

# Advanced Variant Configuration in SAP S/4HANA

Das umfassende Handbuch

» Hier geht's  
direkt  
zum Buch

# DIE LESEPROBE

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

**Tabelle 5.1** Anwendungsgebiete von Beziehungswissen

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

**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
<b>Vorbedingungen</b>	
Merkmal und Wert	–
<b>Auswahlbedingungen</b>	
Merkmal	<ul style="list-style-type: none"> <li>■ Position der Stückliste</li> <li>■ Vorgang, Folge, Fertigungshilfsmittel des Arbeitsplans</li> </ul>
<b>Prozeduren</b>	
Konfigurationsprofil	<ul style="list-style-type: none"> <li>■ Position der Stückliste</li> <li>■ Vorgang, Folge, Fertigungshilfsmittel des Arbeitsplans</li> </ul>
<b>Constraints</b>	
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\* Suchen + Software pflegen Softwareabhängigkeiten Beziehungswissen Löschen \*\*\*

<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	>

Sichern Entwurf verwerfen

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.

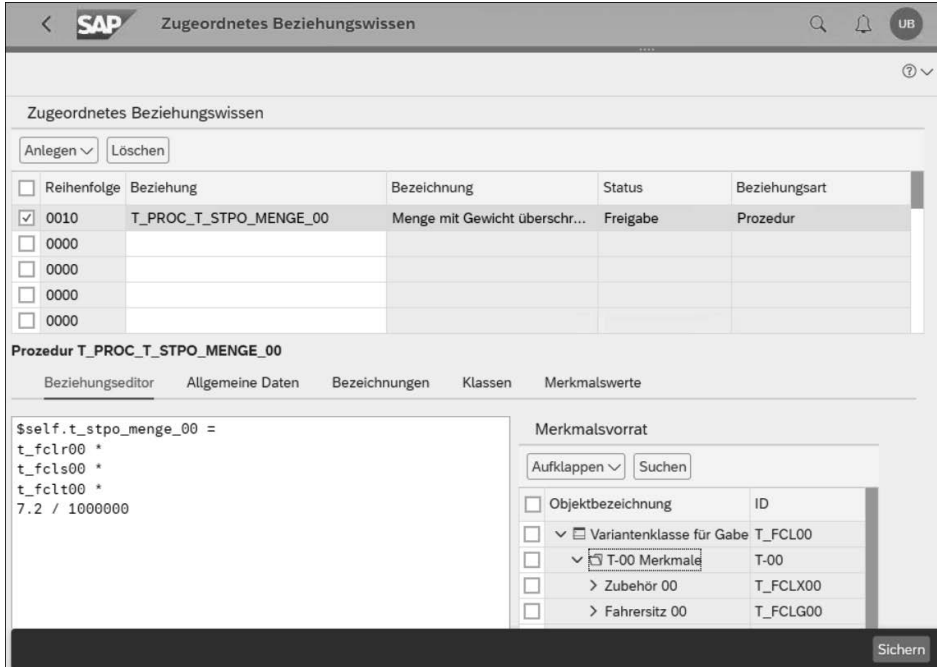


Abbildung 5.6 Die Beziehungswissenspflege in der SAP-Fiori-App »Stücklisten pflegen«

### 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.

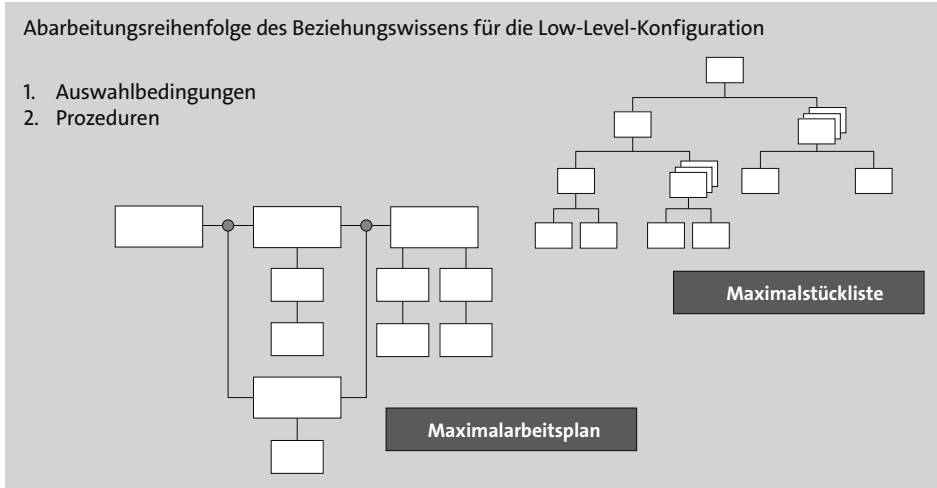
Aus diesen Gründen werden Prozeduren an Merkmalen und Merkmalswerten nicht empfohlen und teilweise sogar verboten. Zum Beispiel werden in die Laufzeitversion keine Prozeduren an Merkmalen und Merkmalswerten übernommen.

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).



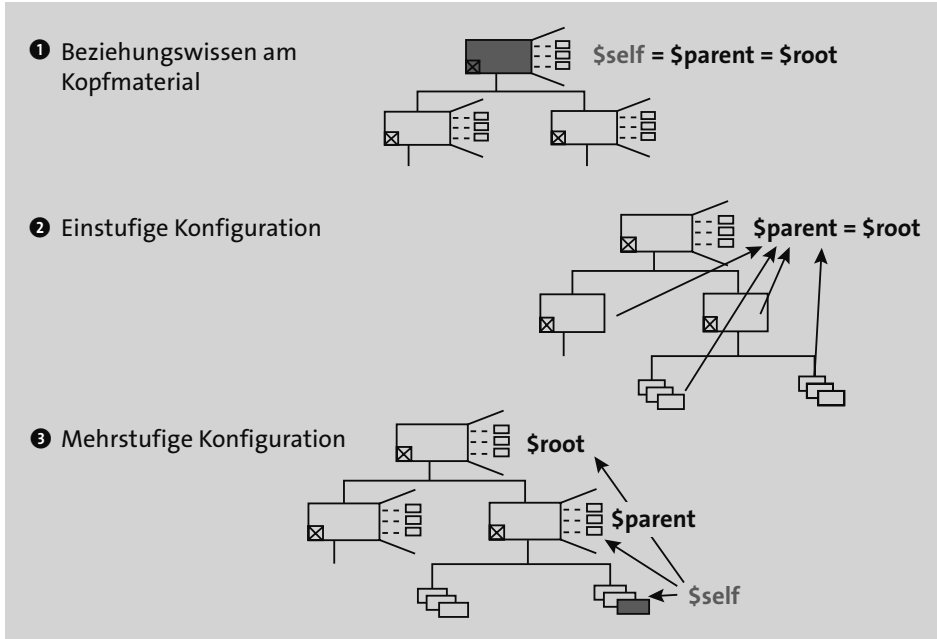


Abbildung 5.9  $\$self$ ,  $\$parent$  und  $\$root$  in speziellen Fällen im Beziehungswissen

### 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  $\geq 0$ ).
- `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).