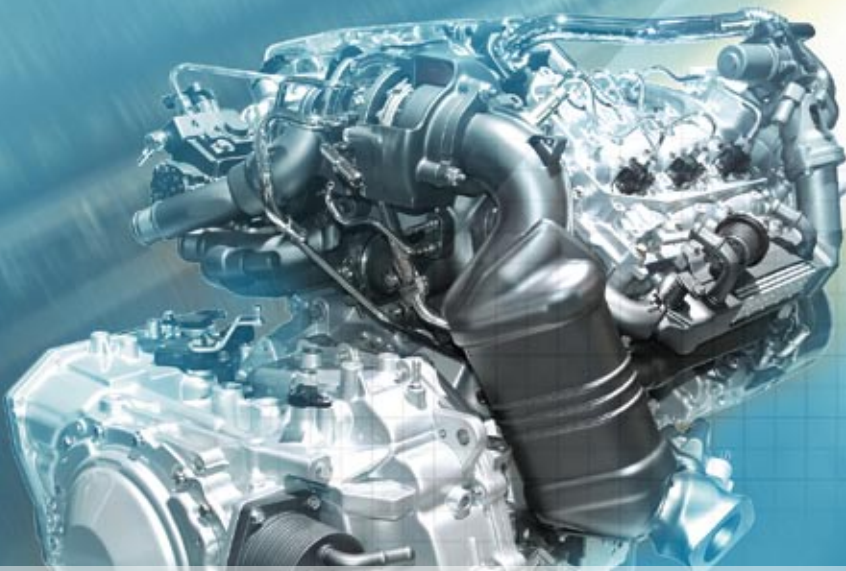


Motorsteuerung alla AUTOSAR

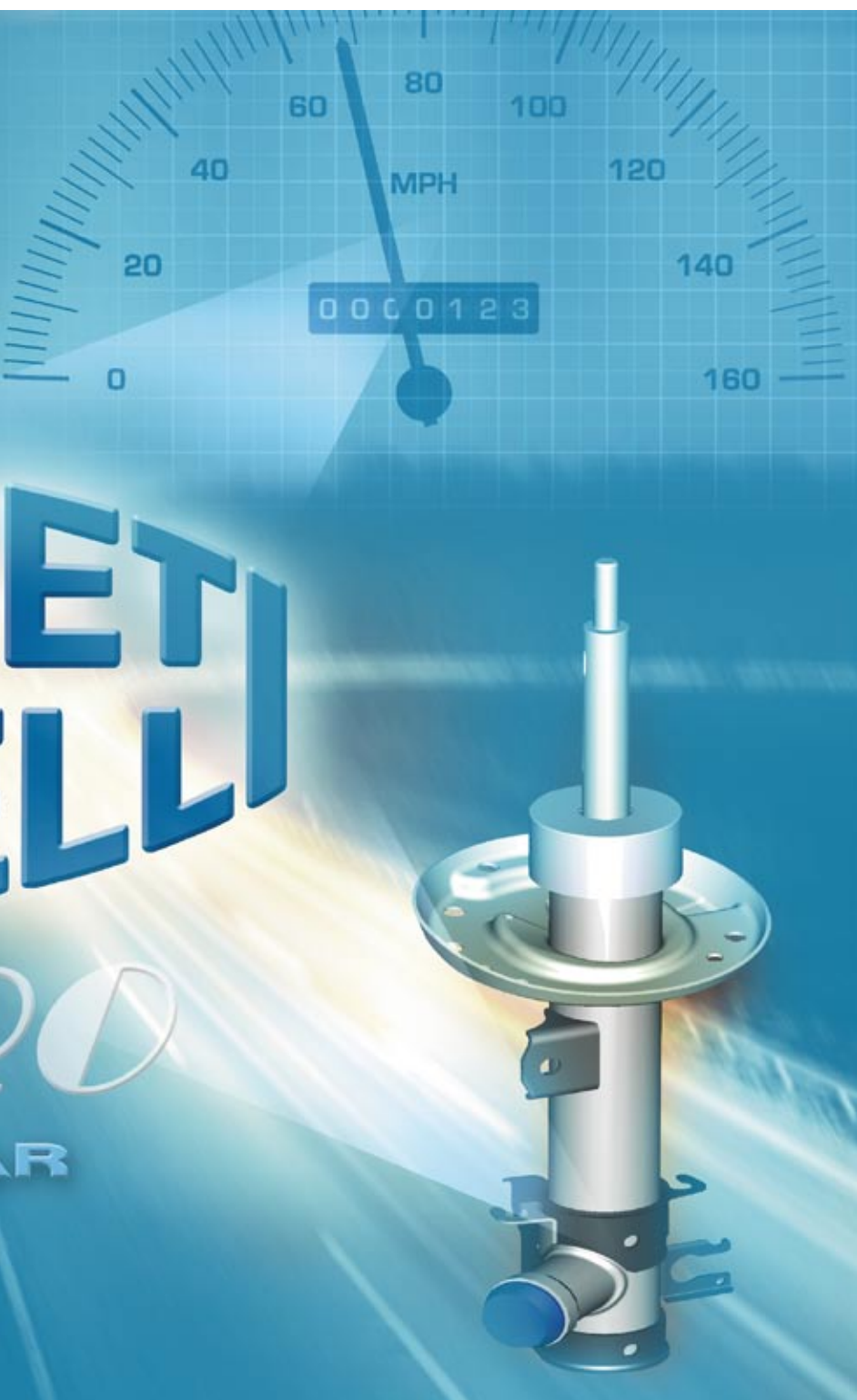


MAGNETI MARELLI

X-PPC AUTOSAR



Magneti Marelli migriert Steuergeräte-Software in den AUTOSAR-Standard



AUTOSAR gehört zu den meistdiskutierten Themen in den Entwicklungsabteilungen der Zulieferer und Fahrzeughersteller. Doch häufig wird dem Standard ein zu großer Overhead bei der Implementierung komplexer Systeme unterstellt.

Magneti Marelli will am Beispiel eines Motorsteuergeräts den Gegenbeweis antreten.

Im Rahmen des Sonderprojekts „Magnet Marelli AUTOSAR Cross-Project X-PRO“ entwickeln wir im Bereich Powertrain einen Demonstrator, um die Machbarkeit AUTOSAR-konformer Softwarearchitekturen selbst für Motorsteuersysteme nachzuweisen. Die konkrete Aufgabe besteht darin, die Software eines bereits existenten Motorsteuergeräts vollständig nach AUTOSAR zu migrieren und wiederum auf demselben Steuergerät zu implementieren. Die Softwareumfänge der Motorsteuerungen sind üblicherweise enorm; umso wichtiger ist es, Module einfach wiederverwenden zu können. Deswegen

interessieren wir uns für die Software-Entwicklung alla AUTOSAR.

In Zusammenarbeit mit dSPACE haben wir uns für eine Vorgehensweise entschieden, bei der die Automatisierungsfunktionalität der Systemarchitektursoftware SystemDesk eine zentrale Rolle spielt. Da es bei diesem Demonstrator-Projekt um Machbarkeit und Erkenntnisse über die Performanz und den Ressourcenbedarf geht, haben wir nicht Top-down eine neue Architektur entworfen, sondern uns eng an den Aufbau der vorhandenen Steuergeräte-Software angelehnt. Um dies umzusetzen, wurden beispiels-

weise Informationen, die zur Rekonstruktion der Softwarearchitektur und des Scheduling wichtig waren, aus den existierenden Steuergerätedaten extrahiert und per Skript nach SystemDesk transferiert (Abbildung 1). Quasi automatisiert erhält man so eine AUTOSAR-konforme Architektur, für die nur noch wenige manuelle Anpassungen in SystemDesk erforderlich sind.

Daten und Informationen aufbereiten

Da es vor AUTOSAR noch keinen Standard zur Beschreibung aller Implementierungsinformationen

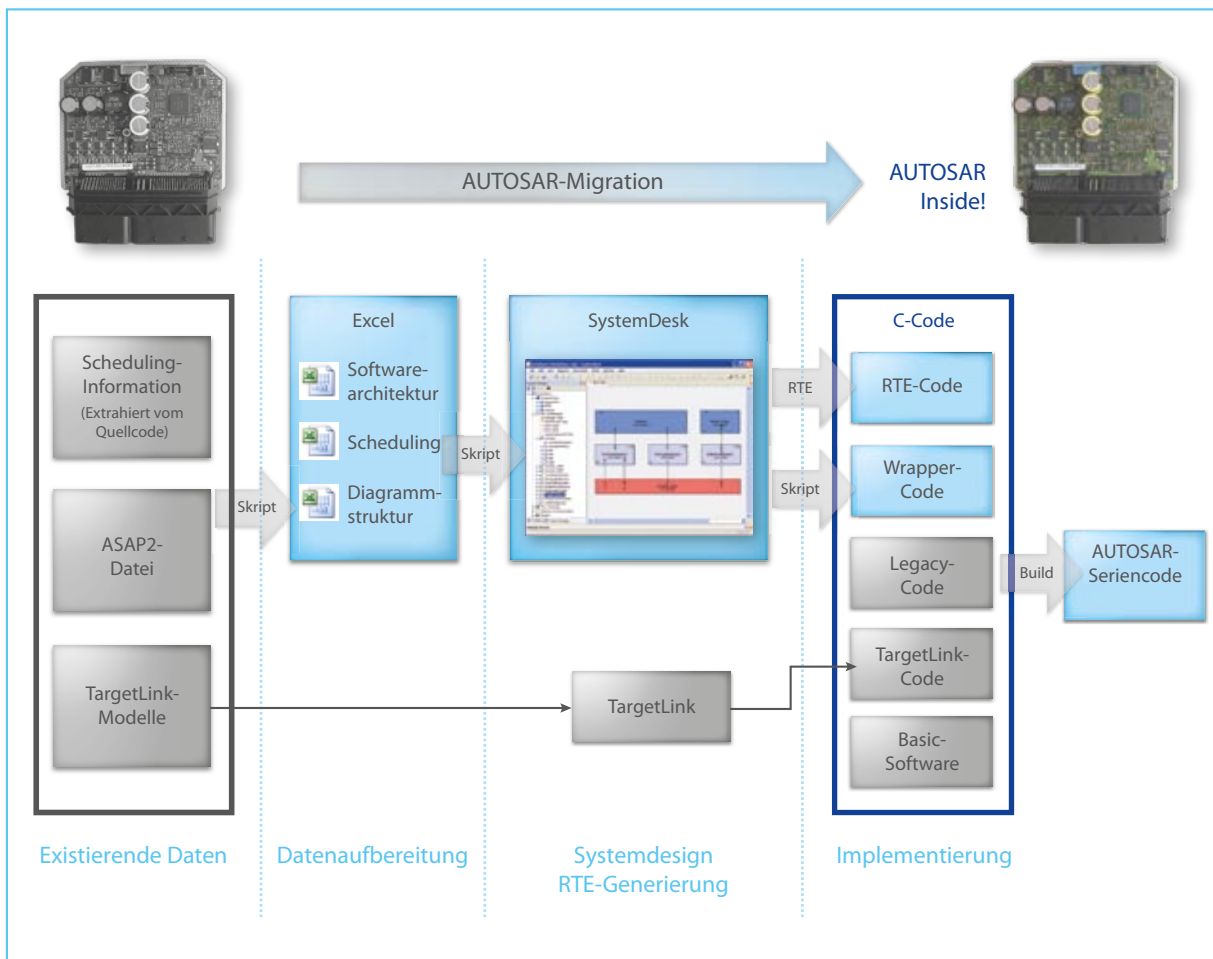


Abbildung 1: Gesamtworkflow für die AUTOSAR-Migration.

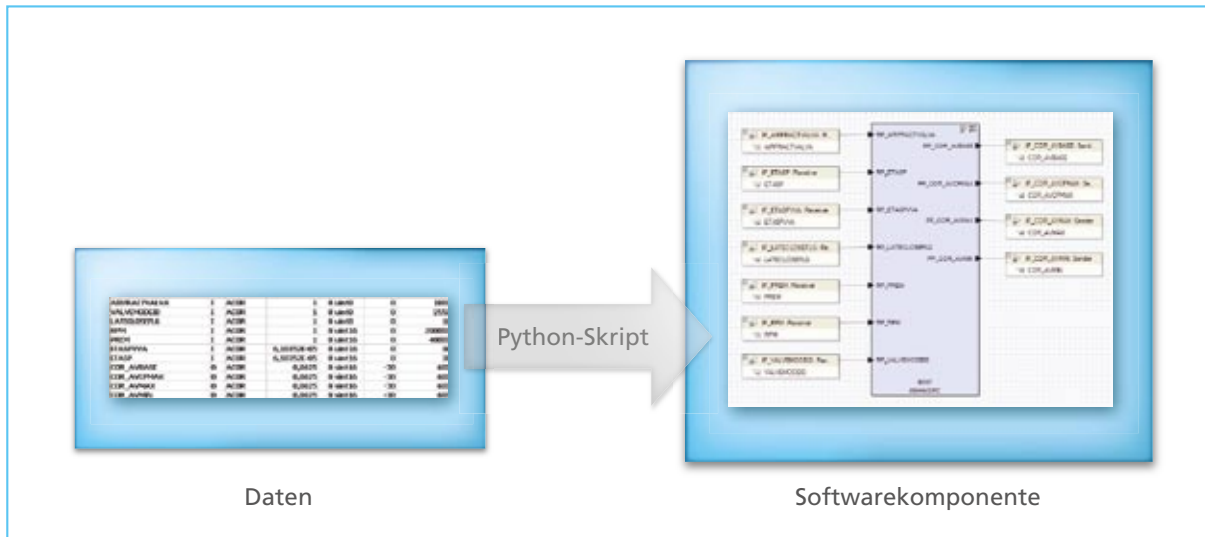


Abbildung 2: Gegenüberstellung aller Variablen einer Softwarekomponente in Excel (links) und der entsprechenden Softwarekomponente in SystemDesk (rechts).

gab, ist eine systematische Wiederverwendung der einzelnen Softwaremodule nicht möglich. Unsere erste Aufgabe bestand daher darin, die Konfigurations-, Parametrierungs- und Implementierungsinformationen des existierenden Steuergeräts aus den verschiedensten Datenquellen zu sammeln und für die weitere Verarbeitung zentral zu archivieren. Beispielsweise wurden die Architekturinformationen skriptbasiert aus den vorhandenen TargetLink-Funktionsmodellen extrahiert. Beschreibungsdateien wie die ASAP2-Datei (Abbildung 1) dienten zusätzlich dazu, die Datentypen, Min/Max-Werte und Skalierungen der implementierten Variablen zu gewinnen. Da auch die Konfiguration des Betriebssystems des alten Steuergeräts übernommen werden sollte, wurde das gesamte Funktions-Scheduling aus den alten Source-Dateien per Skript extrahiert und nach Excel übertragen. Mit Hilfe solcher Standards wie AUTOSAR wird es natürlich in Zukunft viel einfacher werden, bereits existierende Softwaremodule und Implementierungsinformationen wiederzuverwerten.

„Die Zugriffe auf Variablen über die mit SystemDesk generierte RTE erfolgen mit der gleichen Performanz wie in der klassischen Implementierung.“

Luigi Romagnoli, Magneti Marelli

Aufbau der Softwarearchitektur

Die in einer Excel-Tabelle erfassten einzelnen Daten müssen nun in die Systemwelt migriert werden, um eine AUTOSAR-konforme Gesamtarchitektur zu erhalten. Mit Hilfe eines Python-Skripts wurden über die Automatisierungsschnittstelle von SystemDesk alle benötigten Softwarekomponenten inklusive Sender-/Receiver-Interfaces sowie ihrer Datenelemente voll automatisiert angelegt (Abbildung 2). Dass bei einem Importvolumen von über 20.000 Elementen auch Fehler passieren können, lässt sich nicht vermeiden. SystemDesk ist jedoch in der Lage, alle Inkonsistenzen in den Eingabedaten zu erkennen und anzuzeigen. Passt beispielsweise ein Datentyp nicht mit den angegebene-

nen Skalierungsinformationen und Min/Max-Werten zusammen, wird schon beim Anlegen der Softwarearchitektur auf den Fehler hingewiesen. Beim Import der Variablen sind wir nach einem Namensschema vorgegangen und konnten daher alle Verknüpfungen in der Softwarearchitektur automatisiert aufbauen. Dies bedeutet bei über 170 Softwarekomponenten und mehreren Tausend Ein- und Ausgangssignalen eine nicht zu unterschätzende Zeitersparnis. Aufgrund der riesigen Datenmengen ist die komplette Architektur nicht mehr sinnvoll in einem Diagramm darstellbar. In SystemDesk lassen sich daher Teilfunktionalitäten in verschiedenen Kompositionsdigrammen so zusammenfassen,

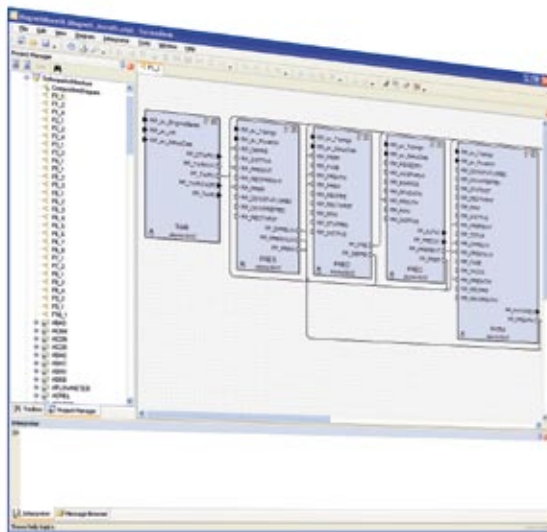


Abbildung 3: Teilaspekt des Gesamtmodells in einem einzelnen Kompositionsdiagramm dargestellt.

„Die Automatisierungsfunktionalität von SystemDesk beschleunigt die Prozesse nicht nur, sondern stellt auch die Integrität der Daten sicher.“

Alessandro Palma, Magneti Marelli

dass unterschiedliche Sichten auf die Softwarearchitektur möglich sind. So behält man auch in großen Projekten immer den Überblick und kann teamübergreifende Aspekte zielorientiert besprechen.

Import des OS-Schedulings

Das alte Betriebssystem-Scheduling sollte auch für das AUTOSAR-Steuerggerät verwendet werden. Nachdem einmal tabellarisch festgelegt wurde, welche Runnables zu welcher Softwarekomponente gehören, musste nur noch die Reihenfolge in den jeweiligen Tasks des

Betriebssystems spezifiziert werden. Danach ließ sich das Scheduling vollständig automatisiert nach SystemDesk übertragen.

Grafische Komplexitätsreduktion

Motorsteuergeräte gehören zu den komplexesten Steuergeräten im Automobil. Um die 10.000 Variablen, 600 Runnables und 16 Tasks müssen definiert und optimal verwaltet werden. Jede Art der Komplexitätsreduktion hilft dabei im Entwicklungsprozess, Fehler zu vermeiden und schneller ans Ziel zu

kommen. SystemDesk bietet neben der bereits erwähnten Möglichkeit, verschiedene Teilaspekte der Softwarearchitektur durch einzelne Kompositionsdiagramme anzuzeigen (Abbildung 3), noch viele weitere Optionen, um die Komplexität zu reduzieren. Beispielsweise lassen sich durch nur drei Klicks alle (verbundenen) Ports einer Softwarekomponente verstecken oder alle noch nicht verbundenen Ports der kompletten Softwarearchitektur in einem Dialog anzeigen.

Was uns besonders bei der Arbeit mit der grafischen Bedienoberfläche von SystemDesk gefallen hat, ist die Möglichkeit, komplexe Systeme auf einfache Darstellungen zu reduzieren. Nur so gelingt es, Reviews erfolgreich durchzuführen und neue Ideen schnell am Bildschirm zu erläutern.

RTE-Generierung

Ein wichtiges Zwischenziel bei der Migration des existierenden Codes in den AUTOSAR-Standard ist eine RTE-Generierung der kompletten Softwarearchitektur unter Berücksichtigung des OS-Schedulings. Die RTE (Runtime Environment) ist eine schlanke Middleware, die softwaretechnisch die Kommunikation zwischen den einzelnen AUTOSAR-Softwarekomponenten umsetzt. Gegenüber der bisherigen Implementierung unserer Software benötigt diese Zwischenschicht unter Umständen weitere Ressourcen. Daher war es uns wichtig, den Speicher- und Laufzeitbedarf des von SystemDesk generierten RTE-Codes abzuschätzen. AUTOSAR-kompatiblen Applikationscode oder Basis-Software benötigt man zu diesem Zeitpunkt noch nicht, um repräsentative Aussagen zu treffen. Bei der Analyse des von SystemDesk generierten RTE-Codes stellte sich heraus, dass der Speicherbedarf nahezu vernachlässigbar war.

„Die vielen Möglichkeiten zur grafischen Komplexitätsreduktion erleichtern und beschleunigen die Arbeit mit umfangreichen Systemmodellen.“

Luigi Romagnoli, Magneti Marelli

Die Zugriffe auf die Variablen konnten fast vollständig durch #define-Statements realisiert werden, sodass es kaum Einbußen bei der Performanz im Vergleich zur klassischen Implementierung gab.

Einbindung des Applikationscodes

Nachdem sich gezeigt hat, dass die Generierung der RTE hinreichend effizienten Code liefert, arbeiten wir derzeit an der Entwicklung der Anwendungsschicht der AUTOSAR-Applikation. Hierfür soll nicht nur existierender handgeschriebener Legacy-Code wiederverwendet werden, sondern auch der gesamte C-Code, der direkt aus TargetLink-Modellen generiert wurde. Um die AUTOSAR-Konformität des Codes zu erreichen, verfolgen wir in dieser Projektphase den Ansatz, sowohl Legacy-Code als auch TargetLink-Code in AUTOSAR-konforme Wrapper einzubetten, wofür wir unter anderem die Automatisierungsfunktionalität in SystemDesk einsetzen. Diese Vorgehensweise ist auch

dadurch motiviert, dass sowohl der Legacy-Code als auch der TargetLink-Code die gleiche Schnittstelle aufweisen, also die gleiche Art von Wrapper genutzt werden kann. Für die Zukunft planen wir allerdings, die spezielle AUTOSAR-Unterstützung in TargetLink zu nutzen und sowohl AUTOSAR-konformen Code als auch die Softwarekomponenten-Beschreibung direkt aus den TargetLink-Modellen zu generieren. Durch diesen Workflow erwarten wir einerseits eine bessere Nutzung der Ressourcen, da die AUTOSAR-konformen Wrapper hierdurch eliminiert werden, andererseits aber auch eine Vereinfachung des Workflows im Zusammenspiel mit SystemDesk und damit einen effizienteren Entwicklungsprozess.

Erkenntnisse und Ausblick

Die von uns durchgeführten Entwicklungen zeigen, dass die Spezifikationen und Beschreibungsdaten eines herkömmlichen Motorsteuergeräts mit vertretbarem Aufwand nach AUTOSAR migriert werden können. Die Unterstützung durch automatisierbare Werkzeuge ermöglicht es, auch komplexe Systeme und große Datenmengen sicher zu handhaben. Erste Performanzmessungen zeigen, dass der AUTOSAR-Standard nicht zwingend zu einem erhöhten Laufzeit- und Speicherbedarf führt, wenn die Entwicklungswerkzeuge für bestmögliche Effizienz ausgelegt sind. Die Arbeiten an der AUTOSAR-konformen Architektur und der Wrapper-Software sind abgeschlossen.

Fazit

- Erfolgreiche Migration eines vorhandenen Steuergeräts nach AUTOSAR
- Effizienter RTE-Code direkt aus SystemDesk
- Komplexität von Motorsteuerungsmodellen mit SystemDesk leicht zu handhaben

Glossar

ASAP2 – Eine ASAP2-Beschreibungsdatei enthält alle Informationen zu den relevanten Datenobjekten im Steuergerät, z. B. Kenngrößen (Parameter, Kennlinien, Kennfelder).

Python – Skriptsprache, entworfen mit dem Ziel, möglichst einfach und übersichtlich zu sein.

Runnables – Die ausführbaren Elemente einer AUTOSAR-SWC, vergleichbar einer Funktion.

Runtime Environment (RTE) – Zwischenschicht, die die Softwarekomponenten von Applikationscode und Basis-Software in AUTOSAR-Designs verbindet.

Scheduling – Definition der zeitlichen Ausführung von Prozessen und Tasks.

Sender-/Receiver-Interfaces – Datenschnittstellen für AUTOSAR-SWC

Softwarekomponenten – Software Components (SWC), Strukturelemente von AUTOSAR mit dem Ziel, wiederverwendbare Softwaremodule zu erstellen.

Task – Ein Prozess, der in einem System abläuft.

Aktuelle Modelldaten

- Softwarekomponenten: 172
- Datenzugriffe: 9696
- Datenelemente: 2650
- Tasks: 16
- Runnables: 624

Wir befinden uns nun in der Implementierungsphase und binden den vorhandenen Applikationscode sowie das Betriebssystem ein. Im letzten Quartal dieses Jahres werden wir das vollständig migrierte Steuergerät vorstellen. ■

*Alessandro Palma
Luigi Romagnoli
Walter Nesci
Manager AUTOSAR Cross-Project X-PRO
walter.nesci@magnetimarelli.com
Magneti Marelli
Italien*