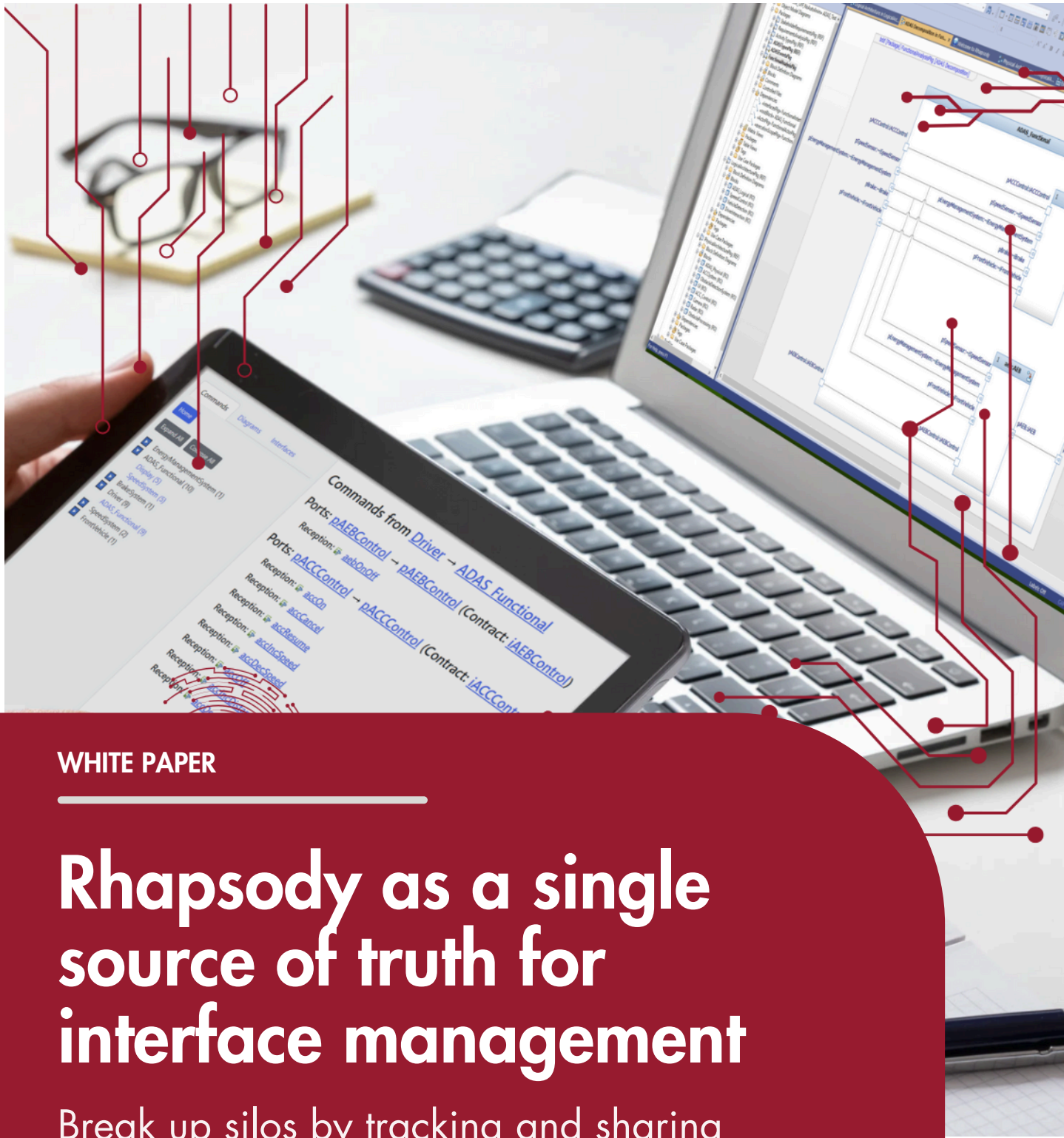
SODIUS WILLERT



WHITE PAPER

Rhapsody as a single source of truth for interface management

Break up silos by tracking and sharing interface specifications with Power Pack

JUNE 2025

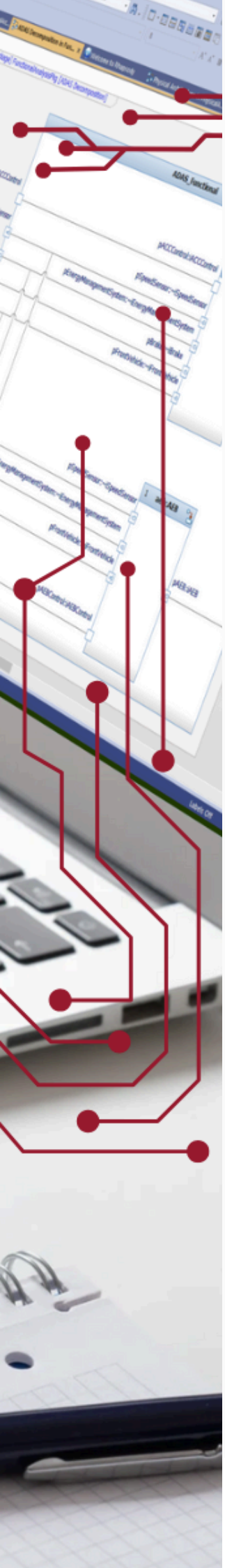


Table of **content**

Introduction

Product development is becoming increasingly complex. A big driver is the addition of software, combined with mechanics, electronics, resulting in tightly integrated products. As the number of disciplines and dependencies grows, so does the potential for miscommunication and misalignment between teams. This complexity often **overwhelms traditional development methods**.

Problems tend to surface late in the process, **during system integration**. By that point, timelines are tight, and every delay is costly. Integration issues are rarely isolated failures. They usually stem from unclear expectations at the boundaries between components.

When teams are not aligned on how systems connect and interact, even small inconsistencies can trigger **significant setbacks**.

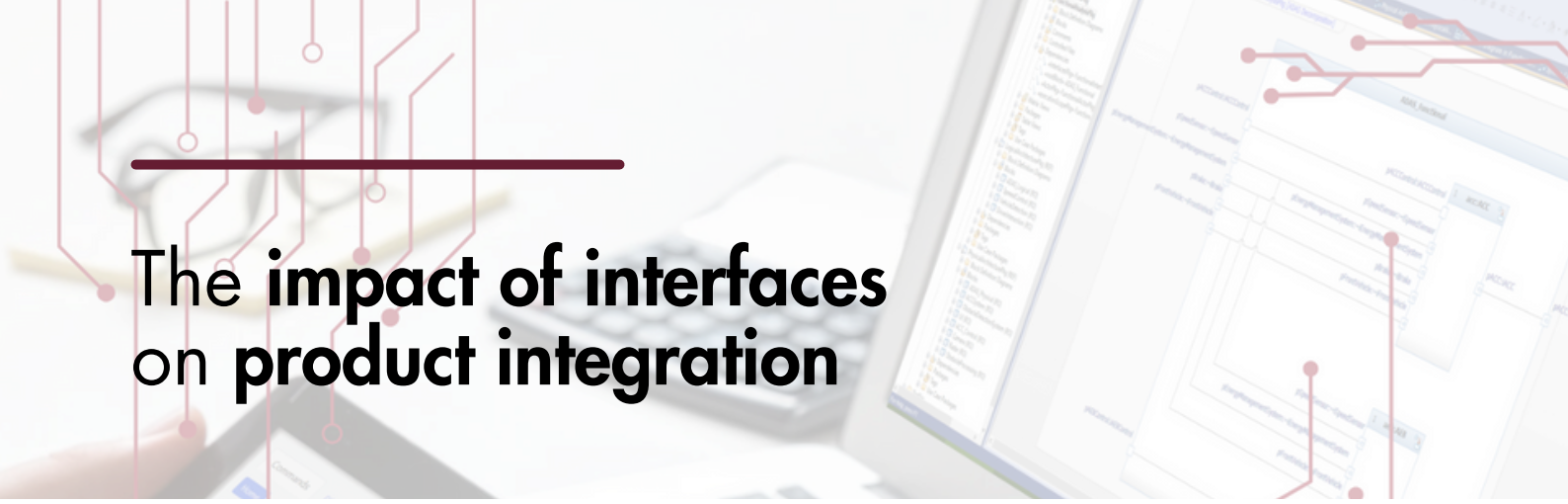
One of the most effective ways to address this challenge is to take control of your interfaces. This whitepaper presents a case study that demonstrates how clear, consistent, and traceable **interface specifications** can streamline integration, improve collaboration, and reduce costly surprises. It outlines an approach that turns interfaces into strategic assets, helping teams build better systems with fewer delays and greater confidence.

Definition: Interface
An interface is the point at which two or more logical, physical, or both, system elements or software system elements meet and act on or communicate with each other.

ISO/IEC/IEEE 24748-6, Section 3.1.3

"Interface specifications constitute a critical tool in guaranteeing that intricate systems are precisely defined, appropriately executed and effectively maintained throughout their entire life cycle."

Ricarda Schüssler, Calibration Engineer at Bosch, *2023 in "Approaching Smart Interface Specifications in the Systems of Systems Context"*



The **impact of interfaces** on **product integration**

Well-managed interfaces are the foundation of successful system integration. In modern product development, where mechanics, electronics, and software must work seamlessly together, even small inconsistencies at an interface can result in significant delays, rework, or outright failure.

Despite advances in development tools and practices, many projects still encounter critical issues during integration, often when it is too late to fix them efficiently. Understanding the role interfaces play is essential to avoiding these outcomes.

Interfaces as the Point of Convergence

Studies and industry experience consistently show that integration is one of the riskiest and most expensive phases of development. Fixing integration issues is more costly than addressing them during design.

Moreover, poor interface management:

- Introduces unnecessary dependencies
- Undermines parallel development
- Leads to finger-pointing when components fail to work together as expected

Types of Interfaces

There is no agreed upon List of interface types. The NASA handbook for instance, states: "*Interfaces may include mechanical, electrical, human user/operator, fluid, radio frequency, data, or other types of interactions.*"

Case study: GM's Missed Opportunity
In the early 2010s, GM intended to launch its semi-autonomous driving system, Super Cruise, on the Cadillac CT6. This system combined adaptive cruise control with lane-centering technology.

However, according to this McKinsey Study, the rollout faced major integration setbacks. Complex interactions between sensors, control logic, and driver monitoring systems led to years of delays.

These challenges were not just technical. They stemmed from misaligned expectations, inconsistent timing, and lack of coordination across teams. The missed launch window allowed competitors to gain ground and forced GM to defer revenue.

Better interface specification could have prevented these issues by aligning teams around shared, reliable information.

Interface **precision with SysML and MBSE**

Mature organizations adopt Model-Based Systems Engineering (MBSE) to manage complexity, ensure consistency, and improve collaboration across disciplines.

SysML

The most widely used notation for this purpose is the Systems Modeling Language (SysML), often applied in tools like IBM Engineering Rhapsody®. When used right, SysML becomes a powerful enabler of interface reuse, automation, and traceability throughout the development lifecycle.

Relevant SysML elements

- **Block:** A modular unit that represents a system, subsystem, or component.
- **Proxy Port:** Used to expose the features of internal parts without revealing internal structure, thereby representing an interface.
- **Interface Block:** Formalizes the definition of an interface by specifying a set of features (operations, signals, or flow properties) that can be provided or required by a port. (Note: No interface block is visible in Fig. 1.)

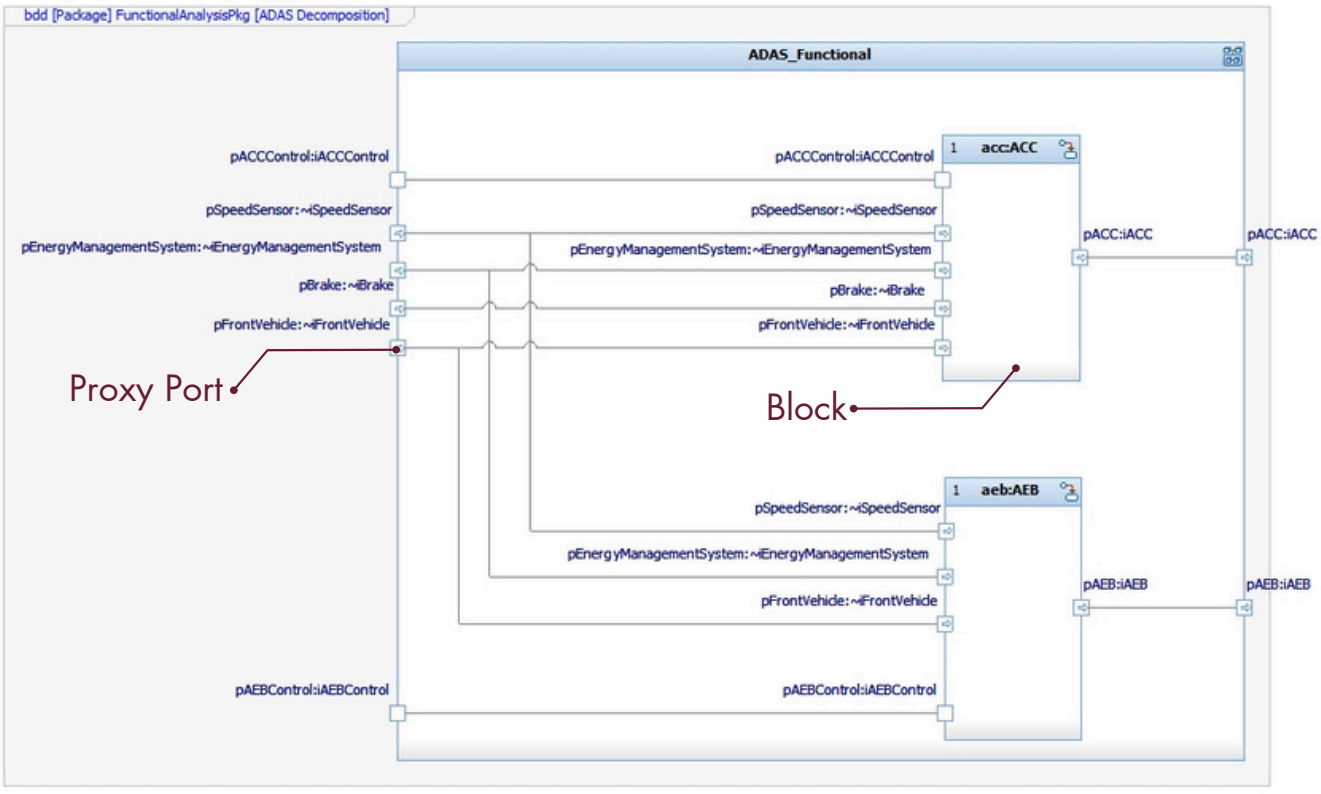


Fig 1. This internal block diagram shows how the model connects ADAS subsystems (ACC and AEB) and exposes interfaces. The model includes precise port definitions that enable automatic interface alignment, even though they are not fully visible in the diagram.

Benefits of **SysML interface models**

Traditionally, interfaces are captured in Interface Control Documents (ICDs). Formalizing them with SysML offers many benefits over ICDs. SysML interfaces allow engineers to define, analyze, and manage systems using consistent, integrated models. This approach improves alignment across disciplines, increases efficiency, and acts as an enabler for many key use cases, like reuse.

Industries and Standards

MBSE with SysML is widely adopted across industries where system complexity and safety are critical. These include:

- **Automotive:** MBSE supports standards like ISO 26262 (functional safety), AUTOSAR (architecture standard), and ASPICE (software process improvement).
- **Aerospace and Defense:** MBSE aligns with DO-178C, ARP4754A, and MIL-STD-499C, enabling systematic verification, interface control, and design traceability for highly regulated airborne systems.
- **Rail:** MBSE supports standards such as EN 50126/8/9, ensuring system-level safety, maintainability, and consistent validation.
- **Medical Devices:** MBSE facilitates compliance with IEC 62304 and ISO 14971, providing structured risk management and traceability.

Reuse & Variants

MBSE promotes reuse of validated system elements, such as architecture patterns, interfaces, and requirements. This allows:

- Faster creation of **new variants**
- Foundation for **product lines** and **platforms**
- Higher **consistency** and reduced **risk** due to simplified **traceability**
- Generally a **higher rate of reuse**

Automation

MBSE enables automation that improves quality and efficiency across the development lifecycle:

- **Consistency checks** identify mismatches between interface definitions and implementations.
- **Model checking** ensures structural correctness and supports formal verification of logical flows.
- **Test case generation** can be automated from behavioral models, helping ensure test coverage aligns with requirements and interface contracts.

These automated capabilities are difficult to achieve with traditional document-based processes and contribute directly to project predictability and product quality.

Making **interfaces available to all stakeholders**

SodiusWillert provides solutions that make interface information from the model accessible to all stakeholders, even those who do not work directly in Rhapsody.

Why they ignore the model

Many stakeholders avoid using the model, which reinforces silos and leads to confusion and rework. This happens for several reasons:

- **Lack of modeling experience:** Not all roles are trained in SysML or system modeling tools.
- **Different company or organization:** External partners or suppliers may not have access to the modeling environment.
- **Tooling limitations:** Licensing, IT policies, or tool incompatibilities can prevent easy access.
- **Time constraints:** Stakeholders may not have the bandwidth to learn or navigate the model.
- **Cultural resistance:** Some teams prefer traditional document-based workflows and resist adopting new practices.
- **Perceived irrelevance:** Roles like procurement, compliance, or test engineering may not see immediate value in working with the model.

Stakeholder of interface descriptions:

- **Requirements Engineering:** Needs interface clarity to derive and validate requirements, especially across system boundaries.
- **Verification and Validation:** Uses interface specs to define test strategies, ensure coverage, and plan regression testing.
- **Configuration Management:** Relies on version-controlled interface definitions to track compatibility across system variants.
- **Engineering Disciplines:** Includes software, mechanical, and electrical teams who design or consume interfaces.
- **Project and Product Management:** Seeks architectural transparency to manage scope, risks, and delivery timelines.
- **Roadmap and Portfolio Planning:** Uses interface maturity to plan reuse and align future product generations.
- **Suppliers and External Partners:** Require accurate interface specs to build compatible and contract-compliant components.
- **Marketing:** Leverages interface information to position modularity and integration capabilities.
- **Executive Management:** Expects interfaces to support reuse, reduce risk, and accelerate delivery.
- **Safety and Compliance Authorities:** Depend on clear, traceable interfaces to assess safety and regulatory conformity.

Power Pack **provides options**

Power Pack is a suite of productivity tools designed to enhance model-based systems engineering in Rhapsody. It includes features for interface sharing and interface management that will make them accessible to all stakeholders.

ICD generator

The ICD is part of the SodiusWillert Power Pack for IBM Rhapsody®. It shows up in the context menu of the model tree if the selected element supports it, e.g. a package.

This functionality requires PUB to be installed, which can be used to adapt the template to your needs.

IBM Publishing

IBM Engineering Lifecycle Optimization - Publishing® (PUB) is an automated document-style report generation solution that generates document-style reports from IBM products like, but not limited to, Rhapsody.

PUB supports multiple formats like Word, Excel, HTML, and PDF, and can combine data from different tools into a single report.

You need a license for PUB to use this feature from Power Pack.

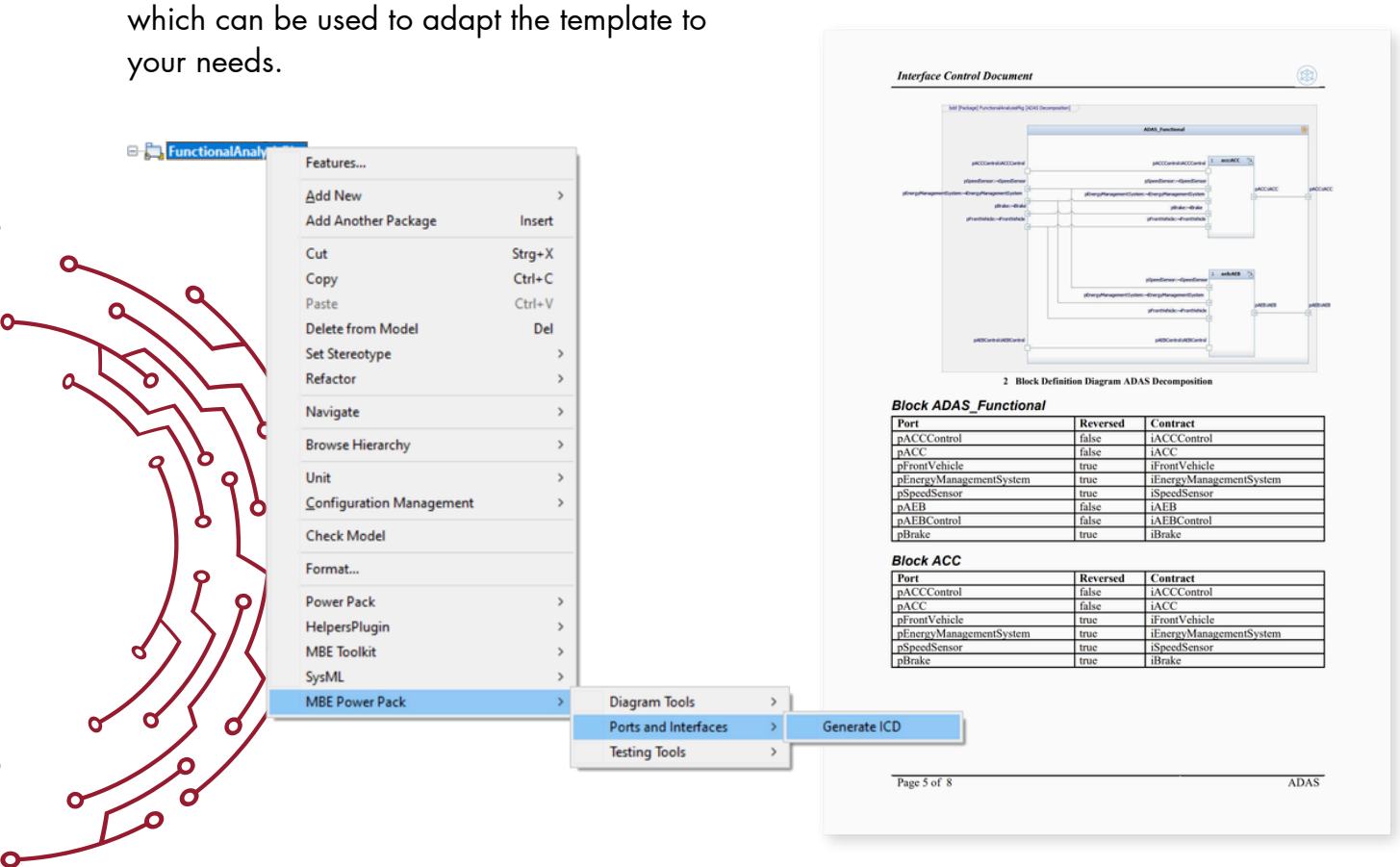


Fig 2. ***Left:*** *With a right-click on a Package, Power Pack allows you to generate an ICD for the given context.* ***Right:*** *The generated document can be shared read-only and provides an easy-to-read view of the interface description.*

Power Pack gives you **even more** options

In addition to the PUB-based solution from the previous page, Power Pack also has an HTML-generator that provides an interactive UI to stakeholders that can be published to the intranet.

Interactive HTML

An even more powerful option is the interactive HTML export. Unlike the document export, it allows stakeholders to explore model content directly in their browser, without needing access to Rhapsody or using PUB.

This solution collects, collates and presents interface data into a consumable, navigable format that simply isn't possible using a document generation tool like PUB. This feature is currently in beta, and we invite you to contact us for more information.

One model, many views

At its core, the system model remains the single source of truth. An Interface Control Document is not a standalone artifact, but one of many tailored views generated from that model. When treated this way, interface specifications become dynamic tools that evolve with the system, not just static documents created for compliance.

Power Pack

Power Pack extends Rhapsody with powerful capabilities that go beyond interface publishing. It includes tools for model validation, automated checks, advanced model navigation, scripting support and more. This makes Power Pack a valuable asset for teams looking to enhance efficiency, ensure consistency, and scale their MBSE practices.

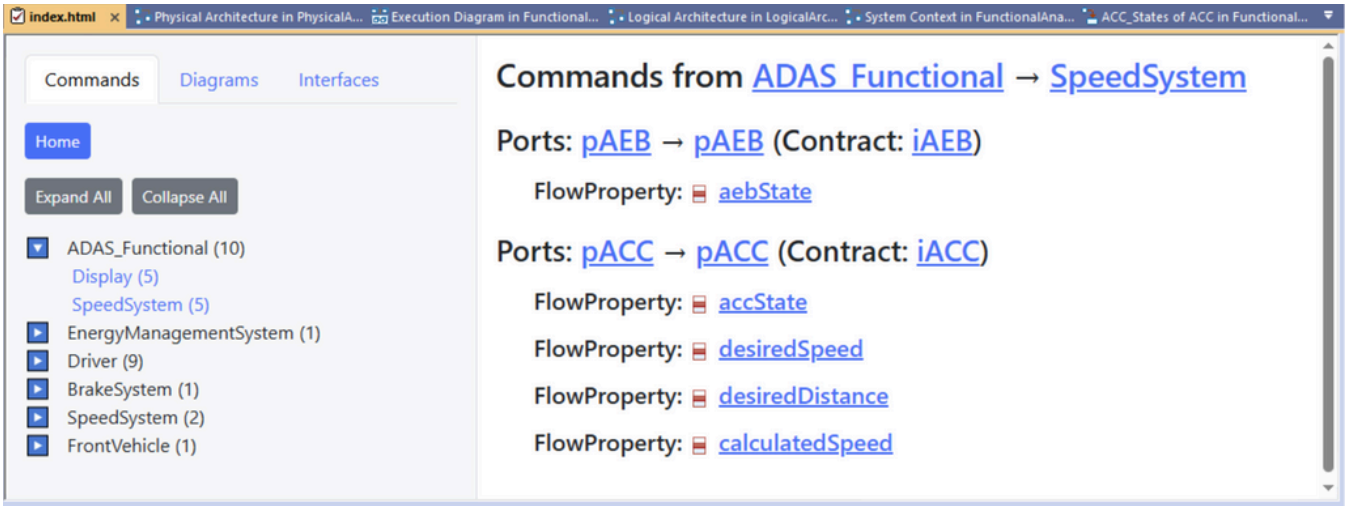


Fig 3. The interactive Power Pack HTML ICD export can be opened inside Rhapsody or in a regular web browser. As it is simple HTML, it can be published to an intranet or even sent to a partner in a ZIP file. The export also includes an interactive diagram view and can open elements in Rhapsody (if installed).

Conclusion

Effective interface management is critical because unclear or outdated interfaces remain a major source of integration problems, delays, and miscommunication in complex systems.

Modeling offers a powerful solution by providing a precise, consistent foundation for defining and maintaining interfaces throughout the development lifecycle. With Power Pack, organizations can bridge the gap between modeling tools and the wider stakeholder community, making interface data accessible and actionable.

This is not the final destination, but the beginning of a broader transformation toward model-driven collaboration and smarter systems engineering.

Benefits of better Interface Management

Managing interfaces in documents leads to errors and misalignment. Modeling solves this, but often excludes key stakeholders. Generating PDF or HTML views bridges the gap, combining the precision of models with the accessibility of documents.

Strategic interface management delivers measurable improvements:

- Fewer integration surprises
- Improved cross-team collaboration
- Reduced documentation overhead
- Better traceability and audit support
- Clearer expectations between suppliers and OEMs
- Easier compliance with standards such as ISO 26262 and ASPICE
- Increased confidence in reuse and modular design

Where to go from here

Improving interface management is a strategic move that boosts quality and speeds up integration. It takes clear leadership and active team engagement.

To move forward:

- Define a **long-term strategy** that supports reuse and consistency
- Provide simple, **accessible outputs** for non-modeling stakeholders
- Encourage adoption by **showing value early and often**
- Align interface data with **change and configuration management**
- Use interface specs to **drive testing and validation**

Curious? Then contact SodiusWillert to try Power Pack for better interface management.

Visit www.sodiuswillert.com to find out how SodiusWillert accelerates the development of complex systems and software.

Willert Software Tools GmbH
Hannoversche Str. 21, 31675 Bückeburg, Germany
www.sodiuswillert.com