

Spezifikation  
OCD  
OFML Commercial Data\*  
(OFML Part IV)

Version 4.3

Status: Release

Thomas Gerth, EasternGraphics GmbH (Editor)

2020-06-25



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Die Tabellen</b>	<b>5</b>
2.1	Übersicht . . . . .	5
2.2	Die Artikeltable . . . . .	6
2.3	Die Artikel-Identifikationstabelle . . . . .	7
2.4	Die Klassifikationstabelle . . . . .	8
2.5	Die Packaging-Tabelle . . . . .	9
2.6	Die Tabelle kompositer Artikel . . . . .	11
2.7	Die Stücklistentabelle . . . . .	12
2.8	Die Merkmalsklassentabelle . . . . .	13
2.9	Die Merkmalstabelle . . . . .	14
2.10	Die Merkmal-Identifikationstabelle . . . . .	18
2.11	Merkmalsgruppen . . . . .	19
2.12	Die Artikelstammtabelle . . . . .	20
2.13	Die Merkmalswerttabelle . . . . .	20
2.14	Die Merkmalswert-Identifikationstabelle . . . . .	22
2.15	Die Beziehungsobjekt-Tabelle . . . . .	23
2.16	Die Beziehungswissen-Tabelle . . . . .	25
2.17	Die Preistabelle . . . . .	26
2.18	Die Rundungsvorschrift-Tabelle . . . . .	27
2.19	Die Serientabelle . . . . .	28
2.20	Die Beschreibungstabellen . . . . .	29
2.21	Wertkombinationstabellen . . . . .	30
2.22	Die Identifikationstabelle . . . . .	31
2.23	Die Versionsinformationstabelle . . . . .	32
2.24	Die Nummernschema-Tabelle . . . . .	33
2.25	Besteuerungsschemata . . . . .	35
<b>3</b>	<b>Die Preisermittlung</b>	<b>37</b>
3.1	Überblick . . . . .	37
3.2	Relevante Tabelleneinträge . . . . .	37
3.3	Gültige Tabelleneinträge . . . . .	38
3.4	Preisfaktoren . . . . .	39
<b>4</b>	<b>Die Endartikelnummererzeugung</b>	<b>40</b>
4.1	Die vordefinierten Schemata . . . . .	40
4.2	Nutzerdefinierte Schemata . . . . .	41
4.3	Mehrwertige Merkmale . . . . .	41

<b>5</b>	<b>Merkmals-text-Steuerung</b>	<b>43</b>
<b>6</b>	<b>Die Ermittlung von Verpackungsdaten</b>	<b>45</b>
<b>A</b>	<b>Sprachdefinition OCD_1</b>	<b>46</b>
<b>B</b>	<b>Sprachdefinition OCD_2</b>	<b>48</b>
	B.1 Constraints . . . . .	48
	B.2 Tabellenaufruf . . . . .	51
	B.2.1 Tabellenaufruf in Vorbedingungen . . . . .	51
	B.2.2 Tabellenaufruf in Aktionen und Reaktionen . . . . .	51
	B.2.3 Tabellenaufruf in Constraints . . . . .	52
<b>C</b>	<b>Sprachdefinition OCD_3</b>	<b>53</b>
	C.1 mehrwertige Merkmale . . . . .	53
	C.2 mehrstufige Konfiguration . . . . .	53
<b>D</b>	<b>Sprachdefinition OCD_4</b>	<b>54</b>
<b>E</b>	<b>Sprachdefinition SAP_LOVC</b>	<b>58</b>
<b>F</b>	<b>Arithmetische Funktionen in Beziehungswissen</b>	<b>59</b>
<b>G</b>	<b>Reservierte Schlüsselwörter</b>	<b>60</b>
<b>H</b>	<b>Steuerarten und Steuerkategorien</b>	<b>62</b>
<b>I</b>	<b>Begriffe</b>	<b>64</b>
<b>J</b>	<b>Änderungshistorie</b>	<b>65</b>
	J.1 OCD 4.3 vs. OCD 4.2 . . . . .	65

# 1 Einleitung

OCD dient allgemein zur Anlage von Produktdaten, die in Geschäftsprozessen des Möbelhandels benötigt und ausgetauscht werden. Mit OCD sollen primär folgende Aufgaben abgedeckt und abgewickelt werden können:

- Konfiguration komplexer Artikel
- Preisermittlung
- Erstellung von Angebots- und Bestellformularen

OCD ist *kein* Format zur Anlage von Katalogdaten. Diese müssen anderweitig bereitgestellt werden. Die Verknüpfung zwischen Katalog- und Produktdaten erfolgt durch das jeweilige Softwaresystem anhand der Artikelnummern.

Das Datenmodell für OCD baut auf dem grundlegenden OFML-Produktdatenmodell auf (siehe Anhang A des OFML-Standards, Version 2.0.2).

Als physisches Austauschformat zwischen OFML-konformen Applikationen werden CSV-Tabellen (comma separated values) verwendet. Hierzu gelten folgende Bestimmungen:

- Jede der unten beschriebenen Tabellen ist in genau einer Datei enthalten. Der Dateiname wird durch den Präfix „ocd\_“, den spezifizierten Tabellennamen und den Suffix „.csv“ gebildet, wobei der Tabellename komplett klein geschrieben wird.
- Jede Zeile der Datei (abgeschlossen durch ein Zeichen für den Zeilenwechsel '\n') repräsentiert einen Datensatz. Leerzeilen (solche aus Null oder mehr Leerzeichen oder Tabulator) werden ignoriert. Als Zeichensatz wird ISO-8859-1 (Latin-1) verwendet.
- Die Felder eines Datensatzes werden durch Semikolon voneinander getrennt.
- Zeilen, die mit einem Doppelkreuz ('#') beginnen, werden als Kommentar interpretiert und von der weiteren Bearbeitung ausgeschlossen.

Bei den folgenden Tabellenbeschreibungen wird ein Feld eines Datensatzes durch folgende Attribute spezifiziert:

- Nummer
- Name
- Kennzeichen, ob das Feld zum Primärschlüssel der Tabelle gehört
- Datentyp (s.u.)
- maximale Länge des Feldes (Anzahl der Zeichen)<sup>1</sup>
- Kennzeichen, ob das Feld unbedingt gefüllt sein muss (Pflichtfeld)

---

<sup>1</sup>Bei CSV-Datensätzen bestehen prinzipiell zwar keine Beschränkungen der einzelnen Feldlängen, bei bestimmten Feldern des Datentyps **Char** werden hier jedoch sich aus dem Verwendungszweck ergebende maximal mögliche bzw. sinnvolle Längen angegeben. Darüber hinaus sind bei der Datenanlage ggf. weitergehende Beschränkungen zu beachten, die durch das im Datenanlageprozeß verwendete Programm auferlegt werden.

Folgende **Datentypen** sind definiert:

**Char** Zeichenkette

Es gelten folgende lexikalischen und syntaktischen Bestimmungen:

1. Es sind alle druckbaren Zeichen bis auf das Feldtrennzeichen (Semikolon) erlaubt.
2. Soll ein Semikolon in der Zeichenkette enthalten sein, muss das ganze Feld in Anführungszeichen (""') eingeschlossen werden (denen kein singuläres Anführungszeichen nach- bzw. vorangestellt ist). Das öffnende und das schließende Anführungszeichen werden beim Lesen des Feldes nicht übernommen.
3. Ist das Feld in Anführungszeichen eingeschlossen, werden beim Lesen des Feldes zwei aufeinanderfolgende Anführungszeichen durch ein einzelnes ersetzt. Ein singuläres Anführungszeichen in einem durch Anführungszeichen eingeschlossenen Feld ist nicht erlaubt.
4. Ist das Feld in Anführungszeichen eingeschlossen, werden Leerzeichen zwischen dem schließenden Anführungszeichen und dem nächsten Feldtrennzeichen bzw. dem Zeilenende ignoriert.

**Num** Zahl

alle Ziffern sowie Dezimalpunkt, evtl. Minuszeichen an erster Stelle

**Bool** boolescher Wert

'1' – ja, '0' – nein

**Date** Datumsangabe

Format: 4 Stellen Jahr + 2 Stellen Monat + 2 Stellen Tag (entspricht ISO 8601, wobei Trennstriche zwischen Jahr, Monat und Tag entfallen)

Das Pflichtfeld–Kennzeichen ist nur für Felder des Datentyps *Char* relevant. Bei Feldern der anderen Datentypen ist immer eine Angabe zu machen. Bei Feldern des Datentyps *Num* sind die jeweils möglichen Werte der Beschreibung der jeweiligen Tabellen zu entnehmen.

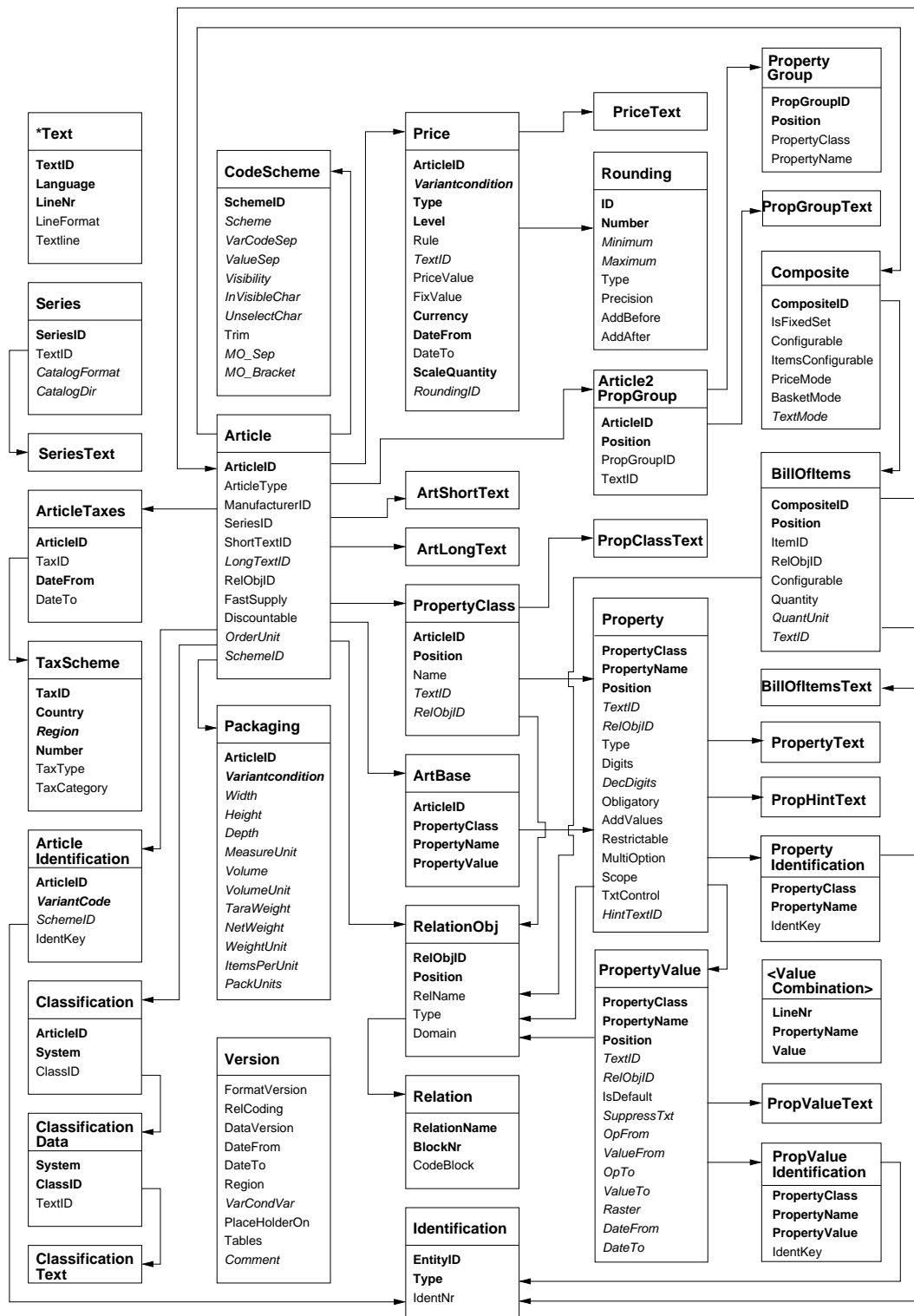
In verschiedenen Tabellen werden Felder für *Einheiten* definiert. Bezüglich der Angabe von Einheiten folgt OCD dem Standard **openTRANS** für den zwischenbetrieblichen elektronischen Austausch von Geschäftsdokumenten. Danach werden Einheiten gemäß dem *Common Code* der UN/ECE Recommendation 20<sup>2</sup> angegeben.

---

<sup>2</sup>[www.unece.org/cefact/rec/rec20en.htm](http://www.unece.org/cefact/rec/rec20en.htm)

## 2 Die Tabellen

### 2.1 Übersicht



Primärschlüsselfelder sind durch Fettdruck hervorgehoben und Felder, die keine Pflichtfelder sind, durch Kursivdruck.

Die Struktur der Text-Tabellen ist nur einmal dargestellt. Die alternativen Text-Tabellen (s. Abschn. 2.20) sind der Übersichtlichkeit halber nicht dargestellt.

## 2.2 Die Artikeltable

Tabellenname: **Article**

Pflichttable: ja

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ArticleID	X	Char		X	Grundartikelnummer
2.	ArticleType		Char		X	Artikelart
3.	ManufacturerID		Char		X	Herstellerkürzel
4.	SeriesID		Char		X	Serienkürzel
5.	ShortTextID		Char		X	Kurztextnummer
6.	LongTextID		Char			Langtextnummer
7.	RelObjID		Num		X	Beziehungsobjekt-Nummer
8.	FastSupply		Num		X	Schnell-Lieferungszähler
9.	Discountable		Bool		X	rabattierbarer Artikel?
10.	OrderUnit		Char	3		Bestell-Einheit
11.	SchemeID		Char			Identifikator des Kodierungsschemas für die Generierung der Endartikelnummer

Anmerkungen:

- Dies ist die Haupttable für alle Artikel. Die Artikelnummer wird als Schlüssel für den Zugriff auf weitere Tabellen zur Preisbestimmung, Klassifizierung usw. verwendet<sup>3</sup>. Als Artikelnummer ist hier die vom Hersteller verwendete Grundartikelnummer (Modellnummer) anzugeben. Über die Artikel-Identifikationstabelle (s. Abschn. 2.3) können weitere Nummern zur Identifikation des Artikels in verschiedenen Kontexten hinterlegt werden.
- Die Artikelart (Feld 2) bestimmt grundlegende Eigenschaften des Artikels. Folgende Artikelarten sind möglich:

Artikelart	Erklärung
P	einfacher Artikel (plain article): nicht konfigurierbar, keine Unterartikel
C	konfigurierbarer Artikel: Eigenschaften des Artikels können durch den Anwender festgelegt werden, keine Unterartikel
CS	zusammengesetzter Artikel (composite): kann Unterartikel enthalten, kann selber konfigurierbar sein

- Die Herstellerkürzel (Feld 3) werden zentral durch die Firma EasternGraphics GmbH vergeben und verwaltet.
- Für das Serienkürzel (Feld 4) sollten idealerweise nur die Großbuchstaben und Ziffern aus dem ASCII-Zeichensatz sowie der Unterstrich '\_' verwendet werden. Die Verwendung folgender Zeichen wird *nicht* empfohlen: \/?:\*"><|, ; = sowie das Leerzeichen<sup>4</sup>.
- Die Textnummern (Felder 5 und 6) dienen als Schlüssel für die Tabellen mit den Kurz- bzw. Langbeschreibungen der Artikel (s. Abschn. 2.20).
- Die Beziehungsobjekt-Nummer dient als Schlüssel für den Zugriff auf das Beziehungsobjekt in der Beziehungsobjekt-Table (s. Abschn. 2.15), an das Beziehungswissen zu dem Artikel gebunden ist. (Wird kein Beziehungsobjekt benötigt, so ist die Nummer 0 anzugeben.)

<sup>3</sup>Die Aufteilung der Artikel-bezogenen Informationen auf verschiedene Tabellen erhöht die Übersichtlichkeit durch Ausblenden von optionalen Informationen und erleichtert die Erweiterbarkeit als auch den inkrementellen Datenaustausch.

<sup>4</sup>Werden diese Zeichen verwendet, ist die korrekte Verarbeitung der Daten auf allen Plattformen und in allen OFML-Anwendungssystemen *nicht* gewährleistet.



- Durch den Schnell-Lieferungszähler (Feld 8) wird die Anzahl der Artikel bestimmt, ab der eine Schnell-Lieferung möglich ist, wobei die Zahl 0 anzeigt, dass generell keine Schnell-Lieferung für den Artikel möglich ist.
- Das Kennzeichen im Feld 9 gibt an, ob auf den ggf. hinterlegten Einkaufspreis des Artikels Rabatte angewendet werden können oder nicht. Der Wert 0 (false) bedeutet dabei, dass für den Artikel abweichend von der allgemeinen Konditionierung des Herstellers *keine* Abschläge vom Einkaufspreis erlaubt sind.
- Im Feld 10 wird die Einheit angegeben, in der der Artikel bestellt werden kann. Auf diese Einheit bezieht sich sowohl die Mengenangabe in einer Bestellung als auch der Preis in der Preistabelle. Die Einheit muß gemäß dem Common Code der UN/ECE Recommendation 20 angegeben werden. Gebräuchliche Einheiten für die Möbelbranche sind C62 – *Stück*, MTR – *Meter* und MTK – *Quadratmeter*. Erfolgt keine Angabe, wird als Standardeinheit *Stück* verwendet.
- Der im 11. Feld angegebene Identifikator dient zur Referenzierung des bei der Generierung der Endartikelnummer zu verwendenden Kodierungsschemas aus der Tabelle `CodeScheme` (Abschn. 2.24). Ist kein oder ein in der Schematabelle nicht referenzierter Identifikator angegeben, wird für den Artikel keine spezifische Endartikelnummer erzeugt<sup>5</sup>.

## 2.3 Die Artikel-Identifikationstabelle

Tabellenname: `ArticleIdentification`

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ArticleID	X	Char		X	(Grund)Artikelnummer
2.	VariantCode	X	Char			Code der Artikelvariante
3.	SchemeID		Char			Schema für Variantencode
4.	IdentKey		Char		X	Schlüssel für Identifikationstabelle

Anmerkungen:

- Die Tabelle dient zur Angabe von zusätzlichen Identifikationsnummern (unterschiedlichen Typs) für Artikel. Die zusätzlichen Identifikationsnummern werden dabei nicht direkt in dieser Tabelle, sondern indirekt über den im Feld 4 angegebenen Schlüssel in der Tabelle `Identification` (s.Abschn. 2.22) abgelegt.
- Soll die Identifikationsnummer nicht dem Grundartikel zugeordnet werden, sondern einer bestimmten Variante des (konfigurierbaren) Artikels, so muß diese im 2. Feld durch einen entsprechenden Code spezifiziert werden. Im Feld 3 muß dazu das dabei verwendete Codierungsschema angegeben werden (Schlüssel für Nummernschema-Tabelle, s.Abschn. 2.24).
- Existieren in der Tabelle zu einem Grundartikel mehrere Einträge, so muss jeweils ein Variantencode angegeben sein. Wird zu einer gegebenen Konfiguration eines Artikels kein passender Variantcode gefunden, so kann für den Artikel keine Identifikation des gewünschten Typs ermittelt werden.

<sup>5</sup>Die Endartikelnummer ist dann gleich der Grundartikelnummer.

## 2.4 Die Klassifikationstabelle

Tabellenname: **Classification**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ArticleID	X	Char		X	Grundartikelnummer
2.	System	X	Char		X	Name des Klassifikationssystems inkl. Versionsangabe
3.	ClassID		Char		X	ID der Klasse des Artikels

Anmerkungen:

- Die Tabelle dient zur Klassifizierung eines Artikels.
- Ein Artikel kann dabei nach verschiedenen Klassifikationssystemen klassifiziert werden. Aktuell sind folgende Systeme erlaubt:

System	Erklärung
ECLASS-x.y	Klassifizierung nach dem eClass-Modell mit Angabe der Version
UNSPSC	Klassifizierung nach dem Standard UN/SPSC
<Manufacturer>_*	Hersteller-spezifische Klassifizierung: die Systembezeichnung wird aus dem Herstellerkürzel, einem Unterstrich und einem beliebigen Nachsatz gebildet

Hersteller-spezifische Klassifizierungen können zur Definition von Warengruppen, Produkthierarchien u.ä. verwendet werden.

Tabellenname: **ClassificationData**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	System	X	Char		X	Name des Klassifikationssystems inkl. Versionsangabe
2.	ClassID	X	Char		X	ID der Klasse
3.	TextID		Char		X	Textnummer

Anmerkungen:

- Die Tabelle dient zur Angabe von Informationen zu einer Klassifizierung<sup>6</sup>.
- Die Textnummer dient als Schlüssel für die Tabelle **ClassificationText** (s. Abschn. 2.20), in welcher sprachspezifische Texte hinterlegt werden können, die die Klassifizierung beschreiben.

<sup>6</sup>Momentan wird nur die Angabe eines Textes zur Beschreibung der Klassifizierung unterstützt.

## 2.5 Die Packaging-Tabelle

Tabellenname: Packaging

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ArticleID	X	Char		X	Grundartikelnummer
2.	Variantcondition	X	Char			Variantenkondition
3.	Width		Char			Breite der Verpackungseinheit
4.	Height		Char			Höhe der Verpackungseinheit
5.	Depth		Char			Tiefe der Verpackungseinheit
6.	MeasureUnit		Char	3		Maßeinheit der Dimensionen 3 bis 5
7.	Volume		Char			Volumen der Verpackungseinheit
8.	VolumeUnit		Char	3		Maßeinheit des Volumens
9.	TaraWeight		Char			Gewicht der Verpackungseinheit
10.	NetWeight		Char			Gewicht des Einzelartikels
11.	WeightUnit		Char	3		Maßeinheit der Gewichte 9 bis 10
12.	ItemsPerUnit		Char			Anzahl der Artikel pro Verpackungseinheit
13.	PackUnits		Char			Anzahl der Verpackungseinheiten, die für den Artikel verwendet werden

Anmerkungen:

- Diese Tabelle dient zur Angabe von Informationen zur Verpackung eines Artikels, der komplett geliefert wird.
- Eine Verpackungseinheit kann dabei mehrere Artikel (derselben Nummer) beinhalten (Feld 12). Teile des Artikels, z.B. optionale Zubehörteile, können aber auch in separaten Verpackungseinheiten geliefert werden (Feld 13).
- Mit Hilfe von *Variantenkonditionen* (Feld 2) können – in Abhängigkeit von speziellen Merkmalsausprägungen – Maße, Volumen, Gewichte und Anzahlen von Verpackungseinheiten hinterlegt werden, die von der Grundauführung des Artikels abweichen. Die Verwendung und Behandlung solcher Variantenkonditionen ist in Abschn. 6 beschrieben. Die Beträge in den Einträgen mit Variantenkonditionen (nicht-leeres Feld 2) werden dabei immer als Differenz zu dem jeweiligen Grundbetrag aus dem Eintrag ohne Variantenkondition angegeben und können negativ sein<sup>7</sup>.
- Wenn das Feld 2 (Variantenkondition) nicht leer ist, kann im Feld 1 der Joker-Artikel „\*“ zur Angabe artikelübergreifend einheitlicher Werte für variantenabhängige Längen, Volumen und Gewichte verwendet werden. Dieser artikel-neutrale Tabelleneintrag wird jedoch nur dann berücksichtigt, wenn für den bearbeiteten Artikel kein eigener, spezifischer Eintrag mit der gleichnamigen Variantenkondition existiert.
- Die Felder 3-5, 7, 9, 10, 12 und 13 sind als optionale Zeichenkettenfelder deklariert, sie können also auch leer sein. Wenn nicht leer, müssen die Felder Zeichenketten-Darstellungen numerischer Werte enthalten<sup>8</sup>.

<sup>7</sup>Dies impliziert, daß für einen gegebenen Artikel immer ein Tabelleneintrag ohne Variantenkondition vorhanden sein muss. Die Datenelemente darin können bei Bedarf den Wert 0.0 aufweisen.

<sup>8</sup>äquivalent zu den Werten numerischer Merkmale in der Wertetabelle

- In den Feldern 6, 8 und 11 sind folgende Maßeinheiten erlaubt<sup>9</sup>:

Längen (Feld 6):

Code der Maßeinheit	Erklärung
CMT	Zentimeter
FOT	Fuß (foot)
INH	Zoll (inch)
MMT	Millimeter
MTR	Meter

Das Feld darf nicht leer sein, wenn eines der Felder 3-5 nicht leer ist.

Volumen (Feld 8):

Code der Maßeinheit	Erklärung
INQ	Kubikzoll (cubic inch)
LTR	Liter
MTQ	Kubikmeter

Das Feld darf nicht leer sein, wenn das Feld 7 nicht leer ist.

Gewicht (Feld 11):

Code der Maßeinheit	Erklärung
KGM	Kilogramm
LBR	Pfund
MGM	Milligramm

Das Feld darf nicht leer sein, wenn eines der Felder 9 und 10 nicht leer ist.

- Das Volumen kann sich von dem rechnerisch aus Breite, Höhe und Tiefe ermittelten Volumen unterscheiden, wenn Verpackungen verwendet werden, die ineinander gestapelt werden können.
- Das Gesamtgewicht (Brutto) einer Verpackungseinheit ergibt sich aus dem Gewicht der Verpackung (Feld 9) plus dem Produkt aus Gewicht des Einzelartikels (Feld 10) und der Anzahl der Artikel pro Verpackungseinheit (Feld 12).

---

<sup>9</sup>entspricht dem Common Code der UN/ECE Recommendation 20 ([www.unece.org/cefact/rec/rec20en.htm](http://www.unece.org/cefact/rec/rec20en.htm))

## 2.6 Die Tabelle kompositer Artikel

Tabellenname: Composite

Pflichttabelle: nein<sup>10</sup>

*Komposite Artikel* sind Artikel, die eine feste oder variable Anzahl von Unterartikeln (sub items) beinhalten. Dies kann eine simple Aggregation im Sinne eines Sets sein, aber auch eine Komposition aus funktionalen Gesichtspunkten.

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	CompositeID	X	Char		X	Artikelnummer des kompositen Artikels
2.	IsFixedSet		Bool		X	Anzahl der Unterartikel fest ?
3.	Configurable		Bool		X	kompositer Artikel konfigurierbar ?
4.	ItemsConfigurable		Bool		X	Unterartikel konfigurierbar ?
5.	PriceMode		Char	3	X	Modus der Preisermittlung
6.	BasketMode		Char	1	X	Modus der Darstellung der Unterartikel im Warenkorb
7.	TextMode		Char			Modus der Textdarstellung der Unterartikel im Warenkorb

Anmerkungen:

- Die Tabelle dient zur Festlegung der allgemeinen Eigenschaften eines kompositen Artikels. Die Unterartikel werden in der Stücklistentabelle (s. nächster Abschnitt) festgelegt.
- Im Feld 2 wird angegeben, ob die Anzahl der Unterartikel fest ist. Wenn *nein* angegeben ist, so variiert die Anzahl der Unterartikel in Abhängigkeit von den in der Stücklistentabelle zu den Unterartikeln zu hinterlegenden Existenzbedingungen.  
Ist *ja* angegeben, werden evtl. dennoch hinterlegte Existenzbedingungen für Unterartikel nicht ausgewertet!
- Im Feld 3 wird angegeben, ob der komposite Artikel selber konfigurierbar ist. Wenn *nein* angegeben ist, kann der komposite Artikel nicht konfiguriert werden, auch wenn ihm Merkmalsklassen (s. Abschn. 2.8) zugeordnet sind.
- Ist der Wert im Feld 4 *nein*, so dürfen die Unterartikel generell nicht konfiguriert werden, auch wenn dies für einzelne Unterartikel in der Stücklistentabelle anderweitig spezifiziert ist.  
Andersherum können einzelne Unterartikel in der Stücklistentabelle von der Konfigurierbarkeit ausgeschlossen werden, auch wenn die Konfigurierbarkeit durch den Wert *ja* im Feld 4 generell freigeschaltet ist.
- Der Modus im Feld 5 bestimmt die Art und Weise der Preisermittlung für den kompositen Artikel:

Preismodus	Erklärung
C	Preis ist an den kompositen Artikel gebunden (ggf. mit Variantenbedingungen)
S	Preis ergibt sich aus der Summe der Preise der Unterartikel
C+S	Preis des kompositen Artikel plus Summe der Preise der Unterartikel

Prinzipiell ist für jede mögliche Wertkombination der Felder 2-4 jeder der drei Preis-Modi denkbar/möglich. Es wird deswegen keine Begrenzung vorgegeben. Bei der Datenerfassung muß die Stimmigkeit beachtet/gewährleistet werden. Sind z.B. die Unterartikel konfigurierbar (Feld 4 wahr), so macht der Preis-Modus "C" in der Regel keinen Sinn, es sei denn, es ist durch Beziehungswissen (s.u.) gewährleistet, daß die Wertemengen aller preisrelevanten Merkmale der Unterartikel auf je genau einen Wert eingeschränkt sind.

<sup>10</sup>Die Tabelle wird nur benötigt, wenn die Datenbank auch tatsächlich komposite Artikel enthalten soll.

- Über den Modus im Feld 6 kann die Darstellung der Unterartikel in kaufmännischen Formularen gesteuert werden:

Basket-Modus	Erklärung
H	Unterartikel werden als Unterpositionen (d.h. hierarchisch) dargestellt, ggf. ohne Preis
T	die Unterartikel werden im Beschreibungstext des kompositen Artikels angeführt

- Der Modus im Feld 7 gibt an, wie die Unterartikel im Basket-Modus "T" im Beschreibungstext des kompositen Artikels beschrieben werden sollen<sup>11</sup>:

Text-Modus	Erklärung
BAN	durch die Grundartikelnummer (base article number)
FAN	durch die Endartikelnummer (final article number)
ST	durch Artikelkurztext (short text)
LT	durch Artikellangtext (long text)
BAN+ST	durch Grundartikelnummer und Artikelkurztext
BAN+LT	durch Grundartikelnummer und Artikellangtext
FAN+ST	durch Endartikelnummer und Artikelkurztext
FAN+LT	durch Endartikelnummer und Artikellangtext
ST+BAN	durch Artikelkurztext und Grundartikelnummer
ST+FAN	durch Artikelkurztext und Endartikelnummer
LT+BAN	durch Artikellangtext und Grundartikelnummer
LT+FAN	durch Artikellangtext und Endartikelnummer

## 2.7 Die Stücklistentabelle

Tabellenname: BillOfItems

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	CompositeID	X	Char		X	Artikelnummer des kompositen Artikels
2.	Position	X	Num		X	Position des Unterartikels
3.	ItemID		Char		X	Artikelnummer des Unterartikels
4.	RelObjID		Num		X	Beziehungsobjekt-Nummer
5.	Configurable		Bool		X	Unterartikel konfigurierbar ?
6.	Quantity		Num		X	Menge
7.	QuantUnit		Char	3		Mengeneinheit
8.	TextID		Char			Textnummer

Anmerkungen:

- In dieser Tabelle wird angegeben, welche Unterartikel (sub items) ein kompositer Artikel (s. vorheriger Abschn.) beinhalten kann.
- Die Position eines Unterartikels innerhalb des kompositen Artikels (Feld 2) wird bei der Bestellausgabe berücksichtigt.
- Über das Beziehungsobjekt (Feld 4) kann eine Existenzbedingung für den Unterartikel angegeben werden (s. Abschn. 2.15). Existenzbedingungen sind als Beziehungsart *Vorbedingung* mit Verwendungsgebiet "BOI" anzugeben. Falls ein Existenzbedingung hinterlegt ist und diese nicht erfüllt ist,

<sup>11</sup>Für den Basket-Modus "H" ist dieser Modus nicht notwendig, da sich die Darstellung der Unterartikel als Bestell-Position dann nach den Vorgaben der jeweiligen Applikation richtet.

wird der Unterartikel nicht in die aktuelle Stückliste des kompositen Artikels aufgenommen. Existenzbedingungen werden jedoch grundsätzlich nur dann ausgewertet, wenn im Feld *IsFixedSet* der Tabelle **Composite** (s. vorherigen Abschn.) für den kompositen Artikel *nein* angegeben ist.

- Im Feld 5 wird angegeben, ob der Unterartikel konfigurierbar ist. Wenn *nein* angegeben ist, kann er nicht konfiguriert werden, auch wenn ihm Merkmalsklassen (s. Abschn. 2.8) zugeordnet sind. Unterartikel sind generell jedoch nur dann konfigurierbar, wenn dies durch den kompositen Artikel freigegeben ist, siehe Feld *ItemsConfigurable* in der Tabelle **Composite**.
- Über die Mengenangabe in Feld 6 können gleichartige Artikel auf einer Stücklistenposition zusammengefaßt werden. Die zu dieser Position gehörenden Unterartikel dürfen dann nicht konfiguriert werden, d.h. der Wert in Feld 5 wird ignoriert<sup>12</sup>.
- Im Feld 7 wird die Einheit angegeben, auf die sich die Mengenangabe in Feld 6 bezieht. Die Einheit muß gemäß dem Common Code der UN/ECE Recommendation 20 angegeben werden. Erfolgt keine Angabe, wird als Standardeinheit *Stück* (C62) verwendet.
- Die Textnummer im Feld 8 dient als Schlüssel für die Tabelle **BillOfItemsText** (s. Abschn. 2.20), in der Texte hinterlegt werden können, die in einem kaufmännischen Formular zusätzlich zu den normalen Artikelinformationen für eine Stücklistenposition auszugeben sind. Die Art und Weise dieser Ausgabe ist abhängig von der verwendeten Applikation.

## 2.8 Die Merkmalsklassentabelle

Tabellenname: **PropertyClass**

Pflichttabelle: ja<sup>13</sup>

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ArticleID	X	Char		X	Artikelnummer
2.	Position	X	Num		X	Position der Klasse
3.	Name		Char		X	Name der Klasse
4.	TextID		Char			Textnummer
5.	RelObjID		Num		X	Beziehungsobjekt-Nummer

Anmerkungen:

- In dieser Tabelle werden den Artikeln die Merkmalsklassen zugeordnet, die die Merkmale des Artikels beschreiben.
- Die Position der Klasse innerhalb der Menge der Merkmalsklassen eines Artikels beeinflusst die Reihenfolge in Aufzählungen der Merkmale des Artikels (Variantentext, Eigenschaftseditoren u.ä.).
- Für den Namen einer Merkmalsklasse (Feld 3) sind alle alphanumerischen Zeichen inklusive dem Unterstrich erlaubt, wobei das erste Zeichen kein numerisches sein darf.
- Über das Beziehungsobjekt (Feld 5) können an die Merkmalsklasse Beziehungen vom Typ *Vorbedingung* und *Aktion* gebunden werden (s. Abschn. 2.15). (Wird kein Beziehungsobjekt benötigt, so ist die Nummer 0 anzugeben.)

<sup>12</sup>Hinweis: Sollen die Unterartikel in einer OFML-basierten Anwendung auch graphisch dargestellt werden, muß vor der OCD-Datenanlage abgeklärt werden, ob die jeweilige Anwendung eine graphische Darstellung für zusammengefasste Stücklistenartikel unterstützt.

<sup>13</sup>Die Tabelle kann entfallen, wenn in der Datenbank keine Konfigurationsdaten, sondern z.B. nur Artikeltexte und -preise angelegt werden sollen.

## 2.9 Die Merkmalstabelle

Tabellenname: Property

Pflichttabelle: ja<sup>14</sup>

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	PropertyClass	X	Char		X	Bezeichner der Merkmalsklasse
2.	PropertyName	X	Char		X	Bezeichner des Merkmals
3.	Position		Num		X	Position des Merkmals
4.	TextID		Char			Textnummer
5.	RelObjID		Num		X	Beziehungsobjekt-Nummer
6.	Type		Char	1	X	Datentyp der Merkmalswerte: C - Char T - Text N - Num L - Length
7.	Digits		Num		X	Anzahl der Stellen (gesamt)
8.	DecDigits		Num			(davon) Anzahl der Nachkommastellen
9.	Obligatory		Bool	1	X	Eingabe erfordert?
10.	AddValues		Bool	1	X	Zusätzliche Werte erlaubt?
11.	Restrictable		Bool	1	X	Wertemenge einschränkbar?
12.	MultiOption		Bool	1	X	mehrwertiges Merkmal?
13.	Scope		Char	2	X	Geltungsbereich, s. Tabelle unten
14.	TxtControl		Num		X	Text-Steuerung
15.	HintTextID		Char			Textnummer für Hinweistext

Geltungsbereich	Erklärung (Details s.u.)
C	konfigurierbar (sichtbar)
R	nur in Beziehungswissen
RV	nicht konfigurierbar, aber für Anwender sichtbar
RG	nicht konfigurierbar, aber Grafik-relevant

Anmerkungen:

- In dieser Tabelle werden die Merkmale pro Merkmalsklasse aufgelistet.
- Die Namen der Merkmale sind symbolische (sprachunabhängige) Bezeichner. Verwendet werden dürfen alphanumerische Zeichen inklusive dem Unterstrich, wobei das erste Zeichen kein numerisches sein darf.  
Sprechende (sprachabhängige) Bezeichner (zur Verwendung in den Benutzeroberflächen) werden in der Tabelle **PropertyText** abgelegt (s. Abschn. 2.20). Dazu wird im 4. Feld eine Textnummer als Zugriffsschlüssel vergeben.  
Die Angabe einer Textnummer ist nur bei sichtbaren Merkmalen gemäß Feld 13 notwendig (siehe dazu auch Anmerkung unten).
- Innerhalb der Merkmalsklassen eines Artikels darf ein gegebenes Merkmal (Feld 2) nur einmal vorkommen<sup>15</sup>.
- Die im Feld 3 angegebene Position des Merkmals innerhalb der Merkmalsklasse beeinflusst die Reihenfolge in Aufzählungen der Merkmale des Artikels (Variantentext, Eigenschaftseditoren u.ä.).

<sup>14</sup>Die Tabelle kann entfallen, wenn in der Datenbank keine Konfigurationsdaten, sondern z.B. nur Artikeltexte und -preise angelegt werden sollen.

<sup>15</sup>Ansonsten kann Beziehungswissen (s. Abschn. 2.16) nicht eindeutig ausgewertet werden, da dort die Merkmale nicht mit ihrer Merkmalsklasse qualifiziert werden.



- Die Beziehungsobjekt-Nummer (Feld 5) dient als Schlüssel für den Zugriff auf das Beziehungsobjekt in der Tabelle `RelationObj` (s. Abschn. 2.15), an welches das Beziehungswissen für das Merkmal gebunden ist. (Wird kein Beziehungsobjekt benötigt, so ist die Nummer 0 anzugeben.)
- Der Datentyp (Feld 6) legt den Typ der Werte fest, die das Merkmal annehmen kann, und bestimmt die Art der Darstellung von Werten des Merkmals in der Merkmalswerttabelle:
  - Werte des Datentyps 'C' sind einfache Zeichenketten mit einer maximalen Länge gemäß Angabe im Feld 7. Es dürfen alle darstellbaren Zeichen aus dem Zeichensatz ISO 8859-1 (Latin-1) außer dem Leerzeichen und dem Backslash ('\') verwendet werden.
  - Der Datentyp 'T' wird für Merkmale verwendet, bei denen der Anwender einen beliebigen Text inklusive Zeilenumbrüche eingeben kann. Werte dieses Typs sind Zeichenketten mit einer maximalen Länge gemäß Angabe im Feld 7. (Ist im Feld 7 der Wert 0 eingetragen, besteht keine Längenbegrenzung.) Zeilenumbrüche werden dabei durch ein Newline-Zeichen ('\n') dargestellt. In der Merkmalswerttabelle dürfen keine Werte hinterlegt sein (bzw. diese werden ignoriert).
  - Werte der Datentypen 'N' und 'L' sind reelle oder ganze Zahlen, die in einfacher Dezimalpunktnotation dargestellt werden. Das Feld 7 gibt dabei die maximale Anzahl aller Stellen ohne Dezimalpunkt vor, und Feld 8 die Anzahl der davon für den gebrochenen Teil verwendeten Darstellung. Bei negativen Zahlen steht an erster Stelle das Minuszeichen (womit für die Ziffernstellen eine Stelle weniger zur Verfügung steht als in Feld 7 angegeben).
  - Die Datentypen 'N' und 'L' (Feld 6) werden im wesentlichen gleich behandelt. Der Unterschied besteht in der Formatangabe des OFML-Merkmals, welches für das jeweilige OCD-Merkmal generiert wird (s.a. OFML-Dokumentation, Abschnitt „Formatspezifikationen für Properties“ im Anhang „Formatspezifikationen“):
    - \* Das Format für Merkmale des Typs 'N' ist `%<Feld7>.<Feld8>f`, wenn die Anzahl der Nachkommastellen (Feld 8) ungleich 0 ist, sonst `%<Feld7>d`.
    - \* Das Format für Merkmale des Typs 'L' ist `@L`, womit der Property-Editor der Applikation aufgefordert ist, zur Darstellung bzw. Eingabe des Wertes die vom Nutzer eingestellte Maßeinheit zu verwenden. Der Property-Editor führt dazu eine Konvertierung zwischen der nutzerdefinierten Maßeinheit und der in OFML für Längenmaßangaben verwendeten Maßeinheit (m) durch. Werte für Merkmale dieses Typs in der Tabelle `PropertyValue` müssen also in Metern abgegeben werden.
- In Feld 9 wird spezifiziert, ob es sich bei dem Merkmal um ein Pflichtmerkmal (1) oder ein optionales Merkmal (0) handelt. Ein Pflichtmerkmal muß durch den Anwender bewertet werden. Solange ein Artikel nicht bewertete Pflichtmerkmale besitzt, ist seine Konfiguration nicht vollständig (ungültig). Das Kennzeichen ist nur für Merkmale des Typs 'C' relevant. Ist für ein als optional gekennzeichnetes Merkmal eine Wertemenge (siehe Tabelle `PropertyValue`, Abschn. 2.13) vorgegeben, so wird vom Anwendungssystem automatisch zusätzlich der Pseudo-Wert `VOID` für den Zustand „nicht ausgewählt“ generiert und verwendet. Bei Pflicht-Merkmalen mit einer vorgegebenen Menge von Merkmalswerten und ohne Möglichkeit der freien Werteingabe (siehe Feld 10) wird durch das Anwendungssystem gewährleistet, daß immer ein Wert aus der Wertetabelle ausgewählt ist; ein solches Merkmal ist also immer bewertet. Bei anderen Merkmalen muß durch Bereitstellung von Produktbeziehungswissen (Auswahlbedingungen, siehe Tabelle `RelationObj`, Abschn.2.15) die Vollständigkeit der Konfiguration sichergestellt werden. Bei der Verwendung dieses Kennzeichens ist zu beachten, daß es sich auf die Eingabepflicht für den Anwender bezieht. Es besteht kein direkter Zusammenhang mit dem produktlogischen Begriff *Option* für Merkmale, bei dem der Anwender aus einer Menge von Werten (Choiceliste) auswählen kann, aber nicht muss. Optionen aus produktlogischer Sicht können auch durch OCD-Merkmale mit Pflicht-Kennzeichen abgebildet werden. Bei numerischen Merkmalen ist dies sogar erforderlich (s.o.), der Zustand „nicht ausgewählt“ kann und muß dann durch einen speziellen Wert (z.B. 0) abgebildet werden.
- Das Kennzeichen im Feld 10 (`AddValues`) gibt an, ob der Anwender frei Werte eingeben kann, ggf. zusätzlich zu den in der Wertetabelle für das Merkmal hinterlegten Werten. Dies kann zur Eingabe

von freien Texten, Mengen oder Maßen genutzt werden.

Hinweis: Ist bei numerischen Merkmalen die Eingabe nur in bestimmten Wertebereichen erlaubt, so müssen in der Wertetabelle entsprechende Interval-Werte hinterlegt werden (siehe Tabelle `PropertyValue`, Abschn. 2.13) und das Kennzeichen muß dann den Wert 0 (*false*) besitzen.

Das Kennzeichen wird nur für einwertige und nicht-einschränkbare konfigurierbare Merkmale ausgewertet, mit folgenden Einschränkungen:

- Merkmalen des Typs T sowie Merkmalen ohne hinterlegte Werte in der Wertetabelle kann generell jeder beliebige Wert zugewiesen werden, d.h. auch, wenn das Kennzeichen den Wert 0 (*false*) hat.
- Sind für das Merkmal Werte in der Stammdatentabelle hinterlegt, können auch nur diese Werte zugewiesen bzw. gesetzt werden, d.h., das Kennzeichen wird dann ignoriert.

Bei allen anderen Merkmalsarten wird das Feld nicht ausgewertet. Bezüglich der Zuweisung von Werten gelten statt dessen folgende Bestimmungen:

- Nicht-konfigurierbaren Merkmalen kann generell jeder beliebige Wert zugewiesen werden, es sei denn, für das Merkmal sind Werte in der Wertetabelle hinterlegt: dann können dem Merkmal auch nur diese Werte zugewiesen werden.
- Einschränkbar bzw. mehrwertigen konfigurierbaren Merkmalen können nur die in der Wertetabelle hinterlegten Werte zugewiesen werden.

- Soll die Wertemenge des Merkmals durch Beziehungen vom Typ *Constraint* eingeschränkt werden können (s. Abschn.2.15), muß das Kennzeichen im Feld 11 auf 1 gesetzt sein.

Die Menge der Werte eines normalen, *nicht* einschränkbar Merkmal, aus denen der Anwender auswählen kann, umfasst alle Werte, die in der Merkmalswerttabelle für das Merkmal angelegt sind **und** die entweder keine Vorbedingung besitzen oder deren Vorbedingungen hinsichtlich der aktuellen Konfiguration des Artikels erfüllt sind. Pflichtmerkmale sind dabei immer mit einem der Werte belegt.

*Einschränkbare Merkmale* werden diesbezüglich anders behandelt. Die Wertemenge dieser Merkmale wird ausgehend von der Menge der Werte in der Merkmalswerttabelle durch Constraints eingeschränkt. Sie gelten erst dann als bewertet, wenn die Wertemenge entweder durch ein Constraint auf genau einen Wert eingeschränkt wurde oder wenn eine Auswahl durch den Anwender erfolgt ist. Die Konfiguration eines Artikels ist erst vollständig, wenn alle einschränkbar Merkmale bewertet sind. Ist ein einschränkbar Merkmal nicht bewertet, kann der Artikel nicht bestellt werden.

- Mehrwertige Merkmale (Feld 12) sind Merkmale, die mehrere Werte auf einmal annehmen können (z.B. „Sonderausstattung“).
- Der *Geltungsbereich* (Feld 13: **Scope**) gibt an, ob das Merkmal vom Anwender eines Konfigurationssystems konfiguriert (verändert) werden darf, ob es für den Anwender sichtbar ist und ob es bei der Erzeugung der grafischen Repräsentation des Artikels benötigt wird:
  - Nur Merkmale des Geltungsbereich „C“ (oder Leerzeichen, d.h. keine Angabe) sind konfigurierbar. Sie sind damit per se auch sichtbar und können bei der Erzeugung der grafischen Repräsentation verwendet werden.
  - Merkmale aller anderen Geltungsbereiche sind Hilfsmerkmale, die in Beziehungswissen verwendet werden können.
  - Merkmale des Geltungsbereichs „RV“ werden dem Anwender darüberhinaus als nur-lesbar (read-only) angezeigt und können auch bei der Erzeugung der grafischen Repräsentation verwendet werden.
  - Merkmale des Geltungsbereichs „RG“ sind für den Anwender nicht sichtbar, werden aber bei der Erzeugung der grafischen Repräsentation benötigt.

Der Geltungsbereich hat auch Einfluß auf die Persistenz und Initialisierung:

- Merkmale des Geltungsbereichs „R“ werden nur innerhalb der Auswertung von Beziehungswissen verwendet. Der aktuelle Zustand dieser Merkmale ist deswegen außerhalb eines Konfigurationsvorgangs nicht verfügbar (d.h., wird nicht persistent am Artikel gespeichert). Eine Konsequenz davon ist, daß Merkmale dieses Scopes zu Beginn jedes Konfigurationsvorgangs initialisiert werden (s.u.).

- Der Zustand der Merkmale aller anderen Geltungsbereiche wird auch nach Ausführung eines Konfigurationsvorgangs benötigt<sup>16</sup> und deswegen persistent am Artikel gespeichert. Die Initialisierung (s.u.) findet bei diesen Merkmalen somit nur einmal unmittelbar bei Artikelerzeugung statt.
- Das Feld 14 enthält einen Code, der die Generierung des Textes steuert, der das Merkmal in kaufmännischen Formularen (Artikelliste u.ä.) beschreibt. In Abschnitt 5 ist die Art und Weise der Steuerung näher beschrieben. (Der Code 0 kennzeichnet dabei das Standardverfahren bei einzeiligen Texten.)
- Im Feld 15 kann eine Textnummer (Zugriffsschlüssel) angegeben werden, unter der in der Tabelle `PropHintText` (s. Abschn. 2.20) ein (mehrzeiliger) Hinweistext zu dem Merkmal hinterlegt ist. Dieser kann von einer Applikation als Hint<sup>17</sup> angezeigt werden, wenn der Nutzer den Mauszeiger über den Merkmalsbezeichner im Eigenschaftseditor der Applikation bewegt. Als grundsätzlicher Aufbau wird folgende Struktur empfohlen:
  - Merkmalsbezeichner (als Sicherheit für den Fall, daß durch die Spaltenformatierung ein Teil des Merkmalsbezeichner im Eigenschaftseditor nicht sichtbar ist)
  - Angaben zu gültigen Wertebereichen
  - sonstige Verwendungshinweise

Zwischen den Attributen eines Merkmals bestehen *Abhängigkeiten*, die die möglichen Kombinationen der Attribute einschränken. Diese Abhängigkeiten werden durch die folgenden Regeln beschrieben:

- Die Kennzeichen *Obligatory* und *AddValues* sind nur bei konfigurierbaren Merkmalen (Scope C) relevant.
- Konfigurierbare numerische Merkmale sind immer Pflichtmerkmale<sup>18</sup>.
- Konfigurierbare einschränkbare Merkmale sind immer Pflichtmerkmale.
- Merkmale zur freien Zeichenketten- bzw. Texteingabe durch den Nutzer sind Pflichtmerkmale, d.h., eine leere Zeichenkette bzw. ein leerer Text ist ein realer Wert.
- Merkmale zur freien Texteingabe durch den Nutzer (Typ T) sind implizit konfigurierbar (d.h. Scope C ist vorausgesetzt). Die Kennzeichen *Restrictable* und *MultiOption* sind für diese Merkmale nicht relevant. Desweiteren sind nur die Text-Steuercodes 0 und 5 erlaubt (s. Anhang 5).
- Nicht-konfigurierbare Merkmale können einschränkbar sein. Im Gegensatz zu konfigurierbaren einschränkbaren Merkmalen müssen diese jedoch nicht (durch das Beziehungswissen) bewertet sein, damit die Konfiguration des Artikels vollständig ist.
- Bei (konfigurierbaren) einschränkbaren Merkmalen ist die Eingabe zusätzlicher Werte nicht möglich.
- Bei konfigurierbaren einschränkbaren Merkmalen muß eine Werteliste vorliegen, oder aber es muß ein Constraint existieren (und wirksam sein), das eine Bewertung des Merkmals vornimmt. Ansonsten wäre die Konfiguration niemals vollständig.
- Interval-Werte sind nur bei konfigurierbaren nicht-einschränkbaren numerischen Merkmalen relevant.
- Der Typ L ist nur bei sichtbaren Merkmalen (Scopes C und RV) sinnvoll.
- Merkmale des Typs T können nicht mehrwertig sein.
- Für mehrwertige Merkmale sind die Kennzeichen *AddValues* und *Restrictable* nicht relevant.
- Interval-Werte sind für mehrwertige Merkmale nicht möglich.

---

<sup>16</sup>z.B. für die Anzeige in Eigenschaftseditoren und zum Aufbau der Grafik

<sup>17</sup>temporäres Textfeld an der Stelle des Mauszeigers

<sup>18</sup>Bei numerischen Merkmalen ist immer eine Eingabe gefordert, da das System ansonsten keine Operationen mit solchen Merkmalen durchführen kann.

Die Merkmale der verschiedenen Typen und Geltungsbereiche werden wie folgt *initialisiert*<sup>19</sup>:

- Merkmale des Typs T werden mit einem Leerstring initialisiert.
- Der initiale Zustand von *einschränkbar*en Merkmalen ist undefiniert (nicht bewertet).
- Nicht-einschränkbare *konfigurierbare* Merkmale werden mit dem als Default gekennzeichneten Wert aus der Wertetabelle initialisiert<sup>20</sup>.  
Ist kein Wert der Wertetabelle als Default gekennzeichnet, so werden Pflichtmerkmale mit dem ersten Wert aus der Wertetabelle initialisiert, während der initiale Zustand von optionalen Merkmalen dann undefiniert ist.  
Sind gar keine Werte in der Wertetabelle hinterlegt, werden Merkmale des Typs C mit einer leeren Zeichenkette und Merkmale der Typen N und L mit dem Wert 0 bzw. 0.0 initialisiert.
- Nicht-einschränkbare *nicht-konfigurierbare Merkmale* werden mit dem ersten Wert aus der Artikelstammdatentabelle initialisiert, nicht jedoch bei Merkmalen der Geltungsbereiche RG und RV, wenn für diese in der Wertetabelle Werte hinterlegt sind.  
Sind keine Werte in der Artikelstammdatentabelle hinterlegt, aber in der Wertetabelle, ist das Verhalten wie bei konfigurierbaren Merkmalen. (Bei Merkmalen des Geltungsbereichs R werden Werte aus der Wertetabelle jedoch nur berücksichtigt, wenn dort genau ein Wert hinterlegt ist oder die Werte mit Vorbedingungen versehen sind.)  
Sind weder in der Artikelstammdatentabelle noch in der Wertetabelle Werte hinterlegt, ist der initiale Zustand von Merkmalen des Geltungsbereichs R undefiniert, während Merkmale der Geltungsbereiche RG und RV je nach Typ mit einer leeren Zeichenkette oder 0 bzw. 0.0 initialisiert werden.

## 2.10 Die Merkmal-Identifikationstabelle

Tabellenname: `PropertyIdentification`

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	PropertyClass	X	Char		X	Bezeichner der Merkmalsklasse
2.	PropertyName	X	Char		X	Bezeichner des Merkmals
3.	IdentKey		Char		X	Schlüssel für Identifikationstabelle

Anmerkungen:

- Die Tabelle dient zur Angabe von zusätzlichen Identifikationsnummern (unterschiedlichen Typs) für Merkmale.
- Die zusätzlichen Identifikationsnummern werden dabei nicht direkt in dieser Tabelle, sondern indirekt über den im Feld 3 angegebenen Schlüssel in der Tabelle `Identification` (Abschn. 2.22) abgelegt.

<sup>19</sup>Unter Initialisierung wird die Wertbelegung eines Merkmals vor der Auswertung von Beziehungswissen verstanden. In Aktionen und Constraints können die Werte dann noch verändert werden.

<sup>20</sup>Es darf nur ein Wert als Default-Wert gekennzeichnet sein. Sind dennoch mehrere Werte als Default markiert, ist das Verhalten der Applikation undefiniert.

## 2.11 Merkmalsgruppen

Merkmalsgruppen können von OFML-Applikationen zur Gruppierung der Merkmale eines Artikels im Eigenschaftseditor verwendet werden. Dazu müssen folgende beiden Tabellen angelegt werden:

Tabellenname: **Article2PropGroup**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ArticleID	X	Char		X	Grundartikelnummer
2.	Position	X	Int		X	Position der Merkmalsgruppe
3.	PropGroupID		Char		X	ID der Merkmalsgruppe
4.	TextID		Char			Textnummer

Anmerkungen:

- In dieser Tabelle wird ein Artikel einer oder mehreren Merkmalsgruppen zugeordnet. Enthält die Tabelle für einen gegebenen Artikel keine Einträge, kann eine OFML-Applikation eine Gruppierung anhand seiner Merkmalsklassenzuordnung vornehmen.
- Die Position (Feld 2) legt die Reihenfolge der Merkmalsgruppen bei der Anzeige im Eigenschaftseditor fest. Ist für einen Artikel nur eine Merkmalsgruppe angegeben und enthält diese alle (aktuell sichtbaren) Merkmale des Artikels, kann die Applikation auf die Anzeige der Gruppe verzichten (flache Darstellung).
- Die ID der Merkmalsgruppe (Feld 3) dient als Fremdschlüssel für die Tabelle **PropertyGroup** (s.u.).
- Die Text-Nummer (Feld 4) dient als Fremdschlüssel für die Tabelle **PropGroupText** (s. Abschn. 2.20) zur Hinterlegung einer (sprach-spezifischen) Merkmalsgruppenbezeichnung. Prinzipiell kann ein und dieselbe Merkmalsgruppe bei verschiedenen Artikeln mit unterschiedlichen Textnummern und damit unterschiedlichen Bezeichnungen versehen werden. Im Normalfall wird jedoch dieselbe Textnummer und damit eine einheitliche Bezeichnung verwendet. Ist für eine gegebene Sprache keine Bezeichnung für eine gegebene Gruppe hinterlegt, wird die ID der Gruppe zur Anzeige im Eigenschaftseditor verwendet.

Tabellenname: **PropertyGroup**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	PropGroupID	X	Char		X	ID der Merkmalsgruppe
2.	Position	X	Int		X	Position des Merkmals
3.	PropertyClass		Symbol		X	Bezeichner der Merkmalsklasse
4.	PropertyName		Symbol		X	Bezeichner des Merkmals

Anmerkungen:

- In dieser Tabelle werden die Merkmale einer Merkmalsgruppe definiert.
- Die Position (Feld 2) legt die Reihenfolge der Merkmale innerhalb der Merkmalsgruppe bei der Anzeige im Eigenschaftseditor fest.
- Merkmale (Felder 3 und 4), die ein gegebener Artikel nicht besitzt, werden ignoriert.
- Aktuell sichtbare Merkmale des Artikels, welche nicht in den Merkmalsgruppen des Artikels gelistet sind, erscheinen nach diesen in einer „künstlichen“ Merkmalsgruppe „Sonstige“ in undefinierter Reihenfolge.

## 2.12 Die Artikelstammtabelle

Tabellenname: ArtBase

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ArticleID	X	Char		X	Artikelnummer
2.	PropertyClass	X	Char		X	Name der Merkmalsklasse
3.	PropertyName	X	Char		X	Name des Merkmals
4.	PropertyValue	X	Char		X	Merkmalswert

Anmerkungen:

- In der Tabelle können artikelspezifische Angaben zu fixen bzw. erlaubten Werten von ausgewählten Merkmalen gemacht werden, indem einem Merkmal des Artikels ein oder mehrere Werte (nacheinanderfolgende Datensätze) zugewiesen werden.
- Bei Merkmalen, für die in der Merkmalswerttabelle (s. Abschn. 2.13) Werte hinterlegt sind, bedeutet dies eine Einschränkung der Wertemenge in Bezug auf die in der Merkmalswerttabelle hinterlegten diskreten Werte<sup>21</sup>. Dies impliziert, daß dann im Feld 4 nur Werte angegeben werden dürfen, die auch in der Merkmalswerttabelle für das Merkmal hinterlegt sind.
- Die Wertzuweisungen für ein Merkmal in der Artikelstammtabelle haben Vorrang vor eventuellen Vorschlagswerten für das Merkmal in der Merkmalswerttabelle!

## 2.13 Die Merkmalswerttabelle

Tabellenname: PropertyValue

Pflichttabelle: ja<sup>22</sup>

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	PropertyClass	X	Char		X	Bezeichner der Merkmalsklasse
2.	PropertyName	X	Char		X	Bezeichner des Merkmals
3.	Position	X	Num		X	Position des Merkmalswertes
4.	TextID		Char			Textnummer
5.	RelObjID		Num		X	Beziehungsobjekt-Nummer
6.	IsDefault		Bool	1	X	Vorschlagswert?
7.	SuppressTxt		Bool	1	X	Text-Unterdrückungskennzeichen
8.	OpFrom		Char	2		Operator Von
9.	ValueFrom		Char			Merkmalswert Von
10.	OpTo		Char	2		Operator Bis
11.	ValueTo		Char			Merkmalswert Bis
12.	Raster		Char			Schrittweite
13.	DateFrom		Date	8		Gültig von
14.	DateTo		Date	8		Gültig bis

<sup>21</sup>In der Merkmalswerttabelle hinterlegte Interval-Werte sind von der Einschränkung nicht betroffen.

<sup>22</sup>Die Tabelle kann entfallen, wenn in der Datenbank keine Konfigurationsdaten, sondern z.B. nur Artikeltexte und -preise angelegt werden sollen.

Anmerkungen:

- In dieser Tabelle werden alle möglichen Werte pro Merkmal aufgelistet.
- Die Werte (Felder 9 und 11) sind String-Darstellungen numerischer Werte oder symbolische (sprachunabhängige) Bezeichner. Sprechende (sprachabhängige) Bezeichner (zur Verwendung in den Benutzeroberflächen) werden in der Tabelle `PropValueText` abgelegt (s. Abschn. 2.20). Dazu wird im 4. Feld eine Textnummer als Zugriffsschlüssel vergeben. Ist keine Textnummer angegeben bzw. kein Text in einer geforderten Sprache hinterlegt, so wird bei Merkmalen des Typs 'C' der symbolische (sprachunabhängige) Bezeichner (aus dieser Tabelle) angezeigt, bei Merkmalen der anderen Typen der entsprechende numerische Wert.
- Innerhalb eines Merkmals darf ein spezifischer Wert nur einmal angegeben werden.
- Die Beziehungsobjekt-Nummer dient als Schlüssel für den Zugriff auf das Beziehungsobjekt in der Tabelle `RelationObj` (s. Abschn. 2.15), an welches das Beziehungswissen für den Merkmalswert gebunden ist. (Wird kein Beziehungsobjekt benötigt, so ist die Nummer 0 anzugeben.)
- Der im Feld 6 (`IsDefault`) als Vorschlagswert gekennzeichnete Wert wird bei der Initialisierung des Merkmals (s. Merkmalstabelle, Abschn. 2.9) verwendet<sup>23</sup>.  
Ist keiner der Werte eines Merkmals als Vorschlagswert gekennzeichnet, wird bei Pflichtmerkmalen der erste Wert als initialer Wert verwendet, bei optionalen Merkmalen der virtuelle Wert „nicht ausgewählt“. Für letzteren ist bei Merkmalen des Typs 'C' das interne Kürzel `VOID` vorgesehen und reserviert. Dieses darf also nicht für einen realen Wert solcher Merkmale verwendet werden.
- Das Kennzeichen im Feld 7 gibt für ein konfigurierbares Merkmal an, ob es in der Beschreibung des Artikels angezeigt werden soll (0 bzw. leer) oder nicht (1), wenn der Wert aktuell durch den Anwender ausgewählt ist.
- Der Merkmalswert wird in den Feldern 8 bis 11 hinterlegt, wobei die Möglichkeit besteht, für den Merkmalswert ein Intervall anzugeben. (Dies kann insbesondere bei Maßmerkmalen Anwendung finden.)
  - Ein fester Merkmalswert (ohne Eingabebereich) wird durch den Operator 'EQ' gekennzeichnet. Ob dabei die Felder für den Von-Wert oder die Felder für den Bis-Wert verwendet werden, ist egal. Die Felder für den jeweils nicht benutzten Wert müssen leer sein.
  - Ein offener Eingabebereich wird durch die Operatoren 'GT', 'GE', 'LT' oder 'LE' für den Von- oder den Bis-Wert gekennzeichnet, wobei die Felder für den jeweils nicht benutzten Wert leer sein müssen.
  - Ein geschlossener Eingabebereich wird entsprechend durch die Operatoren 'GT', 'GE', 'LT' oder 'LE' für den Von- bzw. den Bis-Wert gekennzeichnet, wobei beide Werte bestimmt sein müssen.
  - Folgen dem Intervall-Wert in der Tabelle weitere Einzelwerte für das Merkmal, werden diese Werte ebenfalls in die Auswahlliste der für das Merkmal generierten Property aufgenommen. Dies kann für Standard- bzw. Vorschlagswerte innerhalb des Intervalls verwendet werden.
  - Ist darüberhinaus einer der (dem Intervall-Wert folgenden) Einzelwerte als Default-Wert gekennzeichnet (Feld 6), wird der bisherige Default-Wert überschrieben. Bei einem geschlossenen Intervall könnte damit die obere Grenze als Default-Wert gesetzt werden.
  - Es können mehrere Intervall-Werte angegeben werden. Mit Hilfe von Vorbedingungen (s. Abschn. 2.15) kann dann gesteuert werden, welche Intervalle in einer bestimmten Konfiguration tatsächlich zur Anzeige kommen.
- Im Feld 12 kann bei numerischen Merkmalen eine Schrittweite angegeben werden, die bei der Eingabe innerhalb des durch die Felder 8 bis 11 bestimmten Eingabebereiches eingehalten werden muß.

---

<sup>23</sup>Dabei ist zu beachten, daß der Wert nicht mit einer Vorbedingung versehen ist, die in der initialen Konfiguration des Artikels nicht gültig ist. Ansonsten wird das Merkmal in der initialen Konfiguration letztendlich mit einem anderen Wert oder gar nicht vorbelegt.



- Numerische Werte müssen in den Feldern 9, 11 und 12 gemäß der in der Merkmalstabelle (s. Abschn. 2.9) für das Merkmal spezifizierten Anzahl von Stellen bzw. Nachkommastellen angegeben werden. Führende Nullen bzw. Nullen am Ende des Dezimalteils müssen dabei nicht angegeben werden. Beispiele:
  - Format: 4 Stellen, davon 0 Nachkomma; Wert: 1200 → '1200'
  - Format: 3 Stellen, davon 1 Nachkomma; Wert: 1.5 → '1.5'
  - Format: 4 Stellen, davon 2 Nachkomma; Wert: 20.7 → '20.70' oder '20.7'
- Falls in der OCD-Datenbank mehrere Preislisten abgebildet sind und der Wert bzw. das Intervall nur in einer Preisliste gültig ist, muß in den Feldern 13 und 14 der Gültigkeitszeitraum angegeben werden<sup>24</sup>. Sind die Felder leer, ist der Wert unbegrenzt gültig.

## 2.14 Die Merkmalswert-Identifikationstabelle

Tabellenname: `PropValueIdentification`

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	PropertyClass	X	Char		X	Bezeichner der Merkmalsklasse
2.	PropertyName	X	Char		X	Bezeichner des Merkmals
3.	PropertyValue	X	Char		X	Merkmalswert
4.	IdentKey		Char		X	Schlüssel für Identifikationstabelle

Anmerkungen:

- Die Tabelle dient zur Angabe von zusätzlichen Identifikationsnummern (unterschiedlichen Typs) für Merkmalswerte.
- Die zusätzlichen Identifikationsnummern werden dabei nicht direkt in dieser Tabelle, sondern indirekt über den im Feld 4 angegebenen Schlüssel in der Tabelle `Identification` (Abschn. 2.22) abgelegt.
- Zusätzliche Identifikationsnummern können nur für Werte angegeben werden, die auch in der Merkmalswerttabelle (s. Abschn. 2.13) angelegt sind und die keine Interval-Werte sind<sup>25</sup>.

<sup>24</sup>Dieser sollte mit dem Gültigkeitszeitraum der Preiskomponenten für die entsprechende Preisliste in der Preistabelle übereinstimmen, s.a. Abschn. 3).

<sup>25</sup>Frei durch den Nutzer eingegebene Werte — bei Merkmalen, die dies erlauben — müssen bei einem entsprechenden Export, z.B. zum Bestelldatenaustausch, direkt angegeben werden.



## 2.15 Die Beziehungsobjekt-Tabelle

Tabellenname: RelationObj

Pflichttabelle: ja<sup>26</sup>

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	RelObjID	X	Num		X	Beziehungsobjekt-Nummer (größer 0)
2.	Position	X	Num		X	Position der Beziehung
3.	RelName		Char		X	Beziehungsname
4.	Type		Char	1	X	Art der Beziehung: 1 - Vorbedingung 2 - Auswahlbedingung 3 - Aktion 4 - Constraint 5 - Reaktion 6 - Post-Reaktion
5.	Domain		Char	4	X	Verwendungsgebiet: C - Konfiguration P - Preisermittlung BOI - Stückliste (bill of items) PCKG - Verpackung TAX - Besteuerungsschemata

Anmerkungen:

- In dieser Tabelle werden Beziehungen zu Beziehungsobjekten gebündelt.
- Aktuell sind folgende Beziehungsarten möglich:
  - Mittels *Vorbedingungen* kann generell die Gültigkeit der Entitäten festgelegt werden, für die die Vorbedingungen angelegt sind. Vorbedingungen sind für folgende Entitäten möglich:
    - \* Die Vorbedingung eines *Merkmals* legt fest, ob das Merkmal für den Anwender sichtbar ist (Scope RV) bzw. durch den Anwender bewertet werden darf (Scope C).
    - \* Die Vorbedingung einer *Merkmalsklasse* legt fest, ob die Merkmale der Klasse für den Anwender sichtbar sind (Scope RV) bzw. durch den Anwender bewertet werden dürfen (Scope C).  
Vorbedingungen von Merkmalsklassen haben Vorrang vor evtl. vorhandenen Vorbedingungen der Merkmale der Klasse, d.h., die Vorbedingungen der Merkmale der Klasse werden gar nicht erst ausgewertet, wenn die Klasse aufgrund ihrer Vorbedingung(en) aktuell nicht gültig ist.
    - \* Die Vorbedingung eines *Merkmalswertes* legt fest, ob dieser Wert gesetzt werden darf.  
Vorbedingungen von Werten von konfigurierbaren Merkmalen werden ausgewertet, wenn das Merkmal selber gültig ist.  
Vorbedingungen von Werten von nicht-konfigurierbaren Merkmalen werden ausgewertet, wenn dem Merkmal in der Stammdatentabelle kein Wert zugewiesen ist.
    - \* Die Vorbedingung einer *Stücklistenkomponente* legt fest, ob diese verwendet werden darf.
  - Sind für eine Entität mehrere Vorbedingungen angegeben, so ist die Entität nur dann gültig, wenn alle Bedingungen erfüllt sind. Sind für eine gegebene Entität keine Vorbedingungen angegeben, so ist diese Entität generell gültig.
  - *Auswahlbedingungen* legen fest, ob ein Merkmal bewertet werden muß. Auswahlbedingungen werden bei der Konsistenzprüfung während der Bestelllistengenerierung für aktuell nicht bewertete optionale Merkmale sowie für Merkmale zur freien Zeichenketteneingabe mit aktuell zugewiesener leerer Zeichenkette ausgewertet. Ist eine der ausgewerteten Auswahlbedingungen

<sup>26</sup>Die Tabelle kann entfallen, wenn kein Beziehungswissen benötigt wird.

erfüllt, erscheint eine entsprechende Fehlermeldung und die Bestellistengenerierung wird abgebrochen.

Sind für ein relevantes Merkmal mehrere Auswahlbedingungen angegeben, so muß es bewertet werden, wenn mindestens eine der Auswahlbedingungen erfüllt ist.

- *Aktionen* dienen zur Herleitung von Merkmalswerten oder zur Ausgabe von Nachrichten an den Anwender.

Aktionen an Artikeln und Merkmalsklassen werden bei jedem Konfigurationsschritt angewendet. Aktionen an Merkmalen werden angewendet, wenn diese nicht durch eine Vorbedingung ausgeblendet sind<sup>27</sup>. Aktionen an Merkmalswerten werden angewendet, wenn der Wert in der aktuellen Konfiguration des Artikels gesetzt ist.

- *Constraints* werden zur Überwachung und Sicherstellung der Konsistenz der Konfiguration von Artikeln verwendet. Dabei können auch Werte hergeleitet oder Wertemengen eingeschränkt werden. Constraints müssen an Artikel gebunden sein und werden bei jedem Konfigurationsschritt angewendet.

Nicht jede Sprache, die zur Codierung von Beziehungswissen verwendet werden kann, unterstützt auch Constraints (s.a. Abschnitte 2.16 und 2.23).

- *Reaktionen* dienen wie Aktionen zur Herleitung von Merkmalswerten oder zur Ausgabe von Nachrichten an den Anwender. Sie werden aber nicht bei jedem Konfigurationsschritt angewendet, sondern nur bei bestimmten Ereignissen. Reaktionen können nur an Artikel oder konfigurierbare Merkmale gebunden werden.

Reaktionen von Artikeln werden einmalig zu Beginn der Artikelinitialisierung vor Auswertung aller sonstigen Beziehungen angewendet.

Reaktionen von Merkmalen werden dann angewendet, wenn der Wert des Merkmals durch den Anwender geändert wurde. Die Anwendung erfolgt einmalig *vor* Auswertung aller sonstigen Beziehungen.

- *Post-Reaktionen* von Artikeln oder konfigurierbaren Merkmalen dienen demselben Zweck wie *Reaktionen*, werden im Gegensatz zu diesen aber *nach* Auswertung aller sonstigen Beziehungen angewendet.

Da nach Post-Reaktionen keine weiteren Beziehungen ausgewertet werden, dürfen in Post-Reaktionen keine Änderungen an der Konfiguration des Artikels vorgenommen werden, die Auswirkungen auf abhängige Merkmale haben! Ansonsten kann die Artikelkonfiguration in einem inkonsistenten Zustand verbleiben.

- Die Nummer in Feld 2 bestimmt die Position der Beziehung in der Auswertungsreihenfolge. Die Beziehungen eines Beziehungsobjektes mit demselben Typ (Feld 4) und demselben Verwendungsgebiet (Feld 5) werden in aufsteigender Reihenfolge der Positionsnummern ausgewertet, wobei die Nummernreihenfolge nicht kontinuierlich sein muß.  
Beziehungen mit gleicher Positionsnummer werden in undefinierter Reihenfolge ausgewertet.

- Das Verwendungsgebiet (Feld 5) gibt an, in welchem Kontext die Beziehung anzuwenden ist:

**C** Diese Beziehungen werden bei der Definition der Konfigurationsmöglichkeiten eines konfigurierbaren Artikels während seiner initialen Erzeugung als auch bei jedem Konfigurationsschritt ausgewertet.

**P** Diese Beziehungen werden bei der Preisermittlung ausgewertet (s.Abschn. 3).

**BOI** Beziehungen dieses Verwendungsgebiets steuern die Sichtbarkeit (Existenz) einer Stücklistenkomponente (s.Abschn. 2.7).

**PCKG** Diese Beziehungen werden bei der Ermittlung von Verpackungsdaten ausgewertet (s.Abschn. 6).

**TAX** Diese Beziehungen werden bei der Ermittlung der Besteuerungsinformationen zu einem Artikel auf der Basis von Besteuerungsschemata ausgewertet (s.Abschn. 2.25). Sie werden benötigt, wenn die Zuordnung eines Artikels zu einer Steuerkategorie von einer bestimmten Konfiguration des Artikels abhängig ist<sup>28</sup>.

<sup>27</sup> gilt also generell für Merkmale der Scopes R und RG

<sup>28</sup> Ein Beispiel wäre die Änderung der Materialkategorie in der Steuerart ECO\_FR.

Preis-, PCKG- und TAX-Beziehungen dürfen nur vom Typ *Aktion* sein.

Für Preis-, PCKG- und TAX-Beziehungen ist auch noch folgendes zu beachten:

Diese Beziehungen dürfen keinen Einfluss auf die aktuelle Konfiguration bzw. auf nachfolgende Auswertungen von Konfigurationsbeziehungen haben. Deswegen sind in diesen Beziehungen Zuweisungen nur an interne (Hilfs-)Merkmale (Geltungsbereich R) erlaubt<sup>29</sup>.

Die nachfolgende Tabelle gibt einen Überblick, für welche Daten-Entitäten und in welchen Verwendungsgebieten die einzelnen Beziehungsarten angewendet werden können:

Beziehungsart	Daten-Entität					Verwendungsgebiet				
	Artikel	M.Klasse	Merkmal	M.Wert	BOI-Part	C	P	PCKG	TAX	BOI
Vorbedingung		X	X	X	X	X				X
Auswahlbed.			X			X				
Aktion	X	X	X	X		X	X	X	X	
Reaktion	X		X			X				
Post-Reaktion	X		X			X				
Constraint	X					X				

## 2.16 Die Beziehungswissen-Tabelle

Tabellenname: *Relation*

Pflichttabelle: ja<sup>30</sup>

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	RelationName	X	Char		X	Beziehungsname
2.	BlockNr	X	Num		X	Codeblock-Nummer
3.	CodeBlock		Char		X	Codeblock

Anmerkungen:

- In dieser Tabelle wird das „Wissen“ (die Logik) über die Beziehungen abgelegt. Dazu wird eine verschiedene Sprachen verwendet werden, deren Syntax und Semantik im Anhang beschrieben ist. Welche Sprache verwendet wird, muß in der Versionsinformationstabelle (Abschn. 2.23) angegeben werden.
- Die zu einer Beziehung gehörenden Codeblöcke werden vor der Auswertung entsprechend ihrer Nummer zu einem ganzen Codeblock zusammengesetzt.

<sup>29</sup>s.a. Anmerkung zum Einfluss des Geltungsbereichs auf Persistenz und Initialisierung von Merkmalen in Abschn. 2.9

<sup>30</sup>Die Tabelle kann entfallen, wenn kein Beziehungswissen benötigt wird.

## 2.17 Die Preistabelle

Tabellenname: Price

Pflichttabelle: ja<sup>31</sup>

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ArticleID	X	Char		X	(Grund)Artikelnummer
2.	Variantcondition	X	Char			Variantenkondition
3.	Type	X	Char	1	(X)	Preisart: S - Verkaufspreis (sales) P - Einkaufspreis (purchase)
4.	Level	X	Char	1	X	Preisebene: B - Grundpreis (Base) X - Zuschlagspreis (eXtra charge) D - Rabatt (Discount)
5.	Rule		Char		(X)	Rechenregel
6.	TextID		Char			Textnummer
7.	PriceValue		Num		X	Preis/Betrag
8.	FixValue		Bool	1	X	Festbetrag (vs. Prozentangabe) ?
9.	Currency	X	Char	3	(X)	Währung für Festbetrag
10.	DateFrom	X	Date	8	X	Gültig von
11.	DateTo		Date	8	X	Gültig bis
12.	ScaleQuantity	X	Num		X	Staffelpreis-Mindestmenge
13.	RoundingID		Char			Identifikator für Rundungsvorschrift

Anmerkungen:

- In dieser Tabelle werden zu jedem Artikel die Grund- und die Zuschlagspreise als auch mögliche Rabatte verzeichnet.
- Alle Preise werden als Netto-Preise, also ohne Steuern angegeben! Der Vorgang der Preisermittlung ist in Abschn. 3 genauer beschrieben.
- Ist für einen Preisposten eine *Variantenkondition* angegeben (Feld 2), so wird dieser Preisposten bei der Preisermittlung nur dann berücksichtigt, wenn die angegebene Variantenkondition gültig ist. Die für eine bestimmte Konfiguration gültigen Variantenkonditionen werden durch Preisbeziehungen (aus den Tabellen *RelationObj* und *Relation*) ermittelt.  
Variantenkonditionen müssen komplett groß geschrieben sein (ausgenommen das deutsche Eszett 'ß').
- Für jeden Preisposten kann sowohl ein Verkaufspreis als auch ein Einkaufspreis angegeben werden (Feld 3).
- Bei Einträgen für Zuschläge und Rabatte (Preisebenen 'X' und 'D', Feld 4) kann der Joker-Artikel „\*“ (Feld 1) zur Angabe artikelübergreifender Zuschläge bzw. Rabatte verwendet werden. Das Feld 2 (Variantenkondition) darf dabei nicht leer sein.  
Dieser artikel-neutrale Tabelleneintrag wird jedoch nur dann berücksichtigt, wenn für den bearbeiteten Artikel kein eigener, spezifischer Eintrag mit der gleichnamigen Variantenkondition existiert.
- Die Rechenregel (Feld 5) modifiziert die Art und Weise der Verwendung des Preispostens bei der Preisermittlung:
  - Für Grund- und Zuschlagspreise werden aktuell keine speziellen Rechenregeln unterstützt. Der im Feld 7 angegebene Betrag wird bei der Preisermittlung immer dem bereits akkumulierten Preis hinzugefügt. Zuschlagspreise können auch als prozentuale Werte angegeben werden. In

<sup>31</sup>Die Tabelle kann entfallen, wenn keine Preise angelegt werden sollen.

diesem Fall ergibt sich der absolute Betrag aus dem betreffenden prozentualen Anteil des Grundpreises.

- Bei Rabatten mit Prozentangabe muss eine Rechenregel angegeben werden:  
Rechenregel '1' definiert, dass der Preis in Bezug auf den Grundpreis berechnet wird.  
Rechenregel '2' gibt an, dass der Preis in Bezug auf den während der Preisermittlung bereits akkumulierten Preis berechnet wird.

- Über die Textnummer (Feld 6) kann in der Preistext-Tabelle eine Erklärung zu dem Preiseintrag hinterlegt werden, z.B. Grund für Aufpreis. Ist für einen Preiseintrag keine Beschreibung hinterlegt, wird vom Anwendungssystem ggf. eine automatische Beschreibung generiert.
- Im Feld 8 wird spezifiziert, ob der Betrag in Feld 7 einen festen Betrag in der Währung gemäß Feld 9 darstellt (1) oder eine Prozentangabe ist (0).
- Bei Zuschlägen (Preisebene 'X') kann der Betrag in Feld 7 auch negativ sein. Dies kann zur Abbildung von Abschlägen (Minderpreisen) genutzt werden.
- Währungen (Feld 9) sind gemäß ISO 4217 anzugeben, z. B. EUR, CHF, GBP, USD.
- Feld 12 dient zur Angabe von Staffelpreisen, die ab einer bestimmten Menge von bestellten Artikeln wirksam werden. In dem Feld 12 wird dazu die Anzahl der Artikel angegeben, ab der der Tabelleneintrag verwendet werden kann. Standardmäßig ist hier 1 einzutragen (kein Staffelpreis).  
Achtung: Staffelpreise werden aktuell immer nur pro Bestell- bzw. Auftragsposition berechnet, nicht über die gesamte Bestellung hinweg.
- Im Feld 13 kann auf eine Rundungsvorschrift in der Tabelle **Rounding** (s. nächster Abschn.) verwiesen werden. Diese wird dann anstelle der im Abschn. 3 beschriebenen Standard-Rundung verwendet, bevor der absolute Betrag der Preiskomponente zu dem (aktuellen) Gesamtpreis dazu addiert wird.

## 2.18 Die Rundungsvorschrift-Tabelle

Tabellenname: **Rounding**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ID	X	Char		X	Identifikator der Rundungsvorschrift
2.	Number	X	Num		X	laufende Nummer
3.	Minimum		Char			kleinster Betrag, der nach dieser Methode gerundet werden soll
4.	Maximum		Char			Betrag, ab dem nicht mehr nach dieser Methode gerundet werden soll
5.	Type		Char		X	Rundungsmethode (s.u.)
6.	Precision		Num		X	Genauigkeit der Rundung
7.	AddBefore		Num		X	Betrag, der vor der Rundung aufgeschlagen werden soll
8.	AddAfter		Num		X	Betrag, der nach der Rundung aufgeschlagen werden soll

Anmerkungen:

- Unter einer Rundungsvorschrift-ID können mehrere Einträge angelegt werden. Diese werden in der durch die laufende Nummer im Feld 2 bestimmten Reihenfolge abgearbeitet.
- Die Felder 3 (Minimum) und 4 (Maximum) bestimmen den Betragsbereich, für den die Rundung nach der im Feld 5 angegebenen Methode erfolgen soll, d.h. Beträge ausserhalb des durch die Felder 3 und 4 bestimmten Bereichs werden nicht gerundet. Der im Feld 4 angegebene maximale Betrag

fließt dabei nicht mehr in den Bereich ein<sup>32</sup>.

Die Felder sind als optionale Zeichenkettenfelder deklariert, sie können also auch leer sein. Wenn nicht leer, enthalten die Felder String-Darstellungen numerischer Werte<sup>33</sup>.

Ein leeres Feld 3 bedeutet einen nach unten offenen Bereich, ein leeres Feld 4 einen nach oben offenen Bereich.

- Da unter einer ID mehrere Einträge angelegt werden können, ist es möglich, mehrere Betragsbereiche abzubilden. Werden dabei für dieselbe Rundungsvorschrift-ID mehrere sich überschneidende Betragsbereiche angegeben, finden u.U. mehrere Rundungen statt (wenn die zu rundenden Beträge in dem sich überschneidenden Bereich liegen). Dann ist die durch Feld 2 festgelegte Abarbeitungsreihenfolge entscheidend, da das Ergebnis einer Rundung jeweils den Eingangsbetrag für die darauffolgende Rundung bildet.
- Die erlaubten Rundungsmethoden sind:
  - DOWN** Abrunden
  - UP** Aufrunden
  - COM** kaufmännisches Runden (X.5 rundet auf)
  - ECOM** erweitertes (unverzerrtes) kaufmännisches Runden (X.5 rundet auf gerade Zahlen ab oder auf)
- Die Genauigkeit in Feld 6 bestimmt den Betrag, durch den der gerundete Betrag ohne Rest teilbar sein muß. 0.01 z.B. rundet auf die zweite Stelle nach dem Dezimalpunkt (und entspricht der Genauigkeit bei der Standard-Rundung).
- Die Beträge in den Feldern 7 und 8 können auch negativ sein.

Beispiel:

Angenommen, eine Preiskomponente soll abhängig vom nominalen Betrag wie folgt gerundet werden:

- Betrag < 10 EUR, auf 0.1 EUR kaufmännisch runden
- Betrag >= 10 EUR bis < 100 EUR auf 0.5 EUR kaufmännisch runden
- Betrag >= 100 EUR auf 1.0 EUR aufrunden und 1 Cent abziehen.

Die Tabelle müßte dann folgende Einträge für die Rundungsvorschrift enthalten:

ID	Nr.	Minimum	Maximum	Type	Precision	AddBefore	AddAfter
R1	1	0.0	10.0	COM	0.1	0.0	0.0
R1	2	10.0	100.0	COM	0.5	0.0	0.0
R1	3	100.0		UP	1.0	0.0	-0.01

## 2.19 Die Serientabelle

Tabellenname: **Series**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	SeriesID	X	Char		X	Serienkürzel
2.	TextID		Char		X	Textnummer
3.	CatalogFormat		Char			Format der Katalogdaten
4.	CatalogDir		Char			Verzeichnis der Katalogdaten

<sup>32</sup>d.h., es wird der Vergleichsoperator LT (*kleiner als*) angewendet

<sup>33</sup>äquivalent zu den Werten numerischer Merkmale in der Wertetabelle

Anmerkungen:

- Diese Tabelle unterstützt Applikationen, die nicht das Datenregistrierungssystem *DSR* der Firma EasternGraphics verwenden, bei der Registrierung von kaufmännischen Serien.
- Die Textnummer im Feld 2 dient als Schlüssel für die Beschreibungstabelle **SeriesText** (s. Abschn. 2.20). Der sprachspezifische Text zu einer Serie enthält in der ersten Zeile die Serienbezeichnung. Optional können weitere Zeilen mit zusätzlichen Erklärungen folgen.
- Im Feld 4 kann ein Verzeichnis angegeben werden, daß die Daten enthält, die durch die Benutzeroberfläche zur Artikelauswahl verwendet werden können. Das dabei verwendete Datenformat muss dann im Feld 3 spezifiziert werden. Die Formatangabe wird gebildet aus der Kurzbezeichnung des Formats (s.u.), gefolgt von einem Bindestrich ('-') und der Versionsangabe, die sich aus Hauptnummer, Punkt ('.') und Unternummer zusammensetzt.
- Folgende Formate können im Feld 3 angegeben werden:

Kurzbezeichnung	Erklärung
OAS	OFML Article Selection (OFML Part V)
XCF	eXtensible Catalog Format (Firma EasternGraphics)

## 2.20 Die Beschreibungstabellen

Alle Texttabellen

Artikelkurzbeschreibungen: **ArtShortText** (Pflichttabelle)

Artikellangbeschreibungen: **ArtLongText**

Merkmalsklassenbezeichnungen: **PropClassText**

Merkmalsbezeichnungen: **PropertyText** (Pflichttabelle)

Hinweise zu Merkmalen: **PropHintText**

Merkmalsgruppenbezeichnungen: **PropGroupText**

Merkmalswertbezeichnungen: **PropValueText**

Preistexte (Erklärungen zu Preiskomponenten): **PriceText**

Serienbezeichnungen: **SeriesText**

Zusatztexte für Stücklistenpositionen: **BillOfItemsText**

Nachrichten für den Anwender: **UserMessage**

Bezeichnungen für Klassifikationen: **ClassificationText**

besitzen den gleichen Aufbau:

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	TextID	X	Char		X	Textnummer
2.	Language	X	Char	2	X	Sprache
3.	LineNr	X	Num		X	Zeilennummer
4.	LineFormat		Char		X	Zeilenformatsteuerung
5.	Textline		Char		X	Textzeile

Anmerkungen:

- Der Zugriff erfolgt über Textnummern, die in den jeweiligen Tabellen vergeben werden.
- Die Sprache ist gemäß ISO 639-1 anzugeben, z.B. 'de' (Deutsch), 'en' (Englisch), 'fr' (Französisch)<sup>34</sup>.

<sup>34</sup>Im Sinne einer Durchgängigkeit über alle OFML-Daten hinweg wurde hier die Entscheidung für den 2-stelligen Sprachcode getroffen. Beim Export in ein Format, das eine 3-stellige Kodierung nach ISO 639-2 verwendet, z.B. BMEcat, ist die jeweilige Applikation dafür verantwortlich, eine entsprechende Umwandlung durchzuführen.

- Ein Text besteht aus ein oder mehreren Zeilen. Mehrere Zeilen sind jedoch nur bei den Tabellen `ArtLongText`, `PropHintText`, `UserMessage` sowie `PropValueText` erlaubt. (Bei anderen Texttypen werden zusätzliche Zeilen nicht verarbeitet.)  
Bei der Tabelle `PropValueText` erfolgt die Verarbeitung der Zeilen im Zusammenspiel mit dem Feld `TxtControl` der Merkmalstabelle (s. Abschn 2.9).
- Bei der Gestaltung der Texte sollten die Empfehlungen des OFML–Normungsausschusses zur einheitlichen Gestaltung kundenorientierter Artikelbeschreibungen berücksichtigt werden.
- Das Feld 4 enthält Formatierungshinweise für Formularlayout–Module eines OFML–basierten Vertriebssystems. Das Erscheinungsbild des Textes kann damit in einem gewissen Rahmen beeinflusst werden.

Folgende Formatierungs-codes sind möglich:

\ (Zeilenvorschub)

Die Textzeile wird in einer neuen Zeile ausgegeben. Dies ist der Standard.

Bei einem Merkmalstext-Steuercode 0 wird dieser Code in der ersten Zeile des Merkmalswert-Textes ignoriert und stattdessen der Code ~ (s.u.) verwendet.

~ (Fließtext)

Die Textzeile wird als Fließtext an den vorherigen Text gehängt. Beginnt die Textzeile selber nicht mit einem Leerzeichen, wird dieses vom Formularlayout–Modul eingefügt.

^ (Bedingter Fließtext)

Findet bei Textbausteinen Anwendung, die sich aus verschiedenen Textarten zusammensetzen, z.B. Merkmalsbezeichner plus Wertbezeichner bei Merkmalstext-Steuercode 0: Passt die Textzeile (inkl. der ggf. einzufügenden Trennzeichen) noch in den zusammengesetzten Textbaustein, ohne die vom Formularlayout–Modul vorgegebene Zeilenbreite zu überschreiten, wird sie als Fließtext an den vorherigen Text gehängt. Anderenfalls erfolgt ein Zeilenvorschub zu Beginn der Textzeile.

## 2.21 Wertkombinationstabellen

Verschiedene Sprachen zur Codierung von Beziehungswissen (s. Abschn. 2.16) erlauben die Verwendung von Wertkombinationstabellen. Wertkombinationstabellen werden in Beziehungswissen dazu genutzt, die Konsistenz einer Wertkombination zu prüfen, Werte herzuleiten oder den Wertebereich eines Merkmals einzuschränken.

In einer Wertkombinationstabelle werden alle möglichen Wertkombinationen über eine definierte Menge von Merkmalen hinweg angegeben.

Bsp.:

	AUSF_GRUPPE	FARBE_KORPUS
1	A	F001
2	A	F002
3	B	F002
4	B	F003

Der Dateiname einer Wertkombinationstabelle wird aus dem Bezeichner, unter dem die Tabelle in Beziehungswissen angesprochen wird, dem Nachsatz „\_tbl“ und dem Suffix „.csv“ gebildet, wobei der Tabellename komplett klein geschrieben wird.

Bsp.:

Tabellenbezeichner: `FARBEN_KORPUS`

Dateiname: `farben_korpus_tbl.csv`



Die Tabellendefinition ist für alle OCD-Wertkombinationstabellen gleich:

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	LineNr	X	Num		X	Nummer der Tabellenzeile
2.	PropertyName	X	Char		X	Name des Merkmals
3.	Value	X	Char		X	Merkmalswert

Die Felder einer Zeile der logischen Wertkombinationstabelle (s. Beispiel oben) werden also über ihre (imaginäre) Zeilennummer zusammengefasst<sup>35</sup>.

Sowohl Merkmalsnamen als auch Merkmalswerte müssen komplett groß geschrieben sein (ausgenommen das deutsche Eszett 'ß').

Bsp.:

Die (logische) Wertkombinationstabelle aus dem Beispiel oben würde wie folgt in einer OCD-Wertkombinationstabelle abgebildet werden:

```
1;AUSF_GRUPPE;A
1;FARBE_KORPUS;F001
2;AUSF_GRUPPE;A
2;FARBE_KORPUS;F002
3;AUSF_GRUPPE;B
3;FARBE_KORPUS;F002
4;AUSF_GRUPPE;B
4;FARBE_KORPUS;F003
```

Enthält die Wertkombinationstabelle ein einzelnes Merkmal, das sich auf ein einschränkbares Produktmerkmal bezieht (s. Abschn. 2.9), so kann in dem Feld einer Tabellenzeile für dieses Merkmal auch eine Wertemenge angegeben werden.

Bsp.:

Angenommen, das Merkmal **FARBEN\_KORPUS** aus dem obigen Beispiel ist einschränkbar, so könnten logische Wertkombinationstabelle und entsprechende OCD-Wertkombinationstabelle so aussehen:

	AUSF_GRUPPE	FARBE_KORPUS
1	A	F001, F002
2	B	F002, F003

```
1;AUSF_GRUPPE;A
1;FARBE_KORPUS;F001
1;FARBE_KORPUS;F002
2;AUSF_GRUPPE;B
2;FARBE_KORPUS;F002
2;FARBE_KORPUS;F003
```

## 2.22 Die Identifikationstabelle

Tabellenname: **Identification**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	EntityID	X	Char		X	ID der Entität (Artikel, Merkmal, Wert)
2.	Type	X	Char		X	Typ der Identifikationsnummer
3.	IdentNr		Char		X	Identifikationsnummer

<sup>35</sup>Damit muß nicht für jede logische Wertkombinationstabelle eine eigene Tabellendefinition bereitgestellt werden.

Anmerkungen:

- Die Tabelle dient zur Angabe von zusätzlichen Identifikationsnummern für Artikel, Merkmale und Merkmalswerte.
- Die Art und Weise bzw. der Kontext der Verwendung einer Identifikationsnummer wird durch ihren Typ (Feld 2) bestimmt. Aktuell sind folgende Typen erlaubt (s.a. Begriffe im Anhang)<sup>36</sup>:

Typ	Erklärung
CustomID	Händler- bzw. großkundenspezifische Artikelnummer
EAN.UCC-8	achtstellige ID nach EAN.UCC
EAN.UCC-13	dreizehnstellige ID nach EAN.UCC
EAN.UCC-14	vierzehnstellige ID nach EAN.UCC
GLN	Globale Lokationsnummer
Intrastat	Intrastat-Nummer
CustomsTarif	Zolltarifnummer

Der Typ *CustomID* wird benutzt, wenn der Datenbestand für einen Fachhändler bzw. Großkunden bestimmt ist, der vom Hersteller abweichende Artikelnummern verwendet. Von der OFML-Anwendung ist dann ggf. diese kundenspezifische Artikelnummer anzuzeigen/zu verwenden.

## 2.23 Die Versionsinformationstabelle

Tabellenname: *Version*

Pflichttabelle: ja

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	FormatVersion		Char		X	Nummer der verwendeten OCD-Formatversion
2.	RelCoding		Char		X	verwendete Sprache für Beziehungswissen
3.	DataVersion		Char		X	Datenbank-Version
4.	DateFrom		Date	8	X	Verwendbarkeitsdatum von
5.	DateTo		Date	8	X	Verwendbarkeitsdatum bis
6.	Region		Char		X	Vertriebsgebiet
7.	VarCondVar		Char			Variable für Variantenbedingungen
8.	PlaceholderOn		Bool		X	Platzhalter in IN-Vergleichen?
9.	Tables		Char		X	enthaltene Tabellen
10.	Comment		Char			freie Kommentare, Zusatzinformationen

Anmerkungen:

- Die Tabelle dient zur Angabe von Informationen über das verwendete Format und über die Produktdatenbank. Damit können durch ein Versionskontrollsystem o.ä. Aussagen über den Aufbau und die Verwendbarkeit der Datenbank getroffen werden. Die Tabelle darf nur einen Eintrag enthalten.
- Die OCD-Formatversion (Feld 1) ist in der Form *MajorNumber.MinorNumber* gemäß OCD-Formatspezifikation anzugeben.

<sup>36</sup>Nicht alle der ID-Typen sind auch bei jedem Entitätstyp anwendbar.

- Im Feld 2 muß die Sprache spezifiziert werden, die zur Codierung von Beziehungswissen verwendet wird (s. Abschn. 2.16). Folgende Sprachen können verwendet werden: `OCD_1`, `OCD_2`, `OCD_3`, `OCD_4`, `SAP_LOVC`. Die Beschreibung der Sprachen ist im Anhang enthalten.
- Die Datenbank-Version (Feld 3) ist in der Form `MajorNumber.MinorNumber.BuildNumber` anzugeben. Die Nummern können vom Hersteller frei vergeben werden, sind dabei aber streng monoton wachsend zu vergeben.
- Der Bezeichner für das Vertriebsgebiet<sup>37</sup> (Feld 6) kann frei vergeben werden. Er muss jedoch mit dem Bezeichner korrespondieren, mit dem im jeweiligen Software-System weitere benötigte Daten (Geometrie, Katalog) zu dem Vertriebsgebiet referenziert werden.
- Im Feld 7 kann die Variable spezifiziert werden, die in Preisbeziehungen anstelle von `$VARCOND` (OCD-Sprachsets) bzw. von `$self.variant_condition` (SAP-Sprachsets) zur Zuweisung von Variantenbedingungen verwendet wird (s.a. 3).  
Für den Bezeichner der Variablen sind alle alphanumerischen Zeichen inklusive dem Unterstrich erlaubt, wobei das erste Zeichen kein numerisches sein darf. Bei der Verwendung der Variablen innerhalb von Beziehungen in der Tabelle `Relation` muß dem Bezeichner das Dollar-Zeichen ('\$') vorangestellt werden.
- Im Feld 8 wird angegeben, ob im Beziehungswissen in IN-Vergleichen Platzhalterzeichen in Zeichenkettenkonstanten ersetzt werden sollen (s. Anhang D).
- Im Feld 9 werden, durch ein Komma separiert, die aktuell in der Datenbank enthaltenen Tabellen aufgelistet. Dies betrifft auch die Pflichttabellen, nicht aber Wertkombinationstabellen. Dabei sind die Tabellennamen gemäß der Spezifikation der verwendeten OCD-Formatversion (Feld 1) anzugeben<sup>38</sup>.  
Zusätzliche Leerzeichen nach den Kommata sind erlaubt.

## 2.24 Die Nummernschema-Tabelle

Tabellenname: `CodeScheme`

Pflichttabelle: nein

Anmerkungen:

- Die Tabelle dient zur Angabe von Kodierungsschemata und Parametern für die Generierung von Endartikelnummern.
- Endartikelnummern werden konzeptionell aus der Grundartikelnummer und dem sogenannten *Varietencode* zusammengesetzt, wobei die konkrete Stellung von Grundartikelnummer und Varietencode in der Endartikelnummer durch die einzelnen Kodierungsschemata bestimmt wird. Im Varietencode werden die aktuellen Ausprägungen der konfigurierbaren Merkmale kodiert.
- Das für einen Artikel zu verwendende Kodierungsschema wird in der Artikeltabelle (Abschn. 2.2) anhand des Schema-Identifikators bestimmt<sup>39</sup>.
- Es wird zwischen *vordefinierten* und *nutzerdefinierten* Kodierungsschemata unterschieden. Die jeweiligen Kodierungsvorgänge sind im Abschn. 4 genau beschrieben.
- Die Felder 3 bis 10 dienen der Parametrisierung der Kodierungsschemata wie im Abschn. 4 beschrieben.
- Die Zeichenkette in Feld 3 darf (bei vordefinierten Schemata) nicht leer sein. Ist das Feld 3 dennoch leer, wird eine Zeichenkette bestehend aus einem einzelnen Leerzeichen verwendet.

<sup>37</sup>Großkunden mit spezifischen Preislisten oder abweichenden Konfigurationsdaten werden in diesem Kontext ebenfalls durch das Konzept des *Vertriebsgebiets* abgebildet.

<sup>38</sup>d.h., ohne Präfix `ocd_` und ohne Suffix `.csv`.

<sup>39</sup>Damit kann prinzipiell für jeden Artikel ein eigenes Kodierungsschema definiert werden.

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	SchemeID	X	Char		X	eindeutiger Identifikator zur Referenzierung des Schemas
2.	Scheme		Char			Beschreibung des Schemas
3.	VarCodeSep		Char			Zeichenkette zum Trennen von Grundartikelnummer und Variantencode (nur für vordefinierte Schemata)
4.	ValueSep		Char			Zeichenkette zum Trennen von Merkmalswerten (nur für vordefinierte Schemata)
5.	Visibility		Char	1		Sichtbarkeitsmodus – gibt an, welche Merkmale im Variantencode enthalten sein sollen 0 – nur die aktuell gültigen und sichtbaren Merkmale 1 – alle konfigurierbaren Merkmale
6.	InVisibleChar		Char	1		Ersetzungszeichen für aktuell ungültige bzw. nicht sichtbare Merkmale: Es werden so viele Zeichen dargestellt, wie im Längensfeld der Property-Tabelle für das Merkmal angegeben. Ist das Feld leer, wird '-' verwendet.
7.	UnselectChar		Char	1		Ersetzungszeichen für aktuell nicht bewertete/ausgewählte optionale bzw. einschränkbare Merkmale: Es werden so viele Zeichen dargestellt, wie im Längensfeld der Property-Tabelle für das Merkmal angegeben. Ist das Feld leer, wird 'X' verwendet.
8.	Trim		Bool	1	X	Trimm-Kennzeichen – gibt an, ob die einzelnen Merkmalswerte exakt gemäß der Angabe im Längensfeld der Property-Tabelle dargestellt werden sollen (0), oder ob nicht belegte Stellen (Leerzeichen) am Ende entfernt werden können (1).
9.	MO_Sep		Char			Zeichenkette zum Trennen der Werte von mehrwertigen Merkmalen
10.	MO_Bracket		Char	2*N		Zeichen für Klammersetzung bei mehrwertigen Merkmalen

- Damit eine gegebene Artikelkonfiguration anhand ihrer Endartikelnummer eindeutig rekonstruiert werden kann, darf in den Feldern 6 und 7 kein Ersetzungszeichen verwendet werden, das zu einem regulären Wert von optionalen oder einschränkbaren Merkmalen führen würde.
- Wenn das Trimm-Kennzeichen in Feld 8 gesetzt ist, müssen die Inhalte in den Feldern 6 und 7 von dem Inhalt in Feld 4 verschieden sein und Feld 4 darf nicht leer sein.
- Das Trimm-Kennzeichen darf nicht gesetzt sein, wenn der Artikel (codierte) Merkmale besitzt, für die der Anwender Werte frei eingeben kann. Anderenfalls kann die korrekte Verarbeitung der Endartikelnummer nicht in allen Fällen sichergestellt werden.

## 2.25 Besteuerungsschemata

Zu Artikeln werden in den OCD-Daten keine konkreten Steuersätze hinterlegt, sondern jeder Artikel wird einem sogenannten Besteuerungsschema zugeordnet, welches pro Land (des im OCD-Datenstand abgebildeten Vertriebsgebietes) für jede relevante Steuerart die jeweils zu verwendende (abstrakte) Steuerkategorie beschreibt. Die aktuellen gesetzlichen Steuersätze für jede Steuerkategorie werden dann in den OFML-Anwendungen selber verwaltet und bei der Belegerstellung herangezogen.

Tabellenname: **ArticleTaxes**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ArticleID	X	Char		X	(Grund)Artikelnummer
2.	TaxID		Char		X	ID des Besteuerungsschemas
3.	DateFrom	X	Date	8		Gültig von (inklusive)
4.	DateTo		Date	8		Gültig bis (inklusive)

Anmerkungen:

- Über diese Tabelle erfolgt die Zuordnung eines Artikels zu einem Besteuerungsschema.
- Ist für einen Artikel keine Zuordnung zu einem Besteuerungsschema hinterlegt (z.B. auch, wenn die in diesem Abschnitt beschriebenen Tabellen gar nicht existieren), so wird ein im Anwendungssystem festgelegtes bzw. eingestelltes Besteuerungsschema verwendet.
- Über die Felder 3 und 4 können Stichtage abgebildet werden, an denen gesetzliche Änderungen hinsichtlich der Zuordnung von Artikeln zu Steuerkategorien wirksam werden. Sobald es mehrere Einträge zu einem Artikel in der Tabelle gibt, dürfen diese beiden Felder nicht leer sein (und müssen sich nicht überlappende Zeiträume spezifizieren).

Tabellenname: **TaxScheme**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	TaxID	X	Char		X	ID des Besteuerungsschemas
2.	Country	X	Char		X	Ländercode (ISO 3166-1)
3.	Region	X	Char			Regionscode (ISO 3166-2)
4.	Number	X	Num		X	Nummer in der Anwendungsreihenfolge
5.	TaxType		Char		X	Bezeichner der Steuerart
6.	TaxCategory		Char		X	Bezeichner der Steuerkategorie

Anmerkungen:

- Diese Tabelle definiert Länder-spezifisch und ggf. auch Regionen-spezifisch die Besteuerungsschemata, die für Artikel des OCD-Datenstandes relevant sind.
- Das Feld 3 (*Region*) kann zur Abbildung spezifischer Regelungen für Regionen innerhalb eines Landes genutzt werden. Die Einträge für ein gegebenes Besteuerungsschema und ein gegebenes Land, in denen das Feld 3 leer ist, beschreiben dabei die Regionen-übergreifende Normalregelung für das Land.  
Als Code ist in diesem Feld nicht der vollständige Code gemäß ISO 3166-2 anzugeben, sondern nur der regionsspezifische Teil nach Ländercode und Trennzeichen (z.B. also „TH“ für Thüringen anstatt von „DE-TH“).

- Die Nummer der Anwendungsreihenfolge (Feld 4) ist relevant für die Fälle, in denen mehrere Steuerarten für den Artikel erhoben werden. Die Nummer gibt dann die Reihenfolge vor, in der die jeweiligen Steuersätze angewendet werden. Die letztliche Entscheidungsgewalt bei der Anwendungsreihenfolge hat jedoch die jeweilige OFML-Anwendung, da sie ggf. gesetzlichen Regelungen den Vorrang geben muss.
- Die Bezeichner für Steuerart und Steuerkategorie (Felder 5 und 6) können im Prinzip frei vergeben werden. Dies ist jedoch nur für spezifische OFML-Anwendungen sinnvoll. Um eine einheitliche Verwendung über unterschiedliche OFML-Anwendungen hinweg zu gewährleisten, werden die Bezeichner für die gewöhnlichen Steuerarten und Steuerkategorien standardisiert. Dies erfolgt in einem separaten Anhang zur OCD-Spezifikation (s. Anh. H), der bei Bedarf nach Antrag und Abstimmung durch das OFML-Normungsgremium dynamisch erweitert werden kann.
- Ist ein Artikel keinem Besteuerungsschema zugeordnet oder enthält das Schema weder Angaben zu einer angeforderten Region noch zu einem angeforderten Land, so wird von der OFML-Anwendung in der Steuerart *Mehrwertsteuer* die Standard-Kategorie (mit dem dafür in dem Land geltenden Steuersatz) angewendet.
- Mit Hilfe der Funktion `SET_TAX_CATEGORY` (s. Anh. D) kann in TAX-Beziehungen einem Artikel in einer gegebenen Steuerart, abhängig von bestimmten Artikelvarianten, eine vom Besteuerungsschema abweichende Steuerkategorie zugeordnet werden.

## 3 Die Preisermittlung

In diesem Abschnitt wird beschrieben, wie mit Hilfe der Preistabelle und Beziehungswissen der Preis zu einem Artikel in einer konkreten Konfiguration ermittelt wird.

### 3.1 Überblick

Der Gesamtpreis (Endpreis) eines Artikels in einer Preisart (Verkaufs- vs. Einkaufspreis) setzt sich konzeptionell aus *Preiskomponenten* verschiedener Ebenen zusammen. Es wird zwischen nicht-konditionierten und konditionierten Preiskomponenten unterschieden. *Nicht-konditionierte* Preiskomponenten gelten für den Artikel unabhängig von seiner Konfiguration. *Konditionierte* Preiskomponenten hingegen gelten nur bedingt für bestimmte Konfigurationen des Artikel. Die Bedingung wird mit Hilfe einer *Variantenkondition* (Feld 2 in der Preistabelle) formuliert, welche in Preisbeziehungen entsprechend hergeleitet werden muß (s. Abschn. 3.2).

Das im weiteren beschriebene Verfahren gilt gleichermaßen für beide Preisarten.

Zu einer gegebenen Preiskomponente können in einer Preisebene mehrere Einträge vorliegen, die sich durch ihre Gültigkeitszeiträume, Währungen und/oder sonstige Attribute unterscheiden. Aus diesen Tabelleneinträgen wird anhand der Parameter der Preisermittlung der passendste Eintrag ermittelt (s. Abschn. 3.3).

Die relevanten Preiskomponenten der verschiedenen Preisebenen (Feld 4) werden in folgender Reihenfolge ermittelt:

1. Grundpreise (Ebene 'B')
2. Zuschlagspreise (Ebene 'X')
3. Rabatte (Ebene 'D')

Bei jeder ermittelten (gültigen) Preiskomponente wird der Gesamtpreis für den Artikel entsprechend der Rechenregel (Feld 5 in der Preistabelle) akkumuliert. Dabei erfolgt ggf. eine Rundung des ermittelten absoluten Betrages auf 2 Stellen nach dem Dezimalpunkt nach der Methode des kaufmännischen Rundens (X.5 rundet auf), insofern für die Preiskomponente keine eigene Rundungsvorschrift hinterlegt ist (Feld 13).

Abschließend werden die Steuern für den Artikel ermittelt. Dabei werden die Angaben aus den hinterlegten Besteuerungsschemata verwendet (s. Abschn. 2.25). Die genaue Art und Weise der Steuerbetragsermittlung (auftragsübergreifend oder positionsweise) und die Art der Ausweisung im Formular sind abhängig von der jeweiligen OFML-Applikation.

### 3.2 Relevante Tabelleneinträge

Innerhalb einer Preisebene werden die relevanten Preiskomponenten wie folgt ermittelt:

1. Ermitteln der Preiskomponenten für den Artikel ohne Variantenkondition.
2. Auswerten aller relevanten Preisbeziehungen und Ermitteln der Preiskomponenten für den Artikel mit den Variantenkonditionen, die in diesen Preisbeziehungen hergeleitet wurden. Preisbeziehungen sind in der Tabelle `RelationObj` (Feld 4) durch die Verwendungsart 'P' gekennzeichnet und müssen vom Typ Aktion ('3') sein (Feld 3). Variantenkonditionen werden in Preisbeziehungen (Tabelle `Relation`) durch Zuweisung der Bezeichnung der Variantenkondition an die spezielle Variable `$VARCOND` (OCD-Sprachsets) bzw. an das Hilfsmerkmal `$self.variant_condition` (SAP-Sprachsets) hergeleitet<sup>40</sup>. Sowohl `$VARCOND` als auch `$self.variant_condition` werden dabei wie mehrwertige Merkmale behandelt, d.h., ihnen können in einer Preisbeziehungen mehrere Bezeichner von Variantenkonditionen zugewiesen werden.

<sup>40</sup>Anstelle dieser Standard-Variablen kann auch eine eigene, spezifische Variable verwendet werden, die dann in der Versionsinformationstabelle (s. 2.23) anzugeben ist.

Vor dem Zugriff auf die Preistabelle werden die in den Preisbeziehungen zugewiesenen Variantenbedingungen in Großbuchstaben umgewandelt<sup>41</sup>.

Die relevanten Preisbeziehungen werden in der genannten Reihenfolge aus den Beziehungsobjekten

1. des Artikels
2. der Merkmalsklassen des Artikels
3. der aktuell bewerteten Merkmale des Artikels<sup>42</sup>
4. der Werte der aktuell bewerteten Merkmale

bestimmt.

Wird ein und dieselbe Variantenbedingung in verschiedenen Preisbeziehungen gesetzt, ist nicht definiert<sup>43</sup>, ob die zur Variantenbedingung gehörende Preiskomponente entsprechend oft in die Preisfindung eingeht oder nur einmal<sup>44</sup>.

Neben der Zuweisung einer Variantenbedingung können Preisbeziehungen auch die Bewertung von Hilfsmerkmalen für nachfolgend ausgewertete Preisbeziehungen beinhalten. Eine solche Zuweisung ist jedoch nur während der Auswertung der Preisbeziehungen innerhalb einer Preisebene wirksam.

### 3.3 Gültige Tabelleneinträge

Aus den für eine gegebene Preiskomponente aus der Preistabelle gelesenen Einträgen (s. Abschn. 3.2) wird der gültige (am besten passende) wie folgt ermittelt.

1. Einträge für die Preisebene 'B' (Grundpreis) werden ignoriert, wenn sie keinen festen (absoluten) Preisbetrag (sondern eine Prozentangabe) enthalten.
2. Einträge ohne eine korrekte Datumsangabe in den Feldern 10 und 11 (Gültigkeitszeitraum) bzw. mit einem Gültigkeitszeitraum, der für das geforderte Datum der Preisermittlung nicht erfüllt ist, werden ignoriert.
3. Soll der Preis in einer bestimmten Währung ermittelt werden, werden nur die Einträge mit dieser Währung berücksichtigt. Besitzt keiner der (zeitlich gültigen) Einträge die geforderte Währung, kommen alle Einträge in die weitere Auswahl.
4. Aus den verbliebenen Einträgen werden die Einträge ausgeschlossen, deren Mengeangabe zur Staffelpreisbildung (Feld 12) kleiner ist als die Menge zur aktuellen Bestell- bzw. Auftragsposition.
5. Aus den verbliebenen Einträgen wird der Eintrag mit dem jüngsten Datum im Feld 10 („Gültig von“) verwendet.  
(Gibt es auch hinsichtlich dieses Kriteriums mehrere Einträge, ist nicht definiert, welcher der Einträge durch die OFML–Applikation verwendet wird<sup>45</sup>.)
6. Ist in dem verbliebenen Eintrag die Rechenregel bzw. die Fixpreis-/Prozentangabe für die spezialisierte Preisebene nicht erlaubt, so wird auch dieser Eintrag ignoriert, d.h. die gewünschte Preiskomponente kann nicht ermittelt werden.

---

<sup>41</sup>s.a. Anmerkung zu den Variantenbedingungen in der Preistabelle, Abschn. 2.17

<sup>42</sup>das schließt nicht-konfigurierbare Merkmale mit ein

<sup>43</sup>abhängig von der verwendeten OFML–Applikation

<sup>44</sup>Eine derartige Datenanlage ist also nicht zu empfehlen.

<sup>45</sup>Wenn – bei einer korrekten Datenanlage – für eine Preiskomponente und eine gegebene Währung keine 2 Tabelleneinträge mit demselben Datum in Feld 10 vorhanden sind, kann dieser Fall nur auftreten, wenn für die Preiskomponente Einträge mit verschiedenen Währungen (und demselben Datum in Feld 10) vorliegen, aber keine von den Währungen der angeforderten Währung entspricht. Für die Applikation ist dann nicht ersichtlich, welches der passendste Tabelleneintrag ist.



### 3.4 Preisfaktoren

In Preisbeziehungen (Tabelle **Relation**) können Preisfaktoren zu einem Preisposten angegeben werden, der an eine Variantenbedingung gebunden ist. Dazu wird die Funktion `$SET_PRICING_FACTOR()` verwendet, die in den OCD-Sprachsets wie folgt spezifiziert ist:

- `$SET_PRICING_FACTOR(<Variantenbedingung>, <Faktor>)`

Die Funktion definiert den `<Faktor>`, mit dem der Preis multipliziert werden soll, der für die angegebene `<Variantenbedingung>` in der Tabelle **Price** festgesetzt ist.

`<Variantenbedingung>` kann als Ausdruck angegeben werden, der als Ergebnis eine Zeichenkette liefert.

`<Faktor>` ist ein arithmetischer Ausdruck. (Ist der Ausdruck undefiniert, hat die Funktion keinen Effekt.) Preisfaktoren können auch einen negativen Betrag haben.

Der Aufruf der Funktion folgt in der Regel der Zuweisung der Variantenbedingung an die Variable `$VARCOND` im selben Beziehungswissen.

Beispiel:

Ist das Merkmal „Elektrifizierung“ mit dem Wert „Set 1“ belegt, soll der Aufpreis in Abhängigkeit von der Breite des Tisches festgelegt werden. Die Preisbeziehung, die an das Beziehungsobjekt für den Wert „Set 1“ des Merkmals „Elektrifizierung“ gebunden ist, könnte dann wie folgt definiert werden:

```
$VARCOND = 'ABC123_ELEKTR_1', $SET_PRICING_FACTOR('ABC123_ELEKTR_1', BREITE / 1000)
```

Anmerkung:

Da das Merkmal `BREITE` in mm angegeben ist, wird hier der Aufpreis (aus der Preistabelle) mit der aktuellen Breite in Metern multipliziert.

Der Funktionsaufruf kann auch mit einer Bedingung verknüpft werden. Der Faktor wird dann nur verwendet, wenn die Bedingung eindeutig erfüllt ist.

Beispiel:

Ist das Merkmal „Elektrifizierung“ mit dem Wert „Set 2“ belegt, soll der Aufpreis um 10 Prozent erhöht werden, wenn die Breite des Tisches größer als ein Meter ist. Die Preisbeziehung, die an das Beziehungsobjekt für den Wert „Set 2“ gebunden ist, könnte dann wie folgt definiert werden:

```
$VARCOND = 'ABC123_ELEKTR_2',  
$SET_PRICING_FACTOR('ABC123_ELEKTR_2', 1.1) IF BREITE > 1000
```

In den SAP-Sprachsets besitzt die Funktion `$SET_PRICING_FACTOR()` zwei zusätzliche Parameter am Anfang, für die im OCD jeweils nur ein fester Wert übergeben werden kann:

- `$SET_PRICING_FACTOR($self, variant_condition, <Variantenbedingung>, <Faktor>)`

Erfolgen in einer Preisbeziehung mehrere Aufrufe zum Setzen eines Faktors für ein und dieselbe Variantenbedingung, so wird nur der letzte Aufruf wirksam.

## 4 Die Endartikelnummererzeugung

Die Endartikelnummer für einen Artikel wird gemäß dem Schema erzeugt, das für den Artikel in der Artikeltabelle anhand des Schema-Identifikators angegeben ist. Ist in der Schematabelle (s. Abschn. 2.24) kein oder ein nicht referenzierter Identifikator angegeben, wird für den Artikel keine spezielle Endartikelnummer erzeugt. Diese ist dann gleich der Grundartikelnummer.

In den folgenden Ausführungen wird auf ein einfaches Beispiel eingegangen:

Ein Schrank mit der Grundartikelnummer 0815 gehöre der Merkmalsklasse **Schrank** an und besitze aktuell folgende Merkmalsausprägungen:

Oberflaeche: 03  
Hoehe: 5H  
Zubehoer (optional): nicht ausgewaehlt

Aktuell nicht sichtbar (ungültig) sei das Merkmal **Schloss**. Die Feldlänge für alle Merkmale betrage 2.

### 4.1 Die vordefinierten Schemata

Bei diesen Schemata beginnt die Endartikelnummer grundsätzlich mit der Grundartikelnummer aus der Artikeltabelle. Dieser folgt die im 3. Feld der Schematabelle angegebene Zeichenkette. Der Trennzeichenkette folgt schließlich der Variantencode, der in den einzelnen vordefinierten Schemata wie unten beschrieben generiert wird.

Der Identifikator für ein vordefiniertes Schema (s.u.) muß im 2. Feld der Schematabelle angegeben werden.

Für die Beispiele wird die Trennzeichenkette „-“ angenommen.

- **KeyValueList**

Jedes aktuell gültige Merkmal wird in der Reihenfolge gemäß der Property-Tabelle wie folgt dargestellt:

```
<Merkmalsklasse>.<Merkmal>=<Merkmalswert>
```

Als Trennzeichen zwischen den Merkmalen wird das Semikolon verwendet. Für aktuell nicht ausgewählte optionale Merkmale wird das interne Wertkürzel „VOID“ verwendet.

Die Parameterfelder 4 bis 7 haben für dieses Schema keine Bedeutung. Als Trimm-Kennzeichen wird immer 1 verwendet (unabhängig von der Angabe im Feld 8).

Für das Beispiel ergibt sich damit:

```
0815-Schrank.Oberflaeche=03;Schrank.Hoehe=5H;Zubehoer=VOID
```

- **ValueList**

Die Merkmale werden in der Reihenfolge gemäß der Property-Tabelle alleine anhand des aktuellen Merkmalswertes dargestellt.

Die Darstellung kann dabei mit Hilfe der Parameterfelder 4 bis 10 gesteuert werden. Das Feld 5 (Sichtbarkeitsmodus) sollte bei Neu-Datenanlagen immer gefüllt sein, um ein definiertes Verhalten zu erzielen.

Für das Beispiel sei als Zeichenkette zum Trennen der Merkmalswerte (Feld 4) eine leere Zeichenkette angenommen. Für die Ersetzungszeichen (Felder 6 und 7) seien die Standardzeichen '-' bzw. 'X' verwendet. (Das Trimm-Kennzeichen hat keine Bedeutung, da alle Werte exakt die in der Property-Tabelle angegebene Länge besitzen.)

Damit ergibt sich für das Beispiel im Sichtbarkeitsmodus 0:

```
0815-035HXX
```

und im Sichtbarkeitsmodus 1:

```
0815-035HXX-
```

## 4.2 Nutzerdefinierte Schemata

Im Feld 2 können von den vordefinierten Schemata abweichende Kodierungsvorschriften definiert werden.

Die Syntax der Schemabeschreibung ist:

`<Scheme> := [<PropertyClass>:<PropertyName> | <TableCall> | @ | <Char>], 1:n`

`<PropertyClass>` := Name der Merkmalsklasse in der Tabelle Property

`<PropertyName>` := Name des Merkmals in der Tabelle Property

`<TableCall>` := Tabellenaufruf gemäß Syntax und Semantik im Sprachset `OCD_2`

`<Char>` := Ein einzelnes Zeichen (außer dem Komma und '@')

Beim Aufbau der Endartikelnummer wird die Schemabeschreibung von links nach rechts abgearbeitet. Dabei werden folgende Ersetzungen ausgeführt:

- `<PropertyClass>:<PropertyName>` wird durch den aktuellen Merkmalswert ersetzt. Dabei werden die Formatierungen aus der Property-Tabelle und die Ersetzungs- und Formatierungsanweisungen aus den Feldern 5 bis 10 der Schematabelle berücksichtigt. Ist das Feld 5 (Sichtbarkeitsmodus) leer, wird der Modus 1 verwendet.
- Beim Tabellenaufruf werden Aktualparameter, die auf ein Merkmal verweisen, durch den aktuellen Wert des Merkmals ersetzt. (Der spezielle Aktualparameter `$BAN` wird durch die Grundartikelnummer ersetzt.) Enthält die angesprochene Wertkombinationstabelle für diese Parameter genau eine passende logische Zeile und enthält diese einen Wert für den speziellen Tabellenparameter `$FAN`, so wird dieser an der aktuellen Position der Endartikelnummer eingesetzt. (Wird kein oder kein eindeutiges Ergebnis für `$FAN` geliefert, ist das Verhalten der verwendeten OFML-Applikation undefiniert.)
- '@' wird durch das nächste Zeichen der Grundartikelnummer ersetzt. Die Grundartikelnummer wird ebenfalls von links nach rechts abgearbeitet.
- Für `<Char>` erfolgt keine Ersetzung. Es wird das angegebene Zeichen an der aktuellen Position eingesetzt.  
Es können alle druckbaren Zeichen aus dem ASCII-Zeichensatz verwendet werden außer dem Komma und '@'.

Für das Beispiel ergibt sich bei der Schemabeschreibung:

```
Schrank:Hoehe,_,@,@,-,@,@,_,Schrank:Oberflaeche
```

die Endartikelnummer:

```
5H_08-15_03
```

Hinweis:

Nutzerdefinierte Endartikelnummern sind nur bedingt zur Rekonstruktion einer Artikelkonfiguration geeignet. Dazu müssen prinzipiell alle konfigurierbaren Merkmale kodiert werden, wobei die in Abschn. 2.24 genannten Einschränkungen bzgl. der Felder 6 bis 8 einzuhalten sind. *Nicht* geeignet zur Rekonstruktion einer Artikelkonfiguration ist ein nutzerdefiniertes Schema, wenn es einen Tabellenaufruf verwendet, oder wenn das Trimm-Kennzeichen gesetzt ist und im Schema zwei Merkmale unmittelbar aufeinander folgen, d.h. nicht durch Zeichen getrennt sind.

## 4.3 Mehrwertige Merkmale

Mit den Feldern 9 und 10 der Schematabelle wird in allen Codierungsarten die Darstellung der aktuellen Werte von mehrwertigen Merkmalen gesteuert:

- Prinzipiell werden alle aktuell gesetzten Werte in der Reihenfolge angezeigt, die durch die Positionsangabe in der Merkmalswerttabelle (s. Abschn. 2.13) bestimmt ist.

- Die in Feld 9 angegebene Zeichenkette wird dabei zum Trennen der Werte verwendet. Ist das Feld leer, wird ein Komma („ , “) verwendet. Ist genau ein Zeichen angegeben, so darf es beim vordefinierten Schema `ValueList` nicht identisch sein mit dem Trennzeichen aus Feld 4 der Schematabelle.
- Ist das Feld 10 nicht leer und enthält es eine gerade Anzahl von Zeichen, so wird die erste Hälfte der Zeichenkette der Werte-Auffistung vorangestellt und die zweite Hälfte an die Werte-Auffistung angehängt. Damit kann eine Klammerdarstellung realisiert werden.

Beispiel: „[ABS,ZV]“

## 5 Merkmalstext-Steuerung

In diesem Abschnitt wird beschrieben, wie mit Hilfe des Steuercodes im Feld *TxtControl* der Merkmalstabelle (s. Abschn. 2.9) die Generierung des Textes gesteuert werden kann, der das Merkmal in kaufmännischen Formularen (Artikelliste u.ä.) beschreibt.

Prinzipiell wird der das Merkmal beschreibende Text aus der (einzeiligen) Bezeichnung des Merkmals (aus Tabelle `PropertyText`) und dem Text des aktuell zugewiesenen Wertes gebildet. Merkmalswert-Texte (Tabelle `PropValueText` oder frei eingegebene beim Merkmalstyp T) können mehrzeilig sein. Je nach Steuercode werden davon jedoch nicht alle Zeilen für die Merkmalsbeschreibung verwendet.

Im Gegensatz zu den Merkmalsbeschreibungen in kaufmännischen Formularen kann das Erscheinungsbild eines Merkmals in Komponenten (grafischer) Benutzeroberflächen zur Bewertung von Merkmalen (Eigenschaftseditoren) nicht beeinflusst werden. Ein Merkmal wird dort (außer beim Typ T) immer durch seine (einzeilige) Bezeichnung und die erste Zeile des Textes des aktuell zugewiesenen Wertes dargestellt.

Folgende Codes sind definiert<sup>46</sup>:

- 0 Die erste Zeile der Merkmalsbeschreibung wird aus der Merkmalsbezeichnung und der ersten Zeile des Textes des aktuellen Merkmalswerts gebildet<sup>47</sup>. Es folgen die restlichen Zeilen des Merkmalswert-Textes.

Bei mehrwertigen Merkmale (siehe Feld *MultiOption* in der Merkmalstabelle, Abschn. 2.9) werden jeweils die ersten Zeilen der Texte der aktuell gesetzten Werte aneinander gereiht, wobei die Reihenfolge durch die Positionsangabe in der Merkmalswerttabelle (s. Abschn. 2.13) bestimmt ist und als Trenn-Zeichenkette zwischen den Werten die Angabe aus dem Feld *MO\_Sep* der Schema-Tabelle (Abschn. 2.24) verwendet wird (bzw. „“, wenn das Feld leer ist).

Dies ist das Standardverfahren, insbesondere bei einzeiligen Merkmalswert-Texten.

- 1 Die Merkmalsbeschreibung entspricht dem (mehrzeiligen) Text des aktuellen Merkmalswerts, d.h., die übliche Merkmalsbezeichnung in der ersten Zeile (Standard) wird unterdrückt.

Beispiel:

Ein Stuhl besitze das Merkmal „Mechanik“ mit den Werten:

„Standardmechanik mit Gasdruckfeder“ (1 Zeile)

„Synchronmechanik Optima Plus“ (1 Zeile)

Die Merkmalsbeschreibung im Formular lautet bei Auswahl des ersten Wertes dann

„Standardmechanik mit Gasdruckfeder“

anstelle von

„Mechanik: Standardmechanik mit Gasdruckfeder“

im Standardfall.

- 2 Die erste Zeile der Merkmalsbeschreibung besteht nur aus der Merkmalsbezeichnung. Die restlichen Zeilen entsprechen den Zeilen 2..n des Textes des aktuellen Merkmalswertes.

Beispiel:

Ein Schrank besitze das Merkmal „Fachböden verstärkt“ mit den Werten „Ja“ und „Nein“.

Für den Wert „Nein“ wird in der Tabelle `PropertyValue` das Feld *SuppressTxt* auf True gesetzt.

Bei Auswahl dieses Wertes erscheint im Formular kein Text zu diesem Merkmal.

Beim Wert „Ja“ ist „SuppressTxt“ auf False gesetzt. Als Merkmalsbeschreibung erscheint

„Fachböden verstärkt“

anstelle von

„Fachböden verstärkt: Ja“

im Standardfall.

- 3 Die Merkmalsbeschreibung wird aus den Zeilen 2..n des Textes des aktuellen Merkmalswertes gebildet, d.h. Merkmalsbezeichnung und erste Zeile des Merkmalswert-Textes werden unterdrückt.

<sup>46</sup>Bei Merkmalen des Typs T sind nur die Codes 0 und 5 erlaubt

<sup>47</sup>Die beiden Textbausteine werden dabei durch einen Doppelpunkt und ein Leerzeichen getrennt, falls diese Zeichen nicht schon am Ende der Merkmalsbezeichnung enthalten sind.

Beispiel:

Ein Tisch besitze das Merkmal „Elektrifizierung“ und für dessen Wert E01 sei folgender Text hinterlegt:

Zeile 1: Set 1

Zeile 2: Elektrifizierung bestehend aus:

Zeile 3: - 2x Kabelschlange

Zeile 4: - 2x Vielfach-Steckdose

Im Eigenschaftseditor erscheint: „Elektrifizierung: Set 1“.

Die Merkmalsbeschreibung lautet aber:

„Elektrifizierung bestehend aus:

- 2x Kabelschlange

- 2x Vielfach-Steckdose“

#### 4 Keine Beschreibung des Merkmals im Formular.

Die gleiche Wirkung kann erzielt werden, wenn für alle Werte des Merkmals in der Tabelle `PropertyValue` das Feld `SuppressTxt` auf `True` gesetzt ist.

Sinnvoll ist der Einsatz dieses Modus z.B. für Hilfsmerkmale, die vom Nutzer konfiguriert werden können, aber nicht gedruckt werden sollen.

Beispiel:

Ein Stuhl besitze das Merkmal „zweifarbiger Bezug“ mit den Werten „Ja“ und „Nein“. Bei Auswahl von „Nein“ werde das Merkmal „Farbe Bezug“ freigeschaltet, bei „Ja“ hingegen die Merkmale „Farbe Sitzbezug“ und „Farbe Rückenbezug“.

Das Merkmal „zweifarbiger Bezug“ soll/muss in diesem Fall nicht im Formular beschrieben werden.

Dies wird durch den Code 4 bewerkstelligt.

#### 5 Gilt speziell für mehrwertige Merkmale und Merkmale des Typs T.

Für mehrwertige Merkmale gilt:

Die erste Zeile der Merkmalsbeschreibung besteht aus der Merkmalsbezeichnung. Es folgen die (evtl. mehrzeiligen) Bezeichnungen der aktuell gesetzten Werte in der Reihenfolge, die durch die Positionsangabe in der Merkmalswerttabelle (s. Abschn. 2.13) bestimmt ist. Die Bezeichnungen des ersten bis vorletzten Wertes werden dabei mit der Zeichenkette abgeschlossen, die im Feld `MO_Sep` der Schema-Tabelle (Abschn. 2.24) angegeben ist (bzw. „ ,“, wenn das Feld leer ist).

Für Merkmale des Typs T gilt:

Die erste Zeile der Merkmalsbeschreibung besteht aus der Merkmalsbezeichnung. Es folgen die Zeilen des aktuell durch den Anwender eingegebenen Textes.

## 6 Die Ermittlung von Verpackungsdaten

In diesem Abschnitt wird beschrieben, wie mit Hilfe der Packaging-Tabelle (Abschn. 2.5) und Beziehungswissen (Abschn. 2.15 und 2.16) die Verpackungsdaten (Maße, Volumen, Gewichte, Anzahl Verpackungseinheiten) zu einem Artikel in einer konkreten Konfiguration ermittelt werden.

Die Berechnung der Daten erfolgt in folgenden Schritten:

1. Ermitteln der Grunddaten für den Artikel in Grundauführung durch Auslesen des Eintrags ohne Variantenbedingung (leeres Feld 2) aus der Tabelle **Packaging**.
2. Ermitteln der abweichenden Daten mit Hilfe von Variantenbedingungen:
  1. Ermitteln aller für die aktuelle Konfiguration geltenden Variantenbedingungen anhand der Packaging-Beziehungen.
  2. Für jede ermittelte, aktuell gültige Variantenbedingung:  
Auslesen des zugehörigen Eintrags aus der Tabelle **Packaging** und Addition der Beträge aller nicht-leeren Felder zu den entsprechenden Beträgen aus den Grunddaten bzw. zu den durch vorhergehende Variantenbedingungen evtl. bereits akkumulierten Beträgen.

Zum Schritt 2.1:

- Packaging-Beziehungen sind in der Tabelle **RelationObj** (Feld 5) durch die Verwendungsart 'PCKG' gekennzeichnet und müssen vom Typ Aktion ('3') sein (Feld 4).
- Variantenbedingungen werden in Packaging-Beziehungen (Tabelle **Relation**) durch Zuweisung der Bezeichnung der Variantenbedingung an die spezielle Variable **\$VARCOND** hergeleitet.
- Eine Variantenbedingung kann mit einem Faktor verknüpft werden. Dazu wird folgende Funktion verwendet:

*\$SET\_PCKG\_FACTOR(<Variantenbedingung>, <Datenelement>, <Faktor>)*

Die Funktion definiert den <Faktor>, mit dem der Wert des <Datenelementes> aus dem Eintrag der Packaging-Tabelle multipliziert werden soll, der für die angegebene <Variantenbedingung> gelesen wurde.

<Variantenbedingung> kann als Ausdruck angegeben werden, der als Ergebnis eine Zeichenkette liefert.

<Datenelement> ist der Name des betreffenden Feldes der Packaging-Tabelle in Großschreibung, eingeschlossen in Hochkommata, z.B. 'NETWEIGHT'.

<Faktor> ist ein arithmetischer Ausdruck. (Ist der Ausdruck undefiniert, hat die Funktion keinen Effekt.)

Der Aufruf der Funktion folgt in der Regel der Zuweisung der Variantenbedingung an die Variable **\$VARCOND** im selben Beziehungswissen.

Der Funktionsaufruf kann mittels einer IF-Klausel (s. Anh. B) an eine Bedingung geknüpft werden.

- Die relevanten Packaging-Beziehungen werden in der genannten Reihenfolge aus den Beziehungsobjekten
  1. des Artikels
  2. der Merkmalsklassen des Artikels
  3. der aktuell bewerteten Merkmale des Artikels<sup>48</sup>
  4. der Werte der aktuell bewerteten Merkmale

bestimmt.

---

<sup>48</sup>das schließt nicht-konfigurierbare Merkmale mit ein

## A Sprachdefinition OCD\_1

- Diese einfache Sprache erlaubt die Angabe von Bedingungen (alle Arten von Beziehungen) und von Zuweisungen an die Merkmale eines Artikels (in Aktionen, Reaktionen und Post-Reaktionen). Innerhalb einer Aktion, Reaktion oder Post-Reaktion können dabei mehrere Zuweisungen stattfinden. Diese sind durch Kommata voneinander zu trennen.
- Zeichenkettenkonstanten sind in Hochkommata (') einzuschließen.
- Bei Schlüsselwörtern sowie bei Bezeichnern für Merkmale und bei Werten von Zeichenketten-Merkmalen wird Groß-/Kleinschreibung ignoriert.  
Bsp.: `IF` ist identisch mit `if`.
- In logischen und arithmetischen Ausdrücken können Merkmalsnamen im Sinne von Variablen verwendet werden. Bei der Auswertung des Ausdrucks werden sie durch den aktuellen Wert des Merkmals ersetzt. Die spezielle Variable `$BAN` wird bei der Auswertung durch die Grundartikelnummer des aktuellen Artikels ersetzt.
- **Arithmetische Ausdrücke:**
  - Arithm. Ausdrücke können verwendet werden
    - \* als numerische Operanden in Vergleichen,
    - \* auf der rechten Seite von Zuweisungen und
    - \* als Faktoren in den Builtin-Funktionen `SET_PRICING_FACTOR()` und `SET_PCKG_FACTOR()`.
  - Ein komplexer arithmetischer Ausdruck wird mittels der arithmetischen Grundoperationen und Klammer-Setzung aus untergeordneten arithmetischen Ausdrücken gebildet.
  - Einfache arithmetische Ausdrücke sind:
    - \* numerische Konstanten
    - \* Verweise auf numerische Merkmale
    - \* Aufrufe einer arithmetischen Funktion (s. Anhang F)

Bezieht sich ein einfacher arithmetischer Ausdruck auf ein Merkmal, das der betreffende Artikel nicht besitzt, oder das nicht bewertet ist, so ist der Wert des Ausdrucks *undefiniert*. Ist ein undefinierter arithmetischer Ausdruck Teil eines übergeordneten, komplexen arithmetischen Ausdrucks, so ist auch der Wert dieses Ausdrucks undefiniert.
- **Bedingungen:**
  - Bedingungen sind einfache oder komplexe boolesche (logische) Ausdrücke. Ein logischer Ausdruck wird entweder als *wahr* oder *falsch* bewertet. Unter Umständen kann ein logischer Ausdruck nicht bewertet werden, das Ergebnis ist dann *undefiniert* (s.u.).  
Komplexe boolesche Ausdrücke werden mittels der Operatoren `AND` und `OR` aus Unterausdrücken gebildet.  
Bei verketteten `AND` und `OR` Operatoren werden zuerst die `AND` Verknüpfungen ausgewertet. Die Reihenfolge der Auswertung kann durch Klammer-Setzung gesteuert werden. Vergleiche z.B. `A and B or C` gegenüber `A and (B or C)`.
  - Einfache logische Ausdrücke sind:
    - \* Vergleiche
    - \* Verneinung (Negation)
    - \* Spezielle Bedingungen
  - Vergleiche werden mittels der bekannten Vergleichsoperatoren notiert: `<` (oder `LT`), `<=` (oder `LE`), `=` (oder `EQ`), `<>` (oder `NE`), `=>`<sup>49</sup> (oder `GE`) und `>` (oder `GT`).  
Die Operanden auf beiden Seiten des Vergleichs müssen vom selben Typ sein (nur Zeichenkette oder nur numerisch).

---

<sup>49</sup>alternativ kann die Form `>=` verwendet werden



- Der Vergleich von Zeichenketten basiert auf der lexikographischen Ordnung, d.h. zwei Zeichenketten werden Zeichen für Zeichen verglichen, bis zwei sich unterscheidende Zeichen gefunden werden. Die Zeichenkette, dessen sich unterscheidendes Zeichen lexikographisch kleiner als das entsprechende Zeichen der anderen Zeichenkette ist, gilt dann als die kleinere Zeichenkette. Falls eine Zeichenkette komplett abgearbeitet wurde, bevor ein Unterschied festgestellt wird, gilt die kürzere Zeichenkette als die kleinere Zeichenkette.  
Die lexikographische Ordnung der Zeichen ergibt sich aus der Codierung der Zeichen im verwendeten Zeichensatz. Beim im OCD verwendeten Zeichensatz ISO-8859-1 (Latin-1) gilt somit z.B. 'A' < 'a'.
- Die lexikographische Ordnung ist insbesondere bei Zeichenketten zu beachten, die komplett aus Ziffern bestehen. Dort kann es zu unterschiedlichen Ergebnissen gegenüber dem Vergleich entsprechender Zahlen kommen. So ist z.B. 900 < 1000 aber '900' > '1000' !
- Mittels NOT-Operator können logische Ausdrücke verneint werden.
- *Spezielle Bedingungen* sind:
  - \* SPECIFIED <Merkmalsname>  
Diese Bedingung ist wahr, wenn der Artikel das angegebene Merkmal besitzt und dieses mit einem Wert belegt ist.
  - \* <Merkmalsname> IN (<Wertemenge>)  
Diese Bedingung ist wahr, wenn der aktuelle Wert des im linken Operanden angegebenen Merkmals in der Wertemenge enthalten ist, die im rechten Operanden angegeben ist. Die Werte in der Wertemenge sind durch Kommata zu trennen.
- Bezieht sich ein einfacher logischer Ausdruck auf ein Merkmal, das der betreffende Artikel nicht besitzt, oder das nicht bewertet ist, so kann der Ausdruck nicht bewertet werden und das Ergebnis ist *undefiniert*. Einzige Ausnahme ist der SPECIFIED-Ausdruck, der dazu genutzt werden kann, undefinierte logische Ausdrücke zu vermeiden.  
Für die Verknüpfungsoperatoren AND und OR gelten dann folgende Regeln:
  - \* Das Ergebnis einer OR Verknüpfung ist undefiniert, wenn entweder beide Unterausdrücke undefiniert sind, oder wenn ein Unterausdruck undefiniert und der andere nicht wahr ist. (Ist wenigstens ein Unterausdruck wahr, so ist die OR Verknüpfung in jedem Fall wahr, auch wenn der andere Unterausdruck undefiniert ist.)
  - \* Das Ergebnis einer AND Verknüpfung ist undefiniert, wenn entweder beide Unterausdrücke undefiniert sind, oder wenn ein Unterausdruck undefiniert und der andere wahr ist. (Ist wenigstens ein Unterausdruck nicht wahr, so ist die AND Verknüpfung in jedem Fall nicht wahr, auch wenn der andere Unterausdruck undefiniert ist.)
- Für die verschiedenen Beziehungsarten gelten folgende Regeln in Bezug auf undefinierte logische Ausdrücke:
  - \* Eine *Vorbedingung* ist verletzt, wenn sie eindeutig falsch ist, d.h. sie ist nicht verletzt, wenn der logische Ausdruck undefiniert ist.
  - \* Eine *Auswahlbedingung* ist verletzt, wenn sie nicht eindeutig wahr ist, d.h. sie ist auch verletzt, wenn der logische Ausdruck undefiniert ist.

- **Zuweisungen:**

- Zuweisungen erfolgen über den Zuweisungsoperator =. Der linke Operand ist ein Merkmal oder die spezielle Variable \$VARCOND (s. Abschn. 3).
- Die Operanden auf beiden Seiten der Zuweisung müssen vom selben Typ sein (Zahl vs. Zeichenkette).
- Ist der linke Operand ein numerisches Merkmal, so findet ggf. eine Rundung des rechten Operandes gemäß der für das Merkmal festgelegten Anzahl erlaubter Nachkommastellen statt. Dabei wird die Methode des mathematischen Rundens angewendet.
- Steht auf der rechten Seite der Zuweisung ein undefinierter arithmetischer Ausdruck, so findet keine Zuweisung statt.
- Eine Zuweisung kann mit einer Bedingung versehen werden. Diese ist nach dem Schlüsselwort IF anzugeben. Die Zuweisung findet dann nur statt, wenn die Bedingung eindeutig erfüllt ist (also nicht, wenn der logische Ausdruck undefiniert ist).

## B Sprachdefinition OCD\_2

Diese Sprachdefinition umfasst alle Festlegungen aus der Sprachdefinition OCD\_1. Darüberhinaus gelten folgende weitergehende Festlegungen:

- Zeichenketten können mittels des Operators + verkettet werden. Dadurch entstehen *Zeichenketten-Ausdrücke*, in denen Zeichenkettenkonstanten und Werte von Zeichenketten-Merkmalen miteinander verknüpft werden können.
- Numerische Werte können mittels der Funktion `STRING()` in eine Zeichenkette konvertiert werden: Die Ergebnis-Zeichenkette enthält die einfache Dezimalpunktnotation der übergebenen Zahl mit dem Punkt (‘.’) als Dezimaltrennzeichen und ohne Tausender- oder sonstige Trennzeichen. Nicht-signifikante gebrochene Teile werden nicht dargestellt, z.B. `9.0` → „9“. Die Funktion kann auch in Zeichenketten-Ausdrücken verwendet werden.
- In logischen Ausdrücken können die booleschen Konstanten `FALSE` und `TRUE` verwendet werden.
- Für *einschränkbare Merkmale* gelten folgende Konkretisierungen in Bezug auf Bedingungen:
  - Die Bedingung `SPECIFIED` ist erfüllt, wenn der Wertebereich für das Merkmal auf genau einen Wert eingeschränkt wurde.
  - Ein Vergleich (inklusive der `IN`-Bedingung) ist nur möglich, wenn der Wertebereich auf genau einen Wert eingeschränkt wurde (anderenfalls ist das Ergebnis des Ausdrucks undefiniert).
- Für den Beziehungstyp *Constraint* wird eine Syntax und Semantik definiert, die unten genauer beschrieben ist.
- In Vorbedingungen, Constraints, Aktionen und Reaktionen können Prüfungen und Werteverleitungen mit Hilfe von Wertkombinationstabellen und der Funktion `TABLE()` realisiert werden. Die diesbezügliche Syntax und Semantik wird unten genauer beschrieben.

### B.1 Constraints

- Ein Constraint ist ein komplexes Sprachkonstrukt, das vornehmlich zur Überwachung der Konsistenz einer Konfiguration verwendet wird, aber auch zur Herleitung von Merkmalswerten oder zur Einschränkung von Wertebereichen eingesetzt werden kann.
- Constraints müssen immer an Artikel gebunden sein (s. Abschn. 2.15). Damit können in einem Constraint Aussagen über Merkmale von mehreren Merkmalsklassen des Artikels gemacht werden.
- Ein Constraint besteht aus bis zu vier Teilen, die jeweils durch ein Schlüsselwort plus Doppelpunkt eingeleitet und durch einen Punkt abgeschlossen werden:

- **Objects:**

In diesem Teil werden die Objekte benannt, über die Aussagen im Constraint gemacht werden, wobei hier unter Objekten Merkmalsklassen und Merkmale verstanden werden. Mehrere Objektdeklarationen werden durch Kommatas getrennt.

Merkmalsklassen und Merkmale werden in den folgenden Constraint-Teilen über den Namen von Variablen angesprochen, die für sie im `Objects`-Teil deklariert wurden:

- Variablen für Merkmalsklassen werden mit Hilfe des Konstrukts `IS_A` deklariert:

```
<Variable> IS_A <Merkmalsklasse>
```

Ist einer der deklarierten Merkmalsklassen nicht dem Artikel zugeordnet, an den das Constraint gebunden ist, wird das Constraint nicht ausgewertet.

- Deklarationen von Variablen für Merkmale folgen der Deklaration der Klasse, zu dem die Merkmale gehören. Die Merkmalsdeklarationen werden durch das Schlüsselwort `WHERE` eingeleitet. Mehrere Merkmalsdeklarationen sind durch Semikolon zu trennen. Eine Merkmalsdeklaration hat folgende Form:

```
<Variable> = <Merkmal>
```

Wird für ein Merkmal keine eigene Variable definiert, so kann es in den folgenden Constraint-Teilen über die Variable seiner Merkmalsklasse angesprochen werden: `<Merkmalsklassenvariable>.<Merkmal>`.

- **Condition:**

gibt die Bedingung an, die erfüllt sein muß, damit das Constraint ausgewertet wird.

Für die Syntax dieses Teils gelten die allgemeinen Festlegungen für Bedingungen gemäß Sprachdefinition `OCD_1`.

Ist das Ergebnis der Auswertung der Bedingung undefiniert, so wird das Constraint nicht ausgewertet, da keine sichere Aussage darüber möglich ist, ob es ausgewertet werden darf.

Dieser Constraint-Teil ist optional. Fehlt er, so wird das Constraint ausgewertet.

- **Restrictions:**

gibt die Beziehungen an, die zwischen den Merkmalen des Artikels bestehen müssen, damit die aktuelle Konfiguration des Artikels als konsistent betrachtet wird. Mehrere Beziehungen werden durch Kommata voneinander getrennt.

Ohne den **Inferences**-Teil (s.u.) stellen die Beziehungen einfach Bedingungen dar. Ist eine der Beziehungen nicht erfüllt bzw. keine eindeutige Aussage darüber möglich (undefinierter logischer Ausdruck), ist das Constraint nicht erfüllt und der Artikel besitzt damit eine inkonsistente (ungültige) Konfiguration. Von der Laufzeitumgebung wird sichergestellt, daß entweder ein inkonsistenter Artikel nicht bestellt werden kann oder daß eine Merkmalsänderung, die zu einem inkonsistenten Zustand führen würde, nicht ausgeführt werden darf.

Zusammen mit dem **Inferences**-Teil können die Beziehungen gleichzeitig auch Werteherleitungen oder Wertebereichseinschränkungen bewirken. In diesen Fällen erzwingt das Constraint eine konsistente Konfiguration.

Alle Beziehungen können mit einer Bedingung versehen werden. Diese ist nach dem Schlüsselwort **IF** anzugeben. Die Beziehung wird dann nur ausgewertet, wenn die Bedingung erfüllt ist.

Beziehungen können beschrieben werden durch:

- *Wertevergleich*

Die Ausdrücke auf beiden Seiten des Gleichheit-Operators = müssen gleich sein.

- *Wertemengen-Prüfung: <Merkmalsname> IN (<Wertemenge>)*

Der Wert des auf der linken Seite angegebenen Merkmals muß in der im rechten Teil des Ausdrucks angegebenen Wertemenge enthalten sein.

- *Aufruf einer Wertkombinationstabelle*

Zu der aktuellen Konfiguration des Artikels muß ein passender Eintrag in der Wertkombinationstabelle enthalten sein.

Syntax und Semantik von Tabellenaufrufen werden im nächsten Abschnitt beschrieben.

- **Inferences:**

bestimmt die Merkmale (komma-separierte Aufzählung), für welche durch das Constraint Werte hergeleitet werden sollen oder für die deren Wertebereich eingeschränkt werden soll<sup>50</sup>.

Dieser Constraint-Teil ist optional. Er entfällt, wenn keine Werte hergeleitet oder Wertebereiche eingeschränkt werden sollen.

Die Herleitung bzw. die Einschränkung erfolgt anhand der Beziehungen im **Restrictions**-Teil.

- *Wertherleitung mittels Wertevergleich:*

Ist ein Operand des Gleichheit-Operators eine Merkmalsvariable und ist das Merkmal unter **Inferences** angeführt, bewirkt die Beziehung eine Zuweisung des Werts des anderen Operanden an das Merkmal, vorausgesetzt, der Wert des anderen Operanden ist bestimmt<sup>51</sup>.

<sup>50</sup>Merkmale, deren Wertebereiche mittels eines Constraints eingeschränkt werden können, müssen in der Merkmalstabelle als einschränkbar gekennzeichnet sein.

<sup>51</sup>Der Vergleich ist damit immer erfüllt.

Beispiel:  
 Objects:  
   cup IS\_A cupboard\_a.  
 Condition:  
   cup.ausf\_gruppe = 'A'.  
 Restrictions:  
   cup.farbe\_tuer = 'F002'.  
 Inferences:  
   cup.farbe\_tuer.

Ist der linke Operand eine Merkmalsvariable und das Merkmal *nicht* einschränkbar und der Wert des rechten Operanden bestimmt, so erfolgt in jedem Fall eine Zuweisung, auch wenn das Merkmal nicht unter **Inferences** angeführt ist.

- *Wertebereich-Einschränkung mittels IN-Ausdruck:*

Ist das Merkmal auf der linken Seite des IN-Ausdrucks unter **Inferences** angeführt und einschränkbar, bewirkt der Ausdruck eine Einschränkung der Wertemenge des Merkmals auf die im rechten Teil des Ausdrucks angegebene Menge<sup>52</sup>. Die im rechten Teil des Ausdrucks angegebene Menge darf in diesem Fall keine Intervall-Werte beinhalten.

Beispiel:  
 Objects:  
   cup IS\_A cupboard\_a.  
 Restrictions:  
   cup.farbe\_korpus IN ('F001', 'F002', 'F003').  
 Inferences:  
   cup.farbe\_korpus.

- *Wertherleitung und Wertebereich-Einschränkung mittels Tabellenaufruf:*

Beispiel:  
 Objects:  
   cup IS\_A cupboard\_a  
   where  
     ausf = ausf\_gruppe;  
     korpus = farbe\_korpus.  
 Restrictions:  
   TABLE farben\_korpus (ausf\_gruppe = ausf,  
                           farbe\_korpus = korpus).  
 Inferences:  
   korpus.

Syntax und Semantik des Tabellenaufrufs wird im nächsten Abschnitt näher beschrieben.

Eine einmal eingeschränkte Wertemenge eines einschränkbaren Merkmals kann durch ein nachfolgend ausgewertetes Constraint nicht wieder erweitert werden. D.h., enthält die Wertemenge, die durch einen IN-Ausdruck oder durch einen Tabellenaufruf bestimmt ist, einen Wert, der in der aktuellen Wertemenge nicht enthalten ist, so wird der Wert auch nicht in die neue Wertemenge übernommen.

---

<sup>52</sup>Die IN-Beziehung ist damit immer erfüllt.

## B.2 Tabellenaufruf

Mit Hilfe der Funktion TABLE() kann in Vorbedingungen, Aktionen, Reaktionen und Constraints auf Wertkombinationstabellen (s.Abschn. 2.21) zugegriffen werden.

Die allgemeine Syntax des Aufrufs ist:

```
TABLE <Tabellenname> (<Parameterliste>)
```

Im Tabellennamen dürfen alphanumerische Zeichen inklusive dem Unterstrich verwendet werden, wobei das erste Zeichen kein numerisches sein darf.

<Parameterliste> ist eine komma-separierte Auflistung der Zugriffsparameter:

<Tabellenmerkmal> = <Aktualparameter>.

*Aktualparameter* können sein:

- numerische oder Zeichenkettenkonstante
- Merkmalsvariable

Namen von Merkmalen von Artikel und Tabelle müssen nicht identisch sein, der Aktualparameter muß jedoch in jedem Fall mit dem Wert des zugeordneten Tabellenmerkmals verglichen werden können.

Die Semantik des Tabellenaufrufs unterscheidet sich in den verschiedenen Beziehungsarten und wird in den folgenden Unterabschnitten beschrieben.

### B.2.1 Tabellenaufruf in Vorbedingungen

In Vorbedingungen stellt der Tabellenaufruf einen logischen Ausdruck dar:

- Ist einer der Aktualparameter in der aktuellen Konfiguration nicht bewertet, so findet kein Tabellenzugriff statt und der Ausdruck ist undefiniert. (Zur Behandlung von undefinierten logischen Ausdrücken s.Anhang A).
- Wird zu den Aktualparametern genau ein Eintrag in der Tabelle gefunden, ist das Ergebnis *wahr*, ansonsten *falsch*.

### B.2.2 Tabellenaufruf in Aktionen und Reaktionen

In Aktionen und Reaktionen werden Wertkombinationstabellen dazu verwendet, um Werte für Merkmale herzuleiten.

Die Merkmale, für die durch einen Tabellenaufruf Werte herzuleiten sind, müssen in der Parameterliste als Aktualparameter angegeben sein und mit dem Präfix „\$SELF.“ versehen sein. Alle anderen Aktualparameter dienen als Zugriffsschlüssel. Eine Ausnahme ist die Verwendung der spezifischen Variablen \$VARCOND<sup>53</sup> in Preisbeziehungen zum Setzen von Variantenbedingungen: Ist diese Variable als Aktualparameter angegeben, so erfolgt für diese immer eine Werteverleitung durch den Tabellenaufruf.

Sind keine oder alle Aktualparameter mit dem Präfix „\$SELF.“ versehen, so werden für alle diejenigen Merkmale Werte hergeleitet, die als Aktualparameter angegeben, zum Zeitpunkt des Aufrufs aber nicht bewertet sind.

Es findet kein Tabellenzugriff und damit keine Werteverleitung statt, wenn entweder keine Aktualparameter gemäß der obigen Festlegungen als Zugriffsschlüssel bestimmt werden konnten oder wenn einer der explizit als Schlüsselparameter definierten Aktualparameter nicht bewertet ist.

Der Tabellenaufruf muß für jedes herzuleitende Merkmal zu einem eindeutigen Ergebnis führen. Anderenfalls schlägt der Aufruf fehl und es findet keine Werteverleitung statt.

<sup>53</sup>bzw. der in der Versionsinformationstabelle dafür als Ersatz festgelegten Variablen

### B.2.3 Tabellenaufruf in Constraints

Die Behandlung von Tabellenaufrufen in Constraints ist komplexer als in Aktionen und hängt sowohl vom aktuellen Bewertungskontext hinsichtlich der Aktualparameter als auch vom **Inferences**-Teil ab:

- Fehlt der **Inferences**-Teil oder sind alle Aktualparameter bewertet, so dient der Tabellenaufruf alleine der Konsistenzprüfung:  
Alle Aktualparameter dienen dann als Schlüssel für den Tabellenzugriff.

Ist einer der Aktualparameter in der aktuellen Konfiguration nicht bewertet (bei fehlendem **Inferences**-Teil), so findet kein Tabellenzugriff statt und das Constraint ist nicht erfüllt. Enthält die Tabelle keinen Eintrag für den durch die Aktualparameter bestimmten Schlüssel, so ist das Constraint ebenfalls nicht erfüllt.

- Sind ein oder mehrere einschränkbare Merkmale, die beim Tabellenaufruf als Aktualparameter übergeben werden, nicht bewertet, so dienen die anderen Aktualparameter als Schlüssel für den Tabellenzugriff, ausgenommen von nicht einschränkbaren Merkmalen, die im **Inferences**-Teil angeführt sind. Es erfolgt dann eine Herleitung von Werten für diejenigen Merkmale, die unter **Inferences** angeführt sind:
  - Bei einem einschränkbaren Merkmal wird die Wertemenge auf die Schnittmenge der Wertemenge vor der Auswertung des Constraints und der Wertemenge eingeschränkt, die durch den Tabellenzugriff geliefert wurde.
  - Einem nicht einschränkbaren Merkmal wird der durch den Tabellenzugriff gelieferte Wert zugewiesen. Hat der Tabellenzugriff mehrere Werte geliefert, ist das Constraint nicht erfüllt.

Ist ein als Aktualparameter angegebenes einschränkbares Merkmal nicht bewertet und nicht unter **Inferences** angeführt, so bleibt seine Wertemenge unverändert.

- Sind alle Aktualparameter Merkmalsvariablen und sind alle diese Merkmale einschränkbar, aber in der aktuellen Konfiguration nicht bewertet, so existiert kein Schlüssel für einen Tabellenzugriff.

In diesem Fall wird der Tabellenaufruf wie folgt behandelt:

Es werden alle logischen Zeilen der Tabelle der Reihe nach ausgelesen. Für jedes herzuleitende (unter **Inferences** aufgeführte) Merkmal wird geprüft, ob die für dieses Merkmal in der logischen Zeile angegebenen Werte in der aktuell eingeschränkten Wertemenge enthalten sind. Falls die Werte aller herzuleitenden Merkmale der logischen Tabellenzeile gültig sind, werden die Werte jeweils in eine Liste pro Merkmal übernommen (wobei ein mehrfaches Auftreten von Werten ausgeschlossen wird). Nach der Verarbeitung aller logischen Tabellenzeilen werden die so ermittelten Wertemengen dann den jeweiligen herzuleitenden Merkmalen als neue (eingeschränkte) Wertemenge zugewiesen.

## C Sprachdefinition OCD\_3

Diese Sprachdefinition umfasst alle Festlegungen aus der Sprachdefinition OCD\_2. Darüberhinaus gelten folgende weitergehende Festlegungen zur Verarbeitung mehrwertiger Merkmale und zur mehrstufigen Konfiguration.

### C.1 mehrwertige Merkmale

- Der logische Ausdruck  
SPECIFIED <mehrwertiges Merkmal>  
ist wahr, wenn mindestens ein Wert gesetzt ist.
- Bei IN-Vergleichen darf ein mehrwertiges Merkmal nur auf der rechten Seite stehen. Auf der linken Seite muß eine Konstante stehen.  
  
Beispiel: 'ABS' IN Sonderaustattung
- Bei normalen Vergleichen dürfen 2 mehrwertige Merkmale *nicht* miteinander verglichen werden. Ein mehrwertiges Merkmal darf immer nur mit einer Konstante verglichen werden.  
  
Beispiel: Sonderaustattung = 'ABS' ist wahr, wenn der Wert ABS gesetzt ist (unabhängig davon, ob evtl. noch weitere Werte gesetzt sind).
- Im Condition-Teil von Constraints dürfen auch 2 mehrwertige Merkmale miteinander verglichen werden. Der Vergleich ist wahr, wenn beide Mengen gesetzter Werte gleich sind.
- Im Zuweisungsteil von Aktionen bzw. im Restrictions-Teil von Constraints sind auf beiden Seiten des Ausdrucks mehrwertige Merkmale erlaubt. In diesem Fall werden alle aktuell gesetzten Werte des Merkmals auf der rechten Seite für das Merkmal auf der linken Seite übernommen.

### C.2 mehrstufige Konfiguration

In Beziehungen von Artikeln, die Unterartikel eines kompositen Artikels (s. Abschn. 2.6) sind oder sein können, ist es notwendig zu unterscheiden, ob ein Merkmal des aktuell zu konfigurierenden Artikels oder ein Merkmal eines übergeordneten Artikels referenziert wird. Dazu sind folgende Klassifikatoren für Merkmale definiert:

- **\$self**:  
bezieht sich auf den aktuell konfigurierten Artikel
- **\$parent**:  
bezieht sich auf den unmittelbar übergeordneten Artikel
- **\$root**:  
bezieht sich auf den obersten Artikel in einer mehrstufigen Konfiguration

Die Klassifikatoren werden dem Merkmalsnamen durch einen Punkt getrennt vorangestellt. Ist ein Merkmal nicht klassifiziert, so wird **\$self** angenommen.

## D Sprachdefinition OCD\_4

Diese Sprachdefinition umfasst alle Festlegungen aus der Sprachdefinition OCD\_3. Darüberhinaus gelten folgende weitergehende Festlegungen:

- Anweisungsblock:

In den Beziehungsarten *Aktion*, *Reaktion* und *Post-Reaktion* können Anweisungen mittels geschweifeter Klammern `{...}` in einem Block zusammenfasst werden.

Dies ist insbesondere dann hilfreich, wenn mehrere Anweisungen mit ein und derselben Bedingung verknüpft werden sollen.

- Wertebereiche in IN-Ausdrücken:

Neben diskreten Werten können in der Werteliste auch nach oben bzw. unten offene sowie geschlossene Wertebereiche (Intervalle) angegeben werden.

Operatoren für Obergrenze: `<` (oder `LT`), `<=` (oder `LE`)

Operatoren für Untergrenze: `=>`<sup>54</sup> (oder `GE`) und `>` (oder `GT`)

nach unten offener Bereich: `<Obergrenze-Operator><diskreter Wert>`

nach oben offener Bereich: `<Untergrenze-Operator><diskreter Wert>`

geschlossener Wertebereich: `<diskreter Wert>-<diskreter Wert>`

Die für einen geschlossenen Wertebereich als untere und obere Grenze angegebenen Werte gehören zum Wertebereich.

Beispiel:

`Breite IN (5-10, 20, >30)`

- Platzhalter in IN-Ausdrücken:

Bei Vergleichen mittels IN-Ausdruck<sup>55</sup> können in Zeichenkettenkonstanten in der Wertemenge die Platzhalter-Zeichen `'*'` und `'?'` verwendet werden. Das Zeichen `'*'` ersetzt dabei beliebig viele (0-n) Zeichen und das Zeichen `'?'` genau ein Zeichen des Vergleichsoperanden auf der linken Seite.

Beispiele:

- Der Vergleich mit `IN ('F*O')` ist wahr, wenn der Wert des linken Operanden mit dem Zeichen `'F'` beginnt und mit dem Zeichen `'O'` endet. Dazwischen können sich beliebig viele andere Zeichen befinden (auch keins).
- Der Vergleich mit `IN ('F?O')` ist wahr, wenn der Wert des linken Operanden mit dem Zeichen `'F'` beginnt, mit dem Zeichen `'O'` endet, und sich dazwischen genau ein weiteres Zeichen befindet.

Die Platzhalter-Ersetzung wird jedoch nur wirksam, wenn sie durch das Feld *PlaceHolderOn* in der Versionsinformationstabelle (Abschn. 2.23) frei geschaltet ist. (Anderenfalls sind die beiden obigen Beispielsvergleiche nur wahr, wenn der linke Operand genau die Zeichenkette `'F*O'` bzw. `'F?O'` enthält.)

Werden die Ersetzungszeichen (`'*'`, `'?'`) auch als reguläre Zeichen in Werten von Zeichenketten-Merkmalen verwendet, so kann bei Platzhalter-Freischaltung ein exakter Vergleich mit Hilfe der normalen Vergleichsoperatoren vorgenommen werden, da dort generell keine Platzhalter-Ersetzung stattfindet.

- Optionaler `Objects`-Teil in Constraints:

Fehlt der `Objects`-Teil, so beziehen sich die in den anderen Teilen des Constraints verwendeten Bezeichner auf die gleichnamigen Merkmale des Artikels.

<sup>54</sup>alternativ kann die Form `>=` verwendet werden

<sup>55</sup>also nicht, wenn der IN-Ausdruck in einem Constraint zur Wertebereichseinschränkung verwendet wird



- Logische Ausdrücke im **Restrictions**-Teil von Constraints:

Bei Wertevergleichen im **Restrictions**-Teil sind generell logische Ausdrücke erlaubt (nicht nur der Gleichheitsoperator wie in der Sprachdefinition **OCD\_3**). Das Constraint schlägt fehl, wenn die durch den Ausdruck bestimmte Bedingung nicht eindeutig erfüllt ist.

Die in dem Ausdruck referenzierten Merkmale dürfen nicht im evtl. vorhandenen **Inferences**-Teil des Constraints angeführt sein!

- Ausgabe von Meldungen an den Anwender:

In den Beziehungsarten *Aktion*, *Reaktion* und *Post-Reaktion* des Verwendungsgebiets C (Konfiguration) kann die Funktion

`USER_MESSAGE(<Textnummer>)`

verwendet werden. Diese gibt die in der Texttabelle **UserMessage** (s.Abschn. 2.20) unter der angegebenen Textnummer (ID) hinterlegte Nachricht an den Anwender aus. (Die Textnummer ist dabei als Zeichenkettenkonstante anzugeben, d.h. in Hochkommata einzuschließen.)

Dies kann dazu genutzt werden, bei komplexem Beziehungswissen dem Anwender Hilfestellung zu geben.

Bei der Ausgabe wird die Sprache verwendet, die aktuell in der Applikation eingestellt ist. Ist für diese Sprache in der Tabelle **UserMessage** unter der angegebenen Textnummer kein Text hinterlegt, erfolgt auch keine Ausgabe einer Nachricht.

Wie bei Zuweisungen kann dem Aufruf der Funktion `USER_MESSAGE()` nach dem Schlüsselwort **IF** eine Bedingung folgen, die erfüllt sein muss, damit die Ausgabe der Nachricht erfolgt.

- Abbruch der Auswertung von Beziehungswissen:

In den Beziehungsarten *Reaktion* (für ein Merkmal) und *Post-Reaktion* kann die Funktion

`ABORT()`

verwendet werden, um die Auswertung von Beziehungswissen nach einer Merkmalsänderung abzubrechen. (Der Konfigurationszustand des Artikels ist danach derselbe wie vor der versuchten Merkmalsänderung.)

Dies kann bei Merkmalen zur freien Eingabe von Werten durch den Anwender sinnvoll/notwendig sein, wenn nicht alle Eingaben zulässig sind. Typischerweise ist der Funktion eine **IF**-Klausel nachgestellt, welche die Bedingung formuliert, die eindeutig erfüllt sein muss, damit die Funktion ausgeführt wird.

- Teilzeichenketten-Generierung:

Die Funktion `SUBSTR(<Zeichenkette>, <Position> [, <Laenge>])` liefert als Ergebnis eine Zeichenkette, deren Inhalt der Teilzeichenkette der im ersten Parameter übergebenen Zeichenkette entspricht, die bei der im 2. Parameter angegebenen Position beginnt und die im 3. Parameter übergebene (maximale) Länge besitzt.

Im ersten Parameter kann ein Ausdruck angegeben werden, der als Ergebnis eine Zeichenkette liefert. (Die Funktion selber kann auch in Zeichenketten-Ausdrücken verwendet werden.)

Die Nummerierung der Zeichenpositionen beginnt dabei mit 0. Ergibt die Summe von Position (2. Parameter) und Länge (3. Parameter) eine Position, die über die Länge der übergebenen Zeichenkette hinausreicht, so wird die Teilzeichenkette ab der angegebenen Position bis zum Ende der übergebenen Zeichenkette gebildet.

Der Längenparameter ist optional. Ist er nicht angegeben, so wird die Teilzeichenkette ab der angegebenen Position bis zum Ende der übergebenen Zeichenkette gebildet.

Im Fall von folgenden Fehlern in den Parametern liefert die Funktion eine leere Zeichenkette:

- Der erste Parameter ist keine Zeichenkette.
- Die Werte für Positions- und/oder Längenparameter sind nicht ganzzahlig.
- Die Werte für Positions- und/oder Längenparameter sind negativ.
- Die Position überschreitet die Länge der Zeichenkette.

- Es stehen folgende Funktionen zur Verarbeitung von Zeichenketten zur Verfügung:
  - Die Funktion `SIZE(<Zeichenkette>)` liefert als Ergebnis die Anzahl der Zeichen, die die im Parameter übergebene Zeichenkette enthält.
  - Die Funktion `ToUPPER(<Zeichenkette>)` liefert als Ergebnis eine Zeichenkette, in der alle Kleinbuchstaben der im Parameter übergebenen Zeichenkette in Großbuchstaben umgewandelt sind. Das deutsche Eszett ('ß') wird dabei nicht umgewandelt.
  - Die Funktion `ToLOWER(<Zeichenkette>)` liefert als Ergebnis eine Zeichenkette, in der alle Großbuchstaben der im Parameter übergebenen Zeichenkette in Kleinbuchstaben umgewandelt sind.
  - Die Funktion `TRIM(<Zeichenkette>)` liefert als Ergebnis eine Zeichenkette, in der alle Leerzeichen am Anfang und am Ende der im Parameter übergebenen Zeichenkette entfernt sind.
  - Die Funktion `RTRIM(<Zeichenkette>)` liefert als Ergebnis eine Zeichenkette, in der alle Leerzeichen am Ende der im Parameter übergebenen Zeichenkette entfernt sind.
  - Die Funktion `LTRIM(<Zeichenkette>)` liefert als Ergebnis eine Zeichenkette, in der alle Leerzeichen am Anfang der im Parameter übergebenen Zeichenkette entfernt sind.

Im Zeichenketten-Parameter kann ein Ausdruck angegeben werden, der als Ergebnis eine Zeichenkette liefert. (Die Funktionen selber können auch in Zeichenketten-Ausdrücken verwendet werden.)

- Es stehen folgende Funktionen zur Umwandlung von Zeichenketten in numerische Werte zur Verfügung:
  - Die Funktion `FLOAT(<Ausdruck>, <Fallback>)` liefert als Ergebnis eine gebrochene Zahl in Abhängigkeit vom Typ des im ersten Parameter übergebenen Ausdrucks:
    - \* Ergibt der Ausdruck selber eine gebrochene Zahl, so ist diese auch das Ergebnis der Funktion.
    - \* Ergibt der Ausdruck eine ganze Zahl, so ist das Ergebnis die entsprechende gebrochene Zahl, d.h. mit gebrochenem Teil 0.
    - \* Ergibt der Ausdruck eine Zeichenkette, so wird versucht, diese in eine gebrochene Zahl umzuwandeln. Im Erfolgsfall ist diese das Ergebnis der Funktion, anderenfalls der im zweiten Parameter angegebene Fallback-Wert. Für die Darstellung der Zahl in der Zeichenkette gelten dieselben Bestimmungen wie für die Darstellung von gebrochenen Zahlen in der Merkmalswerttabelle, d.h., als Dezimaltrennzeichen wird der Punkt ('.') verwendet, ein Tausendertrennzeichen wird *nicht* verwendet und an erster Stelle kann ein Minuszeichen stehen.

Ist der 2. Parameter keine konstante gebrochene Zahl, so wird ein Syntaxfehler ausgelöst (und die Auswertung der Beziehung abgebrochen).
  - Die Funktion `INT(<Ausdruck>, <Fallback>)` liefert als Ergebnis eine ganze Zahl in Abhängigkeit vom Typ des im ersten Parameter übergebenen Ausdrucks:
    - \* Ergibt der Ausdruck selber eine ganze Zahl, so ist diese auch das Ergebnis der Funktion.
    - \* Ergibt der Ausdruck eine gebrochene Zahl mit gebrochenem Teil 0, so ist das Ergebnis der ganzzahlige Teil.
    - \* Ergibt der Ausdruck eine gebrochene Zahl mit einem gebrochenem Teil ungleich 0, so ist das Ergebnis der im zweiten Parameter angegebene Fallback-Wert.
    - \* Ergibt der Ausdruck eine Zeichenkette, so wird versucht, diese in eine ganze Zahl umzuwandeln. Im Erfolgsfall ist diese das Ergebnis der Funktion, anderenfalls der im zweiten Parameter angegebene Fallback-Wert. Für die Darstellung der Zahl in der Zeichenkette gelten dieselben Bestimmungen wie für die Darstellung von ganzen Zahlen in der Merkmalswerttabelle, d.h., ein Tausendertrennzeichen wird *nicht* verwendet und an erster Stelle kann ein Minuszeichen stehen.

Ist der 2. Parameter keine konstante ganze Zahl, so wird ein Syntaxfehler ausgelöst (und die Auswertung der Beziehung abgebrochen).

- Sichtbarkeit von Merkmalen:

Die Funktion

`SET_VISIBILITY(<Merkmal>, <logischer Ausdruck>)`

kann in Beziehungen der Arten *Aktion*, *Reaktion* und *Post-Reaktion* dazu verwendet werden, die Sichtbarkeit eines Merkmals für den Anwender einer OFML-Applikation zu steuern:

- Liefert der logische Ausdruck den Wert `false`, wird das angegebene Merkmal für den Anwender unsichtbar geschaltet.
- Liefert der logische Ausdruck den Wert `true`, wird das angegebene Merkmal für den Anwender sichtbar geschaltet.
- Liefert der logische Ausdruck ein undefiniertes Ergebnis, hat der Funktionsaufruf keinen Effekt (d.h., der Sichtbarkeitszustand des Merkmals ändert sich nicht).

Der Funktionsaufruf kann mit einer Bedingung versehen werden. Diese ist nach dem Schlüsselwort `IF` anzugeben. Der Funktionsaufruf findet dann nur statt, wenn die Bedingung eindeutig erfüllt ist.

Die Sichtbarkeitssteuerung mittels `SET_VISIBILITY()` ist eine Alternative zur Verwendung von *Vorbedingungen*:

Die Funktion `SET_VISIBILITY()` wirkt sich nur auf die Sichtbarkeit des Merkmals für den Anwender einer OFML-Applikation aus (Eigenschaftseditor, Variantentext), in technischen Kontexten<sup>56</sup> steht es (mit dem zuletzt zugewiesenen Wert) jedoch noch zur Verfügung.

Ein per Vorbedingung ausgeblendetes Merkmal hingegen ist komplett ungültig (unsichtbar) und steht auch nicht in technischen Kontexten zur Verfügung.

Eine Konsequenz aus der Semantik von Vorbedingungen ist damit auch, dass der Aufruf von `SET_VISIBILITY()` keinen Effekt hat, wenn das angegebene Merkmal aktuell per Vorbedingung ausgeblendet ist<sup>57</sup>! Eine Kombination von Vorbedingungen und `SET_VISIBILITY()` für ein und dasselbe Merkmal ist somit nicht sinnvoll und nicht zu empfehlen.

- Änderung von Steuerkategorien:

Mit Hilfe der Funktion

`SET_TAX_CATEGORY(<Steuerart>, <Steuerkategorie>)`

kann in `TAX`-Beziehungen (s. Abschn. 2.15) die dem Artikel über sein Besteuerungsschemata zugeordnete Steuerkategorie in der angegebenen Steuerart geändert (überschrieben) werden.

Typischerweise wird eine derartige Änderung mittels einer `IF`-Klausel an eine Bedingung geknüpft<sup>58</sup>. (Die Bedingung muß eindeutig erfüllt sein, damit der Funktionsaufruf stattfindet.)

Als Bezeichner für Steuerart und Steuerkategorie sind die im Anhang H vordefinierten Bezeichner zu verwenden.

`TAX`-Beziehungen können an folgende Entitäten gebunden werden. Die Auswertung erfolgt dabei in der angegebenen Reihenfolge:

1. Artikel
2. Merkmalsklassen des Artikels
3. aktuell bewertete Merkmale des Artikels<sup>59</sup>
4. Werte der aktuell bewerteten Merkmale

Finden Aufrufe der Funktion für eine Steuerart in mehreren Beziehungen statt, so gilt die Zuweisung des jeweils letzten Aufrufs.

<sup>56</sup> Auswertung von Beziehungswissen, Grafikdaten, Variantencodes und sonstige technische Beschreibungen der Konfiguration eines Artikels

<sup>57</sup> D.h., das Merkmal hat, wenn es per Vorbedingung wieder gültig wird, denselben Sichtbarkeitsstatus wie zu dem Zeitpunkt, als es per Vorbedingung ausgeblendet wurde.

<sup>58</sup> Ein Beispiel wäre die Änderung der Materialkategorie in der Steuerart `ECO_FR` in Abhängigkeit von einer bestimmten Merkmalsausprägung.

<sup>59</sup> das schließt nicht-konfigurierbare Merkmale mit ein

## E Sprachdefinition SAP\_LOVC

Diese Sprachdefinition bezieht sich auf die Sprache, welche in der Variantenkonfiguration des SAP ERP (Teil des Logistikmoduls) zur Codierung von Beziehungswissen verwendet wird. Standardmäßig werden alle Sprachkonstrukte unterstützt, die syntaktisch und semantisch mit den gleichnamigen Konstrukten aus den OCD-Sprachdefinitionen identisch sind<sup>60</sup>

Demnach werden (aktuell) folgende Sprachelemente und –konstrukte unterstützt:

- Beziehungsarten: *Vorbedingung*, *Auswahlbedingung*, *Prozedur*<sup>61</sup>, *Constraint*
- Bedingungsteil: IF
- Logische Operatoren: AND, OR, NOT
- Built-in Bedingungen: IN, SPECIFIED
- Vergleichsoperatoren: < oder LT, <= oder LE, = oder EQ, <> oder NE, >= oder GE, > oder GT
- Arithm. Operatoren: +, -, /, \*
- Arithm. Funktionen: sqrt(), abs(), sign(), frac(), trunc(), ceil(), floor()
- Preisfaktoren: SET\_PRICING\_FACTOR()<sup>62</sup>
- Auswertung von Variantentabellen: TABLE()

Der Tabellenaufruf in OCD-Aktionen<sup>63</sup> (s. Abschn. B.2.2) weist folgende Besonderheit auf:

Da die im SAP hinterlegten Auswertungsalternativen im OCD nicht bekannt sind, muß auf andere Weise bestimmt werden, welche Merkmale als Schlüssel für den Tabellenzugriff dienen, und für welche Merkmale eine Herleitung eines Wertes stattfinden soll. Im OCD erfolgt dies, indem die Merkmale, für die eine Werteherleitung stattfinden soll, mit `$self` qualifiziert werden. Das bedeutet im Umkehrschluß, dass explizite Schlüsselmerkmale *nicht* mit `$self` qualifiziert werden dürfen<sup>64</sup>!

---

<sup>60</sup>Darüber hinaus können die Anbieter von OCD-Implementierungen weitere Sprachkonstrukte unterstützen. Diese sind bei Bedarf von dem entsprechenden Anbieter der jeweiligen OFML-Anwendung zu erfragen.

<sup>61</sup>wird nur in dem Umfang unterstützt, wie er für die OCD-Beziehungsart *Aktion* definiert ist

<sup>62</sup>etwas abweichende Signatur (s. Abschn. 3.4)

<sup>63</sup>SAP: Prozeduren

<sup>64</sup>In der Variantenkonfiguration des SAP ERP wird für diese Merkmale dann der Qualifizierer `$root` angenommen.

## F Arithmetische Funktionen in Beziehungswissen

Folgende Funktionen können in arithmetischen Ausdrücken im Code von Beziehungen verwendet werden<sup>65</sup>.

Die Argumente selber können arithmetische Ausdrücke sein. Ist ein Argument ein undefinierter arithmetischer Ausdruck, so ist das Ergebnis der Funktion ebenfalls undefiniert.

Bei ungültigen Argumenten wird die Auswertung der Beziehung abgebrochen.

$pow(x(Float), y(Int)) \rightarrow Float$

Die Funktion `pow()` berechnet  $x$  hoch  $y$ . Wenn  $x$  negativ ist, muss  $y$  ein ganzzahliger Wert sein. Wenn  $x$  0 ist, muss  $y$  positiv sein. Das Ergebnis ist 1.0, wenn sowohl  $x$  und  $y$  0 sind.

$sqrt(x(Float)) \rightarrow Float$

Die Funktion `sqrt()` berechnet die nicht-negative Quadratwurzel von  $x$ .  $x$  darf nicht negativ sein.

$fabs(x(Float)) \rightarrow Float$

Die Funktion `fabs()` berechnet den Betrag von  $x$ .

$ceil(x(Float)) \rightarrow Float$

Die Funktion `ceil()` berechnet den kleinsten ganzzahligen Wert, der nicht kleiner als  $x$  ist.

$floor(x(Float)) \rightarrow Float$

Die Funktion `floor()` berechnet den größten ganzzahligen Wert, der nicht größer als  $x$  ist.

$sign(x(Float)) \rightarrow Int$

Die Funktion `sign()` liefert das Vorzeichen (-1 bzw. +1) von  $x$ .

$trunc(x(Float)) \rightarrow Float$

Die Funktion `trunc()` liefert den ganzzahligen Teil von  $x$ .

$frac(x(Float)) \rightarrow Float$

Die Funktion `frac()` liefert den Dezimalteil von  $x$ .

---

<sup>65</sup>Bis auf die Funktionen `sign`, `trunc` und `frac` gehören alle Funktionen auch zu den arithmetischen Standardfunktionen von OFML.

## G Reservierte Schlüsselwörter

Im folgenden werden für jede Sprachdefinition die reservierten Schlüsselwörter aufgelistet.

Diese dürfen bei Verwendung der jeweiligen Sprachdefinition nicht als Bezeichner für Merkmale verwendet werden!<sup>66</sup>

Zur Beachtung:

Hier sind die Schlüsselwörter nur in kompletter Großschreibung angeführt. Da in den OCD-Beziehungen jedoch generell Groß-/Kleinschreibung ignoriert wird (s. A), dürfen auch die gemischten Schreibweisen nicht als Bezeichner für Merkmale verwendet werden.

Sprachdefinition **OCD\_1**:

- AND
- BAN
- EQ
- GE
- GT
- IF
- IN
- LE
- LT
- NOT
- OR
- SET\_PCKG\_FACTOR
- SET\_PRICING\_FACTOR
- SPECIFIED

Sprachdefinition **OCD\_2**:

alle Schlüsselwörter aus OCD\_1 sowie:

- CONDITION
- FALSE
- INFERENCES
- IS\_A
- OBJECTS
- RESTRICTIONS
- TABLE
- WHERE

Sprachdefinition **OCD\_3**: wie OCD\_2

---

<sup>66</sup>Darüberhinaus können Softwareanbieter von OCD-Implementierungen weitere Beschränkungen definieren. Diese sind beim jeweiligen Softwareanbieter zu erfragen.

Sprachdefinition **OCD\_4**:

alle Schlüsselwörter aus OCD\_3 sowie:

- ABORT
- USER\_MESSAGE

Sprachdefinition **SAP\_LOVC**:

alle Schlüsselwörter aus OCD\_2 (außer BAN und SET\_PCKG\_FACTOR) sowie alle Schlüsselwörter gemäß der SAP ERP-Spezifikation

## H Steuerarten und Steuerkategorien

Diese Tabelle enthält die — innerhalb von OCD — standardisierten Steuerarten und zugehörigen Steuerkategorien<sup>67</sup>.

Steuerart (Erklärung)	Steuerkategorie	Erklärung
VAT (Mehrwertsteuer, value added tax)	standard_rate	Normalsatz
	reduced_rate	ermäßigter Satz
	super_reduced_rate	stark ermäßigter Satz
	parking_rate	Zwischensatz
	services	Dienstleistungen
	zero_rate	Nullsatz
	exemption	steuerbefreit
ECO_FR (ECO-Tax Frankreich)	seat_metal	Sitzplatz, mehrheitlich aus Metall bestehend
	seat_metal95	Sitzplatz, zu mehr als 95% aus Metall bestehend
	seat_wood	Sitzplatz, mehrheitlich aus Holz bestehend
	seat_plastics	Sitzplatz, mehrheitlich aus Plastik bestehend
	seat_other	Sitzplatz, sonstige Materialien
	storage_metal	Aufbewahrung, mehrheitlich aus Metall bestehend
	storage_metal95	Aufbewahrung, zu mehr als 95% aus Metall bestehend
	storage_wood	Aufbewahrung, mehrheitlich aus Holz bestehend
	storage_plastics	Aufbewahrung, mehrheitlich aus Plastik bestehend
	storage_other	Aufbewahrung, sonstige Materialien
	workplace_metal	Arbeitsplatz, mehrheitlich aus Metall bestehend
	workplace_metal95	Arbeitsplatz, zu mehr als 95% aus Metall bestehend
	workplace_wood	Arbeitsplatz, mehrheitlich aus Holz bestehend
	workplace_plastics	Arbeitsplatz, mehrheitlich aus Plastik bestehend
	workplace_other	Arbeitsplatz, sonstige Materialien
	other_metal	Sonstiges Möbel, mehrheitlich aus Metall bestehend
	other_metal95	Sonstiges Möbel, zu mehr als 95% aus Metall bestehend
	other_wood	Sonstiges Möbel, mehrheitlich aus Holz bestehend
	other_plastics	Sonstiges Möbel, mehrheitlich aus Plastik bestehend
	other_other	Sonstiges Möbel, sonstige Materialien
none	nicht steuerpflichtig	

Anmerkungen zur ECO-Tax Frankreich<sup>68</sup>:

- Das Non-Profit Unternehmen **Valdelia**<sup>69</sup> ist von der französischen Regierung bevollmächtigt, die rechtlichen Bestimmungen zur Öko-Steuer im B2B-Geschäft von professionellen Objekt- und Büromöbeln umzusetzen<sup>70</sup>. Zweck der Steuer in dieser Branche ist die Beteiligung der Käufer an der Entsorgung und Verwertung von Möbel-Sperrmüll.
- Die Steuer ist abhängig vom Netto-Gewicht des Artikels und seiner Steuerkategorie, welcher ein Preisfaktor zugeordnet ist. Das Netto-Gewicht multipliziert mit diesem Preisfaktor ergibt die Steuer. Die Preisfaktoren variieren jährlich und werden im Oktober bekanntgegeben. Die Steuer wird erstmalig seit dem 1. März 2013 erhoben.

<sup>67</sup>Anträge zur Aufnahme weiterer Steuerarten und Steuerkategorien sind an das OFML-Normungsgremium zu richten.

<sup>68</sup>Die Mehrwertsteuer wird auch auf die ECO-Tax angerechnet und die Steuer ist nicht rabattierbar.

<sup>69</sup>[www.valdelia.org](http://www.valdelia.org)

<sup>70</sup>Für das Recycling von Möbeln, die für Privatkunden bestimmt sind (B2C-Geschäft), ist die Organisation *éco-mobilier* verantwortlich. Deren Steuermodell ist (noch) nicht Gegenstand der OFML-Normung.



- Die oben genannten Steuerkategorien entsprechen den Steuerkategorien gemäß dem Dokument „Barème eco-contribution“<sup>71</sup>. Die Steuerkategorie ist eine Kombination aus Möbeltyp und Materialkategorie<sup>72</sup>. Die Zuordnung zu einer Materialkategorie ergibt sich aus dem Gewichtsanteil. Machen z.B. Materialien aus Holz mehr als 50 Prozent des Gewichts eines Artikels aus, so wird er der Materialkategorie *wood* zugeordnet. Hat keine Materialart die Mehrheit am Gewicht, so ist der Artikel der Kategorie *other* zuzuordnen.

---

<sup>71</sup>herausgegeben von Valdelia

<sup>72</sup>Die Materialkategorie *meta195* ist ab dem 1.7.2014 gültig.

# I Begriffe

- **EAN**

- Abkürzung für European Article Numbering
- 1977 gegründete Non-Profit Organisation mit dem Ziel der Standardisierung von Identifikationen von Produkten, Einheiten, Einrichtungen usw. zur effizienten Abwicklung von Geschäftsprozessen im Handel.
- Seit 1992, nach dem Beitritt von Mitgliedern aus anderen Kontinenten, als EAN International tätig.
- Zusammen mit dem Uniform Code Council (UCC)<sup>73</sup>, der entsprechenden Behörde für Nordamerika, wurde das EAN.UCC-System zur Identifizierung von Produkten (→ GTIN), Lokationen (→ GLN) usw. entwickelt.

- **GTIN**

- Abkürzung für Global Trade Item Number
- Eindeutig im Rahmen des → EAN.UCC-Systems vergebene Identifikationsnummer für eine Ware (Produkt oder Dienstleistung), die innerhalb von Geschäftsprozessen im Handel bestellt und verrechnet werden kann.
- Innerhalb des EAN.UCC-Systems können dabei verschiedene definierte Nummerierungsschemata verwendet werden (EAN.UCC-14, EAN.UCC-13, EAN.UCC-8).

- **GLN**

- Abkürzung für Global Location Number
- Eindeutig im Rahmen des → EAN.UCC-Systems vergebene 13-stellige Identifikationsnummer für physische und elektronische Adressen von Unternehmen, Tochterunternehmen, Niederlassungen sowie organisatorisch relevante Betriebsteile

- **Intrastat**

- Intrastat ist eine vom Statistischen Bundesamt in Wiesbaden geführte Statistik, die den innereuropäischen Handel mit Deutschland umfaßt.
- Jeder deutsche Betrieb, der Handel innerhalb Europas betreibt, muß dies nach genau festgelegten Vorgaben melden. Diese Meldung hat monatlich zu erfolgen.
- Jede Ware besitzt eine achtstellige Nummer, die im Warenverzeichnis für den Außenhandel aufgeführt ist, und die zusammen mit Gewicht, Wert, Transportweg etc. angegeben werden muß.

- **Zolltarifnummer**

- Zolltarifnummern kommen bei der Abwicklung von Geschäften mit Ländern zur Anwendung, die nicht zur EU gehören.
- Ein deutscher Exporteur muss ab einem Warenwert von 1.000 EUR eine schriftliche Ausfuhranmeldung für die Zollbehörden und das Statistische Bundesamt ausfüllen.
- Zur Anmeldung jeder Ware ist eine Warentarifnummer erforderlich. Um die Zuordnung zu ermöglichen, ist eine präzise Deklaration der Waren gemäß dem *Warenverzeichnis für die Außenhandelsstatistik* notwendig.

---

<sup>73</sup>seit 2005 unter *GS1* (Global Standards One) firmierend

## J Änderungshistorie

### J.1 OCD 4.3 vs. OCD 4.2

- Korrektur zur Mehrwertigkeit von Merkmalen in Abschn. 2.9.
- Einführung des Konzepts der *Merkmalsgruppen* (Abschn. 2.11).
- Post-Reaktionen können nun auch für Artikel angegeben werden (Abschn. 2.15).
- Mit Hilfe von TAX-Beziehungen können Steuerkategorien abhängig von bestimmten Artikelvarianten definiert werden (s. Abschnitte 2.15 und 2.25).
- Aufhebung der Beschränkung der Zeilenlänge in den Text-Tabellen (Abschn. 2.20).
- Die alternativen Text-Tabellen wurden entfernt (Abschn. 2.20).
- In der Sprachdefinition OCD\_2 ist nun auch die boolesche Konstante TRUE definiert (Anhang B).
- Die Sprachdefinition OCD\_4 enthält die neuen Funktionen SET\_VISIBILITY zur Steuerung der Sichtbarkeit eines Merkmals für den Anwender einer OFML-Applikation sowie SET\_TAX\_CATEGORY zur Zuordnung einer vom Besteuerungsschema abweichenden Steuerkategorie (Anhang D).
- Die Sprachdefinitionen SAP\_3\_1 und SAP\_4\_6 wurden zu SAP\_LOVC zusammengefasst.