Harmonizing Modeling and Simulation with the Development Lifecycle

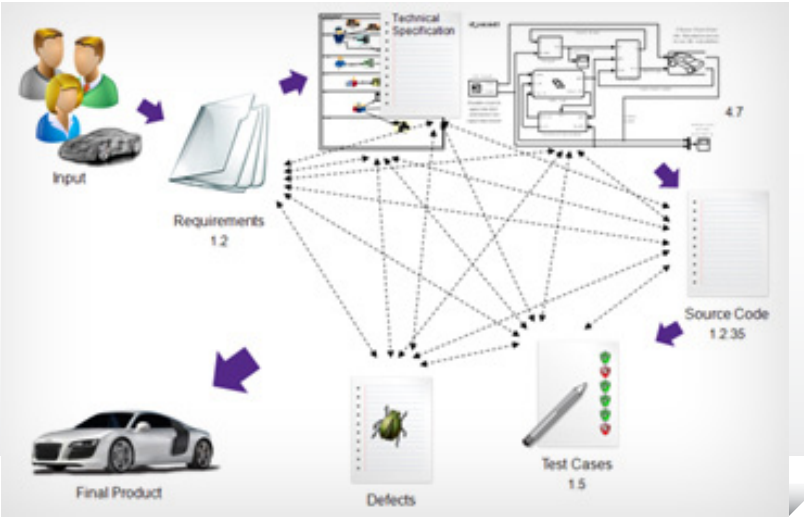
Improved Management of Models Accelerates the Development Cycle

Executive Overview

Product engineering organizations face the incredible challenge of ever shrinking market windows for innovation. These opportunities if missed, can lead to huge costs and overwhelming complexity that can compromise quality and lead to very expensive recalls. Innovating in the face of these pressures requires organizations to rethink their disconnected modeling practices and move to a unified lifecycle management approach integrating models into the core engineering process.

These companies face a perfect storm of dizzyingly short cycle times, quality and compliance demands, and a need to continuously innovate with limited or shrinking resources. A higher level of coordination and collaboration is no longer an option but rather a requirement for organizations to survive and thrive as they deliver products with embedded software that incorporate modeling and simulation into the lifecycle. It is a simple fact, manual methods and disconnected systems can no longer address the demands of today's development (let alone the future).

The solution offered by combining Integrity, a PTC product, with modeling environments and integrating model driven development with engineering processes allows organizations to increase the quality of their products while reducing the amount of time and effort needed to deliver those products.



To meet these challenges engineering teams need to ensure all assets within the development lifecycle, including models are effectively:

- Project managed (together)
- Test managed and recorded
- Configuration managed (again as a coherent unit)
- Leveraged (and reused) from requirements
- Traced to specifications and tests
- Managed in conjunction with calibration data
- Secured as a corporate asset

Communicated across the technology value chain (e.g. Automotive OEMs and Suppliers; Mobile Device Manufacturers and Carriers; etc.)

Harmonizing models produced by these tools with both upstream and downstream development activity creates a single source of truth for development and enables organizations to successfully address the increasing pace and overwhelming complexity of modern embedded system development.

Introduction

Automobiles. Aircraft. Medical Devices. Smart phones and consumer electronics. What do all of these complex products have in common? Embedded software is the primary source of innovation and differentiation, and software drives both the customer experience and perceived value. Today's engineering organizations must achieve excellence in the development, integration and delivery of software to remain competitive. As a case in point, a premium-class automobile contains about one hundred million lines of software code . The price tag of this code can reach 40% of the cost of the car's overall development (or about a billion dollars worth of investment). Ultimately, software is rapidly becoming both the greatest risk and the greatest opportunity for organizations delivering complex products.

The number one pressure on engineering organizations is to accelerate time to market ; this pressure is compounded by the unprecedented complexity in the products being developed. Other critical factors include compliance demands, globalized competition, and a need for continuous innovation (all while doing more with less and maintaining quality). Given all of these challenges, engineering organizations must achieve:

- End-to-end lifecycle management. Coherent, process-based change and configuration management must span physical and software components. Engineering of physical components (electrical, mechanical, etc.) is a well-understood problem, but software cannot be treated as 'just another part' due to the rapid change and interdependence inherent to software systems.
- Strategic integration of modeling and simulation. As complex systems that operate in the physical world, products with embedded software rely on modeling and simulation for design, optimization and testing. Modeling and simulation must be inextricably connected to requirements, test plans and other development assets, in order to mitigate risk, to ensure quality and to cope with the pace and volume of change.
- Control and collaboration to solve software complexity. Models and simulations for products with embedded software add to the already vast matrix of relationships, data, and changes in engineered products; this can no longer be managed effectively with manual effort and disconnected systems. In fact, the manual effort and rework implicit in these disconnected systems serves to stifle innovation and reduce competitiveness.

Engineering organizations need to get this right, because the stakes are high – best in class manufacturers of the most complex products [all of whom use simulation] get to market five to six months earlier with markedly lower product development costs. It is foundational to success and competitiveness that engineering organizations achieve effective management, enforceable processes and traceable relationships across requirements, models, simulations, source code, test cases and all of the other assets in development.

This paper will examine the challenges associated with connecting models and simulations to the greater development lifecycle, and provide an overview of the only solution available today that fully addresses those challenges.

Market Drivers & Challenges

All engineering organizations strive to reduce cycle times and this must be achieved in the context of regulatory and safety concerns, the growing volume and complexity of embedded software, and a constant pressure to innovate. Adding modeling and simulation to the lifecycle compounds the challenges faced by today's engineering organizations:

- Cycle times are put at risk when models and simulations are managed by manual processes and disconnected tools, particularly in environments characterized by high change frequency.
- Regulatory compliance can require that an organization report on the approvals and change histories of models and simulations associated with safety-critical components, leading to delays and overhead.
- The sheer volume of embedded software (particularly if that software is branched across multiple product lines) creates a challenge of scale in relating that software to relevant models and simulations.
- Leveraging generated code can reduce cycle times, but requires control and management to avoid quality and compliance gaps.
- Innovation is stifled when engineers are required to spend time and effort on manual processes and controls that would be better managed by a lifecycle management tool.

So Many Questions, So Little Time to Market

Models and simulations must be connected to the greater engineering lifecycle in order to answer key questions about product readiness and risk/compliance. It is not just about answering these questions accurately (which is important), but also the time and effort needed to answer these questions:

"Does this model address all of the relevant requirements?"

"Were the latest changes to requirements reflected in this version of the model?"

"Can we prove that the delivered product is in compliance with quality and safety standards (i.e. ISO 26262, DO178B, 21 CFR 860, etc.)?"

"Where is the simulation data from the third test run yesterday? Do we still have it? (If not, can we recreate it without too much overhead?)"

"Who approved the changes to the model and when?"

"We need to return to an earlier version of the model and associated simulations - where are they?"

"Were all problems identified by QA reflected in changes to the relevant requirements, models, and simulation runs?"

"Do we know with absolute certainty that the last approved change to the requirement is actually reflected in the functioning system going to production?"

"Where is the simulation data used to validate performance of the system that is currently putting us at risk of a recall?"

Historical Approach

Modeling and simulation was traditionally conducted in isolation by engineers on individual workstations or workgroups. Change control and version management tended to be manual or nonexistent, with models and simulation data often stored on local drives or network shares.

Application of modeling and simulation resulted in improvements to the velocity of the product lifecycle with reduced physical prototyping and efficiencies gained from blocks of automatically-generated code. That said, the 'manual and disconnected' approach to managing these new information assets and processes was neither scalable nor robust, and exposed the engineering organization to new risks and challenges. Without a centralized and connected repository for modeling and simulation assets, the risk to the company is that valuable knowledge is retained solely in the minds of engineers and analysts working on the models and simulations. The lack of processes and controls over this intellectual property also results in massive manual effort to achieve compliance/governance reports as well as an ongoing risk of errors or lost intellectual property.

Particularly as the pace of change increases and products continue to grow in complexity, the risks and costs of manual and disconnected processes can diminish or even outweigh the benefits of modeling and simulation.

Risks and Consequences

Examples: Requirements Change & Root Cause Analysis

The consequences of disconnected models and simulation are painful in the case of requirements change; all too often the domain engineers find out too late (if at all) about requirements change, making it difficult to adjust their models. Likewise when an associated test case fails, this disconnection translates into a tremendous amount of manual effort required to determine the root cause of the failure and determine the appropriate action(s).

Failure to connect models to generated code, test cases, requirements and other development assets leads to longer cycle times, higher development costs, greater risk, and (ultimately) reduced competitiveness.

Key consequences when modeling and simulation are disconnected from engineering lifecycles:

- Difficult and time-consuming to perform root cause analysis on failed test cases
- Costly, error-prone and time-consuming manual communication of requirements change when it impacts models/simulations
- Inconsistency in mapping design to the most current requirements
- Limited ability to assess and address the impact of a particular change before it is accepted
- Significant effort to identify and update test cases to current designs before lengthy test simulation runs
- A lack of rigor in capturing and relating data models/simulations, leading to lost data and rework
- Costly and time-consuming to demonstrate compliance with regulations or governance demands.

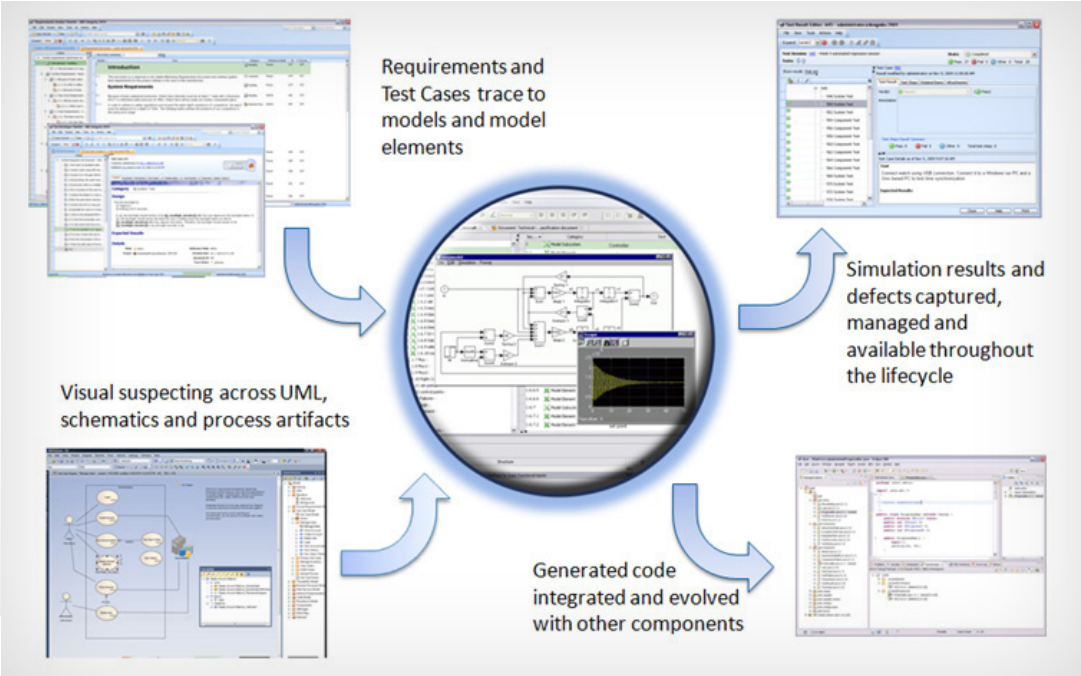
The Solution

Today's engineering organizations require a lifecycle management solution that can capture, manage and control models and simulations as assets in a coherent system. The configuration of and change to these assets must be managed, and the relationships with upstream and downstream assets in the development lifecycle (requirements, test cases, specifications, source code, etc.) must be established and traceable

When all contributors to a product are able to collaborate in a controlled fashion through such engineering change management, it:

- Eliminates unnecessary rework
- Enhances analysis and control of change
- Ensures that the delivered product is consistent with the latest requirements
- Promotes enhanced quality at every stage of the lifecycle.

Engineering organizations that can make models and simulation an integral part of the development lifecycle will reap the benefits of shorter cycle times, superior quality and reduced risk. As automation eliminates manual tasks and unnecessary rework, engineers are freed to innovate (instead of administrate).



Integrity with Modeling Environments

Together, Integrity and partners like MathWorks, Sparx Systems and Atego Artisan Studio enable modeling and simulation to become an integral part of the overall engineering lifecycle. As a result, the value of work by design engineers is extended to and accessible by all contributors to the product. PTC has the only solution that can rapidly and easily harmonize these modeling environments with both upstream and downstream development activities.

This solution extends existing requirements management, test management functions and process into the model-based development process. The end result is seamless visibility and control over the activities and systems being engineered, spanning requirement refinement, specification development, hardware and software engineering, and all verification/validation activities.

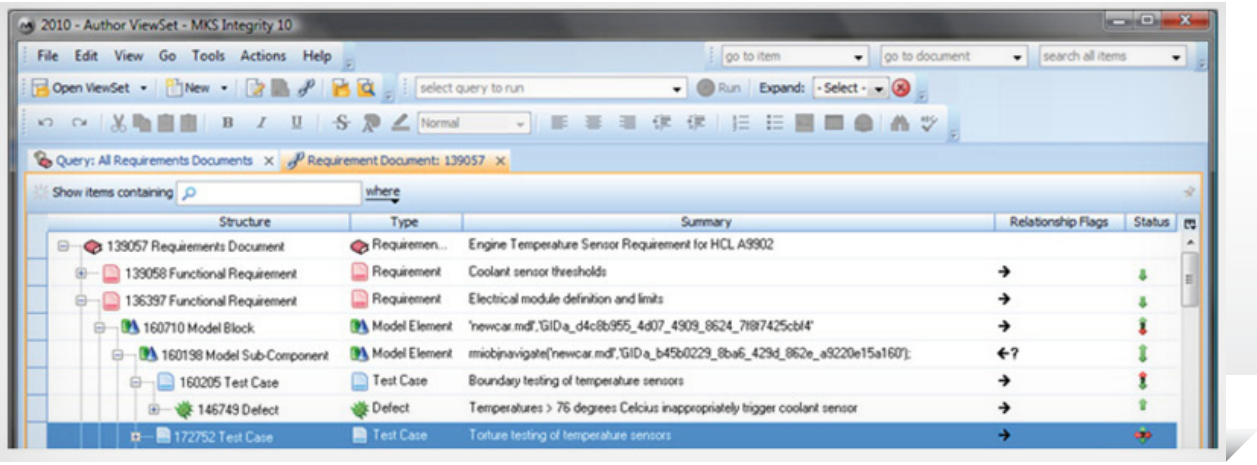


Figure A: Trace view with suspect flag

Solution	Benefit to you
Models are fully managed as part of the greater engineering lifecycle	• More accurate and complete assessments of engineering status and risk. • Project metrics include engineering modeling activities (i.e. size of backlog) and asset data (i.e. how many model elements are suspect?)
Generated code is captured and test managed with full traceability through the related models	• Model confidence is higher because the tests run against it demonstrate its quality. • Reduced effort to iterate models and test suites with fully traceable reuse-by-reference
Models are configuration-managed in the context of the whole project	• Baselines can span all assets including models • Reuse within or across projects with consistent sets of requirements and models • Full traceability across models dependency on other models and libraries
Models are fully traced and leveraged across related Requirements	• More rapid and consistent refinement of requirements. • Reduced effort and time to ensure all requirements are satisfied
Models are traced across Specifications and Tests	• Captured dependencies allow for impact analysis when considering or making a change • Automatic downstream notification when upstream changes are made (i.e. change to a model alerts engineers that dependent test case(s) may need review or updates)
Models are managed in conjunction with Calibration Data within a change process that harmonizes all assets	• A single change authorization can authorize multiple needed requirements and model changes • Full audit ability and accountability for engineers making the changes
Models and data are secured as corporate assets within a common repository	• Model can't be 'lost', modified or deleted without a trace and proper permissions • Greater ability to reuse models and data across projects and locations
Models (and changes to models) are communicated in a delta-cognizant way among OEMs and Suppliers	• Suppliers receive initial models and are notified of all updates in real-time • Reduced effort, cost and time to ensure compliance • Drastically mitigated risk associated with change to requirements and models.

Implementing the Solution

The Integrity solution allows you to start managing all of your model-driven development efforts from day one with automated model synchronization.

Given legacy tools and engrained/siloed practices, implementation of new approaches (and system support for these approaches) can be a challenge for many organizations. With Integrity, manufacturers can implement a solution that starts "narrow" and gradually expands to encompass more upstream and downstream activity.

Example: An initial starting point might be implementation of process-based change and configuration management for models (including test cases and the simulations that aim to prove the evolving models satisfy those test cases). Then requirements management, source and byte code management, end to end process and project management can then be added in steps. Integrations with PLM systems which remain as the system for Bill of Materials for the physical product can also be achieved in a step-by-step manner.

On the first day after implementation, users can:

- Synchronize a model with Integrity
- Easily link a model to test cases and requirements
- Capture the relevant information from simulations
- Leverage the Integrity Test Results Editor for manual testing of models
- As models are synchronized into Integrity, contributors to the development lifecycle will enjoy traceability between the model and related artifacts. All contributors gain the ability to determine which assets are impacted by upstream or downstream change – this permits analysis of change impacts and the mitigation of change risk.

Summary

Model-based development and testing are revolutionizing industry sectors like automotive and aerospace, but the models themselves (as well as the modification and testing of those models) are still disconnected from the overall development lifecycles. This disconnection leads to significant amounts of unnecessary rework and time wasted performing root cause analysis when inevitable changes occur. In today's market, where quality is paramount, governance is increasing, and there is mounting pressure to accelerate delivery cycle times, and this is where disconnected modeling represents a critical risk to the organization.

A higher level of coordination and collaboration is required if organizations are to survive and thrive as they deliver products with embedded software that incorporate modeling and simulation into the lifecycle.

The solution offered by combining Integrity with MathWorks MATLAB and Simulink, Sparx Systems Enterprise Architect or Atego Artisan Studio environments allows organizations to increase the quality of their products while reducing the amount of time and effort needed to deliver those products. Harmonizing models produced by these tools with both upstream and downstream development activity creates a single source of truth for development and enables organizations to successfully address the increasing pace and overwhelming complexity of modern embedded system development.

Integrity Business Unit Locations

North America
1 800 613 7535

United Kingdom
+44 (0) 1252 453 400

Germany
+49 (0) 711 3517 75 0

Asia Pacific
+65 6830 8338

Japan
+81 3 5422 9503

integrityinfo@ptc.com

For more information visit: PTC.com/products/integrity

© 2011, Parametric Technology Corporation (PTC). All rights reserved. Information described herein is furnished for informational use only and is subject to change without notice. The only warranties for PTC products and services are set forth in the express warranty statements accompanying such products and services and nothing herein should be construed as constituting an additional warranty. References to customer successes are based upon a single user experience and such customer's testimonial. Analyst or other forward-looking statements about PTC products and services or the markets in which PTC participates are those of the analysts themselves and PTC makes no representations as to the basis or accuracy thereof. PTC, the PTC Logo, Mathcad, Creo, Elements/Pro, and all PTC product names and logos are trademarks or registered trademarks of PTC and/or its subsidiaries in the United States and in other countries. All other product or company names are property of their respective owners. **The timing of any product release, including any features or functionality, is subject to change at PTC's discretion.**