

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Was verbirgt sich hinter AUTOSAR? . . . . .	3
1.2	Zum Aufbau des Buches . . . . .	4

## Teil I

### Grundlagen

<b>2</b>	<b>Softwarearchitektur in der Fahrzeugentwicklung</b>	<b>9</b>
2.1	Softwarearchitektur oder Systemarchitektur? . . . . .	9
2.2	Was ist überhaupt Architektur? . . . . .	11
2.3	Architektur in der Softwaretechnik . . . . .	13
	2.3.1 Anforderungstypen . . . . .	16
	2.3.2 Abgrenzung zur Systemarchitektur . . . . .	19
	2.3.3 Konzepte über alles . . . . .	19
2.4	Abstraktion erzeugen . . . . .	21
	2.4.1 Umgang mit Abstraktion . . . . .	22
	2.4.2 Domänenspezifische Datentypen . . . . .	24
	2.4.3 Architektursichten . . . . .	24
2.5	Ein System partitionieren . . . . .	26
	2.5.1 Separation of Concerns . . . . .	26
	2.5.2 Lokale und globale Kompliziertheit ausbalancieren . . . . .	30
	2.5.3 Schichtenarchitekturen . . . . .	31
	2.5.4 Komponentenarchitekturen . . . . .	32
	2.5.5 Das KISS-Prinzip . . . . .	33

<b>3</b>	<b>Motive für den Einsatz von AUTOSAR</b>	<b>35</b>
3.1	Der Mythos der stabilen Anforderung	35
3.2	Architektur und Risikomanagement	37
3.3	Komplexität und Zuverlässigkeit	39
3.4	Hardwareunabhängigkeit fördern	40
3.5	Steuergerätezentrierte und funktionsorientierte Sicht	41
3.6	Wiederverwendung von Komponenten	42
3.7	Hilfe für die Systemintegratoren	43
	3.7.1 Schnittstellenaustausch	43
	3.7.2 Basissoftware	43
3.8	Qualitative Aspekte	44
<b>4</b>	<b>AUTOSAR im Detail</b>	<b>47</b>
4.1	Ziele	47
4.2	Schwerpunkte	49
4.3	Organisation	50
	4.3.1 Struktur	50
	4.3.2 Mitglieder	51
4.4	Geltungsbereich	53
4.5	Standardkontext	53
	4.5.1 Am Beispiel JasPar	54
	4.5.2 Am Beispiel HIS	55
4.6	Zeitliche Einordnung	57
	4.6.1 Release 1.0 »proof of concept«	58
	4.6.2 Release 2.0 »RTE und Konfiguration«	58
	4.6.3 Release 2.1 »Abrundung«	58
	4.6.4 Release 3.0 »Weiterentwicklung«	58
	4.6.5 Release 3.1 »OBD-II«	59
	4.6.6 Zukünftige Releases	59
	4.6.7 Erreichter Reifegrad	59
4.7	Einsatzmöglichkeiten von AUTOSAR	59
	4.7.1 Tief eingebettete Automotivsysteme	59
	4.7.2 Nicht im Fokus des Standards	62
	4.7.3 Weitere potenzielle Einsatzgebiete	63
	4.7.4 Rechtliche Aspekte	63

4.8	Grundlegende Begriffe	64
4.8.1	Methodology	64
4.8.2	Virtual Functional Bus (VFB)	65
4.8.3	Software Component (SW-C)	65
4.8.4	Port	65
4.8.5	Port Interface (Interface)	65
4.8.6	Runnable Entity (Runnable)	66
4.8.7	Run-Time Environment (RTE)	66
4.9	Die Spezifikation	67
4.9.1	Struktureller Aufbau: Kategorien, Verzeichnisse und Dateinamen	67
4.9.2	Inhaltlicher Aufbau: Einstiegspunkte und weitere Orientierung	68
4.9.3	Aufbau der Softwarespezifikationen: Effektiv Informationen ermitteln	70

## Teil II

### Engineering

<b>5</b>	<b>Die AUTOSAR-Methodik</b>	<b>75</b>
5.1	Was ist die AUTOSAR-Methodik?	75
5.2	Grafische Notation	76
5.2.1	Bezugnahme auf Elemente des AUTOSAR-Metamodells (Reference to elements of the meta model)	77
5.2.2	Aktivität (Activity)	77
5.2.3	Unterstützung/Führung (Guidance)	78
5.2.4	Fluss von Arbeitsprodukten (Flow of Work Products)	78
5.2.5	Abhängigkeit (Dependency)	78
5.2.6	Komposition/Zusammenstellung (Composition)	79
5.3	Grundlegender Ablauf	79
5.4	Erläuterung wichtiger Konfigurationsschritte	81
5.4.1	Systemkonfiguration	81
5.4.2	ECU-Konfiguration	82
5.5	Aufteilung zwischen den beteiligten Parteien	83
<b>6</b>	<b>Die Systemsicht/der Virtual Functional Bus</b>	<b>85</b>
6.1	Einordnung entsprechend der AUTOSAR-Methodik	85
6.2	Virtual Functional Bus (VFB)	86

6.3	Grafische Notation .....	87
6.3.1	Softwarekomponente (Component) .....	88
6.3.2	Anwendungssoftwarekomponente (Application Software Component) .....	88
6.3.3	Atomare Softwarekomponente (Atomic Software Component) .....	88
6.3.4	Komposition (Composition) .....	89
6.3.5	Sensor/Aktor-Softwarekomponente (Sensor/Actuator Software Component) .....	89
6.3.6	Runnable Entity (Runnable) .....	90
6.3.7	Ports .....	91
6.3.8	Port Interface (Interface) .....	93
6.3.9	Assembly Connector (Connector) .....	93
6.4	Beispiel: Warnblinkfunktion aus VFB-Sicht .....	94
6.4.1	Übersicht .....	94
6.4.2	Detailbetrachtungen .....	95
<b>7</b>	<b>Kommunikationsmechanismen</b>	<b>99</b>
7.1	Einordnung entsprechend der AUTOSAR-Methodik .....	99
7.2	Kommunikation zwischen Softwarekomponenten .....	100
7.3	Sender/Receiver .....	101
7.3.1	Multiplizität .....	101
7.3.2	Explizit .....	102
7.3.3	Implizit .....	104
7.4	Client/Server .....	105
7.5	Events .....	107
7.6	Einschränkungen .....	108
7.6.1	Runnable-Aktivierungsmöglichkeiten .....	108
7.6.2	Empfangsmechanismen und Runnable-Kategorien .....	109
7.7	Kommunikation in Softwarekomponenten .....	109
7.8	Beispiele .....	111
7.8.1	Explizite Sender/Receiver-Kommunikation mit Datensemantik .....	111
7.8.2	Client/Server-Kommunikation .....	114

<b>8</b>	<b>Die Steuergerätesicht (ECU-Sicht)</b>	<b>117</b>
8.1	Einordnung entsprechend der AUTOSAR-Methodik	117
8.2	Aktivitäten	118
8.3	Die AUTOSAR-Schichtenarchitektur (AUTOSAR Layered Software Architecture)	118
8.3.1	Anwendungssoftware (Application Layer)	119
8.3.2	AUTOSAR Runtime Environment (RTE)	119
8.3.3	Die Generierung der RTE	120
8.4	Basissoftware (Basic Software)	121
8.4.1	Serviceschicht (Services Layer)	121
8.4.2	Steuergeräteabstraktionsschicht (ECU Abstraction Layer)	122
8.4.3	Mikrocontrollerabstraktionsschicht (Microcontroller Abstraction Layer)	122
8.4.4	Complex Device Drivers	123
8.5	Die Architektursymmetrie	123
<b>9</b>	<b>Die Basissoftware</b>	<b>125</b>
9.1	Einordnung der Basissoftware	125
9.2	Aufbau der Basissoftware	126
9.3	Einordnung der Basissoftwaremodule	127
9.4	Funktionale Kurzbeschreibung der Basissoftwaremodule	129
9.4.1	Die Systemdienste	129
9.4.2	Der Geräte-Stack	131
9.4.3	Der Speicher-Stack	132
9.4.4	Der Kommunikations-Stack	134
9.4.5	I/O-Stack	138
9.4.6	Die Complex Device Drivers	139
9.5	Kommunikationsbeziehungen in der Basissoftware	140
9.5.1	Erlaubte Kommunikationsbeziehungen	141
9.5.2	In Ausnahmefällen gestattete Kommunikationsbeziehungen	142
9.5.3	Verbotene Kommunikationsbeziehungen	142
9.6	Implementation Conformance Classes 1-3	143
9.7	Beispiele	145
9.7.1	Das Zusammenspiel von BSW-Modulen im Kommunikations-Stack	145
9.7.2	Das Zusammenspiel von BSW-Modulen im Speicher-Stack	147

<b>10</b>	<b>Performance – oder »Was kostet AUTOSAR?«</b>	<b>151</b>
10.1	Ressourcenverbrauch resultiert in Entwicklungs- und Herstellungskosten . . . . .	151
10.2	Kostenoptimierung durch Performance-Optimierung . . . . .	153
10.2.1	Benchmarking – Ermittlung des Ressourcenbedarfs . . . . .	153
10.2.2	Angabe des Ressourcenbedarfs . . . . .	157
10.3	Ressourcenanforderungen an eine AUTOSAR-ECU . . . . .	157
10.4	Objektorientierung zur Performance-Steigerung . . . . .	159
10.5	Schlussfolgerungen . . . . .	164
<b>11</b>	<b>Variantenmanagement</b>	<b>165</b>
11.1	Herangehensweise . . . . .	165
11.1.1	Softwarebasis analysieren . . . . .	166
11.1.2	Weiterentwicklung parallelisieren . . . . .	167
11.1.3	Voraussetzungen schaffen . . . . .	169
11.2	AUTOSAR ermöglicht Variantenmanagement . . . . .	169
11.3	Variantenmanagementsysteme einsetzen . . . . .	170
11.4	Beispiel: Warnblinken und Variantenmanagement . . . . .	171
11.5	Schlussfolgerung . . . . .	173

## Teil III

### Management

<b>12</b>	<b>AUTOSAR kritisch betrachtet</b>	<b>177</b>
12.1	Neue Abhängigkeiten im Entwicklungsprozess . . . . .	177
12.1.1	Bindung an einen Werkzeughersteller . . . . .	177
12.1.2	Werkzeugintegration . . . . .	178
12.1.3	Austauschbarkeit der AUTOSAR-XML-Dateien . . . . .	178
12.1.4	Releases und Revisions des Standards . . . . .	179
12.2	Modellierung und Integration . . . . .	180
12.2.1	Schnittstelle und Semantik . . . . .	180
12.2.2	Das Problem der Modellierung dynamischer Abläufe . . . . .	181
12.2.3	Von der Integrationsnot in die Konfigurationsnot . . . . .	182
12.2.4	Die Dokumentation einer Konfiguration . . . . .	182
12.2.5	Das Prinzip der stabilen Abhängigkeit . . . . .	183
12.2.6	Architekturerosion . . . . .	184
12.3	Besondere technische Aspekte . . . . .	184
12.3.1	Sensorfusion . . . . .	185
12.3.2	Bandbreiten und andere Engpässe . . . . .	185

12.3.3	Dynamik zur Laufzeit	186
12.3.4	Timing und Multitasking	187
12.3.5	Beliebige Verschiebbarkeit von Komponenten	187
12.4	Verwaltung mit technischem Sachverstand	188
12.5	Anforderungen an den Architekten	189
<b>13</b>	<b>Betriebswirtschaftliche Aspekte</b>	<b>193</b>
13.1	AUTOSAR aus mikroökonomischer Sicht	193
13.2	Der Nutzen von AUTOSAR in der Entwicklung	194
13.3	Wiederverwendung	195
13.3.1	Wiederverwendbarkeit auf Anwendungsebene	196
13.3.2	Schutz des geistigen Eigentums – Intellectual Property	197
13.3.3	Die Qualität wiederverwendbarer Komponenten	197
13.4	Wiederverwendbarkeit der Basissoftware	198
13.5	Austauschbarkeit	199
13.6	Geschäftsmodelle	199
13.6.1	Modell 1: Entwicklungsdienstleister	199
13.6.2	Modell 2: Software als Gratisbeilage zur Hardware	201
13.6.3	Modell 3: Verkauf fertiger Bibliotheken	201
<b>14</b>	<b>Produktmanagement mit AUTOSAR</b>	<b>203</b>
14.1	Bisherige Situation	203
14.2	Änderung der Randbedingung	204
14.3	Herausforderungen	206
14.3.1	Soft- und Hardware wird leichter austauschbar für OEMs	206
14.3.2	Welche AUTOSAR-Releases müssen unterstützt werden	207
14.3.3	Was bedeutet »AUTOSAR-compliant«	208
14.3.4	Kompatibilität mit »konventionellen« ECUs	208
14.3.5	Änderung des Fokus	209
14.4	Chancen für den Zulieferer	210
14.4.1	Reines Softwareprodukt	211
14.4.2	Softwareteilprodukt	211
14.4.3	Kombiniertes Hard- und Softwareprodukt	212
14.4.4	Anwendungssoftware mit I/O-Hardwareabstraktion	212
14.4.5	»Software as a product«	213
14.5	Schlussfolgerung	213

<b>15</b>	<b>Migrationsstrategien für bestehende Projekte</b>	<b>215</b>
15.1	Vorbereitungen	215
15.1.1	Der aktuelle Stand	216
15.1.2	Welche Ziele werden langfristig verfolgt?	216
15.1.3	Welche I/O-Schnittstellen sind betroffen?	217
15.1.4	Die Anforderungsstruktur gerade rücken	218
15.1.5	Basissoftware	218
15.2	Werkzeugauswahl	219
15.2.1	Ergonomie	219
15.2.2	Angemessenheit für den gedachten Zweck	221
15.2.3	Anpassung an die Arbeitsweise im Team	222
15.2.4	Den Umgang mit dem Werkzeug beherrschen	223
15.2.5	Pilotprojekte müssen wichtig sein	223
15.3	Legacy-Code weiterverwenden	224
15.3.1	Adapter	224
15.3.2	Fassade	225
15.3.3	Schrittweise Migration	225
15.4	Mögliche Wege für ein Migrationsprojekt	226
15.4.1	Basissoftware und Applikation separieren	226
15.4.2	Nur eine neue Basissoftware einführen	227
15.4.3	Zuerst die RTE und die SW-C-Ports	227
15.4.4	Alles als Complex Device Driver	227
15.4.5	Einsatz vertikaler Prototypen	227
15.5	Tipps zum Vorgehen	229
<b>16</b>	<b>AUTOSAR-Konformität</b>	<b>231</b>
16.1	Was ist Konformität	231
16.2	Was bedeutet AUTOSAR-konform	232
16.2.1	Testspezifikation	233
16.2.2	Testdurchführung	235
16.2.3	Was wird getestet?	235
16.2.4	Wie wird getestet?	237
16.2.5	Testabdeckung	238
16.3	Was bringt der Konformitätstest?	239
<b>17</b>	<b>Ausblick – AUTOSAR in der Zukunft</b>	<b>241</b>
17.1	Release 4.0 »Conformance Testing«	241
17.2	Release 5.0 »...«	242

## Anhang

<b>A</b>	<b>Nützliche Links</b>	<b>247</b>
A.1	Standardlandschaft .....	247
A.2	AUTOSAR-Toolhersteller .....	247
<b>B</b>	<b>AUTOSAR-Entwicklungspartner</b>	<b>249</b>
B.1	Core Partners .....	249
B.2	Premium Members .....	249
B.3	Associate Members .....	251
B.4	Development Members .....	253
<b>C</b>	<b>Abkürzungen</b>	<b>255</b>
<b>D</b>	<b>Glossar</b>	<b>259</b>
<b>E</b>	<b>BSW-Module</b>	<b>265</b>
	<b>Literatur</b>	<b>267</b>
	<b>Stichwortverzeichnis</b>	<b>271</b>