



Advanced Variant Configuration in SAP S/4HANA®

- › Praktische Anleitung für die Produktmodellierung
- › Best Practices für den gesamten Produktlebenszyklus
- › Integration mit SAP Commerce Cloud, SAP CPQ, SAP BTP, Machine Learning u. v. m.

Blumöhr · Kölbl
Neuhaus · Ukalovic

Kapitel 5

Die Intelligenz: Beziehungswissen

Das Beziehungswissen stellt das Herzstück des Variantenmodells dar und steht in diesem Kapitel im Mittelpunkt. Im Folgenden werden sowohl alle Arten von Beziehungswissen als auch alle weiteren Werkzeuge, die im Rahmen von Beziehungswissen genutzt werden können, vorgestellt.

Beziehungswissen spielt im Rahmen der Modellierung eine große Rolle. Das Beziehungswissen stellt ein Regelwerk dar, das sicherstellt, dass von der Person, die die Konfiguration durchführt, zum einen das Richtige »gewünscht« wird und man das Gewünschte auch erhält. Mithilfe von Beziehungswissen werden Abhängigkeiten zwischen Merkmalen und Merkmalswerten beschrieben sowie die Auswahl der Stücklistenkomponenten und Arbeitsvorgänge gesteuert.

Es wird Beziehungswissen benötigt, um die Konfiguration in der Bewertungsoberfläche – meistens im Vertrieb – zu steuern (»das Richtige wünschen«). Über Beziehungswissen sorgen wir dafür, dass das Produkt wunschgemäß gefertigt oder allgemein beschafft wird (»das Gewünschte erhalten«).

Dieses umfangreiche Kapitel soll das Thema Beziehungswissen mit all seinen Details vorstellen: Nach einem Überblick in Abschnitt 5.1 beschäftigt sich Abschnitt 5.2 mit den Variantentabellen. Beziehungswissen für die interaktive Konfiguration lernen Sie in Abschnitt 5.3 kennen. Thema von Abschnitt 5.4 ist das Beziehungswissen für Stückliste und Arbeitsplan.

5.1 Grundlagen

In diesem Abschnitt stellen wir das Beziehungswissen im Überblick vor. Dabei gehen wir auf die High-Level-Konfiguration und auf die Low-Level-Konfiguration ein.

Unter der *High-Level-Konfiguration* wird die Vertriebskonfiguration und die Konfiguration im Dialog, also die Konfiguration »im Hellen« verstanden: das Beziehungswissen für die Bewertungsoberfläche.

Unter der *Low-Level-Konfiguration* wird die Konfiguration ohne Dialog, also die Konfiguration »im Dunkeln« verstanden: das Beziehungswissen für die Stücklisten- und Arbeitsplanauflösung.



Hinweis: High-Level-Konfiguration

Die High-Level-Konfiguration hat die folgenden Merkmale:

- Vertriebskonfiguration
- im Dialog
- »im Hellen«
- für die Bewertungsoberfläche



Hinweis: Low-Level-Konfiguration

Die Low-Level-Konfiguration hat die folgenden Merkmale:

- ohne Dialog
- »im Dunkeln«
- Für die Stücklisten- und Arbeitsplanauflösung

Wir beginnen mit den Grundlagen und allgemeingültigen Regeln des Beziehungswissens.

5.1.1 Arten und Zuordnung von Beziehungswissen

Beziehungswissen wird für die beiden folgenden Aufgaben benötigt:

- **Unterstützt den Konfigurationsprozess der Bewertungsoberfläche**
Beziehungswissen unterstützt den Prozess der Konfiguration in der Bewertungsoberfläche, wie z. B. im Kundenauftrag. Dabei soll sichergestellt werden, dass das Ergebnis der Konfiguration vollständig und konsistent ist. Dies kann beliebig umfangreich gestaltet werden. Das heißt, dass durch entsprechenden Aufwand mit der Modellierung die Bewertung in der Konfiguration beliebig einfach und elegant sein kann. Das Ziel ist, dass nicht erlaubte Bewertungskombinationen nicht angeboten, Bewertungen (wo gewünscht) über Beziehungswissen berechnet und gesetzt oder auch Vorschlagswerte angeboten werden.
- **Unterstützt Stücklisten- und Arbeitsplanauflösungen**
Beziehungswissen erzeugt Stücklisten- und Arbeitsplanauflösungen, die der Konfiguration z. B. im Kundenauftrag entsprechen. Dabei werden aus Maximalstückliste und -arbeitsplan die nicht benötigten Elemente gelöscht. Anschließend können an den nicht gelöschten Elementen per Beziehungswissen Änderungen vorgenommen werden.

Es werden vier Beziehungswissensarten angeboten: Vorbedingungen, Auswahlbedingungen, Prozeduren und Constraints. Jede Art ist für ein bestimmtes Anwendungsgebiet besonders geeignet. Einige Aufgaben können mit unterschiedlichen Arten von Beziehungswissen abgebildet werden. Die einzelnen Arten von Beziehungswissen und ihre Zuordnung werden im Folgenden vorgestellt.

Vorbedingungen

Vorbedingungen können für den Prozess der Konfiguration in der Bewertungsoberfläche genutzt werden. Ohne Vorbedingungen kann jedes Merkmal mit jedem Wert aus der Liste der zulässigen Werte bewertet werden. Dies gilt in Abhängigkeit der Bewertung anderer Merkmale. Abhängigkeiten können über Vorbedingungen abgebildet werden, indem einzelne Werte oder sogar ganze Merkmale nur unter gewissen Bedingungen erlaubt, ansonsten jedoch für die Bewertung verboten werden. Dazu werden die entsprechenden Vorbedingungen dem Merkmal oder dem Merkmalswert zugeordnet.

Falls mehrere Vorbedingungen an einem Element zugeordnet sind, müssen alle Vorbedingungen erfüllt sein, damit das Element erlaubt ist. Es wird somit wie eine *Und-Verknüpfung* interpretiert.

Wird in einer Vorbedingung die Bewertung eines unbewerteten Merkmals abgefragt, bricht die Vorbedingung ab und wird als erfüllt interpretiert. Weitere Informationen finden Sie in Abschnitt 5.3.7, »Facettenänderungen mit Vor- und Auswahlbedingungen«.

Auswahlbedingungen

Auswahlbedingungen dienen zur Auflösung von Stücklisten und Arbeitsplänen mit Bezug zu einer Konfiguration. Alle nicht mit Auswahlbedingungen versehenen Elemente aus Maximalstückliste und Maximalarbeitsplan werden generell in die aufgelöste Stückliste und den aufgelösten Arbeitsplan übernommen. Man kann bei diesen Elementen von *Gleichteilen*, also für alle Varianten benötigten Teilen, sprechen. Im Unterschied zu diesen wird bei Elementen mit zugeordneten Auswahlbedingungen von *Variantenteilen* gesprochen.

Ist keine der Auswahlbedingungen an einem Element erfüllt, wird dieses Element nicht übernommen. Falls mehrere Auswahlbedingungen an einem Element zugeordnet sind, reicht es, wenn eine Auswahlbedingung erfüllt ist, damit das Element übernommen wird. Es wird somit wie eine *Oder-Verknüpfung* interpretiert (im Unterschied zu Vorbedingungen).

Wird in einer Auswahlbedingung die Bewertung eines unbewerteten Merkmals abgefragt, bricht die Auswahlbedingung ab und wird als nicht erfüllt interpretiert (im Unterschied zu den Vorbedingungen).

Elemente in Stückliste und Arbeitsplan, denen Auswahlbedingungen zugeordnet werden können, sind die folgenden:

- Positionen der Stückliste
- Folgen im Arbeitsplan (außer Stammfolge)
- Vorgänge im Arbeitsplan
- Fertigungshilfsmittel im Arbeitsplan

Auswahlbedingungen werden darüber hinaus für das dynamische Umstellen von Merkmalen zu Muss-Merkmalen verwendet. Soll ein Merkmal nur unter gewissen Bedingungen Muss-Merkmal sein, wird das Merkmal als Kann-Merkmal definiert (in der Merkmalsdefinition ist die Aktivierung des Kennzeichens **Eingabe erforderlich** nicht erfolgt), und es wird dem Merkmal eine entsprechende Auswahlbedingung zugeordnet, die genau diese Bedingung enthält. Weitere Informationen hierzu finden Sie in Abschnitt 5.3.7, »Facettenänderungen mit Vor- und Auswahlbedingungen«, und Abschnitt 5.4.1, »Auswahlbedingungen in Stückliste und Arbeitsplan«.

Prozeduren

Prozeduren dienen der Wertsetzung. Das heißt, dass sie Werte für Merkmale setzen können. Dabei können einem Objekt auch mehrere Prozeduren zugeordnet werden. Prozeduren werden genau einmal abgearbeitet. Die Reihenfolge der Abarbeitung kann und soll vorgegeben werden. Eine solche Vorgabe der Reihenfolge wird sogar dringend empfohlen. Auch kann die Wertsetzung sukzessive erfolgen. Ein Feld kann also mehrfach überschrieben werden – auch unter der Nutzung des bisherigen Wertes.

In der Stücklisten-/Arbeitsplanauflösung können für die Elemente, die in die aufgelöste Stückliste und in den aufgelösten Arbeitsplan übernommen wurden, Änderungen in den Details des entsprechenden Elements durchgeführt werden. Hierzu werden, wie in Abschnitt 4.3.1, »Merkmalspflege«, zu den Werkzeugen aus dem Klassensystem bereits beschrieben, Objektmerkmale mit Bezug auf die Tabellen STPO, ESSL, PLPO, PLFH und PLFL genutzt. (Für Vorgänge können nur die Felder geändert werden, die gleichzeitig in der Struktur PLPO_CFMOD enthalten sind.)



Wertsetzungen über Objektmerkmale

Bei Wertsetzungen in Stückliste und Arbeitsplan über Prozeduren werden nur einzelne Felder (ohne jegliche Berücksichtigung von eventuellen Abhängigkeiten zu anderen Feldern) der aufgelösten Stücklisten und Arbeitspläne überschrieben.

Berücksichtigen Sie, dass dabei keine Funktionen aktiv sind, die Sie aus der normalen Stücklisten- und Arbeitsplanpflege kennen. Alle Abhängigkeiten müssen über Bezie-

lungswissen nachmodelliert werden. Werden z. B. Rohmaße in der Position geändert, muss über die Prozedur auch dafür gesorgt werden, dass die Rohteilmenge angepasst wird. Aus diesem Grund raten wir Ihnen davon ab, Felder mit starken Abhängigkeiten zu ändern, wie z. B. die Materialnummer in der Stücklistenposition oder den Arbeitsplatz im Vorgang.

Prozeduren müssen dem Element zugeordnet werden, dessen Detail geändert werden soll. Die Elemente von Stückliste und Arbeitsplan, denen solche Prozeduren zugeordnet werden können, haben wir bereits für die Auswahlbedingungen aufgezählt: Positionen der Stückliste, Folgen im Arbeitsplan, Vorgänge im Arbeitsplan und Fertigungshilfsmittel im Arbeitsplan.

Prozeduren können ebenfalls für die Bewertung von Merkmalen in der Bewertungsoberfläche genutzt werden. Dabei können Sie die folgenden Wertsetzungen durchführen:

- *harte Wertsetzungen* (d. h. der gesetzte Wert kann vom Benutzer nicht gelöscht oder überschrieben werden)
- *weiche Wertsetzungen* (d. h. vom Benutzer überschreibbare, sogenannte *dynamische Vorschlagswerte*)

Prozeduren für die Bewertungsoberfläche werden dem Konfigurationsprofil zugeordnet. Zusätzlich gibt es die Möglichkeit, Prozeduren auch den Merkmalen oder Merkmalswerten zuzuordnen. Das Konfigurationsprofil wird jedoch favorisiert, da nur so die volle Kontrolle über die Abarbeitungsreihenfolge möglich ist. Des Weiteren werden Prozeduren an Merkmalen und Merkmalswerten nicht in die Laufzeitversion der Wissensbasis mit aufgenommen. In Abschnitt 5.3.4, »Prozeduren«, sowie Abschnitt 5.4.3, »Prozeduren in Stücklisten und Arbeitsplan«, finden Sie weitere Informationen.

Constraints

Constraints können zur Wertsetzung und zur Wertprüfung genutzt werden. Sie enthalten damit Funktionen sowohl der Prozeduren als auch der Bedingungen. *Constraints* können für die Gestaltung der Konfiguration in der Bewertungsoberfläche genutzt werden. Dabei gibt es keine Einschränkungen bezüglich der Konfigurationsszenarien.

Constraints haben besondere Bedeutung in der mehrstufigen Konfiguration, da nur über *Constraints* Abhängigkeiten zwischen beliebigen Objekten einer mehrstufigen Konfiguration abgebildet werden können.

Darüber hinaus können *Constraints* Gleichungen und Variantentabellen umfangreicher auswerten, als dies andere Arten von Beziehungswissen können. Auch das Ein-

schränken von Wertevorräten für die Bewertung von Merkmalen lässt sich mit Constraints meist am besten lösen. Weitere Vorteile liegen in der Verbesserung der Performance. Außerdem sind keine Einstellungen bezüglich der Abarbeitungsreihenfolge erforderlich. Constraints werden in sogenannten *Beziehungsnetzen* oder *Constraint-Netzen* gesammelt, und diese Netze können ausschließlich dem Konfigurationsprofil zugeordnet werden. Sie stehen damit im Allgemeinen nicht zur Auflösung von Stückliste und Arbeitsplan zur Verfügung. Lesen Sie hierzu auch Abschnitt, 5.3.1, »Nutzen Sie Constraints«.



Aktionen können unter SAP S/4HANA nicht gepflegt werden!

Aktionen waren eine weitere Art von Beziehungswissen, die jedoch schon in SAP ERP nicht empfohlen wurden und unter SAP S/4HANA nicht mehr gepflegt werden können.

Aktionen erlaubten es auch, Werte zu setzen. Im Hinblick auf die Funktionalität bieten jedoch Prozeduren und Constraints deutlich mehr. Sie können im Unterschied zu Aktionen in Laufzeitversionen der Wissensbasis aufgenommen werden und sind auch bezüglich der Performance besser.

In Tabelle 5.1 und Tabelle 5.2 fassen wir das Wichtigste zusammen. Tabelle 5.1 listet schlagwortartig auf, wofür die einzelnen Arten von Beziehungswissen in den beiden Anwendungsgebieten genutzt werden können.

Anwendungsgebiet: Bewertungsoberfläche	Anwendungsgebiet: Stücklisten-/Arbeitsplanauflösung
Vorbedingungen	
verbieten/erlauben	–
Auswahlbedingungen	
dynamische Muss-Merkmale	Variantenteile
Prozeduren	
<ul style="list-style-type: none"> ■ wertsetzend hart oder weich ■ dynamischer Vorschlagswert ■ prozedural 	<ul style="list-style-type: none"> ■ Änderungen über Objektmerkmale ■ Bewertung von Klassenknoten

Tabelle 5.1 Anwendungsgebiete von Beziehungswissen

Anwendungsgebiet: Bewertungsoberfläche	Anwendungsgebiet: Stücklisten-/Arbeitsplanauflösung
Constraints	
<ul style="list-style-type: none"> ■ wertsetzend ■ wertprüfend ■ ein-/mehrstufige Konfiguration ■ zusätzliche Funktionalität bezüglich Gleichung(ssystem), Variantentabellen und Werteeinschränkungen ■ performant 	–

Tabelle 5.1 Anwendungsgebiete von Beziehungswissen (Forts.)

In Tabelle 5.2 sehen Sie, wo das Beziehungswissen zugeordnet werden kann – auch wieder abhängig davon, für welches der beiden Anwendungsgebiete Sie es nutzen wollen.

Zuordnung für: Bewertungsoberfläche	Zuordnung für: Stücklisten-/Arbeitsplanauflösung
Vorbedingungen	
Merkmal und Wert	–
Auswahlbedingungen	
Merkmal	<ul style="list-style-type: none"> ■ Position der Stückliste ■ Vorgang, Folge, Fertigungshilfsmittel des Arbeitsplans
Prozeduren	
Konfigurationsprofil	<ul style="list-style-type: none"> ■ Position der Stückliste ■ Vorgang, Folge, Fertigungshilfsmittel des Arbeitsplans
Constraints	
Konfigurationsprofil (in Constraint-Netz)	–

Tabelle 5.2 Zuordnung von Beziehungswissen

5.1.2 Prozeduraler und deklarativer Character von Beziehungswissen

Mit den in Abschnitt 5.1.1 vorgestellten Arten von Beziehungswissen wird sowohl eine prozedurale als auch eine deklarative Herangehensweise an den Konfigurator unterstützt. Diese beiden Herangehensweisen kann man den Arten des Beziehungswissens zuordnen.

■ **Deklaratives Beziehungswissen**

Constraints sind deklaratives Beziehungswissen. Bei diesem Beziehungswissen ist das Ergebnis ausschließlich von der Ausgangsbedingung abhängig. Ein Abarbeitungszeitpunkt oder eine Abarbeitungsreihenfolge ist irrelevant. Analog ist hier auch die Syntax aufgebaut.

■ **Semi-deklaratives Beziehungswissen**

Die Vor- und Auswahlbedingungen stellen semi-deklaratives Beziehungswissen dar. Die Syntax ist rein deklarativ. Auch hier wird ausschließlich die Bedingung beschrieben – ohne jede Aussage zur Abarbeitungsreihenfolge zu tätigen. Die Auswertung erfolgt jedoch prozedural, wie in Abschnitt 5.1.7, »Ausführungsreihenfolge von Beziehungswissen«, beschrieben. Die Vor- und Auswahlbedingungen werden zu genau festgelegten Zeitpunkten im Konfigurationsprozess abgearbeitet.

■ **Prozedurales Beziehungswissen**

Bei Prozeduren handelt es sich um rein prozedurales Beziehungswissen. Analog zu den Vor- und Auswahlbedingungen werden auch diese zu einem genau festgelegten Zeitpunkt im Konfigurationsprozess abgearbeitet. Zusätzlich kann in der Zuordnung der Prozeduren die genaue Abarbeitungsreihenfolge festgelegt werden. Prozeduren erlauben die Verwendung von entsprechenden prozeduralen Sprachelementen wie z. B. für sukzessive Berechnungen, etwa $x = x + 1$, oder das Setzen von Vorschlagswerten, wie es in den Ausführungen zur Syntax beschrieben wird.

Beziehungswissen kann nicht nur in verschiedene Arten unterteilt werden, sondern es kann nach dessen Form der Abarbeitung – prozedural oder deklarativ – unterteilt werden. Darüber hinaus kann Beziehungswissen lokal oder global abgelegt werden. Im nächsten Abschnitt stellen wir lokales und globales Beziehungswissen kurz vor.

5.1.3 Lokales und globales Beziehungswissen

Man unterscheidet lokales und globales Beziehungswissen.

Was zeichnet *lokales Beziehungswissen* aus?

- Lokales Beziehungswissen erkennt man am rein numerischen Namen, z. B. Prozedur 4711. Der Name kann nur durch eine interne Nummernvergabe angelegt werden.
- Lokales Beziehungswissen wird generell aus der Zuordnung heraus angelegt, geändert und gelöscht.
- Es kann nur in der Zuordnung existieren. Wird die Verbindung gelöst, ist auch das Beziehungswissen gelöscht.
- Lokales Beziehungswissen kann nicht mehrfach zugeordnet werden.

Was zeichnet *globales Beziehungswissen* aus?

- Globales Beziehungswissen erkennt man an den nicht numerischen Namen, z. B. Prozedur PROC_CHAR1. Hier wird der Name generell extern, sprich manuell vergeben. Es muss ein Kurztext angelegt werden und manuell der Status auf **Freigegeben** gesetzt werden.
- Für die Pflege von globalem Beziehungswissen existieren eigene Transaktionen. Man kann jedoch auch globales Beziehungswissen aus der Zuordnung heraus anlegen und bei einfacher Verwendung ändern.
- Globales Beziehungswissen kann auch ohne Zuordnung existieren.
- Des Weiteren kann globales Beziehungswissen mehrfach verwendet werden, indem es mehrfach zugeordnet wird.
- Globales Beziehungswissen kann leichter verwaltet und ausgewertet werden.

Eine Ausnahme bezüglich dieser Auszeichnungen

Constraints sind lokales Beziehungswissen. Trotzdem wird der Name hier extern (sprich manuell) vergeben und darf nicht numerisch sein.



Empfehlung für globales Beziehungswissen

Wegen der mehrfachen Verwendbarkeit und der damit verbundenen besseren Performance sowie der leichteren Auswertbarkeit wird in der Variantenkonfiguration globales Beziehungswissen empfohlen.



Wie pflegt man lokales und globales Beziehungswissen? Lokales Beziehungswissen kann nur aus der Zuordnung heraus gepflegt, d. h. angelegt, geändert oder gelöscht werden. Dazu müssen Sie in die Pflegetransaktionen der Objekte gehen, zu denen das lokale Beziehungswissen die Zuordnungen besitzt. Damit sind die folgenden Transaktionen und Apps gemeint:

- die Transaktionen und Apps der Stücklisten- und Arbeitsplanpflege
- die Transaktionen und Apps der Merkmalspflege und der Pflege des Konfigurationsprofils
- Eine weitere Pflegeumgebung für lokales Beziehungswissen ist Transaktion PMEVC.

Der Prozess des globalen Beziehungswissens besteht theoretisch aus zwei Schritten:


1. Pflege des globalen Beziehungswissens

Dies ist unabhängig von der Zuordnung in speziellen Transaktionen und SAP-Fiori-Apps der Variantenkonfiguration (CU01, CU02, CU21, CU22 und PMEVC) möglich.

Abbildung 5.2 zeigt als Beispiel die Merkmalspflege. Lokales Beziehungswissen können Sie immer über den folgenden Menüpfad anlegen: **Zusätze • Beziehungswissen • Editor** ❶. Anschließend erscheint ein kleines Fenster, in dem Sie die Art des Beziehungswissens auswählen. Falls eine Spalte für Beziehungswissen erscheint (hier ❸), hat ein Doppelklick auf die Boxen die gleiche Wirkung wie der Menüpfad für lokales Beziehungswissen. (Diese Spalte existiert, abhängig von der Einrichtung Ihres Systems, leider nicht immer.)



Abbildung 5.2 Beziehungswissen anlegen und mit sonstigen klassischen Transaktionen und Apps zuordnen

Globales Beziehungswissen können Sie immer über den folgenden Menüpfad anlegen: **Zusätze • Beziehungswissen • Zuordnungen** ❷. Sollte der Button  auf dem Bild zu sehen sein ❹, bietet er genau die gleiche Funktionalität an wie der genannte Menüpfad für globales Beziehungswissen. (Dieser Button existiert leider nicht immer.)

Im nächsten Bild sehen Sie eine tabellarische Darstellung der Zuordnungen (siehe Abbildung 5.3.) Hier ist zu erkennen, ob bereits Beziehungswissen zugeordnet wurde. Lokales Beziehungswissen erkennen Sie an dem rein numerischen Namen des Beziehungswissens. Der Name von globalem Beziehungswissen kann auch Buchstaben oder wie in Abbildung 5.3 Unterstriche beinhalten.



Abbildung 5.3 Neues Beziehungswissen mit sonstigen klassischen Transaktionen und Apps anlegen

Weiteres lokales Beziehungswissen kann über den Menüpfad **Bearbeiten • Neue Einzelbeziehung • Lokal** angelegt werden.


Globales Beziehungswissen wird in der in Abbildung 5.3 dargestellten Sicht zugeordnet – egal, ob es bereits vorhanden ist oder erst aus der Zuordnung heraus angelegt werden soll –, indem der Name in die Tabelle der Zuordnungen aufgenommen wird.

Existiert noch kein globales Beziehungswissen, erscheint ein Fenster, in dem Sie das neue Beziehungswissen mit oder ohne Änderungsdienst sowie mit oder ohne Kopiervorlage anlegen können. Als Kopiervorlage für globales Beziehungswissen kann sowohl lokales als auch globales Beziehungswissen dienen.

In den Basisdaten des Beziehungswissens wird die Bezeichnung gepflegt (siehe Abbildung 5.4).

Der Status ist generell zunächst **In Erstellung**. Er kann erst auf **Freigegeben** geändert werden, wenn Syntax existiert und diese auch fehlerfrei ist. Die Beziehung kann einer Beziehungsgruppe zugeordnet werden – analog zu den Objekten im Klassensystem als Strukturierungs- und Selektionswerkzeug. Gleiches gilt in Analogie zu den Objekten des Klassensystems für die Pflegeberechtigungen. Der Verarbeitungsmodus ist einzustellen. Der letzte Punkt in den Basisdaten ist die Auswahl der Beziehungsart. Alle Einstellungen in den Basisdaten sind nachträglich änderbar.

Aus diesem Bild können Sie über einen Klick auf die Buttons **Bezeichnungen** und **Beziehungseditor** zu diesen Funktionen abspringen.

In Abbildung 5.5 ist die SAP-Fiori-App **Stücklisten pflegen** (als Beispiel einer nicht Fiori-Transaktion) dargestellt. Hier existiert keine Spalte (wie in Abbildung 5.2), die die Existenz von geordnetem Beziehungswissen anzeigt. Auch der Button  (Beziehungswissen) fehlt.

SAP Beziehung ändern: Basisdaten

Beziehungseditor Bezeichnungen Mehr

Merkmalname: T_FCLF00 Tragfähigkeit 00
 Beziehung: T_SEL_T_FCLD_FCLE_SPEC_00 SCE Format

Allgemeine Daten

*Bezeichnung: erst T_FCLD00 und FCLDE00 bew Dokumentation
 *Status: 1 Freigabe
 Beziehungsgruppe: T-00 T-00 Beziehungswissen
 Pflegeberechtigung:

Beziehungsart

VerarbeitModus: A Erweiterte Variantenkonfiguration

Vorbedingung Constraint
 Auswahlbedingung Prozedur Regel

Verwaltungsdaten

Angelegt von: S4RD2-00 Angelegt am: 16.07.2019
 Geändert von: Geändert am:

Abbildung 5.4 Globales Beziehungswissen pflegen

SAP Stückliste pflegen

T-FCL00 Verwandte Apps

Gabelstapler T-FCL00 konfigurierbar,Werk:Hamburg,STL-Verwendung:Fertigung,Alternative:01

Ausgewähltes Datum: 30.03.2023 Änderungsnummer auswählen: Zurzeit sind keine ... Änderungsmappen anzeigen

Komponenten Kopfatribute Kopfanlagen Kopflangtext

Standard*

<input type="checkbox"/>	Positio...	Positionstyp	Komponente	Komponentenbeschreibung	Komponentenmenge	Gültig ab	
<input type="checkbox"/>	0010	Lagerposition (L)	T-FV3C00	Gegengewicht	1,000 KG	07.06.2019	>
<input type="checkbox"/>	0020	Klassenposition (K)		Klassenknoten Farben	2,000 L	07.06.2019	>
<input type="checkbox"/>	0030	Lagerposition (L)	T-FV5D01	Sitzeinheit fest mit Standardlehne	1,000 ST	07.06.2019	>
<input type="checkbox"/>	0030	Lagerposition (L)	T-FV5D02	Sitzeinheit fest Standard Kopfstütze	1,000 ST	07.06.2019	>
<input type="checkbox"/>	0030	Lagerposition (L)	T-FV5D03	Sitzeinheit fest mit verlängerter Lehne	1,000 ST	07.06.2019	>
<input type="checkbox"/>	0030	Lagerposition (L)	T-FV5D04	Sitzeinheit drehbar mit Standardlehne	1,000 ST	07.06.2019	>
<input type="checkbox"/>	0030	Lagerposition (L)	T-FV5D05	Sitzeinheit drehbar Standard Kopfstütze	1,000 ST	07.06.2019	>

Abbildung 5.5 Die SAP-Fiori-App »Stücklisten pflegen«

Über einen Klick auf den Button **Beziehungswissen** in Abbildung 5.5 gelangen Sie in die Sicht **Zugeordnetes Beziehungswissen**, die in Abbildung 5.6 dargestellt ist. Hier können Sie weiteres globales Beziehungswissen zuordnen oder lokales und globales Beziehungswissen anlegen.

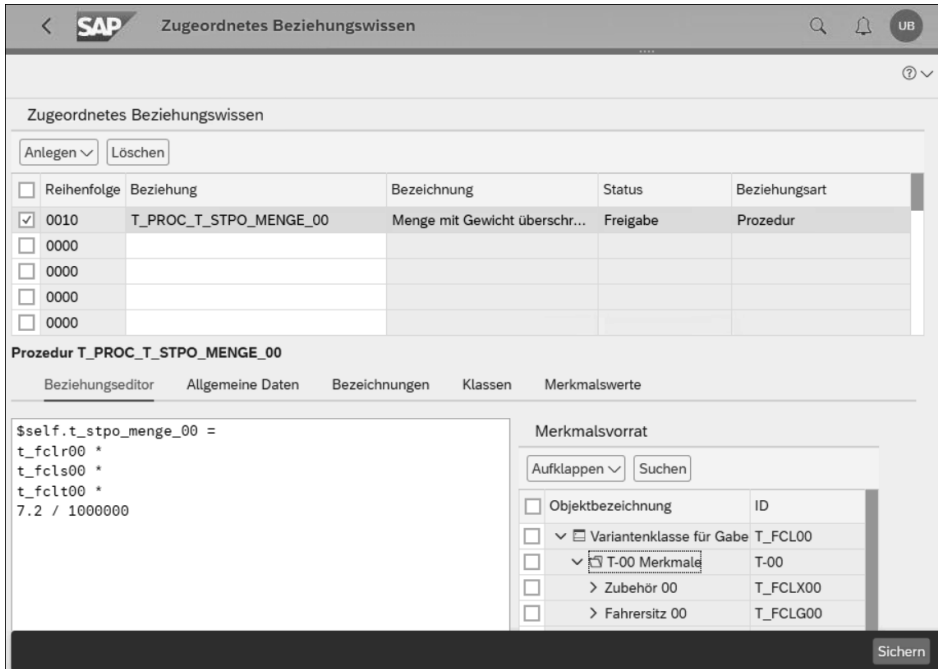


Abbildung 5.6 Die Beziehungswissenspflege in der SAP-Fiori-App »Stücklisten pflegen«

5.1.4 Status von Beziehungswissen

Beziehungswissen – sowohl lokales als auch globales – hat einen *Status*. Es darf erst freigegeben werden, wenn Syntax existiert und diese fehlerfrei ist. Lokales Beziehungswissen wird automatisch freigegeben, wenn vor dem Speichern eine fehlerfreie Syntax angelegt wird. Es ist möglich, fehlerhafte Syntax zu speichern. Das System setzt dann jedoch den Status automatisch auf **Gesperrt**. Dies gilt sowohl bei lokalem als auch bei globalem Beziehungswissen.

Der Status des Beziehungswissens kann im Customizing angepasst werden, falls man nicht mit dem Standard arbeiten möchte. Nach einer Korrektur muss der Status immer manuell auf **Freigabe** gesetzt werden.

Im Unterschied zu lokalem Beziehungswissen muss globales Beziehungswissen meistens manuell freigegeben werden. Ausnahmen sind die Fälle, in denen globales Beziehungswissen vom System angelegt wird. Diese Fälle sind:

- falls mit den Assistenten in Transaktion PMEVC gearbeitet wird
- falls mit Drag & Drop von Merkmalswerten auf Stücklistenpositionen in Transaktion PMEVC gearbeitet wird

Des Weiteren ist zu beachten, dass der Status von Beziehungswissen keine Auswirkung auf seine Zuordnung hat. Globales Beziehungswissen kann mit jedem der folgenden Status zugeordnet werden:

- **In Erstellung**
- **Freigabe**
- **Gesperrt**

Die Zuordnung kann auch bei jedem Status gelöscht werden. Analog können Sie auch für bereits zugeordnetes Beziehungswissen den Status ändern.

5.1.5 Verarbeitungsmodus von Beziehungswissen

Neben dem Konfigurationsprofil hat auch das Beziehungswissen einen *Verarbeitungsmodus*. Auch hier existieren zwei Verarbeitungsmodi:

- Klassisch (LO-VC)
- Erweiterte Variantenkonfiguration (AVC)

Stückliste und Arbeitsplan werden derzeit (Stand: April 2023) grundsätzlich mit der klassischen Konfigurations-Engine (LO-VC-Engine) aufgelöst. Deshalb muss das gesamte Beziehungswissen, das Stückliste und Arbeitsplan zugeordnet wird, den Verarbeitungsmodus **Klassisch** haben. Der Versuch »falsches« Beziehungswissen der Stückliste und dem Arbeitsplan bezüglich Verarbeitungsmodus (d. h., Beziehungswissen mit dem Verarbeitungsmodus **Erweiterte Variantenkonfiguration**) zuzuordnen, führt zu Fehlermeldungen.

Beziehungswissen am Konfigurationsprofil muss den gleichen Verarbeitungsmodus wie das Konfigurationsprofil verwenden. Der Versuch »falsches« Beziehungswissen (d. h., dass der Verarbeitungsmodus des Konfigurationsprofils nicht der gleiche Verarbeitungsmodus des Beziehungswissen ist) bezüglich Verarbeitungsmodus mit Transaktion PMEVC zuzuordnen führt zu Fehlermeldungen. Beziehungswissen mit »falschem« Verarbeitungsmodus wird nicht abgearbeitet.

Beziehungswissen an Merkmalen und Merkmalswerten muss den gleichen Verarbeitungsmodus wie das Konfigurationsprofil verwenden. Es ist möglich, »falsches« Beziehungswissen bezüglich des Verarbeitungsmodus den Merkmalen und den Werten zuzuordnen. Dies liegt daran, dass solche Merkmale in Modellen mit klassischem und mit erweitertem Verarbeitungsmodus enthalten sein können. Beziehungswissen mit »falschem« Verarbeitungsmodus wird nicht abgearbeitet.

In Release 2022 ist in der Klassifizierung nur Beziehungswissen mit klassischem Verarbeitungsmodus wirksam.

Im erweiterten Verarbeitungsmodus (AVC) wird das Beziehungswissen teilweise anders und umfangreicher ausgewertet als im klassischen Verarbeitungsmodus. »Umfangreicher« bedeutet vor allem vorausschauender: Es wird versucht, Inkonsistenzen zu vermeiden. Constraints werden auch ohne Inferences-Teil so abgearbeitet, als hätte der Constraint einen Inferences-Teil; Prozeduren werden nicht so schnell abgebrochen wie ohne AVC.

Bei der Auswertung von Constraints wird über eine Werteeinschränkung versucht, größere Inkonsistenzen zu vermeiden. Diese umfangreichere Auswertung kann vermieden werden, indem Bedingungen grundsätzlich in den `CONDITION`-Teil des Constraints aufgenommen werden und nicht als `IF`-Bedingung im `RESTRICTIONS`-Teil erscheinen. Mehr dazu finden Sie in Abschnitt 5.3, »Beziehungswissen für die interaktive Konfiguration (High-Level-Konfiguration)«.

Vorbedingungen wurden bis Release SAP S/4HANA 2021, On-Premise-Version, oder SAP S/4HANA Cloud 2108 ebenfalls mit diesem umfangreicheren Ansatz wie für Constraints ausgewertet. Mit SAP-Hinweis 3060829 besteht in diesem Release die Möglichkeit, auf die klassische Verarbeitung wie dem klassischen Verarbeitungsmodus auch unter AVC zurückzukehren. Ab SAP S/4HANA, On-Premise-Version 2022 oder SAP S/4HANA Cloud 2111 werden die Vorbedingungen auch in AVC nicht umfangreicher als im klassischen Verarbeitungsmodus ausgewertet.

Sie können mehrere Anweisungen in eine Prozedur aufnehmen. Trennen Sie diese mit Komma ab. Im klassischen Verarbeitungsmodus wird die Prozedur ab der Anweisung abgebrochen, ab der sie nicht ausgeführt werden kann. Im erweiterten Verarbeitungsmodus (in AVC) wird nur die Anweisung abgebrochen, die nicht ausgeführt werden kann. Im Gegensatz zum klassischen Vorgehen versucht das System weiteren Anweisungen abzuarbeiten. Ab SAP S/4HANA, On-Premise-Variante 2022, oder SAP S/4HANA Cloud 2111 ist es möglich, über das Syntaxelement `EXIT` einen Abbruch analog zum klassischen Verarbeitungsmodus zu erzwingen.

5.1.6 Beziehungswissen in der Klassifizierung und in der Variantenkonfiguration

Beziehungswissen kann neben der Variantenkonfiguration auch in der Klassifizierung genutzt werden und ist dort aktiv. Im Folgenden lernen Sie die Gemeinsamkeiten und Unterschiede kennen.

Beziehungswissen an Merkmalen und Merkmalswerten ist in der Klassifizierung und in der Variantenkonfiguration gleich. Dies betrifft die Vor- und Auswahlbedingungen sowie Prozeduren.

Werden Merkmale mit zugeordnetem Beziehungswissen sowohl in der Klassifizierung (Klassen, die keine Variantenklassen sind) als auch in der Variantenkonfiguration (Variantenklassen) genutzt, ist auch das Beziehungswissen dieser Merkmale aktiv. Das kann gewollt sein. Werden z. B. die gleichen Merkmale zur Klassifizierung in Klassenknoten und in der Konfiguration genutzt, wird häufig gewünscht, dass an beiden Stellen die gleichen Regeln gelten. Ansonsten sollten Sie es vermeiden, Merkmale in beiden Anwendungen gleichzeitig zu verwenden, indem Sie die Verwendung von Merkmalen in der Merkmalsdefinition auf bestimmte Klassenarten einschränken oder indem Sie Merkmalen nur in der klassenspezifischen Überschreibung das Beziehungswissen zuordnen.

Anders verhält sich das Beziehungswissen an anderen Objekten: Beziehungswissen an Klassen ist nur in der Klassifizierung aktiv. Auch können Sie den Variantenklassen Beziehungswissen zuordnen. Dies zeigt jedoch keine Wirkung in der Variantenkonfiguration und sollte aus diesem Grund vermieden werden.

Prozeduren an Klassen

Prozeduren sind das einzige Beziehungswissen, das Klassen zugeordnet werden kann. Diese den Klassen zugeordneten Prozeduren werden aber nur in der Klassifizierung abgearbeitet, nicht jedoch in der Variantenkonfiguration. Das heißt, dass die den Klassen zugeordneten Prozeduren in der Klassifizierung die Bedeutung haben, die die Prozeduren in der Variantenkonfiguration am Konfigurationsprofil haben.



Beziehungswissen an weiteren Objekten wie Konfigurationsprofil, Stückliste und Arbeitsplan sind nur in der Variantenkonfiguration aktiv.

Inaktives Beziehungswissen in der Suche im Klassensystem


Bei den Suchfunktionen im Klassensystem ist das Beziehungswissen inaktiv. Das heißt, dass Sie nach Bewertungskombinationen suchen können, die in der Form weder in der Klassifizierung noch in der Konfiguration möglich sind.

Dies liegt daran, dass bei der Suche im Klassensystem alle Einschränkungen bezüglich des Formats der Merkmale abgeschaltet sind. So kann immer mehrwertig und intervallwertig gesucht werden. Es ist weder technisch noch inhaltlich durchführbar, alle Einschränkungen abzuschalten und trotzdem das Beziehungswissen abzuarbeiten.



5.1.7 Ausführungsreihenfolge von Beziehungswissen

Wir betrachten in diesem Abschnitt die Ausführungsreihenfolge von Beziehungswissen. Wir haben gesehen, dass verschiedene Arten von Beziehungswissen an den verschiedensten Stellen genutzt werden können. Außerdem können Sie BADIs nutzen.

Wir beginnen mit der Ausführungsreihenfolge im erweiterten Verarbeitungsmodus (in AVC) bei der *High-Level-Konfiguration*. Das heißt, dass bei der interaktiven Konfiguration, der Konfiguration »im Hellen« bzw. der Konfiguration im Dialog. Starten Sie die Abarbeitung des Beziehungswissens mit der -Taste. Anschließend beginnt die Abarbeitung mit dem \$root-Objekt:

1. Werte, die durch Prozeduren gesetzt wurden, werden zurückgenommen.
2. Das BAdI VCH_HL_PRE_VALIDATE_ASSIGN wird abgearbeitet.
3. Das Beziehungswissen an der \$root-Instanz wird folgendermaßen abgearbeitet:
 - Constraints ausschließlich für das \$root-Objekt werden abgearbeitet. Falls sich an den Bedingungen sich etwas ändert, werden sie wiederholt abgearbeitet.
 - Statische Vorschlagswerte aus der Merkmalsdefinition werden gesetzt, und statische Vorschlagswerte, die zu Inkonsistenzen führen, werden gleich wieder zurückgezogen.
 - Alle Prozeduren werden genau einmal ausgewertet. Dies geschieht in der Reihenfolge: Konfigurationsprofil → Merkmale → Merkmalswerte. Die Ausführungsreihenfolge der Prozeduren pro Objekt wird auf der Ebene der Zuordnung der Prozeduren festgelegt.
 - Alle Vorbedingungen werden abgearbeitet
 - Alle Auswahlbedingungen werden abgearbeitet
4. Laut Stücklistenauflösung der nächsten Ebene und »links« beginnend wird analog mit Schritt 3 fortgefahren. Es wird immer eine weitere Instanz zusätzlich berücksichtigt.
5. Zum Abschluss werden die Prozeduren auf der \$root-Ebene erneut abgearbeitet, u. a., um die Preisfindung zu aktualisieren.
6. Das BAdI VCH_HL_POST_VALIDATE_ASSIGN wird abgearbeitet.
7. Wird ab Schritt 3 beziehungsweise 4 auf einer Instanz eine nicht behebbare Inkonsistenz festgestellt, bricht die Abarbeitung an dieser Stelle ab. (Dies ist nicht so unter LO-VC im klassischen Verarbeitungsmodus.)
8. Es existieren zwei weitere BAdIs, die beim Absprung in die Konfiguration (VCH_HL_MD_DOMAIN_MODIFY) und aus der Konfiguration (VCH_HL_ON_SAVE) abgearbeitet werden.

Die Ausführungsreihenfolge im *klassischen Verarbeitungsmodus* (LO-VC) verläuft folgendermaßen:

1. Die Vorschlagswerte aus der Merkmalsdefinition werden gesetzt
2. Alle Aktionen werden mehrmals ausgeführt, bis keine weiteren Werte hergeleitet werden können.
3. Alle Prozeduren werden in der folgenden Reihenfolge genau einmal ausgewertet: Konfigurationsprofil → Merkmale → Merkmalswerte. Die Ausführungsreihen-

folge der Prozeduren pro Objekt wird auf der Ebene der Zuordnung der Prozeduren festgelegt.

4. Wie in Schritt 2 werden alle Aktionen abgearbeitet, soweit sich durch die Prozeduren Änderungen ergeben haben. Parallel zu den Schritten 1 bis 4 werden sofort alle die Constraints abgearbeitet, für die sich relevante Veränderungen ergeben haben.
5. Erst werden alle Vorbedingungen und dann alle Auswahlbedingungen abgearbeitet.

Prozeduren an Merkmalen und Merkmalswerten vermeiden

Vermeiden Sie Prozeduren an Merkmalen und Merkmalswerten. Bei Prozeduren mit ihrem prozeduralen Charakter ist die Abarbeitungsreihenfolge entscheidend. Es gibt die folgenden festen Regeln:

1. Die Prozeduren am Konfigurationsprofil werden abgearbeitet.
2. Die Prozeduren an den Merkmalen werden abgearbeitet, soweit diese Merkmale bereits bewertet sind.
3. Zum Schluss erfolgt die Abarbeitung der Prozeduren an den Merkmalswerten, soweit diese Merkmalswerte bereits als Bewertung genutzt wurden.

An jeder Stelle, an der Sie die Prozeduren zuordnen können, lässt sich über die Sortierung die Abarbeitungsreihenfolge festlegen.

Eine Einschränkung gibt es jedoch: Wenn mehreren Merkmalen Prozeduren zugeordnet sind, ist die Abarbeitungsreihenfolge zwischen den Merkmalen zufällig. Dies führt zu einer nicht kontrollierbaren Reihenfolge. Wenn mehreren Merkmalswerten Prozeduren zugeordnet sind, ist die Abarbeitungsreihenfolge zwischen den Merkmalswerten zufällig. Auch dies ist nicht erwünscht, da es nicht steuerbar ist.

Aus diesen Gründen werden Prozeduren an Merkmalen und Merkmalswerten nicht empfohlen und teilweise sogar verboten. Zum Beispiel werden in die Laufzeitversion keine Prozeduren an Merkmalen und Merkmalswerten übernommen.

Abschließend betrachten wir mit Abbildung 5.7 noch die Ausführungsreihenfolge des Beziehungswissens für Stückliste und Arbeitsplan. Die Ausführungsreihenfolge des Beziehungswissens für Stückliste und Arbeitsplan, der sogenannten *Low-Level-Konfiguration*, ist wie folgt:

1. Auswahlbedingungen an Objekten
2. Prozeduren an Objekten, soweit diese Objekte aufgrund der Auswahlbedingungen nicht ausgeschlossen wurden.

Das Beziehungswissen an Stückliste und Arbeitsplan nutzt derzeit (Stand: April 2023) stets den klassischen Verarbeitungsmodus. Dabei spielt der Verarbeitungsmodus des Konfigurationsprofils keine Rolle.



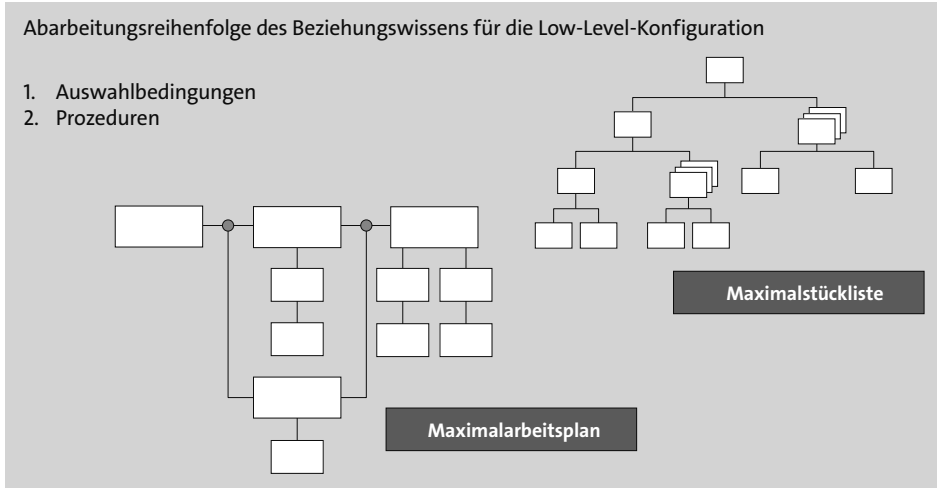


Abbildung 5.7 Ausführungsreihenfolge des Beziehungswissens für Stückliste und Arbeitsplan

5.1.8 Grundregeln der Syntax

Das Beziehungswissen folgt eigenen syntaktischen Regeln. Die Syntax ist keine ABAP-Programmierung und entspricht auch keiner anderen geläufigen Programmiersprache. Über BADIs und sogenannte *Variantenfunktionen* besteht die Möglichkeit, ABAP-Coding in die Abarbeitung von Beziehungswissen zu integrieren – und damit stehen auch ABAP-Funktionen zur Verfügung.



BADIs statt Variantenfunktionen derzeit unter AVC nutzen.

Die vier BADIs aus Abschnitt 5.3.8 stehen ausschließlich unter AVC zur Verfügung. Variantenfunktionen können derzeit (Stand: April 2023) nicht unter AVC genutzt werden.

Das Ergebnis der Abarbeitung von Beziehungswissen kann folgendermaßen aussehen:

- Vor- und Auswahlbedingungen sind boolesche Ausdrücke. Das heißt, dass sie ein binäres Ergebnis liefern.
- Prozeduren liefern beliebige Werte als Ergebnis.
- Constraints liefern im Rahmen der Wertsetzung entsprechende Werte und im Rahmen der Konsistenzprüfung ein binäres Ergebnis.

Wir betrachten im Folgenden die Syntaxregeln detaillierter.

Namen von Klassen und Merkmalen

Die Syntax ist sprachunabhängig und nutzt die sprachneutralen Namen von Klassen, Merkmalen, Merkmalswerten, Materialstämmen und Dokumenten. Diese Namen werden direkt in die Syntax aufgenommen. Einzige Ausnahme sind Merkmalswerte im Zeichenformat (CHAR). Diese werden in Hochkommata gesetzt. Darüber hinaus kann ausschließlich in Constraints mit Variablen für Objekte und Merkmale gearbeitet werden. Diese Variablen werden pro Constraint spezifisch definiert.

Beachten Sie im Hinblick auf Klassen und Dokumente, dass der Objektschlüssel aus mehreren Teilen besteht. Bei Materialstämmen und Dokumenten müssen Sie den Objekttyp mit angeben. Ein Beispiel für Klasse, Materialstamm und Dokument in der Syntax von Beziehungswissen zeigt Listing 5.1.

```
(300)klasse1
(MATERIAL)()(NR='material2')
(document)()(TYPE = 'DRW', VERSION = '00', PART = '000',
             NR = 'D4545')
```

Listing 5.1 Klasse, Materialstamm und Dokument im Beziehungswissen

Groß- und Kleinschreibung, Kommentare

Sie können die Groß- oder Kleinschreibweise beliebig verwenden. Ausgenommen davon sind die Merkmalswerte beim Zeichenformat CHAR, bei denen zusätzlich das Kennzeichen **Groß-/Kleinschreibung** gesetzt wurde. Nutzen Sie dieses Kennzeichen nur, wenn es unbedingt notwendig ist. Das Kennzeichen ist leider bei Objektmerkmalen für Variantenbedingungen erforderlich (siehe Kapitel 6, »Preisfindung«).

Groß- und Kleinschreibung

Syntaxelemente können beliebig groß- oder kleingeschrieben werden. Beides ist gleichbedeutend und kann auch gemischt verwendet werden (Syntaxbeispiele geben wir im Verlauf des Kapitels).

Auch können Sie sprachneutrale Kommentarzeilen in die Syntax aufnehmen, indem Sie einen Stern * in der ersten Spalte voranstellen. Es gibt auch die Möglichkeit der sprachspezifischen Kommentare im Beziehungswissen. Dies wird jedoch nicht in die Syntax aufgenommen, sondern ist unter **Zusätze • Dokumentation** in den Basisdaten des Beziehungswissens zu finden.

Leerzeichen und Zeilenumbrüche können an beliebiger Stelle eingefügt werden; Aufzählungen von mehreren Anweisungen sind in einem Beziehungswissen möglich und werden dann durch Kommata getrennt.



Daraus ergibt sich, dass die folgenden drei Objekte in dieser Beziehung ansprechbar sind:

- **\$self**

Wenn `$self` einem Merkmalsnamen vorangestellt ist, kann die Merkmalsbewertung am Objekt selbst angesprochen werden. Prozeduren können generell nur auf dieser Ebene Werte setzen.

- **\$root**

Wenn `$root` einem Merkmalsnamen vorangestellt ist, kann die Merkmalsbewertung am Kopfobjekt angesprochen werden. Dies ist die oberste Instanz bezüglich der Konfigurationsstruktur und die oberste Ebene, auf der die Merkmalsbewertung gespeichert wird.

- **\$parent**

Wenn `$parent` einem Merkmalsnamen vorangestellt ist, kann die Merkmalsbewertung an der nächsten Instanz oberhalb der `$self`-Ebene angesprochen werden. Soweit das Kopfmaterial der einstufigen Stückliste, in der einer Komponente die Beziehung zugeordnet ist, auch eine eigene Bewertung bezüglich der Konfiguration hat, ist dies die `$parent`-Instanz. Ansonsten wandert diese Ebene weiter nach oben.

```
$self.merkmal3 = $parent.merkmal2 + $root.merkmal1
```

```
$SELF.MERKMAL3 = $PARENT.MERKMAL2 + $ROOT.MERKMAL1
```

Im Beispiel aus Abbildung 5.8 kann die erste Baugruppe unter dem Kopfmaterial nicht angesprochen werden, da die Ebenen zwischen `$PARENT` und `$ROOT` nicht angesprochen werden können. Das Gleiche gilt für Ebenen unterhalb der `$self`-Ebene. Anders gesagt: Es gibt kein `$grandparent`, `$children` oder ähnliche verwandtschaftliche Beziehungen.

Es kann passieren, dass mehrere Ebenen zusammenfallen. Würde in unserem Beispiel auf der Kopfebene z. B. das Szenario **Plan-/Fertigungsauftrag ohne Stücklistenauflösung** genutzt, würden alle darunterliegenden Konfigurationsprofile ignoriert. Damit würden diese Ebenen nicht als Instanzen im Konfigurationsprozess erscheinen, und das Kopfmaterial wäre von allen Elementen aus der Stücklistenstruktur heraus sowohl als `$root` als auch als `$parent` ansprechbar. Damit fallen diese beiden Ebenen zusammen. Betrachten wir das mittlere Beispiel ② in Abbildung 5.9.

Es können sogar alle drei Ebenen zusammenfallen, wenn das Beziehungswissen auf der Ebene des Kopfmaterials, d. h. des dortigen Konfigurationsprofils, zugeordnet wird. Das sehen Sie unter ① in Abbildung 5.9.

Nur in der mehrstufigen Konfiguration werden mit diesen drei Syntaxelementen drei verschiedene Ebenen angesprochen (siehe das Beispiel ③ in Abbildung 5.9).

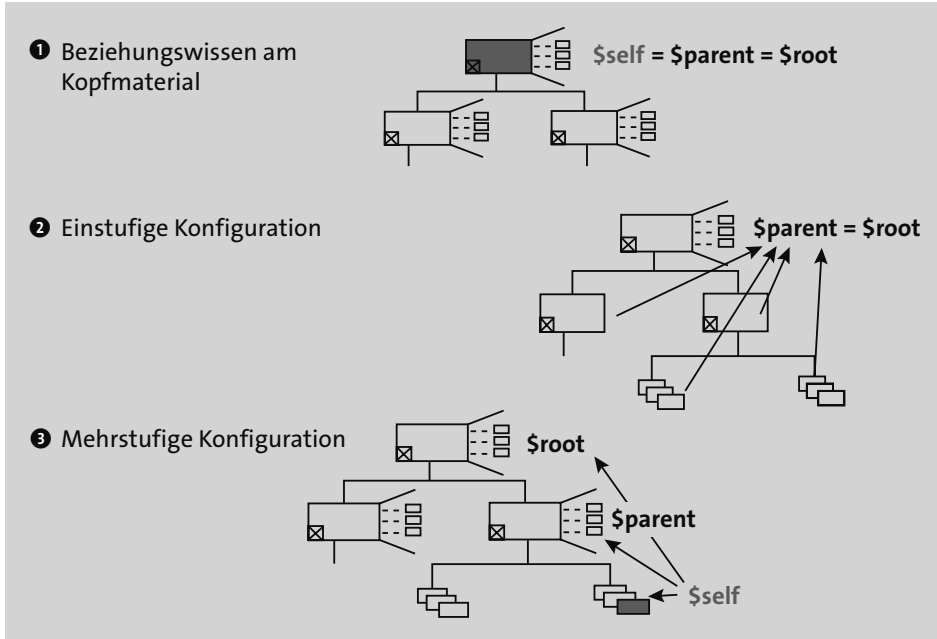


Abbildung 5.9 $\$self$, $\$parent$ und $\$root$ in speziellen Fällen im Beziehungswissen

5.1.9 Syntaxelemente

In Beziehungswissen können neben den bereits vorgestellten Namen von Merkmalen, Merkmalswerten, Materialstämmen und Dokumenten sowie $\$self$, $\$parent$ und $\$root$ weitere Syntaxelemente genutzt werden.

- Sie können beliebige Klammern nutzen.
- Sie können Aufzählungen mit Kommas verwenden.
- Anweisungen können (müssen aber nicht) mit einem Punkt beendet werden.
- Sie können Kommentarzeilen mit einem Stern * beginnen lassen. Ein teilweises Auskommentieren ist hingegen nicht möglich.

Als *logische Operatoren* stehen zur Verfügung:

- Negation: NOT
- Bedingungen: IF
- Verknüpfungen: AND, OR
- Vergleichsoperatoren:
 - kleiner als: <, LT
 - kleiner gleich: <=, =<, LE
 - gleich: =, EQ

- größer gleich: \geq , \Rightarrow , GE
- größer als: $>$, GT
- ungleich: $<>$, \neq , NE

Als *arithmetische Operatoren* stehen zur Verfügung:

- Grundrechenarten: +, -, /, *, **
- Standardfunktionen:
 - Winkelfunktionen: sin, cos, tan
 - Exponentialfunktion (Basis e): exp
 - Logarithmusfunktionen (Basis e und 10): ln, log10
 - Arcus Funktionen: arcsin, arccos, arctan
 - Quadratwurzel: sqrt
 - Absolutbetrag: abs
 - Signum: sign
 - Dezimalanteil: frac
 - ganzzahliger Teil: trunc
 - aufgerundet: ceil
 - abgerundet: floor

Als *Zeichenkettenoperatoren* stehen zur Verfügung:

- die Verknüpfung: ||
- die Umwandlung in Kleinbuchstaben: LC
- die Umwandlung in Großbuchstaben: UC

Des Weiteren können Sie mit dem Syntaxelement `in` Listen abfragen. Bei numerischen Merkmalen können diese Listen auch generell Intervalle enthalten. Beispiele sehen Sie in Listing 5.2:

```
farbe in ('r', 'b', 'g')
breite in (100, 200 - 400, 600).
```

Listing 5.2 Beispiele für eine Listenabfrage mit »in«

Über `specified` kann abgefragt werden, ob ein Merkmal bewertet ist. Auch negative Abfragen sind möglich. Beispiele dazu finden Sie im folgenden Listing 5.3:

```
farbe specified
specified farbe
not specified farbe.
```

Listing 5.3 Beispiele für die Abfragen mit »specified«

In Beziehungswissen außer Constraints können Sie mit dem Syntaxelement `type_of` das Objekt auf der entsprechenden Ebene abfragen. Im Allgemeinen wird hierbei die Variantenklasse auf der Kopfebene abgefragt (siehe Listing 5.4).

```
type_of($root, (300)klasse1)
type_of($root, (material(300)(nr='material1')))
```

Listing 5.4 Beispiele für Abfragen mit »type of«



Keine negativen Abfragen bei Constraints

Als einzige Ausnahme bezüglich negativer Abfragen mit den Syntaxelementen `not in`, `not type_of` und `not specified` lassen Constraints diese Negationen nicht zu. In Constraints sind dies Syntaxfehler.

Weitere Syntaxelemente finden Sie in der folgenden Aufzählung:

- `mdata`
Dieses Element dient der Abfrage der »Masterdaten«, also der Originaldaten in den Stammdaten.
- `?=`
Dieses Element ist für das Setzen von Vorschlagswerten (nur in Prozeduren erlaubt) vorgesehen. Das Syntaxelement `?=` kann auch in Aufrufen von Variantentabellen und -funktionen verwendet werden. Es existiert auch das Syntaxelement `$set_default` mit gleicher Wirkung, aber umfangreicherer und damit umständlicherer Syntax:
`$set_default($self, <merkmal> , <wert>)`
- `$del_default`
Dieses Syntaxelement ist für das Löschen von mit dem Syntaxelement `$set_default` oder `?=` gesetzten Vorschlagswerten (nur in Prozeduren erlaubt) vorgesehen. Diese Syntax ist analog zu der Syntax von `$set_default` aufgebaut:
`$del_default($self, <merkmal> , <wert>)` oder
`$set_default($self, <merkmal> , <merkmal>)`
- `$sum_part`
Dieses Element ist für das Aufsummieren eines numerischen Merkmals in einer aufgelösten Stückliste (nur in Prozeduren erlaubt) vorgesehen.
- `$count_part`
Für das Aufsummieren der Komponentenmenge in einer aufgelösten Stückliste (nur in Prozeduren erlaubt) ist dieses Element vorgesehen.
- `$part_of`
Dieses Syntaxelement ist für das Abfragen von Objekten einer aufgelösten Stückliste (nur in Constraints erlaubt) vorgesehen.

- `$subpart_of`
Für das Abfragen von Objekten einer beliebig tief aufgelösten Stücklistenstruktur (nur in Constraints erlaubt) ist dieses Syntaxelement vorgesehen.
- `$set_pricing_factor`
Mit diesem Syntaxelement kann einem Konditionssatz für die Preisfindung nachträglich noch ein Faktor zugeordnet werden (nur in Prozeduren erlaubt).
- `skey`
Das Syntaxelement `skey` und ein in Hochkommata gesetzter Merkmalsname sind immer auch dann erforderlich, wenn im Merkmalsnamen ein Bindestrich vorkommt.

Ausschließlich unter dem erweiterten Verarbeitungsmodus (AVC) verfügbar:

- `round(x, i)`
Rundet Float-Merkmale `x` auf `i` Nachkommastellen (`i` ganze Zahl ≥ 0).
- `or, and, not`
Auch zur Werteeinschränkung in `RESTRICTIONS` in Constraints verwendbar.
- `set_pricing_factor`
Ist jetzt auch in Constraints anwendbar.
- `min(x), max(x)`
Ermittelt minimale und maximale erlaubte Wert eines eingeschränkten Wertebereichs.
- `$del_user_val(x)` oder `$del_user_val(x, y)`
Löscht Benutzereingaben.
- `$del_val(x)` oder `$del_val(x, y)`
Löscht alle Bewertungen außer Bewertungen, die durch Constraints gesetzt wurden (d. h. Vorschlagswerte aus Merkmalsdefinitionen sowie hart und weich gesetzte Werte durch Prozeduren und Benutzereingaben).
- `substring(x, y, z)`
Teilstring von `x` ab `y`; Stelle der Länge `z`.
- `substring(x, y)`
Teilstring von `x` ab `y`; Stelle bis zum Ende.
- `prefix(x, y)`
Teilstring von `x` ab 1; Stelle der Länge `y`.
- `suffix(x, y)`
Teilstring von `x` der Länge `y` am Ende.
- `assigned_vals(x)`
Bei der Verwendung in einem IF-Teil einer Prozedur erlaubt `ASSIGNED_VALS` Prüfungen, die nur Zuweisungen berücksichtigen.

- `no_of_assigned_vals(x)`
Gibt die Anzahl der Werte zurück, die x zugewiesen werden, wenn die Anweisung ausgeführt wird.
- `exit`
Bricht in einer Prozedur an der entsprechenden Stelle ab. Wird meist mit `if`-Bedingung genutzt. (Kann nur in Prozeduren genutzt werden. Dient dazu, unter AVC eine Verarbeitung von Prozeduren zu ermöglichen, die der Verarbeitung unter LO-VC entspricht.)
- `to_int(x)`
Erzeugt aus einem String eine ganze Zahl.
- `to_string(x)`
Erzeugt aus einer Zahl einen String.

Diese Syntaxelemente wurden hier nur kurz aufgezählt. In den nächsten Abschnitten, insbesondere in Abschnitt 5.3, »Beziehungswissen für die interaktive Konfiguration (High-Level-Konfiguration)«, und Abschnitt 5.4, »Beziehungswissen für Stückliste und Arbeitsplan (Low-Level-Konfiguration)«, gehen wir auf einige Syntaxelemente ausführlicher ein.

5.2 Variantentabellen

Die verschiedenen Eigenschaften eines konfigurierbaren Materials sind häufig voneinander abhängig. Dabei beeinflusst die Bewertung eines Merkmals die noch wählbaren Werte eines anderen Merkmals. Natürlich kann man solche Abhängigkeiten direkt mit einer einfachen Syntax ausdrücken. Wenn jedoch mehrere Wertekombinationen zwischen denselben Merkmalen möglich sind, ist eine Aufzählung jeder einzelnen Kombination in Form einer eigenen Befehlszeile umständlich. In solchen Fällen nutzen Sie Variantentabellen.

Variantentabellen können in jeder Art von Beziehungswissen und sowohl in der High-Level- als auch in der Low-Level-Konfiguration verwendet werden.

5.2.1 Eine Tabellenstruktur anlegen

Eine Variantentabelle ist ein eigenes Stammdatenobjekt, das in der Modellierungsumgebung für Variantenkonfigurationen angelegt werden kann. Hierzu öffnen Sie im Teilbereich **Umfeld** das Kontextmenü zum Unterpunkt **Variantentabelle** und klicken auf den Eintrag **Variantentabelle anlegen...** (siehe Abbildung 5.10).

Vorwort

Ein persönlicher Blick zurück

Die makroökonomische Bedeutung der Mass-Customization und weitergehende Einführungen in das Thema Produktkonfiguration wurden bereits in den bisherigen Vorworten zum Buch »Variantenkonfiguration mit SAP« von unserem sehr geschätzten früheren Entwicklungsvorstand Peter Zencke und von Albert Haag, unserem früheren Entwicklungskollegen (und auch von anderen Personen), bestens beschrieben.

Hier wollen wir daher aus einem sehr persönlichen Blickwinkel schildern, wie es vor vielen Jahren zu drastischen Erweiterungen der Ansätze zur Variantenkonfiguration kam. Als für einige von uns im Jahre 1990 die Karriere bei SAP begann, gab es nur einfachste Zuordnungsbeziehungen von Verkaufsmerkmalen, Verkaufsgruppen und Materialien. Nun blicken wir auf 33 Jahre der Weiterentwicklung der Lösungsansätze und deren Integration in die neuesten SAP-ERP-Systeme zurück.

Das Zusammenleben basiert auf vielen Beziehungen, von denen die wenigsten hierarchisch sind. Und doch haben wir uns viele Jahre lang darauf beschränkt, solche hierarchischen Beziehungen als die einzige Darstellungsform zu ermöglichen, um Objekte zu beschreiben und diese Information durch Softwareanwendungen verarbeitbar zu machen.

Dabei waren entweder einfache singuläre Objekte oder (was die Sache schon deutlich komplizierter machte) ein untergeordneter Teil eines größeren Ganzen möglich. Es gab sogenannte Eltern-Kind-Beziehungen, Positions-Unterpositionsbeziehungen oder Has-Part-is-Part-of-Beziehungen. Und für diese haben wir mühsam Ende der 80er Jahre Ansätze (außerhalb von SAP) entwickelt und flexibilisiert. Ziel war es, variantenreiche Produkte mit einer Masterstückliste und auswählbaren Teilen zu beschreiben und im Arbeitsplan mit dem dazugehörigen Beziehungswissen den Auswahlvorgang zu steuern.

Es war ein Meilenstein, diese Flexibilität in die rigide Auftragsabwicklung und Stücklisten- und Arbeitsplanverarbeitung eines ERP-Systems einzubauen und langsam Stück für Stück durchgängig verarbeitbar zu machen. Ein Prozess der leider (wie so oft) viel länger dauerte als angenommen und der vermutlich nach 37 Jahren immer noch nicht zu Ende ist. Und natürlich führte die Arbeit mit SAP-Kunden immer wieder dazu, die Lösungsansätze zu verbessern, um Probleme besser lösen zu können (z. B. die Variantentabellen als pragmatischen Ansatz einzuführen, um mit großen Daten umzugehen). Aber die wesentlichen Grundlagen waren bereits im letzten Jahrtausend von uns und anderen Personen aufgeschrieben worden.

Der Ansatz, Auswahlbedingungen mit Stücklistenpositionen zu verbinden, geht auf die wissenschaftlichen Arbeiten von Bernhard und seinen damaligen Institutskolle-

gen der Universität Karlsruhe und Duisburg sowie natürlich weiterer »Prior Art« (1985–1990) zurück.

Die Verbindung von Type-of/is-a(-Klassifikation)-Relationen und Stücklisten basiert bei SAP auf dem gemeinsamen Arbeiten des KI- und des Logistik-Entwicklungsteams. Mitte der 90er Jahre haben wir dabei »einfach« Ansätze der Objektorientierung übertragen.

Und natürlich wurden durch die Einführung der Verarbeitung von Constraints alle Ansätze auf ein deutlich anspruchsvolleres Niveau gehoben. Die Constraints waren besonders Albert und den wissenschaftlichen Vorarbeiten des Plakon-Projekts zu verdanken. Eines unserer Lieblingsthemen war die Einführung des Assumption-based Truth Maintenance System, mit dem es endlich möglich wurde, die starre Reihenfolge von Eingaben zu flexibilisieren. Man konnte sich plötzlich auch umentscheiden und sofort die Konsequenzen sehen.

Um aber die immer noch zu starre Welt der Maximalstücklisten (hierarchische Beziehungen) zu verlassen, bedurfte es dreier besonderer Ereignisse.

Die Herausforderung

Wir (das KI-Team) wurden vom SAP-Vorstand aufgefordert zu beweisen, dass unsere Ansätze besser sind als die eines anderen Unternehmens (das gerne mit SAP ins Geschäft kommen wollte. Eine Situation, die ich in den folgenden 30 Jahren in etlichen Bereichen noch sehr häufig erleben musste). Der Haken war, dass die anderen die Aufgabe auswählen durften. Es ging um die Konfiguration aller Teile eines Treppengeländers. In wenigen Wochen haben Albert, Andreas und Kollegen die erste Version der Constraints-Verarbeitung implementiert. Die interne Verwaltung von Objektinstanzen ermöglichte es, bei einer flexibel langen Treppe mit freiwählbarer Steigung und Geländerhöhe (und flexibler Anzahl von Stockwerken) für jeden einzelnen Pfosten die Position auf der Stufe sowie die Höhe des Pfostens und damit alle Angaben der benötigten Materialien auszurechnen. Dies geschah unter der Nutzung der Constraints mit mathematischen Formeln für Anfangspfosten, Endpfosten und Treppenfosten. Ein Ansatz, der das auf Entscheidungstabellen basierende Konkurrenzsystem ganz blass aussehen ließ.

Das Anforderungspapier der Gründungsmitglieder der Configuration Workgroup (CWG)

Bernhard besuchte im Jahr 1993 oder 1994 den Kunden Steelcase und lernte dabei, das Wort »Pedestal« richtig auszusprechen. Außerdem erkannte er dort dank Henk Meeter und anderen Steelcase-Mitarbeitern, dass man eine Schreibtischlandschaft nicht

mit Auswahlbedingungen und nur sehr ungenau mit dem Bilanzkonzept (Offers/Requies) beschreiben konnte. Auch die weiteren Treffen mit IBM, Siemens, Applied Materials und Hewlett Packard führen zu der Erkenntnis, dass alles, was wir im ERP-System implementieren wollten, nicht ausreichend sein würde. Schließlich schrieben alle zusammen ein Anforderungspapier, in dem sie darlegten, was sie von einer zukünftigen SAP-Variantenkonfiguration 2.0 erwarteten – und wir mussten darauf eine Antwort liefern.

Wir wollten ein paar Jahre in Kalifornien arbeiten

Manchmal gibt es ganz einfache Erklärungen dafür, warum bestimmte Dinge passieren. Wir ermunterten die damals noch kleine CWG, ein Meeting mit unserem Entwicklungsvorstand im Rahmen der SAPPHIRE-Konferenz im Jahr 1995 in Phoenix/USA zu fordern. Dort machte die CWG den Vorschlag, dass als Zeichen des guten Willens von SAP ein Kern des KI-Konfigurationsteams nahe beim Kunden an der Umsetzung möglichst aller Anforderungen des Papiers arbeiten sollte. Der SAP-Entwicklungsstandort in Kalifornien schien dafür am geeignetsten zu sein.

Fast wäre daraus nichts geworden, denn Hasso Plattner war in dem Meeting wieder einmal gedanklich 10–15 Jahre weiter und der Ansicht, dass lokal installierte Software schon bald unnötig sein würde, da alle Funktionen online über das Internet abrufbar seien. Schließlich stimmte er der Investition jedoch zu.

Wir integrierten zunächst wie geplant die vorhandenen Ansätze (mehrstufige hierarchische Konfiguration, Constraints, Klassenknoten, Wissensbasen etc.) in das ERP-System und entwickelten ab Mitte 1996 in Kalifornien die IDoc-Schnittstellen für die Behandlung von Variantenkonfigurationen in der R/3-Auftragsabwicklung, unterstützt durch Tester von HP vor Ort. Die räumliche Nähe zu den Gründungsmitgliedern der CWG machte sich sofort bezahlt. Wir unterstützten die Entwicklung der ersten internetbasierten Variantenkonfiguration und erlebten, wie Bill Gates als Gast der Keynote auf der SAPPHIRE eine Jukebox mit Optionen konfigurierte, bestellte und Sekunden später auf der Bühne ausgeliefert bekam.

Wir konnten uns nun endlich auf neue Konzepte und deren Umsetzung als Antwort auf die Anforderungen der CWG konzentrieren. Diese wurden später als SCE Advanced Mode bezeichnet und bilden bis heute die Grundlage des SSC-Konfigurators.

Es handelte sich um frei definierbare Relationen, über die man in Constraints »reden« konnte. Diese Relationen wurden als sogenannte abstrakte Datentypen (ADTs) deklariert und konnten sogar im System als Stammdaten verwendet werden. Auch wenn sie noch viele Jahre nicht zielführend im ERP-System verarbeitbar waren; es bedurfte intensiver Projektarbeit, etwas Sinnvolles damit anzufangen. Schließlich ist es immer schwer, Flexibilität in einen Standard zu pressen.

Lange Zeit wurden die damals erarbeiteten Ansätze unterschiedlich genutzt, verändert, mit anderer Technologie neu implementiert sowie zuweilen erweitert – und manchmal wurden bestimmte Konzepte erst einmal verworfen.

Es ergaben sich immer wieder neue Herausforderungen, um die Konfiguratoren in die neuesten SAP-ERP-Produkte einzubauen oder die Inhalte der Wissensbasen für ganz andere Systemlandschaften in geeigneter Weise und möglichst performant bereitzustellen.

Der Stab wurde dabei schon viele Male an die nächste Expertengeneration weitergereicht. Aus der kleinen CWG, die 1993/1994 von fünf oder sechs Firmen getragen wurde, ist inzwischen eine Organisation mit über 6.500 Mitgliedern geworden.

Im vorliegenden Buch wird AVC als jüngste, in SAP S/4HANA integrierte Konfiguratorlösung detailliert vorgestellt.

Wir blicken mit Stolz und auch mit etwas Wehmut auf die Anfangsjahre der Entwicklung der SAP-Variantenkonfiguration zurück. Besonders auch, da diese dokumentierten wissenschaftlichen Grundlagen es einer Vielzahl von Firmen ermöglicht, ihre Herausforderungen zu bewältigen.

Viel Spaß beim Lesen und Lernen,

Dr. Albert Haag, Dr. Andreas Krämer und Dr. Bernhard Neumann

Einleitung

Die meisten von Ihnen kennen zum Thema *SAP-Variantenkonfiguration* wahrscheinlich die verschiedenen Auflagen unseres Buches »Variantenkonfiguration mit SAP« (Rheinwerk Verlag). Dieses Buch wurde 2009 zum ersten Mal herausgegeben und in zwei weiteren Auflagen 2011 und 2015 aktualisiert und erweitert neu aufgelegt.

Mit der letzten der genannten Auflagen schließen wir das Kapitel der SAP-Variantenkonfiguration mit LO-VC. Wir wagen einen Neuanfang und widmen uns dem Nachfolger von LO-VC: *SAP Advanced Variant Configuration (AVC)* in SAP S/4HANA. Bereits im Jahr 2016 hat SAP damit begonnen, eine neue Variantenkonfiguration als Core-Komponente für SAP S/4HANA zu entwickeln. Dieses neue Buch widmet sich nun erstmals vollumfänglich der neuen Lösung. Dabei haben wir es uns zum Ziel gesetzt, das Produkt in der gleichen Detailtiefe zu beschreiben, wie wir es für LO-VC getan haben.

Auch wenn LO-VC auch in SAP S/4HANA on-premise und in der Private Cloud verfügbar ist, stellt dieser alte Konfigurator eine Brückentechnologie dar. Zum einen befindet sich LO-VC im Wartungsmodus, und zum anderen lassen sich viele neue Prozesse und Komponenten in SAP S/4HANA nur mit AVC nutzen.

Getreu dem Motto »Model Once – Configure Anywhere« gehen wir neben der Betrachtung der Core-Komponente AVC und deren Stammdaten und Prozesse in SAP S/4HANA Cloud, der On-Premise-Variante und der Private Cloud, natürlich auch auf die Gesamtarchitektur ein. Im Mittelpunkt steht dabei die SAP Business Technology Platform (SAP BTP). Denn, ähnlich wie bei LO-VC, gibt es Erweiterungsszenarien, die es erlauben, AVC aus dem »Backend« auch in Applikationen wie SAP Commerce Cloud, SAP CPQ oder Nicht-SAP-Anwendungen weiterzuverwenden.

Das Buch richtet sich maßgeblich an zwei Zielgruppen: zum einen an SAP-Neukunden, die SAP S/4HANA entweder bereits im Einsatz haben oder gerade einführen und die Notwendigkeit einer integrierten Variantenkonfiguration sehen. Zum anderen adressiert es insbesondere die große Anzahl von SAP-Bestandskunden, die bereits auf eine langjährige Historie mit LO-VC in SAP ERP zurückschauen und bereits eine tiefgehende Expertise aufweisen. Der Beweggrund der letztgenannten Personengruppe ist ein anstehender oder bereits erfolgter Wechsel auf SAP S/4HANA. Ob nun Greenfield, Brownfield oder etwas dazwischen – die Beschäftigung mit AVC ist alleine aus strategischer Sicht unumgänglich.

Wie wichtig ein bestimmtes Kapitel für Sie persönlich ist, hängt von Ihrem Background bzw. Ihrem Interesse ab. Wir stellen Ihnen hier die einzelnen Kapitel kurz vor.

Wir beginnen in **Teil I** mit den Grundlagen der Variantenkonfiguration mit AVC.

Kapitel 1, »Produktkonfiguration mit SAP«, adressiert maßgeblich den Wechsel der Softwaregenerationen. Dies gilt sowohl bezogen auf den Umstieg von SAP ERP auf SAP S/4HANA als auch auf den Wechsel von LO-VC auf AVC. Wir führen in die unterschiedlichen Softwarelösungen ein und erläutern die verschiedenen durch AVC unterstützten Geschäftsprozesse. Ziel ist es, Ihnen einen Überblick über AVC in der Cloud zu geben sowie die Veränderungen und Innovationen der Variantenkonfiguration aufzuzeigen. Darüber hinaus wird auch auf die Transformation von LO-VC in AVC und die möglichen Anwendungsszenarien in der Cloud und im On-Premise-System eingegangen.

Kapitel 2, »Der Weg von LO-VC nach AVC in SAP S/4HANA«, beschreibt die Vorgehensweise und die bereitgestellten Tools, die Ihnen dabei helfen, ihre Produktmodelle und Geschäftsprozesse von einem SAP-ERP-System auf SAP S/4HANA umzustellen. Dabei wird auch auf das Thema Greenfield vs. Brownfield eingegangen.

Teil II ist den Grundlagen der Modellierung gewidmet.

In **Kapitel 3**, »Ein erstes konfigurierbares Material anlegen«, beginnen wir mit der Modellierung. Dieses Kapitel ist ein erster Einstieg, um in die Welt der Modelle der Variantenkonfiguration einzutauchen. Anhand eines Beispiels, bei dem ein vorhandenes Produkt nachträglich konfigurierbar gemacht wird, werden Sie mit der Begriffswelt der Modellierung vertraut gemacht.

Kapitel 4, »Stammdaten im Rahmen der Modellierung«, befasst sich mit den notwendigen Stammdaten der Modellierung für AVC. Vorab werden die Produktmodellierungsumgebung PMEVC und die Simulationsumgebung vorgestellt. Anschließend wird die Klassifizierung ausführlich behandelt. Es folgen Materialstamm, Stückliste und Arbeitsplan des konfigurierbaren Produkts. Das Konfigurationsprofil mit seinen Einstellungen zu möglichen Konfigurationsszenarien stellt einen weiteren Schwerpunkt dar. Die Merkmalsgruppen als Oberflächendesign runden dieses Kapitel ab.

Kapitel 5, »Die Intelligenz: Beziehungswissen«, behandelt das »Herzstück« des Variantenmodells. In diesem Kapitel fokussierten wir uns auf die Erstellung der Regeln der SAP-Variantenkonfiguration, das sogenannte Beziehungswissen. Das Beziehungswissen ist Dreh- und Angelpunkt bei der Nutzung der Variantenkonfiguration, wird doch hiermit die Intelligenz ins System gebracht. Es wird sowohl die High-Level- als auch die Low-Level-Konfiguration beschrieben. Dabei werden auch die neuen Möglichkeiten unter AVC vorgestellt.

Kapitel 6, »Preisfindung«, stellt kurz die Funktionen zur Preisfindung in Vertrieb und Einkauf dar. Dabei wird die Frage beantwortet, wie die Konfiguration den Preis beeinflusst. Auch neue Möglichkeiten, die Preisfindung über Constraints abzubilden, werden angesprochen.

Kapitel 7, »Materialvarianten«, beschäftigt sich mit konfigurierten Materialien. Mit den Materialvarianten stehen Ihnen in der Variantenkonfiguration alle Möglichkei-

ten der Planung, Beschaffung, Lagerverwaltung und weiterer logistischer Anforderungen, die Sie gegebenenfalls außerhalb der Variantenkonfiguration nutzen, zur Verfügung. Die folgenden Fragen werden beantwortet: Was ist eine Materialvariante? Welche Funktionen gibt es für Materialstamm, Stückliste, Arbeitsplan und Preisfindung für Materialvarianten? Aber auch die neue flexiblere Variantenfindung und die Möglichkeit, direkt aus dem Kundenauftrag heraus neue Materialvarianten per Klick anlegen zu lassen, kommen zur Sprache.

Teil III beschäftigt sich mit den Optionen, die Ihnen für eine gelungene Integration zur Verfügung stehen.

Das **Kapitel 8**, »Die Integration der Variantenkonfiguration in Plattformen mittels Wissensbasen und Laufzeitversionen«, widmet sich dem Credo »Model Once – Configure Anywhere« und erläutert die dafür notwendigen Prozesse und Stammdaten. So werden hier insbesondere die Microservices auf der SAP BTP und die dafür notwendigen Stammdaten Wissensbasis (Knowledge Base) und Laufzeitversion (Runtime Version) thematisiert. Diese sind Bedingung zur Nutzung der Variantenkonfiguration in Applikationen wie SAP BTP, SAP Commerce Cloud und SAP CPQ.

In **Kapitel 9**, »Neue Integrationsaspekte zur Variantenkonfiguration«, gehen wir auf Themen wie die Product Replication Workbench, die Prozessintegration in Siemens Teamcenter, das Anforderungsmanagement und die Dokumentation in der Modellierung (SAP Enterprise Product Development) ein und erläutern, wie man Machine Learning im Kontext der Variantenkonfiguration einsetzen kann. Auch beleuchten wir die Integration von AVC in das Workflow Management und erklären das 2-Tier-Szenario, das eine systemübergreifende Beschaffung konfigurierter Produkte ermöglicht.

In **Kapitel 10**, »Lösungen von SAP-Partnern«, stellen wir Ihnen Entwicklungen von SAP-Partnern vor, die für die Variantenkonfiguration einen Mehrwert bieten. SAP eröffnet mit der Partnerstrategie, insbesondere mit der SAP BTP, unzähligen Unternehmen die Möglichkeit, z. B. branchenspezifische Add-ons bereitzustellen oder den Standard zu erweitern. In diesem Kapitel finden Sie Erweiterungen zum Thema 3D-Visualisierung, einer neuen Pflegeumgebung für Variantentabelleninhalte auf der SAP BTP, Beispiele zur Nutzung von Machine Learning im Umfeld der Variantenkonfiguration, eine KI-Unterstützung in der Auftragsbearbeitung und eine hybride Systemkonfiguration, die auf APIs basiert.

In **Teil IV** stellt Ihnen Best Practices aus der Industrie sowie Möglichkeiten des Austauschs mit der VC-Community vor.

Kapitel 11, »Erfahrungsbericht: Transformation von LO-VC nach AVC«, zeigt beispielhaft anhand des Unternehmens Trilux, wie eine Transformation nach AVC erfolgen kann.

In **Kapitel 12**, »Communitys zur SAP-Variantenkonfiguration«, gehen wir auf die verschiedenen Kundengruppen ein, die sich zum Teil bereits seit Jahrzehnten mit dem

Thema SAP-Variantenkonfiguration auseinandersetzen. Dazu gehören zum einen die Configuration Workgroup (CWG), die mit Tausenden von Teilnehmern größte internationale Nutzervereinigung. Außerdem stellen wir den DSAG-Arbeitskreis »Variantenkonfiguration« vor, der kurz nach dessen Gründung im Jahre 2018 bereits über 600 Kunden und Partner im deutschsprachigen Raum für sich gewinnen konnte. Zuletzt gehen wir auf den SAP AVC Customer Co-Innovation Council ein, eine ausgewählte Gruppe von Kunden und Partnern, mit denen das SAP-Produktmanagement und die Produktentwicklung sehr eng zusammenarbeiten.

Wie Sie anhand der thematischen Bandbreite erkennen können, haben wir versucht, möglichst viele Aspekte von AVC zu beleuchten. Nichtsdestotrotz können und wollen wir keinen Anspruch auf Vollständigkeit erheben, da das Produkt und die dazugehörigen Prozesse ständig weiterentwickelt werden. Das heißt, es kommen kontinuierlich neue Funktionen und Applikationen hinzu!

Die Variantenkonfiguration lässt sich als »Rückenmark« der IT-Landschaft eines Unternehmens beschreiben, das sich mit kundeneinzelgefertigten Produkten beschäftigt. Alle Prozessbausteine sind bei diesen Unternehmen etwas anders als bei Standardprodukten und meistens auch etwas komplexer. Dennoch wähle ich lieber ein einziges intelligentes Produkt mit dynamischer Stückliste, Arbeitsplan und Preisfindung, als mich mit Milliarden von ausgeprägten Materialstämmen mit festen, aber einfachen Stammdaten zu befassen. Ausnahmen bestätigen die Regel, wie Sie in einigen Abschnitten zu den Materialvarianten im Buch lesen können.

Wir freuen uns auf eine Vielzahl an Rückmeldungen und konstruktives Feedback zu Themen, die Sie in diesem Buch noch vermissen. Senden Sie uns gerne weitere Tipps und Lösungsvorschläge, die dann eventuell in Folgeauflagen veröffentlicht werden. Wenden Sie sich hierzu bitte an eva.tripp@rheinwerk-verlag.de.

Zuletzt möchten wir Sie einladen, der CWG (www.configuration-workgroup.com) einen Besuch abzustatten. Dies ist die ideale Adresse für die Sammlung und Veröffentlichung von Tipps und Tricks (z. B. auf Konferenzen). Wenn Sie dieses Buch in Ihren Händen halten, passen Sie bestens zur CWG. Kommen Sie doch mal vorbei!

Dieses Buch ist allen Anwenderinnen und Anwendern der SAP-Variantenkonfiguration gewidmet. Insbesondere danken wir den Personen aus Modellierung und Beratung sowie den SAP-Partnerunternehmen, die sich in den letzten 20 Jahren mit diesem komplexen Thema auseinandergesetzt haben.

Aber auch den Mitgliedern der Anwendergruppen, insbesondere der Configuration Workgroup (CWG) und des DSAG-Arbeitskreises gebührt Dank. Ohne den regen Austausch wäre diese Publikation nicht möglich gewesen. Und last but not least: Ein herzliches Dankeschön an die Entwicklerinnen und Entwickler in Walldorf, in Budapest, in Indien und in den USA.

Dr. Uwe Blumöhr, Andreas Kölbl, Michael Neuhaus und Marin Ukalovic

Auf einen Blick

TEIL I Grundlagen der Variantenkonfiguration

- 1 Produktkonfiguration mit SAP 27
- 2 Der Weg von LO-VC nach AVC in SAP S/4HANA 57

TEIL II Grundlagen der Modellierung

- 3 Ein erstes konfigurierbares Material anlegen 85
- 4 Stammdaten im Rahmen der Modellierung 109
- 5 Die Intelligenz: Beziehungswissen 221
- 6 Preisfindung 309
- 7 Materialvarianten 339

TEIL III Integration

- 8 Die Integration der Variantenkonfiguration in Plattformen
mittels Wissensbasen und Laufzeitversionen 367
- 9 Neue Integrationsaspekte zur Variantenkonfiguration 419
- 10 Lösungen von SAP-Partnern 497

TEIL IV Best Practices und Möglichkeiten zum Erfahrungsaustausch

- 11 Erfahrungsbericht: Transformation von LO-VC nach AVC 561
- 12 Communitys zur SAP-Variantenkonfiguration 577

Inhalt

Vorwort	17
Einleitung	21

TEIL I Grundlagen der Variantenkonfiguration

1 Produktkonfiguration mit SAP 27

1.1 Wechsel der Generationen: von SAP ERP zu SAP S/4HANA	28
1.1.1 SAP ERP und SAP S/4HANA im Vergleich	28
1.1.2 Die Benutzeroberfläche SAP Fiori	30
1.1.3 SAP HANA oder SAP S/4HANA?	31
1.1.4 Formen des Betriebs	31
1.1.5 Der digitale Kern – the Digital Core	32
1.1.6 Vereinfachung (Simplification)	32
1.2 Wechsel der Generationen: von LO-VC zu AVC	32
1.2.1 SAP S/4HANA: Die Simplification List in LO-VC	33
1.2.2 AVC in SAP S/4HANA Cloud	35
1.3 Welche SAP-Konfiguratoren gibt es?	40
1.3.1 Internet Pricing and Configurator (IPC)	40
1.3.2 Solution Sales Configurator (SSC)	41
1.3.3 SAP Configure, Price, and Quote (SAP CPQ)	41
1.3.4 SAP Advanced Variant Configuration and Pricing (Microservice)	41
1.4 Produktkonfiguration mit AVC im End-to-End-Prozess	42
1.5 Beherrschung von Produktvielfalt	48
1.6 Fertigungsszenarien für konfigurierbare Produkte	52
1.7 Fazit	55

2 Der Weg von LO-VC nach AVC in SAP S/4HANA 57

2.1 AVC und LO-VC – ein Entweder-oder?	57
2.2 Wege der Transformation	61
2.3 Grundlagen der Transformation	62

2.4	Transition Workspace	65
2.5	Transition Workbench	66
2.5.1	Aufbau der Transition Workbench	67
2.5.2	Analyse des LO-VC-Modells	68
2.5.3	Erstellung des AVC-Modells	70
2.5.4	Transformation eines Constraints – ein Beispiel	71
2.6	Ergebnisvergleich	72
2.7	Transfer der Materialvarianten	75
2.8	Ein beispielhaftes Konzept zum Wechsel von VC auf AVC (Greenfield vs. Brownfield)	77
2.8.1	Warum auf AVC umstellen?	78
2.8.2	Brownfield vs. Greenfield	78
2.8.3	Schritt 1: Scoping	79
2.8.4	Schritt 2: Planung	80
2.8.5	Schritt 3: Umsetzung	80
2.8.6	Schritt 4: Testen	81
2.8.7	Schritt 5: Go-live	81
2.8.8	Schritt 6: Hypercare	82
2.9	Fazit	82

TEIL II Grundlagen der Modellierung

3	Ein erstes konfigurierbares Material anlegen	85
3.1	Grundlagen	86
3.2	Ein erstes Beispiel	89
3.3	Der konfigurierbare Materialstamm	90
3.4	Variantenklasse, Merkmale und Konfigurationsprofil	92
3.5	Die Bewertungsoberfläche der erweiterten Variantenkonfiguration	97
3.6	Beziehungswissen für Planung und Fertigung	101
3.7	Konfiguration und Preise	104
3.8	Fazit	107

4	 Stammdaten im Rahmen der Modellierung	109
4.1	 Die Modellierungsumgebung der Variantenkonfiguration	109
4.2	 Die Simulationsumgebung von AVC	120
4.3	 Werkzeuge aus dem Klassensystem	131
4.3.1	Merkmalspflege	131
4.3.2	Klassenpflege	142
4.3.3	Klassifizierung	147
4.3.4	Such- und Auswertungsmöglichkeiten im Klassensystem	153
4.4	 Materialstamm, Stückliste und Arbeitsplan des konfigurierbaren Materials	159
4.4.1	Materialstamm des konfigurierbaren Materials	159
4.4.2	Maximalstückliste des konfigurierbaren Materials	169
4.4.3	Maximalarbeitsplan für das konfigurierbare Material	172
4.4.4	Fertigungsversionen	175
4.5	 Konfigurationsprofil und Konfigurationsszenarien	178
4.5.1	Konfigurationsprofil im Überblick	178
4.5.2	Konfigurationsprofil im Detail	182
4.5.3	Konfigurationsszenarien im Überblick	187
4.5.4	Szenario »Plan-/Fertigungsauftrag ohne Stücklistenauflösung«	188
4.5.5	Szenario »Plan-/Fertigungsauftrag mit Stücklistenauflösung«	191
4.5.6	Szenario »Kundenauftrag (SET)«	194
4.5.7	Szenario »Auftragsstückliste« – der ETO-Prozess	199
4.5.8	Szenario »Auftragsstückliste« – Auftragsstückliste und Auftragsarbeitsplan	209
4.6	 Die Merkmalsgruppen als Oberflächendesign	216
4.7	 Fazit	220
5	 Die Intelligenz: Beziehungswissen	221
5.1	 Grundlagen	221
5.1.1	Arten und Zuordnung von Beziehungswissen	222
5.1.2	Prozeduraler und deklarativer Character von Beziehungswissen	227
5.1.3	Lokales und globales Beziehungswissen	228
5.1.4	Status von Beziehungswissen	234
5.1.5	Verarbeitungsmodus von Beziehungswissen	235

5.1.6	Beziehungswissen in der Klassifizierung und in der Variantenkonfiguration	236
5.1.7	Ausführungsreihenfolge von Beziehungswissen	237
5.1.8	Grundregeln der Syntax	240
5.1.9	Syntaxelemente	244
5.2	Variantentabellen	248
5.2.1	Eine Tabellenstruktur anlegen	248
5.2.2	Kopplung einer Variantentabelle an eine Datenbanktabelle oder an ein Custom Business Object (CBO)	250
5.2.3	Inhalt einer Variantentabelle	253
5.2.4	Zugriff auf eine Variantentabelle	255
5.2.5	Variantentabellen zur Werteinschränkung durch Constraints	256
5.2.6	Variantentabellen zur Wertherleitung durch Prozeduren	258
5.2.7	Variantentabellen in Bedingungen	259
5.3	Beziehungswissen für die interaktive Konfiguration (High-Level-Konfiguration)	262
5.3.1	Nutzen Sie Constraints	262
5.3.2	Exklusive Syntax für Constraints im Detail	264
5.3.3	Constraints im Trace	273
5.3.4	Prozeduren	277
5.3.5	Exklusive Syntax für Prozeduren im Detail	279
5.3.6	Prozeduren im Trace	282
5.3.7	Facettenänderungen mit Vor- und Auswahlbedingungen	283
5.3.8	AVC: Business-Add-Ins	287
5.3.9	Grundsätze der AVC-Modellierung für eine gute Performance	292
5.3.10	Constraints und die Mehrfachverwendung von Klassen	295
5.4	Beziehungswissen für Stückliste und Arbeitsplan (Low-Level-Konfiguration)	296
5.4.1	Auswahlbedingungen in Stückliste und Arbeitsplan	297
5.4.2	Klassenknoten in Stücklisten	299
5.4.3	Prozeduren in Stücklisten und Arbeitsplan	304
5.5	Fazit	307

6	Preisfindung	309
<hr/>		
6.1	Preisfindung im Vertrieb	310
6.1.1	Schritt 1: Objektmerkmal mit Bezug zur Struktur SCDOM-VKOND ...	310
6.1.2	Schritt 2: Konditionsart ermitteln und Variantenkondition anlegen	312
6.1.3	Schritt 3: Kalkulationsschema kontrollieren	314
6.1.4	Schritt 4: Variantenkonditionen zuordnen	315
6.1.5	Überprüfen des Ergebnisses unserer Modellierungsschritte	318
6.2	Preisfindung auf Basis der Kundenauftragskalkulation	321
6.3	Preisfindung im Einkauf	322
6.3.1	Schritt 1: Kalkulationsschema kontrollieren	328
6.3.2	Schritt 2: Konditionsart ermitteln und Variantenkonditionen anlegen	328
6.3.3	Schritt 3: Objektmerkmal zuordnen	331
6.3.4	Schritt 4: Variantenkonditionen zuordnen	332
6.4	Der Microservice zur Preisfindung auf der SAP BTP	333
6.4.1	Funktionsweise des Preisfindungsservice	333
6.4.2	Administration	336
6.4.3	Erweiterbarkeit	337
6.5	Fazit	338
7	Materialvarianten	339
<hr/>		
7.1	Materialstamm	340
7.2	Stückliste	346
7.3	Arbeitsplan	350
7.4	Preisfindung	355
7.5	Variantenfindung	355
7.6	Fazit	363

TEIL III Integration

8	Die Integration der Variantenkonfiguration in Plattformen mittels Wissensbasen und Laufzeitversionen	367
8.1	Die Architektur	367
8.2	Stammdaten der Konfigurationsintegration: Wissensbasen und Laufzeitversionen	370
8.2.1	Replikation der Stammdaten aus den SAP-ERP- und SAP-S/4HANA-Systemen	370
8.2.2	Die Delta-Liste in Bezug auf die Nutzung der Microservices und LO-VC	372
8.2.3	Unterschiede zwischen Konfigurationsmodellen und Wissensbasen in der Delta-Liste	373
8.2.4	Auswahl der Wissensbasis zur Laufzeit	374
8.3	Tipps zur Generierung von Wissensbasis-Laufzeitversionen	374
8.3.1	Objekte der Wissensbasis	375
8.3.2	Generierung neuer Laufzeitversionen	376
8.3.3	Regenerierung existierender Laufzeitversionen	377
8.3.4	Konfiguration mit Laufzeitversionen während Änderungen am Produktmodell	378
8.3.5	Die Tauglichkeit eines Produktmodells zur Erzeugung von Wissensbasen und Laufzeitversionen	381
8.3.6	Die Kompatibilität des Produktmodells	382
8.3.7	Änderungen für die Anwendung von Konfigurationsmodellen im Vertrieb	382
8.3.8	Tipps für die Erstellung von Laufzeitversionen	383
8.3.9	Empfehlungen für die Namenskonvention	384
8.4	Anleitung zur (Re)generierung von Laufzeitversionen	396
8.4.1	Wann müssen neue Laufzeitversionen erstellt werden?	398
8.4.2	Wann müssen bestehende Laufzeitversionen neu generiert werden?	398
8.4.3	Laufzeitversionen in Verbindung mit Änderungsnummern	399
8.4.4	Wie oft sollte man Laufzeitversionen (re)generieren?	400
8.4.5	Neuerungen in Bezug auf die Verwendung von AVC	402
8.4.6	Unnötige Laufzeitversionen filtern	404
8.5	Nutzung des Konfigurationsmicroservice auf der SAP BTP	405
8.5.1	Aufruf des Konfigurationsservice	405

8.5.2	Administration	406
8.5.3	Erweiterbarkeit	407
8.6	Variantenkonfiguration in SAP Commerce Cloud	407
8.7	Integrierte Variantenkonfiguration in SAP CPQ	411
8.7.1	SAP CPQ als führendes System	411
8.7.2	SAP CPQ als Add-on zur integrierten SAP-ERP- oder SAP-S/4HANA-Variantenkonfiguration	412
8.8	Fazit	418

9 Neue Integrationsaspekte zur Variantenkonfiguration 419

9.1	Stammdatenverteilung mit Product Data Replication	420
9.1.1	Herausforderung und Möglichkeiten	422
9.1.2	Setup und Customizing der PDR	424
9.1.3	Weitere Systemvorbereitungen für die PDR	435
9.1.4	Replication Workbench – Objekte und Begriffe	437
9.1.5	Konfigurationsdefinition und -mappe anlegen	439
9.1.6	Baseline erstellen und auflösen	441
9.1.7	Über den Verteilungsauftrag zu Verteilungseinheit und Verteilungspaket	444
9.1.8	Versand des UPS-Pakets	445
9.1.9	Einbuchung des Pakets	447
9.1.10	Korrekturpakete	448
9.1.11	Replikation von Konditionssätzen	452
9.1.12	PDR-Add-on für Arbeitspläne	452
9.1.13	Troubleshooting, Tipps und Tricks	454
9.2	Die SAP-Variantenkonfiguration im Zusammenspiel mit Siemens Teamcenter	458
9.2.1	Die strategische Partnerschaft zwischen SAP und Siemens	459
9.2.2	Die nächste Generation der Siemens-Teamcenter-Integration	461
9.2.3	Aktueller Status und Ausblick zur durchgängigen Variantenkonfiguration	462
9.2.4	Praxisbeispiel: Durchgängige Variantenkonfiguration eines konfigurierbaren Ventils	463
9.3	Anforderungsmanagement und Dokumentation in der Modellierung – SAP Enterprise Product Development	467

9.4	Maschinelles Lernen und künstliche Intelligenz für konfigurierbare Produkte	472
9.4.1	Datenerfassung und -aufbereitung	479
9.4.2	Erstellung von Modellen/Szenarien und Training	480
9.4.3	Nutzung von Machine-Learning-Modellen	480
9.4.4	Maschinelles Lernen in der Industry Cloud	481
9.5	Integration von AVC und SAP Workflow Management	484
9.6	2-Tier-Szenario – systemübergreifende Beschaffung konfigurierter Produkte	491
9.7	Fazit	494

10 Lösungen von SAP-Partnern 497

10.1	3D-Konfiguration	497
10.1.1	Begriffsdefinitionen	498
10.1.2	Funktionen einer in das SAP-System integrierte 3D-Konfiguration ...	500
10.1.3	3D-Konfiguration und Augmented und Virtual Reality	502
10.1.4	Praxisbeispiel: TRILUX GmbH & Co. KG	503
10.1.5	3D-Engine	508
10.2	KI-optimierte Bearbeitung von Kundenanfragen	511
10.2.1	Grundlagen und Rahmenbedingungen	512
10.2.2	Vorüberlegungen	516
10.2.3	Beispiele	519
10.3	Variantentabelleninhalte auf der SAP BTP	524
10.3.1	Systemarchitektur	525
10.3.2	networker, solutions SIMPLE VARIANT TABLE MAINTENANCE	526
10.4	Variantenkonfiguration mit Machine Learning	534
10.4.1	Machine Learning und komplexere Formeln am Beispiel des Brotbackens	536
10.4.2	Der Einsatz von Machine Learning am realen Beispiel einer Verpackungsmaschine »Folder Gluer«	543
10.5	Hybride Systemkonfiguration, durchgängige Automatisierung und API-basierte Integration mit SAP-Lösungen	545
10.5.1	Integrierte Variantenkonfiguration mit SAP-Lösungen und externen Softwaresystemen	546
10.5.2	Konsistente User Experience mit heterogenen Configuration Engines im Parallelbetrieb	548

10.5.3	Mehrstufige hybride Systemkonfiguration und Orchestrierung heterogener Wissensbasen	549
10.5.4	Serviceorientierte Architekturen, SAP-kompatible APIs und komplexe mehrstufige Datenstrukturen	551
10.5.5	Systemneutrale Configuration-IDs und Configuration Lifecycle Management	553
10.5.6	End-2-End-Digitalisierung und Automatisierung mit Workflow Management und Integration	554
10.6	Fazit	556

TEIL IV Best Practices und Möglichkeiten zum Erfahrungsaustausch

11 Erfahrungsbericht: Transformation von LO-VC nach AVC 561

11.1	AVC-Vorstudie	561
11.1.1	Datenmodellanalyse	562
11.1.2	Analyse der Pfunctions	563
11.1.3	Identifizierung von Risiken und Herausforderungen	563
11.1.4	Geschätzte Aufwände	564
11.2	Best-Practice-Transformation – ein Praxisbeispiel	568
11.2.1	Scoping	569
11.2.2	Planung	570
11.2.3	Umsetzung	572
11.2.4	Testen	575
11.2.5	Go-live	575
11.2.6	Hypercare	576
11.3	Fazit	576

12 Communitys zur SAP-Variantenkonfiguration 577

12.1	Die Configuration Workgroup (CWG)	578
12.1.1	Zielsetzung und Aufgaben	579
12.1.2	Entstehungsgeschichte	581
12.1.3	Organisatorischer Aufbau	583

12.1.4	CWG-Konferenzen	584
12.1.5	CWG-Portal	586
12.1.6	CWG-Sandbox-Systeme	588
12.2	Die DSAG-Arbeitsgruppe »Variantenkonfiguration«	588
12.2.1	Interview mit den Sprechern der DSAG-Arbeitsgruppe »Variantenkonfiguration«	590
12.3	SAP AVC Customer Co-Innovation Council	592
12.4	Fazit	593
	Das Autorenteam	595
	Index	607

Die neue Variantenkonfiguration in SAP S/4HANA

Produkte modellieren

Mit diesem Buch legen Sie ein erstes konfigurierbares Material an und lernen nach und nach alle Schritte der Produktmodellierung kennen. Sie erfahren, wie Sie mit AVC in SAP S/4HANA u. a. Ihre Prozesse in Entwicklung, Fertigung und Vertrieb unterstützen.

Integration mit anderen SAP-Werkzeugen

Die Autoren stellen Ihnen das Zusammenspiel von AVC mit anderen Werkzeugen und Lösungen vor: Sie erfahren, wie Sie z. B. SAP Commerce Cloud, SAP CPQ, SAP BTP und Machine Learning in der Variantenkonfiguration nutzen.

Hilfestellung für die Transformation

Schlagen Sie ein ganz neues Kapitel der Variantenkonfiguration auf: Sie lernen die Unterschiede zwischen AVC und LO-VC kennen und erhalten wertvolles Wissen für den Umstieg. So können Sie Ihr eigenes AVC-Projekt schneller meistern.

Auf einen Blick

- Transformation von LO-VC zu AVC
- Produktmodellierung (Transaktion PMEVC)
- Beziehungswissen
- Constraints
- Konfigurierbare Stückliste
- Integration
- Preisfindung
- Materialvarianten
- Lösungen von SAP-Partnern
- Best Practices und Ergebnisberichte

»Sehr gut strukturiert und verständlich geschrieben: für Neulinge und Experten.«

Leser-Feedback zur Voraufgabe



Die Autoren

Dr. Uwe Blumöhr ist als Schulungsberater, Andreas Kölbl als Area Product Owner für AVC und Marin Ukalovic als Chief Product Owner für Digital Configuration Lifecycle bei SAP tätig. Michael Neuhaus ist Geschäftsführer der networker, solutions GmbH und der TRILUX Digital Solutions GmbH.

