

Uwe Blumöhr, Manfred Münch, Marin Ukalovic

Variantenkonfiguration mit SAP®



Galileo Press 

Bonn • Boston

Auf einen Blick

1	Grundlagen der Variantenkonfiguration	33
2	Erstellung eines Produktmodells für die SAP-Variantenkonfiguration	73
3	Geschäftsprozesse in SAP ERP	231
4	Customizing von SAP ERP für die Variantenkonfiguration	323
5	Spezifika der Produktkonfiguration in SAP CRM	351
6	Herausforderungen in der Variantenkonfiguration	381
7	Weiterentwicklungen in der SAP-Branchenlösung DIMP	463
8	Weiterentwicklungen und Add-ons im SAP-Partnerumfeld	481
9	Projektleiter berichten über Projekte und Projektstrukturen	531
10	Kunden berichten über die Einführung der SAP-Variantenkonfiguration	569
11	Configuration Workgroup	617
12	Ausblick auf SAP Business ByDesign	627
A	Datenbanktabellen der Variantenkonfiguration	643
B	APIs der Variantenkonfiguration	647
C	User Exits der Variantenkonfiguration	649
D	Vollständige Beispiele für Variantenfunktionen	651
E	Die Autoren	657

Inhalt

Vorwort	19
Einleitung	23

1 Grundlagen der Variantenkonfiguration 33

1.1 Was ist Produktkonfiguration?	34
1.1.1 Begriffliche Einordnung	34
1.1.2 Elementare Konfigurationsbausteine	39
1.1.3 Produktkonfiguration in logistischen Szenarien	42
1.1.4 Kernproblem Variantenvielfalt	44
1.1.5 Prozedurale und deklarative Herangehensweise	47
1.2 Was ist die SAP-Variantenkonfiguration?	50
1.2.1 Produktkonfiguration mit dem Variantenkonfigurator (LO-VC)	51
1.2.2 Weitere Einsatzgebiete	51
1.2.3 »Hello World«-Beispiel	52
1.2.4 Variantenkonfigurator (LO-VC)	58
1.2.5 Internet Pricing and Configurator (IPC)	62
1.3 Verbesserung der Geschäftsprozesse durch die Variantenkonfiguration	66
1.3.1 Voraussetzung für den Einsatz der Variantenkonfiguration	67
1.3.2 Faktoren für den Einsatz der Variantenkonfiguration	68
1.3.3 Beispielhafte Betrachtung zum Stammdatenvolumen	70
1.4 Zusammenfassung	71

2 Erstellung eines Produktmodells für die SAP-Variantenkonfiguration 73

2.1 Überblick über die Modellierung und Integration der Variantenkonfiguration	74
2.1.1 Variantenreiche Produkte ohne Variantenkonfiguration	74
2.1.2 Variantenreiche Produkte mit Variantenkonfiguration	74

2.2	Werkzeuge aus dem Klassensystem	79
2.2.1	Merkmalsverwaltung	79
2.2.2	Klassenverwaltung	86
2.2.3	Klassifizierung	88
2.2.4	Suche	89
2.3	Materialstamm, Stückliste und Arbeitsplan	91
2.3.1	Materialstamm des konfigurierbaren Materials	92
2.3.2	Maximalstückliste des konfigurierbaren Materials	96
2.3.3	Maximalarbeitsplan für das konfigurierbare Material	100
2.4	Konfigurationsprofil und Konfigurationsszenarien	103
2.4.1	Konfigurationsprofil im Überblick	103
2.4.2	Konfigurationsprofil im Detail	104
2.4.3	Konfigurationsszenarien im Überblick	111
2.4.4	Szenario »Plan-/Fertigungsauftrag ohne Stücklistenauflösung«	111
2.4.5	Szenario »Auftragsstückliste«	113
2.4.6	Szenario »Kundenauftrag (SET)«	120
2.4.7	Szenario »Plan-/Fertigungsauftrag mit Stücklistenauflösung«	124
2.5	Überblick »Beziehungswissen«	127
2.5.1	Arten von Beziehungswissen und Zuordnung	128
2.5.2	Prozeduraler und deklarativer Charakter von Beziehungswissen	133
2.5.3	Globales und lokales Beziehungswissen	134
2.5.4	Status von Beziehungswissen	134
2.5.5	Beziehungswissen in der Klassifizierung und in der Variantenkonfiguration	135
2.5.6	Ausführungsreihenfolge von Beziehungswissen	136
2.5.7	Grundregeln der Syntax	138
2.5.8	Syntaxelemente	142
2.5.9	Variantentabellen und -funktionen	144
2.5.10	Auswertungsfunktionen für Beziehungswissen	146
2.6	Beziehungswissen für die Bewertungsoberfläche bzw. die vertriebliche Sicht	150
2.6.1	Produktmodellierungsumgebung PMEVC	150
2.6.2	Ein erstes Beispiel	154
2.6.3	Variantentabellen im Detail	160
2.6.4	Constraints im Detail	167
2.6.5	Vorbedingungen	172
2.6.6	Auswahlbedingungen	176

2.6.7	Prozeduren	177
2.6.8	Objektmerkmale	179
2.6.9	Variantenfunktionen	182
2.6.10	Oberflächendesign	186
2.7	Beziehungswissen für Stückliste und Arbeitsplan	188
2.7.1	Lokales und globales Beziehungswissen	188
2.7.2	Auswahlbedingungen für Stücklisten und Arbeitsplan	191
2.7.3	Klassenknoten in Stücklisten	192
2.7.4	Klassifizierte Materialien in Stücklisten	197
2.7.5	Prozeduren in Stückliste und Arbeitsplan	199
2.8	Preisfindung für konfigurierbare Materialien	201
2.9	Erzeugniskalkulation für konfigurierbare Materialien	209
2.10	Materialvarianten	211
2.10.1	Materialstamm der Materialvariante	213
2.10.2	Stückliste und Materialvariante	215
2.10.3	Arbeitsplan und Materialvariante	216
2.10.4	Preisfindung und Materialvariante	218
2.10.5	Materialvariantenfindung	218
2.10.6	Materialvariantenfindung auf Kopf- und Baugruppenebene	222
2.11	Wie erstelle ich ein Produktmodell für den IPC?	223
2.12	Fazit	230

3 Geschäftsprozesse in SAP ERP 231

3.1	Einführung – Variantenkonfiguration in betrieblichen Prozessen	231
3.1.1	Stücklisten in der Variantenkonfiguration	231
3.1.2	Order Engineering Workbench	235
3.2	Variantenkonfiguration mit iPPE – Modellierung	242
3.2.1	Produktvariantenstruktur und Produktdesigner	243
3.2.2	Modellierung von den Anforderungen bis zur Fertigung	244
3.2.3	Anforderungsstrukturen	246
3.2.4	Strukturknoten, Positionsvarianten und Beziehungswissen	248
3.2.5	Konzepte	251
3.2.6	Filter: Auflösung und Konfigurationssimulation	253
3.2.7	Stücklistenkonverter	255
3.2.8	PLM WebUI	258

3.3	Integration der Variantenkonfiguration – der klassische Prozess	265
3.3.1	Vertriebsaktivitäten	266
3.3.2	Bedarfsplanung	268
3.3.3	Steuerung der Bedarfsübergabe	269
3.3.4	Beschaffung: Eigenfertigung oder Fremdbeschaffung	272
3.4	Prozesse mit erweiterten Integrationsaspekten	273
3.4.1	Prozess der Eigenfertigung	275
3.4.2	Qualitätsmanagement und Variantenkonfiguration	278
3.4.3	Bestellung und konfigurierbare Muster-Leistungsverzeichnisse	284
3.4.4	Projektsystem, konfigurierbare Standardnetzpläne und Variantenkonfiguration	287
3.4.5	Kundenservice und konfigurierbare Instandhaltungsanleitungen	291
3.5	Vorplanung und Variantenkonfiguration	296
3.5.1	Exkurs: Auswertungen im Umfeld der Variantenkonfiguration	296
3.5.2	Variantenkonfiguration und Vorplanung	301
3.5.3	Reine Baugruppenvorplanung	302
3.5.4	Merkmalsvorplanung/Standarderzeugnisvorplanung	303
3.5.5	Merkmalsvorplanung/Standarderzeugnisvorplanung mit Langfristplanung	309
3.5.6	Typenvorplanung/Vorplanung mit Vorplanungsvarianten	312
3.5.7	Variantenkonfiguration und SCM APO	317
3.5.8	Vorplanung und SCM APO	320
3.6	Zusammenfassung	321

4 Customizing von SAP ERP für die Variantenkonfiguration 323

4.1	Explizites Customizing der Variantenkonfiguration	323
4.1.1	Pflegeberechtigungen	324
4.1.2	Status	325
4.1.3	Gruppen	327
4.1.4	Konfigurierbare Objekte	327
4.1.5	Konfigurationsoberfläche	328
4.2	Customizing des Klassensystems	329
4.3	Für Variantenkonfiguration relevantes Customizing zu den Geschäftsprozessen	336

4.3.1	Konfigurierbarer Materialstamm	336
4.3.2	Positionstypen und deren Findung	339
4.3.3	Bedarfsarten, Bedarfsklassen und deren Findung	342
4.3.4	Planungsstrategien	345
4.3.5	Änderungsprofile für den Fertigungsauftrags-Änderungsdienst (OCM)	347
4.4	Zusammenfassung	349

5 Spezifika der Produktkonfiguration in SAP CRM 351

5.1	Produktkonfiguration in verschiedenen Kanälen	351
5.2	Konfiguration von Produkten versus Services	352
5.3	Vorgehen bei integrierter Produktion in SAP ERP	353
5.3.1	Verkaufskonfiguration versus Produktionskonfiguration	354
5.3.2	Replikation der Stammdaten aus SAP ERP	355
5.4	Erstellung eines Produktmodells mit Hilfe der PME	357
5.4.1	Wesentliche Eigenschaften und Unterschiede zur Modellierung in SAP ERP	357
5.4.2	Aufruf der PME	358
5.4.3	Produktmodelle versus Wissensbasen	359
5.4.4	Versions- und Statusmanagement	360
5.4.5	Klassen, Merkmale und Werte	361
5.4.6	Beziehungswissen in der PME	362
5.4.7	Transport von Wissensbasen	369
5.5	Benutzeroberfläche des IPC	370
5.5.1	JavaServer Pages und J2EE Engine	370
5.5.2	Extended Configuration Management (XCM)	370
5.6	Besondere Funktionen der IPC-Benutzeroberfläche	371
5.6.1	Bilder und andere Objekte	371
5.6.2	Import/Export von Konfigurationsergebnissen	372
5.6.3	Preisübersicht	372
5.6.4	Bessere Handhabung einschränkbarer Merkmale	374
5.6.5	Suchen/Setzen	374
5.6.6	Anzeige von Langtexten (ab CRM 2006s)	375
5.6.7	Vom Konfigurator gesteuerte Meldungen (ab CRM 2006s)	375
5.6.8	Konfigurationsvergleich (ab CRM 2006s)	376
5.7	UI Designer (ab CRM 7.0)	377
5.8	Zusammenfassung	379

6	Herausforderungen in der Variantenkonfiguration	381
6.1	Performance-Optimierung	382
6.1.1	Performance-Engpässe – Auftreten und Einflussfaktoren	382
6.1.2	Ursachen von Performance-Engpässen	385
6.1.3	Performance-Analyse	388
6.2	Änderungsdienst	390
6.2.1	Engineering Change Management (ECM)	390
6.2.2	Order Change Management (OCM)	403
6.3	Komplexe Systemkonfigurationen	409
6.3.1	Was ist eine Systemkonfiguration?	410
6.3.2	Dynamisierung der Stücklistenstruktur	411
6.3.3	Vernetzte Konfigurationsstrukturen im LO-VC	415
6.3.4	Kompositionsprobleme in SCE Advanced Mode	418
6.4	Stammdatenverteilung mit Product Data Replication (PDR) ...	425
6.4.1	Herausforderung und Möglichkeiten	426
6.4.2	PDR-Komponenten (ALE, Konfigurationsmanagement und Workflow)	428
6.4.3	Ablauf der PDR-Einrichtung	429
6.4.4	Vorbereitungen im System	430
6.4.5	Setup und Customizing der PDR	433
6.4.6	Replikation eines VC-Modells mit der PDR	443
6.5	Zusammenfassung	460
7	Weiterentwicklungen in der SAP-Branchenlösung DIMP ...	463
7.1	Überblick	463
7.2	DIMP – Discrete Industries and Mill Products	464
7.3	Spezielle Anforderungen der Mill-Branche	465
7.3.1	Auftragsabwicklung/Fertigungsszenarien	466
7.3.2	Fertigungsdiskrepanzen – Plankonfiguration und Ist-Konfiguration	466
7.4	Erweiterungen zur Produktkonfiguration in SAP for Mill Products	468
7.4.1	Die Merkmalschnellerfassung – vereinfachte Erfassung von konfigurierbaren Belegpositionen	469
7.4.2	Vererbung in Positionsbelegen – globale und lokale Positionen	472
7.4.3	Übernahme von Vorschlagswerten aus dem Kunden-Material-Infosatz	473
7.4.4	Arbeiten mit Kundenauftragsversionen	474

7.4.5	Variantenkonfiguration in Verbindung mit Lagerfertigung	476
7.4.6	Auftragszusammenfassung mit konfigurierbaren Produkten	478
7.5	Zusammenfassung	479

8 Weiterentwicklungen und Add-ons im SAP-Partnerumfeld 481

8.1	Sybit Model Tester (Firma »Sybit GmbH«)	482
8.1.1	Manuelles Testen – Transaktion CU50	483
8.1.2	Vorteile automatisierter Tests	483
8.1.3	Sybit Model Tester	484
8.1.4	Zusammenfassung	488
8.2	Sybit Configuration Visualizer (Firma »Sybit GmbH«)	488
8.2.1	Problemstellung	489
8.2.2	Sybit Configuration Visualizer	490
8.2.3	Anwendersicht	490
8.2.4	Modellierersicht – das Visualization Modeling Environment	492
8.2.5	Systemsicht	494
8.2.6	Zusammenfassung	495
8.3	VCPowerPack (Firma »AICOMP/VCXI Group«)	496
8.3.1	Wie funktioniert VCPowerPack?	496
8.3.2	VCPowerPack – CoreVC	497
8.3.3	VCPowerPack – SmartVC	497
8.3.4	VCPowerPack – SmartPR	498
8.3.5	VCPowerPack – SmartMD	500
8.3.6	VCPowerPack – Branchenlösungen	500
8.3.7	Projektbeschleunigung	500
8.3.8	Zusammenfassung	501
8.4	it.cadpilot (Firmen »itelligence AG« und »ACATEC Software GmbH«)	501
8.4.1	CAD und SAP – zwei Konfigurationswelten?	502
8.4.2	Aufbau moderner 3-D-CAD-Systeme	502
8.4.3	Steuerung von CAD-Systemen	503
8.4.4	Maximalstückliste der Variantenkonfiguration	504
8.4.5	Architektur	504
8.4.6	Durchführung der CAD-Konfiguration	506
8.4.7	Vorteile der SAP ERP-integrierten CAD-Konfiguration	507
8.4.8	Anwendungsszenarien	508

8.4.9	Weitere Möglichkeiten	510
8.5	Komfortfunktionen für Vertrieb, Marketing und Modellierung (Firma »encoway GmbH«)	511
8.5.1	K-Select	512
8.5.2	K-Assistant	514
8.5.3	K-Connect	515
8.5.4	K-Document	517
8.5.5	Quoteassistent	520
8.5.6	Zusammenfassung der Komfortfunktionen	521
8.6	top flow-Framework und top flow-Varianten-Engine (Firma »top flow GmbH«)	522
8.6.1	Optimierung des Konfigurationsdialogs	522
8.6.2	Funktionserweiterungen	525
8.6.3	Neue Möglichkeiten der Beziehungswissenslogik	526
8.6.4	Prozessoptimierung mit der top flow-Varianten-Engine	528
8.7	Zusammenfassung	529

9 Projektleiter berichten über Projekte und Projektstrukturen 531

9.1	»Wir implementieren SAP!« – Erfahrungsbericht eines Projektleiters	531
9.1.1	Der Marketingrummel und was danach kommt – klären Sie die Voraussetzungen für Ihre Arbeit	532
9.1.2	Analysieren Sie Ihre Geschäftsprozesse und verbessern Sie sie	534
9.1.3	Wie viele Instanzen hätten Sie gerne?	536
9.1.4	Regionaler oder globaler Ansatz?	538
9.1.5	Der Umgang mit Modifikationen am Standard	539
9.1.6	Welche Kompromisse sind tragbar und welche nicht?	540
9.1.7	Wie finden Sie die passende externe Unterstützung?	543
9.1.8	Kommunizieren Sie die Veränderungen erfolgreich	545
9.1.9	Kommunizieren Sie die notwendigen Kompromisse erfolgreich	546
9.1.10	Licht am Ende des Tunnels – schulen Sie Ihre Mitarbeiter	546
9.1.11	Nach dem Projekt ist vor dem Projekt – Probleme nach dem Produktivstart	548
9.1.12	Massendaten ändern	550

9.1.13	Wann ist die Zeit für eine Veränderung gekommen? ..	551
9.2	Rollen in einem Variantenkonfigurationsteam	553
9.2.1	Know-how und Know-how-Träger	553
9.2.2	Zusammensetzung und Aufbau des Projektteams	558
9.3	ASAP für Variantenkonfigurationsprojekte	559
9.3.1	Projektvorbereitungsphase	560
9.3.2	Konzeptionsphase	561
9.3.3	Realisierungsphase	561
9.3.4	Vorbereitung zur produktiven Phase	562
9.3.5	Übergang zur produktiven Phase	563
9.3.6	Golden-Client-Ansatz	563
9.3.7	Besonderheiten bei IPC-Szenarien	566
9.4	Zusammenfassung	568

10 Kunden berichten über die Einführung der SAP-Variantenkonfiguration 569

10.1	Projektverlauf bei Getriebebau NORD	570
10.1.1	Ausgangssituation	571
10.1.2	Maßnahmen	572
10.1.3	Ergebnisse	576
10.1.4	Fazit	578
10.2	Konfigurierbare Materialien bei der Krones AG	580
10.2.1	Projekt	580
10.2.2	Ergebnisse	580
10.2.3	Fazit	583
10.3	Projektverlauf bei der Hauni Maschinenbau AG	584
10.3.1	Personalressourcen	586
10.3.2	Ergebnis	586
10.3.3	Nutzung der Order Engineering Workbench	589
10.4	Variantenkonfiguration bei der Felix Schoeller Gruppe	592
10.4.1	Projekt	593
10.4.2	Ergebnisse	594
10.4.3	Erweiterung der Variantenkonfiguration durch Nutzung des IPC	598
10.4.4	Fazit	600
10.5	SAP bei Hülsta und in der Hüls-Unternehmensgruppe	600
10.5.1	Ausgangssituation	601
10.5.2	Vorbereitung	601
10.5.3	Projektziele und -ergebnisse	602
10.5.4	Fazit	609

10.6	Lenze-Gruppe – Konfiguration gestern, heute und morgen	610
10.6.1	Konfiguration heute – das Projekt EuLe	610
10.6.2	Konfiguration morgen – leistungsfähige Prozessintegration	614
10.7	Zusammenfassung	616
11 Configuration Workgroup		617
11.1	Was ist die CWG?	617
11.2	Zielsetzung und Aufgaben	618
11.3	Entstehungsgeschichte	620
11.4	Organisatorischer Aufbau	622
11.5	CWG-Konferenzen	623
11.6	CWG-Portal	624
11.7	CWG Sandbox-System	625
11.8	Zusammenfassung	626
12 Ausblick auf SAP Business ByDesign		627
12.1	SAP Business ByDesign	628
12.2	Produktkonfiguration im Mittelstand	629
12.3	»Make to Order« in SAP Business ByDesign	631
12.3.1	Erweiterung des Produktbegriffs	631
12.3.2	»Make to Specification«	633
12.3.3	Leichtgewichtige Produktvarianten	634
12.4	Produktkonfiguration in SAP Business ByDesign	635
12.4.1	Produktmodell	635
12.4.2	Produktmerkmale	636
12.4.3	Konfiguratoranbindung	638
12.4.4	Prozessautomatisierung	639
12.5	Zusammenfassung	639
Anhang		641
A	Datenbanktabellen der Variantenkonfiguration	643
B	APIs der Variantenkonfiguration	647
C	User Exits der Variantenkonfiguration	649
D	Vollständige Beispiele für Variantenfunktionen	651
E	Die Autoren	657
	Index.....	659

- ▶ eine Zusammenfassung des Beziehungswissens mit der Möglichkeit, in die Syntax abzuspringen (Ein Doppelklick auf BEZIEHUNGSWISSEN zeigt in einem neuen Fenster Name, Status, Kurztext und die gesamte Syntax des entsprechenden Beziehungswissens an.)
- ▶ eine Auflistung aller Stücklistenauflösungen inklusive des zugeordneten Beziehungswissens
- ▶ eine Auswertung zu den Klassenknoten
- ▶ eine detaillierte Auflistung der Merkmalsbewertung inklusive des zugeordneten Beziehungswissens zu Merkmalen und Werten
- ▶ eine Auswertung zu den Konfigurationsprofilen, auch hier mit Beziehungswissen

2.6 Beziehungswissen für die Bewertungsoberfläche bzw. die vertriebliche Sicht

Beziehungswissen wird, wie bereits beschrieben, für zwei Anwendungen benötigt: für die High-Level-Konfiguration (Vertriebskonfiguration, im Dialog, für die Bewertungsoberfläche) und die Low-Level-Konfiguration (Stücklisten- und Arbeitsplanauflösung, auch ohne Dialog). Erstere soll hier im Detail vorgestellt werden.

2.6.1 Produktmodellierungsumgebung PMEVC

Für die Pflege von Beziehungswissen für die Bewertungsoberfläche gibt es verschiedene Transaktionen bzw. Wege. Die wichtigste Pflegeumgebung für Beziehungswissen für die Bewertungsoberfläche, sowohl für lokales als auch globales, ist die Produktmodellierungsumgebung PMEVC (siehe Abbildung 2.29). Aus diesem Grund steht diese hier auch im Vordergrund.

Andere Pflegemöglichkeiten werden vollständig in Abschnitt 2.7, »Beziehungswissen für Stückliste und Arbeitsplan«, vorgestellt.

Was ist die Produktmodellierungsumgebung PMEVC? Diese Funktionalität steht unter der gleichnamigen Transaktion PMEVC seit dem ERP-Release 5.0 zur Verfügung. Eine ähnliche Funktionalität ist ebenfalls Bestandteil des IPC. PMEVC steht für *Product Modelling Environment Variant Configuration*. Die Idee dieser Transaktion ist es, eine einzige Umgebung zu schaffen, in der das gesamte Variantenmodell aus Sicht der High-Level-Konfiguration aus der Modellstruktur heraus gepflegt werden kann. Sie können sich zunächst, ähn-

lich wie im Variantenmodell-Browser CUMODEL, auch hier einen Überblick über die vorhandene Modellstruktur verschaffen und haben die Möglichkeit, in alle Details abzuspringen.

Zusätzlich kann eine große Anzahl von Bestandteilen des Konfigurationsmodells aus dieser Produktmodellierungsumgebung heraus angelegt und geändert werden. So können alle Arten von Beziehungswissen, sowohl globales als auch lokales, angelegt und geändert werden. Das Gleiche gilt für Konfigurationsprofile, Variantentabellen und die Daten für den IPC.

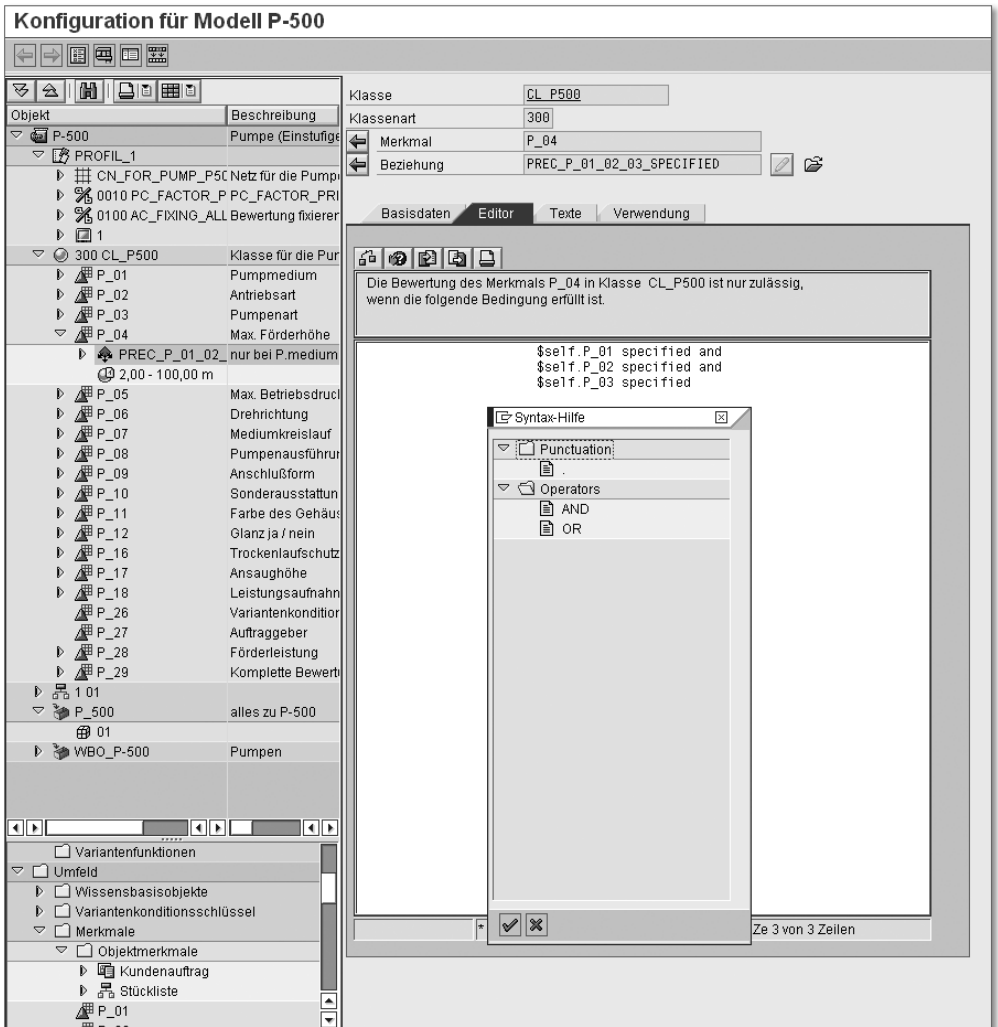


Abbildung 2.29 Produktmodellierungsumgebung PMEVC

Die Produktmodellierungsumgebung nutzt einen erweiterten Editor für die Pflege von Beziehungswissen. Im Unterschied zum klassischen Beziehungswissen-Editor kann hier mit folgenden Elementen gearbeitet werden:

- ▶ kontextsensitive Eingabehilfe
- ▶ Drag & Drop
- ▶ Beziehungswissen-Assistenten für Vor- und Auswahlbedingungen und tabellenbasierte Constraints

Die kontextsensitive Eingabehilfe kann durch die Funktionstasten F4 oder die zweite Drucktaste im Editor aufgerufen werden (siehe Abbildung 2.29). Es werden alle möglichen Syntaxelemente, aber auch Merkmale und Merkmalswerte angeboten, die genau an der entsprechenden Stelle in die Beziehungswissen-Syntax eingefügt werden können.

Die Drag & Drop-Funktionalität erlaubt es, in der PMEVC aus den Listen links (siehe Abbildung 2.29) einzelne Merkmalswerte in den Editor zu übernehmen. Wie das folgende Beispiel aus Listing 2.7 zeigt, werden als Ergebnis sowohl das Merkmal als auch der Merkmalswert als Gleichung übernommen.

```
merkmal1 = 'wert1'
```

Listing 2.7 Beispiel für durch Drag & Drop erzeugte Syntax


Der Beziehungswissen-Assistent erlaubt es, wie bereits erwähnt, Vor- und Auswahlbedingungen und tabellenbasierte Constraints anzulegen, ohne selbst Syntax schreiben zu müssen. Über den Assistenten werden alle notwendigen Informationen abgefragt, und das System legt die Syntax und alle anderen notwendigen Daten an.

Die Produktmodellierungsumgebung PMEVC gestattet nicht nur die komplette Pflege des Beziehungswissens, das für die Bewertungsoberfläche relevant ist. Im Prinzip lässt die PMEVC fast alle Pflegeschritte der Modellierung für die High-Level-Konfiguration zu. Dies umfasst die folgenden Punkte:

- ▶ Pflege (Anlegen, Ändern, Anzeigen, Zuordnen) von Beziehungswissen für Konfigurationsprofil, Merkmale und Merkmalswerte
- ▶ klassenspezifische Merkmalsanpassung
- ▶ einfache Klassifizierung (keine Mehrfachklassifizierung)
- ▶ Anlegen eines Konfigurationsprofils (nicht mehrerer an einem Materialstamm – die Pflege aller vorhandenen Konfigurationsprofile ist aber möglich)

- ▶ Verwendung von Änderungsnummern
- ▶ Pflege der Variantentabellenstruktur
- ▶ Pflege des Variantentabelleninhalts
- ▶ Ändern von Beziehungswissen an der Stückliste
- ▶ Anlegen von Wissensbasis und Laufzeitversion für den IPC
- ▶ materialspezifisches Aktivieren des IPC als Konfigurator auch in ERP
- ▶ Oberflächendesign-Pflege
- ▶ Pflege und Zuordnung von Variantenbedingungen für die Preisfindung

Mit dem Erweiterungspaket 5 (EHP5) des ERP-Releases 6.0 wird voraussichtlich zusätzlich die Möglichkeit geschaffen, nicht nur vorhandenes Beziehungswissen an der Stückliste zu ändern, sondern auch neues Beziehungswissen anzulegen. Dabei kann auch mit Drag & Drop gearbeitet werden.

Weiterhin wird voraussichtlich die Simulationsfunktionalität erweitert. Startet man die Simulation aus der PMEVC heraus (Button  in Abbildung 2.29), erscheint zunächst ein Fenster zur Bewertung der Objektmerkmale. Man kann hier in der Simulation abhängig vom Konfigurationsszenario – damit ist an dieser Stelle die Unterscheidung zwischen Kundenauftrag und Materialvariante gemeint – alle relevanten Objektmerkmale bewerten, bevor das eigentliche Bewertungsbild gestartet wird.

Neben diesen beiden Punkten (Anlegen von neuem Beziehungswissen für Stücklistenpositionen und Simulation mit Objektmerkmalen), wird ab EHP5 voraussichtlich folgende zusätzliche Funktionalität der PMEVC angeboten:

- ▶ Anlegen neuer Merkmalswerte
- ▶ Anlegen neuer Bezeichnungen und Langtexte in weiteren Sprachen für Merkmale und Merkmalswerte
- ▶ Einfachere Pflege von Merkmalsgruppierungen im Rahmen des Oberflächendesigns, inklusive Drag & Drop-Funktionalität
- ▶ Detailsicht für Stücklisten und Stücklistenpositionen

Über die PMEVC können nur wenige Stammdaten des Variantenkonfigurationsmodells gepflegt werden. Das heißt, es wird vorausgesetzt, dass die folgenden Stammdaten bereits vorher über klassische Transaktionen angelegt worden sind:

- ▶ Merkmale
- ▶ Klassen

- ▶ Materialstämme
- ▶ Stücklisten inklusive aller Objekte auf Positionsebene
- ▶ Arbeitspläne inklusive aller Objekte auf Vorgangsebene
- ▶ Variantenfunktionen
- ▶ Änderungsnummern
- ▶ vor Release ERP 6.0 noch nicht in PMEVC anlegbare Objekte

Nach dieser Vorstellung der PMEVC wollen wir ihre Anwendung an einem ersten Beispiel zeigen.

2.6.2 Ein erstes Beispiel

Im Anschluss an diese eher theoretischen Erläuterungen erscheint es uns sinnvoll, ein praktisches Beispiel zur Nutzung der PMEVC für das Anlegen von Beziehungswissen für die Bewertungsoberfläche zu geben. Dazu soll der Beziehungswissen-Assistent der PMEVC genutzt werden.

Für die Abbildung von Abhängigkeiten zwischen den einzelnen Merkmalen der Bewertungsoberfläche empfiehlt es sich, mit Tabellen bzw., wenn sie nicht zu groß werden, mit Variantentabellen zu arbeiten. Tabellen haben den Vorteil, dass in ihnen die Art der Abhängigkeiten leichter ablesbar ist als durch direkte Auswertung der Syntax des Beziehungswissens. Weiterhin hat die Verwendung von Variantentabellen einen großen Vorteil, wenn das Modell »lebt«: Ändern sich die Abhängigkeiten im Modell, muss nur der Inhalt der Variantentabelle geändert werden, ohne nochmals die Syntax des Beziehungswissens bearbeiten zu müssen.

Um die Tabelle auszuwerten, schreiben Sie am elegantesten Beziehungswissen, das die Tabelle abfragt und nur Bewertungen zulässt, die mit der Tabelle konform laufen. Dies soll so geschehen, dass gar keine unerlaubte Bewertung möglich ist. Die Liste der zulässigen Werte pro Merkmal soll dynamisch so eingeschränkt werden, dass nur noch erlaubte Bewertungen möglich sind. Bei der Auswahl von solch »eleganter« Beziehungswissen wird eine Art verwendet, bei der der Benutzer keine Aussage zum Abarbeitungszeitpunkt machen muss. Die Wahl fällt damit auf Constraints.

In Abbildung 2.30 wurde die PMEVC mit dem Material T-VPC aufgerufen. Zu dem Material wurde die Stückliste gefunden. Dies sind zunächst die beiden einzigen Einträge in der Struktur ❶. Es werden nun zumindest noch die Variantenklasse und das Konfigurationsprofil benötigt. Wie bereits erwähnt, muss die Variantenklasse inklusive ihrer Merkmale außerhalb der PMEVC

angelegt werden. In Abbildung 2.30 wurde die vorhandene Variantenklasse bereits zugeordnet.

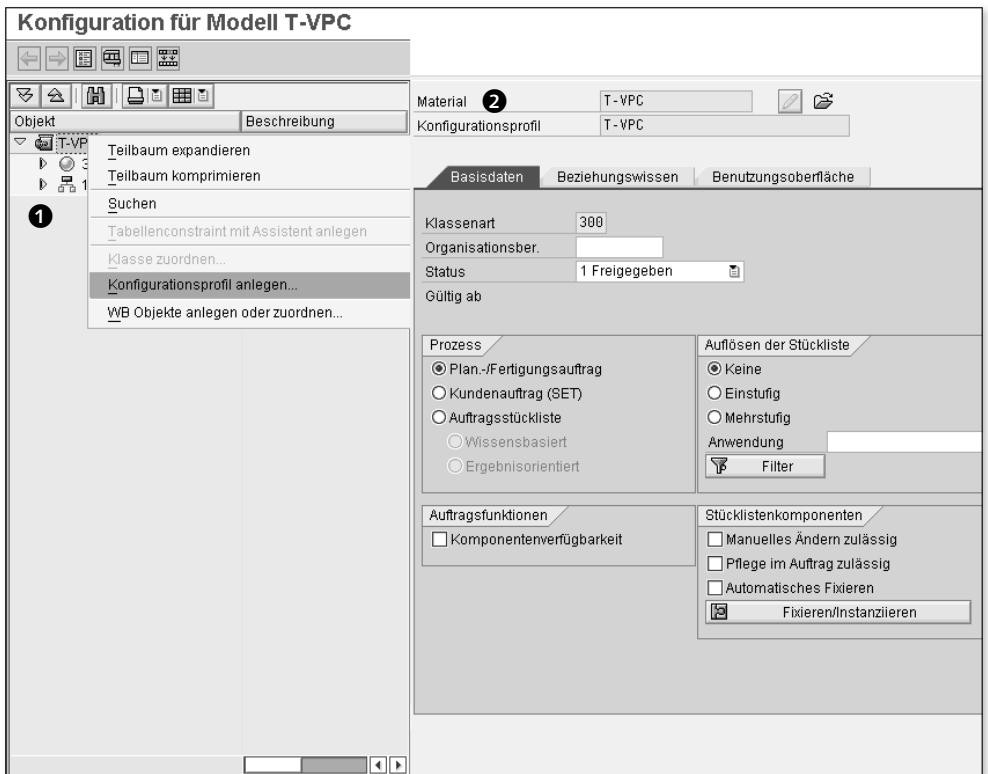


Abbildung 2.30 Erste Schritte in der PMEVC: Klassenzuordnung und Anlegen eines Konfigurationsprofils

Diese Variantenklasse – oder auch gleich eine ganze Gruppe von Variantenklassen – wurde vorher in das Umfeld der PMEVC aufgenommen (siehe Abbildung 2.31 links unten, hier unter UMFELD • KLASSEN und Kontextmenü). Anschließend ordnen Sie die Variantenklasse einfach per Drag & Drop zu.

Im Unterschied zur Variantenklasse kann das Konfigurationsprofil direkt aus der PMEVC heraus angelegt werden. Wie in Abbildung 2.30 unter ② zu sehen, steht diese Funktionalität aus dem Kontextmenü auf der Ebene des Materialstamms zur Verfügung. Analog zur klassischen Transaktion der Anlage von Konfigurationsprofilen können Sie mit den Default-Werten bereits ein Konfigurationsprofil anlegen. Im Unterschied zur klassischen

Transaktion gibt es sogar Default-Werte für den Namen und die Variantenklassenart.

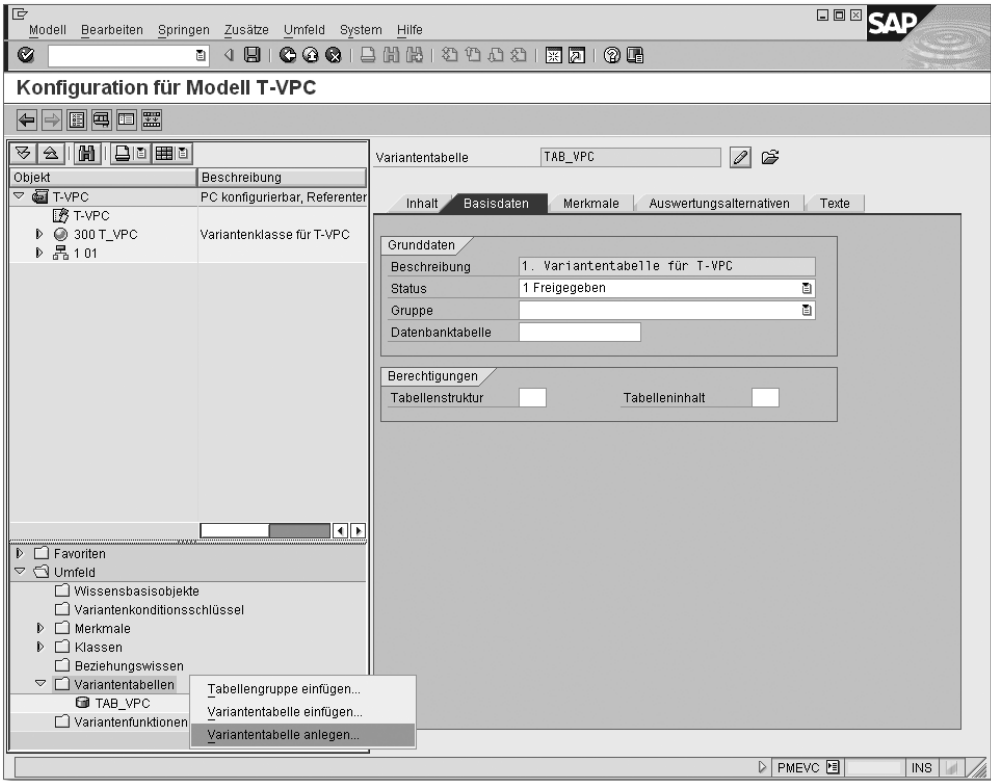


Abbildung 2.31 Variantentabellen – Anlegen und Inhaltspflege mit der PMEVC

Die Modellstruktur besteht nach den beiden obengenannten Schritten aus vier Objekten: dem Materialstamm, dem Konfigurationsprofil, der Variantenklasse und der Stückliste.

Eine Variantentabelle (siehe Abbildung 2.31) soll die erlaubten Kombinationen für die Bewertung einer Reihe von Merkmalen abbilden. In unserem Beispiel nehmen wir an, es seien die drei Merkmale »Sonderwunsch«, »Gehäuse« und »CPU«. Hierzu müssen Sie im Detail folgende Schritte abarbeiten:

1. Variantentabelle anlegen

Im ersten Schritt legen Sie über das Kontextmenü im Umfeld die Variantentabelle an, da sie in diesem Beispiel noch nicht existiert.

2. Name der Tabelle

Damit öffnet sich ein Fenster VARIANTENTABELLE ANLEGEN, in dem Sie der Tabelle einen Namen geben. Der Änderungsdienst ist optional und soll in diesem Beispiel nicht genutzt werden.

3. Beschreibung der Tabelle

Anschließend erscheint die Detailsicht (siehe Abbildung 2.31) mit den fünf Registerkarten. Auf der Registerkarte BASISDATEN tragen Sie eine Beschreibung (sprachabhängig) ein und geben dann die Variantentabelle frei.

4. Merkmale zuordnen

Auf der Registerkarte MERKMALE werden nur die drei genannten Merkmale in beliebiger Reihenfolge angegeben.

5. Tabelleninhalt eingeben

Abschließend können Sie auf der Registerkarte INHALT als Zeilen die zulässigen Kombinationen der Bewertung bezüglich der drei Merkmale eingeben.

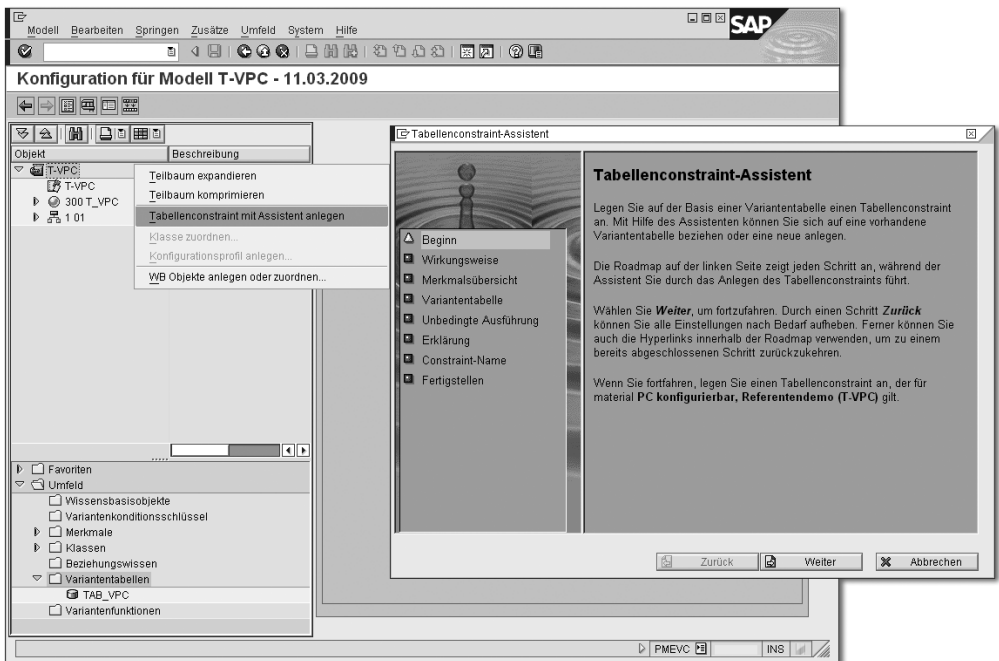


Abbildung 2.32 Anlegen von Beziehungswissen mit Hilfe des Tabellenconstraint-Assistenten

Nach dem Anlegen und der Inhaltspflege der Variantentabellen wird nun noch Beziehungswissen benötigt. Das Beziehungswissen soll die Tabelle

lesen und die Wertelisten der entsprechenden Merkmale immer dynamisch so einschränken, dass nur Bewertungen aus der Tabelle möglich sind. Dazu wird der Tabellenconstraint-Assistent – wie in Abbildung 2.32 erkennbar – über das Kontextmenü am Konfigurationsprofil gestartet.

Der Assistent führt Sie durch die einzelnen Schritte der Anlage von Constraints und fragt alle notwendigen Informationen ab. Dabei kann es sein, dass er die Schritte dynamisch den bereits gegebenen Informationen anpasst. Folgende Schrittfolge arbeitet der Assistent ab:

1. **Beginn**

Der erste Schritt des Assistenten informiert über die weitere Vorgehensweise, um ein Constraint mit Bezug auf eine Variantentabelle anzulegen.

2. **Wirkungsweise**

Im zweiten Schritt wird die Wirkungsweise abgefragt. Hier sollen Wertelisten für Merkmale eingeschränkt werden. Dies stellt in diesem Schritt die erste Auswahlmöglichkeit dar. Der Unterschied zu den anderen Auswahlmöglichkeiten wird weiter unten beschrieben.

Es kann passieren, dass die Auswahlmöglichkeit WERTEEINSCHRÄNKUNG nicht angeboten wird. In dem Fall ist die notwendige Voraussetzung nicht gegeben, dass die Merkmale, deren Wertelisten eingeschränkt werden sollen, auch als einschränkbar in der Merkmalsdefinition gekennzeichnet sind.

3. **Konfigurationsobjekt**

Anschließend wird das Konfigurationsobjekt abgefragt. In Constraints können neben Merkmalen aus Klassen auch andere Bezugsobjekte genutzt und ausgewertet werden. Da im hier vorliegenden Beispiel Merkmale einer Variantenklasse eingeschränkt werden sollen, ist die entsprechende Variantenklasse das Konfigurations- oder auch Bezugsobjekt.

4. **Variantentabelle**

Nach dem Konfigurationsobjekt wird die Variantentabellenselektion durchgeführt. Da in dem konkreten Beispiel nur eine Variantentabelle im Umfeld der PMEVC existiert, ist keine größere Auswahl möglich.

5. **Erklärung**

Im Schritt ERKLÄRUNG kann dem Beziehungswissen, genauer gesagt, dem Tabellenconstraint, ein sprachabhängiger Langtext zugeordnet werden. Weiterhin sind die Merkmale der Variantentabelle aufgelistet, um eventuell Merkmale für das Beziehungswissen auszuschließen.

6. Constraint-Name

Für Constraints muss der Name extern, also vom Benutzer, vergeben werden. Diese Aufforderung inklusive eines Kurztextes umfasst der Schritt CONSTRAINT-NAME.

7. Fertigstellen

Damit sind alle Schritte komplett. Dies wird nochmals ausführlich beschrieben, bevor der Benutzer das Tabellenconstraint fertigstellen kann.

Das fertiggestellte Tabellenconstraint wird abschließend, wie in Abbildung 2.33 zu sehen, dargestellt. Nach dem Speichern können Sie es sofort testen, indem Sie die Konfigurationssimulation aus der PMEVC über den Button TESTEN aufrufen.

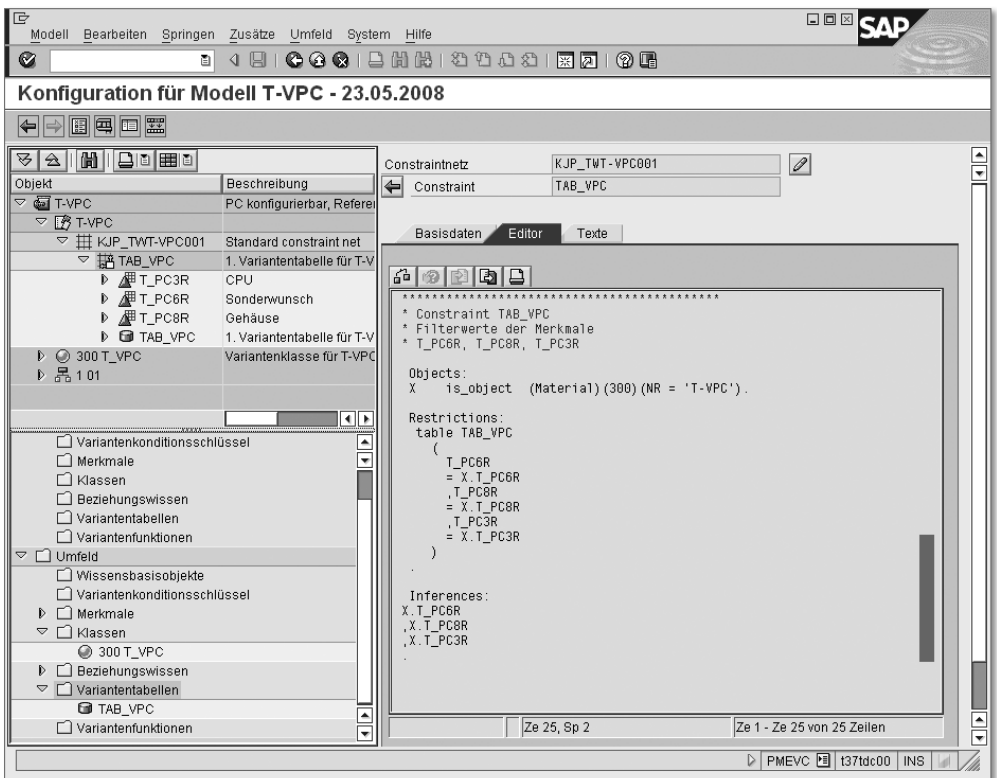


Abbildung 2.33 Ergebnis des Tabellenconstraint-Assistenten

In der Bewertungsoberfläche erscheinen die drei Merkmale »Sonderwunsch«, »Gehäuse« und »CPU«. Zunächst sind pro Merkmal über die jeweilige Eingabehilfe (Taste **F4**) alle Werte aus der Variantentabelle im Ange-

bot. Beginnt man jedoch bei einem beliebigen der drei Merkmale mit der Bewertung, werden sofort für die beiden anderen Merkmale nur die Werte angeboten, die laut Variantentabelle noch zu einer erlaubten Bewertung führen. Analoges gilt für zwei bewertete Merkmale.

Eine Besonderheit des klassischen Konfigurators ist, dass, wenn der zulässige Wertebereich für ein Merkmal auf einen Wert eingeschränkt wird, dieser Wert automatisch auch gesetzt wird. Dies geschieht im Konfigurator des IPC nur für Muss-Merkmale. Zu beachten ist, dass die Merkmale laut Merkmalsdefinition einschränkbar sein müssen. Dies hat zusätzlich zur Folge, dass die Listen der zulässigen Werte für bewertete Merkmale, wie bereits beschrieben, auf die Bewertung eingeschränkt werden.

2.6.3 Variantentabellen im Detail

Eine erste Anwendung von Variantentabellen wurde im obigen Beispiel bereits dargestellt. Variantentabellen können in allen Arten von Beziehungswissen angesprochen werden. Die Spalten von Variantentabellen sind immer Merkmale. Die Zeilen stellen Bewertungskombinationen dar. Variantentabellen können dabei für verschiedene Zwecke genutzt werden:

- ▶ Werteeinschränkungen in Constraints (hier kann es auch bereits zu Wertsetzungen kommen – wie im Beispiel beschrieben)
- ▶ Wertehierleitungen in Constraints oder Prozeduren
- ▶ Bedingungen als Vorbedingungen, Auswahlbedingungen, If-Bedingung in Prozeduren oder Constraints und als Condition-Teil in Constraints
- ▶ Konsistenzprüfungen durch Constraints

Diese Verwendungszwecke werden bei der Vorstellung der entsprechenden Arten von Beziehungswissen noch genauer beschrieben.

Hier sollen der Aufbau und die Pflege von Variantentabellen vorgestellt werden (siehe Abbildung 2.31 und Abbildung 2.34). Die Pflege der Struktur und des Inhalts der Variantentabelle kann über die PMEVC erfolgen. Es gibt auch eigene Transaktionen für die folgenden Punkte:

- ▶ Inhaltspflege (CU60)
- ▶ Anlegen, Ändern und Anzeigen der Tabellenstruktur (CU61 bis 63)

Die Registerkarte BASISDATEN der Variantentabellen – siehe Abbildung 2.31 – enthält Namen, Beschreibung (sprachabhängiger Kurztext), Status und Gruppe. Der Status kann über Customizing angepasst werden und umfasst

unter anderem die Inhaltspflege und die Verwendung in Beziehungswissen sowie Verteilsperren für Inhalt und Struktur.

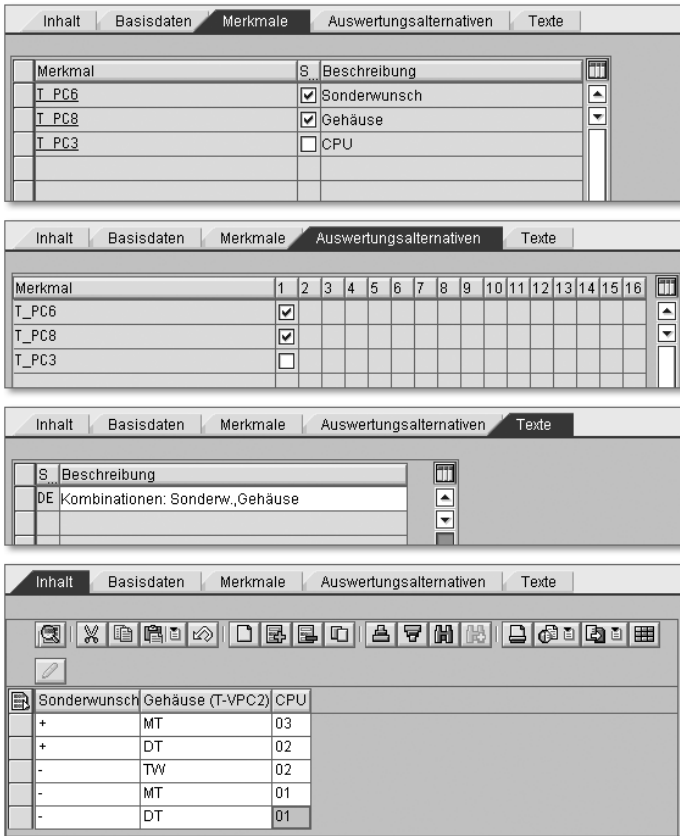


Abbildung 2.34 Aufbau und Pflege der Variantentabelle (PMEVC)

Für die Gruppe gilt Analoges wie für Merkmals- und Klassengruppen, sie stellt auch eine separate Liste aus dem Customizing dar. Weiterhin kann in den Basisdaten die Variantentabelle mit einer Datenbanktabelle gekoppelt werden. Darauf werden wir später noch eingehen. Vervollständigt werden die Basisdaten durch Berechtigungsgruppen für die Inhalts- und Strukturpflege.

Die Registerkarte MERKMALE, die einer Variantentabelle zugeordnet wird, stellt die Spalten der Variantentabelle dar. Für die Pflege des Tabelleninhalts und für die Verwendung in Beziehungswissen sind Einstellungen in der Merkmalsdefinition wie ein-/mehrwertig, einschränkbar, Muss-Merkmal, Vorschlagswerte oder Beziehungswissen irrelevant. In der Merkmalsicht

kann ein erster Schlüssel definiert werden. Dieser Schlüssel ist nur Voraussetzung und von Bedeutung, wenn über die Variantentabelle Werte hergeleitet werden sollen. Dies kann mit Constraints oder Prozeduren erfolgen. Für Werteeinschränkungen, Bedingungen oder reine Konsistenzprüfungen werden Schlüsselinformationen ignoriert.

Constraints mit Inferences-Teil können zur Wertehrerleitung auch mehr als die in der Registerkarte dargestellte Auswertungsalternative (Schlüssel auf der Registerkarte MERKMALE) auswerten. Diese zusätzlichen Alternativen können in der gleichnamigen Sicht angelegt werden.

In der PMEVC können die Elemente der Registerkarte INHALT auch direkt in der Variantentabelle gepflegt werden.

Neben der Inhaltspflege aus der PMEVC heraus kann der Inhalt auch über die klassische Transaktion CU60 erfolgen. Diese Transaktion erlaubt neben der Standarddarstellung (siehe ❶ in Abbildung 2.35) eine Darstellung als Matrix ❷ und als Aufzählung ❸. Mit Hilfe der letzten beiden Darstellungen kann recht einfach entschieden werden, welche Kombinationen aufgenommen werden sollen (Entscheidungstabelle).

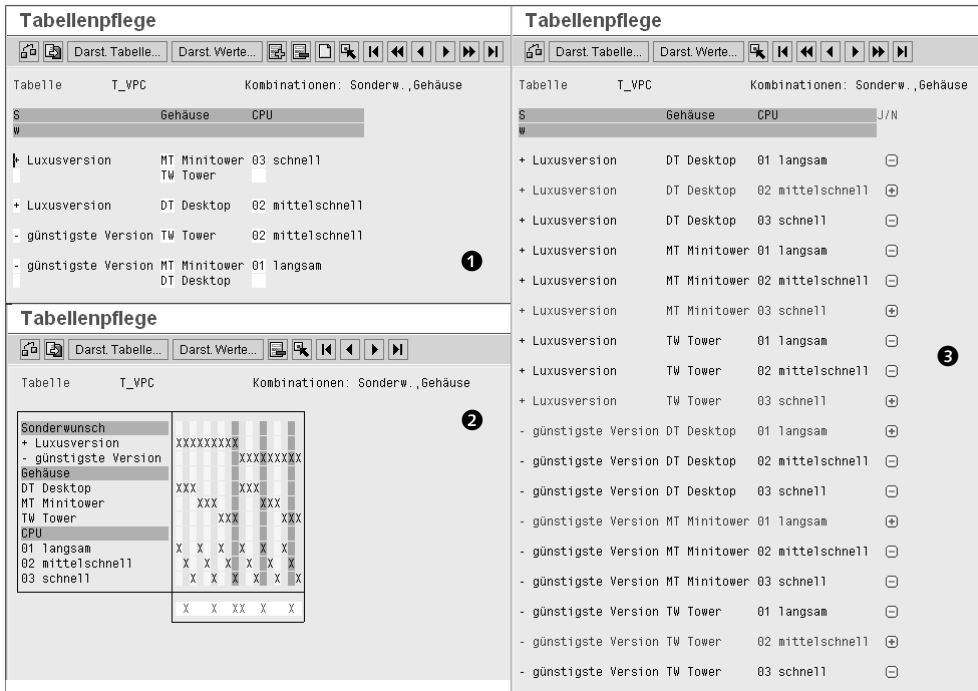


Abbildung 2.35 Variantentabellen – Inhaltspflege mit der klassischen Transaktion

IPC-kompatibler Tabelleninhalt



In der Standarddarstellung können auch mehrere Werte pro Feld ausgewählt werden. Diese Darstellung ist nicht IPC-kompatibel, kann jedoch über den entsprechenden Menüeintrag in eine solche überführt werden (Ausmultiplizieren, Normalisieren).

Neben den hier bereits vorgestellten Möglichkeiten, den Inhalt von Variantentabellen mit Hilfe der PMEVC oder auch der klassischen Transaktion CL60 zu füllen, gibt es auch die Möglichkeit, den Inhalt aus Excel-Tabellen zu importieren. Hierfür existiert die Transaktion CU60E.

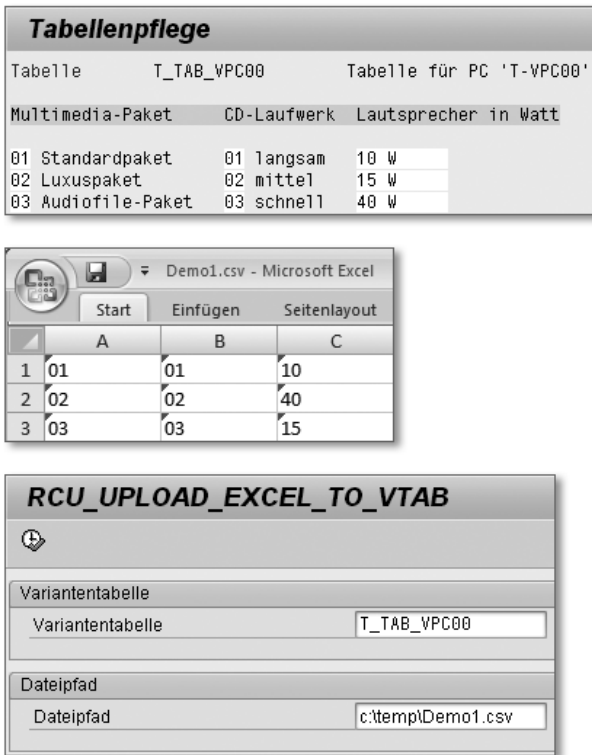


Abbildung 2.36 Transaktion CU60E – Datenimport aus Excel-Tabellen in Variantentabelle

Um den Datenimport über die Transaktion CU60E durchzuführen, wird im ERP-System eine existierende Variantentabelle benötigt. Das heißt, es sind zunächst die benötigten Merkmale und mit Hilfe dieser Merkmale die Struktur der Variantentabelle anzulegen.

Im Beispiel in Abbildung 2.36 gehen wir davon aus, dass als Ergebnis des Datenimports eine Variantentabelle, wie in Abbildung 2.36 oben dargestellt, entsteht. Die genannten Vorbereitungen im ERP-System umfassen das Anlegen der drei hier genutzten Merkmale (Multimediapakete, CD-Laufwerk und Lautsprecher) sowie der Struktur der Variantentabelle (T_TAB_VPC00).

Beim Anlegen und Füllen der Excel-Tabelle sind folgende Punkte zu beachten: Die Spalten der Excel-Tabelle und ihre Reihenfolge müssen genau der der Variantentabelle entsprechen, und alle Werte in der Excel-Tabelle müssen vom Format *Text* sein. Dies gilt auch für numerische Merkmale in der Variantentabelle. Die Excel-Tabelle darf keine Überschriftenzeile enthalten. Auch dürfen keine Bezeichnungen in die Excel-Tabelle aufgenommen werden – nur die sprachneutralen Merkmalswerte. Beim Datenimport wird das Format der Werte geprüft und die Transaktion gegebenenfalls abgebrochen. Das Excel-File muss als *csv*-File gespeichert werden, wie in Abbildung 2.36 erkennbar. Das heißt, als Trennzeichen werden bei diesem Format der Excel-Files Semikolons genutzt.

Die Transaktion CU60E benötigt als Eingaben nur den Namen der Variantentabelle und den Namen und die Adresse des Excel-Files. Der Datenimport wird durch den Button AUSFÜHREN () gestartet. Es ist nur ein vollständiger Datenimport möglich, eine Art Change-Upload existiert nicht. Ein eventuell vorhandener Inhalt der Variantentabelle wird gelöscht und die Variantentabelle neu gefüllt.

[+]

Transaktion CU60E

Im Zusammenhang mit der Transaktion CU60E möchten wir auf den SAP-Hinweis 516885 verweisen. Dieser enthält als Anlage auch eine Dokumentation zu dieser Transaktion.

Wie bereits erwähnt, können Variantentabellen in allen Arten von Beziehungswissen eingesetzt werden. Die Syntax hat dabei im Prinzip immer das gleiche Aussehen (siehe Listing 2.8).

```
table <Tabellenname> (<Spaltenmerkmal>=<Oberflächenmerkmal>,...)
```

Listing 2.8 Prinzip der Syntax von Variantentabellen-Aufrufen

Diese optische Ähnlichkeit können Sie in die folgenden Punkte aufschlüsseln:

- ▶ Der Aufruf der Variantentabelle beginnt mit dem Schlüsselwort `table`.
- ▶ Es folgt der Name der Tabelle.

- ▶ In Klammern folgen nun alle Spalten der Variantentabelle, die ausgewertet werden sollen. Es muss nicht immer die ganze Tabelle ausgewertet werden. Da die Merkmale der Tabelle nicht mit den Merkmalen der Oberfläche übereinstimmen müssen, wird nach jedem Spaltenmerkmal das entsprechende Merkmal der Bewertungsoberfläche zugeordnet.

In Abbildung 2.33 ist ein Beispiel eines solchen Tabellenaufrufs in Constraints dargestellt. Zu beachten ist, dass den Merkmalen aus der Bewertungsoberfläche ein `X` vorangestellt wurde. Damit wird Bezug auf das Material T_VPC genommen, wie ebenfalls im Constraint direkt über dem Tabellenaufruf zu erkennen ist. Wie bereits im letzten Kapitel erwähnt, können Sie außer in Constraints nicht so flexibel Objekte – wie hier den Materialstamm – ansprechen. In dem Fall müssen Sie sich auf die Ebenen `$self.`, `$root.` oder `$parent.` beschränken. Der Tabellenzugriff aus Abbildung 2.33 würde dann folgendermaßen aussehen:

```
table TAB_VPC ( T_PC6R = $self.T_PC6R,
                T_PC6R = $self.T_PC6R,
                T_PC6R = $self.T_PC6R)
```

Listing 2.9 Tabellenzugriff in Vorbedingungen, Auswahlbedingungen und Prozeduren

Merkmale der Bewertungsoberfläche, für die Werte hergeleitet werden sollen, müssen mit `$self.` verknüpft sein. Ansonsten sind auch `$parent.` und `$root.` erlaubt.



Abbildung 2.37 Kopplung von Varianten- und Datenbanktabellen

Im Rahmen der Pflege der Variantentabellen soll noch auf die Kopplung mit Datenbanktabellen eingegangen werden (siehe Abbildung 2.37). Hintergrund dieser Funktion ist der Wunsch, im Beziehungswissen auch Datenbanktabellen anzusprechen. Dies ist jedoch nicht direkt im Beziehungswissen möglich. Es gibt zwei Möglichkeiten, Datenbanktabellen im Beziehungswissen anzusprechen.

- ▶ **Datenbanktabelle direkt ansprechen (Variantenfunktion)**

Man legt eine Variantenfunktion an und spricht in dem mit der Variantenfunktion korrespondierenden Funktionsbaustein die Datenbanktabelle an. Die Variantenfunktion wird im Beziehungswissen verwendet.

► **Datenbanktabelle indirekt ansprechen (gekoppelte Variantentabelle)**

Man legt eine passende Variantentabelle an und koppelt diese mit der Datenbanktabelle. Ist diese Kopplung aktiv, kann in Beziehungswissen beliebiger Art die Variantentabelle direkt aufgenommen werden, wobei hierbei die Datenbanktabelle angesprochen wird.

Die zweite Möglichkeit nutzt in der Pflege in den Basisdaten der Variantentabellenstruktur eine Zuordnung der Datenbanktabelle. Auch hier können wieder zwei Szenarien unterschieden werden.

1. Szenario: Ausgangspunkt »Datenbanktabelle«

Eine vorhandene Datenbanktabelle soll über diese Technik angesprochen werden. In dem Szenario müssen passende Merkmale, also Merkmale mit passendem Format, angelegt werden. Sie müssen nur Merkmale für die Felder, d.h. die Spalten der Datenbanktabelle, anlegen, die im Beziehungswissen angesprochen werden sollen. Anschließend legen Sie eine Variantentabelle an, die als Spalten genau diese Merkmale enthält. In die Basisdaten der Variantentabelle müssen Sie den Namen der Datenbanktabelle aufnehmen. Dadurch ergibt sich die Möglichkeit einer Feldzuordnung. Hier werden alle Merkmale der Variantentabelle mit den Spalten der Datenbanktabelle verknüpft. Abschließend wird die Kopplung aktiv gesetzt. Somit kann in allen Arten von Beziehungswissen die Datenbanktabelle ausgewertet werden, indem in der Syntax die gekoppelte Variantentabelle angesprochen wird.

2. Szenario: Ausgangspunkt »Variantentabelle«

Eine vorhandene Variantentabelle, die eventuell auch bereits in Beziehungswissen genutzt wird, soll in eine Datenbanktabelle umgewandelt werden. Dies wird z.B. dann genutzt, wenn Variantentabellen eine Größe erreichen, die zu Problemen mit der Performance führt. Es kann aber auch andere Gründe geben.

Hier müssen Sie eine passende Datenbanktabelle anlegen. Die Felder der Tabelle müssen bezüglich des Formats so angelegt werden, dass sie mindestens die Werte der entsprechenden Merkmale aufnehmen können. Die Schlüsselfelder der Datenbanktabelle haben keine Bedeutung für die Variantenkonfiguration. Für die Auswertung des Beziehungswissens ist ausschließlich die Schlüsselkombination der Variantentabelle entscheidend. Natürlich muss der Schlüssel der Datenbanktabelle so gewählt werden, dass für den späteren Inhalt die Eindeutigkeit gewährleistet ist.

Im nächsten Schritt müssen Sie analog zur Beschreibung im ersten Szenario die Kopplung der Variantentabelle mit der noch leeren Datenbanktabelle, inklusive Aktivierung und Feldzuordnung, vornehmen. Abschließend wird der Inhalt der Variantentabelle mit Hilfe der gleichnamigen Transaktion CU59 in die Datenbanktabelle transferiert. In diesem zweiten Szenario muss das gesamte Beziehungswissen, das diese Variantentabelle verwendet, nicht angepasst werden. Es hat unverändert die gleiche Funktionalität wie vorher. Zu beachten ist hier, dass der Inhalt der Variantentabelle nach dem Aktivsetzen der Kopplung inaktiv ist.

Im Standard wird man den Inhalt direkt in der Datenbanktabelle pflegen. Die Inhaltspflege der Variantentabelle kann über den Status gesperrt werden. Auch könnte der Inhalt der Variantentabelle gelöscht werden, um Missverständnissen vorzubeugen.

Theoretisch wäre auch eine weitere Pflege in der Variantentabelle mit einem Delta-Transfer dieser Daten in die Datenbanktabelle möglich. Dies ist aber kaum praktikabel, da beim Delta-Transfer keine Zeilen gelöscht werden können und es auch Probleme durch den Schlüssel der Datenbanktabelle gibt.

2.6.4 Constraints im Detail

Alle Arten von Beziehungswissen können, wie bereits erwähnt, für die Beeinflussung der Konfiguration in der Bewertungsoberfläche genutzt werden. Die Constraints stellen dabei die bedeutendste und die Standardart von Beziehungswissen dar. Dies ergibt sich aufgrund der Tatsache, dass sie für fast alle Aufgaben einsetzbar sind. So kann man mit Hilfe von Constraints die folgenden Punkte abhandeln:

- ▶ Werte setzen
- ▶ Werte prüfen (in Form von Konsistenzchecks)
- ▶ beliebige Objekte auch in der mehrstufigen Konfiguration ansprechen (nicht nur `$self`, `$parent` oder `$root`)
- ▶ performant arbeiten
- ▶ deklarativ arbeiten (was bedeutet, dass keine Abarbeitungsreihenfolge oder etwas Ähnliches bei der Modellierung beachtet werden muss)

Constraints werden in Beziehungsnetzen (auch Constraint-Netze genannt) gesammelt. Die Beziehungsnetze werden generell dem Konfigurationsprofil zugeordnet. Dabei spielt es keine Rolle, ob einem Beziehungsnetz eine große Anzahl von Constraints zugeordnet wird. Die Anzahl von Beziehungsnetzen

an einem Konfigurationsmodell sollte gering gehalten werden. Dabei ist es nicht immer realisierbar, nur ein Beziehungsnetz am Konfigurationsprofil des Kopfmaterials des Konfigurationsmodells zuzuordnen.

[zB]

Notwendigkeit mehrerer Constraint-Netze

Werden z. B. einzelne konfigurierbare Baugruppen aus dem Gesamtmodell einzeln verkauft, müssen alle Constraints, die diese Baugruppe betreffen, auch über ein Beziehungsnetz dem Konfigurationsprofil dieser Baugruppe zugeordnet werden.

Constraints sind innerhalb des Beziehungsnetzes nur lokales Beziehungswissen, während das Beziehungsnetz selbst global ist. Zu beachten ist, dass beide einen Status besitzen. Constraints sind somit nur aktiv, wenn folgende Bedingungen erfüllt sind:

- ▶ Das Constraint ist freigegeben.
- ▶ Das Beziehungsnetz ist freigegeben.
- ▶ Das Beziehungsnetz ist einem Konfigurationsprofil zugeordnet, das während der Konfiguration berücksichtigt wird.

Ein Constraint besteht aus mindestens zwei, maximal vier Teilen, wobei der Objects- und der Restriction-Teil verpflichtende Teile sind. Im Folgenden erläutern wir die vier Teile ihrer Reihenfolge nach:

1. Objects

Dieser Teil dient der Deklaration aller verwendeten Klassen und Objekte sowie der Definition von Variablen.

2. Condition

Dieser Teil dient zur (optionalen) Angabe einer zentralen Bedingung, unter der nur das Constraint ausgewertet werden soll.

3. Restriction

Dieser Teil dient zur Angabe von Gleichungen, Tabellen und Funktionen für die Werteverleitung und/oder -prüfung.

4. Inferences

Dieser Teil dient zur (optionalen) Erweiterung der Herleitung und Einschränkung von Merkmalswerten.

Der Objects-Teil ist ein Pflichtteil und muss die Objekte enthalten, die im weiteren Constraint angesprochen werden. Objekte können Klassen, Materialstämme und Dokumente sein. Die Deklaration hat folgende Syntax:

► **Klasse**

(<Klassenart>Klassenname, z.B. (300)T_VPC

► **Materialstamm**

(Material)(<Klassenart>)(Nr = '<Materialnummer>'),
z.B. (Material)(300)(Nr = 'T-VPC')

► **Dokument**

(Document)(<Klassenart>)(Type = '<Dokumentenart>', Version = '<Dokumentenversion>', Part = '<Dokumententeil>' Nr = '<Dokumentennummer>'),
z.B. (Document)(300)(Type='DRM',Version='01',Part='000', Nr='T-VPC')

Im Objects-Teil können weiterhin lokal für das Constraint Variablen definiert werden, das Constraint in Abbildung 2.33 nutzt eine solche Objektvariable. Objektvariablen für Klassen werden durch `is_a` deklariert, für sonstige Objekte wie in Abbildung 2.33 durch `is_object`. Das Constraint in Abbildung 2.33 hätte ohne Variablen folgende Gestalt (siehe Listing 2.10):

```
objects:      (300)t_vpc
restrictions: table tab_vpc ( t_pc6r = (300)t_vpc.t_pc6r,
                             t_pc8r = (300)t_vpc.t_pc8r,
                             t_pc3r = (300)t_vpc.t_pc3r).
```

Listing 2.10 Beispiel-Constraint: Tabellenaufruf ohne Nutzung von Variablen

Hier sollten Sie beachten, dass anstelle des Materialstamms im Objects-Teil auch die Variantenklasse, wie hier geschehen, deklariert werden kann. Die Syntax hat hinter den Gleichheitszeichen die Gestalt `<Objekt>.<Merkmal>`.

Im Objects-Teil können Sie weiterhin Merkmalsvariablen definieren. Diese gelten klassenspezifisch. Damit können Sie in diesem Fall auf Objektvariablen verzichten. In der Syntax geschieht dies durch `where` und die Aufzählung der Variablen, die mit Semikolon getrennt werden. Das Constraint in Abbildung 2.33 hätte mit Merkmalsvariablen folgende Gestalt (siehe Listing 2.11):

```
objects:      (300)t_vpc where ?spe = t_pc6r ;
                             ?cas = t_pc8r ;
                             ?cpu = t_pc3r.
restrictions: table tab_vpc ( t_pc6r = ?spe,
                             t_pc8r = ?cas,
                             t_pc3r = ?cpu).
```

Listing 2.11 Beispiel-Constraint – Tabellenaufruf unter Nutzung von Merkmalsvariablen

[+]

Spezifische Variablenamen

Bei Variablen wird gerne, wie hier gezeigt, mit Fragezeichen gearbeitet. Damit werden sie leichter erkennbar. Für Merkmale aus Tabellen, den Spalten, können Sie nicht mit Variablen arbeiten.

Werden im Objects-Teil mehrere Objekte aufgezählt, geschieht dies durch Komma als Trennzeichen.

Der zweite Pflichtteil in Constraints ist der Restriction-Teil. Hier können Konsistenzprüfungen durchgeführt werden. Das Constraint meldet eine Inkonsistenz, wenn der Restriction-Teil nicht erfüllt ist. Constraints können auch zur Wertehierleitung genutzt werden. Dazu müssen Gleichungen nach der herzuleitenden Größe aufgelöst sein, oder es wird ein Inferences-Teil benötigt. Dazu folgen weitere Ausführungen im Zusammenhang mit dem Inferences-Teil. Der Restriction-Teil kann zur Werteinschränkung genutzt werden (siehe Abbildung 2.33). Es kann mit Variantentabellen und Variantenfunktionen gearbeitet werden.

Im Restriction-Teil kann zudem mit Bedingungen gearbeitet werden. Dazu existiert das Syntaxelement `if`. Weitere Syntaxelemente wie `then` oder `else` gibt es nicht. Zunächst erscheint die Anweisung, dann nachgestellt die `if`-Bedingung wie im Beispiel von Listing 2.12:

```
restrictions:
(300)t_vpc.m1 = 'a'      if (300)t_vpc.m2 = 'x1',
      false            if (300)t_vpc.m2 = (300)t_vpc.m3.
```

Listing 2.12 Restrictions mit `if`-Bedingungen

In diesem Restriction-Teil sind zwei Anweisungen enthalten, die mit Komma aufgezählt wurden. Die erste Anweisung beinhaltet die Wertsetzung des Wertes `a` für das Merkmal `m1`. Die zweite Anweisung führt ausschließlich zu einer Inkonsistenzmeldung, falls die `if`-Bedingung hinter dem `false` erfüllt ist. Die Abarbeitung des Syntaxelements `false` erzeugt generell Inkonsistenzmeldungen und kann nur in Constraints genutzt werden.

In Constraints kann mit einem Condition-Teil gearbeitet werden. Der Condition-Teil muss stets zwischen Objects- und Restriction-Teil aufgenommen werden. Er enthält genau einen logischen Ausdruck. Das Constraint wird dann überhaupt nur abgearbeitet, wenn die Bedingung des Condition-Teils erfüllt ist, was aus Gründen der Performance sehr günstig ist. Im Condition-Teil kann ebenfalls mit Variablen und Variantentabellen gearbeitet werden, wie folgende Beispiele zeigen:

```

objects:  (300)t_vpc where  ?os = t_pc01;
                                ?hd = t_pc04,
                                (300)t_vpr where  ?tr = t_pr02.
condition : specified ?hd.
restrictions : ?tr = ?os

```

Listing 2.13 Beispiel-Constraint mit Condition-Teil

```

objects:  (300)t_vpc where  ?spe = t_pc6r;
                                ?cas = t_pc8r;
                                ?cpu = t_pc3r.
condition:  table tab_vpc (  t_pc6r = ?spe,
                                t_pc8r = ?cas,
                                t_pc3r = ?cpu).
restrictions: false.

```

Listing 2.14 Beispiel-Constraint mit Tabellenaufwurf im Condition-Teil

Im ersten Beispiel (siehe Listing 2.13) wird nur unter der Bedingung, dass das Merkmal `t_pc04` bewertet ist, der Restriction-Teil ausgewertet. In dem Fall wird geprüft, ob die Merkmale `t_pc01` und `t_pr02` gleich bewertet sind. Ist `t_pc01` nicht bewertet, wird die Bewertung von `t_pr02` auf dieses Merkmal kopiert.

Im zweiten Beispiel (siehe Listing 2.14) wird eine Inkonsistenzmeldung ausgegeben, falls die Konfiguration bezüglich der Variantenklasse `(300)t_vpc` in der Bewertungsoberfläche mit einer Zeile der Variantentabelle `tab_vpc` übereinstimmt. Dies wird genutzt, falls in Variantentabellen inkonsistente Bewertungskombinationen gesammelt werden. Solche Constraints lassen sich auch mit dem Tabellenconstraint-Assistenten erstellen. Dabei ist als Wirkungsweise der Eintrag `PRÜFUNG INKONSISTENTER KOMBINATIONEN` auszuwählen. Vergleichen Sie dazu auch Abbildung 2.32 in Abschnitt 2.6.2, »Ein erstes Beispiel«.

Ein vierter Bestandteil des Constraints, der ebenfalls wie der Condition-Teil optional ist, ist der Inferences-Teil. Dieser Teil ist stets der letzte Teil des Constraints und erweitert die Auswertung des Restrictions-Teils. Diese »erweiterte Auswertung« kann sich auf Gleichungen, Variantentabellen, Variantenfunktionen und einschränkbare Merkmale beziehen. Als Syntax enthält der Inferences-Teil ausschließlich eine Aufzählung von Merkmalen.

So wird z.B. eine Gleichung $V = L * B * H$ in einem Restriction-Teil ohne folgenden Inferences-Teil nur in der Form zur Berechnung ausgewertet, dass `V` als Produkt von `L`, `B` und `H` berechnet wird. Sollte das Constraint jedoch die

folgende Gestalt haben, wird aus beliebigen drei Größen die vierte berechnet (siehe Listing 2.15):

```
objects:   (300)t_vpc where
           v = t_pc91 ;   l = t_pc92;
           b = t_pc93 ;   h = t_pc94.
restrictions: v = l * b * h.
inferences:   v, l, b, h.
```

Listing 2.15 Beispiel-Constraint mit Gleichung und Inferences-Teil

Weitere Beispiele sind Variantentabellen und -funktionen, für die mehr als eine Auswertungsalternative definiert sind (siehe Abbildung 2.34 in Abschnitt 2.6.3, »Variantentabellen im Detail«). Wird hier ohne Inferences-Teil gearbeitet, kann nur der erste Schlüssel, also die erste Auswertungsalternative, ausgewertet werden. Sollte das Constraint jedoch die folgende Gestalt haben und sind zwei weitere Auswertungsalternativen mit dem 1. und 3. Merkmal sowie dem 2. und 3. Merkmal in Abbildung 2.34 vorhanden, kann beliebig aus zwei bewerteten Merkmalen das verbliebene dritte Merkmal aus der Tabelle hergeleitet werden (siehe Listing 2.16):

```
objects:   (300)t_vpc where   ?spe = t_pc6r ;
                               ?cas = t_pc8r ;
                               ?cpu = t_pc3r.
restrictions: table tab_vpc ( t_pc6r = ?spe,
                              t_pc8r = ?cas,
                              t_pc3r = ?cpu).
inferences: ?spe, ?cas, ?cpu.
```

Listing 2.16 Beispiel-Constraint mit Variantentabelle und Inferences-Teil

Für die Anwendung im Zusammenhang mit einschränkbareren Merkmalen sei auf das Constraint aus Abbildung 2.33 in Abschnitt 2.6.2, »Ein erstes Beispiel«, verwiesen.

Nachdem hier Constraints recht ausführlich betrachtet wurden, soll noch auf die anderen Arten von Beziehungswissen eingegangen werden. Es gilt, wie bereits erwähnt, dass alle Arten von Beziehungswissen für die Gestaltung der Bewertungsoberfläche im Vertrieb genutzt werden können.

2.6.5 Vorbedingungen

Vorbedingungen können genutzt werden, um einzelne Merkmalswerte oder ganze Merkmale für die Bewertungsoberfläche zu verbieten. Wird ohne Vorbedingungen gearbeitet, kann jedes Merkmal der Bewertungsoberfläche in

beliebiger Reihenfolge bewertet werden. Dabei kann der Wert beliebig aus der Liste der zulässigen Werte gewählt werden – unabhängig von der Bewertung anderer Merkmale. Hierfür müssen zwei Fragen beantwortet werden:

1. Was soll verboten werden? Sprich: Welches Merkmal oder welcher Merkmalswert soll dynamisch verboten werden?
2. Wann soll es erlaubt sein? Sprich: Wann soll das entsprechende Merkmal oder der Merkmalswert im Rahmen der Bewertung erlaubt sein?

Die erste Frage wird durch den Ablageort einer entsprechenden Vorbedingung beantwortet. Die zweite Frage beantwortet die Syntax. Soll z.B. für eine Pkw-Konfiguration der Motor »XYZ« nur für die Sportversion angeboten werden, muss dem Merkmalswert »XYZ« des Merkmals der Motor-Auswahl (»Was verbieten?«) eine Vorbedingung zugeordnet werden. Die Syntax enthält die Voraussetzung, dass die Sportversion ausgewählt wurde, unter der nur der Motor »XYZ« erlaubt ist (»Wann erlauben?«).

Welche Wirkung hat es, wenn die Vorbedingung `$self.version = 'sport'` dem Merkmalswert »XYZ« zugeordnet wird?

► **Merkmal »Version« mit Wert »Sport« bewertet**

Wird das Merkmal »Version« mit dem Wert »Sport« bewertet, stehen in der Liste der zulässigen Werte im Merkmal des Motors alle Werte zur Verfügung, als wäre keine Vorbedingung vorhanden.

► **Merkmal »Version« mit einem Wert außer »Sport« bewertet**

Wird dieses Merkmal »Version« mit einem anderen Wert als »Sport« bewertet, fehlt in der Liste der zulässigen Werte im Merkmal des Motors der Wert »XYZ«.

► **Merkmal »Version« unbewertet**

Zu beachten ist, dass, falls das Merkmal »Version« unbewertet ist, die Vorbedingung als »erfüllt« interpretiert wird. Alle Motor-Werte würden in der Liste der zulässigen Werte des Motor-Merkmals angeboten. Im Standard werden zunächst alle zulässigen Werte angeboten. Im Verlauf der Bewertung werden die nicht mehr zulässigen Werte ausgeblendet. Wird diese Standardlogik nicht gewünscht, muss dies in der Form `$self.version = 'sport' and $self.version specified` ergänzt werden. Jetzt würden zunächst nur die Werte angeboten, die generell auswählbar sind. Die Liste der zulässigen Werte würde schrittweise erweitert.

► **Reihenfolge der Bewertung**

Die obige Vorbedingung würde nur in einer Richtung elegant wirken – nur in dem Fall, dass erst die Version und dann der Motor bewertet werden.

Fängt man die Bewertung am Motor an, kann zunächst ein beliebiger Motor und anschließend eine beliebige Version ausgewählt werden. Erst jetzt würde die Vorbedingung ausgewertet werden und nachträglich eventuell den Motor »XYZ« verbieten. Dies führt zu einer Inkonsistenzmeldung, falls der Motor »XYZ« und anschließend eine andere als die Sportversion ausgewählt wurden. Wie verhindert man dies? Man kann ebenfalls der Sportversion eine Vorbedingung zuordnen. Eine andere Möglichkeit besteht darin, eine Abarbeitungsreihenfolge zu erzwingen. Das Merkmal »Motor« darf erst bewertet werden, wenn die Version bekannt ist. Dies ist eine Anwendung von Vorbedingungen an Merkmalen. Dem Merkmal des Motors (»Was verbieten?«) wird eine Vorbedingung mit der Syntax `$self.version specified` zugeordnet (»Wann erlauben?«).

► Mehrere Vorbedingungen

Man kann einem Merkmalswert oder Merkmal auch mehrere Vorbedingungen zuordnen. In diesem Fall wird die Bewertung nur erlaubt, falls alle Vorbedingungen erfüllt sind. Es kann als Und-Verbindung interpretiert werden. Eine Oder-Verbindung zwischen Vorbedingungen kann nur dadurch erreicht werden, dass die Bedingungen in eine Vorbedingung aufgenommen werden. Eine Vorbedingung kann eine beliebig komplexe Bedingung unter Nutzung von beliebigen Klammern, Negationen und Verkettung mit `and` und `or` sein.

Über Vorbedingungen ausgeschlossene Werte oder Merkmale werden standardmäßig nicht in der Bewertungsoberfläche angezeigt. Über die Einstellungen z.B. im Konfigurationsprofil kann aber gesteuert werden (siehe Abbildung 2.38), dass »verbotene« Merkmalswerte oder auch Merkmale angezeigt, jedoch nicht für die Bewertung genutzt werden können.

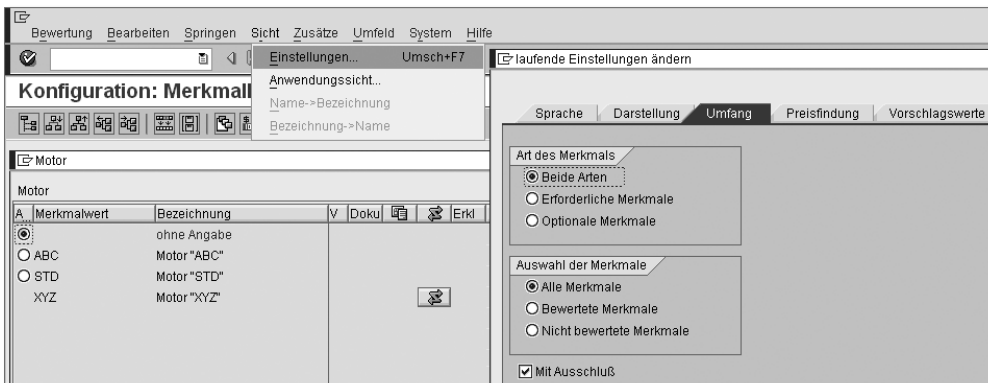


Abbildung 2.38 Anzeigen ausgeschlossener Merkmalswerte (Vorbedingungen)

In Abbildung 2.38 sind während der laufenden Konfiguration über das Menü die Einstellungen aufgerufen worden. Hier ist das Flag MIT AUSSCHLUSS gesetzt worden. Dies hatte zur Folge, dass der »verbotene« Wert »XYZ« angezeigt wird, aber nicht auswählbar ist. Analog würden auch »verbotene« Merkmale angezeigt werden, ohne dass sie bewertbar wären.

In Vorbedingungen kann, wie in allen Arten von Beziehungswissen, auch mit Variantentabellen gearbeitet werden. Die Syntax einer Vorbedingung, die die Tabelle aus Abbildung 2.31 und Abbildung 2.34 nutzt, könnte folgendermaßen aussehen (siehe Listing 2.17):

```
table TAB_VPC (
    T_PC6R = $self.T_PC6R,
    T_PC8R = $self.T_PC8R,
    T_PC3R = $self.T_PC3R)
```

Listing 2.17 Beispiel – Variantentabelle in Vorbedingung

Sie hat damit genau dieselbe Syntax wie eine Prozedur. Statt `$self` kann jedoch hier an allen Stellen auch mit `$parent` oder `$root` gearbeitet werden, soweit es inhaltlich richtig ist. Man könnte eine solche Vorbedingung mit genau dieser Syntax einem der drei angesprochenen Merkmale zuordnen, was jedoch keine elegante Lösung darstellen würde.

Mit zunehmender Eleganz der Lösung gibt es diese drei Möglichkeiten, Werte zu verbieten:

- ▶ Vorbedingung am Merkmal (wie beschrieben)
- ▶ Vorbedingung am Merkmalswert (wie wir im Anschluss beschreiben werden)
- ▶ ein Constraint mit dieser Variantentabelle und einschränkbar Merkmalen (als eleganteste Lösung; siehe Abbildung 2.33)

Sollte man die angegebene Variante mit der Vorbedingung an dem Merkmal wählen, würde während der Bewertung der drei Merkmale keinerlei Wirkung durch die Vorbedingung sichtbar. Erst nach der Bewertung aller drei in der Variantentabelle angesprochenen Merkmale wird die Vorbedingung aktiv und gibt eine Inkonsistenz aus, falls diese Bewertungskombination nicht in der Tabelle enthalten ist.

Eleganter als diese Variante mit einer Vorbedingung am Merkmal – wenn auch nicht an die Variante mit dem Constraint herankommend – sind Vorbedingungen an den Merkmalswerten. Geht man z.B. davon aus, dass die Merkmale in einer festen Reihenfolge bewertet werden, so dass vielleicht

das Merkmal T_PC3R als Letztes bewertet wird, kann man die Merkmalswerte dieses Merkmals mit Vorbedingungen versehen. Die Vorbedingung für den Wert '03' würde folgende Syntax besitzen (siehe Listing 2.18):

```
table TAB_VPC (
    T_PC6R = $self.T_PC6R,
    T_PC8R = $self.T_PC8R,
    T_PC3R = '03')
```

Listing 2.18 Beispiel Variantentabelle in Vorbedingung für Merkmalswert »'03'«

Analoge Vorbedingungen gelten für die anderen Werte dieses Merkmals. Dies hat gegenüber der ersten Variante, der Vorbedingung am Merkmal, den Vorteil, dass für das dritte Merkmal nur noch die Werte angeboten werden, die zu einer konsistenten Bewertung führen. Diese Vorbedingungen an Merkmal oder Merkmalswert benötigen keine einschränkbaren Merkmale.

2.6.6 Auswahlbedingungen

Auswahlbedingungen können aus Kann-Merkmalen dynamisch Muss-Merkmale machen.

[+]

Kann-Merkmale

Diese Technik setzt voraus, dass im Merkmal das Flag EINGABE ERFORDERLICH nicht gesetzt ist, es sich also tatsächlich um ein sogenanntes Kann-Merkmal handelt.

Wird einem solchen Merkmal eine Auswahlbedingung zugeordnet, wird das Merkmal dynamisch zum Muss-Merkmal, falls die Bedingung erfüllt ist. Verdeutlichen wir uns dies an einem Beispiel.

[zB]

Pkw-Sportversion erfordert zwingend ein Navigationssystem

Für die Sportversion soll die Auswahl eines Navigationssystems bei der Pkw-Konfiguration zwingend werden. Dabei müssen sowohl Regelwerk als auch Syntax analog zu den Vorbedingungen sein: Die Auswahlbedingung muss dem Merkmal der Navigationssysteme zugeordnet werden (»Was müssen?«); der Inhalt der Syntax ist die Bedingung der Sportversion (»Wann müssen?«). Die Syntax ist analog zu obiger Vorbedingung: `$self.version = 'sport'`. Die Auswahlbedingung mit dieser Syntax wird dem Merkmal des Navigationssystems zugeordnet. Somit wird für die Sportversion die Auswahl eines Navigationssystems zwingend.

Es gibt zwei Unterschiede zu den Regeln der Vorbedingungen, die zu beachten sind:

Index

? → Vorschlagswertsetzung
|| → Zeichenketten-Verknüpfung

A

A2A → Application to Application
ABAP und Beziehungswissen 138
ABAP-Funktionsbaustein 60, 581
ABAP-Programmiersprache 58, 555
abstrakter Datentyp (ADT) 423
ACWG → American Configuration Workgroup
Adaptable Custom Solution (ACS) 236, 242
Advanced Mode 64
Advanced Planning Optimization (APO) 317, 467
Aggregation 423
Aggregationsmerkmal 425
Ähnlichkeitsfindung 315
Aktion 65, 132
ALE → Application Link Enabling
American Configuration Workgroup (ACWG) 620
Americas' SAP Users' Group (ASUG) 617
Analysetool 149
änderbare Materialvarianten 466
Änderung
 Antrag 393
 Art 395
 Nummer 391, 392, 405
 Stamm 393
 Stammsatz 392
Änderungsdienst 440, 448, 455, 581
Anwendungsgruppe 471
Anwendungslog 455
Anwendungssicht 105, 336
AP Configuration Engine 65, 351, 352, 384
AP Pricing Engine 352
AP-Applikationsplattform 66
APO → Advanced Planning Optimization
Application Link Enabling (ALE)
 ALE-Partnervereinbarungen 429, 433, 434

ALE-Verteilung 427
 ALE-Verteilungsmodell 429, 434
Application Programming Interface (API) 647
Application to Application (A2A) 639
Arbeitsplan 41, 77, 392
 simulativ aufgelöster Arbeitsplan als Kopiervorlage 216
arithmetische Operatoren 142
ASAP Implementation Roadmap 559
Assemble to Order (ATO) 43
ASUG → Americas' SAP Users' Group
asynchron 608
ATO → Assemble to Order
Auflösungsprofil 435, 446
Auftrag
 Arbeitsplan 101, 268
 Art 268
 Stückliste 77, 98, 232, 236, 268, 587
 Zusammenfassung 478
Auswahlbedingung 129, 176, 576, 581
 Auswerten 138
automatisierte Produktkonfiguration 37

B

B2B → Business to Business
B2C → Business to Consumer
BAaI 241
Baseline 444, 446
Baseline-Auflösung 448
Batchjob 447
Baustein 422
Bausteintyp 420
Bautyp 294
Bedarf
 Art 271, 342
 Klasse 271, 342, 404
 Planung 268
Bedarfsart 288
Bedarfsplanung 277
Bedingung 362
Beraterrolle
 IPC-Experte 556
 Modellierer 555

Preisfindungsexperte 556
Projektleiter 557
Solution Architect 554
Stammdatenexperte 557
VC-Experte 554
 Berechtigungsgruppe 81
 Berechtigungsobjekt
 C_LOVC_DEP 324
 C_TCLS_BER 336
 Beschaffungsart 95
 Bestellanforderung 284
 Bestellung 275, 284, 285
 Beziehungsart 128
 Aktion 65, 386
 Auswahlbedingung 59, 60, 129
 Constraint 59, 386, 417
 Monitoring Rule 423
 Prozedur 59, 129, 386
 Reevaluating Rule 423
 Vorbedingung 58, 128
 Beziehungseditor 191
 Beziehungsnetz 167
 Beziehungswissen 58, 59, 77, 132, 133, 392
 Assistent in der PMEVC 152, 154
 Basisdaten 190
 deklaratives 133
 Freigabe 135
 globales 134
 lokales 134
 prozedurales 133
 semi-deklaratives 133
 Status 134
 Bilanzierung 423
 Branchenlösung 465
 Business Object 628
 Business to Business (B2B) 51
 Business to Consumer (B2C) 51
 Business Transaction Event (BTE) 433

C

CCE → Core Constraint Engine
 Chargeneinzelbewertung 477
 Chargenklassifizierung 467
 Chargenselektionskriterium 467
 Coaching-Projekt 572
 Coil 465
 condition 168, 170

Configuration Management 429
 Configuration Workgroup (CWG) 62, 617
 CWG Sandbox 625
 CWG-Konferenz 623
 CWG-Portal 618, 624
 Exekutivkomitee 623
 Mitgliedschaften 622
 Präsident 623
 Satzung 618
 Verein 622
 Vorstand 623
 Configure to Order (CTO) 43, 354
 Constraint 59, 131, 154, 576
 Constraint-Netz 49, 59, 167, 423
 Constraints und Klassenknoten 195
 Core Constraint Engine (CCE) 638
 CTO → Configure to Order
 Customer Relationship Management 51, 352
 Customizing des Materialstamms 336

D

Datenbanktabelle 161, 165, 576
 AUSP 335
 Datenbanktabelle und Variantentabelle 145
 ESLL 287
 PLPO 201, 290
 STPO 199
 TB31 329
 VBAK 180, 266
 Datenbasis
 dynamische 389
 Datentyp 81
 numerisches Format NUM 82
 Zeichenformat CHAR 82
 Datenvolumen 70
 Datumselement 398
 deklarative Herangehensweise 47
 deklarative Modellierung 362
 deklaratives Beziehungswissen 133
 Deltaliste 65, 356
 Dependency Maintenance Table (DMT) 258
 Deutschsprachige SAP-Anwendergruppe (DSAG) 617
 Dienstleistung 285

Dienstleistungsmaterial 295
 DIMP → Discrete Industries and Mill Products
 Discrete Industries and Mill Products (DIMP) 463
 Disposition
 Gruppe 271
 Merkmal 95
 Dispositionsgruppe 95
 Dokument 86, 391
 DSAG → Deutschsprachige SAP-Anwendergruppe
 dynamische Instanziierung 41
 dynamische Stückliste 98
 dynamische Vorgangsfolge 101

E

ECM → Engineering Change Management
 Eigenfertigung 272
 einfache Stückliste 232
 eingeschränkbares Merkmal 154, 158
 Einstiegsknoten 243
 einstufige Stücklistenauflösung 595
 Einteilungstyp 270, 344
 Engineer to Order (ETO) 34, 44, 235, 581, 587
 Engineering Change Management (ECM) 77, 390
 ECM-Historienpflicht 399
 Entkopplungspunkt 466
 Entscheidungstabelle 162
 Equipment 277, 291
 Ereignistyp-Kopplung 442
 Erfahrungsbericht 531, 557
 Erzeugniskalkulation mit Mengengerüst 209
 ETO → Engineer to Order
 Extended Configuration Management (XCM) 598

F

False 170
 Fertigungsauftrag 272, 277, 403

Fertigungsauftrags-Änderungsdienst 77, 273, 347, 390, 403
 Fertigungshilfsmittel-Zuordnung 102
 Fertigungsszenario 42
 Assemble to Order 43
 Configure to Order 43
 Engineer to Order 44
 Make to Order 43, 631
 Make to Specification 633
 Make to Stock 42
 Filter 272
 Finish to Order 466
 Flag »Konfigurierbares Material« 94
 Folgen 102
 Folien 465
 Formel 362
 FOX → Framework for Object Explosion
 Framework for Object Explosion (FOX) 435, 444, 446
 Freigabeschlüssel 394, 405
 Fremdbeschaffung 272
 function 146, 185
 Funktionsbaustein 185, 577
 Funktionsbaustein und Beziehungswissen 145
 Funktionsformel 363

G

Gesamtänderungsprofil 347
 Gesamtprofil 404, 407
 Geschäftsobjekt 628
 globales Beziehungswissen 134, 188
 Gruppen 327
 GSS PSM to BOM 261
 Guided Structure Synchronization (GSS) 258, 261
 Gültigkeitsparameter 253

H

harte Wertsetzung 130, 177
 »Hello World«-Beispiel 52
 Hierarchie erlaubt 334
 High-Level-Konfiguration 107, 354

I

IDoc → Intermediate Document
 Individual Customization 38
 Industry Solution 463
 inferences 168, 171
 InfoSet 298
 Infostruktur S138 296
 in-Listabfragen 143
 Instandhaltung 274, 291
 Instandhaltungsanleitung 51, 274
 Instanz 645
 Instanziierung
 dynamische 41, 412
 manuelle 412
 Materialvariante 414
 integriertes Produkt- und Prozess-Engineering (iPPE) 242, 246
 Anforderungsstruktur 244, 246
 Einstieg 253
 Einstiegsknoten 243
 erweiterter Beziehungswissen-Editor 251
 Filter 254
 Knoten 243
 Konfigurationssimulation 253
 Konzept 245, 251
 Konzeptgruppe (iPPE) 251
 Sichtknoten 244
 Strukturknoten 243
 Stücklistenkonverter 245, 255
 Varianten 244
 Varianzschema 244, 251
 integriertes Produkt-Engineering 246
 Interaction Center 352
 Intermediate Document (IDoc) 426, 428, 451
 Internet Pricing and Configurator (IPC)
 51, 62, 384, 598, 627
 IPC_CONFIGURATION_UI 328
 IPC-Dataloader 225
 IPC-Datenbank 225
 IPC-Modellierung
 Delta 229
 IPC-Produktkonfigurator 66
 invisible 181
 IPC → Internet Pricing and Configurator
 iPPE → integriertes Produkt- und Prozess-Engineering

Ist-Bewertung 467
 Ist-Konfiguration 466
 Ist-Merkmale 467

J

Java-Programmiersprache 62, 556

K

Kabel 465
 Kalkulationsschema 204
 KBIF → Knowledge Base Interchange Format
 Klassenhierarchie 58
 Klassenknoten 60, 86, 192, 228, 334, 386, 581
 Klassensystem 58, 79
 Klassenverwaltung 86
 klassifizierte Materialien 197
 Klassifizierung 88
 KMAT 53
 Know-how-Träger 553
 Knowledge Base 64, 355
 Knowledge Base Interchange Format (KBIF) 620
 Komponentenbedingung 363
 Komponentenformel 363, 368
 Komponentenliste in Plan- und Fertigungsauftrag 272
 Komponentenstruktur 39
 Dynamisierung der Stücklistenstruktur 410
 Komponentenzerlegung 39
 Kompositionsproblem 411, 419
 Top-down-Ansatz 419
 vernetzte Konfigurationsstruktur 410, 415
 Zerlegungsproblem 419
 Konditionsart 203
 VA00 203
 VA01 203
 Konditionstechnik 63
 Konfektionsschritt 476
 Konfiguration 38
 Aufgabe 34
 Baustein 39

Benutzeroberfläche 328
Browser 109
Definition 444, 445
Einstellungen 109, 219
Ergebnis 37, 388
High-Level-Konfiguration 61, 382, 392
interaktive Konfiguration 37, 48
Kennzeichen 58
Konfigurationsergebnis 37
Low-Level-Konfiguration 61, 382, 392
Management 435
Mappe 444, 445, 446
Schritt 48
Stützpunkt 307
Szenario 105, 111
 Konfigurationsmodell 37
 Vervollständigen 55
 Konfigurationsprofil 52, 59, 76, 227, 392, 413
 der konfigurierbaren Komponente als Filter 272
 PMEVC 154
 Konfigurationsprozess
 Auftragsstückliste 412, 413
 Kundenauftrag 413
 Konfigurationsregeln 39, 58
 deklarative Herangehensweise 47, 49, 417
 einfache Regel 47
 prozedurale Herangehensweise 47, 48
 konfigurationsspezifische Bausteine 58
 Konfigurationsstruktur 109
 Bottom-up-Ansatz 419
 Konfiguration-UI 328
 Konfigurator 39
 Konfiguratoranbindung 638
 konfigurierbar
 Baugruppe 108
 Einkaufsmaterial 581
 Instandhaltungsanleitung 292
 Material 53
 Muster-Leistungsverzeichnis 285
 Standardnetzplan 288, 290
 Konfigurierer 584
 kontextsensitive Eingabehilfe in der
 PMEVC 152
 Konvertierung 256
 Kopfmateriale 581, 597
 Kopiersteuerung 268

Kopplung
 Varianten- und Datenbanktabelle 161, 165
 Korrekturpaket 444, 454
 Kundenauftrag 275, 403
 Änderungsdienst 476
 Arbeitsplan 238, 242
 Bestandssegment 269, 270
 Lagerfertigung 466
 Stückliste 232, 238, 581
 Kundeneinzelfertigung 43, 477
 Kundenservice 274, 291

L

Lagerfertigung 42, 477
 kundenauftragsbetrieben 466
 Lagertype 634
 längenorientiert 465
 Laufzeitversion 64, 225, 355, 387, 420
 Lebenszyklusphase 435
 Lebenszyklusprofil 435
 logische Operatoren 142
 Logistikinformationssystem 296
 lokales Beziehungswissen 134, 188, 189
 LO-VC-kompatibler Modus 64
 LO-VC-Variantenkonfigurator 51, 384
 Low-Level-Konfiguration 354

M

Make to Order 42, 43, 593
 Make to Specification 633
 Make to Stock 42, 43, 593
 Mass Customization 38
 Mass Production 38
 Massendaten ändern 550
 Material 52, 391
 Materialstamm 75, 274
 Sichtenkonzept 93
 Materialstamm-Customizing 337
 Materialstückliste 77
 änderbare 466
 Materialvariante 77, 212, 634
 Materialvariantenfindung 212, 218
 Maximalstückliste 40
 mdata 143

Mehrfachklassifizierung 333
 Mehrfachstückliste 97, 233
 Merkmal 39, 76, 385, 392
 Abhängigkeit 39
 Anlegen 54
 Anzeige 471
 Gruppe 81, 186
 Merkmalswert 39, 385
 Merkmalswert verbieten 172
 Name 80
 Schnellerfassung 471
 Status 81
 Vorplanung 304, 343
 Vorschlagswert 637
 merkmalsabhängige Planung 477
 merkmalsbasierte Planung 467
 Merkmalsverwaltung 79
 Middleware (SAP CRM) 225
 Mill Products 463
 Mill-Industrie 465
 Mittelstandslösung 627
 MMCOM 208
 MMCOM-VKOND 208
 Modellbegriff 60
 Montageabwicklung 288, 409
 MTO → Make to Order
 MTS → Make to Stock
 Muss-Merkmal 83
 dynamisches 176
 Muster-Leistungsverzeichnis 52, 273,
 274, 285
 Mustermodell 557

N

Nachrichten-Interface 639
 Nachrichtentyp (IDoc) 426
 NetWeaver Business Client (NWBC) 258
 Netzplan 274, 581
 not specifiable 363
 Nummernkreis 432

O

Oberflächendesign 108, 186
 Oberflächendesign als Filter 272

objects 168
 Objektmerkmal 61, 84, 199, 228, 581
 Zugriff 84
 Objektsuche in Klassen 89
 Objekttyp 438
 Objektverwaltungssatz 397
 OCM → Order Change Management
 On-Demand-Lösung 628
 operatives Umfeld 310
 Operatoren
 arithmetische 142
 logische 142
 Zeichenketten-Operatoren 142
 Order Change Management (OCM) 77,
 273, 347, 390, 403, 476
 Order Engineering Workbench (OEW)
 235
 Originalpaket 444

P

Pakettyp 440
 Papier 465
 Pattern Matching System 389
 PDR → Product Data Replication
 Pflegeberechtigungen 324
 pfunction 185
 Planauftrag 277, 307
 Plankonfiguration 466
 Plankopf 101
 Planprimärbedarf 307
 Plantyp 100
 Planungsstrategie 346, 405
 25 346
 70 347
 PLM WebUI 258
 PLM-Extension 429, 430
 PMEVC 150, 226
 Drag & Drop 152
 Portallösung 592
 Positionstyp 99, 270, 339, 469
 AGC 270
 TAC 270, 340
 TAM 340
 Positionstypenfindung 339
 Positionstypengruppe 270
 Postponement 466

PP-Stammdaten nachlesen 273, 404
 Preisfindung 62, 63, 77, 276, 284
 Problemlösungsprozesse 550
 Product Data Replication (PDR) 425
 Delta-Filterung 440, 450
 Paket-Einbuchung 452
 Paketversand 451
 Replikation eines VC-Modells 443
 Product Modeling Environment (PME)
 63
 Java PME 63, 420
 PMEVC 56
 Product Structure Management (PSM)
 261
 Produkt 35
 Ausprägungen 45
 Designer 246
 konfigurierbares 37, 611, 636
 Merkmal 636
 Modell 635
 Modellierungsumgebung 150
 Spezifikation 36, 632
 spezifizierbares 632
 spezifiziertes 632
 Struktur 245
 Variante 36, 634
 Variantenstruktur 243
 Produkt- und Prozessflexibilität 630
 Produktionskonfiguration 354
 Produktionsregelsystem 49
 Produktionstoleranzen 467
 Produktkonfiguration 34, 36, 38, 42
 automatisierte 37
 Vorgehen 36
 Produktmerkmal 636
 Produktmerkmalsformat
 boolescher Wert 636
 Code 636
 Dezimalzahl 636
 Freitext 637
 ganze Zahl 636
 Mengenangabe 636
 Produktmodell
 Erstellen 73
 Produktspezifikation 36
 Produktvariante 42
 Project Builder 289
 Projektstrukturplan 288

Projektstückliste 232
 Projektsystem 273, 275, 288
 Prozedur 59, 129, 177, 199, 576, 581
 Auswerten 138
 Prozedur und Klassenknoten 194
 prozedurale Herangehensweise 47, 48
 prozedurale Modellierungstechnik 362
 prozedurales Beziehungswissen 134
 Prozessautomatisierung 639
 Prüffart 279
 Prüflos 278
 Prüfmerkmal 278

Q

QM-Sperrbestand 283, 287
 Qualitätsmanagement 274, 278

R

Reconciliation Workbench (RWB) 264
 Registerkarte
 Auftragsstückliste 118
 Basisdaten 81, 105
 Bezeichnungen 83
 Einschränkungen 84
 Konfigurationseinstieg 106
 Konfigurationsparameter 106, 122
 Kundenauftrag 121
 Merkmalschnellerfassung 471
 Oberfläche 108
 Werte 83
 Zusatzdaten 84, 86
 Replication Workbench 443
 Replikationstabelle 441, 455, 459
 Report
 RUPSHIELEV 439
 RUPSPPOST 437, 454
 RUPSEND 437, 451
 restriction 168, 170
 Rohre 465
 Rückmeldung 273, 282
 Rückwärtsverkettung 50

S

- SaaS → Software-as-a-Service
- Sales Configuration Engine (SCE) 63, 390
 - Advanced Mode* 64, 418
 - LO-VC-kompatibler Modus* 64
- Sales Pricing Engine (SPE) 63
- SAP APO 467
- SAP Apparel and Footwear Solution 554
- SAP Business ByDesign 34, 627
- SAP CRM 51, 64
 - CRM 5.0* 66
- SAP Custom Development 236, 242, 568
- SAP Engineering & Construction 465
- SAP Enterprise Solutions 627
- SAP ERP 34, 384
 - ECC 6.0* 66
 - ERP Central Component (R/3)* 51
- SAP for Aerospace and Defense 465
- SAP for Automotive 465
- SAP for High Tech 465
- SAP for Industrial Machinery & Components 465
- SAP for Mill Products 465
- SAP NetWeaver BW 296, 298
- SAP NetWeaver-Technologieplattform 65
- SAP PLM 555
- SAP Vehicle Management System 554
- SAP-Hinweis
 - 1081650 388
 - 1121318 390
 - 173756 297
 - 174758 297
 - 68033 309
 - 844816 352
 - 844817 328
 - 854170 328
 - 901689 390
 - 917987 388
 - 997111 390
- SAP-Philosophie 542
- SCE → Sales Configuration Engine
- SCREEN_DEP 180
- SDCOM-VKOND 203
- semi-deklaratives Beziehungswissen 133
- Serialnummer 277, 283, 291
- Serialnummernprofil 277
- Serviceauftrag 294
- Service-Oriented Architecture (SOA) 628
- Serviceprodukt 352
- Sichtknoten 244
- simulatives Umfeld 310
- SKEY 139
- SOA → Service-Oriented Architecture
- Software-as-a-Service (SaaS) 628
- SPE → Sales Pricing Engine
- specifiable 363
- specified 143, 362
- Sperre 104
- Stammdaten-Änderungsdienst 77, 390
- Stammprüfmerkmal 278
- Standarderzeugnisvorplanung 304
- Standardnetzplan 51, 290
- Standardprodukt 36
- Standard-Projektstrukturplan 289, 290
- Startlogo 106
- Stoffe 465
- Strategie 96, 270, 288
 - Planungsstrategie* 343
- Strategiegruppe 270, 288
- Strukturknoten 243
- Stückliste 40, 231, 386, 391
 - dynamische* 98
 - Erweiterungen* 41
 - Filter* 106
 - konfigurierbare Stückliste* 40
 - manuell Ergänzen* 41
 - Maximalstückliste* 40
- Stücklistenanwendung 106, 234, 436
- Stücklistenauflösung 106, 193
- Stücklistenbeziehung
 - has part* 422
 - part of* 416, 422
- Stücklistenpflege 193
- Stücklistenstruktur 572
- Stücklistensynchronisation 245, 258, 261
- Stücklistenverwendung 233
- Supply Chain Management 317
- Switch Framework 463
- Synchronisationseinheit (GSS) 262
- Syntaxelement
 - \$count_part* 144
 - \$del_default* 144, 178

- \$PARENT. 141
 \$part_of 144
 \$ROOT. 141
 \$SELF. 141
 \$set_default 143, 178
 \$set_pricing_factor 144, 205, 209
 \$subpart_of 144
 \$sum_part 144
 function 185
 inv 182
 invisible 181
 pfunction 185
 Syntaxregel 140
 Systemkonfiguration 64, 409, 410
 Szenario
 Auftragsstückliste 113
 Kundenauftrag (SET) 120
 Plan-/Fertigungsauftrag mit Stücklistenauflösung 124
 Plan-/Fertigungsauftrag ohne Stücklistenauflösung 111
- T**
-
- Tabellen-Constraint 335
 Tabellenconstraint-Assistent
 PMEVC 158
 Tabellenformel 362
 Table 145
 Technikmerkmal 588
 Toleranzen 281
 Trace 148
 Trace-Funktion 388
 Transaction Tax Engine (TTE) 63
 Transaktion
 /OEWB/MAIN – Order Engineering Workbench 236
 BD87 – IDoc-Übersicht 451
 BF01 – BTE-Administration 433
 C223 – Fertigungsversion Massenpflege 551
 CA75 – Massenänderung Fertigungshilfsmittel 550
 CA85 – Arbeitsplatz ersetzen 550
 CA95 – Standardplan in Plänen ersetzen 550
 CAVC_TEST 647
 CC01 – Änderungsstamm anlegen 393
 CC03 – Änderungsstamm anzeigen 400
 CEWB – Engineering Workbench 551
 CFM1 – Integrationsmodell anlegen 318
 CFM2 – Integrationsmodell aktivieren 319
 CL02 – Klassenverwaltung 55
 CL20N – Objekt zu Klassen zuordnen 333
 CL24N – Objekte einer Klasse zuordnen 333
 CLGT – Tabellen für die Suche einrichten 335
 CLMM – Massenänderungen von Bewerbungen 551
 CMOD – User Exits 329
 CN08 – Netzplanparameter aus Kundenauftrag bearbeiten 290
 COCM – Verursachervormerkungen 406
 COCM1 – Beschaffungselemente Vormerkungen 407
 CRWBD – Replication Workbench 443
 CS20 – Massenänderungen in Stücklisten 550
 CS40 – Zuordnung konfiguriertes Material 215, 314
 CS62 582
 CSKB – Auftragsbrowser 236
 CT04 – Merkmalsverwaltung 54
 CT12 – Verwendungsnachweis Merkmale/Merkmalswerte 392, 401
 CTBW – Tabellenpflege für BW und Klassen 299
 CU05 – Verwendungsanzeige von Beziehungen 402
 CU50 – Konfigurationssimulation 403
 CU51 – Auftragsstückliste 236
 CU51E – Auftragsstückliste ergebnisorientiert 571
 CU60E 163
 CU61 – Variantentabelle anlegen 387
 CU62 – Variantentabelle ändern 387
 CUMODEL 146
 EXPO_TEST – Test-Strukturauflösung durch FOX 458
 IP10 – Wartungsplan terminieren 293
 IP41 – Wartungsplan anlegen 293
 MC(B – Standardanalyse Variantenkonfiguration 296
 MCSZ – Copy Management 297

- MD04 – Bedarfs-/Bestandsliste* 277, 284
MD50 – Einzelplanung Kundenauftrag 277
MDP1 – Vorplanungstabelle anlegen 306
MDPH – Vorplanungsprofil 306
MDPV – Ähnlichkeitsfindung 315
MIGO – Warenbewegung 283
MM01 – Material anlegen 52
MM17 – Massenpflege Materialstamm 551
MM50 – Erweiterbare Materialien 551
MS02 – Langfristplanung einzel-/mehrstufig 311
MS31 – Langfristplanungsszenario anlegen 311
MS66 – Simulationssekundärbedarfe übernehmen 310
NWBC – NetWeaver Business Client 258
OISD – Serviceprodukte 295
OMO1 – Fortschreibung aktivieren 297
OVRP – Fortschreibungsgruppe Positionsebene ändern 297
PCFG – Rollenpflege 329
PMEVC – Modellierungsumgebung Variantenkonfiguration 56, 328
PPECS – Stücklistenkonverter 255
QE51N – Ergebniserfassung QM 280
RSA2 – SAPI DataSource Repository 300
SCUO – Customizing Cross-System Viewer 458
SLG1 – Anwendungslog 455, 458
SM37 – Jobauswahl 448, 450
SM50 – Prozessübersicht 450
ST01 – Berechtigungs-Trace 458
ST05 – SAP NetWeaver-Performanceanalyse 390
STAD – Business Transaction Analysis 390
SU03 – Pflege Berechtigungen 420
UPS – UPS-Cockpit (PDR) 452
UPSRCP – UPSRCP-IDocs nachbuchen 459
UPSETUP – Customizing-Vorbereitung für PDR 429, 432
VA01 – Kundenauftrag anlegen 384
WE20 – Partnervereinbarungen 434
 TTE → Transaction Tax Engine
 Typ einer Änderungsnummer 393
 Type 602
 type_of 143
 Typenvorplanung 312
- ## U
-
- Umwidmung 476
 Unternehmensphilosophie 541
 UPS (Uniform Packaging Service) 437
 UPSMAS 428
 UPSRCP 428
 User Exit 555, 649
- ## V
-
- Variabilität 572
 Variable in Constraints 169
 Varianten 36, 38
 Arbeitsplan 60
 Funktion 59, 145, 182
 Klasse 52, 55, 76, 86, 392, 416
 Klasse und PMEVC 154
 Klassenart 333
 Konditionen 576
 Konditionssatz 203
 Konditionsschlüssel 62
 Konfigurator LO-VC 51, 58, 412, 415, 627
 Modell-Browser 146
 Stückliste 60, 97, 233
 Tabelle 59, 144, 154, 387, 392, 576, 581, 638
 Tabelle und Prozedur 179
 Teile 191
 Vielfalt 44
 Variantenkonfiguration 38
 Grundlagen 33
 Hauptaufgaben 61
 Integration 74
 Modellierung 74
 Variantenmodell
 Reihenfolge der Pflege 78
 variantenreiche Produkte 74
 Variantenvielfalt 44
 VCSD_UPDATE 180
 Vererbung 58

Vergleichsfunktion 238
 Verkaufskonfiguration 354
 Verkaufstyp 604
 Versionierung 239, 240
 Versionierung von Auftragsstücklisten
 238
 Verteilungsauftrag 444, 449
 Verteilungspaket 444
 Verteilungssperre 325
 Verteilungsart 443
 Verteilungseinheit 444, 449
 Verteilungspaket 427, 444, 449
 Vertriebsbelegtyp 266
 Vertriebsmerkmal 588
 Verwendungsnachweis Merkmale/Merkmalswerte 90
 Virtual Machine Container (VMC) 65
 VMC → Virtual Machine Container
 Vorbedingung 128, 172, 576, 581
 Vorbedingung und Variantentabelle 175
 Vorgang 102
 Vorgangsfolge
 dynamische 101
 Vorgangsliste im Fertigungsauftrag 273
 Vormerkungen für das OCM 344, 405
 Vorplanung 77, 296
 Profil 297
 Tabelle 297
 Variante 312, 313, 343
 Vorschlagswert
 dynamischer 178
 Vorschlagswertsetzung 143
 Vorwärtsverkettung 49

W

Wartungsplan 292
 WBO → Wissensbasisobjekt
 Web Channel 351
 weiche Wertsetzung 131, 177
 Werkzeuge 79
 Werte einschränken 367
 Wertemenge 37
 Werterücksetzung 48
 Wertigkeit 83
 Wertsetzung
 harte 130, 177
 weiche 131, 177
 WF-BATCH (Workflow-User) 450
 Wissensbasis 64, 355, 387, 420
 Wissensbasisobjekt (WBO) 64, 224
 Workflow-Customizing 442
 Workflow-System 581

X

XCM → Extended Configuration Management
 XCM-Szenario 328

Z

Zeichenketten-Operatoren 142
 Zeichenketten-Verknüpfung 142
 Zielsystem 446