

Spezifikation

OAS
OFML Article Selection*
(OFML Part V)

Version 2.0

Status: Release

Thomas Gerth, EasternGraphics GmbH (Editor)

2014-11-28

Inhaltsverzeichnis

1	Einleitung	2
2	Allgemeine Festlegungen	3
2.1	Bestimmungen zum Format	3
2.2	Tabellenbeschreibung und Datentypen	3
2.3	Sprachspezifische Datenelemente	4
2.4	Externe Ressource-Files	5
3	Die Tabellen	6
3.1	Übersicht	6
3.2	Die Ansicht-Tabelle	7
3.3	Das Ansicht-Struktur-Mapping	7
3.4	Die Tabelle der Struktur-Elemente	8
3.5	Die Tabelle der Katalog-Elemente	9
3.6	Die Text-Tabelle	11
3.7	Die Bild-Tabelle	12
3.8	Die Ressource-Tabelle	14
3.9	Die Artikel-Tabelle	15
3.10	Die Tabelle der Methodenaufrufe	16
3.11	Die MetaPlanning-Tabelle	17
3.12	Die Tag-Tabelle	17
3.13	Die Schlagwort-Tabelle	17
3.14	Die Synonym-Tabelle	18
3.15	Die Kategorie-Tabelle	18
3.16	Die Tabelle der Schlagwort-Beziehungen	19
4	Verteilung von OAS-Daten	20
A	Konzeptionelles Datenmodell	21

1 Einleitung

Diese Spezifikation beschreibt ein Format zur strukturierten Darstellung und Auswahl von Artikeln in digitalen Katalogen.

Die Spezifikation ist Teil des OFML-Standards. Das hier beschriebene Format dient lediglich zur Aufbereitung von Katalogdaten. Der Aufbau sonstiger Daten, z.B. der kaufmännischen und graphischen Daten, ist in entsprechenden anderen OFML-Parts beschrieben.

Die Hauptmerkmale von OAS sind:

- hierarchisch organisiertes Auswahlformat
- Unterstützung von multiplen Hierarchien
- Unterstützung von intelligenten Zugriffs- und Filterszenarien
- Unterstützung flexibler Ressourcen zur Repräsentation von Zusatzinformationen

Obwohl sich die Konzepte teilweise ähneln, besteht keine Kontinuität zur Vorgängerversion 1.0 der Spezifikation.

Dem Tabellenentwurf dieser Spezifikation liegen folgende **Begriffe und Konzepte** zugrunde¹:

- Zentraler Begriff ist der des *Katalog-Elements*.
Es gibt verschiedene *Arten* von Katalog-Elementen (*Article, Information, ...*). Je nach Art haben Katalog-Elemente entweder rein informativen Charakter für den Anwender oder führen bei Auswahl durch den Anwender zu bestimmten Aktionen seitens der OFML-Applikation, z.B. zur Erzeugung und Einfügung des durch das Katalog-Element bestimmten Artikels.
- Eine *Katalog-Ansicht* definiert eine aus Ordnern (*Foldern*) und einfachen Elementen bestehende *Struktur* aus 1 oder mehr Bäumen zur visuellen Repräsentation eines Teils der Katalog-Elemente. Es wird also zwischen *Struktur-Elementen* einer Katalog-Ansicht und den eigentlichen Katalog-Elementen unterschieden. Ein einfaches Struktur-Element referenziert genau ein Katalog-Element. Ein und dasselbe Katalog-Element kann dabei von verschiedenen Struktur-Elementen referenziert werden.
- Katalog-Elemente müssen nicht zwingend in einer Katalog-Ansicht vertreten sein. Sie können auch durch geeignete Suchfunktionen der Applikation durch den Anwender gefunden werden.
- Ein Hersteller/Datenlieferant kann mehrere, nach verschiedenen Gesichtspunkten strukturierte Katalog-Ansichten definieren.
- Sowohl für Katalog-Elemente als auch für Struktur-Elemente können *Ressourcen* für die visuelle Repräsentation (Texte, Bilder) hinterlegt werden. Dabei haben Ressourcen für ein Struktur-Element Vorrang vor Ressourcen (gleichen Typs), die eventuell für das referenzierte Katalog-Element hinterlegt sind.
- Katalog-Elemente können mit *Schlagwörtern* (Deskriptoren) verknüpft (getaggt) werden, wobei ein Schlagwort immer als Gruppe von *Synonymen* angelegt wird.
- Um spezifische Suchanfragen zu ermöglichen, können Schlagwörter mit einem Typ versehen werden, der die Art des Schlagworts beschreibt. Von besonderer Bedeutung sind dabei Schlagwörter des Typs *Kategorie*, die Begriffe bzw. Konzepte aus einem bestimmten Wissensgebiet bezeichnen. Zu einer Kategorie kann eine Ober-Kategorie angegeben werden, wodurch Taxonomien, z.B. eine Produktklassifikation, abgebildet werden können.
- Schlagwörter können untereinander in Beziehung gesetzt werden. Damit lassen sich z.B. Beziehungen zwischen Begriffen und Konzepten abbilden (*Ontologien*).

¹ In Anhang A ist das zugehörige konzeptionelle Datenmodell grafisch veranschaulicht.

2 Allgemeine Festlegungen

2.1 Bestimmungen zum Format

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

1. Jede der unten beschriebenen Tabellen ist in genau einer Datei enthalten. Der Dateiname wird durch den Präfix „oas_“, den spezifizierten Tabellennamen und den Suffix „.csv“ gebildet, wobei der Tabellename komplett klein geschrieben wird.
2. Als Zeichensatz wird UTF-8 verwendet². Am Anfang der Datei kann optional das Byte Order Mark angegeben werden.
3. Jede Zeile der Datei repräsentiert einen Datensatz³. Leerzeilen, d.h., solche aus null oder mehr Leerzeichen oder Tabulator (U+0009), werden ignoriert. Zeilen, die mit einem Doppelkreuz ('#' = U+0023) beginnen, werden als Kommentar interpretiert und ebenfalls ignoriert.
4. Die Repräsentationen der einzelnen Felder eines Datensatzes werden durch Semikolon (U+003B) voneinander getrennt.
5. Der Wert eines Feldes besteht aus null oder mehr Unicode-Zeichen mit gültiger UTF-8-Codierung, ausgenommen die Steuer-Zeichen U+0000..U+001F sowie U+007F..U+009F.
6. Die Repräsentation eines Feldes wird aus dem Wert des Feldes gebildet, indem jedes Anführungszeichen ('" = U+0022) durch zwei Anführungszeichen ersetzt und die resultierende Zeichenkette in Anführungszeichen eingeschlossen wird. Wenn der Wert nicht mit einem Anführungszeichen beginnt und kein Semikolon (U+003B) enthält, kann alternativ der Wert des Feldes unverändert als Repräsentation übernommen werden.

2.2 Tabellenbeschreibung und Datentypen

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

- Nummer
- Bezeichner
- Kennzeichen, ob das Feld zum Primärschlüssel der Tabelle gehört⁴
- Datentyp (s.u.)
- maximale Länge des Feldes (Anzahl der Zeichen)⁵
- Kennzeichen, ob das Feld unbedingt gefüllt sein muss (Pflichtfeld)

² Als Normalform sollte NFC (Normalization Form Canonical Composition) verwendet werden.

³ Eine Zeile wird entweder durch ein LF-Zeichen (U+000A) oder durch eine Folge aus CR (U+000D) und LF abgeschlossen.

⁴ Zu einem Primärschlüssel darf es nur einen Datensatz in der Tabelle geben.

⁵ Theoretisch bestehen bei CSV-Datensätzen zwar keine Beschränkungen der einzelnen Feldlängen, aus praktischer Sicht werden hier bei Feldern des Datentyps `char` jedoch bestimmte maximal sinnvolle Längen angegeben.

Folgende **Datentypen** sind definiert:

Char Zeichenkette

Es sind alle Zeichen gemäß Bestimmung 5 (s.o.) erlaubt.

Symbol Symbol

Erlaubt sind alle alphanumerischen Zeichen aus dem ASCII-Zeichensatz (U+0030..U+0039,U+0041..U+005A,U+0061..U+007A) sowie der Unterstrich ('_'=U+005F), wobei das erste Zeichen keine Ziffer sein darf.

ID Identifikator

Erlaubt sind alle lateinischen Großbuchstaben und Ziffern aus dem ASCII-Zeichensatz (U+0030..U+0039,U+0041..U+005A) sowie der Bindestrich ('-'=U+002D) und der Unterstrich ('_'=U+005F).

Int Integer / Ganzzahl

Erlaubt sind alle Ziffern aus dem ASCII-Zeichensatz (U+0030..U+0039), optional das Minuszeichen (U+002D) an erster Stelle.

Bool boolescher Wert

'1' – ja, '0' – nein

OFML Bezeichner entsprechend OFML-Standard (Part III)

Mögliche Angaben sind:

- OFML-Package
- OFML-Schnittstelle
- vollqualifizierter OFML-Typ (Klasse)

Lang Sprach-Code

Die Angabe setzt sich zusammen aus dem zweistelligen Sprachkürzel nach ISO 639-1 und dem zweistelligen Länder- bzw. Regionenkürzel nach ISO 3166-1 (ALPHA-2), getrennt durch einen Bindestrich (U+002D)⁶. Die Angabe des Regionenkürzels ist optional (s. Abschn. 2.3).

Bsp.:

`en_US` für amerikanisches English

`en_GB` für britisches English

Wenn ein Datenelement für beliebige Sprachen verwendet werden kann, so ist das entsprechende Feld leer zu lassen.

2.3 Sprachspezifische Datenelemente

Für die Verwendung von Sprach- und Regionenkürzel im Sprach-Code im Feld **Language** der entsprechenden Tabellen gelten folgende Bestimmungen:

- Für jeden Tabelleneintrag mit einem Sprach-Code, der Sprach- **und** Regionenkürzel enthält, sollte auch ein Tabelleneintrag mit einem Sprach-Code vorhanden sein, der nur das betreffende Sprachkürzel enthält. Dieser Eintrag dient als Fallback für Anwendungen, die den regionsspezifischen Sprach-Code nicht unterstützen.

⁶ Diese Festlegung orientiert sich an der Spezifikation der IETF für Language Tags. Kleinschreibung des Sprachkürzel und Großschreibung des Länderkürzels ist zu beachten!

- Welche Sprache (Region) für den Fallback-Eintrag verwendet wird, liegt im Ermessensspielraum der Hersteller/Datenanleger.
- Sind beide Tabelleinträge inhaltlich identisch, kann (und sollte) der Eintrag mit dem regionspezifischen Sprach-Code entfallen.

Bsp.:

Ist eine Text-Ressource sowohl für amerikanisches als auch für britisches English angelegt, und wird als Fallback für **en** der Text im britischen English verwendet⁷, wird im Feld **Language** des Tabelleintrags mit dem Text im amerikanischen English der Sprach-Code **en_US** angegeben, während für den Tabelleintrag mit dem Text im britischen English als Sprach-Code nur **en** angegeben wird.

Bei der Auswahl eines Tabelleneintrags aus den Einträgen, die zu einem gegebenen Suchschlüssel⁸ passen, geht eine OFML-Anwendung dann wie folgt vor:

1. Eine Anwendung, in der Sprache und Region eingestellt ist, verwendet den Eintrag mit dem passenden Sprach-Code, der Sprach- **und** Regionenkürzel enthält.
2. Ist kein passender Eintrag mit dem regionspezifischen Sprach-Code vorhanden, oder ist in der Anwendung nur die Sprache, aber nicht die Region eingestellt, wird der Eintrag mit dem passenden Sprach-Code verwendet, der nur das betreffende Sprachkürzel enthält.
3. Ist auch kein passender Eintrag mit dem einfachen (nur aus dem Sprachkürzel bestehenden) Sprach-Code vorhanden, wird der Eintrag mit leerem Feld **Language** verwendet (insofern vorhanden).

2.4 Externe Ressource-Files

Der Ablageort von Ressource-Files wird durch die jeweilige OFML-Applikation spezifiziert.

Dateinamen im Feld **File** in der Tabelle **Image** bzw. im Feld **Value** in der Tabelle **Resource** kann ein Verzeichnispfad vorangestellt werden, wobei der Schrägstrich (‘/’=U+002F) als Trennzeichen der einzelnen Pfadkomponenten zu verwenden ist.

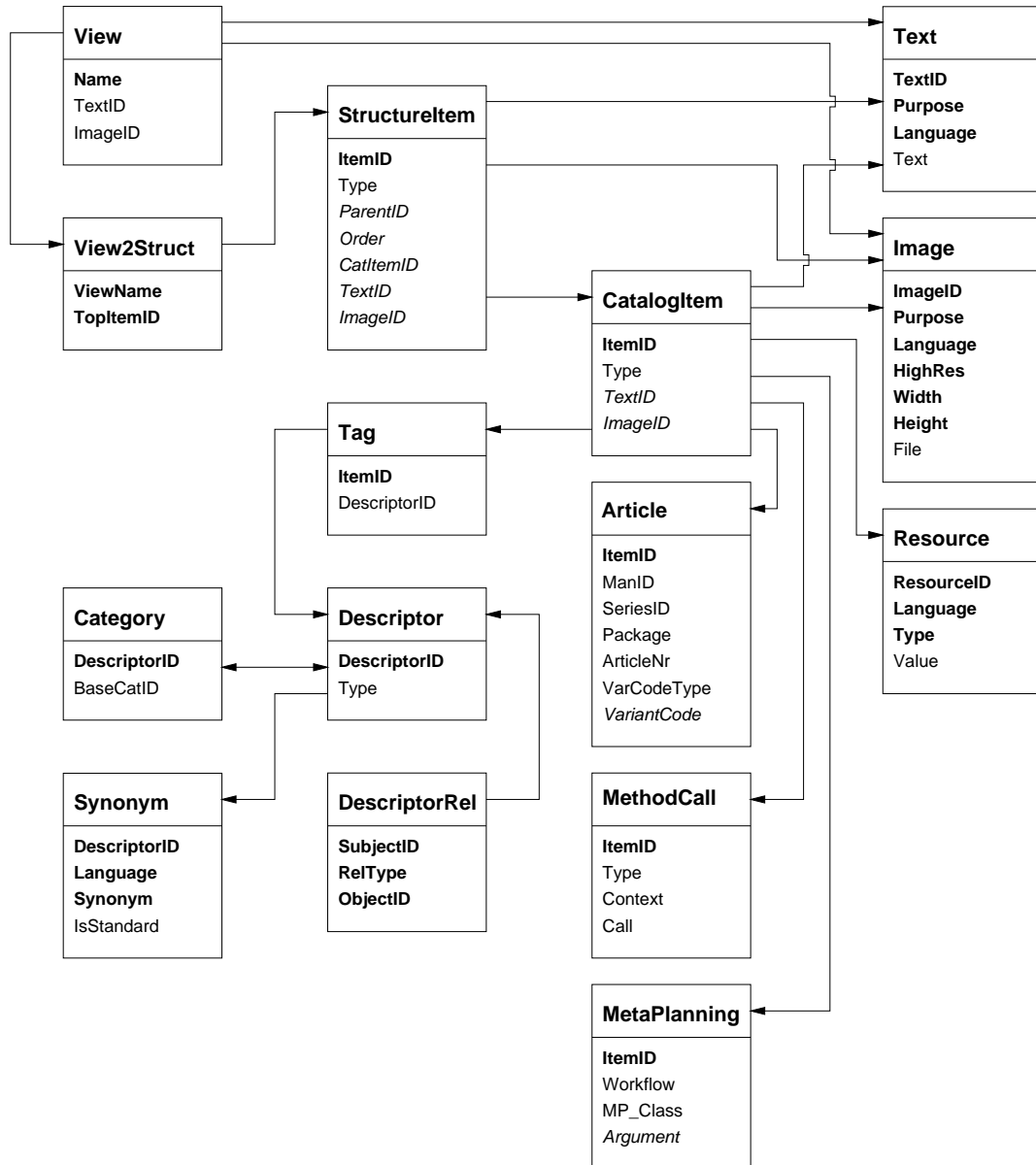
Die Pfadangabe ist ungültig, wenn sie leer ist oder mit einem Schrägstrich beginnt oder die erste Komponente des Pfads aus zwei Punkten (‘.’=U+002E) besteht. Andernfalls wird der Pfad relativ zu dem durch die OFML-Applikation vorgegebenen Ressourcen-Verzeichnis interpretiert.

⁷ z.B. in der Annahme, daß die OFML-Daten vor allem in Europa verwendet werden

⁸ der nicht den Sprach-Code umfasst

3 Die Tabellen

3.1 Übersicht



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

3.2 Die Ansicht–Tabelle

Diese Tabelle legt die verfügbaren Katalog-Ansichten fest.

Ist keine Katalog-Ansicht vorhanden, obliegt die geeignete visuelle Darstellung der Katalog-Elemente (s. Abschn. 3.5) der jeweiligen Applikation.

Tabellenname: **View**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	Name	X	ID		X	Bezeichner der Katalog-Ansicht
2.	TextID		ID			Text-ID
3.	ImageID		ID			Image-ID

Anmerkungen:

- Der Bezeichner (Feld 1) kann bis auf die vordefinierten Bezeichner (s.u.) vom Hersteller frei vergeben werden. Er dient als Fremdschlüssel für die Zuordnung der Baum-Strukturen in der Tabelle **View2Struct** (s. nächster Abschn.).

Folgende Bezeichner sind **vordefiniert**:

DEFAULT Dies ist die Standard-Ansicht. Sie wird verwendet, wenn die Anwendung keine Auswahl einer Ansicht ermöglicht⁹.

- Die Identifikatoren in den Feldern 2 und 3 verweisen auf entsprechende Ressourcen zur Repräsentation der Ansicht in der Anwendung.

Fehlen die Angaben, wird die Ansicht durch die Anwendung auf eine irgendwie geeignete Weise repräsentiert.

3.3 Das Ansicht–Struktur–Mapping

In dieser Tabelle werden einer Ansicht 1 bis N Strukturen (Hierarchie-Bäume) zugeordnet.

Tabellenname: **View2Struct**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ViewName	X	ID		X	Bezeichner der Katalog-Ansicht
2.	TopItemID	X	ID		X	ID des Wurzel-Elements der Baumstruktur

Anmerkungen:

- Die ID im Feld **TopItemID** verweist auf das Struktur-Element (s. Abschn. 3.4), welches das Wurzel-Element der Struktur-Hierarchie darstellt. Typischerweise ist dies ein Element vom Typ **Folder**. Dies ist aber nicht zwingend. Im Extremfall kann die Struktur auch nur aus einem einfachen Struktur-Element bestehen.
- Mehrere Ansichten können sich einen Strukturbaum teilen, d.h., auf ein und dasselbe Wurzel-Element verweisen.

⁹ Ist keine solche Ansicht definiert, ist das Verhalten dieser Anwendungen undefiniert

3.4 Die Tabelle der Struktur-Elemente

In dieser Tabelle werden die Elemente der verfügbaren Katalog-Ansichten und deren hierarchische Beziehungen definiert.

Tabellenname: `StructureItem`

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	<code>StructItemID</code>	X	ID		X	ID des Struktur-Elements
2.	<code>Type</code>		Symbol	12	X	Typ des Struktur-Elements
3.	<code>ParentID</code>		ID			ID des übergeordneten Folders
4.	<code>Order</code>		Int			Position im übergeordneten Folder
5.	<code>CatItemID</code>		ID			ID des referenzierten Katalog-Elements
6.	<code>TextID</code>		ID			Text-ID
7.	<code>ImageID</code>		ID			Image-ID

Anmerkungen:

- Mögliche Typen von Struktur-Elementen (Feld 2) sind:
 - Folder** Dies ist ein Verzeichnis. Es kann andere Elemente als Unter-Elemente enthalten.
 - CatItemProxy** Das Element verweist auf ein Katalog-Element und repräsentiert dieses in der Katalog-Ansicht.
- Für alle Struktur-Elemente außer dem Wurzel-Element einer Katalog-Ansicht muss im Feld `ParentID` die ID des übergeordneten Struktur-Elements angegeben werden. Dieses muss vom Typ `Folder` sein.
- Der optionale Wert im Feld `Order` (nicht-negative Ganzzahl) gibt die Position des Elements innerhalb des übergeordneten Verzeichnisses an. Ist keine Position angegeben oder gibt es mehrere Elemente mit derselben Positionsangabe, ist die Anordnung der betreffenden Elemente undefiniert.
- Ist das Struktur-Element vom Typ `CatItemProxy`, muss im Feld `CatItemID` die ID des Katalog-Elements angegeben werden, welches durch das Struktur-Element repräsentiert wird.
- Mittels der Text-ID im Feld `TextID` werden Einträge in der Tabelle `Text` (s. Abschn. 3.6) referenziert. Diese, sowie die Einträge mit der ggf. vorhandenen Text-ID des referenzierten Katalog-Elements, können dann zur visuellen Darstellung des Elements in der Katalog-Ansicht verwendet werden.
 Beim Zugriff auf die Textinformationen eines Struktur-Elements werden alle Einträge aus der Text-Tabelle ermittelt, deren Text-ID gleich der Text-ID des Struktur-Elements oder gleich der Text-ID des referenzierten Katalog-Elements ist. Befinden sich unter ihnen Einträge, die in den Feldern `Purpose` und `Language` übereinstimmen, so werden von den übereinstimmenden Einträgen die verworfen, deren Text-ID gleich der Text-ID des referenzierten Katalog-Elements ist¹⁰. Aus den so ermittelten Einträgen aus der Text-Tabelle wird entsprechend des in Abschn. 2.3 beschriebenen Verfahrens der Eintrag ausgewählt, der am besten zu der

¹⁰ d.h., die Einträge für das Struktur-Element haben Vorrang vor den ggf. auch vorhandenen (und passenden) Einträgen für das referenzierte Katalog-Element

aktuell in der Anwendung eingestellten Sprache passt.

Kann nach diesem Verfahren kein Eintrag aus der Text-Tabelle ermittelt werden, wird das Struktur-Element in der Katalog-Ansicht ohne Text dargestellt.

- Mittels der Image-ID im Feld **ImageID** werden Einträge in der Tabelle **Image** (s. Abschn. 3.7) referenziert. Diese, sowie die Einträge mit der ggf. vorhandenen Image-ID des referenzierten Katalog-Elements, können dann zur visuellen Darstellung des Elements in der Katalog-Ansicht verwendet werden.

Beim Zugriff auf die Bildinformationen eines Struktur-Elements werden alle Einträge aus der Image-Tabelle ermittelt, deren Image-ID gleich der Image-ID des Struktur-Elements oder gleich der Image-ID des referenzierten Katalog-Elements ist. Befinden sich unter ihnen Einträge, die in den Feldern **Purpose**, **Language**, **HighRes**, **Width** und **Height** übereinstimmen, so werden von den übereinstimmenden Einträgen die verworfen, deren Image-ID gleich der Image-ID des referenzierten Katalog-Elements ist¹¹. Aus den so ermittelten Einträgen aus der Image-Tabelle wird entsprechend des in Abschn. 2.3 beschriebenen Verfahrens der Eintrag ausgewählt, der am besten zu der aktuell in der Anwendung eingestellten Sprache passt. Kann nach diesem Verfahren kein Eintrag aus der Image-Tabelle ermittelt werden, ist das Verhalten von der Anwendung abhängig¹².

3.5 Die Tabelle der Katalog-Elemente

Tabellenname: **CatalogItem**

Pflichttabelle: ja

In dieser Tabelle werden alle Katalog-Elemente aufgelistet, unabhängig von evtl. verfügbaren Katalog-Ansichten.

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	CatItemID	X	ID		X	ID des Katalog-Elements
2.	Type		Symbol	12	X	Typ des Katalog-Elements
3.	TextID		ID			Text-ID
4.	ImageID		ID			Image-ID
5.	ResourceID		ID			Resource-ID

Anmerkungen:

- Mögliche Typen von Katalog-Elementen (Feld 2) sind:

Article Das Katalog-Element repräsentiert einen Artikel, entweder in seiner initialen oder in einer davon abweichenden Konfiguration. Bei Auswahl des Elements durch den Anwender wird der Artikel erzeugt und in das aktuelle Projekt eingefügt. Die für die Erzeugung des Artikels erforderlichen Parameter werden in der Tabelle **Article** hinterlegt (s. Abschn. 3.9).

Info Das Katalog-Element repräsentiert eine Information, die bei Auswahl des Elements durch den Anwender angezeigt wird. Die anzuzeigende (externe) Information wird in der Tabelle **Resource** (s. Abschn. 3.8) spezifiziert und über die ID im Feld **ResourceID** referenziert.

¹¹ s. Fußnote oben zur Text-ID

¹² Die Anwendung könnte z.B. ein Standardbild verwenden oder versuchen, ein evtl. in einer anderen Auflösung vorhandenes Bild zu skalieren.

- Graphics** Das Katalog-Element repräsentiert eine Grafik (3D-Geometrie), die bei Auswahl des Elements durch den Anwender in das Projekt eingefügt wird. Die Datei, welche die Grafik enthält, wird in der Tabelle **Resource** (s. Abschn. 3.8) spezifiziert und über die ID im Feld **ResourceID** referenziert.
- Der Import von Geometrie-Dateien wird nicht von allen Applikationen bzw. Applikationsmodi unterstützt. Der Applikation ist es in diesem Fall freigestellt, ob sie Struktur-Elemente in Katalog-Ansichten anzeigt, die auf Katalog-Elemente dieses Typs verweisen.
- MethodCall** Bei Auswahl von Katalog-Elementen dieses Typs durch den Anwender wird ein OFML-Methode ausgeführt. Die Signatur und die Parameter des Aufrufs müssen dazu in der Tabelle **MethodCall** hinterlegt werden (s. Abschn. 3.10).
- Methodenaufrufe durch den Anwender werden nicht von allen Applikationen bzw. Applikationsmodi unterstützt. Der Applikation ist es in diesem Fall freigestellt, ob sie Struktur-Elemente in Katalog-Ansichten anzeigt, die auf Katalog-Elemente dieses Typs verweisen.
- MetaPlanning** Bei Auswahl von Katalog-Elementen dieses Typs durch den Anwender wird ein *MetaPlanning*-Workflow gestartet. Die dafür benötigten Parameter müssen in der Tabelle **MetaPlanning** hinterlegt werden (s. Abschn. 3.11).
- MetaPlanning-Workflows werden nicht von allen Applikationen bzw. Applikationsmodi unterstützt. Der Applikation ist es in diesem Fall freigestellt, ob sie Struktur-Elemente in Katalog-Ansichten anzeigt, die auf Katalog-Elemente dieses Typs verweisen.
- Mittels der Text-ID im Feld **TextID** und der Image-ID im Feld **ImageID** werden Einträge in der Tabelle **Text** (s. Abschn. 3.6) bzw. in der Tabelle **Image** (s. Abschn. 3.7) referenziert, welche von der Applikation zur visuellen Darstellung des Elements in Katalog-Ansichten bzw. in Ergebnis-Ansichten von Suchanfragen verwendet werden können. In Katalog-Ansichten haben jedoch (insofern vorhanden) Tabelleneinträge für das referenzierende Struktur-Element Vorrang (s.a. Beschreibung zu den gleichnamigen Feldern in der Tabelle **StructureItem**, Abschn. 3.4).
 - Im Feld **ResourceID** wird auf die typ-spezifisch benötigte Ressource in der Tabelle **Resource** verwiesen (s. Abschn. 3.8). Aktuell ist das Feld nur für die Katalog-Element-Typen **Info** und **Graphics** relevant.

3.6 Die Text-Tabelle

In dieser Tabelle werden Texte abgelegt, die zur visuellen Darstellung von Struktur- bzw. Katalog-Elementen verwendet werden. Die Zuordnung erfolgt mittels der Text-ID, siehe Feld `TextID` in den Tabellen `StructureItem` und `CatalogItem` (Abschn. 3.4 und 3.5)¹³.

Tabellenname: `Text`

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	<code>TextID</code>	X	ID		X	ID des Textes
2.	<code>Purpose</code>	X	Symbol	12	X	Verwendungszweck
3.	<code>Language</code>	X	Lang	5	X	Sprach-Code
4.	<code>Text</code>		Char		X	Text-Inhalt

Anmerkungen:

- Neben den unten genannten vordefinierten Verwendungszwecken kann im Feld `Purpose` ein beliebiger Bezeichner angegeben werden. Letzteres ist sinnvoll, wenn die Daten auch in speziellen Anwendungen (z.B. Web-Clients) mit spezifischen Anforderungen verarbeitet werden sollen.

Folgende Verwendungszwecke sind **vordefiniert**:

CatTextShort Der Kurztext ist hauptsächlich zur Darstellung eines Elements in der Katalog-Ansicht in Verbindung mit einem Bild (Icon) vorgesehen.

CatTextLong Der Langtext kann in Hinweis-Dialogen (Tool-Tips) oder sonstigen Kontexten mit größerem verfügbarem Platz verwendet werden.

- Texte können sprachspezifisch angelegt werden (Feld `Language`). Details zur Verwendung des Sprach-Codes siehe Abschn. 2.3.
- Im Feld `Text` sind bei Langtexten (Verwendungszweck `CatTextLong`) folgende, aus HTML bekannte Formatierungstags und Zeichenbeschreibungen erlaubt bzw. gefordert:

`
` Erzwingt einen Zeilenumbruch¹⁴.

`<` Bewirkt die Darstellung eines Kleiner-als-Zeichens ('<').

`>` Bewirkt die Darstellung eines Größer-als-Zeichens ('>').

`&` Bewirkt die Darstellung eines Kaufmanns-Und-Zeichens ('&').

¹³ Mehrere Struktur- bzw. Katalog-Elemente können sich somit einen Text-Eintrag teilen.

¹⁴ Dabei ist zu beachten, daß von der Anwendung je nach Layout ggf. noch zusätzlich Zeilenumbrüche eingefügt werden können.

3.7 Die Bild-Tabelle

In dieser Tabelle werden Bild-Dateien referenziert, die zur visuellen Darstellung von Struktur- und Katalog-Elementen bzw. als Ressource für Katalog-Elemente des Typs `Info` verwendet werden. Die Zuordnung erfolgt mittels der Image-ID, siehe Feld `ImageID` in den Tabellen `StructureItem` und `CatalogItem` (Abschn. 3.4 und 3.5)¹⁵ bzw. Feld `Value` in der Tabelle `Resource` (Abschn. 3.8).

Tabellenname: `Image`

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	<code>ImageID</code>	X	ID		X	ID des Bildes
2.	<code>Purpose</code>	X	Symbol	12	X	Verwendungszweck
3.	<code>Language</code>	X	Lang	5	X	Sprach-Code
4.	<code>HighRes</code>	X	Bool	1	X	für hochauflösendes Display?
5.	<code>Width</code>	X	Int		X	Breite in Pixel
6.	<code>Height</code>	X	Int		X	Höhe in Pixel
7.	<code>File</code>		Char		X	relativer Pfad zur Bild-Datei

Anmerkungen:

- Neben den unten genannten vordefinierten Verwendungszwecken kann im Feld `Purpose` ein beliebiger Bezeichner angegeben werden. Letzteres ist sinnvoll, wenn die Daten auch in speziellen Anwendungen (z.B. Web-Clients) mit spezifischen Anforderungen verarbeitet werden sollen.

Folgende Verwendungszwecke sind **vordefiniert**:

SmallIcon Dieser Verwendungszweck ist für wirklich kleine Icons gedacht, wie sie in einer aus einzeiligen Texten bestehenden Liste von Katalogeinträgen unmittelbar vor dem Text verwendet werden würden.

Da die Skalierung von sehr kleinen Bildern oft unbefriedigende Ergebnisse liefert, gelten folgende Größenempfehlungen:

Höhe: 20 (`HighRes=0`) bzw. 40 (`HighRes=1`) Pixel

Breite: maximal das fünffache der Höhe

Es wird erwartet, daß Anwendungen diese Bilder, soweit in beiden Auflösungen vorhanden, unskaliert darstellen.

Icon Dieser Verwendungszweck ist für Bilder gedacht, die in einer normalen gekachelten Katalogansicht verwendet werden.

Richtwert für die Höhe dieser Bilder sind 40 (`HighRes=0`) und 80 (`HighRes=1`) Pixel.

Es wird erwartet, daß Anwendungen diese Bilder in der Standardeinstellung (100% Katalogbildgröße) unskaliert darstellen.

Image Dieser Verwendungszweck ist für größere, auf dem Bildschirm darzustellende Bilder gedacht.

Richtwert für die Größe sind etwa 1000x1000 Pixel.

Es wird erwartet, dass diese Bilder von Anwendungen auf die gewünschte Größe skaliert werden.

¹⁵ Mehrere Struktur- bzw. Katalog-Elemente können sich somit einen Eintrag in dieser Tabelle teilen.

Info Dieser Verwendungszweck ist für Bilder gedacht, die bei Auswahl von Katalog-Elementen des Typs **Info** durch den Anwender als Information angezeigt werden sollen.

Es wird erwartet, dass diese Bilder von Anwendungen unskaliert dargestellt werden, zumindest nicht vergrößert werden..

- Für die Verwendungszwecke **SmallIcon** und **Icon** gibt das Feld **HighRes** an, ob das Bild für die Darstellung auf einem herkömmlichen Display¹⁶ (Wert 0) oder auf einem hochauflösenden Display¹⁷ (Wert 1) gedacht ist.

Wenn auf Grund des vorhandenen Displays das Bild mit **HighRes=1** gefordert aber nicht vorhanden ist, so wird statt dessen das Bild mit **HighRes=0** (soweit vorhanden) mit dem Faktor 2 skaliert und verwendet. (Umgekehrt wird entsprechend der Faktor 0.5 angewendet.)

Für alle anderen Verwendungszwecke ist die Bedeutung dieses Feldes undefiniert. Im Zweifelsfall sollte es auf 0 gesetzt werden.

- In den Feldern **Width** und **Height** wird – optional – die exakte Breite bzw. Höhe des Bildes in Pixeln angegeben¹⁸. Der Wert 0 bedeutet dabei keine Angabe. Ist eines der beiden Felder ungleich 0, muss auch das andere ungleich 0 sein.

Existieren mehrere Einträge, bei denen die Felder 1-4 übereinstimmen, dann muß für alle dieser Einträge die Breite und die Höhe des Bildes angegeben werden. Dies kann in Web-Anwendungen zur Realisierung eines responsiven Webdesigns verwendet werden.

- Das Bildformat der in Feld **File** angegebenen Datei muss **JPEG** oder **PNG** sein.

Bilder im **JPEG**-Format müssen gemäß der Spezifikation für das „JPEG File Interchange Format (JFIF)“¹⁹ hinterlegt werden:

- müssen sequenziell (nicht interlaced/progressive) strukturiert sein
- müssen Huffman-Kodierung (nicht arithmetische Kodierung) nutzen
- müssen das YCbCr-Farbmodell nutzen (kein Schwarz/Weiß)
- müssen 8 Bit-Farbkanal nutzen (nicht mehr)

Bilder im **PNG**-Format müssen gemäß der „PNG (Portable Network Graphics) Specification, Version 2.2“²⁰ hinterlegt werden:

- müssen sequenziell (nicht interlaced/progressive) strukturiert sein
- müssen das RGB-Farbmodell nutzen
- müssen 8 Bit-Farbkanal nutzen (nicht mehr, kein Schwarz/Weiß)
- optional ist ein 8 Bit Alphakanal für transparente Bilder erlaubt

¹⁶ca. 100 dpi Auflösung

¹⁷üblicherweise, abhängig von dem für das Gerät üblichen Augenabstand, zwischen 200 und 400 dpi

¹⁸Der Wert dieser Felder darf nicht negativ sein.

¹⁹<http://www.jpeg.org/public/jfif.pdf>

²⁰<http://libpng.org/pub/png/spec/1.2/png-1.2-pdg.html>

3.8 Die Ressource-Tabelle

In dieser Tabelle werden spezifische externe Ressourcen definiert.

Tabellenname: **Resource**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	ResourceID	X	ID		X	ID der Ressource
2.	Language	X	Lang	5	X	Sprach-Code
3.	Type	X	Symbol	13	X	Datentyp der Ressource
4.	Value		Char		X	Wert/Inhalt der Ressource

Anmerkungen:

- Ressourcen für Katalog-Elemente des Typs **Info** können sprachspezifisch angelegt werden (Feld **Language**):
Aus den Einträgen mit der Ressource-ID des Katalog-Elements wird entsprechend des in Abschn. 2.3 beschriebenen Verfahrens der Eintrag ausgewählt, der am besten zu der aktuell in der Anwendung eingestellten Sprache passt.
- Folgende Datentypen (Feld **Type**) sind definiert:

Image

Die Ressource verweist auf eine Bilddatei.

Im Feld **Value** wird nicht die Bilddatei direkt angegeben, sondern die Image-ID, die auf die entsprechenden Einträge in der Tabelle **Image** verweist (s. Abschn. 3.7).

Aus diesen Einträgen kommen diejenigen in Frage, bei denen das Feld **Purpose** den Wert **Info** besitzt und der Wert des Feldes **Language** mit dem Wert des gleichnamigen Feldes aus diesem Eintrag übereinstimmt. Existieren mehrere solcher Einträge, kann/muß die Anwendung anhand der Werte in den Feldern **HighRes**, **Width** und **Height** der Bild-Tabelle das passend(st)e Bild auswählen.

HTML

Die Ressource verweist auf eine Datei im HTML-Format.

URL

Die Ressource enthält einen vollqualifizierten Uniform Resource Locator für eine Internet-Ressource entsprechend RFC 1738²¹. Die Anzeige erfolgt entweder im integrierten Internet-Browser der Applikation oder im System-Browser.

MIME

Die Ressource enthält die Deklaration eines Inhalts gemäß der MIME-Spezifikation. Das Format der Ressourcenangabe ist:

`<MIME-Content-Type>;<Content-Filename>`

Ressourcen dieses Typs können zur Anzeige von Dokumenten spezifischer Formate (PDF, Excel, ...) oder auch Videos verwendet werden.

Beispiele:

```
application/pdf;info.pdf
application/ms-excel;info.xls
video/x-msvideo;info.avi
```

Die Anzeige erfolgt entweder im integrierten Internet-Browser der Applikation oder im System-Browser ggf. unter Zuhilfenahme von separaten Viewer-Programmen. Der Umfang der verarbeitbaren Content-Types hängt somit vom System des Anwenders ab.

²¹ <http://tools.ietf.org/html/rfc1738>

Graphics Die Ressource verweist auf eine Geometrie-Datei.

Die durch den Standard unterstützten Formate sind: 3DS, DXF und DWG.

Über diese Standardformate hinaus können bestimmte Anwendungen weitere Formate unterstützen, u.a. proprietäre Erweiterungen von DXF und DWG zum Übertragen von Informationen, die über die reine grafische Repräsentation hinausgehen (z.B. kaufmännische Informationen).

Je nach Typ des Katalog-Elements sind nur bestimmte Datentypen (Feld **Type**) erlaubt/relevant:

Typ des Katalog-Elements	Daten-Typ				
	Image	HTML	URL	MIME	Graphics
Info	X	X	X	X	
Graphics					X

3.9 Die Artikel-Tabelle

In dieser Tabelle werden die Parameter spezifiziert, die für die Erzeugung von Artikeln (Katalog-Elemente vom Typ **Article**) benötigt werden.

Tabellenname: **Article**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	CatItemID	X	ID		X	ID des Katalog-Elements
2.	ManID		Char		X	kaufm. Kürzel des Herstellers
3.	Series		Char		X	kaufm. Kürzel der Serie
4.	Package		OFML		X	OFML-Package, das die Daten des Artikels enthält
5.	ArticleNr		Char		X	Grundartikelnummer
6.	VarCodeType		Symbol	12	X	Schema des Variantencodes
7.	VariantCode		Char			Variantencode

Soll der Artikel in einer von der initialen Konfiguration abweichenden Konfiguration erzeugt werden, muss dazu im Feld **VariantCode** der entsprechende Variantencode angegeben werden.

Das bei der Codierung des Variantencodes verwendete Schema wird im Feld **VarCodeType** spezifiziert. Folgende Werte sind erlaubt:

None Der Artikel soll in seiner initialen Konfiguration erzeugt werden. (Das Feld **VariantCode** wird dann ignoriert.)

Manufacturer Der Variantencode im Feld **VariantCode** ist nach dem hersteller-spezifischen Schema codiert.

OFML Der Variantencode im Feld **VariantCode** ist nach einem standardisierten, hersteller-neutralen Schema codiert. Dieses entspricht dem vordefinierten OCD-Codierungsschema **KeyValueList**, welches wie folgt definiert ist:

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

[<OCD-Merkmalsskategorie>.] <OCD-Merkmal>=<OCD-Merkmalwert>

Die Angabe der Merkmalsklasse kann entfallen, wenn das Merkmal nur in einer Merkmalsklasse des Artikels definiert ist.

Als Trennzeichen zwischen den Merkmalen wird das Semikolon verwendet.

Für aktuell nicht ausgewählte optionale Merkmale wird das interne Wertkürzel „VOID“ verwendet. Nicht belegte Stellen (Leerzeichen) am Ende von Merkmalswerten (in Bezug auf die für das Merkmal definierte maximale Anzahl von Wertstellen) werden weggelassen.

Die gewünschte Konfiguration muss nicht vollständig im Variantencode spezifiziert werden; es ist ausreichend, nur diejenigen Merkmale zu codieren, deren Werte von der initialen Konfiguration abweichen.

3.10 Die Tabelle der Methodenaufrufe

In dieser Tabelle werden die Parameter spezifiziert, die für die Ausführung eines Methodenaufrufs (Katalog-Elemente vom Typ `MethodCall`) benötigt werden.

Tabellenname: `MethodCall`

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	<code>CatItemID</code>	X	ID		X	ID des Katalog-Elements
2.	<code>Type</code>		Symbol	8	X	Art des Aufrufs
3.	<code>Context</code>		OFML		X	Aufruf-Kontext
4.	<code>Call</code>		Char		X	Aufruf

Es werden 2 Arten von Methodenaufrufen unterstützt. Die jeweils vorliegende Art muss im Feld `Type` spezifiziert werden:

Instance Die Methode wird auf einer OFML-Instanz (Objekt) aufgerufen, z.B. auf einer Instanz, die einen Artikel repräsentiert.

Im Feld `Context` ist dazu der vollqualifizierte Bezeichner des OFML-Typs oder der OFML-Schnittstelle anzugeben, für die die Methode definiert ist, und im Feld `Call` die Signatur des Methodenaufrufs (inkl. Argumenten) entsprechend des Syntax von OFML Part III.

Bei Auswahl des Katalog-Elements durch den Anwender wird die Methode auf dem aktuell selektierten Objekt ausgeführt, wenn dessen Klasse von dem im Feld `Context` spezifizierten Typ abstammt bzw. die dort spezifizierte Schnittstelle implementiert.

Diese Form des Methodenaufrufs ist somit nur in Applikationen bzw. Applikationsmodi möglich, die die Erzeugung und Selektion von OFML-Instanzen erlauben.

Class Es wird eine (statische) Klassen-Methode aufgerufen.

Im Feld `Context` ist dazu das OFML-Package anzugeben, welches die Klasse enthält, und im Feld `Call` die Signatur des Methodenaufrufs (inkl. Argumenten) entsprechend des Syntax von OFML Part III. Als Argument kann dabei der Platzhalter `%SELECTION%` verwendet werden. Dieser wird vor dem Aufruf der Methode durch eine Liste der aktuell selektierten OFML-Instanzen ersetzt²².

²² Ist kein Objekt selektiert, wird eine leere Liste übergeben.

3.11 Die MetaPlanning-Tabelle

In dieser Tabelle werden die Parameter spezifiziert, die für die Ausführung eines MetaPlanning-Workflows (Katalog-Elemente vom Typ `MetaPlanning`) benötigt werden.

Dabei handelt es sich um die vollqualifizierte ID des Workflows, die vollqualifizierte Klasse, welche die User Exits des Workflows implementiert, und (optional) die Argumente, welche die `initialize()`-Funktion der UserExit-Klasse zusätzlich zu den Standard-Argumenten der Basis-Klasse `::ofml::go::GoMetaPlanning` benötigt.

Tabellenname: `MetaPlanning`

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	<code>CatItemID</code>	X	ID		X	ID des Katalog-Elements
2.	<code>Workflow</code>		OFML		X	ID des Workflows
3.	<code>MP_Class</code>		OFML		X	UserExit-Klasse
4.	<code>Argument</code>		Char			Argument

3.12 Die Tag-Tabelle

Diese Tabelle dient der Anbindung von Schlagwörtern (Tabelle `Descriptor` in Abschn. 3.13) an Katalog-Elemente.

Tabellenname: `Tag`

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	<code>CatItemID</code>	X	ID		X	ID des Katalog-Elements
2.	<code>DescriptorID</code>		ID		X	ID des Schlagworts

3.13 Die Schlagwort-Tabelle

Diese Tabelle legt die für den Katalog definierten Schlagwörter fest.

Tabellenname: `Descriptor`

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	<code>DescriptorID</code>	X	ID		X	ID des Schlagworts
2.	<code>Type</code>		Symbol		X	Art des Schlagworts

Es werden folgende Arten von Schlagwörtern (Feld `Type`) unterstützt:

Untyped Das Schlagwort hat keine spezifische (standardisierte) Bedeutung.

Category Das Schlagwort benennt eine Kategorie (Begriff aus einem Wissensgebiet).

In der Tabelle `Category` (s. Abschn. 3.15) können Vererbungsbeziehungen zwischen Kategorien (Taxonomien) abgebildet werden.

Designer Das Schlagwort benennt einen Designer (eines Produkts).

Characteristic Das Schlagwort benennt eine Produkteigenschaft.

Die Schlagwort-IDs `CAT_PRODUCT` und `CAT_ENVIRONMENT` sind reserviert (s. Tabelle `Category` in Abschn. 3.15).

3.14 Die Synonym-Tabelle

In dieser Tabelle werden die Synonyme für ein Schlagwort definiert.

Tabellenname: `Synonym`

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	<code>DescriptorID</code>	X	ID		X	ID des Schlagworts
2.	<code>Language</code>	X	Lang	5	X	Sprach-Code
3.	<code>Synonym</code>	X	Char		X	Synonym
4.	<code>IsStandard</code>		Bool		X	Ist Standard-Synonym ?

Anmerkungen:

- Synonyme können sprachspezifisch angelegt werden (Feld `Language`).
- Als Standard gekennzeichnete Synonyme (Feld `IsStandard`) werden von der Applikation bei der Anzeige und Suche bevorzugt. Sind von den für eine geforderte Sprache passenden Synonymen mehrere als Standard gekennzeichnet, ist undefiniert, welches von diesen durch die Applikation als Haupt-Synonym verwendet wird.
- Für die reservierten Schlagwort-IDs `CAT_PRODUCT` und `CAT_ENVIRONMENT` müssen keine Synonyme hinterlegt werden.

3.15 Die Kategorie-Tabelle

In dieser Tabelle werden Vererbungsbeziehungen zwischen Kategorien abgebildet, wobei eine Kategorie genau eine übergeordnete Kategorie besitzen kann.

Tabellenname: `Category`

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	<code>DescriptorID</code>	X	ID		X	ID des Schlagworts (Kategorie)
2.	<code>BaseCatID</code>		ID		X	ID der übergeordneten Kategorie

Anmerkungen:

- In beiden Feldern dürfen nur IDs von Schlagwörtern verwendet werden, für die in der Tabelle `Descriptor` (s. Abschn. 3.13) der Typ `Category` angegeben ist.
- Folgende Top-Kategorien (IDs) sind vordefiniert:

CAT_PRODUCT

Die Basis-Kategorie von Produkt-Klassifikationen.

Konzeptionell sind alle Katalog-Elemente vom Typ **Article** per se dieser Kategorie zugeordnet, ohne dass sie explizit mit der Schlagwort-ID **CAT_PRODUCT** getaggt werden müssen.

CAT_ENVIRONMENT

Die Basis-Kategorie von Klassifikationen für Einrichtungsumgebungen.

3.16 Die Tabelle der Schlagwort-Beziehungen

In dieser Tabelle werden Beziehungen zwischen Schlagwörtern definiert. Konzeptionell werden damit Aussagen aus dem Wissensgebiet in der Form „*Subject Prädikat Object*.“ abgebildet.

Tabellenname: **DescriptorRel**

Pflichttabelle: nein

Nr.	Name	Key	Typ	Länge	Pflicht	Erklärung
1.	SubjectID	X	ID		X	ID des Subjekt-Schlagworts
2.	RelType	X	Symbol		X	Art der Beziehung
3.	ObjectID	X	ID		X	ID des Objekt-Schlagworts

Die Beziehung (Feld **RelType**) ist immer vom Subjekt zum Objekt gerichtet. Handelt es sich in der Realität um eine symmetrische Beziehung, so muss ein zweiter Eintrag in dieser Tabelle angelegt werden, in dem Subjekt und Objekt ihre Rollen tauschen.

4 Verteilung von OAS-Daten

Die OAS-Daten eines Katalogs können auf mehrere OFML-Pakete verteilt sein²³.

Die OAS-Daten der OFML-Pakete, die einen Katalog definieren, werden beim Import durch die Anwendung zu einer gemeinsamen Datenbank zusammengefaßt. Dabei gelten folgende Bestimmungen, die z.T. auch schon bei der Datenanlage zu berücksichtigen sind:

1. Identifikatoren in den Feldern `StructItemID`, `CatItemID`, `TopItemID`, `ResourceID`, `TextID` und `ImageID` in den relevanten Tabellen werden mit der ID des jeweiligen OFML-Pakets verknüpft, um sicher zu einer eindeutigen ID zu gelangen²⁴.
2. Bezeichner von Katalog-Ansichten (Feld `Name` in der Tabelle `View`) müssen über alle OFML-Pakete des Katalogs hinweg eindeutig sein und werden der Anwendung beim Import 1:1 übernommen.
3. Aufgrund der Bestimmung 1 können die Wurzel-Elemente der Katalog-Ansichten (Feld `TopItemID` in der Tabelle `View2Struct`) aus verschiedenen OFML-Paketen problemlos für eine gegebene Ansicht übernommen (zusammengefaßt) werden²⁵.
4. Identifikatoren von Schlagwörtern (Feld `DescriptorID` in der Tabelle `Descriptor`) müssen über alle OFML-Pakete des Katalogs hinweg eindeutig sein und werden der Anwendung beim Import 1:1 übernommen.
5. Für die Tabelle `Descriptor` wird gefordert, dass einer konkreten Descriptor-ID in jedem OFML-Paket des Katalogs der gleiche Typ (Feld `Type`) zugeordnet ist. Andernfalls schlägt die Verarbeitung fehl, d.h. die Tabelleneinträge mit dieser Descriptor-ID werden nicht in die gemeinsame Datenbank übernommen.
6. Beim Zusammenfassen der Tabellen `Synonym`, `Category` und `DescriptorRel` kommt es zu einem Konflikt, wenn derselbe Primärschlüssel auftritt. Um derartiges zu vermeiden, wird empfohlen, die genannten Tabellen sowie die Tabelle `Descriptor`²⁶ komplett in einem OFML-Paket anzulegen und zu verteilen.

²³ Dies kann z.B. bei Serien-zentrierten Ansichten sinnvoll sein. Die OAS-Daten der Serie können dann direkt im Serien-Paket angelegt und verteilt werden.

²⁴ Diese Identifikatoren müssen also über die OFML-Pakete des Katalogs hinweg *nicht* eindeutig sein.

²⁵ d.h., die Hierarchie-Bäume einer Ansicht, so es denn mehrere gibt, können in je einem eigenen OFML-Paket angelegt und verteilt werden

²⁶ d.h., die Taxonomie und Ontologie des Katalogs

A Konzeptionelles Datenmodell

Abb. 1 veranschaulicht das dieser Spezifikation zugrundeliegende konzeptionelle Datenmodell²⁷.

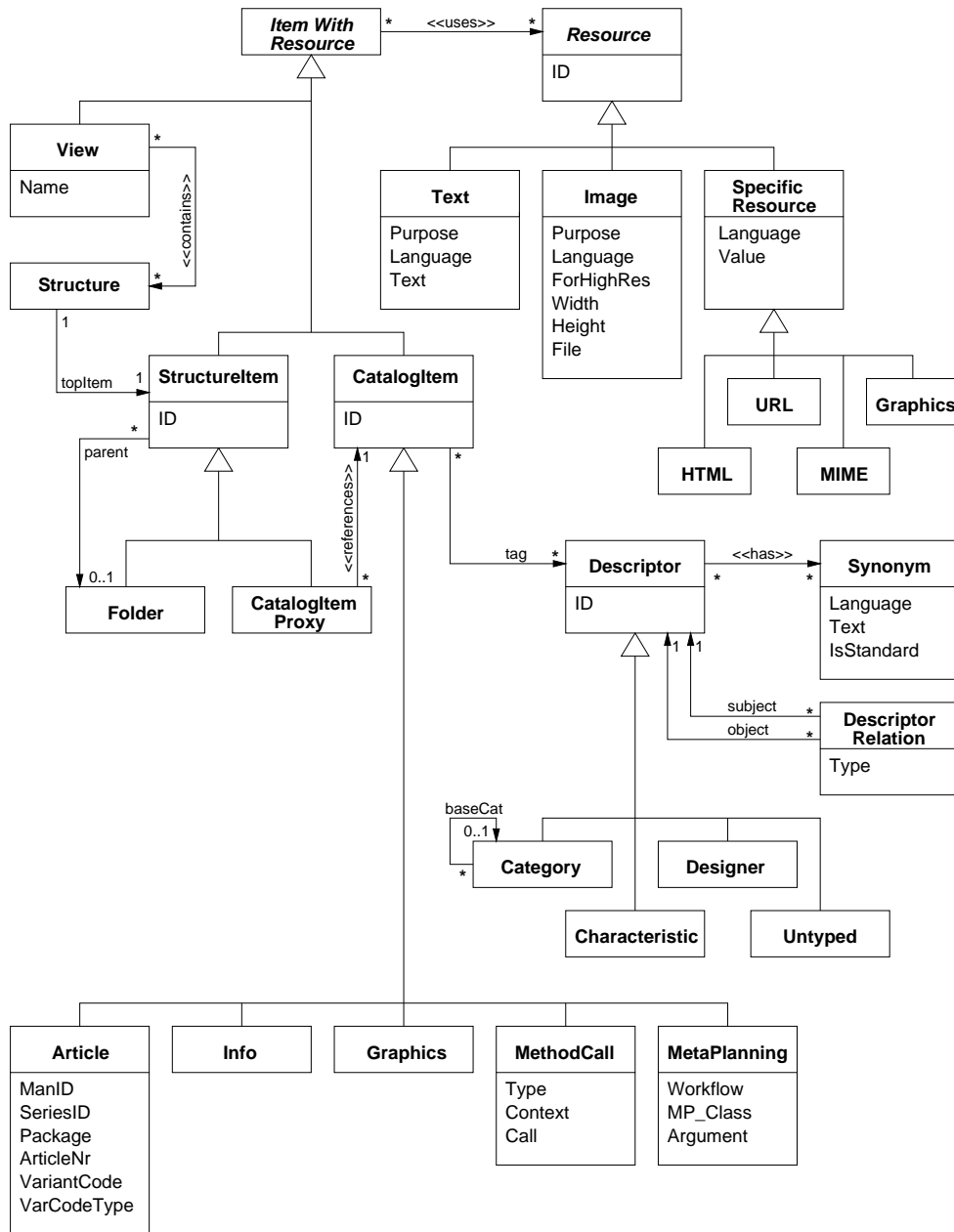


Abb. 1

²⁷ Die in diesem Diagramm verwendete Notation entspricht der *Unified Modeling Language (UML)* Spezifikation (www.omg.org/uml).