

eclipse

VSA

AUTOSAR-TOOL-ENTWICKLUNG MIT ECLIPSE

AUTOSAR

meets Eclipse

Parallel zu AUTOSAR hat sich mittlerweile der OSGi-Standard Eclipse in der Automobilindustrie fest etabliert. Ob als Entwicklungsumgebung oder als Target-Plattform zur Entwicklung neuer Applikationen findet das mächtige Framework unter anderem aufgrund seiner Offenheit und seiner Skalierbarkeit immer mehr Anhänger.

**K**aum fahren die ersten AUTOSAR Steuergeräte auf der Straße, steht schon die nächste AUTOSAR-Version in den Startlöchern. Die aktuelle AUTOSAR Version 4.0, wartet mit einer ganzen Reihe von neuen Konzepten auf, u. a. Variantenmanagement, einem Timing Modell, Ethernet und Teilnetzbetrieb (Partial Networking), um nur die Wichtigsten zu nennen. Diese spiegeln sich in einem noch mächtigeren Metamodell wieder, was den Standard nicht unbedingt einfacher macht. Abermals sind besonders die Toolhersteller gefordert, in kürzester Zeit eine benutzerfreundliche, stabile und leistungsfähige Toollösung umzusetzen, die den AUTOSAR-Standard voll unterstützt. Gleichzeitig muss jedoch ein tooltechnischer Migrationspfad von existierenden Designdaten auf die neueste AUTOSAR-Version gewährleistet sein.

Besonders für die AUTOSAR-Toolentwicklung hat sich daher Eclipse als Target Plattform in den letzten Jahren mehrfach bewährt.

#### Toolentwicklungsunterstützung

Mit der Eclipse-Basisplattform werden bereits einige Grundmechanismen und Möglichkeiten bereitgestellt, die auch für die Entwicklung von AUTOSAR-Werkzeugen sehr nützlich sind.

#### Plug-Ins und Extension Points

Plug-Ins ermöglichen den modularen Aufbau und die gezielte Funktionserweiterung einer bestehenden Eclipse Applikation. Aufgrund des „Lazy-Loading“-Prinzips werden Plug-Ins nur bei Bedarf geladen. Durch den Eclipse Built-in-Update-Mechanismus können Updates oder zusätzliche Funktionalitäten, die zentral eingespielt werden, einfach

nachinstalliert werden. Dies begünstigt besonders den Software-Rollout-Prozess oder die Softwarewartung in einem Großunternehmen.

Mit dem Extension-Point-Konzept in Eclipse können Plug-In-Funktionalitäten anderen Plug-Ins zur Verfügung gestellt werden. Auch User Interfaces können über Extension Points erweitert werden.

#### Eclipse Perspektiven

AUTOSAR definiert bewusst keine Anwenderrollen, da diese abhängig von der Unternehmensorganisation, ja sogar vom Steuergeräteprojekt sehr unterschiedlich gehandhabt werden. Hier können jedoch Eclipse-Perspektiven helfen. Diese ermöglichen das Gruppieren und Anordnen von vorhandenen Editoren und Viewern innerhalb der Applikation, um zum Beispiel eine bestimmte Anwender-Rolle abzubilden. Perspektiven können auch vom Anwender definiert werden.

#### Wizards

Das AUTOSAR-Metamodell definiert sämtliche Objekte, deren Relationen und Parameter, um ein Distributed Embedded System zu beschreiben, bietet dem Anwender jedoch keine Hilfestellung in der Vorgehensweise im Modellierungs- und Konfigurationsprozess.

Hier bieten sich Wizards an, die durch eine definierte Reihenfolge von Dialogen den Anwender durch einen Design- und Konfigurationsprozess führen. Gerade im AUTOSAR-Konfigurationsprozess finden sich zahlreiche, immer wiederkehrende Design-Tasks. Ein Beispiel hier wäre die Erstellung eines Port-Interfaces und dessen Konfiguration wie es in **Bild 1** dargestellt ist.

Hier bietet Eclipse dedizierte Unterstützung in der Erstellung von Wizards, die den Anwender durch einen Konfigurationsprozess führen. Dadurch wird besonders dem AUTOSAR-Neuling der Einstieg in den Umgang mit dem mächtigen Standard bedeutend erleichtert.

### Eclipse und Modell-basierte Entwicklung

Eclipse bietet eine Reihe von Modellierungsframeworks, die bereits wertvolle Basisfunktionalitäten zur Verfügung stellen. Als Input ist eine Metamodell-Beschreibung gefordert, wie sie z. B. vom AUTOSAR-Konsortium zur Verfügung gestellt wird. Hieraus können mithilfe von unterschiedlichen Frameworks ein Großteil der Komponenten eines AUTOSAR-Werkzeugs wie (grafische) Editoren, View-er oder Konsistenzregeln automatisch generiert werden. Die wichtigsten Vertreter hier sind EMF, GEF und GMF.

#### ■ Eclipse Modeling Framework (EMF)

EMF ist ein Modeling Framework, welches auf Basis eines strukturierten Datenmodells (z. B. UML) Codegenerierungsmöglichkeiten von Tools und Applikationen zur Verfügung stellt.

#### ■ Eclipse Graphical Editing Framework (GEF)

GEF ist ein Projekt, das es Entwicklern ermöglicht, mithilfe eines existierenden Datenmodells schnell einen grafischen Editor zu erstellen.

#### ■ Eclipse Graphical Modeling Framework (GMF)

Das Framework GMF stellt eine generative Komponenten-

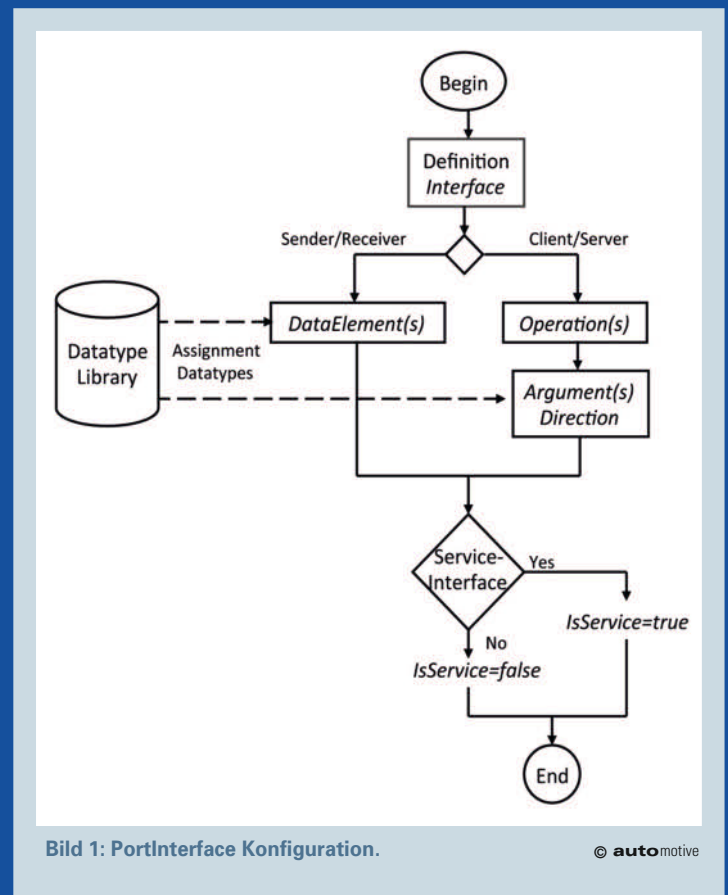


Bild 1: PortInterface Konfiguration.

© automotive

und Runtime-Infrastruktur für die Entwicklung grafischer Editoren auf der Basis der Projekte EMF und GEF bereit.

Um Teile der AUTOSAR-Komplexität zu abstrahieren und so die Benutzerfreundlichkeit zu erhöhen, gibt es gerade in AUTOSAR-Werkzeugen einen großen Bedarf an grafischen Editoren. Beispiele hierfür sind grafische Editoren zur Erstellung von Compositionen (Softwarearchitektur), Softwarekomponenten, Internal Behavior oder Topologien.

### Toolerweiterungen durch

#### Eclipse Add-Ons

Die Anzahl an verfügbaren Eclipse Tools und Plug-Ins scheint mittlerweile endlos zu sein. Viele der frei oder kommerziell verfügbaren Add-Ons sind gerade für die tool-gestützte Entwicklung von Embedded Systemen eine große Bereicherung. Interessante Toolerweiterungen sind zum Beispiel Konfigurationsmanagement-Tools wie SVN, ClearCase, Synergy oder Reporting Frameworks wie BIRT (Business Intelligence Reporting Technology) oder Jasper-Reports, um nur ein paar zu nennen. Durch die im Eclipse Framework definierten Schnittstellen lassen sich Eclipse-basierte Applikationen mit derartigen Add-Ons einfach erweitern.

### AUTOSAR Authoring am Beispiel Volcano VSA

Grundbestandteil einer AUTOSAR-Entwicklungstoolkette ist ein AUTOSAR-Autoren-Tool zur Erstellung einer Systembeschreibung, die auf im AUTOSAR-Standard

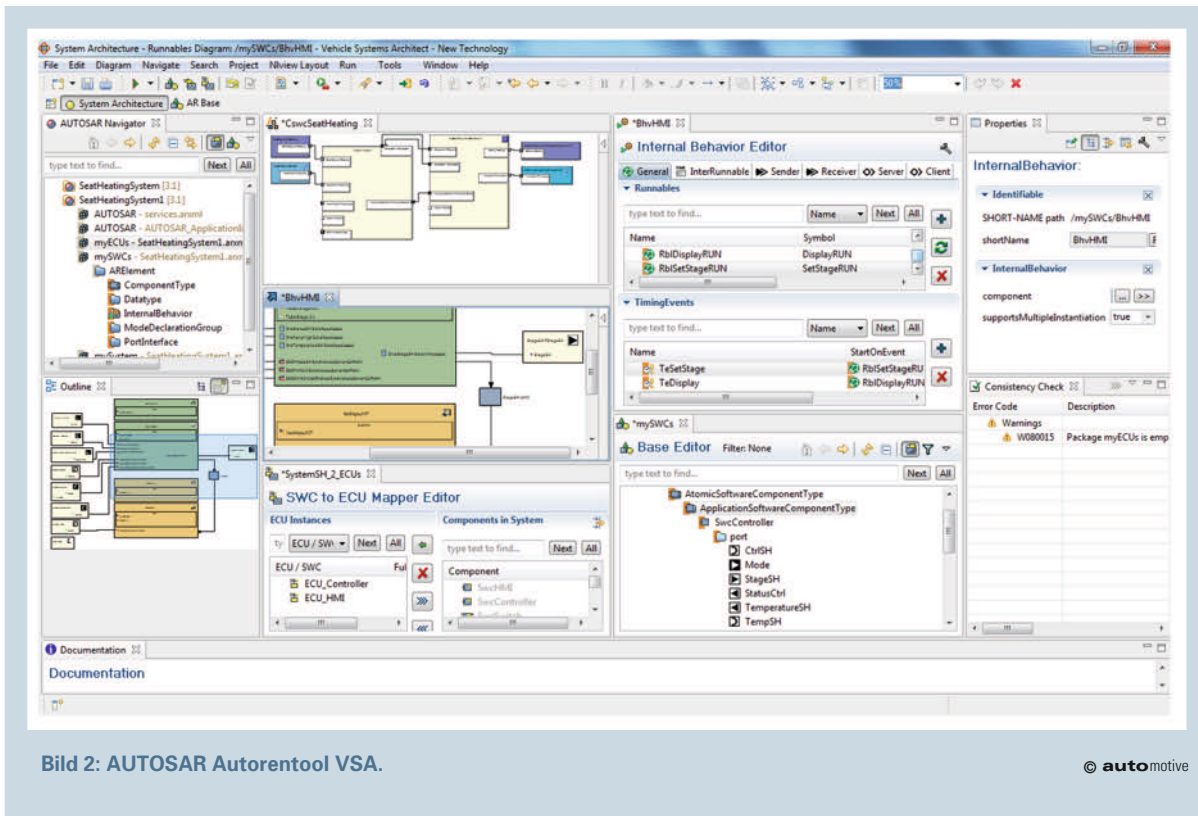


Bild 2: AUTOSAR Autorentool VSA.

© automotive

definierte Templates basiert (Bild 2). Zur Erstellung dieser Systembeschreibung sind Designschritte wie Software-Komponenten, -Behavior und -Architekturerstellung, COM-Matrix-Design, Topology-Beschreibung, Funktionsmapping (SWC-zu-ECU-Mapping) und Datenmapping (DataElement-zu-SystemSignal-Mapping) erforderlich.

Sämtliche Designschritte gilt es durch begleitende Konsistenzprüfungen zu verifizieren, um eine valide und vollständige Konfiguration zu gewährleisten. Aus der Gesamt-Systembeschreibung wird letztendlich der ECU Extract generiert, welcher als Input für das BasisSoftware-Konfigurationstool benötigt wird. Genau diese Designaufgaben werden durch das AUTOSAR-Authoring-Tool Volcano VSA abgedeckt, welches bereits seit einigen Jahren in der Automobilindustrie erfolgreich eingesetzt wird. Viele der durch Eclipse verfügbaren Mechanismen und Technologien sind in dieser Lösung umgesetzt.

Die daraus in VSA resultierende Komplettabdeckung des AUTOSAR-Standards garantiert einen zuverlässigen und verlustlosen Datenaustausch mit anderen AUTOSAR-Entwicklungswerkzeugen.

### Mit Artop in die Zukunft

Im Zusammenhang mit der Eclipse-basierten AUTOSAR Toolentwicklung darf das Artop-Projekt (AutosAR Tool Platform) nicht unerwähnt bleiben.

Dieses, ursprünglich von BMW Car IT initiierte Projekt, resultiert mittlerweile in einer robusten und industrieprobten AUTOSAR-Tool-Basisplattform. Zahlreiche Grundfunktionalitäten wie z. B. Persistence Layer, Workspace Management, ein Konsistenzprüfungs-Framework und die Implementierung von verschiedenen AUTOSAR-Metamodellversionen sind in diesem Framework bereits

enthalten. Auf Basis dieser Plattform können sich nun Toolanbieter auf die Implementierung erweiterter Funktionalitäten fokussieren, die vor allem die Handhabung des mittlerweile sehr komplexen AUTOSAR-Standards erleichtern und das Handling von großen Designdaten adressieren. An Ideen mangelt es hier schließlich nicht.

Die neue Generation der AUTOSAR-Toolkette von Mentor Graphics basiert auf der Artop Plattform und ist demnächst verfügbar. (oe)

### Literatur

- [1] [www.autosar.org](http://www.autosar.org)
- [2] *AUTOSAR\_TPS\_SoftwareComponent Template.pdf*
- [3] [www.eclipse.org](http://www.eclipse.org)
- [4] *VSA User Guide*
- [5] [www.artop.org](http://www.artop.org)



**Michael Seibt** ist bei Mentor Graphics als Produkt Manager für AUTOSAR-Tooling verantwortlich.



**Tibor Kovacs** arbeitet bei Mentor Graphics als Teamleiter in der AUTOSAR Tool-Entwicklung.

# ▶▶ Steuergeräte-Test

## Bringen Sie System in Ihre Testabläufe

mit systematischen und reproduzierbaren Steuergeräte-Tests. Effiziente Testsysteme beschleunigen Ihre CAN-, LIN-, FlexRay- und MOST Entwicklung.



- > Schaffen Sie reale Testumgebungen bereits in sehr frühen Entwicklungsphasen mit generierten Restbussimulationen.
- > Nutzen Sie automatisch erstellte Testfälle und Testreports für systematische Komponenten- und Systemtests.
- > Beschleunigen Sie den Test Ihrer Steuergeräte durch den Einsatz der gleichen Plattform für Entwicklung und Test.
- > Greifen Sie in Ihren Tests auf die Schnittstellen für Kalibrierung, Diagnose oder die I/O-Ports zu.
- > Testen Sie die vollständige Steuergerätefunktion mit dem VT System, das die Umgebung simuliert oder ansteuert.

**Mit den Werkzeugen und Services von Vector erreichen Sie ein Maximum an Testqualität und Testtiefe in allen Phasen der Entwicklung.**

- ▶ **Informationen und Downloads:**  
[www.vector.com/ecu-test](http://www.vector.com/ecu-test)

**Vector VT System**  
Die modulare Testhardware  
[www.vector.com/vt-system](http://www.vector.com/vt-system)

