



Arbeitsgruppe „IT-gestützte Vorgangsbearbeitung“

XDOMEA 1.0

Technische Beschreibung

Versionsangaben

Version	1.0
Stand vom	31.05.2005
Bearbeitet von	I. Bauer
Status	Endfassung

Änderungshistorie

Datum	Änderung	durch
01.02.2005	Neuerstellung	Irina Bauer
25.05.2005	Überarbeitung	Irina Bauer
31.05.2005	Redaktionelle Überarbeitung	Rolf-Dieter Wenzel

INHALTSVERZEICHNIS

1	Zielsetzung	3
2	Lösungsansatz.....	4
3	Genereller Aufbau von XDOMEA	4
3.1	Komponenten	4
3.2	Aufbau der XML-Schemadateien.....	5
3.2.1	Schema: Typen	6
3.2.2	Schema: Root.....	7
3.2.3	Schema: Werteliste	7
3.3	Namensraum	8
4	Globale Typen	10
4.1	T_XDOMEA.....	10
4.2	T_Identifizierung.....	12
4.3	T_Verweis.....	14
4.4	T_Adresse	14
5	Lokale Elementbeschreibungen.....	16
5.1	Zum Dokument.....	16
5.1.1	Beschreibung	17
5.1.2	Primärdokument	17
5.1.3	Adress_Informationen	18
5.2	Zum Vorgang.....	18
5.2.1	Beschreibung	19
5.2.2	Adress_Informationen	19
5.3	Zur Akte	20
5.3.1	Beschreibung	20
5.3.2	Adress_Informationen	21
6	Identitätsbeschränkungen (Identity Constraints).....	22
6.1	Schlüssel & Referenzen	22
6.2	Eindeutigkeit.....	24
7	Instanziierung	25
7.1	Wertelisten.....	26
8	Version	28
8.1	Bestandteile der Versionsnummer.....	28
8.2	Angabe der Versionsnummer im Dateinamen.....	29
9	Abbildungsverzeichnis.....	30

1 Zielsetzung

Der XML-Standard XDOMEA wurde im Auftrag des KoopA ADV durch die Arbeitsgruppe „IT-gestützte Vorgangsbearbeitung des KoopA ADV“ und Vertretern des Unterausschusses „Allgemeine Verwaltungsorganisation“ des AK VI (Verwaltungsorganisation, Aus- und Fortbildung sowie öffentliches Dienstrecht) der IMK (Ständige Konferenz der Innenminister und -senatoren der Länder) entwickelt. Er dient zum elektronischen Austausch von Akten, Vorgängen und Dokumenten.

Der Standard besteht aus einer Beschreibung der Metadatenfelder für den Austausch von folgenden Komponenten

- XDOMEA –Vorgang
- XDOMEA – Akte
- XDOMEA – Dokument
- XDOMEA – Adresse

Zum elektronischen Austausch von Daten werden die jeweils erforderlichen Komponenten nach Bedarf miteinander kombiniert, die gleichzeitige Verwendung aller Komponenten ist möglich, aber nicht erforderlich.

Um die Komponenten möglichst vielseitig nutzen zu können, sind sie durch viele optionale Felder sehr flexibel gehalten.

Partner, die Daten austauschen wollen, können die Menge der verwendeten Felder dadurch selbst festlegen. Damit wird ein Austausch auch dann möglich, wenn unterschiedlich parametrisierte Vorgangsbearbeitungssysteme verwendet werden. Gleichzeitig kann die Weitergabe von den Informationen verhindert werden, die aus Datenschutzgründen nicht weitergegeben werden sollen. Die absendende Behörde kann somit festlegen, welche Informationen dem Empfänger mit der Übertragung zur Verfügung gestellt werden sollen. So kann auch die Anforderung nach einer ggf. erforderlichen Beschränkung der übertragenen Metadaten erfüllt werden (z.B. auf Grund datenschutzrechtlicher oder sonstiger gesetzlicher Bestimmungen).

2 Lösungsansatz

Die Datenbeschreibung zu XDOMEA bedient sich der Beschreibungssprache XML (Extensible Markup Language). Daten, die im Format XDOMEA übermittelt werden, sind in einer XML-Datei (einem so genannten Instanzdokument) enthalten, die nach bestimmten Regeln aufgebaut ist. Diese Regeln ihrerseits sind in Dateien hinterlegt, die nach dem Standard XSD (XML-Schema-Definition) aufgebaut sind. Die Definition der XDOMEA-Datenstruktur ist in einer XML-Schema-Definitionsdatei hinterlegt. In ihr werden grundlegende Strukturen und Datentypen definiert und somit festgelegt, welche Daten ausgetauscht werden können.

XML-Schema	http://www.w3.org/2001/XMLSchema
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
XML Version 1.0 und Version 1.1	http://www.w3.org/TR/REC-xml 1.0

XML Spezifikationen

3 Genereller Aufbau von XDOMEA

Grundsätzlich sind die Schemata des Datensatzes XDOMEA so gestaltet, dass dieser mit geringem Aufwand fortgeschrieben und erweitert werden kann.

Die XML-Schemadateien enthalten Vorgaben für den formalen Aufbau der zu übermittelnden Dateien. Die übermittelten Daten bestehen aus einer Datei im Format XML (Extensible Markup Language) und in der Regel aus Dokumenten, auf die in der XML-Datei Bezug genommen wird (Primärdokumente). Der formale Aufbau der XML-Datei muss den im XML-Schema definierten Vorgaben entsprechen.

Ausgangspunkt für die Prüfung, ob eine XML-Datei dem Standard XDOMEA entspricht, ist entweder das Rootschema oder ein anderes Schema, das den XDOMEA-Typ verwendet.

3.1 Komponenten

Der Datensatz XDOMEA umfasst drei XML-Schemadateien:

? Typen-Schema: `xdomea_typen_1_0.xsd`:

In dieser Datei sind globale komplexe Datentypen und grundlegende Strukturen definiert, die alleine nicht instanziiert werden können. Hierzu ist folgendes Root-Schema notwendig: `xdomea_root_1_0.xsd`.

? Root-Schema: xdomea_root_1_0.xsd:

Hier findet eine Instanziierung der erwähnten Typen-Schemadatei statt. Sie enthält ein Root-Element, das mit dem Typ T_XDOMEA aus der Datei xdomea_typen_1_0.xsd typisiert ist.

? Wertelisten-Schema: xdomea_wl_allgemein_1_0.xsd:

Dieses Schema ist eine Ansammlung von Aufzählungstypen. Hier werden mögliche Werte zum Ausfüllen einzelner Elemente der Typdefinitionen festgelegt.

3.2 Aufbau der XML-Schemadateien

XDOMEA umfasst drei voneinander abhängige Schemata. Die Elementbeschreibungen, die in XDOMEA relevant sind, wurden auf diese Schemadateien verteilt. Diese drei Schemata lauten wie folgt:

- xdomea_typen_1_0.xsd
- xdomea_root_1_0.xsd
- xdomea_wl_allgemein_1_0.xsd

xdomea_typen ist eine reine Ansammlung von globalen komplexen Datentypen. Die Verwendung dieser Typen ist stets an ein Element gebunden.

Dies ist üblicherweise das Root-Element von XDOMEA, das in der Schemadatei Root_1_0 deklariert wird. Die Trennung der Typdefinitionen von dem Root-Element ist notwendig und eröffnet die Möglichkeit, dass auch andere XDOMEA-fremde Bereiche einfach die Typdefinitionen von XDOMEA in ihren jeweiligen Anwendungsgebieten verwenden können, um Daten auszutauschen. Ein solcher externer Bereich ist beispielsweise das Justizwesen. Dadurch kann ein Element aus einem externen Bereich die Typdefinitionen von XDOMEA verwenden. Der Normalfall jedoch ist bis auf weiteres, dass die Typdefinitionen von XDOMEA über das Root-Element der Schemadatei xdomea_root eingesetzt werden.

In den Typdefinitionen sind Elemente definiert, deren Werte vorgegeben sind. Diese Werte sind in der Schemadatei der Werteliste xdomea_wl_allgemein zu finden.

Auf den Aufbau und den Inhalt der drei genannten Schemadateien wird im Folgenden näher eingegangen.

3.2.1 Schema: Typen

Die bewusste Trennung und Auslagerung der Definition der Datentypen und der Deklaration des Root-Elements ermöglicht eine einfache Wiederverwendung der Datentypen und eine bequeme Modularisierung. Man bedient sich gängiger Auslagerungstechniken, wie sie auch in verschiedenen anderen Programmiersprachen erhältlich sind, um die Pflege von größeren Schema-Dokumenten und umfangreichen Datenstrukturen zu ermöglichen. Dadurch können größere Datenstrukturen in kleinere Einheiten zerlegt und diese den Ansprüchen gerecht modular zusammengesetzt werden. Gemäß der Modularisierungstechnik entstanden die Schemadateien `xdomea_typen_1_0` und `xdomea_root_1_0`.

Das Typen-Schema (`xdomea_typen_1_0.xsd`) ist somit eine Art Modul, das als Baukasten gesehen werden kann. Die Bestandteile des Baukastens sind globale komplexe Typdefinitionen. Globale komplexe Typen eignen sich dazu, wieder verwendbare Komponenten in Form von benannten Bereichen durch qualifizierte Namensbezeichnungen zu erstellen, die später innerhalb der Definitionen anderer Elemente in ihrem Type-Attribut aufgerufen werden. Dadurch überträgt man die im globalen komplexen Typ definierten Eigenschaften in das deklarierte Element und verzichtet dabei auf eine lokale Definition der jeweiligen Kind-Elemente.

Es ist anzumerken, dass alle in XDOMEA enthaltenen Elemente optional sind, d.h. ihre Kardinalität ist auf *minOccurs=0* gesetzt, so sind sie nicht obligatorisch erforderlich. In ihren Häufigkeiten unterscheiden sie sich und sind dementsprechend einzeln definiert.

Das Typen-Schema besitzt kein Wurzel(Root-)element und kann deshalb nicht als Ausgangspunkt für die Erstellung oder Prüfung eines XML-Instanzdokuments verwendet werden. Für diesen Zweck ist das Root-Schema oder ein entsprechendes Schema aus einem anderen externen Fachbereich zu verwenden.

3.2.2 Schema: Root

Das Root-Schema (xdomea_root_1_0.xsd) enthält nur ein Element, welches so angelegt ist, dass es in einem XML-Instanzdokument das Wurzel(Root-)element darstellt. Der Bezeichner für dieses Element lautet <XDOMEA>.

In dieses Schema werden die Typdefinitionen von XDOMEA inkludiert. Die Inklusion ermöglicht die Verwendung der Typdefinitionen der inkludierten XDOMEA-Typen-Datei, so dass nun dem Element <XDOMEA> durch die Typisierung mit *T_XDOMEA* die komplette Struktur von XDOMEA zugewiesen wird.

```
<xs:include schemaLocation="xdomea_typen_1_0.xsd" />
<xs:element name="XDOMEA" type="T_XDOMEA" />
```

Das Root-Schema enthält ansonsten keine weiteren Elemente.

3.2.3 Schema: Werteliste

Hierunter versteht man die Aufzählung möglicher Werte, die den Wertebereich bereits definierter Elemente einschränken. In der Werteliste von XDOMEA stehen komplexe Typen, die jeweils Aufzählungen von Werten enthalten. In XDOMEA werden diese quasi Aufzählungstypen mit einem vorangestellten „WL_“ im Bezeichner gekennzeichnet, so dass sie sofort als Wertelistentyp erkannt werden. Es existieren folgende Wertelistentypen:

- WL_String
- WL_Anschriftstyp
- WL_Dokumenttyp
- WL_Kommunikationsart
- WL_Medium
- WL_Referenzart
- WL_Telekennzeichnung
- WL_Vertraulichkeitsstufe

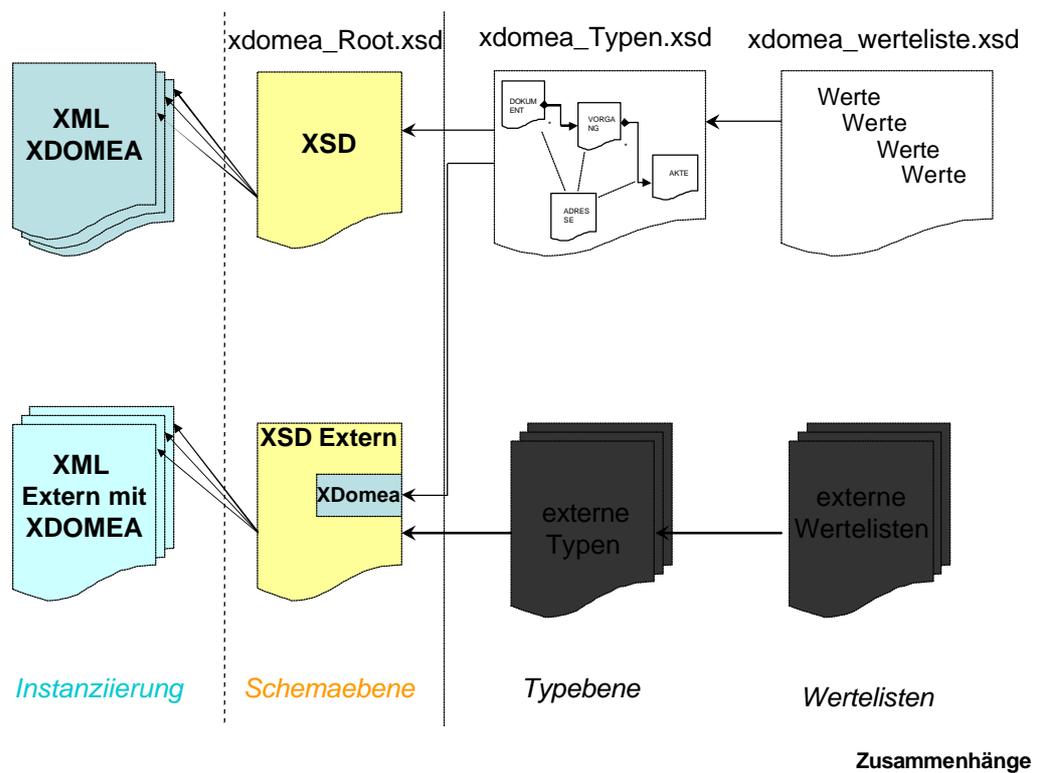
Es besteht die Möglichkeit ein Element entweder mit einem Stringwert zu belegen oder mit einem Wert aus einer Werteliste. Entweder wird ein Wertelistentyp oder der *WL_String-Typ*, der ebenfalls in der Werteliste definiert wird, verwendet.

Diese Typen besitzen dieselbe Struktur: die Attribute <version> und <fassung>, über die die Wertelisten versioniert werden können.

Zur Verwendung der vordefinierten festgelegten Werte muss ein Bezug zwischen den Typdefinitionen und der Werteliste vorhanden sein. Die Inklusion geschieht in der Schemadatei `xdomea_typen_1_0.xsd` mit folgendem Befehl:

```
<xs:include schemaLocation="xdomea_wl_allgemein_1_0.xsd" />
```

Ein Überblick der Zusammenhänge zwischen Root-Schema, Typen-Schema und Werteliste zeigt folgendes Schaubild:



3.3 Namensraum

Für die Einrichtung eines Namensraumes benötigt man nur das `xmlns`-Attribut, dessen Wert eine URI-Angabe enthält. Ist zusätzlich noch ein Präfix angegeben, kann dieses als Kurzform für den Namensraum stehen. Das Präfix besteht aus einer kurzen Bezeichnung, die jedem Element vorangestellt werden kann. Üblicherweise steht der Namensraum im Wurzelement und gilt somit für alle Elemente im XML-Instanzdokument.

Der Standard-Namensraum, der in dem XML-Instanzdokument mit dem `xmlns`-Attribut angegeben wird, gilt überall und ist insbesondere ohne weitere Angaben ei-

nes Präfixes gültig. Das heißt, dass alle Bezeichner der Elemente innerhalb eines XML-Instanzdokuments dem gesetzten Standard-Namensraum zugeordnet werden und somit eindeutig identifizierbar sind. Diesem Namensraum ist also über das Schema-Attribut `<xmlns: >` kein besonderes Präfix zugeordnet. In XDOMEA lautet der Standard-Namensraum <http://www.xdomea.de>.

Darüber hinaus kann ein Präfix gesetzt werden. Wird ein Präfix dem Bezeichner vorangestellt, ist dieses Element dem bestimmten Namensraum des definierten Präfixes zugeordnet.

Beispielsweise ist das Vokabular der Definitionssprache XML Schema selbst durch den Namensraum „www.w3.org.2002/XMLSchema“ identifiziert. Auch um Schreibaufwand zu sparen und den Code übersichtlicher zu halten, wird für diesen Namensraum im Allgemeinen das Präfix `xs:` eingesetzt.

Globale Objekte, denen kein Präfix vorangestellt ist, werden automatisch dem Standardnamensraum (`xmlns:http://www.xdomea.de`) zugeordnet. Lokal deklarierte Elemente werden ihm nur dann zugeordnet, wenn das Attribut `<elementFormDefault>` den Wert *qualified* hat. Dies ist in XDOMEA der Fall. Lokal deklarierte Attribute werden dem Standard-Namensraum zugeordnet, wenn das `<attributeFormDefault>` den Wert *qualified* hat. Dies ist in XDOMEA nicht der Fall.

Schließlich kann noch ein Ziel-Namensraum (`targetNamespace`) definiert werden. Dies ist der Standard-Namensraum für die zum Schema gehörigen Instanzdokumente. In XDOMEA lautet auch der Ziel-Namensraum <http://www.xdomea.de>. Damit sind alle zu XDOMEA gehörigen Objekte dem einheitlichen Namensraum zugeordnet. Dies vereinfacht die Erweiterung durch Vererbung von Typen und die Anbindung von externen Listen.

Zusätzlich zu dem definierten Standard-Namensraum <http://www.xdomea.de> ohne Präfix wird der Standard-Namensraum mit einem Präfix „xdo:“ angelegt. Dies eröffnet die Möglichkeit, dass andere Bereiche (z.B. Justizwesen) die Typdefinitionen von XDOMEA nutzen können. Damit sind automatisch die beschriebenen Elemente von XDOMEA in einem anderen externen Bereich dennoch an den für XDOMEA zugehörigen Standard-Namensraum `http://www.xdomea.de` gebunden und können so von anderen extern beschriebenen Elementen unterschieden werden.

4 Globale Typen

Innerhalb eines XML-Schemadokumentes lassen sich Typen sowie Elemente sowohl global also auch lokal definieren.

Globale Komponenten eignen sich dazu, wieder verwendbare benannte Bereiche zu erstellen, die später innerhalb der Definition anderer Elemente in ihrem <type>-Attribut aufgerufen werden. Dies bildet sozusagen eine Schnittstelle zu den Datentypen. Globale Komponenten lassen sich jeweils in anderen Dokumenten über diverse Techniken der Wiederverwendbarkeit abrufen und auch in ihren Eigenschaften mit Hilfe von Import, Redefine oder Inklusion bzw. durch Ableitung mit Einschränkung oder Erweiterung verwenden. Um ein Schema also erweiterbar zu machen, müssen entsprechende Strukturen erstens global in einem einzubindenden Dokument vorliegen und dürfen zweitens nicht durch die Techniken der Ableitungskontrolle in ihrer Ableitungsfähigkeit beschränkt sein.

Betrachtet man nun eingehend die Schemadatei `xdomea_typen_1_0.xsd`, befindet sich in ihr der Grundtyp `T_XDOMEA`, der die Hauptstruktur von XDOMEA darstellt.

4.1 `T_XDOMEA`

Dieser Typ ist gemäß der inhaltlich fachlichen Beschreibung von XDOMEA hierarchisch aufgebaut, d.h. die in ihm enthaltenen Elemente bzw. Inhaltsmodelle sind in einer über mehrere Ebenen gegliederten Baumstruktur angeordnet. So kann gemäß den fachlichen Anforderungen von XDOMEA ein entsprechendes Inhaltsmodell aufgebaut werden.

Diese globale komplexe Typdefinition ist sozusagen als Ankerpunkt von XDOMEA zu sehen und heißt `T_XDOMEA`. Der Typ besitzt ein Element `<XDOMEA_Daten>`, welches folgende Unterelemente enthält:

- Dokument 0..*
- Vorgang 0..*
- Akte 0..*
- Adresse 0..*

Aus fachlichen Überlegungen heraus werden dem Element `<Dokument>` folgende Unterelemente zugeordnet:

- Identifizierung
- Verweis
- Beschreibung

- Primärdokument
- Adress_Informationen

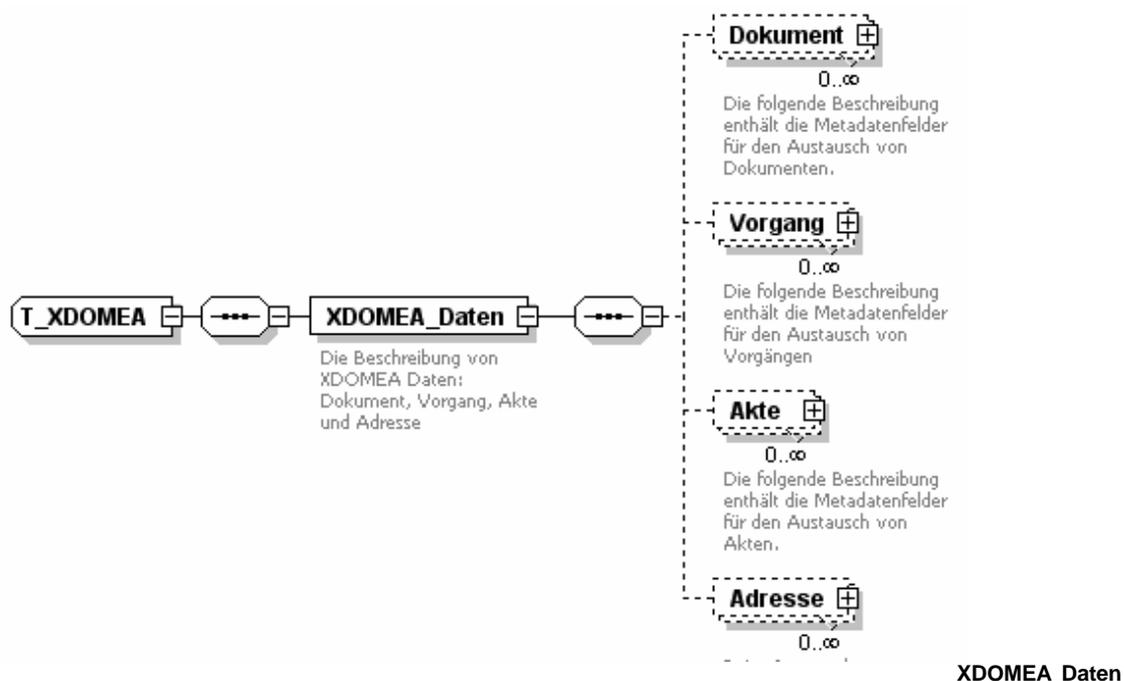
Das Element <Vorgang> besitzt folgende Unterelemente:

- Identifizierung
- Verweis
- Beschreibung
- Adress_Informationen

Das Element <Akte> besitzt folgende Unterelemente:

- Identifizierung
- Verweis
- Beschreibung
- Adress_Informationen

Der vorstehend beschriebene Grundaufbau ist in nachfolgendem Schaubild nochmals zusammengefasst.

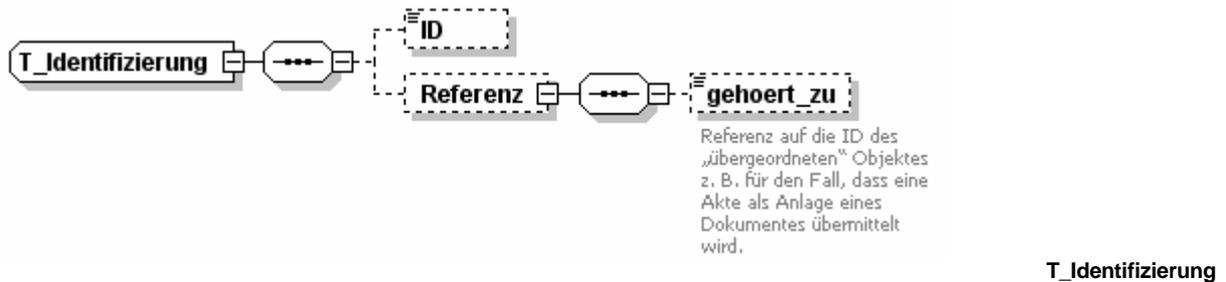


Die Beschreibung der Elemente kann den inhaltlichen Beschreibungen XDOMEA-Dokument, XDOMEA-Vorgang, XDOMEA-Akte und XDOMEA-Adresse in ihren jeweiligen Versionen entnommen werden.

Die Elemente <Dokument>, <Vorgang> und <Akte> besitzen alle das Unterelement <Identifizierung> mit identischem Inhaltsmodell. Das wieder verwendbare Inhaltsmo-

dell wird in Form eines benannten globalen komplexen Typs definiert: T_Identifizierung.

4.2 T_Identifizierung



In der vorliegenden Struktur können Dokument, Vorgang und Akte durch eine ID identifiziert werden. Diese ID muss im XML-Dokument eindeutig sein.

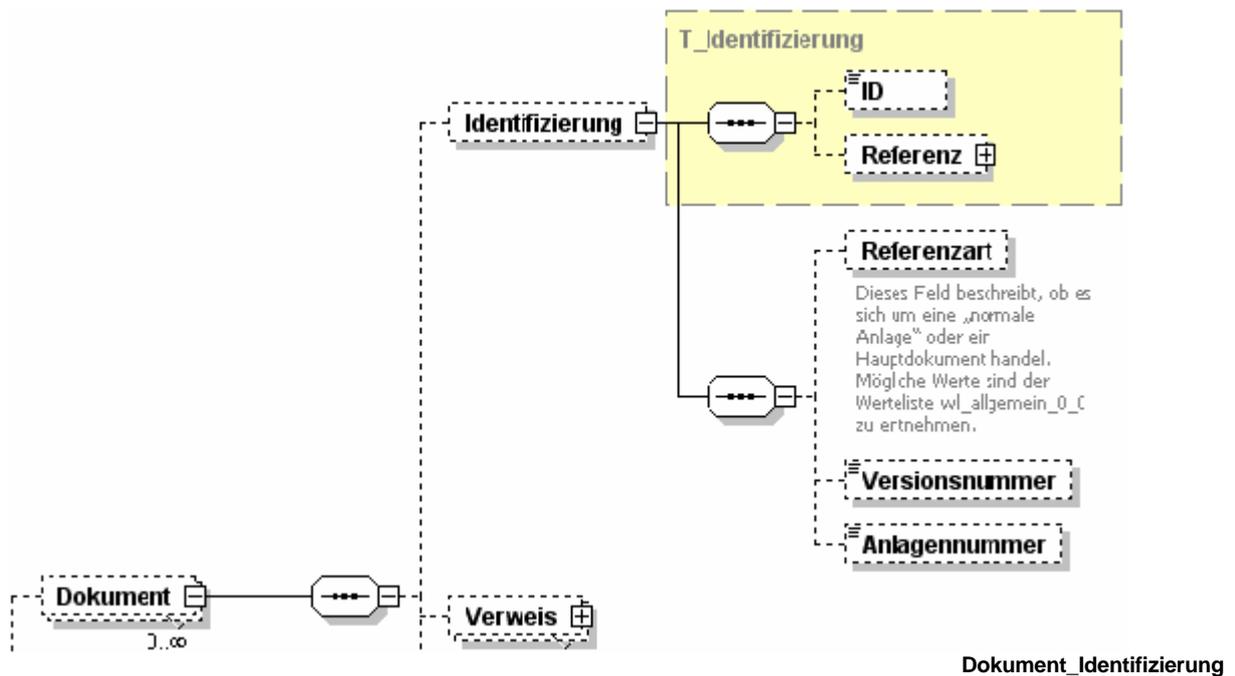
Das Element <Referenz/gehört_zu> verweist auf andere IDs im XML-Dokument. So kann, fachlich gesehen, ein Dokument einem übergeordneten Vorgang oder einem anderen gleichgestellten Dokument zugeordnet werden. Ebenfalls kann eine Akte eine Anlage eines Dokumentes sein.

Die Verweisteknik basiert auf Schlüsseldefinitionen, die weiter unten erläutert werden.

Das Element <Identifizierung> unter dem Vorgang und der Akte besitzt genau diese Struktur, einzig beim <Dokument> werden zusätzlich zu der beschriebenen Identifizierung Elemente benötigt. Dies geschieht über eine Erweiterung des komplexen Typs *T_Identifizierung* (<xs:extension base="T_Identifizierung">). Ein Dokument besitzt also den Typ T_Identifizierung und darüber hinaus folgende zusätzliche Elemente:

- Referenzart: hinter diesem Element liegt eine Werteliste mit den möglichen Werten „Anlage“, „Hauptdokument“, „Frühere Version“ und beschreibt somit das referenzierende Element <Dokument>. Zur Kennzeichnung der Anlage und der Früheren Version sind nachfolgende Elemente vorgesehen:
- Versionsnummer: Falls eine Version vorliegt, ist hier die Versionsnummer anzugeben.

- Anlagenummer: Falls eine Anlage vorliegt, ist hier die Anlagenummer anzugeben, falls eine Nummerierung vorhanden ist.

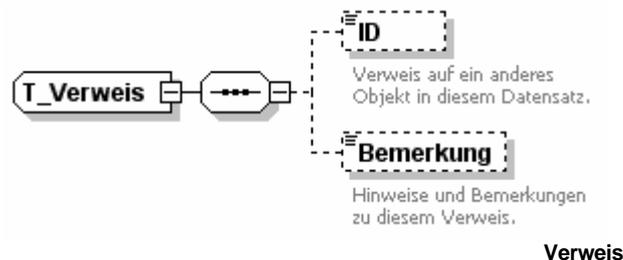


In obigem Schaubild sind das Element <Identifizierung> mit dem Typ *T_Identifizierung* und die zusätzlichen Elemente des Dokuments zu sehen.

Betrachtet man nun wieder eine Ebene höher die Gemeinsamkeiten der drei Objekte (Dokument, Vorgang, Akte) ist zu erkennen, dass die Gemeinsamkeiten nicht nur die Identifizierung betreffen, sondern auch den Verweis und die Adress_Informationen. Deswegen existiert für die Elemente <Verweis> und <Adress_Informationen> jeweils ein globaler komplexer Typ. Die <Beschreibung> unterscheidet sich allerdings bei den Objekten (siehe lokale Elementbeschreibung).

4.3 T_Verweis

Der Verweis bezieht sich auf das Unterelement <ID> eines im XML-Dokument befindlichen Objektes, und verfügt zudem über ein Element <Bemerkung>, das die Natur der bestehenden Beziehung genauer kennzeichnet.

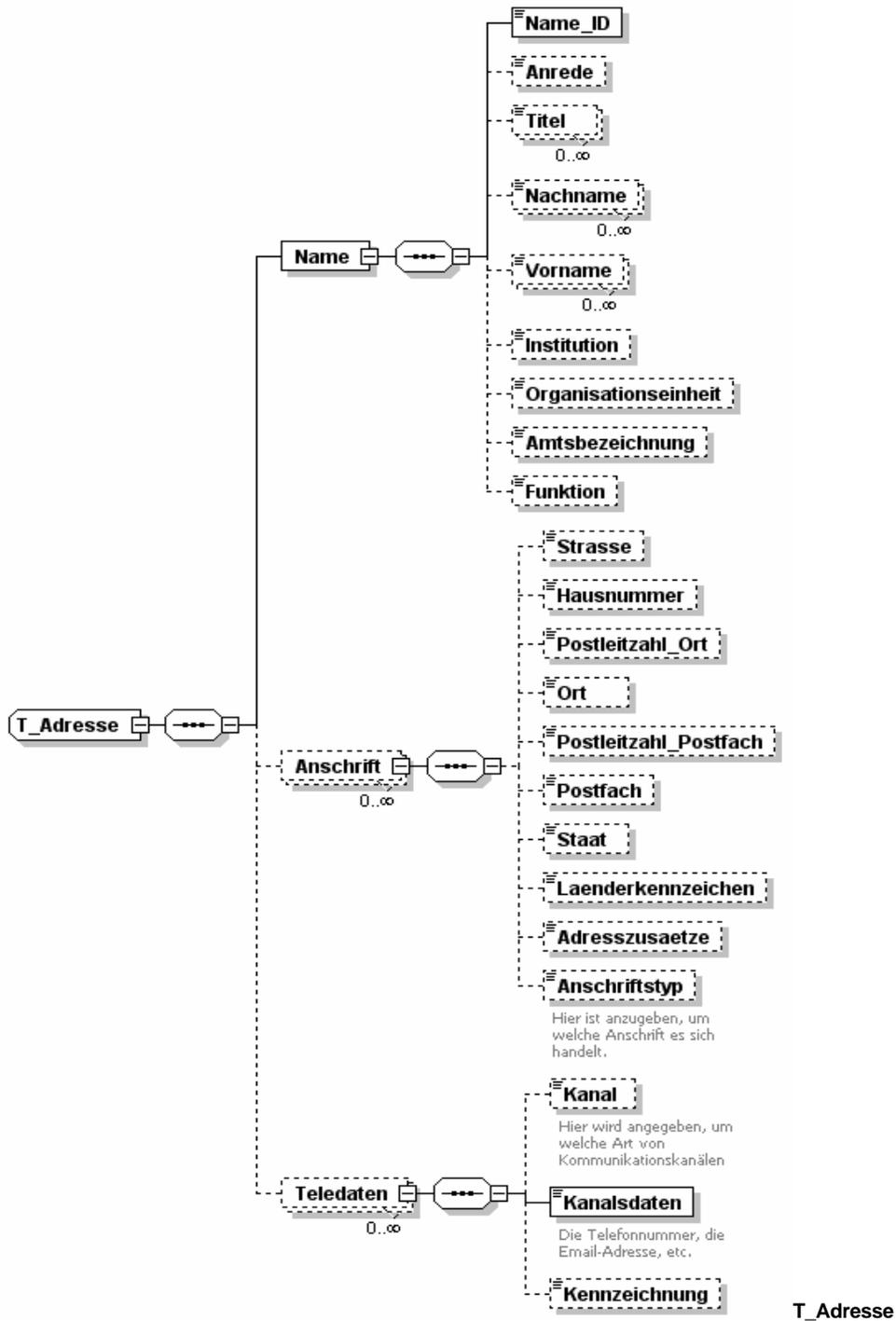


4.4 T_Adresse

In dem globalen komplexen Typ ist die Adresse mit Name, Anschrift und Telekommunikationsangaben eines Absenders oder Empfängers zu finden. Die Differenzierung, ob es sich um einen Absender oder Empfänger handelt, findet jeweils auf der lokalen Ebene der Objekte (Dokument, Vorgang, Akte) statt (siehe lokale Elementbeschreibung).

Über das Element <Adresse> 0..* können alle vorkommenden Personen oder Institutionen mit ihren Anschriften und Teledaten hier nur einmal erfasst werden. Durch eine Name_ID, die eine Person oder eine Institution über den NAMEN identifiziert, kann auf diese von anderer Stelle (z.B. von Absender in dem Objekt Dokument) referenziert werden.

Eine Person/Institution mit einem Namen kann mehrere Anschriften besitzen, sowie auch mehrere Teledaten.



5 Lokale Elementbeschreibungen

Lokale Elementbeschreibungen sind unter den Objekten Dokument, Vorgang, Akte und Adresse zu finden. Die Elementbeschreibungen, die sich in einer Typdefinition befinden, werden lokale Elemente genannt (da sie nicht direkt unter dem Schema-Rootelement stehen). Die lokalen Elemente beschreiben den Aufbau eines Dokuments, eines Vorgangs und einer Akte.

5.1 Zum Dokument

Das Element <Dokument> besitzt neben den Unterelementen, die mit einem globalen komplexen Typ ausgestattet sind, noch weitere Unterelemente.

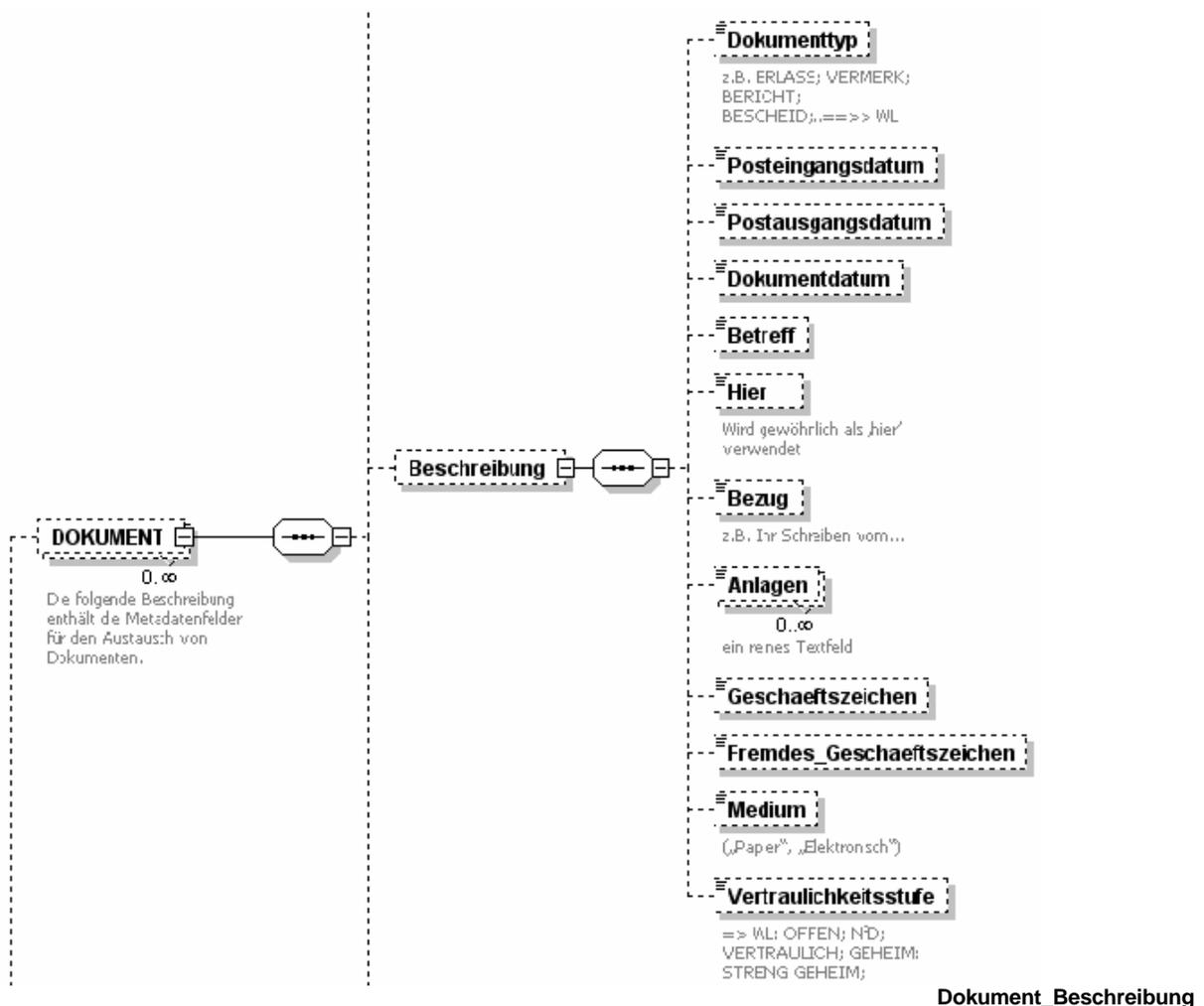
Die wesentlichen Unterelemente des Dokuments zu XDOMEA lauten wie folgt:

- Identifizierung
- Verweis
- Beschreibung
- Primärdokument
- Adress_Informationen (indirekter globaler Typ)

Die Identifizierung und der Verweis werden mit den globalen Typdefinitionen abgedeckt. Einzig die <Beschreibung> und das <Primaerdokument> sind nicht in einem globalen Typ definiert. Die Adress_Informationen besitzen indirekt einen globalen Typ.

5.1.1 Beschreibung

Die Beschreibung des Dokuments lässt sich aus folgendem Schaubild erkennen:

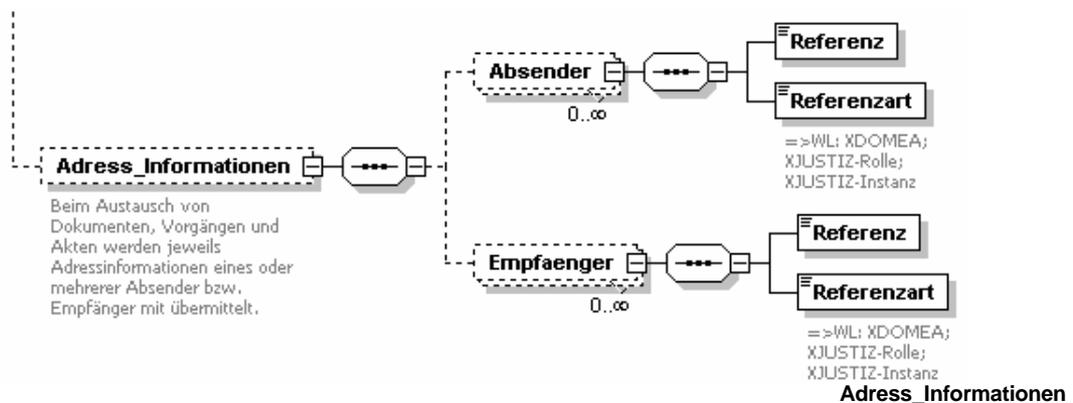


5.1.2 Primärdokument

Das Primärdokument enthält Metainformationen über das eigentlich beschriebene Dokument. Hierunter sind Dateiangaben zu finden wie folgt:

- **Dateiname:** Der innerhalb eines Datensatzes eindeutige Name der Datei inklusiv der Dateierweiterung, die das Dokument enthält in Form von „nnnnnn*.fff“ (*name*.format*)
- **Dateiformat:** Das Dateiformat des Dokuments wird hier beschrieben, sowie die Software mit Versionsnummer, die zur Erstellung des Dokuments verwendet wurde. z.B. MS Word 10.0; Acrobat 6.0
- **Ursprungsformat:** Dieses Element ist notwendig, falls eine Signatur eines Dokuments vorliegt. Das Format der Datei, die signiert wurde, wird angegeben.
- **Ersteller:** Hier steht der Ersteller des Primärdokuments.
- **Datum:** Datum der Erstellung des Primärdokuments in JJJJ-MM-TT.
- **Uhrzeit:** Uhrzeit der Erstellung des Primärdokuments in HH:MM:SS.

5.1.3 Adress_Informationen



Die Adress_Informationen bestehen jeweils optional aus einem oder mehreren Absendern und/oder Empfängern, die jeweils wieder Unterelemente besitzen. In dem Unterelement Referenz ist ein Verweis vorgesehen, der sich auf eine bereits angelegte Adresse dieses XDOMEA-Instanzdokuments bezieht. Realisiert wird der Verweis durch Angabe des Identifikators der Adresse: somit ist hier die Name_ID eines Beteiligten anzugeben. Nunmehr wird deutlich, wer als Absender bzw. Empfänger des beschriebenen Objekts (Dokument, Vorgang, Akte) verstanden wird.

Über die <Referenzart> wird beschrieben, ob es sich um eine XDOMEA-Adresse oder eine externe Adresse (z. B. XJustiz) handelt. Für gewöhnlich steht hier XDOMEA.

5.2 Zum Vorgang

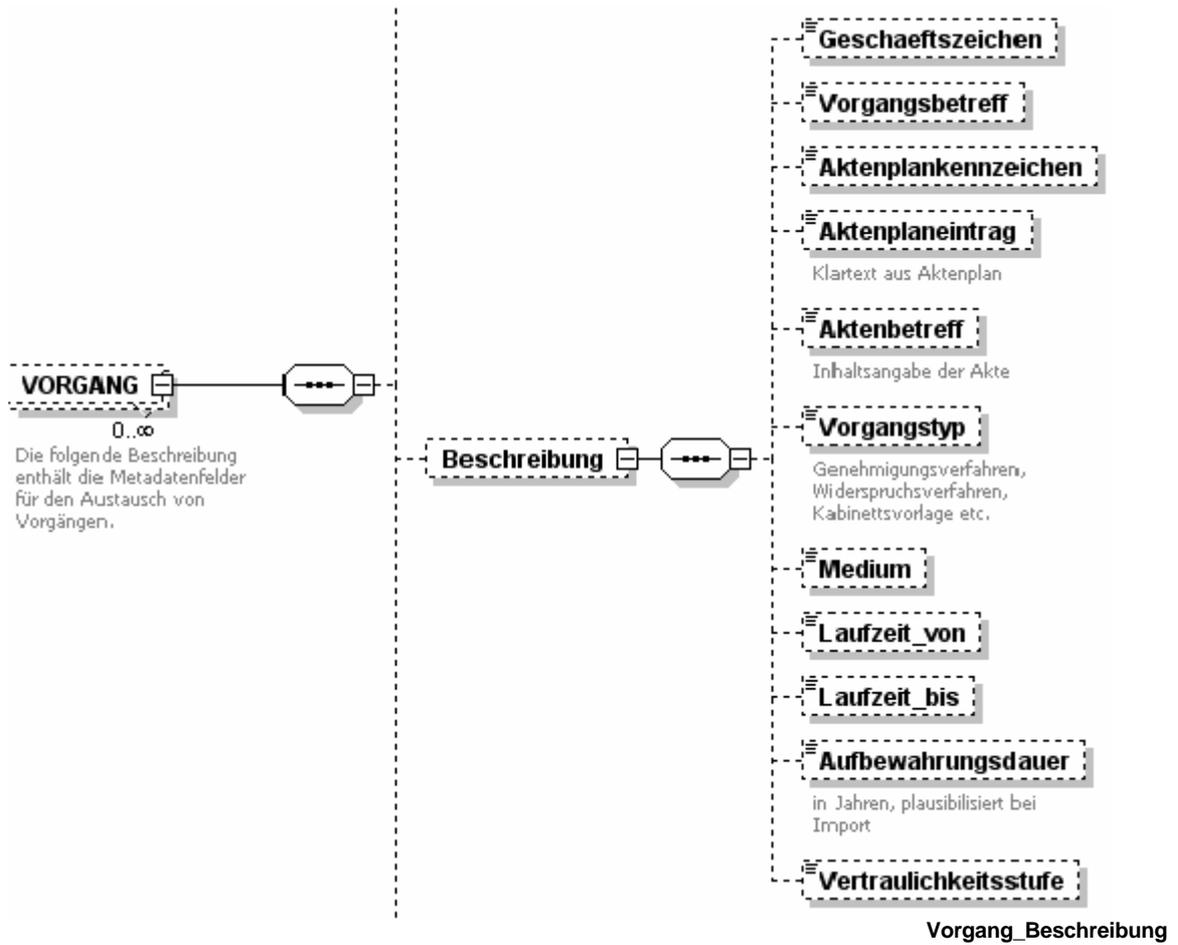
Das Element <Vorgang> hat folgende Unterelemente:

- Identifizierung
- Verweis
- Beschreibung
- Adress_Informationen

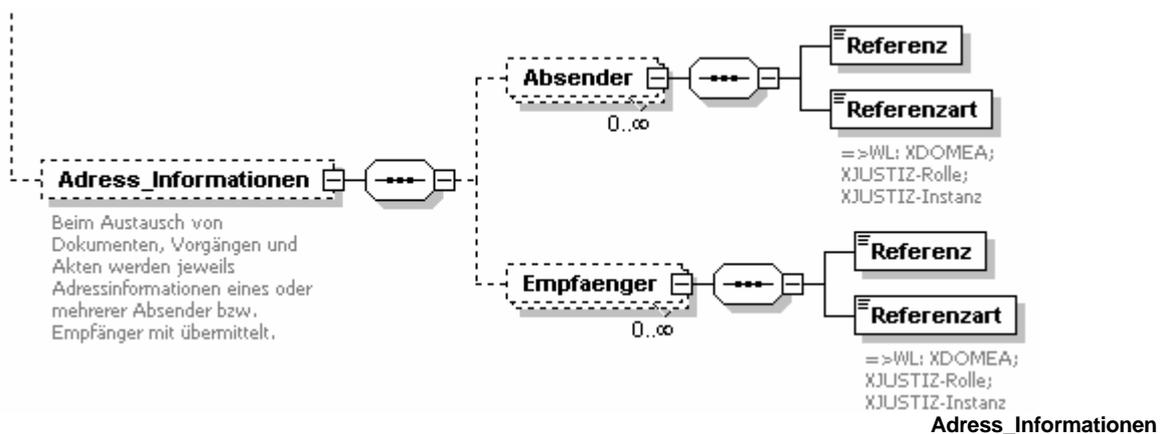
Auch beim <Vorgang> wie auch beim <Dokument> werden die Unterelemente <Identifizierung> und <Verweis> mit den globalen Typdefinitionen abgedeckt. Einzig die <Beschreibung> wird nicht in einem globalen Typ definiert. Die Adress_Informationen besitzen indirekt einen globalen Typ.

5.2.1 Beschreibung

Das Element <Beschreibung> ist aus folgendem Schaubild ersichtlich:



5.2.2 Adress_Informationen



Die Adress_Informationen beim Vorgang sind mit der Struktur der Adress_Informationen des Dokuments identisch (siehe Dokument).

5.3 Zur Akte

Die Akte enthält ebenfalls die Unterelemente <Identifizierung> und <Verweis>, die mit den globalen Typdefinitionen abgedeckt sind. Einzig die <Beschreibung> wird nicht in einem globalen Typ definiert. Die Adress_Informationen besitzen indirekt einen globalen Typ.

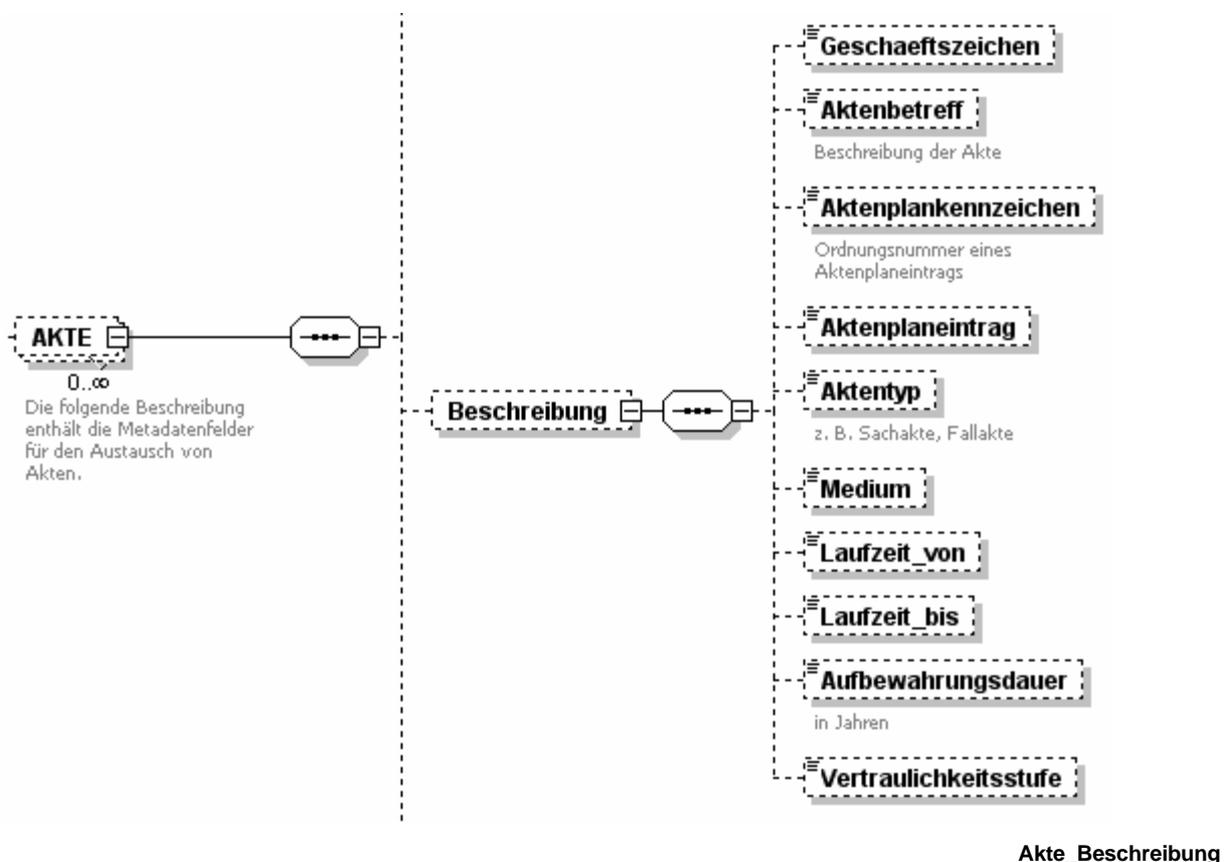
Das Element <Akte> hat folgende Unterelemente:

- Identifizierung
- Verweis
- Beschreibung
- Adress_Informationen

Die Angaben zu den Adress_Informationen sind mit denen des Vorgangs und des Dokuments vergleichbar. Die Beschreibung einer Akte setzt sich wie folgt zusammen.

5.3.1 Beschreibung

Das Element <Beschreibung> der Akte besitzt folgende Unterelemente:



5.3.2 Adress_Informationen

Diese Angaben sind vergleichbar mit den Adress_Informationen des Vorgangs und des Dokuments (siehe oben).

Ein überblickender Vergleich der Elemente <Beschreibung> und <Adress_Informationen> der drei Objekte soll nachfolgend eine Zusammenfassung geben.

DOKUMENT	VORGANG	AKTE
<Beschreibung>		
Geschaeftszeichen	Geschaeftszeichen	Geschaeftszeichen
Fremdes_Geschaeftszeichen		
Dokumentdatum		
Betreff	Vorgangsbetreff	Aktenbetreff
	Aktenplankennzeichen	Aktenplankennzeichen
	Aktenplaneintrag	Aktenplaneintrag
	Aktenbetreff	
Hier		
Bezug		
Dokumenttyp	Vorgangstyp	Aktentyp
Medium	Medium	Medium
Posteingangsdatum	Laufzeit_von	Laufzeit_von
Postausgangsdatum	Laufzeit_bis	Laufzeit_bis
	Aufbewahrungsdauer	Aufbewahrungsdauer
Vertraulichkeitsstufe	Vertraulichkeitsstufe	Vertraulichkeitsstufe
Anlagen		
<Adress_Informationen>		
Absender/Referenz	Absender/Referenz	Absender/Referenz
Absender/Referenzart	Absender/Referenzart	Absender/Referenzart
Empfaenger/Referenz	Empfaenger/Referenz	Empfaenger/Referenz
Empfaenger/Referenzart	Empfaenger/Referenzart	Empfaenger/Referenzart

Vergleich der Objekte

6 Identitätsbeschränkungen (Identity Constraints)

Um bestimmte Dokumentbereiche eindeutig zu halten und über Verweise zuordnen zu können, werden Schlüssel- und Eindeutigkeitsdefinitionen eingesetzt.

6.1 Schlüssel & Referenzen

Es kann vorkommen, dass dieselbe Information an verschiedenen Stellen innerhalb desselben XML-Instanzdokuments angegeben ist. Um dies zu vereinfachen, wird die mehrfach zu nutzende Information einmal angelegt und von anderen Stellen darauf Bezug genommen. Es besteht nun die Notwendigkeit innerhalb des Instanzdokuments von anderen Stellen auf eine bestimmte Stelle zu verweisen. XML-Schema bietet hierfür Verweistechiken und Einschränkungsmöglichkeiten durch die Verwendung von Schlüsseldefinitionen.

In XDOMEA ist beispielsweise der Bezug eines Dokuments auf eine bereits angelegte Akte notwendig.

Um also in solchen Fällen Redundanzen zu vermeiden, enthält XDOMEA an einigen Stellen Verweisungsmechanismen, die ähnlich funktionieren wie Schlüsselbeziehungen in einer relationalen Datenbank. Dadurch brauchen Daten eines Dokuments, auf die in einem XML-Instanzdokument mehrfach Bezug genommen wird, in diesem XML-Instanzdokument nur einmal aufgeführt zu werden. An allen anderen Stellen des Dokuments, an denen diese Daten erforderlich sind, genügt dann ein Verweis auf das entsprechende Element, das die vollständigen Daten enthält.

In XDOMEA ist ein Schlüssel definiert, der dem Element XDOMEA_Daten zugeordnet ist. Er bezieht sich auf die ID einer Identifizierung eines Objektes. Diese ID ist somit eindeutig und kommt instanzweit nur einmal vor.

Beispiel:

Es können verschiedene Dokumente, Vorgänge oder Akten vorhanden sein, die jeweils eine eigene ID-Nummer besitzen. Über diese ID können sie referenziert werden. Neben der ID besitzen sie eine Referenz (gehört_zu) über die eine Beziehung zu einer anderen ID eines anderen Objektes dargestellt werden kann.

Der Schlüssel in XDOMEA ist wie folgt aufgebaut: er wird durch das Schema-Element <xs:key> eingeleitet und besitzt einen Namen, über den er verwendet wer-

den kann. Zur Kennzeichnung der Elemente, aus denen der eindeutige Schlüssel bestehen soll, werden XPath-Ausdrücke verwendet. Dabei wird mit `<xs:selector>` eine Menge von Elementen spezifiziert, auf die der Schlüssel angewendet wird. Mit `<xs:field>` wird angegeben, aus welchen (Unter-)Elementen der Schlüssel besteht.

In dem XPath-Ausdruck, der über den Selektor gewählt wird, wird der Pfad angegeben, an dessen Stelle sich das Element befindet, welches eindeutig gehalten und worauf referenziert werden soll.

In XDOMEA wird eine Schlüsseldefinition für die Identifizierung eines Objekts (Dokument, Vorgang oder Akte) eingesetzt. Aber auch für die diversen Adressen wird ein Schlüssel eingesetzt.

Identifizierung der einzelnen IDs der Elemente:

```
<xs:key name="Schluessel">
  <xs:selector xpath="*/Identifizierung" />
  <xs:field xpath="ID" />
</xs:key>
```

Identifizierung der Person/Organisation über ihren Namen:

```
<xs:key name="Schluessel_Adresse">
  <xs:selector xpath="Adresse/Name" />
  <xs:field xpath="Name_ID" />
</xs:key>
```

Mit der Schema-Komponente `<xs:keyref>` sind Verweise eingerichtet, die sich mit ihrem `refer`-Attribut auf genau einen im Dokument definierten Schlüssel beziehen. Hier sind im Selektor und im Field-Element ebenfalls wieder über einen XPath-Ausdruck der Pfad und der Knoten angegeben, der die Verweisquelle darstellt.

Folgende Referenzen sind in XDOMEA definiert.

Verweis der Zugehörigkeit der Objekte (Dokument, Vorgang oder Akte):

```
<xs:keyref name="Referenz" refer="Schluessel">
  <xs:selector xpath="*/Identifizierung/Referenz" />
  <xs:field xpath="gehört_zu" />
</xs:keyref>
```

Verweis der Adresszuordnungen:

```
<xs:keyref name="Referenz_Adresse" refer="Schluessel_Adresse">
  <xs:selector xpath="*/Adress_Informationen/*" />
  <xs:field xpath="Referenz" />
</xs:keyref>
```

Es spielt eine große Rolle, in welchem Kontext ein Schlüssel definiert ist, denn dadurch wird sein Geltungsbereich festgelegt. Es ist erforderlich, dass die Definitionen der Schlüssel und der Schlüsselverweise dem Root-Element zugeordnet werden. Für XDOMEA bedeutet dies, dass die Schlüsseldefinitionen im Element „XDOMEA“ des Root-Schema definiert werden.

Externe Schemata, die XDOMEA einsetzen, müssen die entsprechenden Schlüsseldefinitionen von XDOMEA in ihr eigenes Root-Element übernehmen.

6.2 Eindeutigkeit

In XDOMEA wird eine Eindeutigkeitsdefinition verwendet, mit der festgelegt wird, dass instanzweit der Wert eines entsprechenden Elementes nur einmal existiert. Dies betrifft das Element Dateiname des Primärdokuments. Durch die Eindeutigkeitsdefinition kann der Dateiname als Wert nur einmal im Instanzdokument auftreten.

Eindeutigkeitsdefinition des Dateinamen:

```
<xs:unique name="Eindeutigkeit_Dateiname">  
  <xs:selector xpath="*/Dokument/Primaerdokument" />  
  <xs:field xpath="Dateiname" />  
</xs:unique>
```

Das Instanzdokument ist invalide, wenn das Element Dateiname zweimal denselben Wert besitzt.

7 Instanziierung

Ein XML-Dokument, das den Regeln der XML-Schemadatei XDOMEA entspricht, wird Instanzdokument genannt. In einem XML-Instanzdokument werden die eigentlichen Inhaltsdaten mit ihren Bezeichnungen transportiert. Der Aufbau und die Struktur des XML-Instanzdokuments richten sich nach den Regeln, die in dem XML-Schemadokument festgelegt wurden.

Zur Instanziierung wird der vom W3C vorgeschlagene Namensraum für Instanzdokumente mit dem Präfix xsi verwendet: `xmlns:xsi= "http://www.w3.org/2001/XMLSchema-instance"`. Dem ersten Element, auch Wurzelement genannt, werden alle weiteren Elemente untergeordnet. In ihm werden auch die Namensräume festgelegt. Im Falle von XDOMEA wird das Wurzelement mit `<XDOMEA>` bezeichnet. Die Verbindung des Instanzdokumentes zu der Schemadatei XDOMEA geschieht über das xsi-Attribut des Instanznamensraums `<xsi:schemaLocation>`, in dessen Wert der relative Pfad zu der relevanten Schemadatei anzugeben ist.

Darüber hinaus besitzt das Wurzelement `<XDOMEA>` ein Attribut `<XDOMEA_Version>` mit dem derzeitigen Wert 1.0. Über diese Version können Instanzdokumente den jeweiligen Versionen der Schemadateien zugeordnet werden.

Beispiel:

```
<?xml version="1.0" encoding="UTF-8" ?>
<XDOMEA xmlns="http://www.xdomea.de"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.xdomea.de xdomea_root_1_0.xsd"
  XDOMEA_Version="1.0">
<XDOMEA_Daten>.....
```

Ein Auszug eines Beispiels für ein XDOMEA valides XML-Instanzdokument wie folgt:

```
<XDOMEA_Daten>
  ± <DOKUMENT>
  ± <VORGANG>
  ± <AKTE>
  <ADRESSE>
    <Name>
      <Person_ID>ADR1</Person_ID>
      <Anrede>Frau</Anrede>
      <Titel>Prof.</Titel>
      <Nachname>Beispiel</Nachname>
      <Vorname>Berta</Vorname>
      <Amtsbezeichnung>Rätin</Amtsbezeichnung>
    </Name>
  <Anschrift>
```

```

    <Strasse>Beispielstrasse</Strasse>
    <Hausnummer>13</Hausnummer>
    <Postleitzahl_Ort>77777</Postleitzahl_Ort>
    <Ort>Beispielhausen</Ort>
    <Staat>Beispielland</Staat>
    <Laenderkennzeichen>DE</Laenderkennzeichen>
    <Anschriftstyp xsi:type="WL_Anschriftstyp"
wl_version="1.0">Hauptsitz</Anschriftstyp>
  </Anschrift>
  <Teledaten>
    <Kanal>EMail</Kanal>
    <Kanalsdaten>beispiel@beispiel.de</Kanalsdaten>
    <Kennzeichnung xsi:type="WL_Telekennzeichnung"
wl_version="1.0">geschäftlich</Kennzeichnung>
  </Teledaten>
</ADRESSE>
</XDOMEA_Daten>

```

Somit können hierüber die Nutzdaten zu XDOMEA für Dokumente, Vorgänge und Akten transportiert werden.

7.1 Wertelisten

Der Einsatz von Wertelisten ist an die entsprechenden Wertelistentypen gebunden. Die in XDOMEA vorhandenen Wertelistentypen sind wie folgt aufgezählt:

- WL_String
- WL_Anschriftstyp -> Element: Adresse/Anschrift/Anschriftstyp
- WL_Dokumenttyp -> Element: Dokument/Beschreibung/Dokumenttyp
- WL_Kommunikationsart -> Element: Adresse/Teledaten/Kanal
- WL_Medium -> Element: */Beschreibung/Medium
- WL_Referenzart -> Element: Dokument/Identifizierung/Referenzart
- WL_Telekennzeichnung -> Element: Adresse/Teledaten/Kennzeichnung
- WL_Vertraulichkeitsstufe -> Element: */Beschreibung/Vertraulichkeit

Es besteht für das entsprechende Element die Wahl, einen hierfür bestimmten Wertelistentyp zu verwenden oder einen gewöhnlichen Stringwert einzusetzen. Diese Entscheidung wird bei der Instanziierung getroffen und zwar über das Attribut des Instanznamensraumes <xsi:type>.

Beispielsweise wird dem entsprechenden Element (hier Anschriftstyp) das Attribut <xsi:type> hinzugefügt. Der Wert des Attributs erhält nun den Wertelistentyp. Ist dieser angegeben, ist ein weiteres Attribut <wl_version> erforderlich und mit der entsprechenden Version des Wertelistentyps anzugeben. Folglich muss der Wertinhalt des Elementes zwingend einem Wert des entsprechenden Wertelistentyps der Wer-

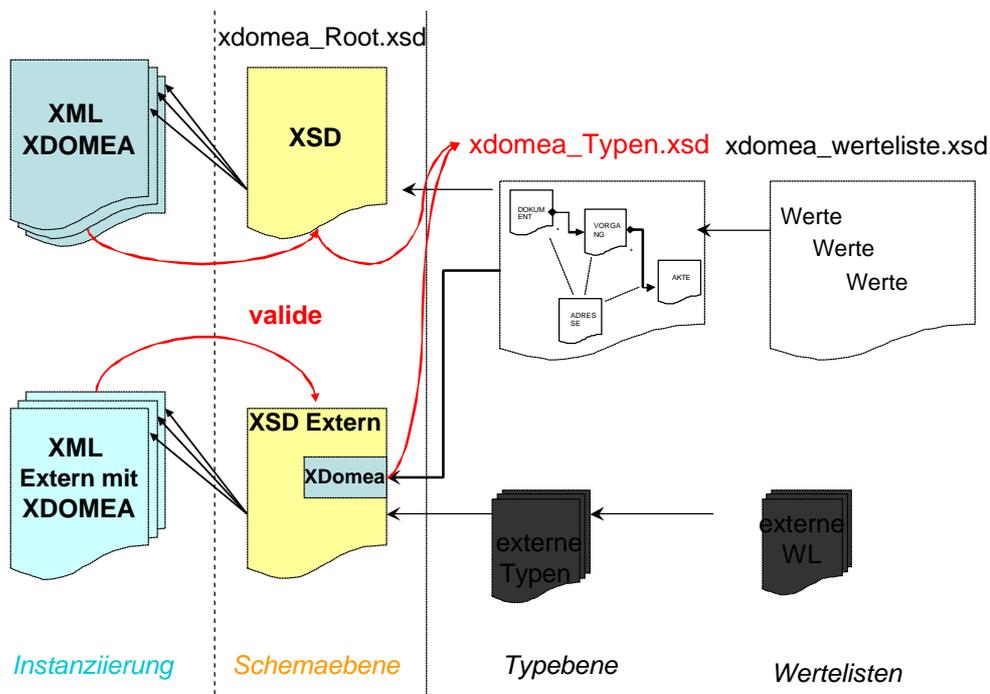
teliste gleichen. Optional kann noch eine Fassung über das Attribut <wl_fassung> hinzugefügt werden. Bei Fehlerbereinigungen beispielsweise könnte über die Fassung versioniert werden, ist aber nicht zwingend.

Liegt keine derartige Typisierung vor, kann der Wert des Elementes ein Stringwert aufweisen. Somit ergeben sich zwei gültige Möglichkeiten:

```
<Anschriftstyp xsi:type="WL_Anschriftstyp"
wl_version="1.0">Hauptsitz</Anschriftstyp>
oder
<Anschriftstyp>irgendetwas</Anschriftstyp>
```

Durch die Definition der Wertelistentypen ist über das Attribut <wl_version> und über das optionale Attribut <wl_fassung> eine Versionierung möglich.

Im Allgemeinen läuft die Instanziierung von XDOMEA über das Root-Schemata xdomea_root_1_0.xsd. Es besteht aber auch die Möglichkeit durch Inklusion des Typenschemata in einen externen Bereich die Typdefinitionen zu XDOMEA zu verwenden. Hierbei sind die entsprechenden XML-Instanzdokumente ihren Schemata gegenüber valide. Zur Veranschaulichung dient folgendes Bild.



Zusammensetzung

8 Version

Um eine entsprechende Validierung der Instanzdokumente gegenüber den korrekten Schemadateien zuverlässig durchführen zu können, muss klar sein, welche Schemadatei als Maßstab heranzuziehen ist. Sobald mehrere Versionen einer Schemadatei vorliegen, kann dies zu Schwierigkeiten führen. Das Problem der Versionierung der Schemadateien wird über ein eingeführtes Attribut gelöst.

Die Versionierung der Schemadateien geschieht also über ein definiertes Attribut, das dem komplexen Typ *T_XDOMEA* (*xdomea_typen*) zugeordnet ist und somit im Instanzdokument im Rootelement `<XDOMEA>` erscheint.

Definition:

```
<xs:attribute name="XDOMEA_Version" type="xs:string" use="required" fixed="1.0"
```

Instanziierung:

```
<XDOMEA xmlns="http://www.xdomea.de"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.xdomea.de xdomea_root_1_0.xsd"
  XDOMEA_Version="1.0" >
```

Dadurch ist auch gewährleistet, dass ein externer Bereich (bspw. Justizwesen), der XDOMEA verwendet, auch die aktuelle Versionsnummer von XDOMEA benutzt und eine entsprechende Versionierung sichergestellt ist.

8.1 Bestandteile der Versionsnummer

In der ersten Version von XDOMEA hat die Versionsnummer über das Attribut `<XDOMEA_Version>` bei allen Schemadateien den Wert *1.0*.

Die Stellen der Versionsnummer haben folgende Bedeutung:

Die erste Stelle (1.x.x) identifiziert die Haupt-Version. Sie wird nur dann heraufgesetzt, wenn das Design von XDOMEA grundlegende strukturelle Änderungen oder Erweiterungen erfährt.

Die zweite Stelle (x.0.x) identifiziert die Unter-Version. Diese Stelle wird heraufgesetzt, wenn Änderungen an der Datenstruktur vorgenommen werden, die das grundlegende Design unberührt lassen.

Die dritte Stelle (x.x.0) ist optional für die Beseitigung von Fehlern und die Vornahme kleinerer Änderungen ohne Auswirkungen auf die Datenstruktur vorgesehen. Sie entspricht einer Art Fassung.

8.2 Angabe der Versionsnummer im Dateinamen

Die ersten beiden Stellen der Versionsnummer sind zugleich Bestandteil des Dateinamens jeder XML-Schemadatei.

Beispiel: Die Version 1.0 der `xdomea_typen` hat den Dateinamen `xdomea_typen_1_0.xsd`. Dieser Dateiname wird bei Versionen mit derselben Datenstruktur (z.B. 1.0.1, 1.0.2 usw.) beibehalten.

Eine spätere Version 1.1 erhält dagegen den Dateinamen `xdomea_typen_1_1.xsd`.

Eine Änderung dieses Dateinamens hat zur Folge, dass von dem Root-Schema `xdomea_root_1_0`, das auf die neue Version zugreifen soll, ebenfalls eine neue Version angelegt werden muss, in der die `xs:include`-Befehle an den neuen Dateinamen angepasst sind.

Diese und weitere Informationen sind auf <http://www.xdomea.de> erhältlich.

9 Abbildungsverzeichnis

Zusammenhänge.....	8
XDOMEA_Daten.....	11
T_Identifizierung	12
Dokument_Identifizierung	13
Verweis.....	14
Dokument_Beschreibung.....	17
Adress_Informationen.....	18
Vorgang_Beschreibung	19
Adress_Informationen.....	19
Akte_Beschreibung.....	20
Vergleich der Objekte	21
Zusammensetzung	27