

Auf einen Blick

1	Entwurf von Anwendungssystemen	31
2	Anwendungsobjekt	61
3	Klassen, Interfaces und Ausnahmen	135
4	Anwendungsarchitektur	159
5	Anwendungsschicht	223
6	GUI-Programmierung	291
7	SAP-Geschäftspartner	401
8	Techniken der Anwendungsprogrammierung	459
9	Entwicklung von Webservices	515
10	Anwendungsentwicklung mit SAP HANA	579
11	Informationsquellen in der Projektplanungs- und Realisierungsphase	629
12	Management von Entwicklungsprojekten	649

Inhalt

Geleitwort	19
Vorwort	21
Einleitung	23

1 Entwurf von Anwendungssystemen 31

1.1 Anforderungen	32
1.1.1 Anforderungsanalyse als Prozess	32
1.1.2 Funktionale Anforderungen	33
1.1.3 Nichtfunktionale Anforderungen	34
1.1.4 Grenzbereich funktionaler und nichtfunktionaler Anforderungen	43
1.1.5 Systemspezifikation	46
1.2 Allgemeine architektonische Überlegungen	49
1.2.1 Produktfamilien: Trennung von Rahmen und Inhalt	49
1.2.2 Metadaten	54
1.2.3 Generative Programmierung	56
1.2.4 Modellgetriebene Architekturen	58
1.3 Verwendung des SAP-Standards	59

2 Anwendungsobjekt 61

2.1 Was ist ein Anwendungsobjekt?	62
2.2 Modellierung des Anwendungsobjektes auf Datenbankebene	66
2.2.1 Strukturiertes Entity-Relationship- Modell	67
2.2.2 Datenmodellierung auf ABAP-Dictionary-Ebene	71
2.3 Realisierung der Objektpersistenz	88
2.3.1 Notwendigkeit von Datenbank-Zugriffsschichten	88
2.3.2 Object Services	93
2.3.3 Vererbung persistenter Klassen	99
2.3.4 Zugriff auf abhängige Tabellen	102
2.3.5 Query-Dienst	105

2.3.6	Entwicklung eines eigenen Persistenzdienstes	106
2.3.7	Business Object Processing Framework	107
2.3.8	BAPI-Zugriffsmethoden	115
2.4	Transaktionskonzept	116
2.4.1	Spezielle Techniken des klassischen Transaktionskonzeptes	120
2.4.2	Objektorientiertes Transaktionskonzept	121
2.5	Best Practices	123
2.5.1	Bildung von Primärschlüsseln	123
2.5.2	Modellierung des Anwendungsobjektes im ABAP Dictionary	125
2.5.3	Service-Funktionen für persistente Objekte	125
2.5.4	Speicherung un- und semistrukturierter Daten	129
2.5.5	Performante Datenbankzugriffe	132
2.5.6	Weiterführende Überlegungen	133
2.5.7	Wichtige Transaktionen	134

3 Klassen, Interfaces und Ausnahmen 135

3.1	Vorteile von ABAP Objects	137
3.1.1	Definition von Konstanten in Klassen und Interfaces	137
3.1.2	Funktionsgruppen versus Objekte	138
3.1.3	Events	138
3.2	Ausnahmen	139
3.2.1	Klassische und objektorientierte Ausnahmen	140
3.2.2	Assertions	142
3.2.3	Ausnahmebehandlung	144
3.3	Grundprinzipien des objektorientierten Entwurfs	145
3.3.1	Umkehrungen von Abhängigkeiten	146
3.3.2	Open-Closed-Prinzip	147
3.3.3	Vererbung und das Substitutionsprinzip	149
3.3.4	Testbarkeit durch Unit-Tests	151
3.4	Klassische Modularisierungseinheiten	153
3.4.1	Funktionsbausteine	154
3.4.2	Reports	154
3.5	Best Practices	156

3.5.1	Allgemeine Überlegungen zum objektorientierten Entwurf	156
3.5.2	Wichtige Transaktionen	158
4	Anwendungsarchitektur	159
4.1	Anforderungen an die Anwendungsarchitektur	159
4.2	Softwarestrukturierung aus Sicht der Softwaretechnik	161
4.3	Wie strukturiert man ein Softwaresystem?	165
4.3.1	Strukturierung nach fachlichen und technischen Kriterien	165
4.3.2	Identifizierung von Schichten	169
4.3.3	Zerlegung in Teilanwendungen	170
4.3.4	Schaffung von Basiskomponenten	171
4.3.5	Abhängigkeit vom SAP-Standard	172
4.3.6	Strukturierung der Beispieldatenbank	172
4.4	Paketkonzept	174
4.4.1	Paketschnittstellen und Prüfungen	176
4.4.2	Sichtbarkeit von Paketschnittstellen	179
4.4.3	Strukturpakete und SAP-Softwarekomponenten	179
4.4.4	Durchführung von Paketprüfungen	183
4.4.5	Exkurs: Kompatibilitätsprobleme	184
4.4.6	Exkurs: Namenskonventionen und Namensräume	186
4.5	Komposition von Paketen	189
4.6	Laufzeitkonfiguration von Softwarekomponenten ...	190
4.6.1	Realisierung von Schnittstellen durch Erweiterungen	195
4.6.2	Schaltbare Erweiterungen – das Switch Framework	199
4.6.3	Ereignisbasierte Schnittstellen	204
4.7	Best Practices	212
4.7.1	Architektdokumentation	213
4.7.2	Eigenschaften der Paketzerlegung	214
4.7.3	Schnittstellenkonzeption	216
4.7.4	Paketprüfungsmodus	218
4.7.5	Auslieferung von Softwarekomponenten ...	220
4.7.6	Einführung des Paketkonzepts	220

4.7.7	Operationales Paketkonzept	221
4.7.8	Wichtige Transaktionen	222
5	Anwendungsschicht	223
5.1	Anwendungslogik	224
5.1.1	Realisierung des Anwendungsobjektes	226
5.1.2	Trennung von Objekt und Prozess	229
5.2	Customizing	232
5.2.1	Grundlagen	233
5.2.2	Technisches Customizing	235
5.3	Suchdienste	240
5.4	Workflows	245
5.4.1	Beispielszenario: Wiedervorlage zu einem bestimmten Termin	247
5.4.2	Wichtige Transaktionen	263
5.5	Geschäftsregeln mit BRFplus	263
5.5.1	Implementierung von Geschäftsregeln	265
5.5.2	Entwicklung von Geschäftsregeln mit BRFplus	266
5.5.3	Aufruf einer BRFplus-Funktion in ABAP	281
5.5.4	Architektur von regelbasierten Anwendungen	284
5.5.5	Best Practices	286
5.5.6	Zusammenfassung	288
5.5.7	Wichtige Transaktionen	289
6	GUI-Programmierung	291
6.1	Überblick über die UI-Technologien von SAP	292
6.2	Ergonomiebeispiele und Dialogstandards	294
6.2.1	SAP R/3 Style Guide	294
6.2.2	Ergonomiebeispiele	295
6.2.3	Menüstandards	296
6.2.4	Bildaufbau und Benutzerführung	296
6.3	Tabellenpflegedialoge	302
6.3.1	Generierung und Erweiterung von Tabellenpflegedialogen	303
6.3.2	Tipps zum Umgang mit Pflege-Views	312
6.4	Bereichsmenüs	314
6.5	Objektorientierte Dynpro-Programmierung	316

6.5.1	Pro und contra Subscreens	316
6.5.2	Subscreens als Modularisierungseinheit	317
6.5.3	Kapselung mit Dynpros	318
6.5.4	Message Handling mit Dynpros	318
6.5.5	Das BUS-Screen-Framework	319
6.5.6	Vorzüge objektorientierter Dynpros	321
6.5.7	Verwendungen des BUS-Screen-Frameworks	322
6.5.8	Normale Dynpros und modale Dialogfenster	322
6.5.9	Ablauflogik definieren	324
6.5.10	Instanzen erzeugen	325
6.5.11	Dynpro aufrufen	326
6.5.12	Reihenfolge der Verarbeitungszeitpunkte	326
6.5.13	Eigene Dynpro-Logik definieren	328
6.5.14	Titel und GUI-Status setzen	328
6.5.15	Benutzereingaben behandeln	328
6.5.16	Fehlermeldungen sammeln und ausgeben	332
6.5.17	Anbindung des Business Application Logs	335
6.5.18	Table Controls und ALV-Grids	337
6.5.19	Dynpros mit Subscreen-Areas	338
6.5.20	Subscreens definieren	340
6.5.21	Datentransport zwischen Dynpro-Feldern und Dynpro-Klasse	341
6.5.22	Tabstrips	342
6.5.23	Für Fortgeschrittene: Selektionsbilder und Dynpro Painter	348
6.5.24	Selektionsbilder in Verbindung mit dem BUS-Screen-Framework	354
6.5.25	Ausblick	359
6.6	Web Dynpro ABAP	359
6.6.1	Grundlagen	360
6.6.2	Erstellung einer Beispielanwendung	363
6.6.3	Web-Dynpro-ABAP-Konfigurations- Framework	370
6.6.4	Beispiel für eine modifikationsfreie Erweiterung durch Konfiguration	374

6.7	Floorplan Manager	380
6.7.1	Grundrisse	381
6.7.2	Generische und andere UIBBs	382
6.7.3	Feeder-Klassen	385
6.7.4	Floorplan Manager und Konfigurations-Framework	387
6.8	Herausforderung der Dialogintegration	388
6.9	SAP NetWeaver Business Client	391
6.9.1	Desktop- und HTML-Client	391
6.9.2	Index-Seite	392
6.9.3	Side Panel	393
6.10	Best Practices	398
6.10.1	Wahl der richtigen GUI-Technologie	398
6.10.2	Softwaretechnische Aspekte	398
6.10.3	Wichtige Transaktionen	399
6.10.4	Wichtige Web-Dynpro-Anwendungen	400

7 SAP-Geschäftspartner

7.1	Hintergrundinformationen	401
7.1.1	Entstehung des SAP-Geschäftspartners	402
7.1.2	Konzeptueller Überblick	403
7.1.3	Erster Eindruck	404
7.2	Erweiterung des SAP-Geschäftspartners	406
7.2.1	Beispiel für eine Erweiterung	406
7.2.2	Anwendung pflegen	408
7.2.3	Datenset pflegen	409
7.2.4	Tabellen pflegen	409
7.2.5	Feldgruppen pflegen	410
7.2.6	Sichten (Transaktion BUS3)	410
7.2.7	Abschnitt (Transaktion BUS4)	412
7.2.8	Bilder (Transaktion BUS5)	412
7.2.9	Bildfolgen (Transaktion BUS6)	414
7.2.10	GP-Sichten (Transaktion BUSD)	414
7.2.11	Rollentyp und Rolle anlegen	414
7.2.12	Funktionsgruppe »ZVHM_BUPA«	416
7.2.13	Aufbau des Dynpros 0100	418
7.2.14	Zeitpunkte	418
7.2.15	BDT-Namenskonventionen	431
7.2.16	Test der Erweiterung	432

7.2.17	Fehlersuche	433
7.2.18	Zusammenfassung	436
7.3	Erweiterung des SAP Locators	437
7.3.1	Einführung in den SAP Locator	437
7.3.2	Ziel der Erweiterung	438
7.3.3	Transaktion LOCA_CUST	439
7.3.4	Definition der Hierarchie	440
7.3.5	Append-Suchhilfe anlegen	442
7.3.6	Elementare Suchhilfe anlegen	442
7.3.7	Suchhilfe zur Append-Suchhilfe zuordnen	443
7.3.8	Funktionsgruppe anlegen	444
7.3.9	Such-Dynpro anlegen	444
7.3.10	Formroutine zum Initialisieren der Suche	446
7.3.11	Formroutine zum Holen der Suchfelder	447
7.3.12	Formroutine zum Setzen der Suchfelder	447
7.3.13	Formroutine zum Erzeugen des Dynpro-Objektes	448
7.3.14	Funktionsbaustein anlegen	449
7.3.15	Lokale Suchklasse anlegen	450
7.3.16	Suche-ID im Locator-Customizing bekannt machen	452
7.3.17	Suche testen	453
7.3.18	Zusammenfassung'	453
7.4	Geschäftspartner in SAP Master Data Governance	454
7.5	Wichtige Transaktionen	458

8 Techniken der Anwendungsprogrammierung 459

8.1	Realisierung des Anwendungsprotokolls	460
8.1.1	Adressat der Protokolle	461
8.1.2	Protokollrecherche als Geschäftsprozess	461
8.1.3	Business Application Log (BAL)	464
8.1.4	Datenmodell des BAL	464
8.1.5	Entwicklerschnittstelle	465
8.1.6	Beispiel: Protokoll erzeugen und anzeigen	466
8.1.7	Beispiel: Protokoll speichern	468

8.1.8	Transaktionskonzept	469
8.1.9	Protokolle anreichern	472
8.1.10	Komplexe Zusatzdaten speichern	480
8.1.11	Weitere Callbacks in der Anzeige nutzen	482
8.1.12	Benutzerdefinierte Buttons	483
8.1.13	Protokolle löschen und archivieren	484
8.1.14	Zusammenfassung und weiterführende Informationen	484
8.2	Anwendungen parallelisieren	485
8.2.1	Anwendungsfall	485
8.2.2	Voraussetzungen	488
8.2.3	Asynchroner Remote Function Call (aRFC)	492
8.2.4	Parallelisierung mit Hintergrundjobs	504
8.2.5	Parallelisierung mit dem PV-Tool BANK_PP_JOBCTRL	506
8.2.6	Zusammenfassung	512
8.2.7	Weiterführende Informationen	513
8.3	Wichtige Transaktionen	514

9 Entwicklung von Webservices 515

9.1	Überblick über die Konnektivitätstechnologien des AS ABAP	516
9.1.1	Weiterentwicklung der RFC-Technologie	517
9.1.2	OData für moderne Webanwendungen und leichtgewichtige Integrationsszenarien	518
9.1.3	Stärken von Webservices	518
9.1.4	Legacy-Technologien	521
9.1.5	Rolle der SAP Process Integration	521
9.1.6	Zusammenfassung	523
9.2	Grundlagen zu Webservices	525
9.2.1	Einführung in die grundlegenden Standards	525
9.2.2	Unterstützte Standards und Interoperabilität	526
9.2.3	SOAP-Runtime der ABAP-Plattform	528
9.3	Modellierung von Webservices	528
9.3.1	Modellierung von Webservice-Signaturen	530

9.3.2	Kommunikationsmuster und Standards für Datentypen	535
9.3.3	Eigene XML-Namensräume bei der Modellierung	536
9.4	Entwicklung eines Beispiel-Webservice	538
9.4.1	Entwicklung von Enterprise Services mit dem Metadata Repository	538
9.4.2	Durchführung der Implementierung	550
9.4.3	Konfiguration des Webservice mit dem SOAMANAGER	554
9.4.4	Weitere Implementierungsdetails von Webservices	558
9.5	Konsumieren von Webservices	563
9.5.1	Aufruf eines externen Webservice	564
9.5.2	Anmerkungen zum LUW-Konzept beim Aufruf von Consumer-Proxys	569
9.5.3	Service-Gruppen	570
9.5.4	Weitere Funktionen der Transaktion SOAMANAGER	571
9.6	Weitere Themen	572
9.6.1	Nutzung des Enterprise Services Repository	572
9.6.2	Inside-Out-Entwicklung	573
9.6.3	Service Implementation Workbench	574
9.6.4	Versionierung von Webservices	575
9.6.5	Test von Webservices	575
9.6.6	Informationen zu weiteren Themen	577
9.6.7	Wichtige Transaktionscodes	577

10 Anwendungsentwicklung mit SAP HANA 579

10.1	Einsatzszenarien für SAP HANA	580
10.1.1	Anpassung bestehender Anwendungen an SAP HANA	580
10.1.2	Neue Anwendungen für SAP HANA entwickeln	581
10.1.3	Abwärtskompatible SAP-HANA- Anwendungen entwickeln	581
10.1.4	Side-by-Side-Anwendungen entwickeln	582
10.2	Besonderheiten von SAP HANA	583
10.2.1	Spaltenorientierung	583

10.2.2	Schnelle Rechen-Engine	583
10.2.3	Besondere Funktionen	585
10.3	Analysewerkzeuge für die Codeoptimierung	588
10.3.1	Abgleich nach einem Release-Upgrade	589
10.3.2	Unicode-Prüfungen	590
10.3.3	Code Inspector	591
10.3.4	Systemlastanalyse	592
10.3.5	SQL-Monitor	593
10.3.6	Arbeitsvorrat für SQL-Optimierung	595
10.3.7	Weitere Analysewerkzeuge	596
10.4	Architekturempfehlungen für den Entwurf neuer Anwendungen	597
10.4.1	Hybrider Charakter der Anwendung	597
10.4.2	Datenbanknahes Programmiermodell	600
10.4.3	Kapselung	601
10.4.4	Erweiterbarkeit beachten	602
10.5	Architekturempfehlungen für abwärtskompatible Anwendungen	603
10.5.1	Nichtfunktionale Anforderungen priorisieren	604
10.5.2	Funktionsliste mit und ohne SAP HANA aufstellen	605
10.5.3	Fallunterscheidung realisieren	606
10.6	Architektur von Side-by-Side-Anwendungen	607
10.6.1	Einsatzfelder von Side-by-Side-Szenarien	608
10.6.2	Datenreplikation	608
10.6.3	Technik für den Zugriff	611
10.6.4	Optionales Sidecar	612
10.7	Transaktionale Anwendungen und Analytik	613
10.8	Praxisbeispiel	617
10.8.1	Voraussetzungen	617
10.8.2	InfoSet anlegen	618
10.8.3	InfoSet verwenden	620
10.8.4	Fazit des Anwendungsbeispiels	625
10.9	Zusammenfassung	625
10.10	Wichtige Transaktionen	626

11 Informationsquellen in der Projektplanungs- und Realisierungsphase	629
11.1 SAP Service Marketplace	629
11.1.1 SAP Help Portal	629
11.1.2 SAP Support Portal	632
11.1.3 SAP Community Network	633
11.1.4 SAP Learning Hub und weitere Trainingsplattformen	634
11.2 ABAP-Schlüsselwortdokumentation	634
11.3 SAP Design Guild	635
11.4 Innenleben des AS ABAP	636
11.4.1 Debuggen	636
11.4.2 Informationsquellen im SAP-System	637
11.4.3 Laufzeitanalyse	639
11.4.4 Datenbank-Trace	642
11.4.5 Umfeldermittlung	643
11.5 Wissensmanagement	644
11.6 Wichtige Transaktionen	647
12 Management von Entwicklungsprojekten	649
12.1 Rollen in Entwicklungsprojekten	649
12.1.1 Rolle des Chefdesigners	650
12.1.2 Frameworks und Tools	650
12.2 Qualitätsmanagement	651
12.2.1 Risikomanagement	652
12.2.2 Entwicklungsrichtlinien	653
12.2.3 Code Inspections und Erweiterung des Code Inspectors	655
12.2.4 Dokumentation anlegen	665
12.2.5 Prüfung freischalten	666
12.2.6 Softwaretest	666
12.2.7 Dokumentation	668
12.2.8 Wartung von Anwendungen	671
12.2.9 Wichtige Transaktionen	672

Anhang	673
A Literaturverzeichnis	675
B Zitatverzeichnis	679
C Die Autoren	681
Index	683