

Spezifikation  
OFML Artikel-Mappings\*  
(OFML Part VI)

Version 1.0

Status: Release

Thomas Gerth, EasternGraphics GmbH (Autor und Editor)

2004-09-30

---

\*Copyright © 2003, 2004 Verband der Büro-, Sitz- und Objektmöbelhersteller e.V.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Überblick</b>	<b>3</b>
<b>2</b>	<b>Allgemeine Festlegungen</b>	<b>4</b>
2.1	Lexikalische und syntaktische Festlegungen . . . . .	4
2.2	Verzeichnisstruktur . . . . .	5
<b>3</b>	<b>Die Tabellen</b>	<b>7</b>
3.1	Konzeptionelles Datenmodell . . . . .	7
3.2	Übersicht . . . . .	8
3.3	Das Artikel-OFML-Mapping . . . . .	8
3.4	Das Artikel-ODB-Parameter-Mapping . . . . .	10
3.5	Das Merkmal-Material-Mapping . . . . .	11

# 1 Einleitung und Überblick

Die in diesem OFML-Part beschriebenen und definierten Tabellen dienen zur Festlegung von Zusammenhängen zwischen Daten, die gemäß der Spezifikation verschiedener anderer OFML-Parts angelegt wurden und jeweils unterschiedliche Aspekte von Artikeln beschreiben. Abb. 1 veranschaulicht diese Zusammenhänge.

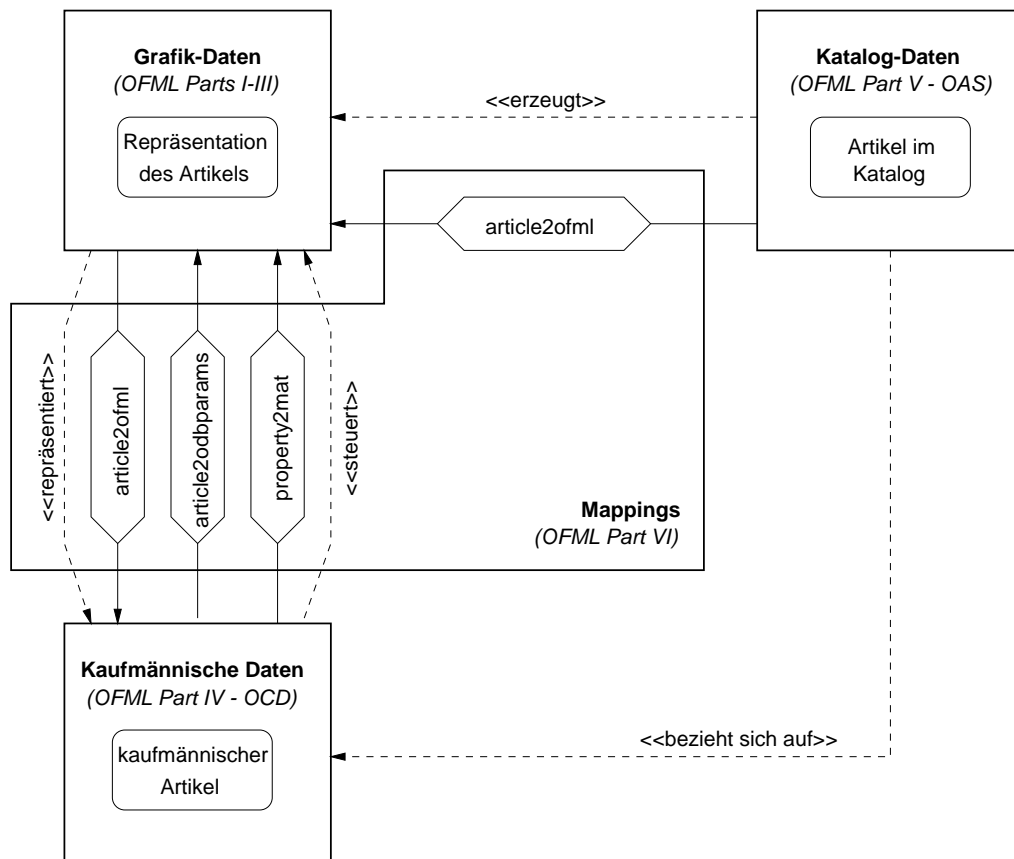


Abbildung 1: Stellung der Mapping-Tabellen zwischen den OFML-Parts

Es werden folgende Mappings verwendet:

- *Artikel-OFML-Mapping (article2ofml)*  
Legt fest, durch welchen OFML-Typ mit welchen Parametern ein Artikel in einer OFML-Laufzeitumgebung repräsentiert werden soll.
- *Artikel-ODB-Parameter-Mapping (article2odbparams)*  
Legt für eine gegebene Artikelkonfiguration die Werte von Hilfsvariablen fest, die in den ODB-Daten zur grafischen Darstellung des Artikels verwendet werden (optional).
- *Merkmal-Material-Mapping (property2mat)*  
Legt fest, welche Merkmalsausprägungen welchen Material-Layer-Belegungen entsprechen (optional).

## 2 Allgemeine Festlegungen

### 2.1 Lexikalische und syntaktische Festlegungen

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 „oam\_“, den spezifizierten Tabellennamen und den Suffix „.csv“ gebildet.
- 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.
- 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)

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.

---

<sup>1</sup>Da bei CSV-Datensätzen formal keine Beschränkungen der einzelnen Feldlängen bestehen, werden hier bei Feldern des Datentyps **Char**, wo möglich, lediglich bestimmte, aus praktischer Sicht maximal sinnvolle Längen angegeben.

## Params Parameterliste

Es gelten folgende lexikalischen und syntaktischen Bestimmungen:

1. Ein Parameter wird wie folgt spezifiziert:  
<Parameter-Bezeichner>=<Parameter-Wert>  
Die einzelnen Parameter in der Liste sind durch Komma zu trennen.
2. Für den Parameter-Bezeichner sind alle alphanumerischen Zeichen inklusive dem Unterstrich „\_“ erlaubt, wobei der Bezeichner mit einem Buchstaben oder einem Unterstrich beginnen muß.
3. Als Parameter-Wert können ganzzahlige und Gleitkomma-Konstanten sowie literale Symbole und Zeichenkettenkonstanten angegeben werden. Bezüglich der genauen Syntax von ganzzahligen und Gleitkomma-Konstanten sei auf die OFML-Sprachbeschreibung im OFML-Part III verwiesen. Für literale Symbole gelten dieselben Bestimmungen wie für die Bezeichner oben, sie beginnen jedoch immer mit dem Zeichen „@“ Für Zeichenkettenkonstanten gilt die von der OFML-Sprachbeschreibung abweichende Regelung, dass diese in Hochkommata „'“ einzuschliessen sind, nicht in Anführungszeichen „““. Einem in der Zeichenkette selber vorkommenden Hochkomma ist ein Backslash „\“ voranzustellen. Einem in der Zeichenkette vorkommenden Anführungszeichen muß kein Backslash vorangestellt werden, es gelten jedoch dieselben Regelungen wie beim Feldtyp *Date* oben für den Fall, dass die Zeichenkette ein Semikolon enthält.

## 2.2 Verzeichnisstruktur

Die konkrete Verzeichnisstruktur für die Ablage von Objekt-, Katalog- und kaufmännischen Daten wird von den jeweiligen OFML-Applikationen festgelegt. Dabei gilt jedoch generell, dass die Grafik-Daten für eine oder mehrere kaufmännische Serien eines Herstellers in einem OFML-Paket (Verzeichnis) enthalten sind, während für die Katalog- und kaufmännischen Daten separate Datenbanken (Verzeichnisse) je Vertriebsgebiet angelegt werden, siehe Abb. 2.

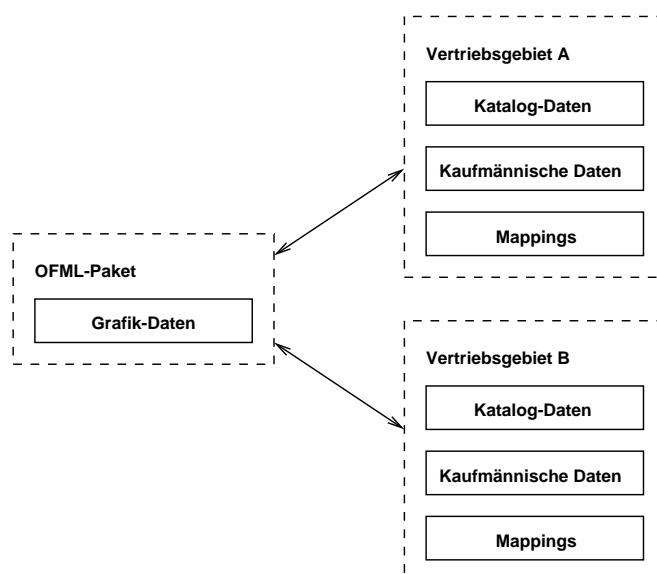


Abbildung 2: Verzeichnisstruktur

Bezüglich dieser allgemeinen Verzeichnisstruktur wird festgelegt, dass die Mappings prinzipiell parallel zu den kaufmännischen Daten je Vertriebsgebiet angelegt werden. Zwar besteht auch eine Abhängigkeit der Mappings von den Grafik-Daten (was eine Ablage der Mappings parallel zu diesen motivieren würde), die Abhängigkeit der Mappings von den kaufmännischen Daten ist jedoch zwingender, da in verschiedenen Vertriebsgebieten unterschiedliche Sets von Artikeln relevant sein können bzw. sich sogar die Artikelnummern von Grundartikeln unterscheiden können, und da die Wahrscheinlichkeit und Häufigkeit von Änderungen in den kaufmännischen Daten größer ist als die von Änderungen in den Grafik-Daten.

In vielen Fällen werden sich die Mappings von Vertriebsgebiet zu Vertriebsgebiet jedoch nicht unterscheiden. Zur Verringerung des Datenvolumens bei der Auslieferung muß die OFML-Applikation deswegen das spezielle Vertriebsgebiet `COMMON` unterstützen. Die Applikation nutzt dann die Mappings aus diesem Vertriebsgebiet, wenn sie nicht in dem gerade aktiven Vertriebsgebiet enthalten sind.

### 3 Die Tabellen

#### 3.1 Konzeptionelles Datenmodell

Abb. 3 veranschaulicht das konzeptionelle Datenmodell der durch die Mappings verbundenen OFML-Parts und soll die Zielstellung und Wirkungsweise der einzelnen Mappings verdeutlichen<sup>2</sup>. Das Modell stellt eine Konkretisierung der Übersicht in Abb. 1 in Abschn. 1 dar.

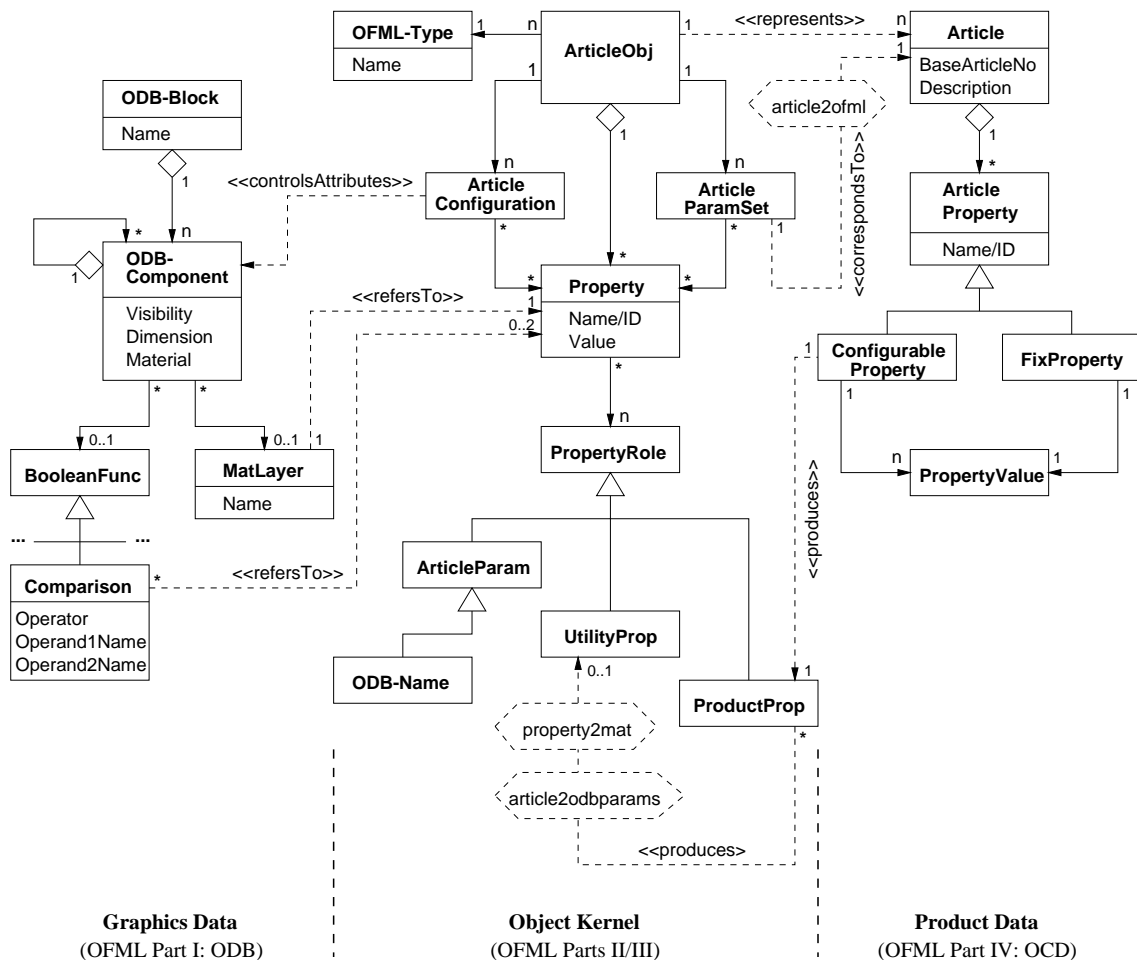


Abb. 3

<sup>2</sup>Die in diesem Diagramm verwendete Notation entspricht bis auf die Darstellung der Mappings der *Unified Modeling Language (UML)* Spezifikation ([www.omg.org/uml](http://www.omg.org/uml)).

## 3.2 Übersicht

<b>article2ofml</b>	<b>article2odparams</b>	<b>property2mat</b>
<b>ArticleID</b> <i>OFMLType</i> <i>ODBName</i> <i>ArtParams</i>	<b>ArticleID</b> <i>VarCodeType</i> <i>VariantCode</i> <i>OdbParams</i>	<b>ArticleID</b> <i>PropertyID</i> <i>PropValueID</i> <i>MatLayer</i> <i>Material</i>

Abbildung 4: Tabellen-Übersicht

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

## 3.3 Das Artikel-OFML-Mapping

Tabellenname: `article2ofml`

Pflichttabelle: ja

Durch dieses Mapping wird festgelegt, durch welchen OFML-Typ mit welchen Parametern ein Artikel in einer OFML-Laufzeitumgebung repräsentiert werden soll.

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ArticleID	X	Char	18	X	Grundartikelnummer
2.	OFMLType		Char	*		vollqualifizierter OFML-Typ
3.	ODBName		Char	*		vollqualifizierter ODB-Name
4.	ArtParams		Params (Char)	*		Artikelparameter

Anmerkungen:

- Die Tabelle kann einen Eintrag mit dem Zeichen '\*' anstelle einer realen Artikelnummer im Feld 1 besitzen. Die Angaben in den anderen Feldern des Eintrags werden dann von der OFML-Laufzeitumgebung bei der Erzeugung von Artikeln herangezogen, für die kein expliziter Eintrag in der Tabelle existiert.
- OFML-Typen und ODB-Namen sind vollqualifiziert anzugeben, d.h. inklusive Bezeichner des OFML-Pakets, welches die Implementierung des Typs enthält.
- Als OFML-Typ kann einer der OFML-Basistypen (siehe OFML-Part III) oder ein Typ aus einem OFML-Erweiterungspaket angegeben werden, wobei der Typ direkt oder indirekt von *OiPElement* oder *OiPart* abgeleitet sein muß. Die Initialisierungsfunktion des Typs darf nur die beiden Standard-Parameter für das Vater-Objekt und den Objektbezeichner besitzen. OFML-Basistypen werden von der OFML-Laufzeitumgebung bereitgestellt. OFML-Erweiterungspakete müssen vom Daten-Hersteller geliefert und von der OFML-Laufzeitumgebung geladen werden.
- Das Feld für den OFML-Typ kann leer sein, wenn vom Daten-Hersteller keine explizite grafische Darstellung des Artikels vorgesehen ist. Von der OFML-Laufzeitumgebung wird dann eine Standard-Darstellung für den Artikel verwendet. (Die nachfolgenden Felder werden in diesem Fall nicht ausgewertet.)



- Wird für die grafische Darstellung des Artikel die ODB (OFML-Part I) verwendet, so ist im Feld 3 der vollqualifizierte Name des ODB-Blockes anzugeben, der in den ODB-Tabellen die 2D- und 3D-Geometrie des Artikels definiert. Der im Feld 2 angegebene OFML-Typ muß in diesem Fall direkt oder indirekt von *OiOdbPElement* abgeleitet sein. Der ODB-Name wird dann von diesem Typ in einer unsichtbaren Property des Artikelobjekts mit dem Key `@ODB_NAME` gespeichert.  
Das Feld 3 ist leer, wenn die Geometrie nicht durch einen ODB-Block beschrieben wird, sondern dies durch den OFML-Typ selber erfolgt. Dies ist nur in der OFML-Ausbaustufe III (Sprache) möglich.
- Die Artikelparameter in Feld 4 werden benötigt, wenn der OFML-Typ mehrere Grundartikel repräsentieren kann, die Grundartikelnummer aber nicht am Artikel-Objekt selber gespeichert wird. Welchen Grundartikel das Artikel-Objekt repräsentiert, ergibt sich dann aus anderweitigen Variablen des internen Objektzustandes. Die Variablen, die eine Zuordnung zu einem Grundartikel ermöglichen, werden als *Artikelparameter* bezeichnet. Dies können Membervariablen und/oder OFML-Properties sein. Jede mögliche Kombination von Werten dieser Artikelparameter entspricht genau einem Grundartikel. Diese Kombination ist im Feld 4 anzugeben.
- Der im Feld 3 angegebene ODB-Name stellt eine spezielle Form eines Artikelparameters dar (s.a. Abb. 3). Dies bedeutet, dass keine weiteren Artikelparameter in Feld 4 notwendig sind, wenn jedem Grundartikel bereits ein eindeutiger ODB-Name zugeordnet ist.

Folgende 2 Beispiele veranschaulichen diesen Sachverhalt:

```
854;::foo::bar::PElement;::foo::bar::T8xx;W=@W1,UA=@U
855;::foo::bar::PElement;::foo::bar::T8xx;W=@W2,UA=@U
884;::foo::bar::PElement;::foo::bar::T8xx;W=@W1,UA=@A
885;::foo::bar::PElement;::foo::bar::T8xx;W=@W2,UA=@A

854;::foo::bar::PElement;::foo::bar::T854;
855;::foo::bar::PElement;::foo::bar::T855;
884;::foo::bar::PElement;::foo::bar::T884;
885;::foo::bar::PElement;::foo::bar::T885;
```

Generell gilt somit, dass die Tabelle eine eindeutige Abbildung von Grundartikelnummer (Feld 1) einerseits auf OFML-Typ, ODB-Name und Artikelparameter (Felder 2-4) andererseits ermöglichen muß. Die umgekehrte Abbildung muß ebenfalls eindeutig sein, wenn die Grundartikelnummer nicht am Artikel-Objekt gespeichert wird oder diese nicht durch eine entsprechend überschriebene Implementierung der Methode *getArticleSpec()* der OFML-Schnittstelle *Article* bereitgestellt wird<sup>3</sup>. In diesem Fall benutzt die Standardimplementierung dieser Methode dieses Mapping, um aus dem aktuellen Objektzustand die aktuelle Grundartikelnummer zu ermitteln.

- Artikelparameter (Feld 4) können auf zwei verschiedene Weisen angegeben und verarbeitet werden:

1. *Parameterliste* (Feldtyp *Params*)

Dies ist die Standard-Angabe, ein Beispiel dafür ist oben gegeben. Die Zuweisung der Parameterwerte erfolgt automatisch durch die OFML-Laufzeitumgebung unmittelbar nach Zuweisung der Grundartikelnummer via Standardmethode *setArticleSpec()*. Das Artikel-Objekt muß zu diesem Zeitpunkt für jeden Artikelparameter eine entsprechende OFML-Property besitzen, deren Key mit dem in der Parameterliste angegebenen Bezeichner übereinstimmt. (Dem Parameter-Bezeichner "W" aus dem Beispiel oben entspricht also eine Property mit dem Key `@W`.)

---

<sup>3</sup>Im Zweifelsfall ist immer eine eindeutige Abbildung zu sichern.

Die Zuweisung des Artikelparameterwertes erfolgt dann mittels der Methode *setProperty()* der OFML-Schnittstelle *Property*. Dabei kommt es zu einem Fehler, wenn der Typ der Property nicht mit dem Datentyp des Wertes aus der Parameterliste übereinstimmt. Es ist undefiniert, ob die OFML-Laufzeitumgebung dann mit der Zuweisung eventuell folgender Artikelparameter fortsetzt, oder ob der Vorgang abgebrochen wird. In jedem Fall befindet sich das Artikel-Objekt danach in einem ungültigen Zustand.

Für eine gegebene Kombination aus OFML-Typ (Feld 2) und ODB-Name (Feld 3) müssen die Artikelparameter in derselben Reihenfolge und Anzahl angegeben werden.

#### 2. Codierung (Feldtyp *Char*)

Wird ein mittels OFML-Ausbaustufe III (Sprache) entwickelter OFML-Typ verwendet, so kann dieser ein eigenes Codierverfahren für die Artikelparameter implementieren. Dies erfolgt mittels spezieller Methoden, die im OFML-Part III beschrieben sind. Das Feld 4 muß in diesem Fall mit dem Zeichen „#“ beginnen, welches selber aber nicht zur Codierung der Artikelparameter gehört.

### 3.4 Das Artikel-ODB-Parameter-Mapping

Tabellenname: `article2odbparams`

Pflichttabelle: nein

Legt für eine gegebene Artikelkonfiguration die Werte von Hilfsvariablen fest, die in den ODB-Daten zur grafischen Darstellung des Artikels verwendet werden.

#### Motivation:

Wie aus Abb. 3 ersichtlich, können in ODB-Daten Variablen verwendet werden, die sich auf Properties des Artikelobjekts beziehen. Von der OFML-Laufzeitumgebung werden diese Variablen bei der Auswertung der ODB-Daten dann durch den aktuellen Wert der Property ersetzt. In der Regel reicht es aus, in den ODB-Daten Variablen zu verwenden, die sich auf ein kaufmännisches Merkmal beziehen (zu dem am Artikelobjekt eine entsprechende Property generiert wird). In bestimmten Situationen kann es jedoch sinnvoll sein, Hilfsvariablen zu verwenden, die aus einem oder mehreren kaufmännischen Merkmalen abgeleitet werden, also einer oder mehreren konkreten Konfigurationsvarianten entsprechen.

Die Tabelle wird für einen Artikel also nur dann benötigt, wenn seine Geometrie durch ODB-Daten beschrieben ist *und* wenn dabei über die kaufmännischen Merkmale hinausgehende Hilfsvariablen verwendet werden.

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ArticleID	X	Char	18	X	Grundartikelnummer
2.	VarCodeType		Char	*		Typ des Variantencodes
3.	VariantCode		Char	*		Variantencode
4.	OdbParams		Params	*	X	ODB-Parameter

Anmerkungen:

- Für einen Grundartikel (Feld 1) kann es ein oder mehrere Einträge in der Tabelle geben. Jeder Eintrag definiert ein separates Set von ODB-Parametern (Feld 4).
- Der Variantencode (Feld 3) bestimmt die Konfigurationen des Artikels, für die das Parameter-Set anzuwenden ist.
- Der Variantencode kann in verschiedenen Formen angegeben werden, weswegen die verwendete Form im Feld 2 angegeben werden muss. Aktuell werden folgende Formen unterstützt:

**FS** Bei dieser Form werden die Konfigurationen, die ein Parameter-Set definieren, anhand der **Endartikelnummer** (Final Article Specification) festgelegt. Diese kann dabei teilbestimmt sein, indem nicht relevante Stellen durch das Fragezeichen „?“ ersetzt werden bzw. die Endartikelnummer nach der letzten relevanten Stelle abgebrochen wird. Eine teilbestimmte Endartikelnummer bezieht sich somit auf eine definierte Anzahl von möglichen Konfigurationen des Artikels.

Zu einer gegebenen, aktuellen Endartikelnummer ermittelt die OFML-Laufzeitumgebung dann den passenden Eintrag. Werden zu einer Endartikelnummer mehrere passende Einträge gefunden, wird der Eintrag mit der höchsten Anzahl an übereinstimmenden Stellen verwendet.

Diese Form kann nicht verwendet werden, wenn die relevanten kaufmännischen Merkmale nicht in den Endartikelnummern codiert sind. Teilbestimmte Endartikelnummern können nicht verwendet werden, wenn die relevanten kaufmännischen Merkmale in den Endartikelnummern für einen Grundartikel an verschiedenen Stellen codiert werden.

In der Zukunft müssen weitere alternative Formen definiert werden, um die genannten Einschränkungen der obigen Form aufzuheben.

- Die Felder 2 und 3 können leer sein, wenn es für den Grundartikel unabhängig von seiner Konfiguration nur ein Set von ODB-Parametern gibt. In diesem Fall enthält die Tabelle auch nur einen Eintrag für den Grundartikel.
- Für jeden ODB-Parameter in der Parameterliste (Feld 4) legt die OFML-Laufzeitumgebung eine entsprechende Property an, die dann für den Zugriff durch die ODB-Daten zur Verfügung steht.

### 3.5 Das Merkmal–Material–Mapping

Tabellenname: `property2mat`

Pflichttabelle: nein

Legt fest, welche Merkmalsausprägungen welchen Material-Layer–Belegungen entsprechen.

Die **Motivation** für dieses Mapping ist ähnlich wie die für das Artikel–ODB-Parameter–Mapping (siehe Abschn. 3.4). *Material-Layer* sind spezifische ODB-Hilfsvariablen, die speziell im Material-Feld der ODB-Tabelle für die 3D-Daten verwendet werden. Konzeptionell wird dadurch das jeweilige ODB-Teilobjekt (Komponente) dem durch die Variable definierten Material-Layer zugeordnet (s.a. Modell in Abb. 3). Bei der Darstellung der 3D-Grafik des Artikels durch die OFML-Laufzeitumgebung wird dann dem ODB-Teilobjekt das Material (Oberfläche) zugewiesen, das aktuell der Property des Artikelobjektes zugewiesen ist, die für das jeweilige Material-Layer angelegt wurde. Mit dem vorliegenden Mapping wird nun festgelegt, welche Material-relevanten kaufmännischen Merkmale sich auf welche Material-Layer auswirken und für welche Werte dieser kaufmännischen Merkmale welche Materialien dem Layer zugewiesen werden sollen.

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ArticleID	X	Char	18	X	Grundartikelnummer
2.	PropertyID		Char	*		Merkmalskürzel
3.	PropValueID		Char	*		Merkmalswertkürzel
4.	MatLayer		Char	*	X	Name des Material-Layers
5.	Material		Char	*		Materialbezeichnung

Anmerkungen:

- Feld 2 enthält das (sprachunabhängige) Kürzel des Merkmals, für das eine Layerzuordnung erfolgen soll. Feld 3 enthält das (sprachunabhängige) Kürzel des Merkmalswertes, zu dem eine Layerbelegung erfolgen soll.
- Für ein gegebenes Tripel aus Grundartikelnummer, Merkmalskürzel und Merkmalswertkürzel (Felder 1-3) können in der Tabelle mehrere Einträge enthalten sein, die sich auf jeweils verschiedene Layer (Feld 4) beziehen. Dies wird benötigt, wenn ein kaufmännisches Merkmal mehrere Layer steuert.

Beispiel:

Ein Stuhl besitze (u.a.) die Konstruktionsgruppen Sitz und Lehne. In der Polsterungsart *einfarbig* sollen Sitz und Lehne mit dem gleichen Material belegt werden. In der Polsterungsart *mehrfarbig* sollen den beiden Konstruktionsgruppen unterschiedliche Materialien zugewiesen werden können. In Abhängigkeit vom Wert des kaufmännischen Merkmals "Polsterungsart" werde dazu durch Beziehungswissen in den kaufmännischen Daten einmal (einfarbig) das Merkmal "Farbe", und das andere Mal (mehrfarbig) die Merkmale "Farbe Sitz" und "Farbe Lehne" freigeschaltet. Da Sitz und Lehne prinzipiell mit verschiedenen Materialien belegt werden können, sind für beide Konstruktionsgruppen unterschiedliche Material-Layer notwendig (SITZ\_ und LEHNE). In der Polsterungsart *einfarbig* muß das Merkmal "Farbe" (Kürzel F1) dann beide Layer steuern:

```
ABC123;F1;M600;SITZ_;LM600
ABC123;F1;M600;LEHNE;LM600
ABC123;F1;M601;SITZ_;LM601
ABC123;F1;M601;LEHNE;LM601
ABC123;F1;M602;SITZ_;LM602
ABC123;F1;M602;LEHNE;LM602
```

...

- Feld 3 (Merkmalswert) kann leer sein oder das Zeichen „\*“ enthalten. In diesem Fall wird das Feld 5 nicht ausgewertet, sondern dem Layer der aktuelle Wert des Merkmals zugewiesen.

Würden in dem obigen Beispiel die Materialien unter demselben Namen angelegt wie die Merkmalswertkürzel, könnte die Tabelle wie folgt stark vereinfacht werden:

```
ABC123;F1;*;SITZ_;
ABC123;F1;*;LEHNE;
```

...

- Existiert für ein Material-Layer eines Grundartikels nur eine feste Materialbelegung (unabhängig von irgendwelchen kaufmännischen Merkmalen), so sind die Felder 2 und 3 leer zu lassen. Pro Grundartikel und Layer darf es nur einen solchen Eintrag in der Tabelle geben.
- Existieren Grundartikel-übergreifende Layerbelegungen, so können sie in einem gemeinsamen Tabelleneintrag angegeben werden, der im ersten Feld das Zeichen „\*“ enthält. Existieren für einen Grundartikel jedoch explizite Einträge, so haben die dort spezifizierten Layerbelegungen Vorrang.
- Materialnamen (Feld 5) können optional voll qualifiziert angegeben werden. Sind sie nicht oder nur teilweise qualifiziert, wird dem Materialnamen durch die OFML-Laufzeitumgebung implizit der Name des Paketes vorangestellt, in dem sich die ODB-Daten befinden.