Summary

Catalyst

Platform-based engineering is a perennial goal of automotive product development. Ideally, applying platform-based engineering practices should ensure that the product is comprised of design elements that are fully validated and therefore reusable across product families. However, as products such as automobiles have grown more complex, platform-based engineering has grown more challenging.

For example, embedded software systems have increased the level of interdependencies to the point where the modern automobile has become a classic “system of systems”. Modeling, a well-established design activity, is also a key pillar in helping automakers practice platform-based engineering because it enables them to reuse or repurpose pre-validated components or mechanical subsystems. With ever-growing product complexity, automakers now need to apply these principles to the development of all of their validated systems, subsystems, components, and features, across all domains including mechanical, electrical, and software. Furthermore, with growing product complexity, it is becoming increasingly challenging to keep all design artifacts, such as specifications, models, and underlying assumptions, synchronized. Compounding the challenge is not only the sheer diversity of models used in designing and validating systems, but...
also the fact that the various elements of complex systems often evolve in parallel, from conception through to manufacturing, and even after the vehicle enters service.

**Ovum view**

Systems engineering practices are an essential ingredient of platform-based engineering, and the growing role of embedded software is reinforcing the need for them. Today’s automobile has become a full-fledged cyber-physical system where the design and operation of physical parts is closely tied to the development of the embedded software that is implemented on a number of distributed electronic control units. Automotive OEMs and their top-tier suppliers therefore need to apply a fully integrated systems engineering approach that covers data management, modeling, simulation, and validation of the systems. This approach assumes that there will be multiple sources of the truth in the modeling activity in the automotive product lifecycle, and that there will continue to be multiple modeling and simulation tools and approaches. The goal is not simply to link the models, but instead to ensure that they reflect the most valid design assumptions, and that they will be used in the correct context with all the necessary interdependencies, in a product lifecycle where design assumptions are a constantly moving target. To support this approach, the underlying data management and business process infrastructure must support a federated data and process model that ensures that all users are working with the latest information. This information must be presented in the context of an overarching integrated systems definition.

**Key messages**

- Software is introducing more complexity and interdependencies to automotive design, prompting the need for automotive manufacturers to extend the practice of platform engineering to the systems engineering domains.

- The proliferation of tool chains and modeling languages is a barrier to effective traceability, impact analysis, and information-sharing across product organizations.

- Systems modeling practices must evolve from being stand-alone analysis of individual sub-systems, to being federated approaches that fully account for the interdependencies in automotive design.

- An integrated requirements, logical, functional, and physical (RFLP) approach to product definition and decomposition opens the path for attaining cross-domain traceability, and makes a logical starting point for applying systems engineering best practice to multi-domain modeling and simulation.
Standards are opening paths, not only for integrating modeling processes, but also for providing new, flexible capabilities that can help automotive companies develop the mixed-domain models that are increasingly required by today's vehicles.
WITH MORE COMPLEX AUTOMOBILES COME MORE COMPLEX LIFECYCLES

Software compounds the challenge for platform-based engineering

While platform-based engineering has long been established in automotive design, obstacles remain for its adoption in the systems engineering world. The growing role of embedded software has compounded the issue because it raises the level of design interdependencies. No longer electromechanical products, today’s automobiles have evolved to true cyber-physical systems where the design and operation of physical parts is closely tied to the software domain through distributed, networked electronic control unit architectures. While many OEMs have achieved the ability to design product families on common chassis, they have not been able to realize the same efficiencies with embedded software. For example, it is not uncommon for some OEMs to continue to develop multiple variations of relatively simple generic systems such as those that control the electric windows inside a car. Although this is a trivial example, it typifies the lack of reuse of validated embedded systems that occurs within diverse product organizations such as automotive companies.

Tool chains are proliferating

For many automakers and other organizations across sectors that deliver complex engineered products, a key indicator of product design complexity is the growing proliferation of systems design, simulation, and testing tools. Every engineering domain typically has its own unique tool chain (see Figure 1), and it is not unusual for automotive OEMs or top-tier suppliers to have literally hundreds of systems engineering tools, with little or no way of integrating them. Examples include:

- Product requirements that may be recorded in documents, spreadsheets, domain-specific requirements management tools governing embedded software, geometric models, state models, and a variety of other sources.
- Compliance documentation addressing applicable regulatory standards such as ISO 26262 or Euro NCAP, which are typically represented in MS-Word or PDF formats.
- Physical definitions that are created and represented through 3D-rendering capabilities in CADCAM systems, and often managed in product lifecycle management (PLM) vaults.
Dozens of test, analysis, and simulation tools across each of the domains that help engineers and developers refine product design features. These include product geometries, parts configurations and specifications, heat dispersion properties, materials choice, electrical circuit capacity, software systems architecture, and the allocation of software functionality to ECUs.

**Figure 1: Existing toolchain proliferation**

Source: Ovum
MODELING IS ESSENTIAL, BUT SPEED BUMPS REMAIN

So many models, so little time

As one of the pillars of platform engineering, modeling can provide the means for sharing designs for components or systems that have already been validated. Modeling enables automotive companies to validate components that are core to a vehicle platform, and then validate other components that are used for distinguishing specific products that are part of the family. Table 1 provides a high-level view of some of the different categories of models that are used at different stages in the development of a vehicle.

Just as there are many types of models, there are multiple modeling languages and tools. Two of the more commonly used approaches are:

- Simulink, a proprietary tool developed by The MathWorks, which has become a de facto standard within systems engineering communities. Simulink provides a data-flow-oriented state machine modeling environment that can be applied to the problems of multiple engineering domains. However, models from different domains cannot be readily mixed or run in the same environment. Furthermore, the low-level data-flow approach of Simulink makes the models difficult to reuse.

- SysML has had appeal to the embedded software development community because it adapts the familiar UML language to systems engineering. However, SysML has a critical limitation. It does not fully represent the software in the context of a highly complex, virtual product at the implementation, testing, and validation phases where there is no direct relation between the flat SysML diagram and the mechanical properties of the hardware it is controlling.
The popularity and variety of models has proven both a boon and a bane for product developers. On the plus side, the sheer variety of models provides extensive coverage across different stages of the product development process. However, for complex products, there have never been standards for exchanging the various types of modeling data or artifacts across disciplines, domains, and tools. Because of the lack of logical links between the systems architecture and physical engineering domains, the challenge is compounded for cyber-physical products, such as today’s automobile. For example, the results of a behavioral model simulation of a subsystem and its control logic may not necessarily automatically populate a new issue in a software development issue-tracking system if the simulation revealed a flaw in the underlying logic. Similarly, it is not likely that the results of such a simulation would update the performance model based on the latest state model assumptions.

Specifically, there is no ability to:

- Integrate systems models to provide complete functional and logical decomposition and persistence, and document their context.
- Manage system architecture configurations down to the entity level.
- Share systems architecture definitions with physical engineering domains in a consistent and unified way.
• Integrate artifacts of the product definition, from high-level requirements through to functional, logical, and physical models.

• Close the loop for verifying and validating the artifacts, and the models representing them, in the way that they will be implemented.

• Integrate the embedded software development process with other engineering modeling environments.

• Integrate or correlate modeling parameters and scenario assumptions with product configurations as they evolve over the lifecycle.

Not surprisingly, validating the integration of individually modeled and validated design features has traditionally been a manual, highly error-prone, document-centric process. At best, integration occurs at data exchange rather than at the process or semantic levels.

**Design interdependencies make model integration difficult**

Design interferences introduce additional challenges to modeling complex products because changes to assumptions in one model can have cascading effects on others. The performance of antilock braking systems provides a good case in point. Braking distance can be shortened by increasing the size of the tires, but bigger tires will in turn increase vehicle weight, increase vehicle rolling resistance, and ultimately penalize fuel economy. Adjustments may also dictate changes in embedded logic. The challenge is that it is extremely difficult to conduct change or impact analysis across multiple models and engineering domains, or to ensure that all the parameters and assumptions of the various models are in sync. There is a need for tools that can manage change and versioning, and provide a metadata layer and integration backbone that allows the artifacts from each domain to be logically and causally linked.
SOLUTIONS FOR ENABLING MODEL-BASED SYSTEMS ENGINEERING

The need for cross-domain integration

The first step toward utilizing modeling as a cornerstone to achieving full systems engineering best practice is to abstract the functional elements so they can be composed regardless of how they are implemented. This provides the logical basis for associating them and integrating the steps for verification, validation, and qualification.

The RFLP decomposition of systems, which initially grew popular among embedded software development teams, is now becoming more widely adopted across automotive OEMs and tier-1 suppliers. The benefit of the RFLP approach (see Table 2) is that it enables product teams to analyze design elements independently, opening the possibility of reuse, and providing a logical path for integration to gain a holistic view of the product definition.

Table 2: RFLP

<table>
<thead>
<tr>
<th>Artifact</th>
<th>Definition</th>
<th>Traditional tooling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements</td>
<td>Outlines the product or component’s purpose, and the capabilities essential for meeting that purpose</td>
<td>Word documents or Excel spreadsheets, or through specialized management tools provided by ALM or PLM vendors</td>
</tr>
<tr>
<td>Functional Design</td>
<td>Defines the tasks that the user, system, product, or component must perform</td>
<td>SysML, Visio, Simulink-based tools, Modelica-based tools</td>
</tr>
<tr>
<td>Logical Design</td>
<td>Provides 2D representations and simulations of control logic and system behavior</td>
<td>SysML, Visio, Simulink-based tools, Modelica-based tools, or 2D schematic CAD systems</td>
</tr>
<tr>
<td>Physical Design</td>
<td>Traditionally, provides 3D visualizations for specifying the envelope, form, and appearance. Ideally, extends that by encompassing the physical instantiation of software, devices, and the virtual experience that represents the interaction of the software and the hardware.</td>
<td>3D rendering capabilities in CADCAM systems, and in some cases managed in PDM design vaults</td>
</tr>
</tbody>
</table>

Source: Ovum
RFLP treats the product definition as a series of logically related and integrated artifacts, instead of as a rigid, linear workflow where elements of the definition are tied to the functions or logic that implement them. For modeling, a fully integrated RFLP approach lays the blueprint for rationalized modeling because the artifacts are more self-contained, more readily linked to related upstream or downstream artifacts, and are therefore better suited for repurposing.

Using integration, validation, verification, and qualification (IVVQ) feedback loops to validate elements of the product definition, RFLP also provides an important first step toward achieving the traceability that is essential for platform engineering because of the way it decomposes the product definition. The tool chains used for RFLP are often quite diverse. Table 2 shows that in defining a product there will be multiple sources of the truth. For example, there will be specialized sources of the truth for mechanical, electrical, and software domains, and across these domains there will be different sources of critical information addressing functional, logical, and physical design requirements. Traditionally, diverse stand-alone tool chains have hampered traceability across domains and disciplines. The key is to ensure traceability through:

- Federated metadata backbones that link each source of the truth.
- Process integration for ensuring consistent workflows between the right players for associating changes and results with artifacts.
- Adoption of standards in place of fragile, dedicated, and often proprietary point-to-point interfaces.

Configuration management and model integration are essential to keeping development on track

Stand-alone, siloed modeling approaches to RFLP-IVVQ are no longer adequate for addressing the combined pressures of reducing product lead times, improving responsiveness, and accommodating the greater levels of complexity introduced with software. Models must contribute to the validation of the entire system, not just the individual parts or sub-systems. Innovations in multi-physics analysis are enabling designers to combine simulation of multiple physical properties in the same modeling run. However, increasingly complex automotive designs, such as hybrid vehicles, are triggering the need to chain together models of multiple parts, subsystems, or systems to reliably represent operating scenarios. For example, there are different patterns of interplay between gas engines, electric motors, braking, energy recovery, and battery charging and discharging in hybrid vehicles that must be modeled in conjunction with one another under acceleration, deceleration, cruising, and braking conditions.
In complex systems such as these it is more important than ever to ensure that the modeling parameters for each of the components, subsystems, or systems are operating from current, consistent specifications to ensure that the systems can be modeled in the same way that they operate in real life. It is equally critical to support the capability to link the running of multiple simulations by different tools that represent each aspect of operation in order to optimize the complete system.

Ensuring that the correct version of any given model is being used requires the ability to manage the lifecycle and configuration of models down to individual modeling artifact levels. In turn, making sure that models are realistic representations of actual operations requires the ability to conduct multi-domain simulation. This allows numerous models spanning multiple domains to exchange results, and for design changes in one model to be propagated to other models, even if they are being executed with different tools or languages. Ideally, multiple models representing related phenomena should be logically orchestrated to deliver validated results for the complete system.

Standards are pivotal

Standards are pivotal for integrating modeling and providing end-to-end traceability. These will enable OEMs and suppliers to concentrate on developing new functionality rather than designing interfaces. Standards have come through several routes, including bodies that have developed standards such as TCP/IP and other Internet protocols, and open-source projects that have been responsible for technologies such as the Linux operating system and the Apache HTTP web server.

In the product-engineering world, some industry sectors are embracing open standards. An early precedent was set in the aerospace community where the US Defense Research Projects Agency (DARPA) originally fostered development of what eventually became the Internet. DARPA has continued to push for open standards in its latest research projects, an example of which includes the System F6 program, which is sponsoring the development of a new network architecture for inter-satellite communications that will be based on open standards. Another example is the Adaptive Vehicle Make (AVM), a portfolio of programs to streamline the design, verification, and manufacturing of complex defense systems and vehicles. The three primary programs, including META, Instant Foundry Adaptive through Bits (IFAB), and Fast Adaptable Next-Generation Ground Vehicle (FANG GV) programs, will extensively leverage open-source technologies.

In the automotive industry, the relevant standards that apply to modeling, software development, and co-simulation of systems include AUTOSAR, Modelica, and the Functional Mockup Interface (FMI), which came out of the recently completed MODELISAR project.
AUTOSAR

AUTOSAR (Automotive Open System Architecture), an initiative for standardizing the architecture and interfaces of software embedded into automotive ECU, is being developed by automobile manufacturers, suppliers, and tool developers. The standard, which is now stabilizing with the recent release of version 4.0, is designed around a tiered architecture that includes an application layer that contains all functionality, the interface to the hardware, and the runtime environment.

AUTOSAR frees OEMs of the need to devise their own proprietary systems architectures or spend time writing unique interfaces to different ECUs. By decoupling the application from how it is physically implemented on the device, automakers can focus software development on the embedded applications that define the driving experience and differentiate the car. Furthermore, by providing a standardized architecture, AUTOSAR allows designers the flexibility of taking advantage of the distributed, networked architecture that provides the information backbone within the modern automobile. As the level of software content in the vehicle grows, systems designers must determine how to distribute processing loads across different ECUs, and with AUTOSAR they have the freedom of deployment without the burden of having to port software to different ECU devices. In turn, AUTOSAR frees suppliers from having to worry about writing custom APIs to devices designed by different OEMs.

Geensoft, one of Dassault’s recent acquisitions, donated the code base for Artop, the AUTOSAR tooling platform. Although early AUTOSAR development originated from European OEMs, the standards organization has become very broad-based. The AUTOSAR standards organization includes most of the major automotive OEMs across North America, Europe, and Japan, along with many top-tier automotive suppliers, software developers, tools providers, and chip makers. With such a broad constituency, AUTOSAR adoption is about to become far more widespread.

Modelica

Modelica is a declarative, open-source modeling language that can support extensions for multiple domains including mechanical, electrical, electronic, hydraulic, thermal, control, electric power, and process-oriented components. Significantly, Modelica can support mixed domains within the same model, a benefit that has become important for automakers that are developing hybrid vehicles. It provides a higher-level alternative to Simulink, which is a tool that represents models as data flows in and out of blocks. It also provides a more dynamic alternative to SysML, which provides snapshots of a system state. Modeling languages such as Simulink and SysML will retain a considerable presence in the mechanical, electrical, control, and software engineering communities. However, Modelica will address the need for a higher-level language that is
extensible and supports artifacts relevant to the engineer's domain, and that can be easily integrated into a 3D-based virtual simulation environment.

Most importantly, Modelica is designed to be extensible, so each engineering discipline can add external libraries or design its own. There are already numerous open-source and commercial libraries available for adding domain-specific constructs to Modelica, including:

- Standard libraries that provide a large number of standard components in different engineering domains that can be used to build multi-domain systems models.
- Free libraries for areas such as discrete event-modeling and simulation of networks with embedded controllers.
- Commercial libraries for modeling dynamics of vehicle chassis, vehicle dynamics, hybrid electric vehicles, electric drives, power-trains, thermo-fluid systems, and other areas.

The growing availability of domain-specific libraries in Modelica makes automotive systems designers and engineers more productive because they are modeling relevant engineering concepts, rather than low-level, cryptic data flows. Modelica also facilitates communication and sharing of models because they are declarative and use terms that are relevant to the specific engineering domain or analysis type. In addition to the standard libraries, Modelica provides:

- Electrical thermal, fluid, control systems, hierarchical state machines;
- Provides numerical, string, file, and stream functions; and
- A growing selection of free and commercial libraries that target various forms of differential equations, property computation, neural mathematics, fuel cell behavior, vehicle chassis dynamics, and other areas.

The selection of commercial libraries is also expanding. For automotive companies, these address areas such as hydraulics, power trains, smart electric drives, vehicle dynamics, and spark and compression ignition engines.
FMI provides a new standard for model exchange and co-simulation

The only practical approach for integrating the models is through the definition and adoption of standards. The Functional Mockup Interface (FMI), a new standard developed for the automotive industry as part of the MODELISAR project, defines an open executable interface that can allow dynamic system models, or any embedded software associated with them, from different systems to interoperate (see Figure 2). This enables co-simulation, which is the ability to dynamically interoperate multiple related models. MODELISAR and FMI are coming to fruition. With the MODELISAR project having been completed in December 2011, FMI has begun picking up critical mass support. As this report was being published, FMI has drawn support in more than 30 tools from 20 third-party vendors. Going forward, FMI will continue to evolve under the auspices of the Modelica association.

FMI exposes models that are deployed as executables, called Functional Mock-up Units (FMUs) that may be executed directly or through a third-party simulation engine. Even though these standards are early in their development, they can provide automakers with guidance and a framework to implement co-simulation either locally or distributed over a network. In upcoming
releases, the MODELISAR Consortium will be enhancing FMI to improve support for existing simulation tooling, particularly for coupling with third-party tools such as testing and diagnostic tools that are required during testing activities.

Although initially developed in the context of the automotive industry, the MODELISAR consortium is domain-agnostic. The FMI standard will provide the same benefits for other complex product sectors.
DASSAULT'S MODELING-BASED SYSTEMS ENGINEERING VISION

Integration through a unified architecture

Dassault Systèmes, a provider of product lifecycle management (PLM) solutions, has focused on several strategies to support a fully integrated model-based systems engineering approach. The architecture of its current V6 products is built on a unified data and process backbone that enables globally distributed product teams to work with a common version of the truth and to support traceability. Dassault is delivering this through a combination of web and rich clients that keeps design teams around the globe connected, maintaining a central metadata layer to keep teams in sync, while locally replicating large engineering files to optimize access.

Dassault's platform-based engineering support begins with a common product data model that encompasses requirements, platform, program, project, product, system definition, and configuration-management capabilities. This is core to the ENOVIA PLM family of products that enables collaboration between creators, collaborators, and consumers. It also drives the digital product experience provided by CATIA, Dassault's platform for systems engineering, mechanical design, equipment and systems design, and shape design.

From this model, Dassault's systems engineering solution supports an RFLP decomposed view of the product definition that:

- Combines requirements management, functional, logical, and physical design processes on a common platform
- Closes the loop with the model, scenario, result, and qualification (MSRQ) paradigm to support systems validation, verification, and qualification; and
- Supports full change and configuration management across the lifecycle, from hardware to software, and full traceability and impact analysis between the steps of the systems engineering "V model" (a classical software development process often employed in the systems engineering world). A simplified version of which is shown in Figure 3.

The premise of the V model is that every stage is tested, verified, and validated according to the IVVQ methodology. In spite of its premise, the effectiveness of the V model has been limited by the lack of integration between the tools that are employed at each stage of the process. For example, while requirements can be validated through operational testing, with existing tooling
Because Dassault uses a common product data model and process integration backbone, it can deliver a unified environment that closes the loop in the V model. Dassault implements the IVVQ feedback loop through an MSRQ-based compliance process (see Figure 4). This is a multi-step process that:

- Organizes the V model steps that are essential for validation into a sequence that comprises documenting the requirements;
- Generates a physical model from the requirements, creating different usage and load-case scenarios;
- Executes the model against the scenarios, generates and analyzes the results; and
Qualifies the results against the goals set by the original requirements.

Integration to PLM in turn helps automakers define the broader process including planning, a resource model for defining and validating product manufacturing, delivery, operation, maintenance, and end of life.

**Support for multi-domain modeling and emerging standards**

Modeling is a major pillar in Dassault's platform engineering product strategy. Dassault integrates modeling into the core systems engineering process through a lifecycle management approach. For example, simulation tasks can be integrated into higher-level business processes such as project management, engineering change management, and risk mitigation.

![MSRQ process](image)

Dassault supports the emerging MODELICA, AUTOSAR, and FMI (MODELISAR) modeling standards in its tool chain, and is also an active member of the respective standards consortia. In
addition to Dassault's support, these open modeling languages are also supported by numerous third parties. Dassault's modeling solution leverages the multi-domain support of MODELICA by supporting mixed domains within the same modeling activity. This capability is critical for validating entire subsystems or systems, and is typically required at design stages such as validating embedded code in the controller (software-in-the-loop), followed by validating the controller in the assembled system (hardware-in-the-loop).

Dassault's SIMULIA family of modeling tools provides finite element analysis, multi-physics capability, and simulation lifecycle management that help product organizations to keep their models and parameters current and in sync. Through its support for FMI, the SIMULIA tools can be used for cross-domain systems optimizations, such as linking embedded systems with computational fluid dynamics, finite element, and other analyses.

Acknowledging the reality that modeling is a highly diverse set of activities that typically involves multiple tools, Dassault's tool chain also provides open interfaces that support co-simulation through accepting the output of models developed in Simulink, SysML, and other languages. Its multi-disciplinary design optimization products are built on the SIMULIA Isight technology that loosely couples relevant modeling codes and automates their execution, analysis, and iteration.

**Seeing is believing**

The results of product development must be communicated to stakeholders, and Dassault's PLM vision can be summed up in the phrase “seeing is believing”. Along with other major CADCAM vendors, Dassault has promoted 3D modeling for its utility in defining the physical design of a product, and in communicating the look and feel of the virtual product to other stakeholders. By showing in vivid 3D photo-realistic visualization the appearance and behavior of a design, members of the product team, suppliers, and customers can more easily understand the properties of the product, which can help reduce product development lead time.

3D visualization is long-established in the CADCAM market. However, Dassault's vision of a "3D digital product experience" seeks to carry the capabilities of 3D further, embedding intelligence into design models to generate more realistic virtual simulations, with 3D geometry ultimately being controlled by the embedded control systems that will be instantiated in the real product. Modeling is a core pillar of Dassault's strategy because it provides the ability to understand not only the functionality and performance of a part or component, but also a fuller understanding of (literally) the bigger picture.
RECOMMENDATIONS FOR AUTOMOTIVE OEMS AND SUPPLIERS

Riding the innovation wave

There is little question that the automotive industry is undergoing major structural transformation. The OEM and supplier ecosystem are rapidly consolidating, end markets are fragmenting by geography and economic cycle, and the core technology is on the cusp of major reinvention with new platforms such as hybrids, plug-in electric, and possibly even fuel-cell-powered vehicles.

Meanwhile, the growth of "soft" content in the vehicle has placed the industry on a convergence path with the rapidly innovating world of high technology. Each of these innovations is transforming the vehicle into a more complex product that is starting to become a piece of "embedded hardware" that participates in a much larger "system of systems". Given the pace of structural change to world markets, traditional five-year product lead times can no longer keep pace, and lead times of between 24 and 36 months are becoming the new norm. The onus is on OEMs and top-tier suppliers to find more effective ways of riding the innovation wave.

Amid all this change is the growing role of embedded software. It is estimated that embedded IT content and related electronics will comprise at least half of new car manufacturing costs by 2015. The addition of software as an intrinsic design element of the car not only adds another engineering domain to the product team, but also greatly increases the web of design interdependencies. Admittedly, the ALM and PLM vendor communities have yet to firm up the rules of engagement and process of integration. However, OEMs and top-tier suppliers cannot afford to wait for the vendor community to fully align with definitive standards that cross domains, and they must therefore begin to rationalize engagement from the ground up.

Modeling is pivotal to performing effective systems engineering, and it is critical for automotive OEMs for which platform-based engineering is a core design strategy. Although modeling is hardly new to the automotive industry, demands for the design of more complex platforms, with a growing web of system interdependencies, has dictated a new approach for integrating, federating, and managing models throughout their lifecycle. Although integration between traditional product design domains and the systems modeling domain remains imperfect, standards such as Modelica and FMI are beginning to fill the gap. In the automotive software engineering space, standards such as AUTOSAR are commoditizing the core architecture of software embedded into ECUs, eliminating a non-value-added activity for OEMs and suppliers. Clearly, the impact of AUTOSAR will add agility to the design process for embedding software control. Although this might appear to
provide a clear benefit to OEMs and to potentially complicate life for tier-1 suppliers, in the long run adherence to the standard will make suppliers more competitive and agile in their own businesses.

Dassault's strategy has been to help drive standards to promote a more integrated approach to systems modeling that takes a systemic view that sees models not as individual sub-optimizations, but instead as product-level exercises that optimize the design of the whole car. This approach is critical to its vision of delivering the "3D digital product experience" with the goal of embedding intelligence in 3D simulation.

However, the ultimate direction is unmistakable: modeling must become an integrated activity that can federate the legacy with the new. Multi-physics simulation and openness are critical strategies for automotive OEMs and top-tier suppliers to master the design of a new generation of vehicles that as a result of embedded software carry more design complexity than ever. OEMs and suppliers alike should factor this in when selecting their systems design, management, and simulation tool chains.
APPENDIX

Further reading

Software Development in the product lifecycle, Ovum, April 2010

ALM in Systems and Product Engineering: Two Case studies, Ovum OI00068-006, March 2011

Links

Modelica Association -- https://modelica.org/

MODELISAR Consortium -- http://modelisar.org/

AUTOSAR Development Partnership -- http://www.autosar.org/

Authors

Tony Baer, Principal Analyst, Ovum IT Software

tony.baer@ovum.com

Michael Azoff, Principal Analyst, Ovum IT Software

michael.azoff@ovum.com

Ovum Consulting

We hope that this analysis will help you make informed and imaginative business decisions. If you have further requirements, Ovum's consulting team may be able to help you. For more information about Ovum's consulting capabilities, please contact us directly at consulting@ovum.com.

Disclaimer

All Rights Reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior permission of the publisher, Ovum (an Informa business).
The facts of this report are believed to be correct at the time of publication but cannot be guaranteed. Please note that the findings, conclusions, and recommendations that Ovum delivers will be based on information gathered in good faith from both primary and secondary sources, whose accuracy we are not always in a position to guarantee. As such Ovum can accept no liability whatever for actions taken based on any information that may subsequently prove to be incorrect.