Siemens PLM Software

Systems-driven cross-industry product development

White Paper
Development of complex products through a systems-driven process

A plan to succeed
Siemens PLM Software believes that today's complex products require a systems-driven approach to product development that combines systems engineering with an integrated product definition and the ability to unify your product development framework with manufacturing and shop floor operations. To facilitate systems-driven product development, Siemens PLM Software provides systems engineering tools, functional networking, a consistent process-enabled framework, advanced modeling and simulation capabilities, an intuitive user experience and an open product lifecycle management (PLM) environment.
Contents

Executive summary .......................................................... 3
Increasing product complexity ........................................... 4
Requirements engineering and management ................. 5
Functional and behavioral networks ............................... 6
Consistent process framework ...................................... 11
Modeling and simulation ............................................. 14
Immersive user experience ......................................... 16
Conclusion ....................................................................... 17
Executive summary

Twenty-first century engineering enterprises face new challenges as they strive to create the best products or systems in increasingly shorter time-frames. For many industries, each new generation of products is inevitably more sophisticated and complex because new product versions require the participation of more engineering disciplines than ever before. Many industries have evolved from the delivery of purely mechanical devices to the development of sophisticated software-driven electronic products that can modify their own functions and performance in real time.

This continued introduction of electrical, electronic and software components into the product development process has created the need for more efficient and effective integration of all participating engineering disciplines. Previously, much of this cross-domain knowledge was held in the heads of individual engineers, but that approach is no longer possible. Today's product complexity, combined with a shortened time-to-market, translates into a product development and manufacturing complexity that often spans the entire supply chain. It is imperative that the entire supply chain understand the impact that individual decisions have on other aspects of product design or manufacturing processes in a timely manner.

Unfortunately, taking extra time to get it right can cause your resource costs to increase significantly and extend your time-to-market. On the other hand, if you rush your process too much, you may end up not meeting system requirements, which leads to increased warranty cost or noncompliance penalties. Trying to solve this dilemma by reducing content complexity doesn't work either, because eliminating new features means removing the innovations that your consumers demand.

Simply put, traditional product development and manufacturing processes no longer work; they can't manage all the process complexities and dependencies, especially when your supply chain is globally dispersed.

All of these factors combine to create a new imperative for product manufacturers. The need to manage these sophisticated products requires a systems-driven process that keeps all disciplines and suppliers connected and informed. Siemens PLM Software's systems-driven approach to product development marries systems engineering with an integrated product definition as well as your manufacturing processes and assembly plants.

Siemens PLM Software’s approach to systems-driven product development is enabled by the following key capabilities.

- **Requirements engineering and management** – a single, secure source for managing the evolution of requirements and linking them to physical implementation and testing across multiple sub-systems, domains and functional areas, enabling you to verify compliance and more effectively meet product targets and schedules.

- **Functional and behavioral definition** – a series of networks that facilitate the functional definition and behavioral analysis of the entire system or product, including requirements allocated to the various functions and the development of the logical constructs of the product and into their physical implementation.

- **Consistent process framework** – a framework that crosses all of your system development and manufacturing domains and integrates your evolving product definition with change and issue management, configuration management and schedule management – thereby unifying your mechanical, electrical, software and electronic development domains with your process planning and plant operations.

- **Advanced modeling and simulation** – the ability to derive engineering requirements from user needs and validate that your engineering specifications fulfill these needs early in the system or product development process.

- **Open PLM environment** – a PLM-driven infrastructure in which your home-grown and commercial software tools can be easily integrated and efficiently leveraged.

- **Intuitive user experience** – the ability to enable all product development, manufacturing, delivery and maintenance stakeholders to access all of the information they need to intuitively understand your product as a complete system, as well as to help these users proactively leverage this information in the context of their specific needs and job tasks.

This white paper describes how these capabilities can be leveraged to provide all stakeholders with a robust systems-driven product development environment.
Increasing product complexity

Each new generation of products is more sophisticated and complex than the previous generation:

- Cameras capture, store, enhance and share images
- Home appliances such as washers/dryers are internet hot-spots and use electronic brains to control washing/drying functions
- Aerospace systems rely on advanced composite materials and sophisticated fly-by-wire technologies
- Infotainment and safety systems make automobiles “electronic processors on wheels”
- Medical devices, robotics and imaging technologies must “do no harm”
- Robotics and machinery achieve amazing levels of precision or perform in hostile environments

The rapid increase of software-driven electronics, introduction of new materials and adoption of sophisticated manufacturing processes combined with the increasing emphasis of regulatory and environmental compliance demands big-picture thinking; systems-level thinking across the product lifecycle. This increased complexity creates the need for more efficient and effective integration and communication across all engineering disciplines and functional domains.

Leading companies are leveraging systems engineering concepts and approaches to create a systems-level view of the product and subsystems interactions, along with ensuring that mechanical, electrical controls, electronics and software implementation are kept in sync and therefore contribute to a compliant, reliable, high-quality product. Correctly implemented, this can play a significant role in mitigating risk, accelerating time-to-market, reducing engineering change requests during development, and significantly lowering warranty costs.
Requirements engineering and management

Requirements drive the product lifecycle

Addressing today’s increasing product complexity requires a fully integrated approach to systems engineering and requirements management. It requires a solution in which customer and contractual requirements are captured in a single source, shared with all stakeholders and managed so as to efficiently and effectively drive the product development and delivery process. Requirements must be supplemented with quantifiable constraints in terms of their cost and delivery schedules, as well as their ability to satisfy established performance, ergonomic, safety, usability, reliability, maintainability, recycling/disposal and other compliance-related metrics. Managed properly, requirements engineering is an activity that spans the entire product lifecycle, to ensure that product development, realization and service remain aligned with original goals.

To accomplish this, you must allocate or link requirements to the various domain’s (mechanical, electrical, electronic and software) design and analysis functions, as well as across the other functional areas of the company including testing, manufacturing, service and maintenance. For example, an aerospace manufacturer may have a requirement to meet certain crash and safety standards, an automobile manufacturer may have requirements to meet certain performance and fuel efficiency objectives, or a consumer electronics manufacturer must meet Restriction on the Use of Hazardous Substances (RoHS) and conflict minerals.

To ensure compliance, a critical process in any program, product or project is the verification that requirements have been met. To ensure quality and on-time delivery, you must associate requirements to tasks and project schedules, to the appropriate product configuration, as well as to issue and change management activities.

Teamcenter provides an intelligently integrated solution for capturing and communicating requirements throughout the enterprise. Providing a single, secure source for all requirements, Teamcenter supports “live” integrations with

“ATK develops and manufactures armament, mission and space systems, with the goal to ensure that their customers accomplish their objectives – whether they involve a military operation, a satellite launch or a technological breakthrough. The systems engineering and requirements management capabilities of Teamcenter® software help ATK meet its goals. “We can go all the way from a customer’s set of requirements, their statement of work, to the tasks in a schedule and we can trace all the way back to that statement of work and know why we’re doing that task. We’re doing that task to comply with this portion of the ATK’s PLM strategy starts at the onset of a project.”

Jon Jarrett
Senior Manager
Engineering Systems

Microsoft Office applications. You can review, edit and publish the requirements as well as send approval notifications via Outlook. You can segment functional requirements by domains or functional areas and distribute them to all the decision-makers in the product lifecycle.

Embedded requirements traceability reports give you the ability to see all the relationships between different objects and to validate compliance to the requirements. And lastly, because they are Teamcenter objects, requirements can participate in schedule, resource, configuration, workflow and change processes. Using Teamcenter to manage and trace requirements across the entire system or product lifecycle provides an intelligently integrated decision support system that enables you to reduce costs, improve quality and ensure compliance.
Functional and behavioral networks

Functions drive the product definition
Today’s products can be seen as an integrated set of functions. For example:

- A camera has to meter the light source, combine or modify images, and output multiple formats
- A home appliance must wash, dry, optimize energy usage and access the internet
- An airplane must take off, maintain safe flight, communicate with ground systems and land
- A car has to accelerate, stop and provide a safe and comfortable passenger environment

These are just a few examples of the high-level functions that products must provide. They can be broken down – decomposed – into lower levels of more specific, detailed functions that eventually are delivered or implemented by physical or software components. Most of the functions of today’s products are the same or similar from one generation, version or configuration to the next, but with the advent of advanced software-driven electronics and new materials, some of that is now changing. Performing functional analysis may be important for your new, ground-breaking solution or as you integrate new functions into existing products.

What makes each product unique is the set of requirements that are applied to these functions. What is the resolution of the image? What is the energy efficiency optimization function? How comfortable and safe does the pilot or passenger compartment have to be? How fast does the car have to accelerate? The answers to all of these questions can be found in the product’s requirements.

Requirements are allocated (related) to specific functions so that product teams understand what criteria to use when designing specific functions. And of course, requirements and functions have dependencies between each other that need to be understood so that proper trade-off decisions can be made. For example, if an airplane is required to have a greater transport capacity, then that increased capacity will have an impact on the aerodynamics, thrust and other functions. If a car is required to have a top speed of 175 mph, then that speed will have an impact on the function of stopping the car, and soon. The dependencies between the requirements and functions in both examples have to be understood and their impacts managed.

Basically all of these impacts and dependencies must be modeled to be understood so that as design or manufacturing decisions are made, the relationships and dependencies across the entire lifecycle of the products can be intelligently evaluated – and the best possible solutions developed.

The way functions will be implemented is defined in a logical architecture. Logical models can, and often do, exist at each level of the functional decomposition – and they define how the different functions interact with each other. In electrical terms, this would be expressed in a schematic diagram that indicates what inputs and outputs on one device or component are connected to specific inputs and outputs on another device or component.

Siemens PLM Software’s systems-driven product development solution provides a complete set of capabilities to put all of this together – requirements, functional models and logical models, ensuring all relationships and dependencies are defined and understood throughout the entire product.

Optimize re-use and manage evolving technologies
Most new products rarely start from scratch. For example, multiple versions or variants of an aircraft are based on the original design. The iPad is a logical extension of the iPod and iPhone. Sixty to eighty percent of a vehicle is functionally the same from one vehicle to the next. Logically, the way the functions interact is fairly similar, as well.

Requirements can come from many sources, including customers, regulatory bodies, government, internal development standards and manufacturing best practices. Product teams collect these requirements and manage them as individual objects. This object approach allows requirements to be related to one or multiple functions that they impact, even when these functions cross different products, programs, systems and design domains. Requirements can also determine variation in the functional and logical models because they might require new features or different technologies.
Systems engineering plays a crucial role in facilitating an effective requirements process. Because requirements originate from many different sources, they need to be evaluated, refined and ultimately focused on a particular system, product platform or program. As the product team moves from a very coarse definition of the system at the concept level to a fine-grained or detailed representation at the component level, the product’s requirements, functions and logical models are also refined, gaining more detail and granularity as they move through each of the design and manufacturing processes.

**Validate early in the development process**

Because interfaces are where all the bad things in a system or product seem to happen, one way to reduce problems are the creation, management and validation of a well defined interface. Functional networks and the interactions between product functions describe the interfaces and constraints that will eventually be implemented by various functional organizations and design domains (mechanical, electrical, software, manufacturing process, risk, reliability, cost, service and disposal).

This network includes descriptions and agreement on the definition of each interface, in conjunction with the exchanged (subscriber and publisher) inputs and outputs – as well as the functional definitions, and their related performance data – required to validate the behavior of each function. In our washing machine example, consider the washing machine transmitting an opposing frequency to cancel the agitation cycle noise. Or in the aircraft example, the window film automatically darkening when the passenger’s infotainment system is activated. Or in the vehicle example, a sunroof that automatically closes when it starts to rain. These capabilities are high-level functions allocated to the components necessary to carry out that function – the sensors, ECU, electrostatic film, motor, mechanical parts, wire harness and other equipment. The functional description, in conjunction with the logical definition, is the basis for creating behavioral models that simulate the function as well as its interdependencies to other functions.

**One model across all variants – the 150 percent view**

The result of this process is a complete functional model of an entire system (also known as the product platform.) This “150 percent model” covers all possible product configurations – including combinations that may not be actually produced. Because this model describes the logical interaction of the product at different granularity levels (such as the system and subsystem level), it can play a crucial role in helping the product team find conflicts that escaped notice and might cause extra expense, late redesign, field recalls or even catastrophic failures.

Many different models are required to refine functions and gain important insights, including signal analysis, energy transfer diagram, state diagrams, failure mode effects analysis (FMEAs), electromagnetic interference/electromagnetic compatibility (EMI/EMC), network diagrams, fault trees, behavior and manufacturing processes, to name just a few. The end result is a specification for the product’s physical elements.

---

“Strukton Civiel develops, implements and manages large-scale infrastructure projects from the initial planning phase up to and including final operation. To be able to successfully carry out design and construction projects and a public-private partnership (PPP), you need continuous insight into everything that influences budgets, technology and planning. The aim of their systems engineering project was to improve risk analysis and to provide the organization with more and better information so that project complexity can be managed more effectively. “What we have done with the Teamcenter systems engineering project is rise to a higher level so that we get a genuine picture of all the systems, tasks, responsibilities, schedules and risks,”

Frank Hoekemeijer
Manager
Strategy and Development Department
Managing complexity and tracking against high-level goals – integration architecture

Product or system alternatives are evaluated in terms of their ability to achieve goals that, ideally, are formulated as measurable targets – such as weight, cost, power, fuel economy, flight time or safety. In system-driven product development, these high-level targets are decomposed ("engineered") into targets for subsystems or specific product properties – such as current draw, RPM, lift, engine parameters, transmission parameters, mass and aero drag. This translation is refined from high-level parameters for subsystems to increasingly detailed parameters of the subsystem components based on heuristic or quantitative models that are continuously refined to describe these dependencies as accurately as possible.

This refinement of high-level targets to lower-level targets creates a structure that ultimately enables individual engineers to understand how changes made to a component or subsystem impact the goals of the overall program. This structure also allows the program manager to track if the program is “on target” for its high-level goals. If it is not, the structure enables the program manager to determine what causes the deviation and how it can be mitigated.

![Diagram of Vehicle System Integration Architecture](image)

Figure 1. Subsystems for automotive production.
Configurable plant and process management – systems-driven process throughout the product lifecycle

The systems engineering process extends into manufacturing, including designing, validating and testing production processes and manufacturing equipment needed for a particular product. The translation of manufacturing requirements into the equipment and operations definition and driving validation through virtual commissioning is analogous to the systems engineering process for product design.

The process starts by identifying manufacturing requirements based on the product definition. In the example below, requirements are listed for the sheet metal used for a washing machine’s cabinet, the aircraft’s airframe or vehicle’s engine compartment. From the product definition (which is mostly a combination of 3D CAD and PMI data), product teams can derive specific requirements for the production of the detailed product design. Once these requirements are understood, the product team might identify an existing plant in which the product will be produced or select the most applicable plant if additional equipment will be needed.

Once the equipment has been defined, the process continues with the definition of work orders, the equipment bill of materials (BOM) and layout, which then serve as a basis for developing and validating the required automation for a given production line or cell. At each step of this process, the product team will perform some form of validation based on associated product and production requirements. It is important to validate both the product and production requirements. For example, finishing tolerance of the parts is a product requirement that can only be validated during production definition.

Perkin Engines designs and manufactures high-performance diesel engines. Their “virtual factory systems” link the factory to product planning and development that enable their manufacturing engineers to model production facilities in detail for process optimization. These models enable them to create a virtual production line and manufacturing routing sequence. PLM systems support this “virtual factory” and allow multiple design teams to synchronize product development and validate factory-floor processes.

Visualizing the shop floor effectively links the factory into the product planning and development process, thus closing the loop between finished goods and product designers. Because every step of the manufacturing process is synchronized along the value chain, they have seen a threefold increase in process planning capacity and significant value in terms of financial return and time saved.

Describing the mechatronic system – physical architecture

For mechatronic systems like many of today’s products, the physical architecture contains all mechanical components and assemblies, the electrical architecture (including electronic control units (ECU’s), sensors, wire harnesses and corresponding schematics), signals and messages, as well as the complete software bill of materials, including bootloaders, application software, configuration files and calibration data. Logical and functional components are associated to the physical architecture. For example, a behavioral model for the function of a washing machine’s spin cycle, an aircraft’s landing gear or a vehicle’s electric sunroof are each associated with the corresponding elements in the physical architecture so that engineers can understand how a changed part or software component might require the underlying behavioral model to be updated and revalidated.

The integration architecture and the physical architecture organize the requirements, functional, logical and physical view of the product so that complexity at the lowest level is manageable even if mechanics, electronics and software are all involved. These factors are also highly re-usable from one system or program to the next and across multiple product platforms.

Perkin Engines designs and manufactures high-performance diesel engines. Their “virtual factory systems” link the factory to product planning and development that enable their manufacturing engineers to model production facilities in detail for process optimization. These models enable them to create a virtual production line and manufacturing routing sequence. PLM systems support this “virtual factory” and allow multiple design teams to synchronize product development and validate factory-floor processes.

Visualizing the shop floor effectively links the factory into the product planning and development process, thus closing the loop between finished goods and product designers. Because every step of the manufacturing process is synchronized along the value chain, they have seen a threefold increase in process planning capacity and significant value in terms of financial return and time saved.

Siemens PLM Software’s approach to systems-driven product development takes the physical aspect of all dimensions of the design into account – including parts, physical design, processes, plants and the actual product represented by a serial number. This comprehensive approach is known as the RFLP5 method.
Traceability and consistency – integration from development to manufacturing and service
Applying the functional network across the entire product lifecycle, including manufacturing and service, helps remove the “work and reconcile” tasks that occur in most product or system development programs. Today, gates or milestone reviews provide an opportunity to bring everyone together and reconcile the integration of all the pieces that must work together to deliver the function. Functional networks enable you to build-in an integration framework from the beginning. This enables you to align all of your functional organizations at any point in the development process, not just at certain milestones.

By leveraging the systems-driven product development process, anybody touching a physical design is able to understand which functions the design supports and the role each component plays in fulfilling a requirement. This is especially valuable in projects that span mechanical, electrical and software engineering, in which complete traceability is crucial for facilitating collaboration and knowledge re-use.

In addition, the physical design can also act as a template, bringing automation and knowledge re-use into the model. Template values can be derived from requirements or other relationships and dependencies. Rules can be applied through templates that ensure boundaries established by related requirements are automatically checked and flagged if an invalid condition is encountered.

Figure 2. Systems-driven design for production.
Figure 3. The complete physical definition of the product.
Consistent process framework

In order to make the systems-driven approach work consistently for every engineer, Siemens PLM Software’s systems-driven product development solution provides five key capabilities, including:

- The ability to connect all engineering data in a common data model, which facilitates a deep understanding of dependencies created in the functional and logical architecture
- Open PLM platform for interacting with specific authoring tools, which enables product teams to interface with a variety of authoring tools across multiple domains
- The ability to minimize variability across all domains, by establishing one common, consistent product configuration framework for all engineers
- Common program and platform management, which drives the PLM process in a single program context from program initiation to service
- Change and issue management across all domains, which ensures complete traceability throughout the PLM process

Connect all engineering data in a common object data model

Driving the development process through functions creates consistency between your mechanical, electronic, electrical and software processes. Each domain can use a different set of methods and tools, while sharing the product functions and leveraging a common definition that engineers can work with on a cross-domain basis.

However, the dependencies between engineering domains can go very deep into the actual design data. For example, an electrical designer will develop a schematic for a wire harness based on the function definition. This schematic contains all of the information about the components that are interconnected. Nevertheless, the engineer will have to make an assumption on the length of the connection. Later, another engineer will design the actual routing of the cables, determining the actual length. This information is important for the electrical engineer. With this in mind, the system-driven product development solution needs to provide a data model that represents the schematic and is able to link parameters in the schematic to parameters of the actual “mechanical” design of a wire harness.

Siemens PLM Software addresses these needs by providing an open, standards-based PLM solution with a single source of product and process knowledge and a common data object model that can be leveraged across multiple engineering domains. Openness means two things:

- Providing a design data model based on open standards wherever possible
- Supporting and driving integration with domain-specific design tools

Working with our PLM-enabled framework, individual development teams can retain their focus by continuing to use their mechanical, electrical, electronic or software design methods and tools of choice, while working together in-context to meet overall development goals. This approach enables them to reduce risk and achieve higher product-related quality goals, performance standards and safety requirements, which result in lower warranty cost, better customer satisfaction and shorter time-to-market.

Leverage an open platform to facilitate use of the “right” authoring tools

Rarely do entire supply chains, user communities or technology vendors start fresh. Established ecosystems of software tools and manufacturing processes already exist. It is not realistic to require your engineering and manufacturing domains to adopt new tools as you implement your system-driven product development solution. As a result, you will need an open PLM environment that enables you to consolidate a data model based on open, published standards.

Siemens PLM Software provides the following integrations to today’s most widely used authoring, design and diagnostics tools to ensure the consistency of your PLM-driven processes.

- **Mechanical lifecycle**, including support for major MCAD tools such as NX™ software, CATIA, Pro/ENGINEER, SolidWorks, Solid Edge® software and Inventor.
- **Electronics lifecycle**, including integrations with ECAD PCB design tools from Mentor, Cadence, Altium, Intercept and a gateway for other EDA tools. Teamcenter is used to manage multi-CAD part libraries, maintain links to vendor data, facilitate compliance management, support check-in/check-out and manage native design archives, fabrication and assembly data, derived files and extracted BOMs. Teamcenter also is used to help facilitate native ECAD data visualization, design for test and assembly (DFT/DFA) and BOM grading processes that enable usage of 3D visualization tools, automatically validate design against assembly rules and evaluate the quality of the BOM before being submitted to your contract manufacturer.
• **Software lifecycle**, including supporting source code, software design elements, configuration and calibration parameters, binaries and hardware as part of the same product structure and linking this product structure to requirements and their dependencies across multiple engineering domains.

Within each engineering domain, the cycle of design, validation and testing is executed based on the common PLM backbone. Domain-specific requirements are transformed into a detailed design and then validated against the requirements outlined for that domain.

**Integrate nominal and failure-mode behavior**
Most engineers automatically focus on the “working” (nominal) behavior of the product, and less on the “not working” behavior (failure modes). One of the reasons the latter is harder to do is that understanding, simulating and testing against failure modes is not possible without a systems perspective. Systems-driven product development directly supports failure mode and effects analysis (FMEA).

FMEA can provide an analytical approach when dealing with potential failure modes and their associated causes. When considering possible failures in a design – such as safety, cost, performance, quality and reliability – engineers can get a lot of information about how to alter the development/manufacturing process to avoid these failures. FMEA is an easy way to determine which risk raises the greatest concern, thereby identifying a potential problem before it arises.

**Manage variability across all domains**
Variability occurs everywhere in the development process: at the user requirements and functional network levels, within the logical architecture and across all of your engineering domains.

Robust cross-domain traceability has to be able to manage the variability in the interrelationships between requirements, functions and logical components, e.g., which versions of which requirements apply to which versions of which functions performed by which versions of which components, assembled in this factory for this customer. Many different variables can affect configuration including customer location, governmental regulations, customer features, environment variables and engineering and order release status. Configuration and multi-domain navigation must be able to combine different kinds of variables in the right context to find relevant results quickly.

For example, analysis of a washing machine simulation may show that during the high-speed spin cycle for extra large laundry loads the vibration dampener doesn’t function properly. Or, an aircraft flight simulation may detect window darkening failures due to harness vibration when wings are under excessive loads. Or, a car simulation may show that a sun roof does not close properly at 100 mph. To understand the results, engineers may want to view all the components in the product that specifically configure together with the failure item. But they may also need to look further, because some components which may affect the failed item’s operation (such as an oversize load vibration sensor, electrostatic windows, or roof support bar assembly) are always in the product when configuring engineering design options and may not show up with a specific configuration expression as direct as “extra_large_load = yes,” “window_darkening = yes,” or “sunroof = yes.”

Relevance can be determined by many different conditions involving a combined configuration of requirements, functions, logical components and designs, such as:

• Sharing a common variable expression such as “extra_large_load = yes,” “window_darkening = yes,” or “sunroof = yes.”

• Configuration rules which imply a common variable expression such as in the case of the washing machine “if ‘model=industrial_xyz’ then ‘extra_large_load = yes’, “ or the aircraft “if ‘model= commercial_xyz’ then ‘window_darkening = yes,’” or in the case of the vehicle “if ‘model= sport_xyz’ then ‘sunroof = yes.’”

“The growing prevalence of software and electronics in cars has huge implications for after-sales activities and controlling warranty costs at Ford Motor Company. These issues required a new way of testing, validating and managing the software content that goes into vehicles. Ford uses Teamcenter systems engineering functionality to solve the problem of communication between ECUs by monitoring and tracking software dependencies. Teamcenter helps identify where a particular software component is being used: in which vehicle programs, in which series, in which variants of those programs and in which global locations. The manufacturing data model tracks information for in-plant flashing, ensuring that correct software assemblies are flashed during manufacturing. This has several major benefits. “First, it enables Ford to perform impact studies whenever a software change is made. Another is that Ford can now trace the ECUs to an individual customer’s vehicle by the vehicle identification (VIN). Teamcenter allows us to fully re-use software components without any changes.”

Chris Davey  
Technical Leader  
Software and Control Systems Engineering
Overlap in related functions or requirements can be relevant to the same logical component configuration. For example, the car or aircraft’s functions for protecting the occupant during a crash might include a specific reference to a particular roof or bulkhead support component implemented by a specific design assembly re-used from a previous program without the sunroof or electrostatic windows. Multi-domain configuration combines configuration requests within domains (i.e., to find all requirements relevant to a particular program) and configuration for related data across domains.

To illustrate the concept of variant management, consider the following examples from our washing machine and car examples. First, as part of the washing machine’s vibration control, two options are available – mechanical control and electronic sensor control. Next, as part of the car’s sunroof, two options are available – manual control and control based on a rain sensor. As illustrated below, these product options can be mapped to an option package (e.g. laundry care or vehicle trim package) called basic and fully loaded. As users work their way through the systems-driven product development process, they will be able to consistently refer to the option values that are defined as part of the product.

**Leverage program and platform management**

Program management supports enterprise project planning and execution. Team performance is enhanced through real-time and goal-directed collaboration. This form of organized collaboration should be available to all layers of the organization, including your value network of suppliers and customers.

The integrated program management functionality in Siemens PLM Software’s systems-driven product development solution includes capabilities for managing programs and their individual projects, allowing organizations to establish best practices and support continuous improvement. Real-time interaction among groups and members from distributed locations on specific business objectives and critical deadlines is enabled.

Program management is more than just establishing schedules and managing tasks. It also facilitates product platform configurability. In many companies, only a very few team members understand and control the variant data in their business systems. CAD designers and part engineers are not expected to author the correct variability without guidance, although they must be aware of variability to support their design. Siemens PLM Software’s solution enables configuration experts to define-in variability using nomenclature that designers already understand. This use of the product architecture enables individual companies to use unique taxonomy to define their generic parts or modules. These generic parts are placeholders for design and can include parts that may not yet exist.

**Teradyne needed a technology solution to replace the largely manual and impromptu processes that were hindering engineering and manufacturing performance. Separate workflows and databases were consolidated into a single workflow and database that made part information consistent and easily accessible. All project requirements, parts, specifications, engineering changes, bills of materials and engineering documentation were centralized in and managed using Teamcenter.**

By automating formerly manual activities, streamlining the change management process and providing better access to more accurate information, the solution helped Teradyne cut cycle times for engineering change orders from 90 days to just 14 days and engineering change orders are also implemented more accurately. The rejection rate has declined from 70 percent to 26 percent. Together, the greater speed and accuracy of the change order process have reduced the costs of implementing change orders by 60 percent and contributed to reduced scrap and rework costs.

**Establish change and issue management across all domains**

A system-driven product development process must be able to manage the complexities of product variation by accurately representing the “whole product” bill of materials and common architecture breakdown for the product line – including software and electronics – for all product configurations. To address these needs, Siemens PLM Software’s solution provides:

- **Context management**, including unique capabilities to simplify design collaboration by defining, managing and sharing working contexts for product development

- **Change management**, including a common infrastructure to consistently manage change across multiple domains and different development stages

- **Lifecycle representations**, including the ability to associatively define and relate multiple BOM representations of the product structure for different lifecycle stages or processes

- **Option and variant capabilities**, including the ability to organize products into modules and marketing options, thereby facilitating faster response to market opportunities and increased product/part re-usability
Modeling and simulation

Early, virtual validation enables product teams to leverage models as much as possible instead of tying them to physical artifacts. Models can be used to represent component, subsystem, system and product-level behaviors. Models serve three important purposes within systems-driven product development by letting product teams:

• Convert marketplace and customer needs into engineering requirements
• Drive early validation
• Accelerate trade-off decisions during the early stages in product development

Behavioral models convert customer needs into engineering requirements
Models play a critical role in cascading customer needs down into engineering requirements. From earlier product examples, a customer-driven need might be to reduce vibration during washing machine spin cycles, in an aircraft during take-off, or while a vehicle’s engine is idling. The customer usually perceives this vibration in the floor, the seat and/or the steering wheel. In addition, perception to vibration can vary with vibration frequency.

How then can the customer’s need for a smooth vibration-free experience be translated into engineering requirements for the suspension and the suspension bushings?

Systems-level models can play a very useful role in this process. Models can be built to capture the frequencies of the main subsystems. The model is then excited with frequencies corresponding to basket rotation, runway surface or engine idle. Through a process of adjusting the model parameters, engineers can arrive at targets for the subsystems and subsequently for the components that comprise the suspension system and its interaction with other parts of the product.

Models drive early validation
A second important role that models play is in validating the design to ensure that the requirements have indeed been met. Often these models are more detailed and provide higher fidelity information.
Models support trade-off decisions
The third important role for models is to enable the right trade-offs to be made between conflicting requirements. Continuing with our earlier example, the suspension bushings also need to satisfy requirements arising from washing machine loading location, airframe load distribution, or vehicle handling and rough road impact considerations. The most effective way to consider multiple requirements and their trade-offs and to optimize the design is through the use of simulation.

Several hurdles must be overcome to make the process efficient. Siemens PLM Software’s approach to simulation is specifically designed to overcome these challenges and make engineers more productive by providing integrations, simpler access to geometry data and a consistent data model:

Integration
Integration with other simulation tools and with product development applications is important to both the efficiency and effectiveness of a systems-driven product development solution. Integration is a core philosophy that permeates Siemens PLM Software’s simulation solutions, providing them with unique and true differentiation.

Simplified geometry format
Analysts often need to edit models but are inhibited by a lack of expertise in knowing how to work with complex CAD geometry. Siemens PLM Software addresses this dilemma by providing a data model that separates design models from analysis models while providing tools that enable analysts to modify geometry without having to master the intricacies of CAD technology. Most importantly, when the design is changed, analysis models and results can be updated quickly and automatically. This integration dramatically reduces the amount of time it takes to extract information, use the simulation, process the results and feed this information back into the design process.

Consistent data object model
In systems-driven product development, the true value of simulation is not limited to islands of automation, but ripples downstream in the product lifecycle through re-use of the simulation results, as well as other CAE-derived knowledge. To facilitate this, Siemens PLM Software provides the right integrated tools for your PLM environment – tools that enable you to capture, update and share relevant engineering data and product knowledge across your entire global enterprise and supply chain.

Cofely Energy & Infra operates across the entire spectrum of transportation and traffic, including engineering and automation, consultancy and project management, telecommunications and security, maintenance and management, assembly and operation startup. “We use the systems engineering capabilities of Teamcenter predominantly for requirements management, systems definition, analysis and design management. The extra effort in the engineering phase saves time later in the process. Impact analyses show the largest benefit. When the customer changes requirements, we now have the ability to say where to expect the impact of these changes. This allows us to change the quotation accordingly. This removes an important part of the risk for both our clients and us.”

Rico Verhage
Systems Engineer and Consultant
Immersive user experience

The rich information that systems-driven product development provides must be at the fingertips of every person involved in your product lifecycle. Information needs to be provided intuitively, in accordance to context and situation of the engineer, focused on the business activity at hand and leveraging the geometric model.

With this in mind, Siemens PLM Software’s system-driven product development solution is:

- **Intuitive** Users should feel comfortable interacting with the PLM-enable environment. The solution understands each user’s role and presents information to the user commensurate with tasks associated with that role. This information is intuitive because it is consistent with the training and expectations of the user’s role.

- **Context and situational sensitive** When a user initiates any activity in a systems-driven product development environment, the solution will identify a body of information as being critical to support that activity. In addition, as the activity progresses, further information can be identified as critical to the successful completion of the activity. The user’s productivity is optimized because the software automatically manages the context that determines the scope of this information.

- **Activity focused** When a user performs an activity, the solution will invoke business processes that provide context to its execution. In certain instances, this might be a higher level process that defines specific actions that will take place. In other cases, the process may actually execute the activity. The solution anticipates the user’s needs for information and application usage.

- **Intelligent in terms of its geometry** As a design matures to the point at which actual product geometry is created, users must be able to use the geometry model to interact with and validate the system model in meaningful ways. The solution provides geometric product models with a fully functional window into the interrelated functions and requirements that helped define that geometry. These models can also serve as a critical input to more advanced validations of the system’s performance.
Conclusion

Siemens PLM Software provides the foundation to support system-driven product development throughout your enterprise, including capabilities that enable you to:

- Capture the voice of the customer
- Support your design, manufacturing and service operations
- Facilitate enterprise program management, change and issue management as well as configuration management
- Ensure consistent data and business processes

Siemens PLM Software’s open architecture is a necessary prerequisite to facilitating consistency in the real world – enabling you to integrate your processes across all engineering and manufacturing disciplines, leverage specialized design and simulation tools of choice and quickly adopt methods and rising innovations that nobody anticipates today but may become the de facto standard in the coming years.
Siemens PLM Software

Headquarters
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 972 987 3000

Americas
Granite Park One
5800 Granite Parkway
Suite 600
Plano, TX 75024
USA
+1 314 264 8499

Europe
Stephenson House
Sir William Siemens Square
Frimley, Camberley
Surrey, GU16 8QD
+44 (0) 1276 413200

Asia-Pacific
Suites 4301-4302, 43/F
AIA Kowloon Tower, Landmark East
100 How Ming Street
Kwun Tong, Kowloon
Hong Kong
+852 2230 3308

About Siemens PLM Software
Siemens PLM Software, a business unit of the Siemens Digital Factory Division, is a leading global provider of product lifecycle management (PLM) and manufacturing operations management (MOM) software, systems and services with over nine million licensed seats and more than 77,000 customers worldwide. Headquartered in Plano, Texas, Siemens PLM Software works collaboratively with its customers to provide industry software solutions that help companies everywhere achieve a sustainable competitive advantage by making real the innovations that matter. For more information on Siemens PLM Software products and services, visit www.siemens.com/plm.

www.siemens.com/plm
© 2015 Siemens Product Lifecycle Management Software Inc.
Siemens and the Siemens logo are registered trademarks of Siemens AG. D-Cubed, Femap, Fibersim, Geolus, GO PLM, I-deas, JT, NX, Parasolid, Solid Edge, Syncrofit, Teamcenter and Tecnomatix are trademarks or registered trademarks of Siemens Product Lifecycle Management Software Inc. or its subsidiaries in the United States and in other countries. All other logos, trademarks, registered trademarks or service marks belong to their respective holders.

31890-Y2  3/15  C