



# **GS1 XML**

**Technischer Leitfaden  
zum Release 2.x**





## GS1 Germany

GS1 Germany hilft Unternehmen aller Branchen dabei, moderne Kommunikations- und Prozess-Standards in der Praxis anzuwenden und damit die Effizienz ihrer Geschäftsabläufe zu verbessern. Unter anderem ist das Unternehmen in Deutschland für das weltweit überschneidungsfreie Artikelidentensystem GTIN zuständig - die Grundlage des Barcodes. Darüber hinaus fördert GS1 Germany die Anwendung neuer Technologien zur vollautomatischen Identifikation von Objekten (EPC/RFID) und bietet Lösungen für mehr Kundenorientierung (ECR - Efficient Consumer Response).



Das privatwirtschaftlich organisierte Unternehmen mit Sitz in Köln gehört zum internationalen Netzwerk "Global Standards One" (GS1) und ist die zweitgrößte von mehr als 100 GS1-Länderorganisationen. Paritätische Gesellschafter von GS1 Germany sind der Markenverband und das EHI Retail Institute.

## **Zu dieser Schrift**

Die GS1 XML-Standards bestehen aus einem Set elektronischer Geschäftsnachrichten, mit denen Unternehmen unabhängig von den verwendeten internen Soft- und Hardware-systemen Informationen schnell, präzise und effizient austauschen können. Die GS1 XML-Standards unterstützen – u. a. auf Basis der klassischen Nachrichtenarten Bestellung, Liefermeldung und Rechnung – Prozesse entlang der gesamten Supply Chain.

Zusätzlich zu den klassischen Einsatzgebieten wird GS1 XML mit Priorität für neue Branchen und die Abbildung neuer Geschäftsprozesse verwendet.

So basiert beispielsweise das Global Data Synchronisation Network (GDSN), das den Unternehmen der Konsumgüterwirtschaft einen vereinfachten Umgang mit Artikelstammdaten über weltweit nur einen Zugangspunkt ermöglicht, vollständig auf den GS1 XML-Standards.

Dieser technische Leitfaden dient als Umsetzungshilfe bei der Einführung des GS1 XML-Standards. Organisatorische Fragestellungen zu einzelnen Geschäftsnachrichten sind nicht Bestandteil dieses Dokuments.

Köln, im April 2010

Abschnitt	Seite
<b>1 Anwendungsbereich dieses Dokuments .....</b>	<b>8</b>
<b>2 Allgemeiner Überblick zur Verwendung des XML-Standards .....</b>	<b>9</b>
2.1 Voraussetzung für die Verwendung von XML .....	9
2.2 Die GS1 XML-Standards .....	12
2.2.1 Implementierungs-Paket .....	13
2.2.2 Business Message Standard (BMS) .....	13
2.3 Mapping von UML zu XML .....	14
<b>3 GS1 XML-Publikationsstrategie .....</b>	<b>17</b>
<b>4 GS1-Namensräume .....</b>	<b>18</b>
4.1 GS1-Namensraum-Struktur .....	18
4.2 Präfix für GS1-Namensraum .....	18
4.3 XML-Schema Namensraum und Präfix .....	19
<b>5 GS1 XML-Versionierungsstrategie .....</b>	<b>20</b>
5.1 Nebenversion (Minor Version) .....	20
5.2 Hauptversion (Major Version) .....	20
5.3 Darstellung der Versionsnummer .....	21
5.3.1 Versionsnummern in XML-Schemas .....	21
5.3.2 Versionsnummern in Geschäftsnachrichten (XML-Instanzen) .....	22
5.3.3 Versionsnummern in den Geschäftsnachrichten und im Standard Business Document Header .....	22
<b>6 Kontext .....</b>	<b>24</b>
6.1 Kontext-Kategorien .....	24
6.2 Abbildung von Kontext im GS1 XML-Standard .....	25

<b>7</b>	<b>Code-Listen</b> .....	<b>26</b>
7.1	Externe Code-Listen .....	26
7.2	Interne Code-Listen.....	26
7.2.1	Interne Code-Listen im Kontext.....	27
7.2.2	Versionierung der internen Code-Listen .....	27
<b>8</b>	<b>Vordefinierte XML-Datentypen (built-in types)</b> .....	<b>28</b>
8.1	Handhabung von Datum und Zeit .....	30
8.1.1	Anwendung von Millisekunden.....	31
8.1.2	Anwendung von Zeitzonen.....	31
<b>9</b>	<b>Standard Business Document Header</b> .....	<b>33</b>
<b>10</b>	<b>Nachrichten-Architektur</b> .....	<b>34</b>
<b>11</b>	<b>Nachrichten-Schicht</b> .....	<b>36</b>
11.1	Proxy-Dateien .....	36
11.2	Geschäftsnachrichten .....	37
11.3	Common Library.....	37
<b>12</b>	<b>Extension-Mechanismus</b> .....	<b>39</b>
12.1	Allgemeine Komponenten einer Extension .....	39
12.2	Validierung der Komponenten einer Extension.....	39
<b>13</b>	<b>XML Werkzeuge</b> .....	<b>40</b>
<b>14</b>	<b>Checkliste für Solution Provider</b> .....	<b>41</b>
<b>15</b>	<b>Anhang: Wesentliche Änderungen zwischen dem Release 1.3.1 und 2.0</b> .....	<b>43</b>
	<b>Impressum</b> .....	<b>44</b>

Abbildung	Seite
Abb. 1: Validierung unterschiedlicher Instanzen mit einem XML-Schema .....	10
Abb. 2: Validierungen von XML-Dokumenten auf Seiten des Senders .....	11
Abb. 3: Validierungen von XML-Dokumenten auf Seiten des Empfängers .....	11
Abb. 4: Entwicklung des XML-Standards .....	14
Abb. 5: Hierarchie der in GS1 XML-Schemas verwendeten Kontext-Kategorien .....	25
Abb. 6: Architektur-Komponenten für GS1-Nachrichten .....	34
Abb. 7: Die Struktur der Extension-Komponente .....	39

### 1 Anwendungsbereich dieses Dokuments

---

#### **1 Anwendungsbereich dieses Dokuments**

Dieser XML-Leitfaden deckt die technischen Aspekte des GS1 Business Message Standard (GS1 BMS) der Version 2.0 ab. Technische Details der früheren Versionen werden in diesem Dokument nicht behandelt. Die wesentlichen Unterschiede zwischen dem aktuellen Release 2.0 und dem vorherigen 1.3.1 sind im Anhang zusammengefasst.

Weitere Informationen zum Release 1.3.1 entnehmen Sie bitte dem Dokument: "How to use EAN.UCC XML Standards version 1.3.1".

Empfehlungen zur Anwendung einzelner Nachrichten für bestimmte Geschäftsprozesse sind nicht Bestandteil dieses Dokuments. Diese Aspekte werden im "Business Message Standard"-Dokument (GS1 BMS) für jede Geschäftsnachricht beschrieben.

## 2 Allgemeiner Überblick zur Verwendung des XML-Standards

---

### 2 Allgemeiner Überblick zur Verwendung des XML-Standards

#### 2.1 Voraussetzung für die Verwendung von XML

Die Anwendung des XML-Standards erfordert die Verwendung bestimmter Software-Werkzeuge. Im einfachsten Fall sind dies ein XML-Prozessor und ein Validierer.

Der XML-Prozessor ist im Allgemeinen in den Applikationen bereits eingebunden und dient dazu, den Inhalt der XML-Datei darzustellen. Er überprüft, ob die XML-Datei wohlgeformt (wellformed) ist. Wohlgeformt bedeutet, dass das XML-Dokument den XML-Syntaxregeln des W3C entspricht.

Der Validierer überprüft, ob das XML-Dokument (auch XML-Instanz genannt) mit dem XML-Schema übereinstimmt. Schemas beinhalten Definitionen und Strukturen, die in dem XML-Dokument verwendet werden. Die XML-Dokumente enthalten in Form von Elementen und Attributen die eigentlichen Daten und die jeweiligen Tags.

Die Software zur Validierung der Dokumente vergleicht die XML-Instanz mit dem zugehörigen XML-Schema. Dabei werden die im XML-Dokument verwendeten Tags auf Gültigkeit überprüft. Zur Kontrolle der gültigen Struktur werden die Daten mit den Schema-Definitionen abgeglichen etc.

In einem XML-Schema lassen sich mehrere Datenstrukturen definieren, die sich in verschiedenen XML-Dokumenten einbinden lassen. Allerdings muss sich der Inhalt dieser Dokumente innerhalb der im Schema definierten Grenzen und Beschränkungen bewegen.

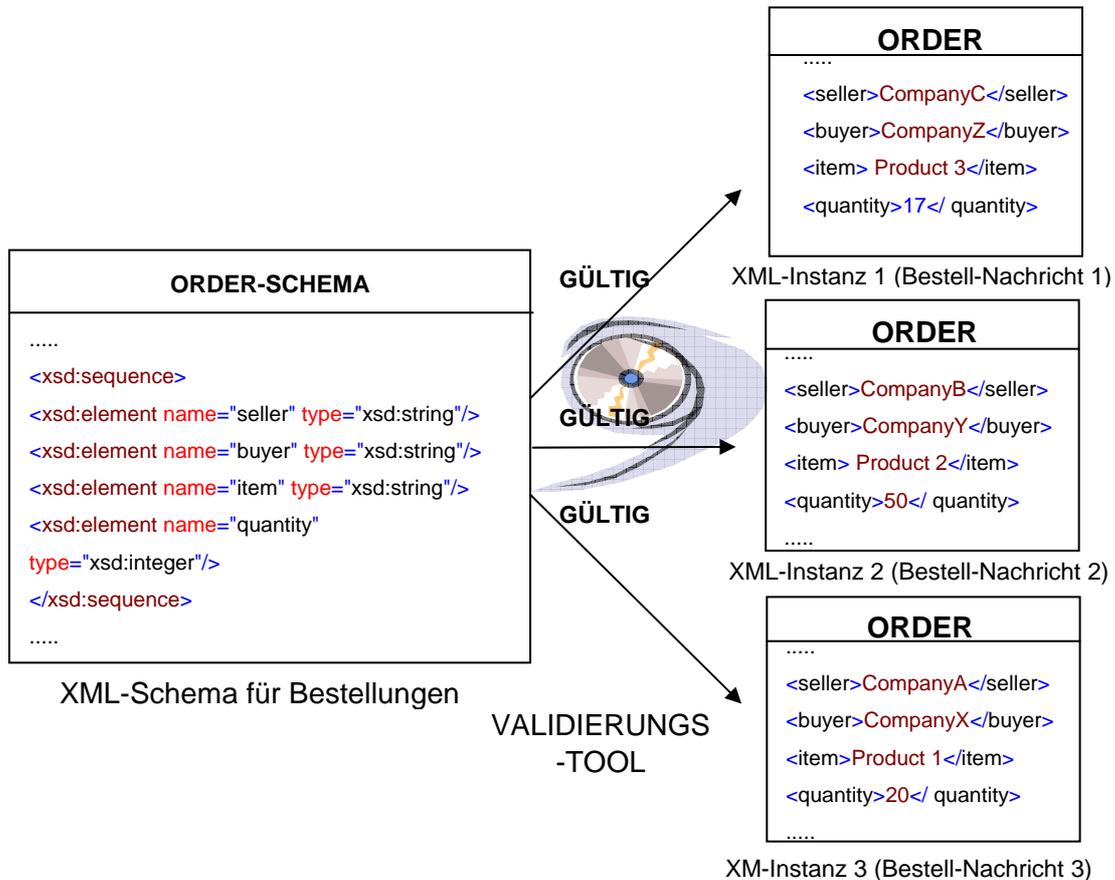


Abb. 1: Validierung unterschiedlicher Instanzen mit einem XML-Schema

Validierungen lassen sich in Geschäftsszenarien verwenden. Jeder der am Datenaustausch beteiligten Handelspartner besitzt eine Kopie des Standard-Schemas, um die Gültigkeit jeder gesendeten bzw. empfangenen XML-Instanz zu überprüfen. Da alle Geschäftsnachrichten, wie zum Beispiel Bestellungen, Lieferavise, Rechnungen etc. unterschiedliche Inhalte und Strukturen aufweisen, gibt es für jeden Nachrichtentyp ein separates Schema. Ungültige Dokumente (die nicht den vorgegebenen Strukturen entsprechen) sind abzulehnen. Auf Seiten des Absenders sollte die Software zur Validierung der XML-Dokumente an der Stelle installiert werden, an der das Dokument erzeugt wird. Dadurch kann jede Nachricht vor dem Versenden geprüft und mögliche Fehler vorab korrigiert werden.

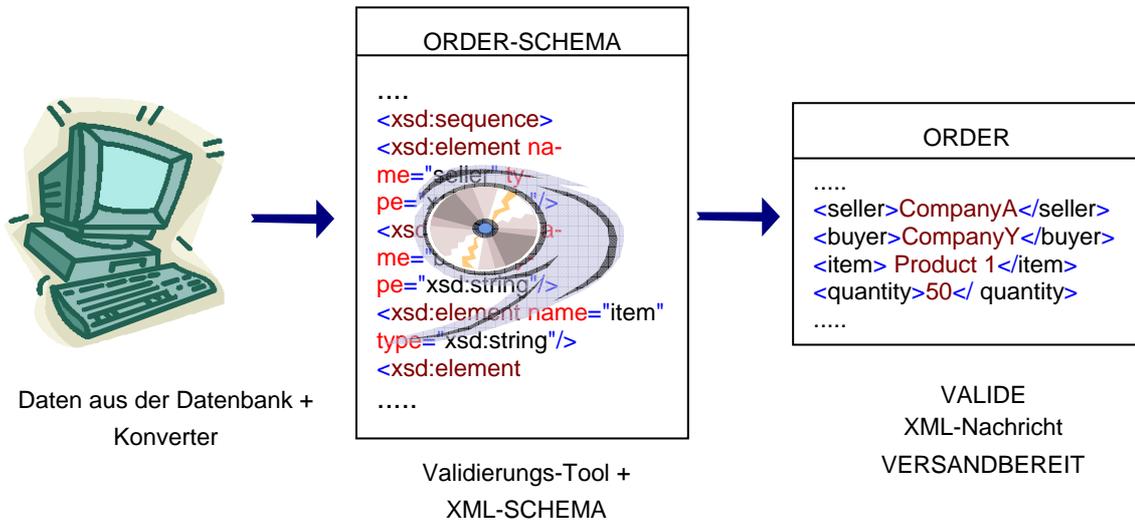


Abb. 2: Validierungen von XML-Dokumenten auf Seiten des Senders

Auf Seiten des Empfängers erfolgt die Prüfung auf Gültigkeit direkt beim Empfang der Nachricht innerhalb der EDI-Software, d. h. bevor die Daten an die Benutzeranwendungen übertragen werden.

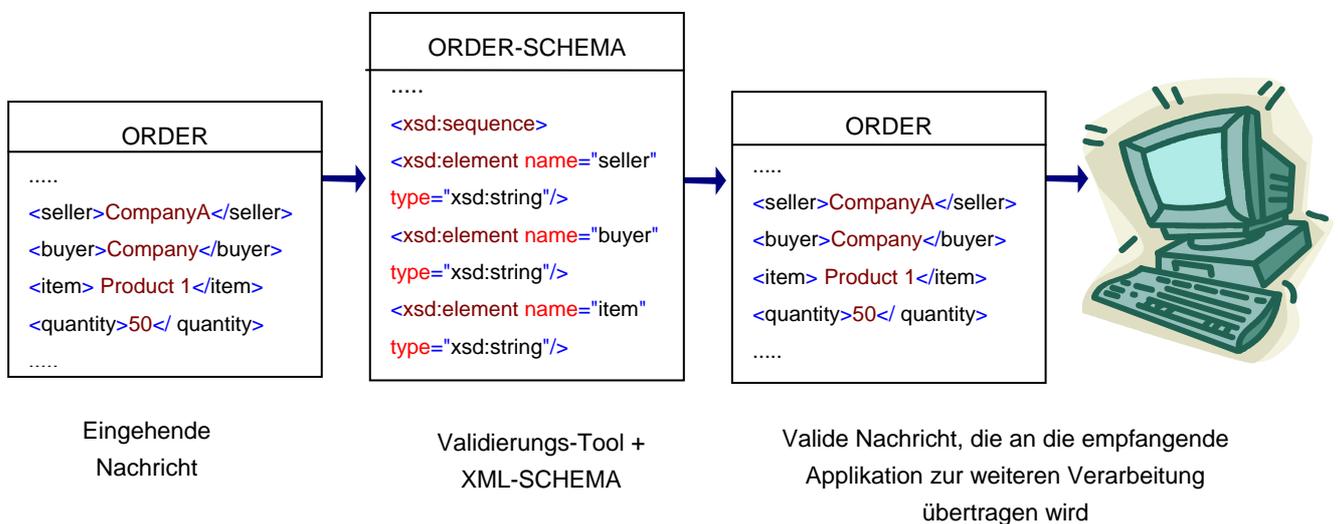


Abb. 3: Validierungen von XML-Dokumenten auf Seiten des Empfängers

Im Allgemeinen werden die nachfolgenden Funktionalitäten in einer XML-Software angeboten:

- Parsen, zum Umwandeln der Daten in ein beliebiges Format
- Aufbereitung bzw. Editierung
- Überprüfung der Wohlgeformtheit
- Validierung

Es sind jedoch auch zahlreiche eigenständige Validierer und Editoren erhältlich.

Für weitere Informationen verweisen wir auf das Kapitel 13 XML-Werkzeuge.

#### 2.2 Die GS1 XML-Standards

GS1 entwickelt zu den jeweiligen Standards zahlreiche Hilfestellungen. Zwei dieser Hilfestellungen sind Schemas sowie die Business Message Standards - kurz BMS-Dokumente.

Der Nachrichten-Standard ermöglicht den Anwendern, ihre Geschäftsdokumente in ein elektronisches Austausch-Format umzuwandeln. XML-Geschäftsdokumente werden "Nachrichten" oder "Dokumente" genannt. Deren Format wird in den "GS1 XML data format message standards" definiert. Der Austausch dieser Geschäftsdokumente ist Bestandteil des elektronischen Handels (e-Commerce).

Ein GS1-Schema beschreibt die Strukturen der XML-Dokumente. Ziel eines Schemas ist die Definition gültiger Bausteine eines XML-Dokuments. Das GS1-Schema-Design hilft dem Entwickler die zu übertragenden Informationen in XML-Dokumente einzubetten.

Die GS1 XML-Standards bestehen aus einem Set von abgestimmten XML-Schema-Modulen. Das aktuelle Release beinhaltet ein Architektur-Upgrade. Diese Änderungen wurden eingearbeitet, um alle Funktionalitäten der XML-Schemaspezifikation in vollem Umfang nutzen zu können.

Die "Business Message Standards" (BMS) sind Arbeitsergebnisse des GSMP und dokumentieren für eine Geschäftsnachricht den verabschiedeten Standard. Jedes BMS enthält die entsprechenden Klassen, Attribute und Werte, die notwendig sind, das Ziel der Nachricht zu erfüllen. Um Klarheit hinsichtlich der Klassen, Attribute und Werte sicherzustellen, wurden präzise Definitionen bereit gestellt und Syntaxeinschränkungen kenntlich gemacht. Zusätzlich beinhaltet der Standard Diagramme auf High-Level sowie auf Detail-Ebene, um den Umfang der Nachricht und die Beziehungen einzelner Elemente untereinander abzubilden. Die Diagramme zeigen Anwendern das Datenmodell und einen Weg, wo und wie sich Erweiterungen für andere Funktionen anbinden lassen. In jedem Standard sind Auszüge aus dem Global Data Dictionary enthalten.

Die GS1 XML-Standards wurden für jede Geschäftsnachricht veröffentlicht und bestehen aus einem Satz von Dokumenten. Anwender müssen zur Implementierung einer Nachricht diesen gesamten Satz herunterladen. Folgende Dokumente sind enthalten:

- Implementierungs-Paket
- Business Message Standard (BMS)

#### 2.2.1 Implementierungs-Paket

Das Implementierungs-Paket ist eine ZIP-Datei die alle XML-Dateien enthält, die zur Validierung einer bestimmten XML-Nachricht notwendig sind. Es enthält:

- **TableofContents.txt** - eine Text-Datei, die alle in dem Paket enthaltenen Dateien auflistet
- **Ordner mit XML-Instanz**, enthält für die jeweilige Nachricht ein XML-Beispiel
- **Ordner mit HTML-Beispiel**, enthält die HTML-Darstellung des XML-Beispiels
- **Schema-Ordner** mit den folgenden Inhalten:
  - EAN.UCC Ordner – enthält zwei Unterordner:
    - ⇒ Common – beinhaltet Schemas aus der common library, mit dem Ziel-Namensraum: `xmlns:eanucc="urn:ean.ucc:2"`. Diese Dateien können in vielen Geschäftsdokumenten, unabhängig vom Kontext, wiederverwendet werden.
    - ⇒ [Name des Geschäftsprozesses] – enthält Schemas eines bestimmten Geschäftsprozesses, die für die Validierung zugehöriger Geschäftsnachrichten notwendig sind. Diese Schemas haben den Ziel-Namensraum:  
`xmlns:[context-specific prefix]="urn:ean.ucc:[context value(s)]:2"`. Weitere Informationen finden Sie in den Kapiteln Kontext und Versionierungsstrategie.
  - **SBDH-Ordner** – enthält die Standard Business Document Header Schemas
  - **Proxy-Schema** – Zusammenfassende Schemas für die bestimmten Geschäftsnachrichten

#### 2.2.2 Business Message Standard (BMS)

Dieses Dokument enthält das Business Solution Document (BSD) sowie das vollständige UML-Modell der Nachricht.

Der Global Data Dictionary Report dokumentiert alle Komponenten des Nachrichtenmodells (Klassen, Rollennamen, Werte für Aufzählungen und Attribute), ihre Definitionen, Kardinalitäten, Längen der Datenfelder sowie den Namen des XML-Schemas, in dem sie definiert sind.

Für die Implementierung einer bestimmten GS1 XML-Nachricht liefert der Business Message Standard dem Anwender alle notwendigen Informationen.

### 2.3 Mapping von UML zu XML

Die Entwicklung des GS1 XML-Standards basiert auf UML-Klassendiagrammen. Ein Klassendiagramm wird in einem bzw. mehreren XML-Schemas abgebildet. Anschließend wird eine Beispiel XML-Datei entwickelt. Diese entspricht den im Schema definierten Datenstrukturen und den im Global Data Dictionary hinterlegten Datendefinitionen. Die Klassendiagramme mit den zugehörigen Modellbeschreibungen und die XML-Beispiele werden für die Erstellung des Business Message Standards verwendet und gemeinsam mit den Schemas veröffentlicht.

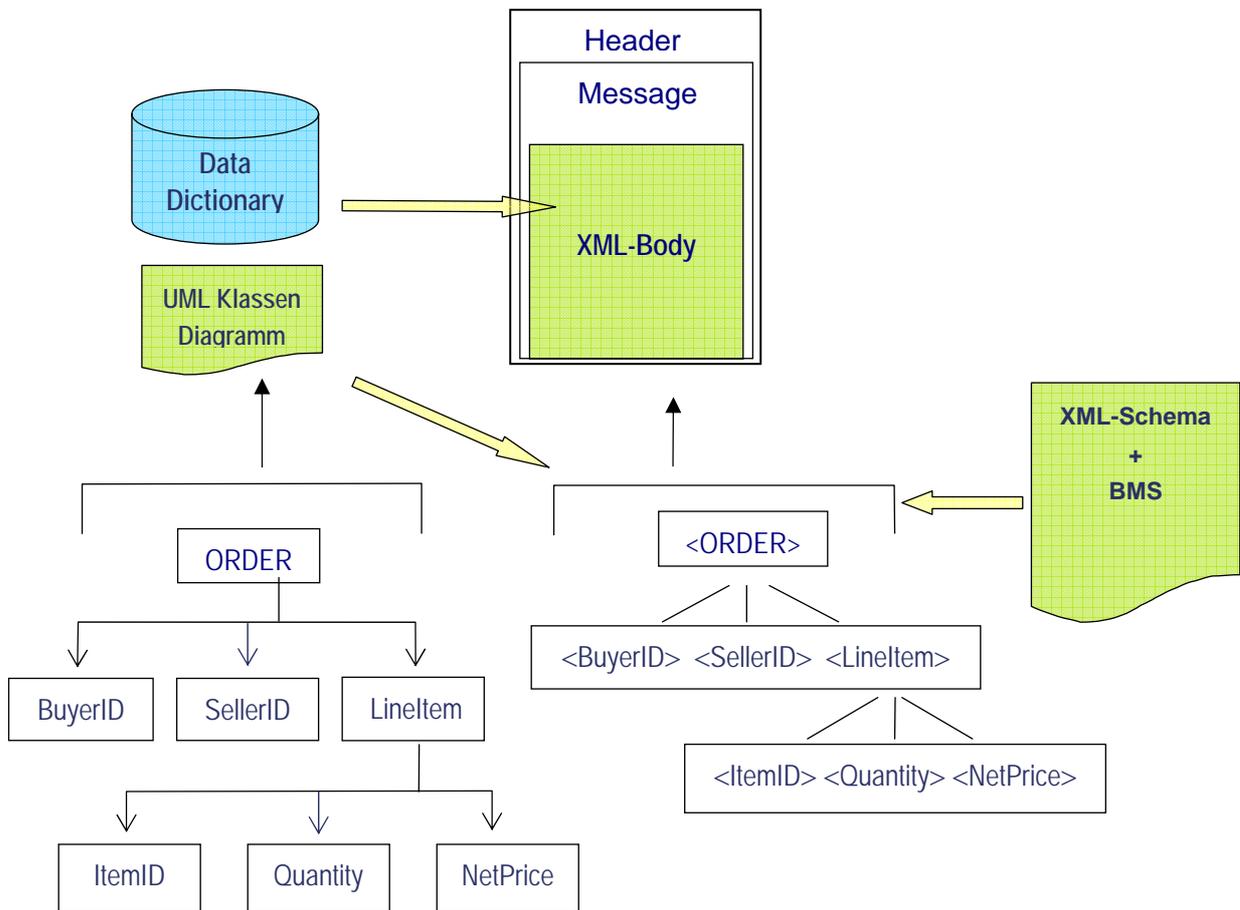


Abb. 4: Entwicklung des GS1 XML-Standards

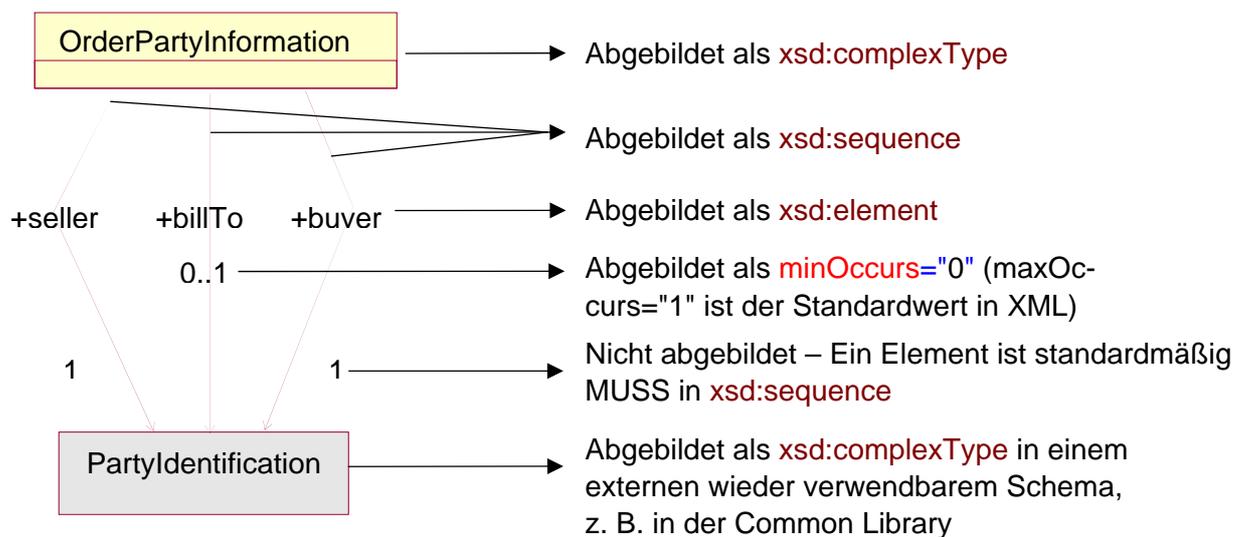
Das tatsächliche Mapping der einzelnen Komponenten von UML nach XML wird bestimmt durch die Beschränkungen der XML-Syntax sowie des gewählten Schema-Design-Modells. Allerdings werden, sofern möglich, auch einige allgemeine Anwendungsregeln genutzt (Ausnahmen dieser Regeln sind selten und werden durch die Syntax-

Beschränkungen ausgelöst). Der Anwender muss nicht alle Details und die Komplexität des UML / XML-Mappings verstehen. Der die Kenntnisse über die wesentlichen Grundsätze können beim Lesen des GS1-Standards hilfreich sein.

Die Basisregeln zur Abbildung von UML-Komponenten in XML werden anhand der drei nachfolgenden Beispiele erklärt:

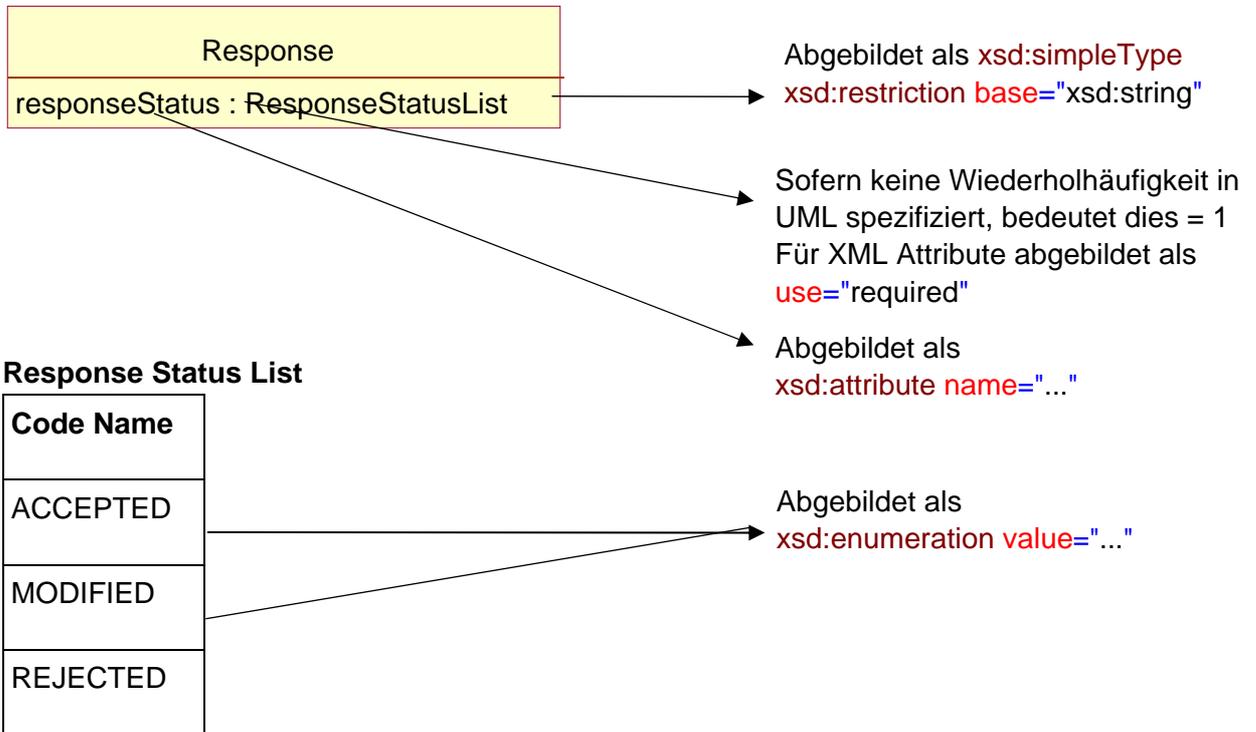
#### Beispiel 1

Abbildung von UML-Komponenten als komplexe XML-Typen (Complex Types) und Elemente



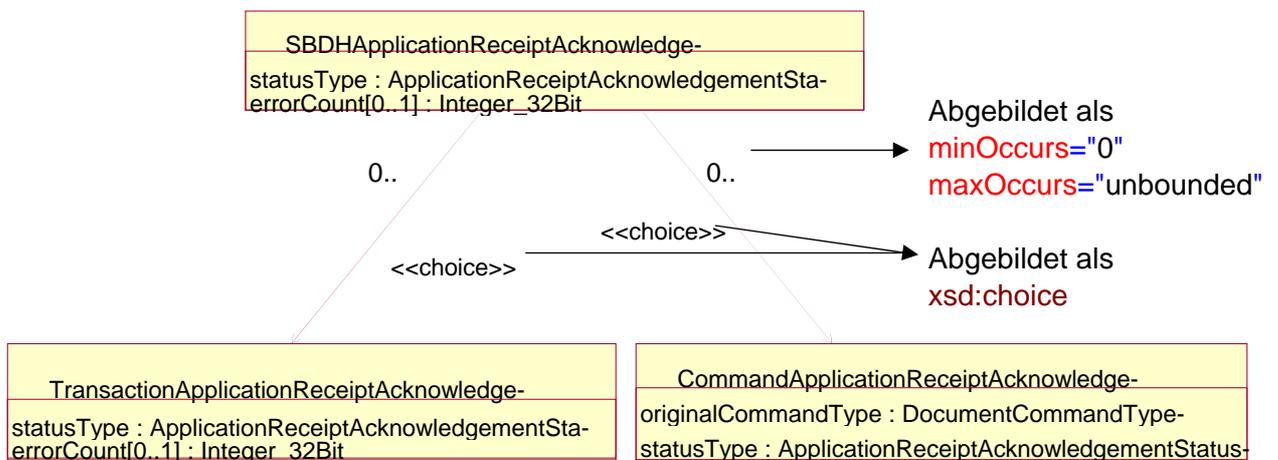
Beispiel 2

Abbildung der UML-Komponenten als einfache XML-Typen (Simple Types) und Attribute



Beispiel 3

Abbildung von UML-Komponenten als XML-Auswahl (Choice)



### 3 GS1 XML-Publikationsstrategie

---

#### 3 GS1 XML-Publikationsstrategie

Die GS1 XML-Standards unterziehen sich eines genauen formellen Freigabe-Verfahrens, um ihre Korrektheit und Integrität sowohl innerhalb der XML-Nachrichten als auch mit den anderen GS1-Standards sicherzustellen.

Sobald die XML-Nachrichten, als Teil des globalen Standardisierungsprozesses (GSMP), fertig gestellt wurden, erfolgt eine weitere Überprüfung der "Information Technical Requirements Group" (ITRG). Nachdem die Gruppe sichergestellt hat, dass die Nachrichten die Geschäftsanforderungen erfüllen und technisch korrekt sind, werden die Nachrichten im Rahmen eines Abstimmungsprozesses von der ITRG-Gruppe formal freigegeben. Zu dieser Zeit erhalten die Standards den Status "ENTWURF" und werden auf der GSMP-Website veröffentlicht. Der Entwurf der Nachrichten wird veröffentlicht, um den Anwendern die Möglichkeit zu geben, den Inhalt zu analysieren und Pilotimplementierungen zu planen und zu testen. Der allgemeine Inhalt der Nachrichten wird sich wahrscheinlich nicht ändern, aber es kann die Notwendigkeit bestehen, vor der finalen Release-Freigabe nochmals Änderungen vorzunehmen, sodass der Entwurf nicht als Basis für die eigentliche Implementierung genutzt werden sollte.

Den finalen Status erhält der Standard, sobald er durch das GS1 Management Board ratifiziert wurde. In der Regel geschieht dies ein- bis zweimal pro Jahr und hängt von der Anzahl der entwickelten Nachrichten und von größeren architektonischen Änderungen ab, die ein neues Release aller XML-Nachrichten erfordern.

Nach dieser Ratifizierung werden die Nachrichten als endgültiger Standard auf den GSMP-Webseiten veröffentlicht und können für die endgültige Implementierung verwendet werden.

Alle vorherigen Releases des Standards (von 1.0 bis 1.3.1) können unter dem nachfolgenden Link heruntergeladen werden:

[http://www.gs1.org/services/gsmp/kc/ecom/xml/xml\\_archive.html](http://www.gs1.org/services/gsmp/kc/ecom/xml/xml_archive.html) und alle späteren Versionen (2.0 und höher) unter:

[http://www.gs1.org/services/gsmp/kc/ecom/xml/xml\\_bms.html](http://www.gs1.org/services/gsmp/kc/ecom/xml/xml_bms.html).

- ✓ **Hinweis:** Ab dem Release 2.0 werden alle Nebenversionen (minor versions) der XML-Schemas als Teil der Hauptversion (major version) des Standards veröffentlicht. Die nächsten Hauptversionen werden nur noch die aktuellsten Nebenversionen (minor versions) des vorherigen Release enthalten.

### 4 GS1-Namensräume

---

#### 4 GS1-Namensräume

##### 4.1 Struktur der GS1-Namensräume

Die GS1-Namensräume spiegeln den Kontext und die Hauptversion (major version) der Schema-Komponenten wider. Die GS1-Namensräume entsprechen dem Format des Uniform Resource Names (URN - einheitlicher Name für Ressourcen). Alle URNs weisen die folgende Struktur auf:

$$\langle \text{URN} \rangle ::= \text{"urn:"} \langle \text{NID} \rangle \text{" :"} \langle \text{NSS} \rangle$$

$\langle \text{NID} \rangle$  **N**amespace **I**dentifier (Präfix - Namensraum)

$\langle \text{NSS} \rangle$  **N**amespace **S**pecific **S**tring (Suffix - numerischer oder alpha-numerischer String)

In allen GS1-Namensräumen wird "ean.ucc" als NID verwendet. Der von GS1 zugewiesene NSS aller URNs hat die folgende hierarchische Struktur und spiegelt die wesentlichen Kontext-Kategorien und die Hauptversionsnummer (major version) wieder:

$$\text{urn:ean.ucc:} \_\_\_\_\_\_ \text{:} \_\_\_\_\_\_ \text{:} \_\_\_\_\_\_ \text{:} \_\_\_\_\_\_$$

BP      IC      GP      Major Version

BP – Business Process

IC – Industry Classification

GP – geopolitischer Kontext

✓ **Hinweis:** Der NSS ist in den GS1-Namensräumen von der Groß- und Kleinschreibung abhängig.

In jeder Komponente sind alle Kontext-Kategorien definiert - entweder "spezifisch", "global" oder "alle". Sofern für die Kontext-Kategorie "global" gilt, wird die Kategorie im Namensraum nicht näher bestimmt. So hat zum Beispiel eine für alle geographischen Regionen und Industriesektoren allgemeingültige Komponente im Global Data Synchronization Network (GDSN) den folgenden Namensraum: **gdsn="urn:ean.ucc:gdsn:2"**.

##### 4.2 Präfixe für GS1-Namensräume

GS1 XML-Schemas verwenden keine Standardwerte für Namensräume. Die GS1-Informationskomponenten müssen einem Namensraum zugeteilt werden, der den Kontext widerspiegelt, in dem er definiert wurde. Dieser Namensraum (Kontext) sollte für jede Komponente deutlich angegeben werden.

Der GS1-Standard schreibt die folgenden Standard-Präfixe für Namensräume vor. Im Allgemeinen leitet sich der Name des Präfixes vom Geschäftsprozess ab:

- "pay" für PAY (Zahlungsprozess), z. B.: **pay**="urn:ean.ucc:pay:2"
- "align" für ALIGN (Stammdatenprozess, z. B.: **align**="urn:ean.ucc:align:2"
- "plan" für PLAN (Planungsprozess), z. B.: **plan**="urn:ean.ucc:plan:2"
- "deliver" für DELIVER (Lieferprozess), z. B.: **deliver**="urn:ean.ucc:deliver:2"
- "order" für ORDER (Bestellprozesse), z. B.: **order**="urn:ean.ucc:order:2"
- "gdsn" für Global Data Synchronisation (Artikelstammdatenprozesse im GDSN), z. B.: **gdsn**="urn:ean.ucc:gdsn:2"
- "eanucc" für COMMON (alle allgemein gültigen Komponenten), z. B.: **eanucc**="urn:ean.ucc:2"

Lokale Erweiterungen (Extension) haben ihre eigenen Namensräume und eigene Standard-Präfixe, so nutzt zum Beispiel Schweden für spezielle Komponenten für Handelseinheiten einen separaten Namensraum mit dem Präfix "sw":

**sw**="urn:ean.ucc:align:sweden:2"

Um die Einheitlichkeit und Interoperabilität im GS1-System zu gewährleisten, sollte für jeden Namensraum-URI ein einzelner, eindeutiger Präfix verwendet werden. Die gleiche Vorgehensweise empfiehlt GS1 auch für die XML-Dokumente. Sofern möglich, sollten Entwickler an dieser Stelle die gleichen Präfixe verwenden, wie sie in den Standard GS1 XML-Schemas vergeben wurden.

Aus Sicht der XML-Syntax lässt sich jedes beliebige Präfix nutzen. Es muss lediglich auf den richtigen Namensraum verweisen. Die Zuordnung nicht-standardkonformer Präfixe würde die globale Interoperabilität der Standards gefährden.

Außerdem reagieren einige XML Mapping- und Verarbeitungs-Tools sensibel auf unterschiedliche Präfixe. Sofern Geschäftspartner keine abgestimmten Präfixe verwenden, kann dies zu ernsthaften Problemen im Validierungsprozess und der Verarbeitung führen.

#### 4.3 XML-Schema Namensraum und Präfix

Für die XML-Schema-Komponenten wird das World Wide Web Consortium (W3C) URL <http://www.w3.org/2001/XMLSchema> als Namensraum verwendet. Auch hier wurde der Präfix für den Namensraum im GS1-System standardisiert und alle existierenden Schema-Module nutzen xsd: für den XML-Schema Namensraum.

#### 5 GS1 XML-Versionierungsstrategie

Seit Beginn des Release 2.0 wird die Versionierung der GS1 XML-Schemas nicht mehr wie bei früheren Versionen (1.1, 1.3, 1.3.1 und 1.3.2) an die Veröffentlichung der GS1 BMS-Dokumente gekoppelt. Dies bedeutet, dass alle Schemas, einschließlich der für Geschäftsdokumente und die Schemas der Common Library nicht unbedingt die gleiche Versionsnummer haben müssen.

Dank dieser Politik erfordert eine Veränderung innerhalb der Common Library keine erneute Veröffentlichung aller Schemas, wie es in früheren Versionen der Fall war. Darüber hinaus wird die Abwärtskompatibilität zwischen zwei aufeinander folgenden Releases von GS1-Schemas gewährleistet.

Abwärtskompatibilität bedeutet, dass der Absender eine Geschäftsnachricht auf Basis der alten Schemas erzeugen und versenden kann. Der Empfänger dieser Nachricht kann diese erfolgreich gegen ein aktuelles Schema validieren.

##### 5.1 Nebenversion (Minor Version)

Nebenversionen sind zwischenzeitliche XML-Schema-Releases, die nur Änderungen enthalten, die mit der vorhandenen Version des Schemas abwärtskompatibel sind.

Änderungen, die in einer Nebenversion enthalten sind:

- Hinzufügen optionaler Elemente oder optionaler Attribute
- Änderung der Statusangabe bei Attributen vom Status "MUSS" auf "KANN"
- Änderung der Wiederholbarkeit von Elementen im alten Schema von [0..1] auf [0..\*] im neuen Schema
- Änderung der Wiederholbarkeit von Elementen im alten Schema von [1..1] auf [1..\*] im neuen Schema
- Hinzufügen eines Codes zu einer Code-Liste
- Korrekturen von Druck- und Schreibfehlern

##### 5.2 Hauptversion (Major Version)

Um die Implementierung und die laufende Versionierung der GS1 XML-Schemas zu erleichtern, besitzen alle Schemas innerhalb eines Geschäftsprozesses die gleiche Hauptversionsnummer. Sofern ein einziges Schema für einen bestimmten Geschäftsprozess in eine höhere Hauptversion geändert werden muss, sind alle Schemas innerhalb dieses Geschäftsprozesses auf die nächst höhere Hauptversion zu aktualisieren. Diese Geschäftsprozesse werden durch die Anwender aus den Unternehmen definiert.

Änderungen, die eine Erhöhung der Hauptversion erfordern:

- Änderung der Statusangabe bei Attributen vom Status "KANN" auf "MUSS"
- Hinzufügen eines MUSS-Elements
- Änderung des XML-MUSS-Namens von Attributen oder Elementen
- Änderung der Wiederholbarkeit von Elementen im alten Schema von [0..\*] auf [0..1] im neuen Schema
- Änderung der Wiederholbarkeit von Elementen im alten Schema von [1..\*] auf [1] im neuen Schema
- Änderung der Reihenfolge von Elementen in einer <xsd:sequence>

### 5.3 Darstellung der Versionsnummer

#### 5.3.1 Versionsnummern in XML-Schemas

Die Nummer der Hauptversion wird im Namensraum des Schemas angegeben.

##### Beispiel

urn:ean:gdsn:fmcg:1

Die Nebenversion wird in dem Versions-Attribut des xsd:schema-Elements angegeben. Für eine einfachere Implementierung spiegelt der Wert des Versions-Attributes die Nummer der Haupt- und Nebenversion des Schemas wider.

##### Beispiel

```
<xsd:schema targetNamespace="urn:ean.ucc:gdsn:fmcg:2"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="urn:ean.ucc:gdsn:fmcg:2"  
elementFormDefault="unqualified" attributeFormDefault="unqualified" version="2.3">
```

Dieser Ansatz gewährleistet die Abwärtskompatibilität. Aufgrund der Tatsache, dass die Nebenversionen nicht in dem Namensraum des Schemas angegeben werden, lassen sich XML-Dokumente gegen ältere Nebenversionen validieren. Alle Schemas, die dieses Schema einbinden ('include') bedürfen keiner Änderung, da der Ziel-Namensraum (target namespace) der eingebundenen Komponenten mit dem Ziel-Namensraum des eingebundenen Schemas identisch ist.

#### 5.3.2 Versionsnummern in Geschäftsnachrichten (XML-Instanzen)

Seitdem die Hauptversion im Namensraum angegeben wird, enthält die Geschäftsnachricht (XML-Instanz) standardmäßig die Nummer der Hauptversion. Allerdings benötigt das empfangende Anwendungssystem die vom Geschäftspartner verwendete Haupt- und Nebenversion. Daher ist die vollständige Version in der XML-Instanz anzugeben.

GS1-Geschäftsnachrichten verwenden das "documentStructureVersion"-Attribut. Dieses wird als komplexer Datentyp (complex Type) unter "DocumentType" definiert und spiegelt die vollständige Schema-Version der Geschäftsnachricht wieder. Das Attribut "documentStructureVersion" gibt zum Beispiel für eine Bestellung "Order.xml" die Versionsnummer des Schemas "Order.xsd" an. Die Versionen der Common Library-Schemas werden in dem XML-Dokument nicht angegeben, da sie sich aufgrund der Abwärtskompatibilität nicht auf den Validierungsprozess auswirken.

Dieser Datentyp wird wie folgt definiert:

```
<xsd:complexType name="DocumentType" abstract="true">
  <xsd:attribute name="contentVersion" type="eanucc:VersionType" default="2.0"/>
  <xsd:attribute name="documentStructureVersion" type="eanucc:VersionType" default="2.0"/>
  <xsd:attribute name="lastUpdateDate" type="xsd:date"/>
  <xsd:attribute name="creationDate" type="xsd:dateTime" use="required"/>
  <xsd:attribute name="documentStatus" type="eanucc:DocumentStatusListType" use="required"/>
</xsd:complexType>
```

Bitte beachten Sie, dass ab der Version 2.0, dem Attribut "documentStructureVersion" kein fester Wert mehr zugewiesen wird, wie es in früheren Versionen der Fall war. Stattdessen wird der Wert des Attributes von den Handelspartnern übermittelt, um die von ihnen verwendeten Haupt- und Nebenversion anzugeben.

#### 5.3.3 Versionsnummern in den Geschäftsnachrichten und im Standard Business Document Header

Das Schema des Datei-Kopfs, der sog. Standard Business Document Header (SBDH), enthält das Element "TypeVersion". Dieser dient als Platzhalter für die Version der Geschäftsnachricht oder für Nachrichten, die mit einem Nachrichten-Kopf versendet wurden. Die SBDH-Beschreibung fordert dieselbe Versionsnummer für alle unter einen gemeinsamen Nachrichten-Kopf versendeten Dokumente. Um diesen Anforderungen zu entsprechen, empfiehlt GS1, dass nur die Hauptversionsnummer der Geschäftsnachricht in den "TypeVersion"-Element des SBDH angegeben wird. Dadurch ist es möglich, nur Doku-

mente (nur Rechnungen, nur Bestellungen etc) des gleichen Typs mit der gleichen Hauptversionsnummer in einer Nachricht zu versenden. Die Nebenversionen dieser Dokumente kann unterschiedlich sein, da sie innerhalb der gleichen Hauptversion kompatibel sind.

#### 6 Kontext

Geschäftsnachrichten werden normalerweise immer unter bestimmten Alltagsbedingungen eingesetzt. Hierbei gibt es im Aufbau der Nachrichten zahlreiche strukturelle und inhaltliche Gemeinsamkeiten, aber auch viele Unterschiede. GS1-Schemas basieren auf einer Begriffswelt, die diese gemeinsamen und unterschiedlichen Umstände wiedergibt. Um diese Komplexität zu bewältigen, wurde eine Kontext-Methodik eingeführt. Diese reflektiert die Umstände, in denen der Austausch von Geschäftsinformationen stattfindet.

Als Beispiel lässt sich die Beschreibung einer Handelseinheit erwähnen. Es gibt gewisse Attribute, die für jede Art von Produkten Gültigkeit besitzen, z. B. Markenname und Identifikationsnummer. Die pharmazeutische Industrie hingegen benötigt spezielle Attribute, die von den Attributen der Fahrzeugindustrie und der Textilbranche abweichen. Diese verschiedenen Attribute sind kontextspezifisch, während der Kontext in diesem Fall der Industriesektor ist, durch den der Bedarf an gewissen Informationen bestimmt wird. Die allgemeinen Attribute sind unabhängig vom Kontext.

#### 6.1 Kontext-Kategorien

Der Kontext, in dem geschäftliche Zusammenarbeit stattfindet, kann mit Hilfe einer Reihe von Kategorien und entsprechenden Werten angegeben werden. Es gibt insgesamt acht Kategorien, derzeit werden aber in GS1 XML nur drei dieser Kategorien verwendet:

- Geschäftsprozesse, in denen eine Zusammenarbeit stattfindet, z. B. Bestellung, Lieferung etc.
- Industriesektor bzw. Branche, in denen Geschäftspartner involviert sind, z. B. FMCG (Fast Moving Consumer Goods – Konsumgüter) , Hardlines (Werkzeuge) etc.
- Geopolitisch, womit die geografischen Faktoren beschrieben werden, die die Geschäftssemantik beeinflussen könnte. Hierbei kann es sich entweder um länderspezifische Faktoren (z. B. nur für Frankreich oder Schweden), um bestimmte Wirtschaftsregionen (z. B. NAFTA, Europäische Union) oder letztlich um Faktoren, die in der ganzen Welt anwendbar sind, handeln.

Je nach Bedarf lassen sich in zukünftigen Versionen des GS1 XML-Standards weitere Kategorien überführen.

Komponenten, die in den gesamten Kontext-Kategorien verwendet werden können, werden mit dem Wert "In All Contexts" belegt. In diesem Fall wird aus Gründen der Übersichtlichkeit kein Wert angegeben.

Einige Komponenten können mit mehr als einem Wert je Kategorie in Verbindung gebracht werden, z. B. "Pharma" und "Bergbau".

#### 6.2 Abbildung von Kontext im GS1 XML-Standard

Im Allgemeinen handelt es sich bei einem Kontext um ein nicht-hierarchisches Konzept, da alle Kategorien auf gleicher Ebene mit gleicher Bedeutung stehen. Allerdings wurde in den XML-Schemas eine Hierarchie eingeführt, um die Handhabung des Kontext zu erleichtern.

Die Reihenfolge der Hierarchie beginnt mit der Kategorie "Geschäftsprozess", gefolgt von der "Branche" und der geopolitischen Kategorie.

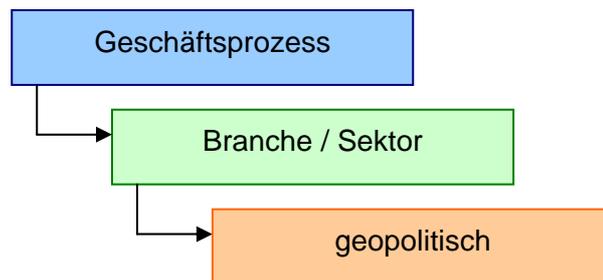


Abb. 5: Hierarchie der in GS1 XML-Schemas verwendeten Kontext-Kategorien

In den XML-Schemas wird der Kontext durch die Namensräume (Kapitel 4) abgebildet.

#### 7 Code-Listen

In GS1 XML-Nachrichten gibt es zwei Arten von Code-Listen: externe und interne Code-Listen.

##### 7.1 Externe Code-Listen

Externe Code-Listen werden in der Regel von anderen Normungsgremien außerhalb der GS1-Gemeinschaft definiert und gepflegt, z. B. ISO. Beispiele sind:

- Länder-Codes – ISO 3166-1:1997, definiert im "Country.xsd"-Schema, als ISO3166\_1CodeType
- Codes für staatliche Untereinheiten – ISO 3166-2:1998, definiert im "Country.xsd"-Schema, als ISO3166\_2Code Type
- Währungs-Codes – ISO 4217:2001, definiert im "MonetaryAmount.xsd"-Schema, als CurrencyISOCODEType

Diese Code-Listen sind ein Teil der so genannten "Common Library". Sie sind als Zeichenketten definiert und auf eine passende Anzahl von Zeichen begrenzt. Sie enthalten jedoch nicht die eigentlichen Code-Werte. Der Grund für diese Entscheidung liegt darin, dass die Aufzählung der Code-Werte zu Problemen bei der Pflege führen würde, da diese außerhalb des GS1-Systems erfolgt. Zudem würde auch das Urheberrecht verletzt.

Detaillierte Informationen zur Verwendung dieser Listen können auf der Webseite der International Standard Organisation unter [www.iso.org](http://www.iso.org) herunter geladen werden.

##### 7.2 Interne Code-Listen

Interne Code-Listen sind jene, die innerhalb des GS1-Systems entwickelt und gepflegt werden. Ab dem Release 2.0 sind diese Code-Listen in separaten Schemas definiert und nicht mehr in dem Schema der Common Library integriert. Diese Vorgehensweise führt zwar zum Verlust einiger Vorteile, besonders die Fähigkeit für den Parser sich an strikten Validierungsregeln zu halten. Die in das Schema eingebetteten Code-Listen lassen sich nicht ändern, ohne den Validierungsprozess zu beeinflussen. Im Gegensatz dazu lassen sich jedoch diese extern definierten Code-Listen durch Referenzierung in ein Hauptschema einbinden und verschiedene Versionen als Basis für Validierungen nutzen.

Auf der anderen Seite wird sichergestellt, dass eine Änderung im Code-Listen-Schema nicht zu einem neuen Release des gesamten GS1-Schemas führt und damit auch nicht ein zusätzliches Upgrade für den Anwender erfordert.

Code-Listen werden als `xsd:enumeration` definiert. Die Code-Listen werden in das Schema, in dem sie verwendet werden, eingebunden (`include`) oder importiert (`import`).

Dieser Ansatz bietet dem Anwender folgende Vorteile:

Verringerte Wartungskosten, schnellere Implementierung von Änderungen in den Code-Listen, weniger Schema-Versionen und eine einfachere Veröffentlichung. Darüber hinaus ist die Anwendung der Code-Listen ähnlich der neuen Verwendung der Komponenten.

#### 7.2.1 Interne Code-Listen im Kontext

Code-Listen-Schemas folgen mit Rücksicht auf die Namensräume den gleichen Richtlinien wie auch alle anderen GS1-Schemas. Wird eine bestimmte Code-Liste übergreifend in unterschiedlichen Kontexten benötigt, ist sie innerhalb des Common Library Namensraum zu definieren, z. B. `TimePeriodList.xsd` oder `AllowanceChargeList.xsd`.

Sofern Code-Listen in nur einem Kontext verwendet werden, sind sie nur in dem Schema zu definieren, in dem sie verwendet werden. So werden zum Beispiel `ReplenishmentRequestStatusList.xsd` und `CollaborationPriorityCodelist.xsd` im Namensraum "Plan" definiert.

Dieser Ansatz stellt eine Abwärtskompatibilität sicher. Da die Nebenversionen nicht im Namensraum des Schemas erscheinen, lassen sich Dokumente gegen ältere Nebenversionen validieren. Jedes Schema, das dieses Schema einbindet (`Include`), benötigt keinerlei Änderungen, da der Zielnamensraum der eingebundenen Komponente mit dem Zielnamensraum des eingebundenen Schemas identisch ist.

#### 7.2.2 Versionierung der internen Code-Listen

Jede Code-Liste wird in einer eigenständigen Schema-Datei definiert und getrennt versioniert. Bei Änderungen am Code-Listen-Schema erfolgt die Versionierung auf gleiche Art und Weise wie bei jeder anderen Komponente. Sobald neue Code-Definitionen durch die Business Requirement Group (BRG) genehmigt wurden, wird das Code-Listen-Schema aktualisiert und die Nebenversionsnummer innerhalb des Code-Listen-Schemas erhöht. Damit wird dem Anwender ein effizienter Zugriff auf die Codes ermöglicht.

Die Aufnahme neuer Code-Werte führen zu einem Wechsel der Nebenversionen. Zu löschende Code-Werte oder Änderungen der Definitionen werden nur unter einer neuen Hauptversion implementiert.

#### 8 Vordefinierte XML-Datentypen (built-in types)

Der vom World Wide Web Consortium (W3C) entwickelte XML-Standard schließt in seiner Spezifikation 44 Datentypen ein. Diese Datentypen können in XML-Schemas verwendet werden, ohne dass sie vorab definiert werden müssen. Diese Datentypen sollten von allen XML-fähigen Softwaretools bedingungslos verstanden werden. Die vordefinierten Datentypen weisen bestimmte Standard-Aspekte auf, um sie im Schema abzubilden oder ihre Wertebereiche einzugrenzen.

Im GS1-Standard werden nur wenige vordefinierte Datentypen verwendet. Einige der W3C-Datentypen werden im Business Message Standard (GS1 BMS) und in den vom Standard verwendeten Schemas unterschiedlich dargestellt. Dieser Unterschied ist darauf zurückzuführen, dass die UML-Darstellung aus dem Business Dokument übernommen wurde, um von der zukünftig gewählten Syntax unabhängig zu sein.

Die Tabelle zeigt die im GS1-Standard verwendeten W3C-Datentypen, ihre Darstellung in den Schemas sowie im BMS:

<b>W3C-Datentyp</b>	<b>W3C XSD</b>	<b>GS1 BMS-Datentyp</b>
String		AN String
Boolean	true/false 1/0	Y/N
Decimal	totalDigits=5 fractionDigits=3	Numeric 5,3
Integer	totalDigits=5	Number 5
nonNegativeInteger	totalDigits=5	Number 5

W3C-Datentyp	W3C XSD	GS1 BMS-Datentyp
dateTime	<p>ISO8601</p> <p><u>Jahr:</u> YYYY (z. B. 1997)</p> <p><u>Komplettes Datum:</u> YYYY-MM-DD (z. B. 1997-07-16)</p> <p><u>Jahr und Monat:</u> YYY-MM (z. B. 1997-07)</p> <p><u>Ges. Datum plus Stunden und Minuten:</u> YYY-MM-DDThh:mmTZD (z. B. 1997-07- 16T19:20+01:00)</p> <p><u>Ges. Datum plus Stunden, Minuten und Sekunden:</u> YYY-MM-DDThh:mm:ssTZD (z. B. 1997-07- 16T19:20:30+01:00)</p> <p><u>Ges. Datum plus Stunden, Minuten, Sekunden und dezimale Nachkommastellen der Sekunden:</u> YYY-MM-DDThh:mm:ss.sTZD (z. B. 1997-07- 16T19:20:30.45+01:00)</p> <p>wo:</p> <p><b>YYYY</b> = vierstellig Jahr</p> <p><b>MM</b> = zweistellig Monat (01 = Januar)</p> <p><b>DD</b> = zweistellig Tag (01 bis 31)</p> <p><b>hh</b> = zweistellig Stunde (00 bis 23) (am/pm ist nicht erlaubt)</p> <p><b>ss</b> = zweistellig Sekunden (00 bis 59)</p> <p><b>s</b> = ein oder mehrstellige Darstellung der Nachkommastellen der Sekunden</p> <p><b>TZD</b> = Bezeichner Zeitzone (Z oder +hh:mm oder -hh:mm)</p>	CCYYMMDDThh:mm:ss

Time	ISO 8601 Strukturen bezüglich der Zeit:  hh:mm:ss.s - Erläuterungen und Beispiele - siehe <u><a href="#">dateTime</a></u>	hh:mm:ss
date	ISO 8601 Strukturen bezüglich des Datums:  YYYY-MM-DD - Erläuterungen und Beispiele - siehe <u><a href="#">dateTime</a></u>	CCYYMMDD

Von GS1 derzeit nicht verwendete W3C-Datentypen:

- normalizedString
- token
- Name
- NCName
- ID
- IDREF
- IDREFS
- ENTITY
- ENTITIES
- NMTOKEN
- NMTOKENS
- nonPositiveInteger
- negativeInteger
- long
- language
- int
- short
- byte
- unsignedLong
- unsignedInt
- unsignedShort
- unsignedByte
- positiveInteger
- double
- duration
- gYearMonth
- gYear
- gMonthDay
- gDay
- gMonth
- hexBinary
- Base64Binary
- anyURI
- QName
- NOTATION

#### 8.1 Handhabung von Datum und Zeit

Innerhalb des GS1 XML-Standards werden zur Darstellung der Zeit zwei vordefinierte Datentypen (built-in types) verwendet: "dateTime" und "time". Die Beschreibung und der Gebrauch dieser Datumsformate und Zeitangaben sind im Artikel 5.3 der ISO 8601 festgelegt.

Weitere Hinweise finden sie unter den folgenden Links:

<http://www.w3.org/TR/xmlschema-2/#time>

<http://www.w3.org/TR/xmlschema-2/#dateTime>

Zusätzlich lassen sich Millisekunden und Informationen zu den Zeitzonen angeben.

#### 8.1.1 Anwendung von Millisekunden

Beide Datentypen, "dateTime" und "time", ermöglichen die Präzisierung von Zeitangaben durch Verwendung zusätzlicher Stellen. Sofern gewünscht, lassen sich Millisekunden im Format ss.s darstellen.

ss.s zeigt zwei Stellen für die Sekunden (00 bis 59), gefolgt von einer oder mehreren Stellen, die den dezimalen Nachkommateil der Sekunden (in Millisekunden) darstellen. Der Nachkommateil wird durch die Verwendung eines Punktes als Trennzeichen von den zwei Stellen für die Angabe der Sekunden getrennt.

Für den Nachkommateil wird eine beliebige Anzahl von Stellen unterstützt. Innerhalb der GS1 XML-Nachrichten sollten für die Darstellung der Millisekunden jedoch nur drei Stellen verwendet werden.

##### Beispiel 1

Für das Attribut "creationDate" vom XSD Datentyp "dateTime" sind in allen Versionen des GS1-Standards folgende Werte zulässig:

- **creationDate="2003-03-22T09:30:47"**  
Das Beispiel zeigt 47 Sekunden und 0 Millisekunden.
- **creationDate="2003-03-22T09:30:47.0"**  
Das Beispiel zeigt 47 Sekunden und 0 Millisekunden und ist gleichbedeutend mit dem ersten Beispiel.
- **creationDate="2003-03-22T09:30:47.233"**  
Das Beispiel zeigt 47 Sekunden und 233 Millisekunden.

#### 8.1.2 Anwendung von Zeitzonen

Beide Datentypen, "dateTime" und "time", ermöglichen eine genaue Angabe der Zeitzone. Angaben zur Zeitzone, die von der UTC-Zeit (Universal Time Coordinated) abweichen, müssen innerhalb des GS1-Systems durch die Angabe der tatsächlichen Differenz der jeweiligen Landeszeit zur UTC zum Ausdruck gebracht werden. Direkt nach der Zeitangabe wird ein + oder - Zeichen angegeben, gefolgt von der Differenz zur UTC-Zeit. Die Angabe erfolgt nach dem Format hh:mm (Hinweis: die Angabe der Minuten ist zwingend erforderlich).

##### Beispiel 2

- **creationDate="2003-03-22T09:30:47.233-05:00"**  
Das Beispiel zeigt die Zeitangabe der Eastern Time (ET) Zeitzone, die 5 Stunden hinter der UTC-Zeit liegt.

### 8 Vordefinierte XML-Datentypen (built-in types)

---

- `creationDate="2004-11-06T12:43:17.000+09:00"`  
Das Beispiel zeigt die Zeit in Tokio, die der UTC-Zeit 9 Stunden voraus ist.

#### Beispiel 3

Nachfolgend ein zusammengesetztes Beispiel des XSD-Datentyps "dateTime", welches den oben genannten Richtlinien entspricht:

- `creationDate="2003-03-22T09:30:47.233-05:00"`

Dabei kennzeichnet:

- '2003' das Jahr
- '03' den Monat
- '22' den Tag
- 'T' das Trennzeichen für die Zeit
- '09' die Stunden
- '30' die Minuten
- '47' die Sekunden
- ',' das Trennzeichen für die Nachkommastellen der Sekunden
- '233' den Nachkommateil der Sekunden (in Millisekunden)
- '-' den Indikator zur Bestimmung des Abstands hinter der UTC-Zeitzone
- '05' die Stunden als Differenz der jeweiligen Landeszeit zur UTC-Zeit
- ':' das Trennzeichen für die Minuten und '00' steht für die Anzahl der Minuten

#### 9 Standard Business Document Header

Der in dem GS1 XML-Releases verwendete Standard Business Document Header (SBDH) ersetzt die in der Vergangenheit genutzte Envelope-Schicht einschließlich des AS2 Message-Headers. Er liefert Routing- und Verarbeitungsinformationen zu dem XML-Dokument (Instanz). Er ist so konzipiert, dass er unabhängig von dem verwendeten Transport-Protokoll genutzt werden kann. Mit Hilfe der im SBDH enthaltenen Informationen können Kommunikations-Anwendungen herausfinden, ob es sich für das beim Routing verwendete Protokoll um ebMS, AS2 oder ein anderes Protokoll handelt.

Wahlweise kann der SBDH auch Informationen zum Kontakt wie Name, Telefonnummer und E-Mail-Adresse enthalten. GS1-Schemas verwenden die SBDH's als integralen Bestandteil eines XML-Dokuments (in anderen Standards könnte der SBDH ein mit dem XML-Dokument verknüpftes Objekt sein).

Das SBDH-Schema enthält ein Element, welches die Versions-Nummer des Dokuments oder die enthaltenen Dokumente spezifiziert. Weitere Informationen zur Verwendung dieses Headers entnehmen Sie bitte dem Dokument: "Version numbers in the business documents and the Standard Business Document Header".

Die detaillierte UN/CEFACT-Spezifikation für den SBDH und den Leitfaden zur Verwendung, können Sie von der GS1-Webseite herunterladen.

#### 10 Nachrichten-Architektur

Die Architektur der GS1 XML-Nachrichten basiert auf getrennten Funktionsschichten, wobei jede Schicht in die andere eingebettet ist. Die drei Schichten sind:

- Die Transport- und Routing-Schicht enthält Informationen zum Routing und der Verarbeitung des XML-Dokuments (Instanz).
- Die Service-Schicht bietet der empfangenden Anwendung Anweisungen, die an dem Dokument durchgeführt werden sollen.
- Die Dokumenten-Schicht beinhaltet die eigentlichen Geschäftsdokumente oder Dokumente wie zum Beispiel Bestellungen, Lieferavise, Rechnungen etc.

Zusammen bilden alle drei Schichten eine GS1 XML-Nachricht.

Das nachfolgende Diagramm zeigt die Schichtenarchitektur für GS1-Nachrichten.

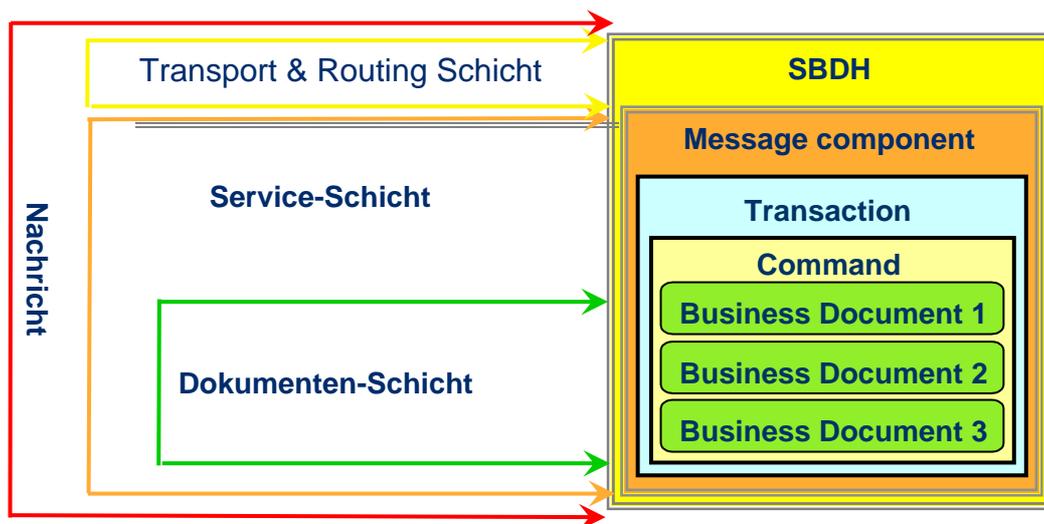


Abb. 6: Architektur-Komponenten für GS1-Nachrichten

Die Verwendung geschichteter Nachrichten bringt große Vorteile:

- Reduzierung der Gesamtanzahl der Nachrichten, die vom GS1-System benötigt werden, um einen effizienten Austausch der Geschäftsinformationen durchführen zu können. Das gleiche Dokument kann mit verschiedenen Anweisungen versehen werden. Daher sind keine zusätzlichen Dokumente, wie zum Beispiel "Bestellung hinzufügen", "Bestellung ändern" oder "Bestellung löschen", erforderlich. Die gleiche Bestellung könnte mit allen notwendigen Anweisungen, wie zum Beispiel, "hinzufügen", "ändern" oder "löschen" versendet werden.
- Mehrere Nachrichten können dabei die gleichen Befehle verwenden, sodass bei Bedarf neue Befehle schnell und einfach hinzugefügt werden können, ohne Änderung an den bestehenden Definitionen (XML-Schemas) vornehmen zu müssen.

Neue Anweisungen können hinzugefügt werden, indem weitere Befehle zur Command-Liste hinzugefügt werden. Neue Werte sind mit den Beteiligten bilateral abzustimmen. Alle übrigen Benutzer werden von dieser Änderung nicht beeinträchtigt.

Detaillierte Informationen zur Struktur der einzelnen Schichten sowie ihrer Verwendung werden gesondert im "GS1 XML Message Architecture Guide" behandelt. Der Leitfaden ist auf der GS1-Webseite erhältlich.

#### 11 Nachrichten-Schicht

Die Dokumenten-Schicht enthält die eigentlichen Geschäftsnachrichten. Diese Schicht wird im GS1 BMS-Dokument vollständig beschrieben.

##### 11.1 Proxy-Dateien

Funktionell ist eine Geschäftsnachricht ein Teil der Nachrichten-Architektur, die das Vorhandensein anderer Ebenen - den Standard Business Document Header und die Komponenten der Service-Schicht - erfordert.

Nicht alle Parser unterstützen es, wenn innerhalb einer XML-Instanz die Schemas an unterschiedlichen Stellen hinterlegt sind. Um den Prozess zur Einbindung gültiger XML-Nachrichten in die gesamte Architektur zu erleichtern, werden für jede Geschäftsnachricht Proxy-Schemas erstellt, mit denen die erforderlichen Dateien eingebunden und importiert werden. Proxy-Schemas sind nicht normativ und dienen ausschließlich zur Implementierung der GS1-Standards.

Sofern in XML-Dokumenten das Attribut "xsi:schemaLocation" verwendet wird, empfiehlt GS1 als Best Practice, dass das Attribut auf die aktuelle Schema-Datei verweisen sollte, in der das Root-Element deklariert wurde. Proxy-Dateien sind nicht Teil des GS1-Standards und sollten daher möglichst vermieden werden.

Ein Beispiel für ein Proxy-Schema am Beispiel einer Rechnungsbestätigung:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetName-  
space="urn:ean.ucc:2">  
<xsd:import namespace="urn:ean.ucc:pay:2"  
schemaLocation="ean.ucc/pay/InvoiceResponse.xsd"/>  
<xsd:include schemaLocation="ean.ucc/common/AdditionalPartyIdentificationList.xsd"/>  
<xsd:include schemaLocation="ean.ucc/common/Command.xsd"/>  
<xsd:include schemaLocation="ean.ucc/common/Document.xsd"/>  
<xsd:include schemaLocation="ean.ucc/common/DocumentCommand.xsd"/>  
<xsd:include schemaLocation="ean.ucc/common/DocumentCommandList.xsd"/>  
<xsd:include schemaLocation="ean.ucc/common/DocumentReference.xsd"/>  
<xsd:include schemaLocation="ean.ucc/common/DocumentStatusList.xsd"/>  
<xsd:include schemaLocation="ean.ucc/common/EntityIdentification.xsd"/>  
<xsd:include schemaLocation="ean.ucc/common/Extension.xsd"/>  
<xsd:include schemaLocation="ean.ucc/common/Message.xsd"/>  
<xsd:include schemaLocation="ean.ucc/common/PartyIdentification.xsd"/>
```

```
<xsd:include schemaLocation="ean.ucc/common/ResponseStatusList.xsd"/>
<xsd:include schemaLocation="ean.ucc/common/Transaction.xsd"/>
</xsd:schema>
```

In diesem XML-Dokument lassen sich die zahlreichen Dateinamen durch den Namen eines einfachen Proxy-Schemas ersetzen. Sofern diese Datei das Attribut "xsi:schemaLocation" verwendet, ist der Parser in der Lage, dieses XML-Dokument zu validieren. Nachfolgend ein Beispiel für eine XML-Instanz, die auf ein Proxy-Schema weist:

```
<?xml version="1.0" encoding="UTF-8"?>
<sh:StandardBusinessDocument
xmlns:sh="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
xmlns:eanucc="urn:ean.ucc:2" xmlns:pay="urn:ean.ucc:pay:2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader
../Schemas/sbdh/StandardBusinessDocumentHeader.xsd urn:ean.ucc:2
../Schemas/InvoiceResponseProxy.xsd">
```

#### 11.2 Geschäftsnachrichten

Die Geschäftsnachrichten werden von ihren jeweiligen Geschäftsprozessen organisiert. Jede Geschäftsnachricht benötigt alle eingebundenen und importierten Dateien, die für eine erfolgreiche Validierung erforderlich sind. Jedes Anwendungs-Paket (Implementers Packet) enthält für jede Nachricht eine Textdatei mit dem Namen "Table of Content". Diese Datei listet alle benötigten Dateien auf, die die für die Validierung des Schemas einer bestimmten Geschäftsnachricht notwendig sind.

#### 11.3 Common Library

In jeder Geschäftsnachricht werden Komponenten aus der Common Library verwendet. Sie enthält alle Dateien, die in mehr als einer Geschäftsnachricht und in mehreren Kontexten verwendet wird.

Die Komponenten, die in nur einem Kontext verwendet werden, sind im Namensraum dieses bestimmten Kontextes definiert. Komponenten, die nur in einer Nachricht benötigt werden, sind innerhalb des Nachrichten-Schemas lokal definiert.

Dieser Ansatz ermöglicht in allen Geschäftsnachrichten die Wiederverwendung gleicher Informationsstrukturen, wodurch die Interoperabilität erhöht und die Pflege der Schemas vereinfacht wird.

In früheren Versionen des GS1 XML-Standards (1.1, 1.3 und 1.3.1) hatten die Komponenten der Common Library keinen eigenen Namensraum. Sie haben den Namenraum der Schema-Dokumente übernommen, in denen sie verwendet wurden. Dies hat zur Folge, dass eine Änderung in der Common Library sich auf die Bibliothek und auch auf die Geschäftsnachrichten auswirkt.

Common Library Schemas folgen der gleichen Versionierungsstrategie, wie auch alle anderen Schemas. Ab der Version 2.0 haben die Dateien der Common Library ihren eigenen Ziel-Namensraum: `xmlns: eanucc = "urn: EAN.UCC: 2"`.

Dieser Namensraum enthält nur die Hauptversion. Die Nebenversion wird nicht aufgeführt und ist abwärtskompatibel. Zur Anzeige der Nebenversionsnummer wird, wie in anderen Schemas, das Versions-Attribut im `xsd: schema-Element` verwendet.

Schemas der Common Library besitzen eigene Namensräume und sind von den Geschäftsnachrichten getrennt. Das hat den Vorteil, dass sie sich separat versionieren lassen. Dies bedeutet, dass eine Änderung an einer beliebigen Komponente sich nur auf die Version der Dokumente und nicht auf das gesamte Release auswirkt.

Sollte jedoch die Hauptversionsnummer der Common Library geändert werden, müssen alle Geschäftsnachrichten, die diese Bibliothek nutzen, ebenfalls auf diese neue Version wechseln.

## 12 Extension-Mechanismus

### 12.1 Allgemeine Komponenten einer Extension

Ab der Version 2.0 enthält jedes Nachrichtenschema einen Platzhalter, über den zusätzlich kontext-spezifische Komponenten an die Nachricht angefügt werden können. Dieser Platzhalter, XML-Schema-Element "##any" genannt, hat die folgende Form:

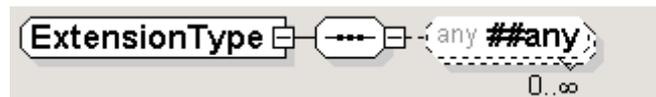


Abb. 7: Die Struktur der Extension-Komponente

Das XML "any"-Element enthält das vordefinierte (built-in) Attribut: "processContents". Es dient als Indikator und zeigt Anwendungen oder einem XML-Prozessor wie die Validierung abzuwickeln ist. Berücksichtigt werden alle im "any"-Element definierten Elemente. In der GS1 Extension-Komponente wird der Wert dieses Attributes auf den Wert "lax" gesetzt. Dies bedeutet, dass der XML-Prozessor versucht, das Schema für den erforderlichen Namensraum zu beschaffen und jedes beliebige Element des Namensraumes zu validieren. Kann das Schema nicht beschafft werden, tritt kein Fehler auf.

### 12.2 Validierung der Komponenten einer Extension

Obwohl die XML-Syntax Regeln für das "any"-Element die Platzierung eines beliebigen wohlgeformten XMLs erlauben, sollten Anwender daran denken, dass Geschäftspartner möglicherweise nicht in der Lage sind, diese nicht-standardisierten Informationen zu überprüfen. Dem Empfänger der Nachricht sollte zur Validierung der Extension das entsprechende Schema bereitgestellt werden. Es wird empfohlen, nur die innerhalb des GSMP-Prozesses freigegebenen Attribute für die Extension zu verwenden. Andernfalls entsprechen die Nachrichten keinem Standard und sind damit nicht weltweit miteinander kompatibel.

Durch die Vorbelegung des "processContents"-Attribut mit dem Wert "lax", wird die Validierung der Extension-Elemente optional: Bei der Validierung von XML-Schemas wird ein Parser nur Elemente validieren, für die er Definitionen (Extension-Schemas) findet und sofern sie ungültig sind, entsprechende Fehler melden. Findet der Parser für bestimmte kontext-spezifische Elemente keine kontext-spezifischen Schemas, wird er für diese Elemente keine Fehler ausweisen. Der Platzhalter ist ein optionales Element des Root-Elements.

#### 13 XML Werkzeuge

Auf dem Markt ist eine große Auswahl an XML-Werkzeugen erhältlich: von einfachen, webbasierten und frei zugänglichen Validierern bis hin zu integrierten Entwicklungsumgebungen, die Validierer, Parser, Dokumentationsgenerator, Repository für Schnittstellen, Skriptsprachen, Konvertierung von XML-Nachrichten in andere Formate und viele andere Funktionen in einem Paket miteinander kombinieren.

Die komplexen Produkte neigen generell dazu, kostenintensiver zu sein. Der Benutzer benötigt in der Regel nicht alle der oben genannten Features oder er weiß zunächst noch nicht genau, welche Funktionen sinnvoll sein könnten und für die es sich lohnt Geld zu bezahlen. Für die XML-Verarbeitung gibt es eine breite Palette an kostenlosen Software-Tools, die beim Aufbau des Verständnisses für den XML-Standard und dessen Verwendung hilfreich sein könnten. Eine recht umfangreiche Liste finden Sie unter den folgenden Links:

<http://www.garshol.priv.no/download/xmltools/> oder die vom W3C gepflegte Liste:  
<http://www.w3.org/XML/Schema#Tools>

Für die Suche im Internet nach verfügbarer XML-Software, könnten die folgenden Schlüsselwörter hilfreich sein:

- "XML tools"
- "XML parser"
- "XML validator"
- "XML integrated development environment"
- "XSLT editor"
- "XSLT generator"
- "XSLT engine"

Die Checkliste für Solution Provider ist hilfreich für Unternehmen, die eine Entscheidung treffen wollen, welche Werkzeuge sie für die Umsetzung der GS1-Standards einsetzen möchten.

Die GS1-Schemas und Beispieldateien wurden mit der letzten Version der nachfolgenden Validierungs-Tools getestet:

- XML Schema Validator (XSV)
- XMLSpy (Version 2006 oder höher) von der Firma Altova
- Xerces (Version 2.7 oder höher) von der Firma Apache

#### 14 Checkliste für die Auswahl von Solution Providern

Für die Umsetzung der GS1 XML-Nachrichten sollten Anwender vor dem Kauf eines Produktes darauf achten, dass der Standard ausreichend unterstützt wird.

Die nachfolgenden Fragen sollen Ihnen eine Hilfestellung für die geeignete Auswahl eines Produktes bieten:

1. Unterstützt Ihr Produkt GS1 XML-Schemas?
2. Welche Version(en) der GS1 XML-Schemas unterstützt Ihr Produkt?
3. Wie schnell unterstützt Ihr Produkt eine neue Version des von GS1 veröffentlichten Standards?
4. Kann Ihr Produkt mehrere Versionen des GS1-Standards verarbeiten?
5. Nimmt Ihr Unternehmen am globalen GS1-Entwicklungsprozess (Global Standards Management Process – GSMP) teil? Wenn ja, in welcher Funktion?
6. Wie importiert Ihr System die GS1-Schemas?
7. Lassen sich mit Ihrem Produkt XML-Dokumente auf Basis der importierten Schemas generieren?
8. Welche verschiedenen Möglichkeiten bietet Ihr Produkt zur Abbildung von GS1-Schemas und XML-Dokumenten?
9. Unterstützt Ihr Produkt die Bearbeitung von GS1 XML-Nachrichten, die Validierung von Nachrichten oder beide – Bearbeitung und Validierung?
10. Wie stellt Ihr Produkt sicher, dass die zu verarbeitenden Daten der XML-Nachrichten dem Standard entsprechen?
11. Wie effizient ist eine Validierung gegen GS1-Schemas?
12. Lassen sich Validierungen vonseiten der Handelspartner und / oder der Nachrichten aktivieren / deaktivieren? Wenn ja, wie?
13. Wie leicht kann Ihr Produkt GS1-Nachrichten in andere Formate konvertieren (z. B. anwenderdefinierte Formate oder andere XML Formate)? Bitte erklären Sie, wie Ihr Konvertierungs-Tool arbeitet.
14. Haben Sie Kunden, die bereits Ihr Produkt zum Austausch und zur Übersetzung von GS1-Nachrichten nutzen? Wenn ja, wie viele und mit welchen Nachrichtenarten?
15. Sofern Sie Kunden haben, die Ihr Produkt in Verbindung mit GS1-Nachrichten nutzen, stellen Sie bitte eine Kunden-Referenzliste mit Kontaktdaten zur Verfügung.

### 14 Checkliste für die Auswahl von Solution Providern

---

16. Bitte Beschreiben Sie, wie Sie Ihren Support organisiert haben. Besitzen Ihre Mitarbeiter ausreichende Kenntnisse über GS1-Nachrichten?
17. Ermöglicht Ihr Produkt eine dynamische XML-Validierung, sofern sich die GS1-Schemas außerhalb der Firewall befinden?
18. Unterstützt Ihr Produkt die W3C-Empfehlungen für XML-Schemas?
19. Unterstützt Ihr Produkt den Befehl `<<include>>` für Schemas mit mehreren Speicherstellen und den Befehl `<<import>>` für mehrere Namensräume?

### 15 Anhang: Wesentliche Änderungen zwischen dem Release 1.3.1 und 2.0

---

#### **15 Anhang: Wesentliche Änderungen zwischen dem Release 1.3.1 und 2.0**

1. Anwendung der Namensräume
  - a) Einzelne Namensräume für unterschiedliche Kontexte
  - b) Hauptversionen, die sich in den Namensräumen widerspiegeln
2. Neue Versionierungsstrategie
  - a) Rückwärtskompatible Änderungen, die sich in Nebenversionen widerspiegeln
  - b) Nicht kompatible Änderungen, die sich in den Hauptänderungen widerspiegeln
3. Envelope-Schicht, wurde ersetzt durch den Standard Business Dokument Header
4. Neue Code-Listen Verwaltung
  - a) Jede Code-Liste ist in einem einzelnen Schema definiert
  - b) Jede Code-Liste wird einzeln versioniert
  - c) Das Hinzufügen neuer Codes erfordert kein Upgrade der gesamten Nachrichten-Suite
5. Neuer Extension-Mechanismus
  - a) In allen Schemas werden neue Extensions eingeführt
  - b) Xsi:type wird nicht mehr unterstützt
6. Neue Verwaltung der Common Library – jedes common component schema ist von weiteren Release-Änderungen unabhängig
7. GTIN und GLN werden zwingend als primäre Identifikation für Produkte und Partner verwendet

### Impressum

Herausgeber:  
GS1 Germany GmbH, Köln

Geschäftsführer:  
Jörg Pretzel

Text (Übersetzung):  
Tanja Thomsen

Redaktion:  
Elisabeth Kikidis

GS1 Germany GmbH  
Maarweg 133 · D-50825 Köln  
Postfach 30 02 51 · D-50772 Köln  
Telefon (02 21) 9 47 14-0  
Telefax (02 21) 9 47 14-990  
eMail: [info@gs1-germany.de](mailto:info@gs1-germany.de)  
[www.gs1-germany.de](http://www.gs1-germany.de)

© GS1 Germany GmbH, Köln, 2010  
GTIN 40 00001 01883 3