

**Systems Engineering and Simulation:
Essential for Smart Products and
Virtual Production**

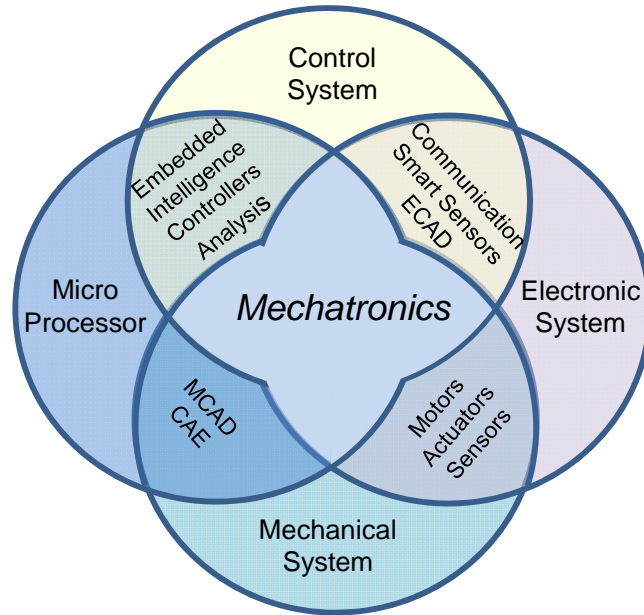
Executive Overview 3

Principles of Systems Engineering and Model-Based Design 4

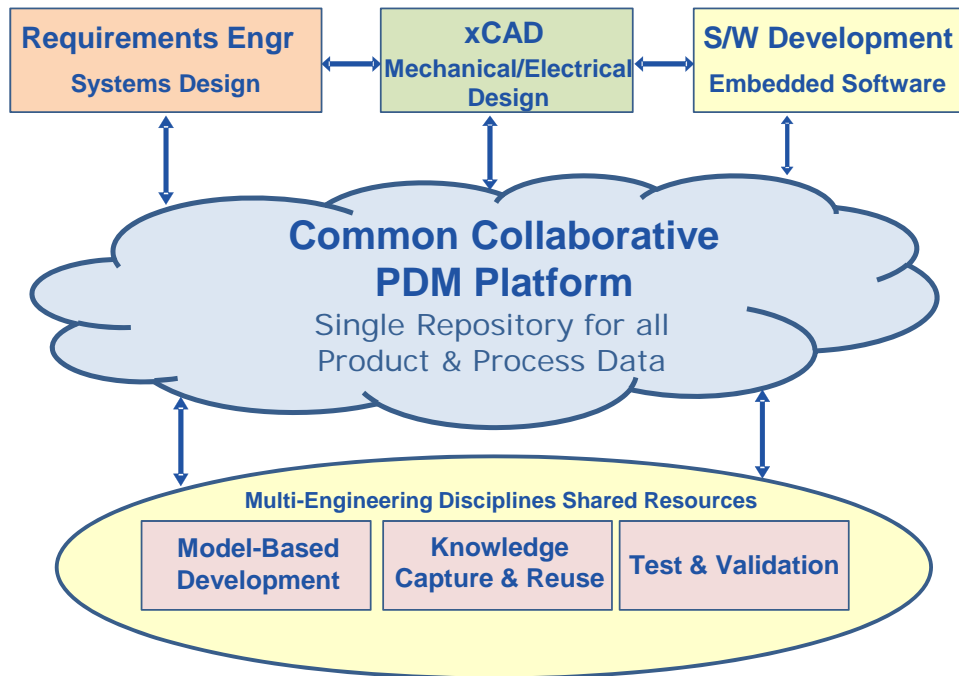
Single, Open Platform for All Design and Build Domains10

Conclusions13





Mechatronic Systems Integrate Multiple Disciplines



Common Platform Provides Single Source for All Data

Executive Overview

Increasingly, today's consumers are being exposed to an environment of intelligent products – from toasters to automobiles. The level of intelligence embedded in everything from our cars, to our homes, communication devices, consumer electronics, and even mundane items as our toasters and toothbrushes, increases every day. In the very near future, not only will humans interact with a rapidly growing array of smart products, but many of these products will interact autonomously with each other and other systems

To meet the engineering challenge of developing the next generation of smart products, many consumer goods companies are turning to highly sophisticated, multi-engineering development practices that have been proven in industry sectors like embedded systems, aerospace & avionics, automotive, and high-tech.

to monitor and control power usage for smart grids; automatically make our dentist appointments; monitor, diagnose, and schedule service for our vehicles; and lots more.

While manufacturers across all industrial sectors are ramping up to meet demand for this growing “smart product” market, they face major challenges in developing and manufacturing

new products that are significantly more complex. These mechatronic products require mechatronic design approaches in which mechanical, electrical, and software engineering disciplines are integrated. To meet the engineering challenge of developing the next generation of smart products, many consumer goods companies and other manufacturers are turning to highly sophisticated, multi-engineering development practices that have been proven in high-tech industry sectors like embedded systems, electronics, aerospace & avionics, and automotive.

In response, product lifecycle management (PLM) solution providers now offer systems engineering design platforms and solutions across multiple domains of the product lifecycle, including production systems. These allow engineers to apply a model-based, systems engineering approach to mechatronic product and process development. Manufacturers of smart products across all industries are adopting systems engineering design solutions to deal with the ever-increasing complexity of multi-discipline engineered products to meet cost, quality, and time-to-market demands. Systems engineering-based design platforms provide product developers and production system engineers with tools to integrate mechatronic systems using a functional design approach that allows for early stage validation. This helps reduce the time and cost of design recycles and late stage integration.

Principles of Systems Engineering and Model-Based Design

As we move to a world of more smart products and systems it will be rare to find mechatronic devices without some kind of embedded intelligence built into the design. Furthermore, the level of firmware intelligence provided by an embedded system will often be the key that differentiates the product from its competitors. However, the benefits of embedded intelligence come at a price. As mechatronic systems take advantage of more powerful microprocessors and the software that runs on them, the interaction between hardware and software becomes more complex. Managing

As mechatronic systems take advantage of more powerful microprocessors and the software that runs on them, the interaction between hardware and software becomes more complex. Managing this complexity is the challenge.

this complexity can prove challenging to hardware and software engineering teams that develop requirements, describe functionality, and test and implement the concepts in a variety of ways. The fact that most of these systems include closed-loop control methodologies that compensate for electromechanical interactions and other variables adds to the complexity.

Mechatronic design methods used today typically initialize the design process with mechanical modeling, followed by electrical design. Traditionally, when developing new software, engineers addressed software validation at a late stage in the development process, only testing the software through emulation on hardware prototypes. Just as electrical design imposed constraints on mechanical systems, software typically imposes significant constraints on the overall electromechanical system design. Compensating for constraints and errors found in hardware or software at this late stage creates costly delays in the overall development process. It can be time consuming to trace problems back to their root cause. Errors related to incomplete, incorrect, or conflicting requirements may even necessitate a fundamental redesign.

In contrast, embedded systems developers typically use a more concurrent approach in which the electro-mechanical device or equipment is targeted based on the overall function of the system and the software developed to meet the functional system requirements. This is the basic premise of systems engineering; developers consider the entire system, with the various engineering disciplines being evaluated together as functional whole. They functionally decompose the system to be able to fully understand and ac-

count for the effects and constraints of each sub-system within the context of complete system. Without a systems engineering approach, it is not uncommon for the iterative lifecycle of software development to undergo frequent engineering design changes due to late stage integration. Systems engineering utilizes an integrated approach because:

- **Mechanical** components place constraints on the design and layout of electronics, electrical interconnects, and automation software.
- **Electrical & electronic** subsystems require integration and optimization among multiple disciplines, from design and sourcing through assembly processes.
- **Software** development requires a wider range of skills, but depends upon design decisions in other domains.

Systems Engineering Design Methodology

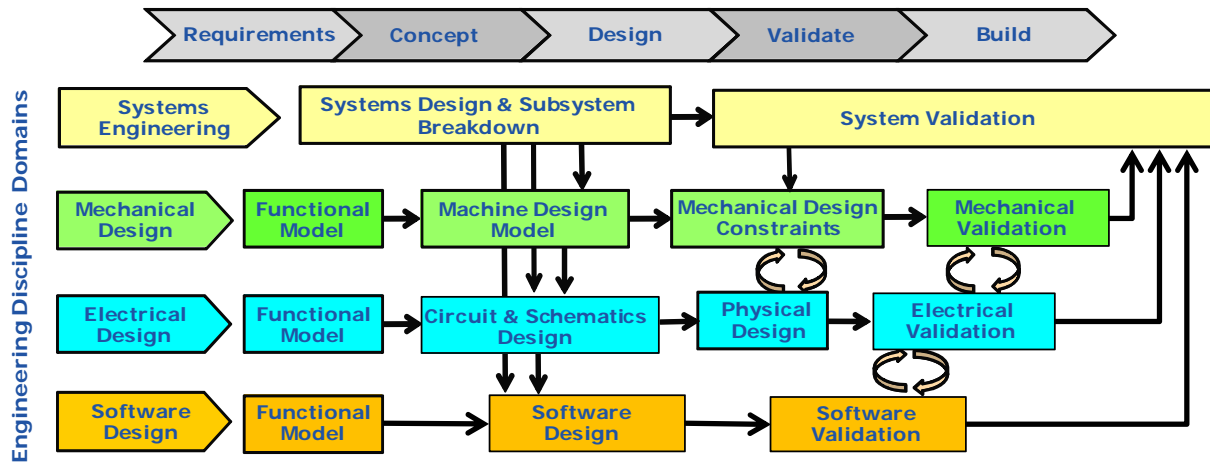
A systems engineering approach decomposes the design requirements of the overall system into subsystems and components, which are then associated with their individual functional specifications. This enables the various engineering design disciplines to define immutable integration

A systems engineering approach to mechatronic development often helps reduce complexity, lower component costs, and increase integration efficiency.

interfaces early on in the design cycle. This process naturally lends itself to concurrent development within each engineering discipline involved in the overall system design and validation. The systems engineering method provides the reference point for each design discipline and how they will interface. This allows developers to vali-

date the basic subsystems requirements and functionality concurrently for mechanical, electrical, and software design, and determine how each design discipline interacts within the context of the system, device, equipment, or machine. Furthermore, design tradeoffs between engineering disciplines are evaluated during an early validation process. This avoids late state integration that typically results in development delays.

Aside from encouraging concurrent design, a systems engineering approach to mechatronic development often helps reduce complexity, lower component costs, and increase integration efficiency. With an increasing amount of products being designed with built in intelligence, mechatronic system design drives engineering organizations toward more efficient engineering practices.



Systems Engineering Approach Drives Concurrent Design Process

It is not uncommon for product developers today to perform the mechatronic system validation per the system requirements as the last step in the development process. This can inhibit incorporating design tradeoffs, alternative component selection, and sensor locations that occur between engineering disciplines. All aspects of a mechatronic system, which include integration of mechanical, electrical, and software, benefit from this approach. The multi-discipline, mechatronic systems design approach is employed widely in product development for high-tech, consumer electronics, communication, telematics for automotive, and avionics for aerospace. However, as products within other industry sectors incorporate more intelligence (machine tools, production systems, and consumer goods, tec.), product developers will turn to systems engineering-based mechatronic development platforms. Typical systems engineering solutions sets would include:

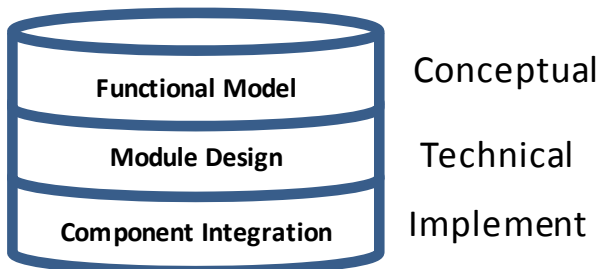
- Requirements management
- Functional analysis
- Logical analysis and design
- Systems modeling and analysis
- Simulation (virtual and logical)
- System emulation and validation

Several PLM providers now offer collaborative systems engineering solutions with full integration across their design/build solutions.

Functional Models

Functional modeling supports the overall design activity by describing problems and solutions in terms of their functions and allowing designers to create a hierarchy of relationships. Since the process is at a higher level of abstraction, functional models provide a basis to generate design solutions without getting into the mechanical or electrical design detail. Developers can decompose a functional model into sub-functions or sub-systems to fully define the relationships between the engineering disciplines involved while maintaining the overall integrity and requirements of the device or equipment as defined and designed in the context of a functioning system.

A functional model is developed to reflect the required system behavior along with any associated special or physical constraints. This enables the development process to be linked to the functional requirements of the me-



chanical and electrical systems and modeled in a unified environment. Additionally, developers can use a range of simulation tools to perform concurrent and continuous testing of the software against the electro-mechanical system requirements for each design iteration. In this way, developers can find specification and integration errors immediately and at the

beginning of the systems development process. This approach has proven to be an effective development method where embedded software must be functionally integrated with targeted mechatronic devices.

Model-based development also works well within the context of a systems engineering design approach. The system designer typically begins the process by developing a functional model of the complete system and the various sub-systems, creating high-level system views, and linking requirements and constraints to parts, assemblies, and modules. By modeling the complete system, including the software interdependencies and relationships between mechanical and electrical designs, the designer can validate the functionality of the system very early in the design process and continue to validate with each design iteration of any of the subsystems.

Model-Based Design Enables Multi-Discipline Engineering Integration

Model-based design (MBD) addresses the challenges of multi-discipline engineering integration, providing a structured methodology to implement systems engineering. It simplifies development of mechatronic systems by providing a common environment for design and communication across different engineering disciplines. MBD began in the early 1990s in the aerospace and automotive industries, which were continuously building more intelligence (and more microprocessors) into their products. Aerospace and automotive engineers recognized the advantages of simulating multi-domain systems when developing embedded controls.

During the last few years, other industries involved in mechatronic development have discovered MBD. The combination of more sophisticated software applications, more powerful desktop computers, and model-based

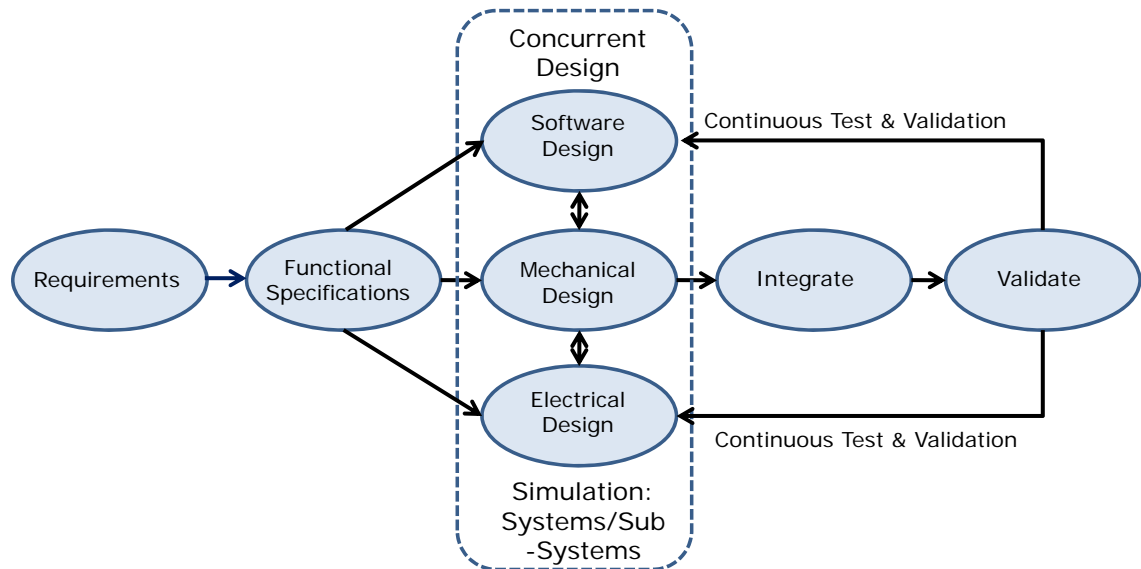
The combination of more sophisticated software applications, more powerful desktop computers, and model-based development platforms has helped MBD proliferate.

development platforms has helped MBD proliferate in much the same way that all industries ultimately adopted CAD/CAE software.

Where CAD provides a 3D virtual environment for the designer to geometrically define products and equipment, MBD incorporates the functional and performance requirements to properly describe the overall system. Since this approach is software driven, engineers can easily investigate a variety of designs and explore new concepts without the time and expense of building physical prototypes. Engineers can test the design continuously as it evolves, checking it against requirements and finding mistakes earlier in development when they are easier and less costly to correct. In addition, MBD automates code generation for the embedded system by eliminating the need to code the open- and closed-loop control algorithms manually.

MBD works in conjunction with a system-level model, providing developers with a tool to design and implement the system-engineered product. Typically, this model defines an executable specification by describing the natural and controlled behavior of the device or equipment in a logical and functional model that can be expressed in mathematical form. Engineers can execute the model by using applications that simulate the actual physical dynamics and performance of the system. The model specifies a rigorous mathematical definition of the expected performance of the mechatronic system.

The traditional development sequence in mechatronic systems begins with the mechanical design followed by the electrical design. Electrical design is highly dependent upon constraints imposed by mechanical systems, which dictate the location of sensors, electrical panels, wiring harnesses, and infrastructure. Typically, automation software functionality is not validated until late in the overall system development, even though mechanical system modeling is generally simulated prior to the actual build. Most automation software development can be based on systems engineering requirements, but without an accurate mechanical and electrical model, it is not always developed in an integrated engineering environment. Without modeling, functional errors found in the automation software when tested on a late-stage physical prototype can cause costly development delays.



Model-Based Design Allows Integration and Validation of Mechatronic Systems

Model-based design can address these issues confronting the machine tool builder by linking the system requirements for each engineering discipline to a model of the total system or modular subsystem.

Simulation: Essential Component for Verification and Validation

Simulation tools remain an essential element of the modeling, verification, and validation process associated with systems design and model-based development. Simulation is used whenever trying things in the physical world would be inconvenient, expensive, impractical, or simply impossible.

Simulation allows developers to try things with more control over parameters and better insight into the results. It reduces the time and cost of experiments and makes it possible to work with systems that do not yet exist in physical form. This eliminates the need for physical prototypes. When used in lieu of “real” (physical) systems, simulation and virtual systems can cut lead times and improve product quality.

Developers can use a simulated environment for two different purposes. One is to provide “typical” data to the system simulation, trying to mimic the behavior of the final physical system under normal operating conditions. The other is to provide “extreme” data, corresponding to boundary cases in the system behavior, and “faulty” data corresponding to broken sensors or similar cases outside normal operating conditions. The ability to inject extreme and faulty cases is a key benefit from simulation.

Taking the product models and placing them in a virtual environment allows developers to simulate, verify, and validate the actual physical behavior of the product.

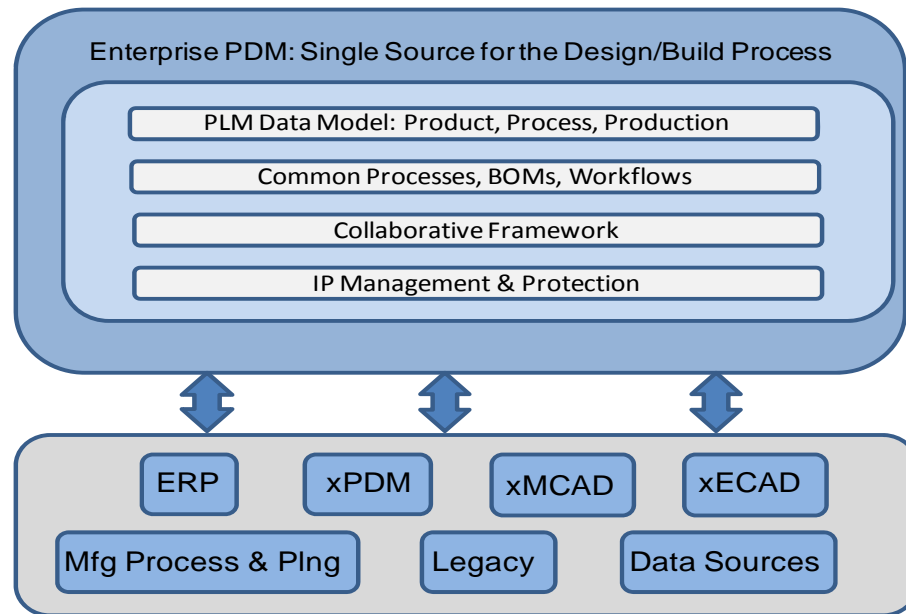
Today’s PLM providers offer CAD tools for product design and CAE tools (FEA, CFD, multi-physics) for product testing. Now, they offer systems engineering

solutions that provide virtual (3D) modeling tools that allow developers to use the virtual models of the product to simulate the physics-based (motion, torque, force, etc.) attributes of the device, machine, or equipment. Taking the product models and placing them in a virtual environment allows developers to simulate, verify, and validate the actual physical behavior of the product. Additionally, it provides means to validate the functional component interaction of the device based on the systems engineering functional decomposition and model-based design.

Single, Open Platform for All Design and Build Domains

Almost all major PLM solution providers have adopted the fundamental principle that their collaborative PDM systems must function as a single source and data model for all intellectual property (IP) across all enterprise domains of the product lifecycle. This means that collaborative PDM systems will be the core repository for all product, process, and production information. These systems will understand PLM context from all data sources and provide a single architecture for all collaboration between applications, engineering disciplines, business processes, and IP security. The

notion of a “single version of the truth” on a single platform for IP management and PLM context has been a goal for PLM users since the emergence of digital product development.



Collaborative PDM Platforms Manage Machine Tool Development

Clearly, systems engineering principles are fundamental to mechatronics product development. A systems engineering approach facilitates collaboration among multiple departments and disciplines – not simply within the enterprise but also among suppliers and strategic partners across the value chain. When supported by enterprise PLM, systems engineering provides a holistic design view.

Information Management in System Design

For product and production system developers to maintain, optimize, and realize the full potential of a multi-discipline engineering design team they need to enhance the collaborative capabilities of knowledge tools. Much information is contained and stored in domain-specific design tools. The entire organization can leverage this information to streamline design, development, manufacturing, and – ultimately – service of the machinery in the field. Alternatively, information can be lost simply by the lack of access to a single collaborative system. Collaboration environments are critical to support rapid deployment, concurrent engineering, problem solving and troubleshooting, dynamic learning, and decision making, while protecting intellectual assets. Collaboration early in the design and development pro-

cess provides the foundation for sustaining and improving production machinery reliability.

Reuse contributes significantly to reducing the time and cost of the overall design/build process, as well as allowing engineers to assimilate proven designs and features.

A prerequisite for applying continuous improvement principles is the ability to capture and reuse knowledge with all requirements, specifications, models, and design documents in a shared environment. The shared environment enables a development process based upon design reuse and common models across domains. Reuse specifically allows domain-specific engineering groups to access and search a knowledge base of existing designs, processes, and established best design practices. Reuse helps reduce the time and cost of the overall design/build process and allows engineers to assimilate proven designs and features. Ultimately, this can lead reduced component variations, moving toward a system of common components that comprise the basis for all machine designs.

From the perspective of mechatronics development, both in product design and production systems, a single collaborative platform for all mechanical, electrical, and software designs; plus access to engineering bills of material (BOMs), manufacturing processes, and requirements engineering; would help streamline the overall development process. Collaborative PDM systems offer engineering process management solutions that allow machine tool builders to integrate multi-discipline engineering teams by providing the ability to capture, manage, and synchronize product design data, as well as automating engineering change, validation, and approval processes. Typically, an open, interoperable PDM system can manage data from all major CAD/ECAD systems. This mitigates the need to translate and convert design data. In this collaborative environment, designers can bring parts, assemblies, and components from different MCAD and ECAD systems into a visual mockup, where they can incorporate, visualize, analyze, and mark up drawings during the design process.

Collaborative PDM platforms can also provide digital validation capabilities that allow designers to aggregate design changes continuously. Within a digital mockup environment, engineers can see design changes in real time. Designers can validate their changes, not only in the current configuration, but also against multiple configurations and an integrated environment that includes the various engineering disciplines involved in the overall design. Additionally, automated workflow capabilities allow engineers to execute change processes that provide all design/build partic-

ipants with easy access to necessary data. Designers can collaborate and evaluate the impact of proposed changes with the PDM system's configuration management capabilities. Cumulatively, this can result in an optimally designed machine.

Designers can collaborate and evaluate the impact of proposed changes with the PDM system's configuration management capabilities.

Manufacturers must consider continuous innovation as a factor. In today's highly competitive market, innovation in product design is essential to gain and maintain market share. Designers must not only have access to as much design information as possible, but also need to be able to share and exchange new concepts and processes, both within their own companies and beyond the walls of the enterprise. Collaborative design platforms can help OEMs create a culture of information and idea sharing that is the key to successful innovation.

Conclusions

Today, product developers and production system designers face a range of challenges. In product design, they must meet customer requirements as well as remain competitive in a market that demands constant product improvements and the inclusion of the latest technology. For the production system designer they must deal with integrating multiple engineering discipline designs and model, as well a large range of components and equipment that comprise today's complex production lines and work cells. Perhaps the most daunting challenge is to design and build products and production systems that not only integrate the latest advances in mechanical and electrical design, but also incorporate more complex embedded software that will differentiate the product from competitors' products.

Based on market projections there will be a significant need for next-generation smart products, devices, equipment, and systems to support energy efficiency at every level, as well as connected consumers, where intelligent products and systems will be ubiquitous across industries, business, and society. Managing this complexity across multi-discipline engineering design teams mandates that companies and their engineering staffs adopt a systems engineering approach coupled with a model-based design development environment.

Analyst: Dick Slansky

Editor: Paul Miller

Distribution: MAS-D, MAS-H, and EAS Clients

Acronym Reference: For a complete list of industry acronyms, refer to our web page at www.arcweb.com/Research/IndustryTerms/

BOM Bill of Material	IP Intellectual Property
CAD Computer Aided Design	IT Information Technology
CAE Computer Aided Engineering	MBD Model-Based Design
CFD Computational Fluid Dynamics	MCAD Mechanical Computer Aided Design
CMM Collaborative Management Model	OEM Original Equipment Manufacturer
ECAD Electrical Computer Aided Design	PDM Product Data Management
ERP Enterprise Resource Planning	PLM Product Lifecycle Management
FEA Finite Element Analysis	

Founded in 1986, ARC Advisory Group is the leading research and advisory firm for industry. Our coverage of technology from business systems to product and asset lifecycle management, supply chain management, operations management, and automation systems makes us the go-to firm for business and IT executives around the world. For the complex business issues facing organizations today, our analysts have the industry knowledge and first-hand experience to help our clients find the best answers.

ARC Strategies is published monthly by ARC. All information in this report is proprietary to and copyrighted by ARC. No part of it may be reproduced without prior permission from ARC.

You can take advantage of ARC's extensive ongoing research plus experience of our staff members through our Advisory Services. ARC's Advisory Services are specifically designed for executives responsible for developing strategies and directions for their organizations. For membership information, please call, fax, or write to:

ARC Advisory Group, Three Allied Drive, Dedham, MA 02026 USA
 Tel: 781-471-1000, Fax: 781-471-1100
 Visit our web pages at www.arcweb.com



3 ALLIED DRIVE DEDHAM, MA 02026 USA 781-471-1000

USA | GERMANY | JAPAN | KOREA | CHINA | INDIA | BRAZIL | ARGENTINA