Digital Engineering for Electrical and Electronic Design eBook – Zuken USA

Introduction

Today’s products are growing in complexity and driving change in how they are defined, designed and manufactured.

The composition and complexity of today's products has irrevocably changed. Engineers are including an ever-increasing amount of electronics and software to satisfy the demand for more features, safety and connectivity. In many cases, achieving this level of new functionality necessitates better requirements, analysis, architecture optimization and continuous verification.

As a result, today’s electrical and electronic product content is dramatically more complex than in the past. Many of today’s products are comprised of numerous embedded subsystems communicating with an array of on-board sensors, cameras, motors and other subsystems. Electrical subsystems must deliver power to electronic endpoints and provide enough bandwidth for high volume communication. This array of embedded subsystems and components connected by wires must work collaboratively to achieve safety and performance targets.

Model-Based Design

All these changes introduce significant new challenges into the development process.

Mechanical, electrical, electronic, and software engineering teams have their own unique development processes and procedures. While the development within each engineering discipline may proceed well, getting designs to work across domains is often difficult. And make no mistake—interfacing and integrating designs across domains is at the root of the growing complexity in product development today.

Due to this reality, many companies see development progress in each domain, only to come to a screeching halt during system integration and verification. Engineering then becomes caught in a repetitive loop of a testing failure, root cause analysis, design modification, and testing again. Such iterative efforts cause significant delays and spiraling costs which can ultimately derail a project.

In response to this growing complexity, some organizations are exploring and deploying foundational changes to their development processes. The high-level goal behind these efforts is to get—and keep—all engineering domains on the same page throughout the development process. Such an initiative, called Digital Engineering, encompasses the following key characteristics:

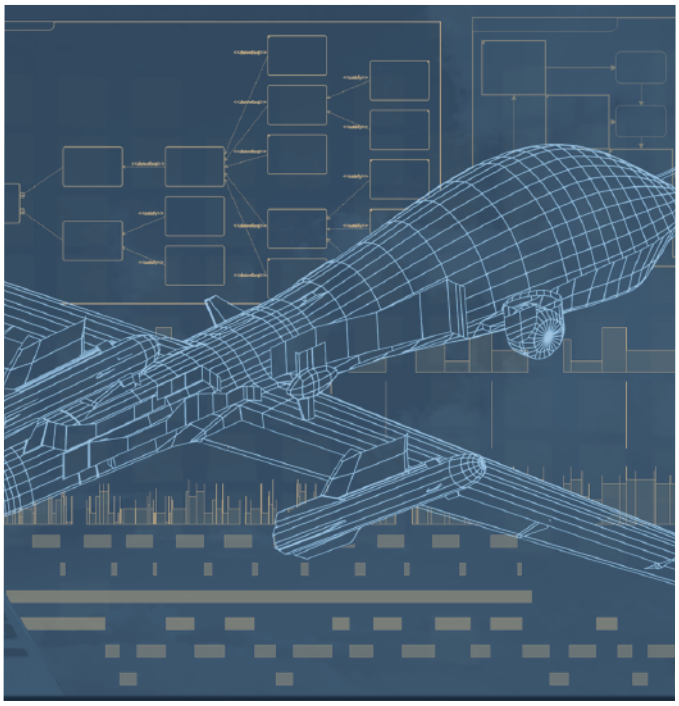
- Complete and unambiguous definition of the product or mission, using model-based methods, across engineering domains. The definition is comprised of requirements, functions, logical and physical architectures. The model is the single source of truth.
- Model-based design processes that require an analyze and build approach versus a traditional design, build and test approach.
- Requirement verification events throughout the design process, ensuring the implementation is meeting the product or mission definition captured in the model.

Adopting such Digital Engineering practices reduces the churn throughout development and eases system integration concerns. Moreover, these changes provide additional advantages, including:

- Faster exploration of architecture alternatives while maintaining requirement compliance.
- Better decision making based on more rigorous product and behavior definition in the model.
- Avoidance of development disruptions.
- Traceability of requirements and verification from end to end in the process.

Implementing a Digital Engineering process delivers numerous benefits, including:

- Hitting deadlines in the development schedule more frequently.
- Designing and delivering more innovative products.
- Reducing development costs by identifying and eliminating design errors early in the process.



Return on Investment in a Model-based Process

An ongoing challenge to model-based systems engineering (MBSE) is the ROI at the project level. Clearly, systems engineering embraces the benefits, but if the model does not remain relevant throughout the product lifecycle, the ROI is significantly limited.

In many cases the benefits of a model-based design process are limited because the model remains in Systems Engineering and is manually transferred to the design process. Showing ROI in this use case is difficult and does not provide senior management with the reason for broad adoption.

To make the model relevant through the product lifecycle, there are several requirements to meet:

- Programmatic connection to the architecture and/or detailed design. Manually translating the model to design disconnects the model.
- Exposing the design envelope to the design team. The envelope can consist of, but not be limited to, the architecture, requirements and related documents. The design team needs to view the model through a lens that meets the needs of implementation.
- Exposing the design verification requirements to the design and test teams to build out their test plans and verify the requirement has been met. Requirements must be verified throughout the design process (e.g., architecture, design, first article, system integration).

The relevance of the model is measured by its use during the lifecycle. Providing the proper lens for each purpose ensures model relevance. Model interaction throughout the design cycle provides the foundation for ROI calculations.

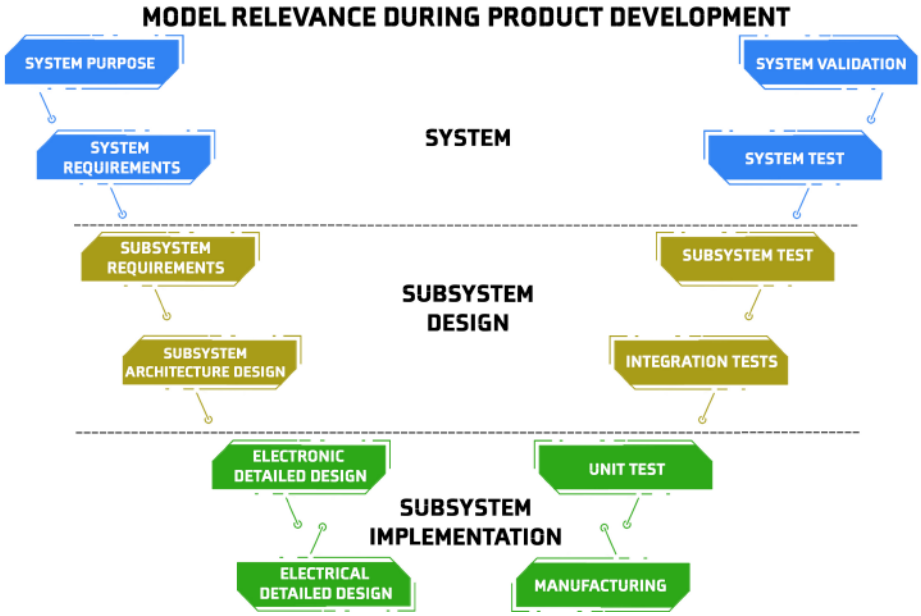
Building the E/E System Model

This document will not go into the details of MBSE practices or standards. Instead it will discuss constructing a model for the purpose of defining and realizing the electronic and electrical subsystems of a product. Some key design concepts to consider:

- The model is the single source of truth and must be part of the product lifecycle.
- The model is not a detailed design tool and should not contain schematic-like detail or part numbers, with few exceptions.
- The model structure is not changed by the design teams but can be supplemented with documents as test or simulation results.
- Parts of the model (e.g., requirements) need to be exposed to the design and test teams without requiring the launch of the MBSE tool.
- The verification requirement status in the model can be updated by the design or test team as they execute the verification requirements and add status.
- The model is hierarchical and is initially composed of functional units or subsystems. The subsystems are later decomposed into a physical architecture.

The model must be relevant throughout the design and manufacturing process for an effective Digital Engineering implementation (see Figure 1). Model ownership resides with Systems Engineering, but multiple teams must access and utilize the model.

Figure 1



Decomposition of the System Design

Utilizing a top-down approach, engineers first develop potential designs composed of logical subsystems as solutions that satisfy the requirements in the system purpose. Once the logical elements or subsystems are identified, the subsystems are decomposed into physical architectures (see Figure 2).

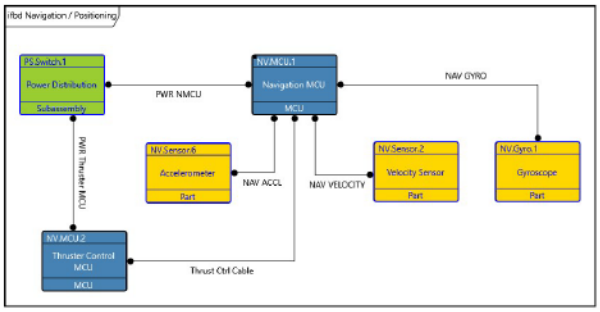


Figure 2

- Logical Architectures: This high-level functional block structure provides a means to allocate requirements across a system. For instance, a vehicle may have a power-train subsystem and a satellite may have a power subsystem. Logical architectures can be explored to act as alternative solutions for meeting system purpose. System-level requirements are allocated to the subsystems.

- Physical Architectures: The logical architecture is decomposed into physical entities that fulfill the subsystem requirements. The components can include an electronic control unit (ECU), sensor, actuator, batteries, etc. Components are connected through an interface relationship and links may be added to the interface if signals are to be assigned.

Digital Engineering Approach

The Digital Engineering approach to building and managing functional, logical, and physical architectures is through a single, unambiguous definition of a model-based approach. This extends the work of defining initial and implementation requirements as it is part of the same model. The advantages of this approach include:

- Functions, logical components, and physical items are design elements that can change and version individually. They are used to build behaviors, logical architectures, and physical architectures.

- Multiple stakeholders work against a singlemodel, so they are always on the same page. No one works against out-of-date information. But just as importantly, they can all work simultaneously, seeing each other's feedback, facilitating real-time collaboration.

- Engineers can conduct trade studies against requirements in the system purpose, comparing architectures against one another to determine the best fit.

With a model-based approach, developing a system definition is a natural extension of the system purpose. Traceability and visibility into the impact of a change is a natural outcome. All this helps companies realize the value of a Digital Engineering initiative.

Creating the Design Envelope

It is important to note that each component may be assigned functions and requirements, but it is unlikely that the design element has been identified (e.g., sensor part number is ABC-001).The assignments of requirements (e.g., weight, cost, memory size/type, dimensions) create the design envelope. The completeness of the design envelope is critical for success when the design team begins detailed design (see Figure 3).

Furthermore, configuration management across all these structures is crucial. Each of these artifacts can progress through change, manifested as versions, at different rates. Maintaining traceability during version changes is a core concept in the digital thread. Model history is a vital element to success.

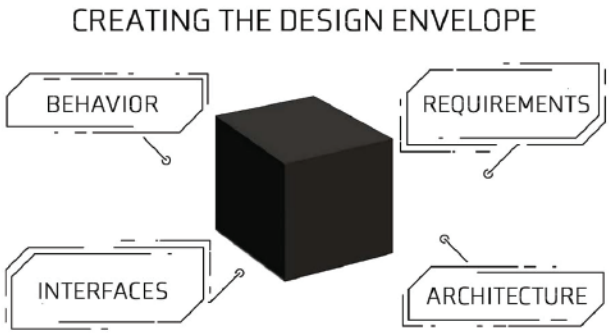


Figure 3

Analyzing System Designs

Next, systems engineers assess one or more system designs against the requirements in the system purpose. These assessments range from simple properties like weight and cost to highly complex calculations such as mean-time-to-failure and peak power draw. Comparing and contrasting many alternative architectures, and even running trade studies by varying design parameters to see how requirements satisfaction is affected, allows engineers to make highly informed decisions. Most importantly, it enables them to select one system design to proceed to the next step in the process.

Note that this activity isn't only valuable in selecting the right architecture, it also builds up an audit trail of the development process that becomes part of the digital thread. It captures which architecture was selected to proceed and why. Furthermore, it records the validation of the work to this point as part of the audit trail in the digital thread.

Digital Engineering Approach

The Digital Engineering means of assessing systems design is a 1D systems simulation. These analyses work like a block diagram, passing the output of one block to another block as an input. Each block processes inputs through sets of equations that represent the functional behavior of that part of the system. When assembled and run, the 1D systems simulation presents the behaviors and performance of the system. Engineers can check the satisfaction of requirements from the system purpose for multiple architectures. These analysis models can provide a more accurate view of performance early and can progressively improve fidelity as development continues.

Tracking and managing the configuration of the architectures during these assessments is also key. Without such context, engineers would not know which properties or analysis results apply to which architectures. The results would be meaningless. No one would know to which architecture they apply.

Realization of the System Model

The system purpose and E/E system design has been developed in a top-down approach through decomposition levels. The engineering teams responsible for the implementation of each subsystem have been assigned and it is time verify that their proposed designs are feasible in a bottom-up approach.

Exposing and Transferring the Model

Up to this point, the product or system model has resided in the hands of Systems Engineering. Implementing the model will require the subsystem design teams to access the model to understand the design envelope they have been assigned. This transition from model creation to model realization is one of the biggest obstacles to MBSE adoption. Manually transferring the model to the design tools takes time, is error-prone and is a snapshot in time. Ideally, the architecture should programmatically transfer from the model to the design tools for wire harness and electronic subsystem design.

The requirements assigned to a specific design element should be viewable by the design team in the context of their design tool. It is important to note that data flow is unidirectional from the model to the design tools. Any requested or necessary changes in architecture or requirements must be made and approved by Systems Engineering. Systems Engineering will update the model and the new version will propagate to the design tools as an incremental update.

Architecture Verification

Product complexity is driving the need for architectural verification before detailed design begins. The requirements and architectures form the system definition and represent a design envelope for subsystem engineers. Once development starts, those engineers must comply with the constraints of the envelope while satisfying all requirements. But what happens if it is infeasible to develop the subsystem within the envelope? For instance, a weight or power consumption requirement cannot be met.

Without this verification gate, sometime later, deep in detailed design, engineers might finally realize that there is no possible solution. Such a discovery late in development is highly disruptive, potentially threatening the entire process.

The purpose of this bottom-up verification prior to detailed design is to ensure that the development of the subsystem is feasible given the envelope. Designers can identify an over constraint and take the necessary step for resolution.

For instance, a systems engineer can change weight or power allocation between subsystems to meet a subsystem need while maintaining system level design integrity. This allows for design changes when the system design is still flexible, instead of later in design when everything is more constrained. This bottom-up verification aligns all stakeholders involved on the system purpose and system definition.

This period of development, when many subsystem engineering teams are verifying the feasibility of the system architecture and requirements, is chaotic. It is during this phase when the formal management of the digital thread is most needed. Stakeholders must know what feedback applies to which versions of the requirements, functions, logical components, physical items, and architectures. Digital conversations are required across multi-discipline teams for review, discussion, negotiation, modifications and ultimately approval. As these conversations take place, it is key for all stakeholders to stay up to date. Clearly managing all this change is imperative to the success of a Digital Engineering-based development process.

Digital Engineering Approach

The Digital Engineering approach is to transfer the model content to the subsystem design teams to mock up a variety of design alternatives as a solution within the design envelope (Figure 4: Subsystem Decomposition). This approach allows for quick and easy changes and exploration while checking requirements, constraints, and performance. A model-based approach acts as the lynchpin for all collaboration amongst stakeholders in this process. Any model changes must be approved and implemented by Systems Engineering. This approach enables close tracking of all changes and verification steps. It is also important to note that the model remains relevant throughout the development process.

All these activities must be captured as part of the digital thread. As the model is evolving, the digital conversations behind the change are as important as the change itself. Other solutions may be necessary to supplement the model to accommodate these conversations.

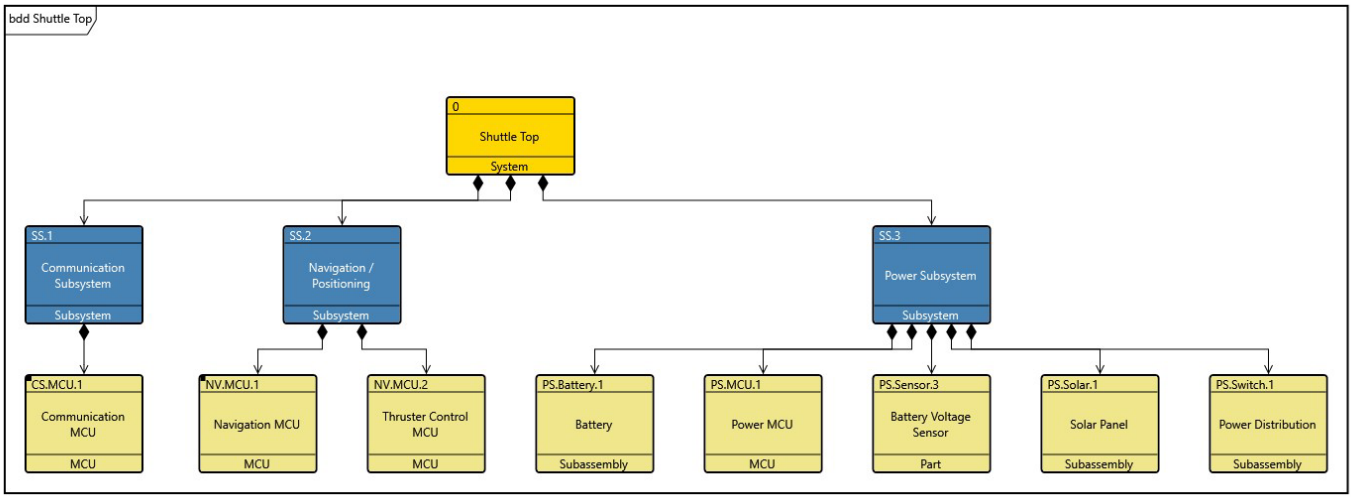
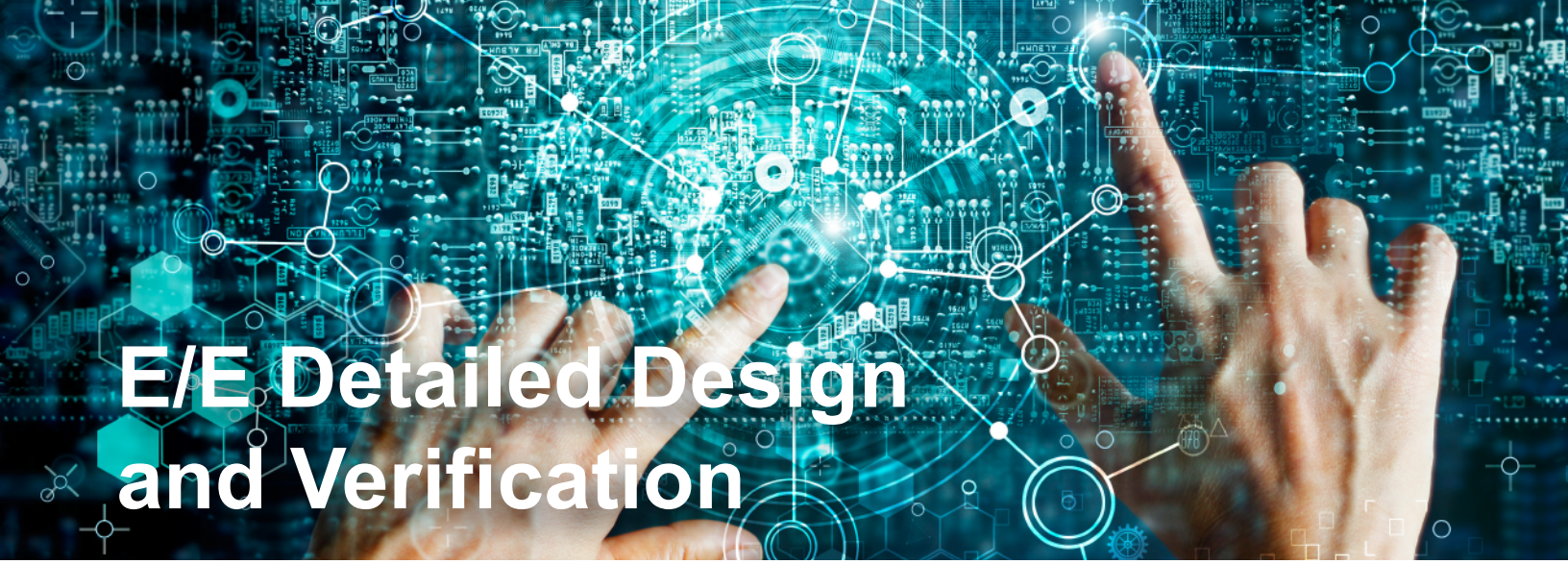


Figure 4: Subsystem Decomposition



E/E Detailed Design and Verification

Once the architecture has been verified, the next step of a Digital Engineering-based development process is to develop detailed designs for use in verification and validation as well as manufacturing.

At this point, the envelope for each subsystem has been verified both from a top-down and bottom-up perspective. Engineers now have subsystem requirements and architecture split out across mechanical, electrical, electronic, and embedded software domains. The design process now moves into realizing a manufacturable implementation of the requirements. The design process must be digital, 3D and cross-functional. For instance, the ECU in a vehicle will involve one or more PCBs in a mechanical enclosure connected through wire harnesses. Collision checks must be done in 3D to ensure fit during manufacturing. Digital design representations must be easily shared and synchronized to maximize design collaboration and accuracy.

As the detailed design of the subsystem progresses, there will be numerous design reviews on Bill of Materials (BOM), power, signal integrity, functionality, etc. The design reviews will consist of both verbal and digital conversations of cross-functional teams ensuring the design meets the system defined requirements plus design quality and safety thresholds. These conversations, just as the ones described earlier, are part of the digital thread to provide history, traceability and status.

Digital Engineering Approach

The Digital Engineering approach is to develop and analyze detailed designs as part of the digital thread.

- The definitions and simulations of the detail designs are in a high-fidelity, native 3D models across electrical, electronic, and mechanical engineering. These shared detailed definitions enable visualization and co-design between domains.
- Requirements and architectures from the system definition natively extend into these models. Here, they can be resolved with specific aspects of the detailed design.

This approach naturally extends the digital thread defined at the very beginning of the process into the lowest-level details of the design. This approach fulfills the need for traceability.

Additionally, shared detailed definitions based on high-fidelity, native 3D models across domains acts as a verification mechanism. Board layouts are always in agreement with 3D assembly models, presenting both electronic and mechanical engineers with a single, unambiguous definition of a board system. Electrical diagrams and harness routings through 3D assembly models are two representations of the same netlist, presenting a single definition in two different representations to both electrical and mechanical engineers.

System Model Verification

A Digital Engineering-based development process verifies that specific requirements are met at multiple design phases. The multiple verification gates throughout the design process ensure that design teams don't stray from the design envelope. It is possible to meet a BOM or power requirement at the architecture verification gate and miss it at the first article gate. Design integrity is maintained, and the probability of success increases. Confidence grows that subsystems will seamlessly integrate up to higher-level system builds.

This series of events, verifying integration, performance, and requirements satisfaction for subsystems and systems, is recorded as part of the audit trail in the model. The conversations are stored as part of the digital thread, but the verification requirement status defined in the model is updated to reflect the success or failure of that requirement. Test teams have specific verification requirements to meet that will drive test plan development.

A requirement verification analysis can be run at each verification event to provide critical data on the alignment between the model and the realization of the model. For instance, the architecture verification event is scheduled to be complete in two weeks and only 50% of the requirements have been met and 25% could not be met.

Digital Engineering Approach

The Digital Engineering approach defines verification events in the model that include verification requirements such as integration, performance, function and requirement satisfaction.

- Updated, more detailed 1D system simulations that leverage reduced order models of high-fidelity 3D analyses.
- Hybrid digital-physical simulation and test configurations of systems, including Model-in-the-Loop, Software-in-the-Loop, Processor-in-the-Loop, and Hardware-in-the-Loop methods.
- Traditional physical prototyping and testing.

While these methods are beneficial, they alone are not enough. It is imperative to capture these activities and results as part of the digital thread. This means capturing the configuration of every test, whether it is digital, physical, or some mix of the two. This means documenting the results of each test and relating them back to the test configuration. This extends the digital thread to its end in development at design release.

Summary and Recommendations

Today, products are undergoing a dramatic transformation. They include far more electronics and software than ever before. Just as importantly, integrating all these technologies into a single, integrated system presents significant challenges to engineering. In response, many are fundamentally changing their development processes, adopting a new initiative called Digital Engineering. This new effort includes:

- Engineers using a single, unambiguous product or system definition from a model-based approach, enabling traceability throughout the design of the system. It is important that the model remains relevant throughout the product lifecycle.
- Engineers develop system designs that include functional architectures, logical architectures, and physical architectures as part of the same model-based approach. Configuration management of changes to this model becomes key as it grows and evolves as part of the digital thread.
- Engineers assess performance using 1D simulations connected to the model-based definition of the system. This effort represents the first closed-loop verification of the initial requirements.
- The model must transfer programmatically from the tool used to create the model to the tools used to implement the model. Access to the model must exist throughout the product lifecycle.
- Engineers begin to realize the system design by verifying the design envelope from a bottoms-up approach at the architecture level with all the discipline-specific design teams. Detailed design begins after the physical architecture is verified against the model.
- Engineers verify cross-systems integration and the satisfaction of requirements using the model-based approach. Verification requirements provide the foundation for test teams to develop test plans. The status of each verification requirement is updated in the model.
- Model verification throughout the development process insures the final product is a realization of the system model.