



# **Codier-Richtlinien**

**XDOMEA 2.0**

**Erstellt durch: AG XDOMEA**

**Erstellt am: 12.11.2008**

# Inhaltsverzeichnis

|    |   |   |
|----|---|---|
| 0  | Einleitung .....  | 3 |
| 1  | Codierung der XML-Schemata .....                        | 4 |
| 2  | Globale Elemente und globale Typen .....                | 4 |
| 3  | Kennzeichnung globaler Schematypen .....                | 4 |
| 4  | Kennzeichnung globaler Schematypen für Codelisten ..... | 4 |
| 5  | Versionierung .....                                     | 4 |
| 6  | Dateinamen der XML-Schemata .....                       | 4 |
| 7  | Schreibweise von XML-Namen .....                        | 5 |
| 8  | Sprechende XML-Namen .....                              | 5 |
| 9  | Zusammengesetzte XML-Namen .....                        | 5 |
| 10 | Inhalte von XML-Elementen .....                         | 5 |

## 0 Einleitung

Nachfolgend werden die wichtigsten formalen Regeln aufgeführt, die bei der Erstellung der XDOMEA-2-XML-Schemata zu Grunde gelegt worden sind. Die Aufzählung dient als Richtlinie für künftige Weiterentwicklungen von XDOMEA.

## 1 Codierung der XML-Schemata

Die für XDOMEA eingesetzten XML-Schemata werden in der Codierung UTF-8 erstellt.

## 2 Globale Elemente und globale Typen

Nachrichten werden als globale Elemente definiert – alle anderen Elemente werden lokal, unter den entsprechenden komplexen Typen definiert. Die Definition von komplexen Typen erfolgt grundsätzlich in globaler Form (ausgenommen „choice“), um eine Wiederverwendung zu vereinfachen.

## 3 Kennzeichnung globaler Schematypen

Globale Datentypen werden mit dem Suffix „Type“ gekennzeichnet. Hierbei spielt es keine Rolle, ob es sich um einen komplexen oder einfachen Typen handelt.

## 4 Kennzeichnung globaler Schematypen für Codelisten

Einfache Typen, für die eine Werteliste vorgesehen ist, erhalten einen besonderen Suffix: Typen, die aus einer Aufzählung einzelner Enumerations-Werte und zusätzlichen Attributen bestehen, erhalten den Suffix „CodeContentType“.

Komplexe Typen, die als Basis eine Werteliste besitzen, erhalten den Suffix „CodeType“.

## 5 Versionierung

Jedes Typen-Schema besitzt ein Attribut „version“ und komplexe Typen zur Darstellung einer Werteliste enthalten ein Attribut „codelistVersionIdentifier“.

## 6 Dateinamen der XML-Schemata

Folgende Dateinamen existieren:

- Baukasten.xsd
- Datentypen.xsd
- Codelisten.xsd
- Nachrichten\_\*\*\*.xsd (\*\*\*) = Systemname, z.B. VBS, DMS, Archivsystem, Fachverfahren)

Eine Angabe der Version in dem Dateinamen ist nicht notwendig, da bereits im targetNamespace die Versionsnummer enthalten ist.

## 7 Schreibweise von XML-Namen

Die Schreibweise der XML-Namen für Elemente, Attribute und Typen entspricht der deutschen Rechtschreibung. Ausnahmen bilden Sonderzeichen (Umlaute, ß, etc.).

Die Namen von Elementen und Typen fangen grundsätzlich mit einem Groß- und Attribute mit einem Kleinbuchstaben an. Ausnahme bilden hier Typen aus der Datei „Datentypen.xsd“, die auf einem W3C-Datentyp basieren – sie erhalten den gleichen Namen wie der W3C-Typ sowie eine Ergänzung, z.B. „stringUUID“.

## 8 Sprechende XML-Namen

Für sämtliche Bezeichner (Elemente, Attribute, Typen) werden aussagekräftige Bezeichnungen verwendet, die erkennen lassen, welchen Inhalt bzw. welchen Zweck das Objekt hat. Um Bezeichner jedoch nicht zu lang werden zu lassen, ist es in Einzelfällen auch gestattet, Abkürzungen zu verwenden, die dann in der Spezifikation beschrieben werden bzw. deren Sinn sich aus der Beschreibung des zu bezeichnenden Objekts erkennen lässt, z.B. SGO, Nk.

## 9 Zusammengesetzte XML-Namen

Sofern die verwendeten Namen aus mehreren Teilworten bestehen, werden diese durch einen Großbuchstaben beim nächsten Wort voneinander getrennt (z.B. FremdesGeschaeftszeichen).

Worte, die zwei Substantive durch ein Genitiv-„s“ verbinden, werden als ein einziges Wort behandelt.

## 10 Inhalte von XML-Elementen

Die Wertehalte der XML-Elemente dürfen Sonderzeichen enthalten.