DASSAULT SYSTEMES

HIGH-TECH

3DEXPERIENCE FOR SOFTWARE LIFECYCLE MANAGEMENT (SWLM)

A federated platform for software driven innovation

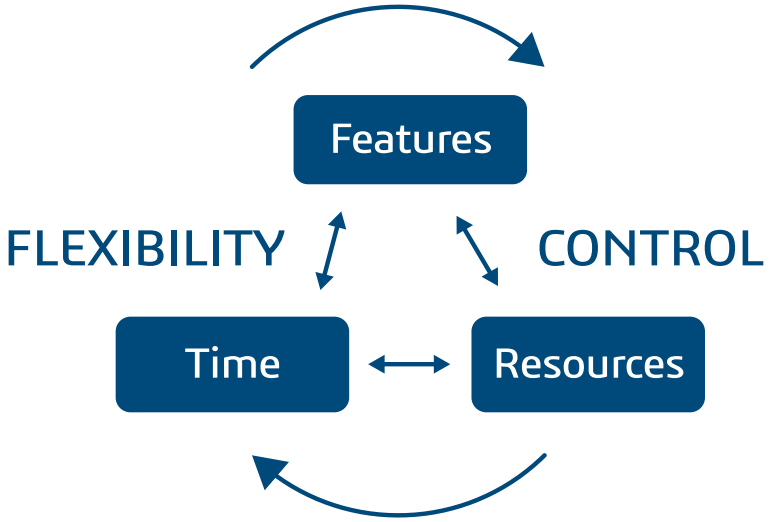
TABLE OF CONTENTS

EXECUTIVE SUMMARY

In all industries, delivering more software is becoming the best way to answer customers' demands for unique experiences. But companies are struggling to manage this transition to software intensive products.

One of the key elements for success is to master the hardware and software together, in common complex configurations, and to master the compatibility of software and hardware components as they evolve at different rates and are managed by different teams.

Another key element is to continue delivering quality products, even though the delivery cycle is quickly increasing. In other words: how do you become more agile, but maintain control of the product you design and the experience you offer?



Software design has always been characterized by a tradeoff between time available for development, resources to do the work, and features you want to implement. The best solution today to optimize this is an agile methodology that advocates continuously readjusting decisions with maximum flexibility.

The **3D**EXPERIENCE® platform from Dassault Systèmes enables Software teams to work in a rapid, nimble manner, using best-in-class development tools (commercial, open source, home grown, etc.). At the same time, engineers can control the work, monitor progress, trace IP (internal and external, managing security and protection of critical data) and guarantee quality. To provide both flexibility and control, the environment can support traditional waterfall methodologies as well as agile and modular approaches for all types of multidiscipline complex products. Frequently, for more complex architectures in a regulated space, some kind of hybrid approach is used—an agile process elicits key architecture aspects, then reverts to a waterfall method when key architecture and requirements aspects have been adequately fleshed out. Ultimately, the **3D**EXPERIENCE platform provides the opportunity to simulate the full virtual experience of the product with its embedded and application software, before actually delivering the experience to the end customer.

This white paper explains how the **3D**EXPERIENCE platform supports managing software and how platform openness allows development teams to work with their preferred software tools while governing the software for synchronization, configuration and compatibility with the system and hardware.

MANAGING SOFTWARE

The need for more software

Today's products are getting smarter. A smart product can react to situations: monitor the world through sensors, understand the data coming from the sensors through software, and adapt its behavior according to the situation, again through software. Products are not just smarter, but also adaptable, because even though the hardware cannot change easily, the software embedded in the product can change in an instant by a simple download over the air.

This potential to build smarter and adaptable products is changing their nature. Products can be perfectly tailored to the needs of its users. For example, not only does a scale measure weight, it monitors its user's weight evolution and helps reach fitness goals. Today's scales offer a personalized experience, comprising the physical scale, the human interface that shows progress (and gives the weather forecast), the mobile app on a smartphone to follow progress and set goals, and the web community to discuss weight loss strategy and find an exercise partner.

Manufacturing companies today, therefore, need to master the relationship between the embedded software (that makes the product smart) and the accompanying application and server software (that creates the full experience). Such mastery is necessary for most physical products on the market, whatever the complexity, whether a scale, a car or something else.

In order to deliver continuous value and enhance the customer experience, companies need to manage the design, lifecycle and compatibility of software and hardware. These processes need to be supported not only during the design of new products, but also during their deployed life while delivering value to the customer. This is how the experience can be continuously enhanced with new features while the physical product stays the same.

ONE GOVERNANCE: MANAGING SOFTWARE AND HARDWARE TOGETHER

Managing a product's lifecycle requires managing all its components: hardware, software and even services from inception to delivery and services through one global platform. Even though software is different than hardware in how it is designed and managed, its design needs to be in sync with the hardware, at all steps of its life, to optimize architecture, design and reuse of the product's components.

More specifically, the alignment of software and hardware during all steps of the lifecycle offers extremely valuable services to the end user:

- **Configuration management**: Define precisely which hardware and software configuration items go together in each version of a product or component, for reuse and traceability
- **Compatibility**: Manage compatibility between hardware and software modules or between different software modules
- **Multidiscipline simulation**: Enable simulation of multidiscipline systems and sub-systems, mixing software and hardware at all levels of definition and maturity, from functional behavior simulation to hardware and software
- **Configurable reusable components**: Enable reuse at all levels of granularity of simple software modules, but also of more complex systems combining software and hardware. Each module is defined as a generic configurable and reusable asset contributing to the product's definition
- **Reuse of all related assets**: Enable reuse not just of the code or definition of a module, but also its requirements, behavior, test cases, rules of usage, rationale and all information that is part of its release

MAP OF PROCESSES

There are four big families of processes necessary to manage software:

- **Software design** processes cover the definition of what a certain software module should do, from requirements definition to architecture and detailed modeling of the module's function, behavior and internal data model. This process ends with a detailed code specification or even with the generation of the code corresponding to the specification, when the tools offer this possibility.

- **Software development** processes cover the actual coding of the software, which is still mostly done manually. Software engineers usually work in their Integrated Development Environment (IDE) like Eclipse, Visual Studio and others. They access the code by check-out and check-in from a Software Code Management tool (SCM), like GIT, Subversion, Clearcase or ENOVIA DesignSync; and they access tasks for debugging or creating new code using a Defect and Activity Management tool (DAM), like JIRA or RTC.
- **Continuous integration processes** cover all the processes that happen once the code is checked-in. This comprises building the code, testing the code through unit tests, integration tests, incremental verification in successive layers of integration and ultimately validation in the intended deployment context. All these steps are today becoming more and more automated in a continuous process called continuous integration. The code is then packaged to be delivered to the customer or used inside hardware/software simulations for system integration and verification.
- **Governance processes** ensure that the software is correctly managed, inside products or as products. This is where Product Lifecycle Management (PLM) brings industrial-strength expertise on managing complex products inside product lines using a modular architecture, ensuring global configuration management of the entire system. PLM enables and facilitates collaboration of all parties, in particular, third parties in the definition and implementation of complex products where parts may be outsourced or open-source. Finally, PLM supports management of certification and compliance to regulations and standards, such as ISO26262, ARP4754 for airborne systems, DO-178C for airborne software, or DO-254 for airborne complex hardware, or security for EAR (Export Act Regulations) export considerations for products under export control, as indicated by export control classification numbers or ECCN.

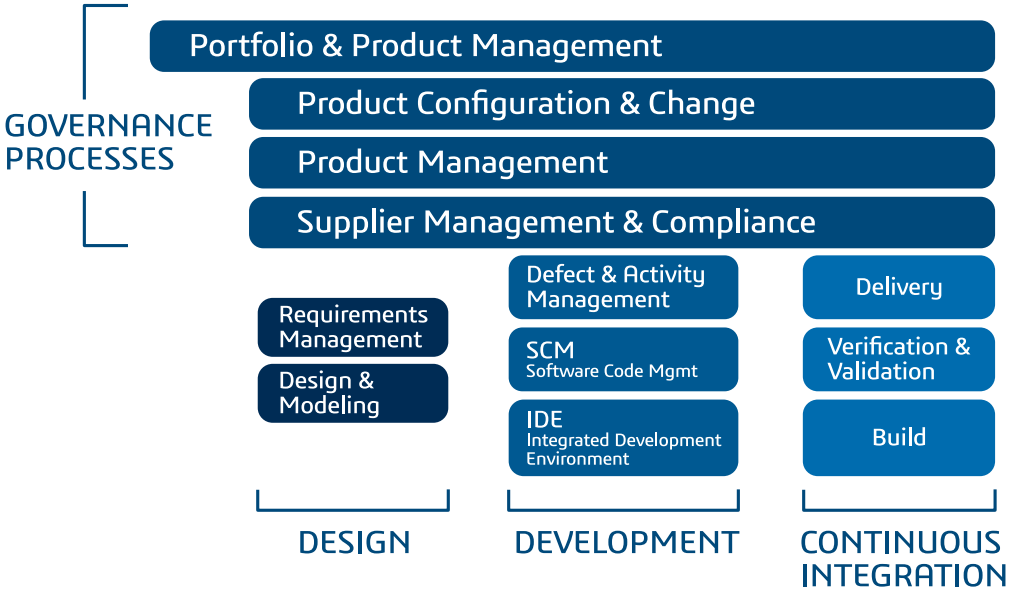


Figure 1
Software Lifecycle Management processes

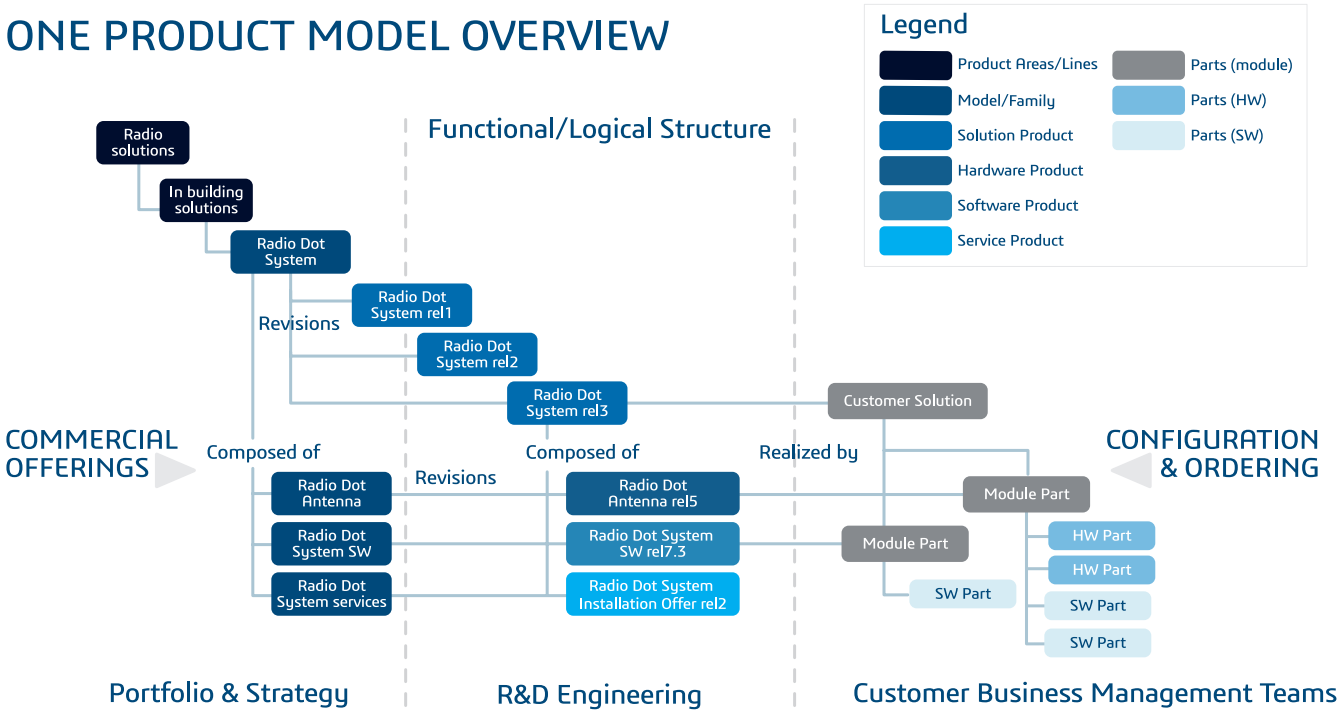
The **3DEXPERIENCE** platform provides a rich set of applications and infrastructure to help execute and manage systems, software and hardware lifecycle processes; however, inevitably, there are best-in-class tools in use by enterprises that need to be incorprated into the process[1]. Therefore, we provide open solutions to federate those tools into the platform.

[1]One of the things that needs to be done in this case is understanding the operational context of the particular tool and the affected core processes and related objects, as well as the various completion points and the temporal nature of the processes, objects, and relationships and collaboration workflows. With this operational context, the system interfaces can be archtiected in accordance with single-source of truth principles to affect the necessary implementation that affords a well-performing system configuration.

Network Equipment Company example

The **3D**EXPERIENCE platform has a unique data model that offers multiple views and levels of analysis, given the point of view. Product Strategy and Portfolio teams work at a very high level to define the content of new solution offerings (reusing and evolving modules from previous solutions or building new ones). R&D teams work on their specific family of products to define the engineering data, as well as the rules of use and configuration of new versions of technology assets. Sales and Marketing teams configure the R&D products into specific solutions tailored to customers' needs. The data model manages the software and hardware modules together and the compatibility between those modules.

ONE PRODUCT MODEL OVERVIEW



A MODEL BASED DATA CENTRIC PLATFORM

The **3D**EXPERIENCE platform enables data-driven processes by unifying product data into one single referential. This approach allows developers to manage hardware and software products in synchronization. The authoring tools provide data that can then be combined inside models of functional systems for experience modelling. This model-based approach provides smooth integration between the different layers of specification, design and implementation, as well as full traceability from the requirements that describe product needs to the architecture of modules that realize the necessary functionality. Traceability extends to the line of code and the built executables. These can then be directly tested with the virtual representation of the physical product, allowing model, software and hardware in the loop verification and validation of the designed systems and systems of systems.

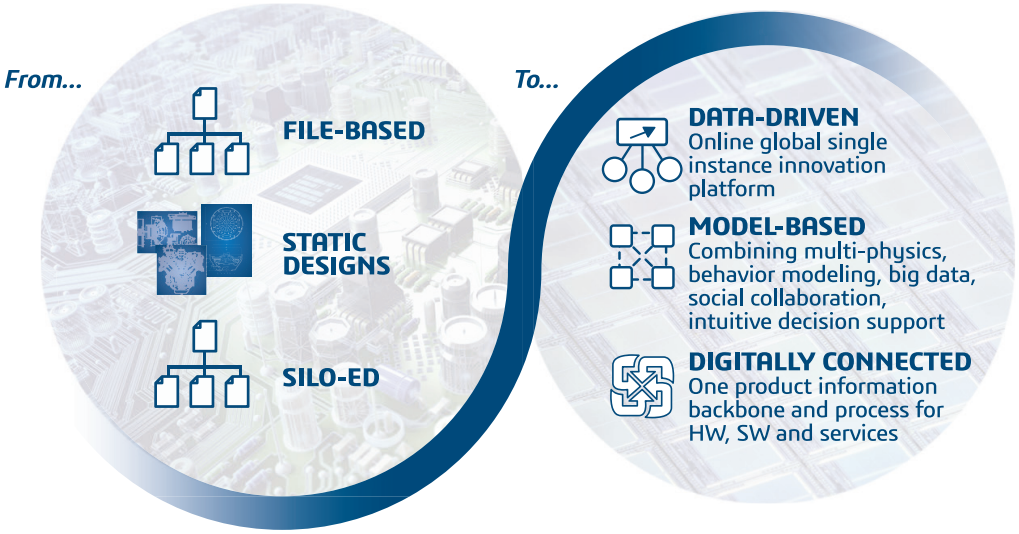


Figure 2
A data-driven, model-based and digitally connected platform

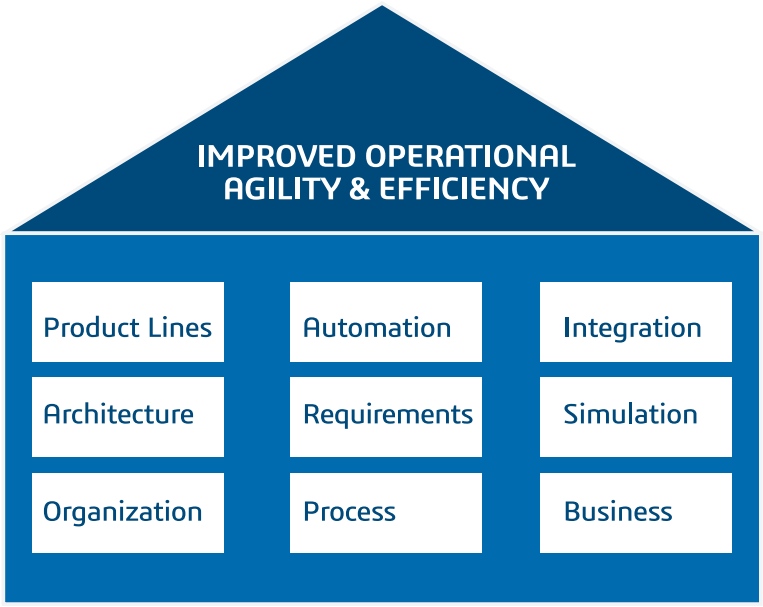
An aerospace industry example

Dassault Systèmes has deployed solutions that effectively and economically enable the conceptualization, development, manufacture, acceptance, deployment and maintenance of products (and in particular software products) in a product-line centric development context. In the aerospace & defense domain, one publicly acknowledged application is from Pratt & Whitney in the development of engine software. Pratt & Whitney had bold business aspirations related to following questions:

Goal: "Where are we going?"
Path: "How are we going to get there?"
Assurance: "Why are we going to be successful?"

It is important to note that the goals of the deployment were so bold in terms of operational agility and efficiency that it was clear to them that the underlying cost structure was the target and that the execution model was the solution; turn-backs and non-value-added activities and compensating behaviors were the enemies of success. Focusing solely on product line engineering (PLe) was necessary, but not sufficient to achieve the desired outcomes. Only by holistically addressing the initiative with a product line enterprise (PLE) approach could there be assurance of success.

Pratt & Whitney, to continue to be successful in one of their engine product lines, established a number of goals, one of which was to reduce variant non-recurring engineering cost by 75%.



For complex systems, such deployments are necessarily facilitated when all executional aspects are embedded in the authoritative system of record and the feature models are related to the product breakdown structure (PBS). The key activities that roll up into the work breakdown structure which includes all elements of the PBS as a subset, are detailed in the deliverable-driven integrated master plan (IMP). The IMP is resourced by the associated organizational breakdown structure with resources reflected in the integrated master schedule.

Critical for this enterprise execution success was the integration of the enterprise execution and governance, as well as the rationalization of all systems and item (software and hardware) governance. A particular effort was made on the validation of requirements and testing. In summary, a full model-based enterprise approach was necessary, bringing project data, resources, product data, requirements and tests all into one enterprise-wide, data-centric repository: the **3D**EXPERIENCE platform.

THE 3DEXPERIENCE PLATFORM OPENNESS

Because software needs to be managed in sync and in configuration with hardware to ensure global traceability and governance of the whole system, software related data and lifecycle data need to be accessible in the platform either directly or through a link. But software teams use very different tools across the whole lifecycle: requirement management, modeling, developing, building, testing, packaging and delivery processes are supported by different tools. These tools are numerous and they can change often just for serving specific tasks. The **3D**EXPERIENCE platform allows federating the product data authored in those tools in one data-centric referential. To do that, it complies with all the standards in the market and offers a set of data connection solutions.

Openness is multifaceted. There is not one solution, but as many as there are user stories. The platform allows many types of openness that we will detail in the rest of this paper. These types can be grouped into three classes:

- **Traceability** allows navigating from data inside the platform to data outside the platform and vice versa, to understand impact of changes, coverage of data by processes and collaboration around this information.
- **Data exchange** allows exchanging data between the platform and other external tools. It is based on standard interchange formats to ensure a correct semantic when exchanging. It entails duplication of data in different systems, which can lead to data discrepancy.
- **Data federation** is a means to share data across tools without duplication. It can be based on synchronization mechanisms or on linked data like OSLC. It can also be supported by real-time collaboration between tools through the use of web services.

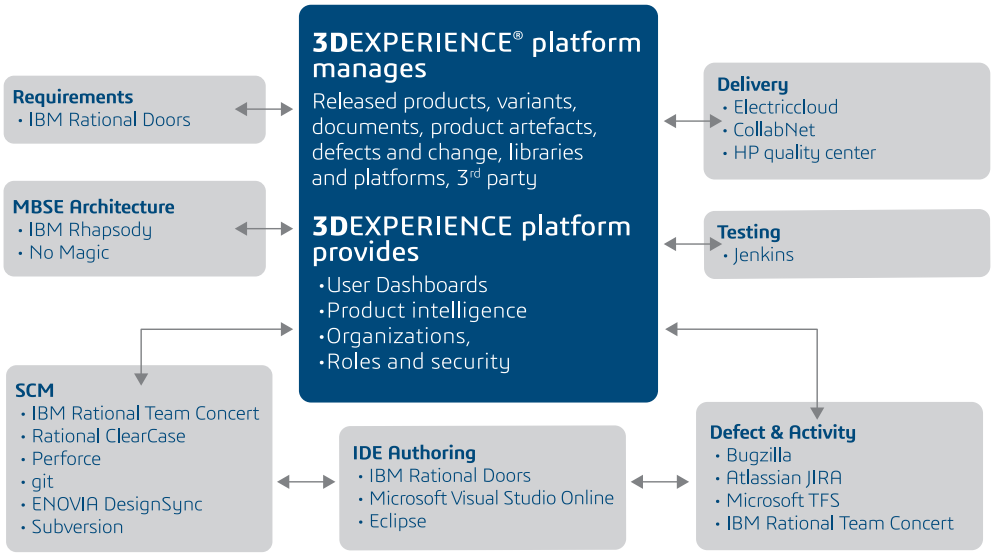


Figure 3
Openness for ALM

HERE IS THE LIST OF OPENNESS TOOLS AND STANDARDS DESCRIBED IN THE FOLLOWING PAGES.

	FUNCTIONS	TOOLS	STANDARDS
Traceability	Visualization of links between objects of different systems, and navigation through those links	CATIA System Navigation and Traceability (TRA)	n/a
Data Exchange	Based on standard formats, allows exchanging data between systems	n/a	ReqIF, FMI, AUTOSAR
Data Federation	Sharing data between systems by synchronization or/and linked data	ENOVIA DesignSync (BLD)	OSLC, RESTful

Table 1
List of discussed tools and standards

THE AUTOMOTIVE CASE: ADOPTION OF DATA EXCHANGE AND FEDERATION STANDARDS

Prominent standards such as ReqIF, AUTOSAR, and FMI that have been adopted at a worldwide level are now fully supported for both export and import by the **3D**EXPERIENCE platform. These standards were mainly driven by the needs of automotive industry OEMs and suppliers to address growing complexity of processes dominated by embedded systems software.

The automobile industry knows more than most how expensive the inconsistent handling of requirements between manufacturers and suppliers can be. For many years, requirements have been gathered and managed in dedicated requirement tools. When it comes to requirement refinement and integration with business, software and system architecture, different requirements interchange formats have been used, which were not following any standard. This is why the German automotive industry started the open, non-proprietary format for requirements exchange development: ReqIF, an international standard for requirement data exchange. Standardized by the Object Management Group (OMG), it today has solid recognition and adoption in the industry.

REQIF

ReqIF contains an XML data model that permits automated transfer of specifications, including graphics and tables. With ReqIF, an exchange file is exported from a RM tool that can be imported and understood by another system. In addition, it includes clear rules for the description and identification of the data. The **3D**EXPERIENCE platform can export and import requirements using ReqIF.

FMI

Functional Mock-up Interface (FMI) is a tool independent standard to support both model exchange and co-simulation of dynamic models. Daimler AG and Dassault Systèmes initiated FMI development in 2008 through an ITEA2 European Project called MODELISAR, with the goal to significantly improve the design of systems and embedded software in vehicles. Its main purpose is to improve the exchange of simulation models between suppliers and OEMs, building a strong foundation for the collaborative development of complex mechatronic systems. It also supports the internal and cross-enterprise creation of system level simulation capabilities through model exchange and/or co-simulation techniques that are independent of individual models authoring tools.

The first version, FMI 1.0, was published in 2010, followed by FMI 2.0 in July 2014.

As of today, development of the standard continues with participation of 16 companies and research institutes, including Dassault Systèmes, which is supporting all versions of the standard. Today, FMI is supported by over 101 tools and is used by automotive and non-automotive organizations throughout Europe, Asia and North America. Its scope is expanding beyond automotive, embracing aerospace, industrial equipment, energy and smart systems at large, including smart buildings.

AUTOSAR

AUTOSAR (AUTomotive Open System ARchitecture) is a worldwide development partnership of vehicle manufacturers, suppliers and other companies from the electronics, semiconductor and software industry. It delivers an open and standardized automotive software architecture, jointly developed by its Core and Premium members, among which is Dassault Systèmes who supports all successive releases of the standard.

AUTOSAR has been introducing new software architecture and development methods, including a standardized layer between application software and hardware (ECU). AUTOSAR was launched in 2003 with the goals of:

- Definition of an open embedded software architecture
- Standardization of basic software (BSW) functionality of automotive ECUs
- Transferability of software
- Development of highly dependable systems
- Scalability to different vehicle and platform variants
- Collaboration between various partners
- Support of applicable automotive international standards and state-of-the-art technologies

In 2012, AUTOSAR extended its standard into non-automotive areas, including rail, agriculture and forestry machinery, construction machinery, compressors, pumps or power generators, and marine and military transportation vessels.

At the end of 2016, the AUTOSAR development partnership added a new standard to its line-up called "AUTOSAR Adaptive Platform" and based on POSIX operating systems. Among the game changers, or main drivers for new automotive software systems identified by AUTOSAR, are highly automated driving, Car-2-X applications, vehicle in the cloud and increased connectivity. Support required for highly automated driving systems includes dependable architecture with fail-safe operational systems, support for cross-domain platforms, support for high-performance microcontrollers and computing, and distributed and remote diagnostics.

The object-oriented approach of the AUTOSAR Adaptive Platform provides an optimal foundation for computing-intensive tasks with large amounts of data, e.g., algorithms for automated driving (ADAS sSystems). It offers flexible options for the in-vehicle ECU architecture, such as support for dynamically adding, migrating and/or removing software components to adapt the system at runtime.

In December 2016, Dassault Systèmes became a premium member in the AUTOSAR Adaptive Partnership and is continuing to support AUTOSAR in its "adaptive" evolution, with the objective to cover three main aspects of ECU and vehicle level software development process:

- Software authoring (supporting AUTOSAR Adaptive Standard Descriptions and Methods)
- Car level software integration on E/E topologies with lifecycle, configuration and variant management
- Simulation and validation (ECU level and E/E integration level)

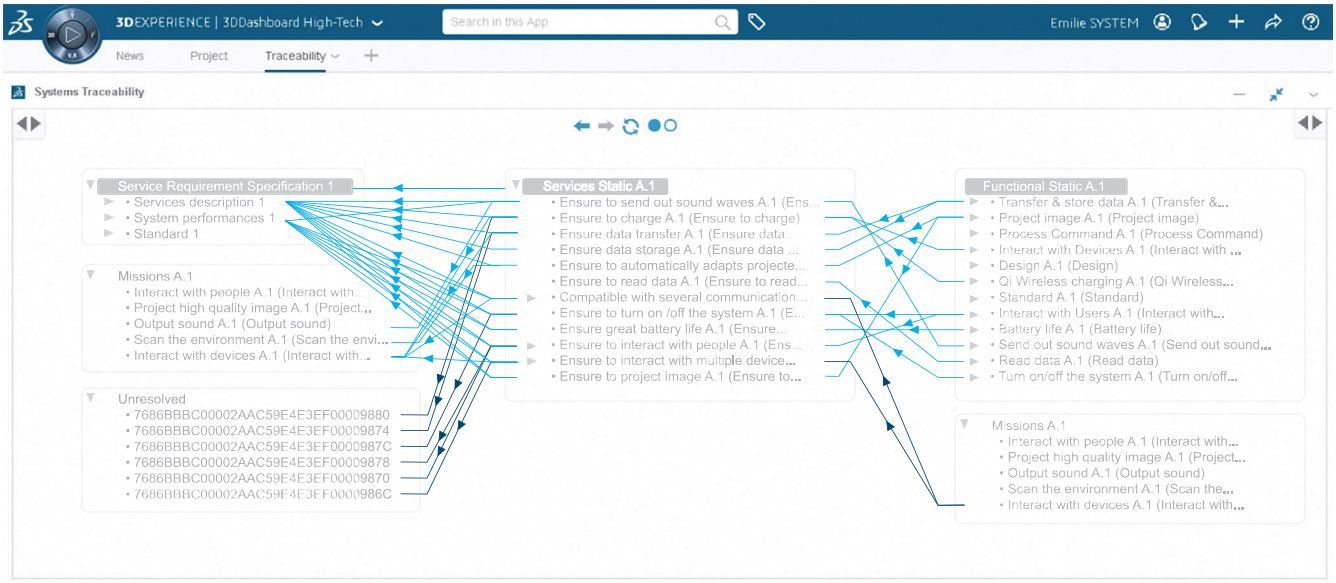
All these standards—ReqIF, FMI, AUTOSAR—are at the core of the **3DEXPERIENCE** platform to enable today's (automotive) and tomorrow's (many) industries to develop and simulate software intensive experiences. In particular for ADAS, as seen above but more generally, Dassault Systèmes will continue to integrate the latest standards to allow industries to master the hardware/software complexity and define, build, manage and experience multidiscipline products. Among those standards, Dassault Systèmes will build OSLC connectors, but also other means for Openness, as presented next.

SYSTEM TRACEABILITY

Before needing to exchange data, simply knowing what data in one system is linked to data in other systems (and if those links cover the whole scope necessary for the systems definition) is a basic need of systems and software engineering. System Navigation and Traceability (TRA) is a solution to ensure end-to-end traceability and collaboration on system and software models whatever the system behind them.

Whatever the tool, whether inside the **3DEXPERIENCE** platform, like Requirement Management or F/L Editor (Functional and Logical Systems Architecture Authoring), or outside the platform, like DOORS, Rhapsody or Matlab/Simulink, you can navigate the traceability link and understand the impact of changes.

Figure 4
System Navigation and Traceability on **3DEXPERIENCE**



For internal and external (supplier) traceability and system reviews among teams, TRA offers the unique possibility to understand, share, review and collaborate on one holistic system view, even with different tools and different organizations. Connectors will progressively be added and customers can easily add new connectors for home grown tools. In addition, system traceability supports OSLC-RM as a consumer and can therefore navigate to any OSLC compliant provider tool. (See the next section for details.)

APPLICATION GROUP	CONNECTOR NAME	DETAILS
Requirements	Doors 9X Doors NG ReqIF Reqtify	IBM Doors, based on DXL IBM DoorsNext generation, based on OSLC RM Format REQIF Reqtify
Modeling	Magic Draw Rhapsody Simulink SystemArchitect PTC Integrity Modeler PLCOpen	No Magic mdzip format IBM Rhapsody The Mathworks, Simulink Unicom System Architect Former Artisan studio now PTC PLCOpen
Software	C source file	Source files, .c.h
Testing	FMU	FMI, FMU format
Office	Images MSExcel MSPowerPoint MSWord Pdf Zip	.jpg, .png, .svg + structure .xlsx, .xls, .csv .pptx, .ppt .doc .pdf .zip
3DX Platform	CATIA EEW electrical Electronics architecture CATIA F/L ENOVIA change ENOVIA parameters ENOVIA product structure ENOVIA project ENOVIA requirement ENOVIA test case ENOVIA use case ENOVIA workspace	Functional, software, hardware and mapping Functional & logical models

Table 2
System Traceability Connectors

OPEN SERVICES FOR LIFECYCLE COLLABORATION (OSLC)

OSLC is an open community hosted by OASIS that defines specifications for integrating software engineering tools. OSLC is built upon internet standards like RDF, Linked data or RESTful web services. OSLC is the basis for integration between the IBM Jazz suite of tools, like DOORS or RTC. It has also been adopted by most of the players in the ALM world, such as JIRA for defect and activity management or HP QC for test management.

OSLC specifies a common tool protocol for creating, retrieving, updating and deleting (CRUD) lifecycle data, based on internet standards like HTTP and RDF using the Linked Data model. This protocol can be used by any tool or other programmatic client to talk to any other tool that implements the specifications. Linking is achieved by embedding the HTTP URL of one resource in the representation of another.

On top of the core definition of OSLC, which covers authentication mechanisms and allows declaring an OSLC provider and what it provides, domains are defined. The most mature domains today are Requirement Management and Change Management. The future domains will cover things like quality management to link verification data or architecture management for linking architecture components.

The core advantage of using OSLC for integrating domain-specific ALM tools with the **3D**EXPERIENCE platform is that with one connector you connect to all tools that are compliant and can provide or receive the shared data. Since the ALM world is changing very rapidly you usually will not connect to each new tool that appears on the market, but simply wait for that tool to offer the right connector.

Today, Dassault Systèmes uses OSLC to connect to tools from TRA. Dassault Systèmes is committed to embrace and extend OSLC coverage. Also, Dassault Systèmes is a member of the OSLC OASIS group and actively involved in the common definition of core services and PLM services for software providers.

SCM INTEGRATION USING ENOVIA DESIGNSYNC

The most disconnected area of the software product lifecycle is the development phase. Traditionally, software development happens in a separate environment with separate tools, managed and administrated by different teams. Any handoff or integration between teams happens through a disconnected process, with traceability only through system requirements allocated to software, and through black box delivery of the software build or executable in the final bill of materials for the product.

ENOVIA DesignSync (BLD) from Dassault Systemes' ENOVIA® brand brings together the software development environment and content metadata into the **3D**EXPERIENCE platform. The solution integrates Software Code Management (SCM) into the product lifecycle management tool chain and platform, providing an effective and integrated management of software assets throughout the (holistic) product lifecycle. Beyond providing traceability and impact intelligence from the highest design architecture to the lowest software leaf node, the SCM integration using ENOVIA DesignSync offers users the value of:

- Bringing SWLM/PLM value to existing code/repositories
- Federating multiple SCM repositories and their content for collaboration and integration
- Managing software in context of the entire system design, including semiconductor SCM

Integrating SCM with a holistic PLM solution and offering lifecycle and content intelligence are not trivial. Most companies have large amounts of software intellectual property (IP) code managed in existing Source Code Management repositories. Their development teams have often invested a lot in training and tooling for these existing systems. Migrating source code to a new system is a difficult and time consuming exercise. Retraining developers on new tools is costly and requires a long learning curve. Software developers are performing best in their existing environments and do not have time to change systems. Through leveraging the ENOVIA DesignSync solution, software development organizations can take advantage of SWLM benefits without the major investment associated with switching existing SCM systems.

The **3D**EXPERIENCE platform with ENOVIA software and SOC Designer (DesignSync/BLD) provides the ability to manage the entire software lifecycle while managing source code in different legacy repositories. Developers continue to work in their current development environments with their existing code management tools. The product team and product/company management gains the benefits of a complete managed lifecycle and accelerated development. Development teams are not disturbed, and there is no unnecessary additional cost of migration, retooling and retraining.

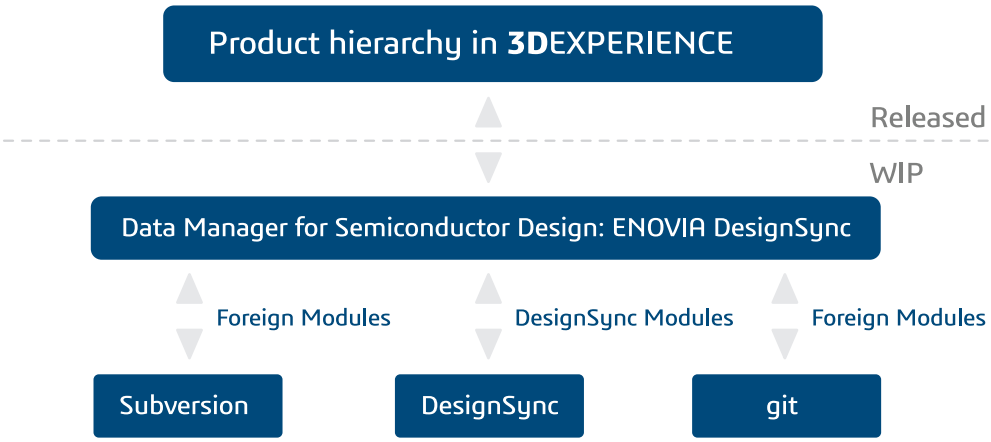


Figure 5
ENOVIA DesignSync federating SCMs

RESTFUL WEB SERVICES

Standard connectors and exchange formats facilitate the federation of tools and the data they manage. But in real life, it is often necessary to tailor those connections to the exact business process in a company. To do that, it is often necessary to model the actual workflow of data collaboration and implement it with direct integrations between tools. This is where a high level and light weight framework for integration based on web services is needed.

RESTful web services are becoming an integral part of the architecture of the **3D**EXPERIENCE platform, and the publication of REST APIs has been rapidly increasing. The REST service interface is more lightweight, well-suited and supported by modern programming languages and frameworks. Dassault Systèmes has embraced REST services and today they are a fundamental part of application development. As new applications are developed and enhanced, new REST services APIs are being developed, providing key interfaces and openness that enables the system to function in various, safe ways without requiring modification of the source code.

RESTful services are developed on top of the **3D**EXPERIENCE platform which provides the proper authentication, licensing and access control. The web services are executed in the context of an authenticated user and access to the data is based on the user's access rights. The REST services provide complete management of the underlying resources in terms of read and write, while ensuring data integrity and completeness. They are designed independently of the user experience apps they enable, allowing them to be used in various use cases and applications.

Today, many ENOVIA applications are based on REST services and as a result, many of the underlying modelers are supported by REST APIs. The support of REST services will continue to evolve and will be the basis for the openness provided by the ENOVIA brand from Dassault Systèmes.

CONCLUSION

The **3D**EXPERIENCE platform delivers complete end-to-end software product lifecycle management, federating data coming from diverse authoring tools. The platform manages the definition of all the products' Intellectual Property (IP) in configuration system, software, hardware (mechanical, electrical, electronic) and services. This key capability of designing software and hardware together in a model-based approach gives companies the flexibility to answer rapidly evolving, complex customer demand.

The system simulation capabilities allow verification and validation at all steps of the process, from model in the loop to hardware and software in the loop. The platform's openness is the basis for federating all data into one end-to-end experience modeling flow.

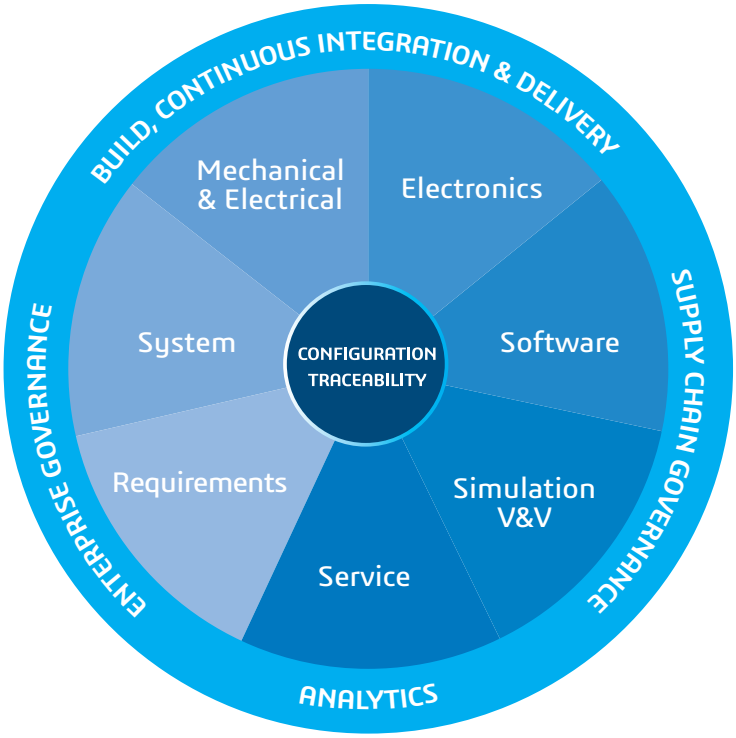


Figure 6
The **3D**EXPERIENCE platform to plan, build and test modular hardware and software products

REFERENCES

- OSLC – Open Standard for Lifecycle Collaboration: http://open-services.net/
- ISO26262 – Road Vehicles Functional Safety: https://www.iso.org/standard/54591.html
- ECCN – Export Control Classification Numbers: https://www.bis.doc.gov/index.php/licensing/commerce-control-list-classification/export-control-classification-number-eccn
- ReqIF – Requirements Interchange Format™: http://www.omg.org/spec/ReqIF/
- MODELISAR: https://itea3.org/project/modelisar.html
- FMI – Functional Mockup Interface: http://fmi-standard.org/
- FMI – Functional Mockup Interface: http://fmi-standard.org/
- Innovation through software – Olivier Ribet on creating the connected experience: https://www.youtube.com/watch?v=nt0XN-k7ER4

Our **3D**EXPERIENCE® platform powers our brand applications, serving 12 industries, and provides a rich portfolio of industry solution experiences.

Dassault Systèmes, the **3D**EXPERIENCE® Company, provides business and people with virtual universes to imagine sustainable innovations. Its world-leading solutions transform the way products are designed, produced, and supported. Dassault Systèmes' collaborative solutions foster social innovation, expanding possibilities for the virtual world to improve the real world. The group brings value to over 220,000 customers of all sizes in all industries in more than 140 countries. For more information, visit **www.3ds.com**.

3DEXPERIENCE®

DASSAULT SYSTEMES | The 3DEXPERIENCE® Company

Americas
Dassault Systèmes
175 Wyman Street
Waltham, Massachusetts
02451-1223
USA

Europe/Middle East/Africa
Dassault Systèmes
10, rue Marcel Dassault
CS 40501
78946 Vélizy-Villacoublay Cedex
France

Asia-Pacific
Dassault Systèmes K.K.
ThinkPark Tower
2-1-1 Osaki, Shinagawa-ku,
Tokyo 141-6020
Japan