hopeless. And that would be wrong, because this complicated situation is anything but hopeless.

Indeed, there is a need for a more disciplined, digital approach to systems engineering. You really can't rely on documents to get a handle on the complexities. 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. Two decades ago, systems engineering was already an elephant. Now it's thousands of elephants.

Fortunately, this is an increasingly digital age, 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 brings a multiplying effect to teams using a digital twin by enabling numerous data processes across multiple systems. Merging the physical and digital worlds with a digital thread enables systems engineering teams to predict performance and optimize their product. Teams can effectively deliver on their programs in a proven and secure manner.

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 is not serial processes within a single function, nor are these functions intended to operate independently of one other. The exact opposite is true. With the digital thread, task automation is achieved and functions are interconnected, integrated and linked so 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 rethink them. You still have a herd of elephants to contend with, but it's no longer a circus with a new trainer on the job.

Meanwhile, digital innovations in the world of systems engineering continue to march forward. SysML is just one example. The open-source Systems Modeling Language grew out of Unified Modeling Language, or 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 innovation 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 V1 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.

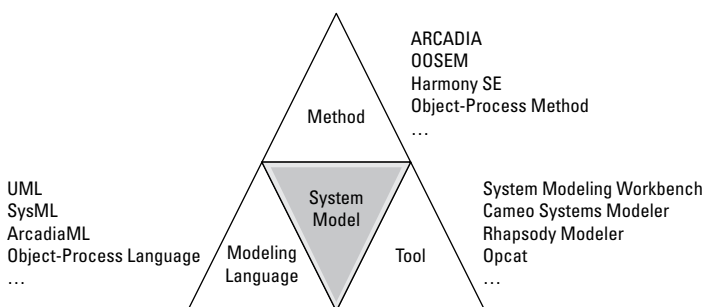


FIGURE 2-2: MBSE Essentials Pyramid.

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

Turning to MBSE

More and more companies are turning to MBSE processes to manage product development, requirements flow down, and overall integration. These are all things that traditional systems engineering has attempted to tackle — 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.



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 doesn'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 no dynamic view of the program as a whole. Thinking again in terms of an orchestra, it's as though the woodwinds are playing from one score, and the strings are following another. It's more of a cacophony than a symphony.

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. (Incidentally, there are numerous digital threads available today. In addition to the MBSE digital thread, you can find a digital thread for Intelligent Manufacturing or Product Support, for example.) Through the MBSE digital thread, models are shared from one domain to another and also with suppliers as needed. The information moves back and forth smoothly and efficiently — and when compared with dealing with countless separate documents, it's far easier to manage.

The concept works miracles with 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 important, it's a recipe for far better control of and visibility into the entire technical program, 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. MBSE supports a 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 a model-based systems engineering approach, simulation becomes more than just simulation — it can also help make informed decisions about what architecture is the most appropriate. 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

include costs, maintenance, and reliability. Because you can get quick turnaround of 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 be helpful at times. Think in terms of the computer-aided design space. You can draft a component and create 3D digital models. And you can then combine components until the point that 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 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.



REMEMBER

In short, by adopting MBSE, you're moving away from a niche modeling approach, with several very specialized individuals producing architectures and analyses. You're moving instead to a digital thread view where multi-domain architectures are used to propel and orchestrate product development, incorporating key design decisions to fulfill the products' performance targets.

- » Connecting with the digital thread
- » Taking control of the technical program
- » Deriving data from disparate tools
- » Doing things better

Chapter 3

Painting a New Vision for the Future

Model-based systems engineering (MBSE) unlocks a whole new way of orchestrating your aerospace and defense (A&D) engineering operations. It's a vast step beyond document-centric methods that required a whole lot more work. Using a model-based approach drives greater insights throughout the program, builds more effective connections and more scalable processes, and gets all players working together in new ways.

This chapter explores how the digital thread 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 like. And it describes ways your processes and results can improve.

Weaving the Digital Thread

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 he left at home while he's on vacation.

Clearly, with a complex project, this haphazard 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 in MBSE solves this problem. It centralizes information, making it visible and accessible to those who need it.



REMEMBER

That's great, but it's really just a tiny piece of the value of the digital thread. Each element in the program is connected to the thread, and together they create a fabric. The weave of that fabric offers visibility into all things impacted by any particular change.

The digital thread starts at the front end of the program lifecycle. It's established by decisions made by architects there. 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.

The digital thread is, in short, a connection of everything. It allows you to drive your desired functionality to the system level, connecting with the logical architecture – across the entire product lifecycle.



REMEMBER

MBSE maintains the thread of behavioral, structural and interface requirements throughout the program. The thread of requirements influences how the system is designed, tested, and delivered.

The digital thread must comprehend and prioritize verification. Developing an aircraft virtually is a tremendous innovation, through virtual modeling and simulation — physical testing alone is no longer sufficient. 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.

Ultimately, the MBSE-driven digital thread improves program execution by allowing you to:

- » Define your system of systems using modeling to continuously refine product requirements and architecture.
- » Turbocharge your design space by optimizing multidisciplinary design, ultimately leading you to the best design faster than ever.

- » 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/programs, improving program-specific execution.
- » Manage integration across the supply 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 authorized team members can view files and author changes.

Orchestrating Your Technical Program

Who wouldn't want to orchestrate the technical scope of program more effectively throughout the entire product lifecycle?



REMEMBER

MBSE changes how everyone interacts across the process. It means rolling 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 A&D development, as the players are disconnected. Without the insights delivered through MBSE, changes in one domain may not immediately be visible in other places, including program planning and execution.

MBSE is a new way to work. It's like moving from black-and-white television to HD color. It's like evolving from a rotary phone to a smartphone. It really is that much better.

With a digital thread in place, 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.

A single digital thread drives the entire program. The digital thread builds tremendous synergy between players across various areas. Starting with 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 from Siemens.

As illustrated in Figure 3-1, 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.

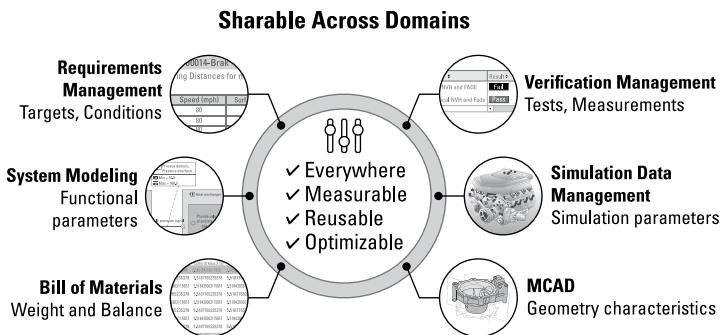


FIGURE 3-1: Managing parameters across multiple domains.

Here are more dreams you can fulfill by using MBSE to orchestrate the technical program:

- » Eliminate the integration issues between complex systems and between suppliers before moving into detail design.
- » Easily trace all design decisions within the product from the concept phase to the final certified product.
- » Trace final verification and certification results back through design and analysis, back to the original requirements.
- » 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 product lifecycle management (PLM) system, such as Siemens Teamcenter. With this particular MBSE digital thread, the focus is on orchestrating the technical program and driving the technical scope, by moving from system modeling alone to a highly digitalized approach.

This comprehensive digital approach tracks all technical requirements and architecture implementation from conceptual design to verification. It captures design decisions as the product matures. All of this can be accomplished securely in the cloud.



WARNING

That last point is vital considering 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 have left the company) isn't the best course of action.

The Teamcenter PLM approach lets users integrate all the product design and supplier interfaces within a very flexible and open multi-tool solution.

Making it Comprehensive

MBSE doesn't work unless it works for everybody in the program. It's one thing to have an active digital thread, but another to create content in the thread. This must be made possible regardless of the tool the content creator prefers to use.

The thread itself is created with a database structure to organize the data. But in the interest of allowing comprehensive information and full compatibility, that data can come from multiple different tools.

The caveat, however, is something that's truly comprehensive can quickly become overwhelming. As you connect into the digital thread to exploit the details of a certain point, you want only the right details to be shared. To borrow a term from texting, you don't want TMI — "too much information."



TIP

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

Getting Results

In the realm of A&D 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 by 25 or 30 percent. They're getting through programs with reduced risk, too.



TIP

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

MBSE's digital thread brings new levels of scalability and breadth of focus into the modeling process. With MBSE, your gaze can shift from the big picture to the finer details and from component level to systems of systems. As an example, you must track braking behavior within the context of the landing gear, but also certify it at the integrated aircraft level, while ensuring compliance with the global aviation infrastructure.

Achieving certification and compliance depends on meeting the engineering challenges. Through MBSE you can model not only the form and function of a particular item but also 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 OEMs, often threatening to derail programs. MBSE and the digital thread have proven their ability to eliminate many major integration issues.

- » 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 all Domains with MBSE

This chapter dives into the nuts and bolts of how model-based systems engineering (MBSE) gets its hands into all of the various domains that collectively create new A&D products. It offers more detail on the digital thread that weaves the whole program together and keeps all players informed and playing from the same orchestra score. It details requirements management, system modeling, the key role of software, the critical importance of safety, the need to verify every requirement, the challenge of integrating many operations, and the need for up-to-date data.

Getting with the Program

Chapter 2 referenced 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. Simply put, all the definition and distillation of requirements occurs on the left, and the detailed work of integrating the many interconnected systems takes place on the right. As you climb up

the right side, you're verifying those requirements and ensuring that there are no problems.

That sounds like a fine approach, but it can seemingly take forever to get from one side of the V to the other. And these days, no one is willing 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 a potential runaway situation.

What's more, the reality is that you can't view the V as simply a here-to-there proposition. In one part of the V, there are changes made, problems addressed, and important realizations, which all 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 digital thread through the entire process — from conceptual design, through preliminary design, through detailed design, to the test and build stage, to certifications and determination of airworthiness. It's the MBSE thread that guides all other threads that are supporting various functional teams and domains. Such areas might include product design, verification, manufacturing, and program and supplier management.

The ambition of solution providers, such as Siemens, is to compose the entire product lifecycle process along the digital thread, from design to manufacturing to service, and all points in between. MBSE crosses the whole lifecycle of the product.

It also crosses between the many different tools found across the individual domains. For example, a CAD engineer works in one world, and a simulation engineer takes CAD data to see how it performs, using a different tool in a different world. These simulation results 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. The digital thread weaves it all together in a way that can make sense for everyone.

As outlined elsewhere in the book, systems engineering processes have been around for decades. Still, today's aerospace and defense (A&D) challenges necessitate a move from document-centric to model-based methods to more of a comprehensive model-based

systems engineering approach. That's the only way to effectively manage product development, requirements flow-down, and overall integration of design, analysis, validation, and verification activities.

Managing Requirements

The thread of your project begins with a determination of features and functions — if it's a plane, you begin with the general vision of what 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 plane can fly 2,000 miles, with a specific number of passengers and a particular top speed. You're getting a general idea here, but you're running out of room on the napkin already.

These features and functions lead to requirements that inform the architecture, and the bigger the project, the more requirements you're defining. Designing a big plane? You're likely talking about hundreds of thousands of conditions. Definitely, more detail than your cocktail napkin can handle.



REMEMBER

Thus is the challenge of managing requirements. As you work through the flow-down of requirements, from big picture to smaller and more specific systems, you encounter more and 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 to which it will be attached, the drag, the fuel tanks. 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. 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 for which 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 as 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.



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. Not just A&D work but also your tax planning or the details of a home remodeling project. Most likely, you've written those numbers down or plugged them into a spreadsheet on your computer. It's document-based, one way or another.



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 airplane.

That's why the MBSE approach links requirements in a database, where each requirement is an individual object which is 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. Better to extract values in the requirement, create measurements on the geometry of a component using a 3D model in CAD systems such as NX. Further, users may choose to plug in the metrics of a performance-based requirement such as stresses or strains or temperatures, or volume flows into CAE systems such as Simcenter.



REMEMBER

Your digital thread can flow through all of this. It can tie into the relationships between requirements and logical architecture, and, ultimately, physical components. Understanding and accounting for these relationships is the key to wrestling the complicated beast of A&D engineering complexities — so you can trim both costs and development cycle times. And everything is verified, so you do not have a conflict of requirements.

Modeling the System

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



Through such technology as Capella and tools such as Siemens' System Modeling Workbench (SMW) for Teamcenter, you're starting with a top-level set of requirements to develop these functional models to examine how different systems are doing the work. The system architecture model must be integrated with PLM 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 sub-functions. You're doing the same thing when modeling control of the airplane in the air, considering such sub-functions 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 above example, 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 digital thread winds its way through parameters (system design attribute, performance metric, etc.) 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. The MBSE digital thread takes various forms while winding through architecture modeling in SMW, starting from high-level mission concept through identifying the physical components and their behavior. These various digital threads then traverse outside of the architecture model, connecting the various domain design implementations through the realization of the final physical product and beyond. Much of this can be achieved via collaboration through the cloud with product lifecycle management tools to plan, develop, and deliver aerospace programs faster. See Figure 4-1.

Indeed, system modeling is not 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 multi-physics, MCAD, ECAD, DFMEA, RAMS, and so on.

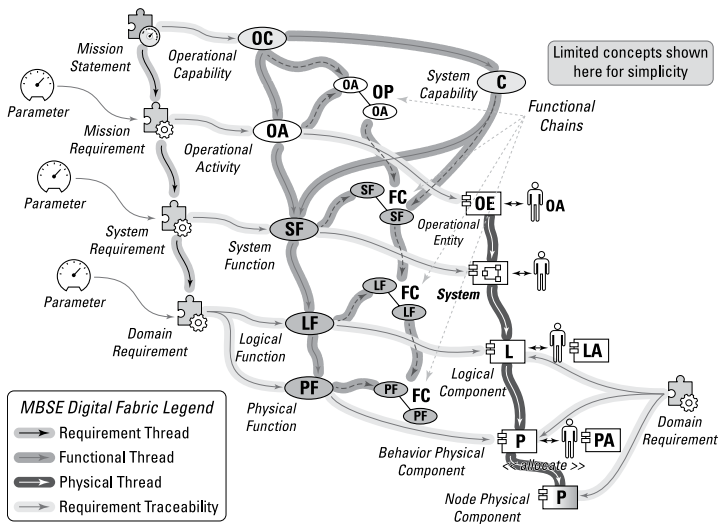


FIGURE 4-1: MBSE Digital Fabric within SMW.

Ensuring Safety

In the world of A&D, nothing could be more important than safety. The safety process is, in fact, a sub-process in overall systems engineering.

This runs through the process of 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 need to plan for parallel, redundant electrical systems. You must ensure that the loss of one system won't cause loss of the aircraft.



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.

Focusing on Software

As with so many other products we see today, the use of software in aircraft has been increasing dramatically. Software development starts with requirements definition and considers the level of integrity you have 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 software systems. In the area of software, it's vital to bring it all into one place since software creation uses a wide range of 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 all kinds of 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 driven 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. The critical importance of software is similar in A&D but on steroids.

This underscores the importance of software within your modeling environment and the digital thread you're building. 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 SMW can effectively enable model-based software architecture modeling, supporting numerous process improvements. These improvements range from concept

space exploration, requirements elicitation, and specification, predicting early behaviors, to even automating test-case generation.



TIP

Integrating software with PLM and plugging it into a solution such as Polarion helps ensure that you're fully addressing the impact that software has on the overall product and its capabilities. 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, not as obvious. Still, its presence has every bit as much an impact on operations and capabilities as all of the aluminum, composites, and every other physical element.

Software development is happening in parallel with every other activity needed to produce the product. As the software team creates its part of the ecosystem, it must move at the right speed.

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. Your digital thread 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, a particular certification. 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?

For example, perhaps there's an electrical system requirement related to resilience during an electrical storm. That may necessitate a physical test of a component in a lightning chamber. Other physical tests may involve ground testing or flight testing.

The bottom line is, 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.

As long as we're talking in terms of goals, you can look to sports to illustrate how this works. In basketball, for example, the overarching requirement is getting the ball into the rim and through the net. That's the equivalent of getting your plane safely into the air.

But there are smaller requirements that must be fulfilled along the way. A player must successfully inbound the ball to a teammate, and there may be a time limit for initiating that inbound pass. There also may be a time limit requiring that a shot be taken. The ball must be passed and dribbled if a player is moving. Referees are handling the verification of these steps along the way to the basket.

Meanwhile, your players must steer clear of various kinds of fouls or missteps. Again, the verification plan is handled by referees. Ultimately, your team will never score a basket if you can't successfully navigate and fulfill a lengthy series of requirements.



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). Your MBSE digital thread includes all of this data for the entire program, collected automatically. 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 already 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 of the smaller images within.

Ultimately, your team's work isn't finished until you've closed all of the loops and verified that all requirements have been met. Your digital thread helps you ensure 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.



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. Your digital thread may provide that proof.

Managing Integration

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 4-2). Components of all kinds: avionics, hydraulic systems, and electrical systems, to name a few. Talk about "it takes a village!" In these situations, you involve lots and lots of villages, potentially all over the world.

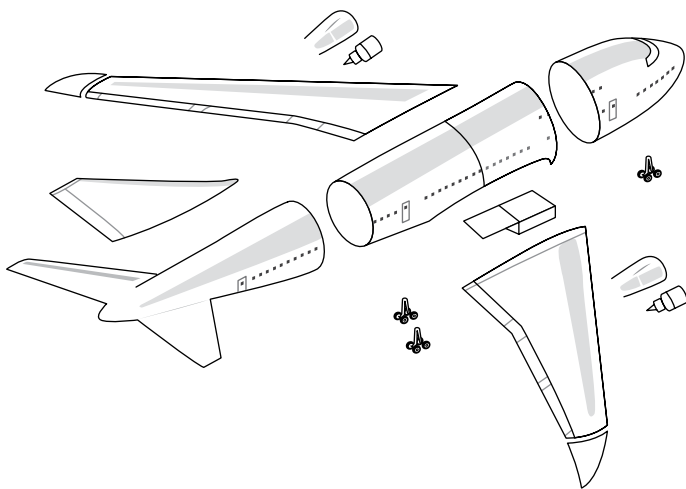


FIGURE 4-2: Increasing interface complexity of subsystems involving multiple suppliers.

Now add in the complexities of the interfaces between the work products of these many villagers. 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. 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

There are many opportunities for failure. 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 for the kinds of integrated interface management that is part of PLM. Interface definition happens during system modeling and carries throughout the lifecycle to enable the integration process.

Automated control over this integration and all of 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 program, through its digital thread, 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 up and down and across the supply chain? How will you

communicate a specific design change from the OEM to a supplier who must deliver a critical component?

Model-based design chains are becoming increasingly more 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 quickly becoming "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. No question today's supply chains must be integrated into the overall MBSE environment.

Teamcenter has been mentioned as a key partnering software with the MBSE digital thread. Here too, Teamcenter offers a common PLM environment where systems architects within the OEM and supplier organization can digitally come together.

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.

Perhaps you have a supplier of a certain avionics system. You may not want that supplier to be aware of what the payload crew is working on. You don't necessarily want to provide intellectual property from one domain of the system to another part of the system.



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 must support a multiple-level 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 are 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 their kids are doing okay.

A similar reality applies in the world of systems engineering.

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



Today's MBSE digital thread, on the other hand, gives 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, as well as throughout the whole program.

As the program moves along, progress is visible through dashboards that display the many aspects of performance and the key performance indicators 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 a pause to generate a static report card every few weeks, but up-to-the-minute information about how on-track you are as the plan matures. That sure beats those days when the only time you had a truly complete snapshot was when you were flight-testing the aircraft and certifying, with fingers crossed.

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 — a common bit of advice is “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.

- » Determining where you're headed
- » Making the future happen today
- » Finding the right expertise

Chapter 5

Charting the Path Forward

This chapter focuses on the vision that materializes as you adopt 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.

Painting a Vision

To create an aircraft or other aerospace and defense (A&D) products, you'll be tapping into complex design chains. You can't do this work alone — you need to engage others and leverage all available knowledge. An 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 what things will be accomplished and in what discipline. You're dividing problems up to be more scalable.

Through the process, your team determines how you'll deliver different system functionalities. You'll identify what things need to be bought and 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 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 four basic patterns of architecture interoperability:

- » Capturing the problem and the solution
- » Engaging others
- » Leveraging knowledge
- » Closed-loop verification and validation

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

The vision of a digital thread that loops in all domains opens the door to the knowledge you need for your solution. 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 designs that include how systems interact with each other (signals, power, hydraulic pressure, control laws, etc.) 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 have quite 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 risks more specifically. There may always be some unknown risks, but not nearly as many.

Just as important, once you discover a risk, you can capture it in the digital thread 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, between functions and geometry, and between 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 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. They're getting to market faster, at reduced costs, with fewer schedule glitches. They've cut risk, too. They're getting through testing and manufacturing with fewer failures.

In short, they're notching more wins. That's how the MBSE process helps companies be more effective and more competitive.

Implementing the Future State

If you want to implement a future state, the first step is admitting you have a problem.



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.

That said, the connection seems like a logical place to address at the outset. The future state starts with the digital thread because the data management piece is undoubtedly huge. It's a massive matrix that you're automating, bringing together the best tools for modeling and simulation and design optimization, orchestrating them with a strong product lifecycle management platform.



TIP

Perhaps a more logical place to begin, though, is tackling organizational culture. You need to sell MBSE company-wide by focusing on its myriad benefits.

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.

Consider, for example, that science fiction promised us decades ago that one day we'd have a robot to do all of the mundane tasks around the house, from cleaning and cooking to maintenance. That still seems a bit farfetched. But you might own a robot that cleans the floors, and while it's pretty cool, we accept it and take it for granted now. That's an example of how you take a big change and break it into a manageable, incremental piece.



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 product lifecycle management system and create new project parameters. Show how the pain is eased, prove the value, then move on to another pain point.

Aerospace engineers are clearly very smart individuals. Come 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 Expertise

Many companies are hungry 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.

Maximizing the benefits of MBSE requires transformation both in technologies and in the mindset of the users. The good news is no one has to go it alone as they continue on this path.



TIP

Success depends upon choosing the right technology and vendor to help implement MBSE. It's far more than buying a piece of software — the vendor must bring a robust suite of technology and the expertise needed to achieve successful adoption.

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

- » **Product lifecycle management:** MBSE must fully integrate into a PLM solution. It has to bring together multi-domain product development, including mechanical, electrical, and software domains. And it must fully address such considerations as configuration, baseline, workflow, cost, reliability, and manufacturability. The PLM solution must define what will be built, instruct players how to do it, and orchestrate 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 product lifecycle, allowing complete visibility into design decisions and integrates 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 the entire team is working consistently.
- » **Technical risk management:** Look for a product that includes 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.

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 portfolio is one such example. 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 a personalized and adaptable approach to helping companies become digital enterprises. It may not have everything listed above, but its comprehensive nature and approach to things is definitely worthy of consideration.



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 MBSE environment.

Ultimately, you need a software vendor with deep engineering expertise, the technical know-how required to develop mission-critical systems and is ready to help you with the paradigm shift that MBSE demands.

- » Bringing system modeling to the forefront
- » Making collaboration more feasible
- » Gaining solid buy-in

Chapter 6

Ten 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. As with just about everything else in the world, there are pressures to deliver more innovation, deliver it faster, and deliver it at a lower cost.

The model-based systems engineering (MBSE) approach promises gains in all of those areas, as this book has outlined. Following are just some ways your organization can win by adopting MBSE, and some of the lessons to remember as you move forward in your digital journey.

Transform the Process

To be clear on what's needed — transform, transform, and transform again. Adopting the MBSE approach certainly requires bringing on the right product lifecycle management system that builds the digital thread and 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. MBSE is not an app. It's a transformational mindset that must

spread throughout the entire organization. You won't win if you don't transform.

Remain Welcoming

As stated previously, 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 and adaptable to processes that already exist or that key domains are hoping to put in place.

Redefine System Modeling

As you implement 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 Collaboration

Some processes take time. Just ask any maker of fine wine or whiskey — you can't just bring in more people to make faster work of fermentation. But 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, brewing up their own subprocesses without regard to the big picture.

Digitalized MBSE removes the silo boundaries so 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.

Keeping 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.

Tracing Your Way to Success

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, activity, and all the interactions that happen along the way. Having a traceable process 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. Some have referred to MBSE as “CAD for systems engineering.” It is, perhaps, an oversimplification, but it seems to resonate well and make sense.

On the other hand, many early adopters of SysML believe that SysML is MBSE. But the reality is that SysML is merely an enabler of the overarching MBSE approach. Once you begin to make these distinctions, you’ll be well on your way.

Circle the Wagons

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, mechanical/physics, and electronics/hardware.

Tame Your Information

The biggest challenge for companies today is no longer within the engineering domain or level of modelers. The real challenge is the consolidation of all of the gigabytes of information coming in from all over into a single product architecture to view customer feedback, product requirements, and the latest from engineering, to name a few. The product architecture also includes areas in which companies have little means of managing the integration of electrical, electronic, and software with the functions to ensure the safety of the final product.

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, MBSE significantly reduces the need to do physical testing. Your MBSE program serves as that tight handshake between the virtual and physical worlds.



SIEMENS DIGITAL INDUSTRIES SOFTWARE

Accelerate your product development

Orchestrate your technical program and become more agile
and efficient with **Model Based Systems Engineering**.

www.siemens.com/plm/mbse

SIEMENS

Get to know model-based 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. This book walks you through the process of creating the latest and greatest A&D products using MBSE. It explores the product demands that have brought us to the current place where dramatic technological innovation is required and expected to coexist with faster, shorter, and more cost-efficient cycle times.

Inside...

- Help your team collaborate effectively
- Leverage the digital twin
- Mitigate risk
- Prioritize safety
- Integrate processes more easily
- Make complexity your secret weapon

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.

Cover image created by Oweb www.oweb.es

Go to **Dummies.com™**
for videos, step-by-step photos,
how-to articles, or to shop!

ISBN: 978-1-119-79316-8
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.