



<b>E-Government XML Strukturen für Geschäftsobjekte</b>		Konvention	
		xml-g 1.0.1	
		Empfehlung	
Kurzbeschreibung:	<p>XML-Strukturen sind ein wesentlicher Bestandteil der Kommunikation zwischen den einzelnen E-Government-Systemen. Verfahrensinformationen bilden dabei den Kern dieser Strukturen.</p> <p>Der vorliegende Entwurf schlägt eine Gliederung dieser sogenannten Geschäftsobjekte in Containerobjekte und Fachobjekte vor und definiert ein Regelwerk für die Generierung dieser Strukturen.</p>		
Autor(en):	Luef, Grandits	Projektteam / Arbeitsgruppe:	Kommunikations- architektur

Stelle:	Vorgelegt am:	Angenommen am:	
Bund	08.03.2004	11.05.2004	z.K.g.
Länder	08.03.2004	29.03.2004	
Gemeindebund	08.03.2004	29.03.2004	z.K.g.
Städtebund	08.03.2004	29.03.2004	

XML-Gesellschaft e.V.

---

**Dokumentklasse:** Erläuterung

**Dokumentstadium:** Entwurf öffentlich

## E-Government – XML-Strukturen für Geschäftsobjekte

### (1) Einleitung

XML-Strukturen sind ein wesentlicher Bestandteil der Kommunikation zwischen den einzelnen E-Government-Systemen. Verfahrensinformationen bilden dabei den Kern dieser Strukturen.

Der vorliegende Entwurf schlägt eine Gliederung dieser sogenannten Geschäftsobjekte in Containerobjekte und Fachobjekte vor und definiert ein Regelwerk für die Generierung dieser Strukturen.

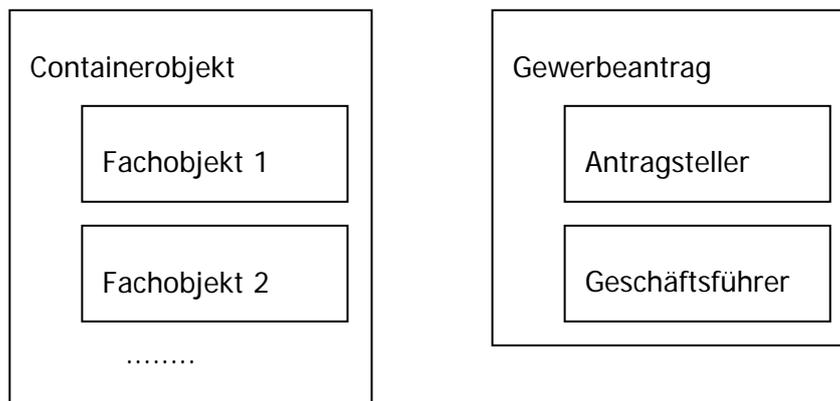
Dadurch sollen folgende Ziele erreicht werden:

- Verbesserung der Interoperabilität zwischen den e-Government-Systemen
- Geringere Kosten für die Definition von XML-Strukturen
- Automatisierte Generierung von Strukturdefinitionen

### (2) Grundstruktur

Als Geschäftsobjekte werden jene Informationen bezeichnet, welche zwischen E-Government-Systemen ausgetauscht werden, wie z.B. ein Antrag auf Gewerbeausübung oder die Übermittlung von Wahlergebnissen.

Die Geschäftsobjekte bestehen aus Containerobjekten und darin eingebetteten Fachobjekten.



### (3) Containerobjekte

Die Containerobjekte bilden quasi das Kuvert, in dem Fachobjekte übermittelt werden. Sie lassen sich in unterschiedliche Typen unterteilen:

- Anträge
- Zustellung
- Verwaltungsinterne Übermittlung
- .....

Von diesen Grundtypen werden die speziellen Objekte abgeleitet, wie z.B. ein Antrag auf Gewerbeausübung.

### (4) Fachobjekte

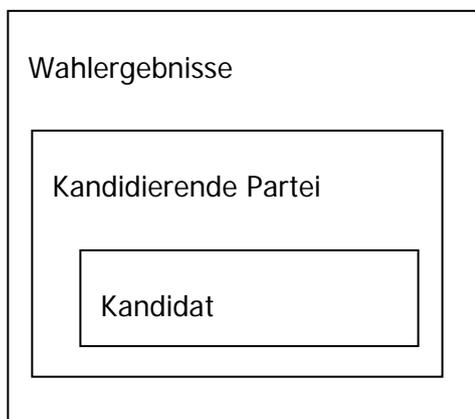
Hier handelt es sich um Objekte des natürlichen Lebens, welche in den einzelnen Verfahren relevant sind:

- Personen
- Adressen
- Bankverbindungen
- Grundstücke
- Betriebe
- .....

Jedes dieser Objekte kann in verschiedenen Rollen auftreten. Folgende Rollen sind für eine Person denkbar:

- Antragsteller
- Ehepartner
- Kind
- Geschäftsführer
- Usw.

Fachobjekte bestehen aus einer Menge von Attributen sowie weiteren Fachobjekten.

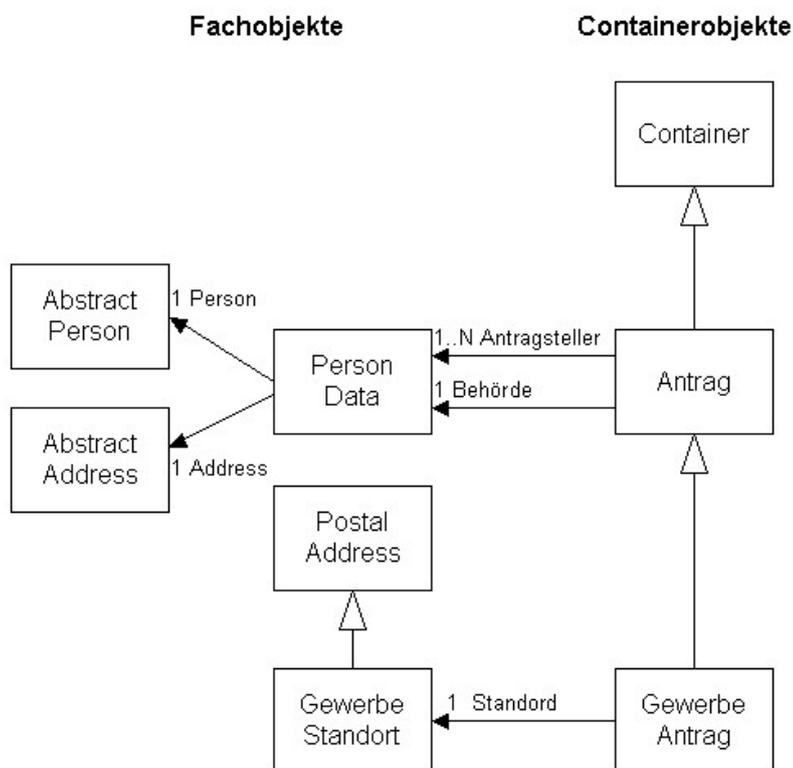


## (5) UML-Darstellung

Als Darstellung der Geschäftsobjekte und Beziehungen wird UML (Unified Modeling Language) empfohlen. Diese Darstellung erlaubt die Ableitung von XML-Schema, Datenbankschema, Objektmodell usw. aus einer einzigen Quelle. Durch geeignete Tools können diese Ableitungen auch generiert werden.

In UML werden Objekte und Beziehungen als Kästchen und Linien dargestellt. Attribute eines Objektes werden innerhalb seines Kästchen dargestellt. Die wichtigsten Beziehungen sind Assoziation (symbolisiert durch einen Pfeil) und Vererbung (senkrechte Linie mit einer Pyramide an der Spitze).

Am Beispiel eines Gewerbeantrages sieht man im folgenden UML-Modell das allgemeine Objekt *Container*, und davon durch Vererbung abgeleitet, als spezielle Container, den *Antrag*, und wieder spezieller den *Gewerbeantrag*. Ein allgemeiner Antrag hat mindestens einen Antragsteller und genau eine Behörde zugeordnet, wobei beides *PersonData*-Objekte sind. Das PersonData-Objekt gliedert sich weiter in *AbstractPerson* und *AbstractAddress*, usw. Der Gewerbeantrag hat, als Spezialisierung des Antrages, noch zusätzlich einen *GewerbeStandort* zugeordnet, der eine erweiterte Form einer Adresse darstellt (Die Eigenschaften von *PostalAddress* und zusätzlich die Gemeinde sowie ein Erläuterungsattribut).



Es ist zu beachten, dass Objekte wie *Container*, *Antrag*, *PersonData* Use Case übergreifend sind, während andere, wie *GewerbeStandort*,

spezifisch für einen Use Case, hier den Gewerbeantrag, definiert sind. Die allgemeinen Objekte werden als Rahmen für die Use Cases entwickelt. Spezifische Objekte können, wie oben gezeigt, solche allgemeinen Objekte durch Vererbung spezialisieren. Eine Vererbungsbeziehung ist aber keinesfalls zwingend; spezifische Objekte können auch durch Assoziationen mit den allgemeinen Objekten verknüpft sein.

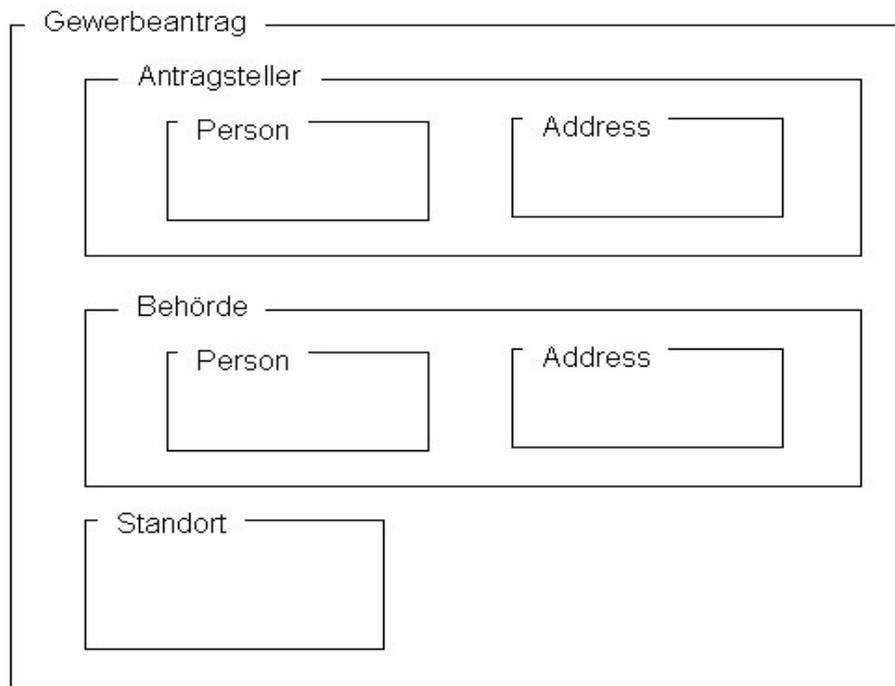
Weiters können Use Cases zusätzliche Felder bei generellen Objekten notwendig machen. Diese spezifischen Attribute können auch durchaus in das allgemeine Modell eingehen, sodass sie auch allen anderen Use Cases zur Verfügung stehen, auch wenn sie vielleicht nicht überall benötigt werden.

## (6) Ableitung von XML-Strukturen aus UML

UML ist geeignet, ein hinreichend abstraktes Modell der Geschäftsobjekte zu beschreiben. Die Objekte werden dann in verschiedenen Formen verarbeitet, z.B. in Datenbanktabellen, Strukturen (Objekten) von Programmiersprachen und XML-Dokumenten. In der Folge wird auf die Darstellungsform der XML-Dokumente näher eingegangen.

Die hier betrachteten XML-Dokumente entsprechen Container-Objekten, die wiederum Fachobjekte enthalten, usw. In der folgende Grafik wird ein XML-Dokument gezeigt, das dem Gewerbeantrag des zuletzt beschriebenen UML-Modelles entspricht. Assoziationen aus dem UML-Modell sieht man dabei als Einbettungen: Der Standort des Gewerbeantrages, im UML-Modell durch eine Assoziation dargestellt, wird in den Gewerbeantrag als Teildokument eingebettet. Der Pfeil im UML-Modell zeigt die Richtung der Einbettung: Der Standort ist Teil des Gewerbeantrages und nicht umgekehrt.

Vererbungen aus dem UML-Modell sieht man im XML-Dokument so, dass alle Attribute und Assoziationen, die geerbt werden, als Attribute und Teildokumente vorhanden sind: In *GewerbeAntrag* sind *Antragsteller* und *Behörde* eingebettet, weil *GewerbeAntrag* von *Antrag* erbt.



Aus dem UML-Modell kann die Gestalt der XML-Dokumente als XML-Schema abgeleitet werden. Im Allgemeinen sind aus einem UML-Modell viele XML-Schemata ableitbar, weil die Umsetzung verschiedener Konzepte wie Attribute, Assoziationen, Vererbung in ein XML-Schema unterschiedlich erfolgen kann. Aus diesem Grund bedarf es bei der Ableitung eines Schemas bestimmter Konventionen.

Wenn ein bestimmtes XML-Schema bereits vorliegt, kann daraus auch ein UML-Modell abgeleitet werden. Bestimmte Konzepte eines XML-Schemas, wie z.B. die Unterscheidung zwischen Attributen und Subelementen, sind in UML verschieden darstellbar. Aus diesem Grund

```

<xsd:element name='Gewerbeantrag'>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:sequence minOccurs="1" maxOccurs='unbounded'>
        <xsd:element ref='Antragsteller' />
      </xsd:sequence>
      <xsd:sequence minOccurs="1" maxOccurs="1">
        <xsd:element ref="Behörde"/>
      </xsd:sequence>
      <xsd:element ref='Standort' />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
  
```

bedarf es auch bei dieser Ableitung bestimmter Konventionen.

Das oben stehende Schema-Exzerpt für den Gewerbeantrag ist eine mögliche Ableitung. Für die Subelemente des Gewerbeantrages wurde ein `<xsd:sequence>`-Element verwendet. Das stellt eine an sich

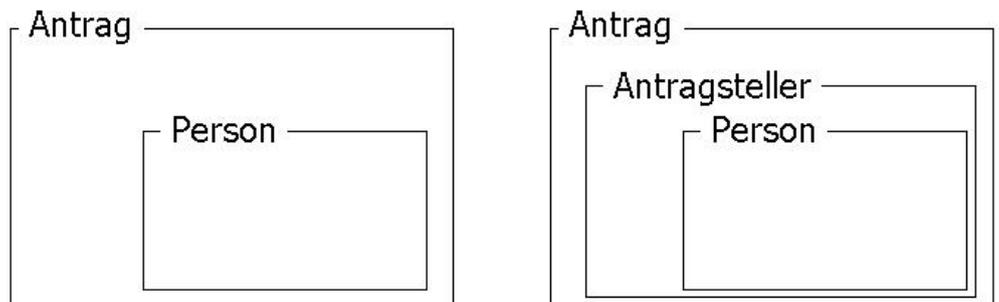
nicht zwingende Konvention dar, weil in UML die Reihenfolge der Assoziationen nicht definiert ist und statt dessen auch `<xsd:all>` verwendet werden könnte.

## (7) Personen und Rollen

### a) Mantel-Elemente für Rollen

Ein wesentliches Gestaltungsmittel im XML-Schema für Antragsdaten sind *Rollen*. Der Zweck von Rollen sind die Einbeziehung von Personen in den Antrag, z.B. die Einbeziehung einer natürlichen Person als Antragsteller.

Ein Antrag kann eine Person inkludieren, wie im linken Teil der folgenden Abbildung zu sehen ist. Wenn mehrere Personen beteiligt sind, muss man immer wissen, welche Person welche Rolle spielt. Zu diesem Zwecke wird die Person in einen „Mantel“ gehüllt, der ihre Rolle angibt, wie im rechten Teil der Abbildung. Dort ist der Mantel der Person ein Element mit der Bezeichnung „Antragsteller“, welches der Rolle der Person im Antrag entspricht.



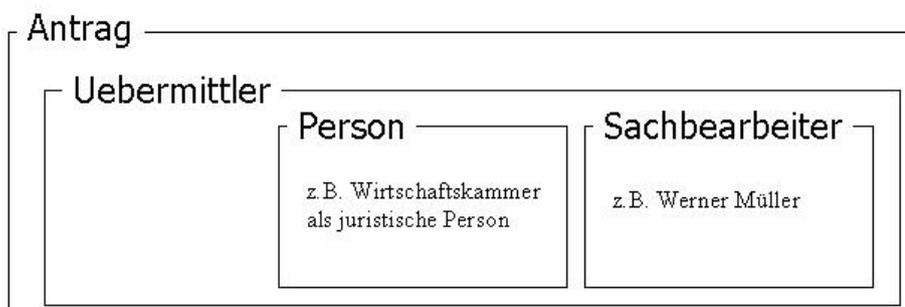
In reiner XML-Notation stellt sich der Rollen-Mantel des Antragstellers wie im folgenden Bild als einfaches Element `<Antragsteller>` dar. Das Mantel-Element hat in diesem Beispiel keine weiteren Attribute oder Subelemente als nur die Person. Diese Einschränkung auf sonst leere Mantel-Elemente wird im Allgemeinen nicht gelten, wie der nächste Abschnitt zeigt.

```
<Antrag>
  <Antragsteller>
    <Person xsi:type="pd:PhysicalPersonType">
      <pd:Name>
        <pd:GivenName>Max</pd:GivenName>
        <pd:FamilyName>Blumenmann</pd:FamilyName>
        <pd:Affix type="academicGrade">Mag.</pd:Affix>
      </pd:Name>
      <pd:MaritalStatus>married</pd:MaritalStatus>
      <pd:Sex>male</pd:Sex>
      <pd:DateOfBirth>1952-02-11</pd:DateOfBirth>
      <pd:PlaceOfBirth>Graz</pd:PlaceOfBirth>
      <pd:Nationality>Österreich</pd:Nationality>
      <pd:Geburtsname>Müller</pd:Geburtsname>
      <pd:Beruf>Blumenbinder</pd:Beruf>
    </Person>
  </Antragsteller>
</Antrag>
```

#### b) Besondere Eigenschaften bei Rollen

Wie bereits im Abschnitt **Fehler! Verweisquelle konnte nicht gefunden werden.** dargestellt wurde, sind mit Rollen besondere Eigenschaften verbunden. So etwa kann vom Antragsteller eine Bankverbindung erfasst sein, aber vom Übermittler nicht. Andererseits kann vom Übermittler eine Kontaktperson erfasst sein, die beim Antragsteller nicht vorkommt.

Diese „besonderen“ Eigenschaften von Personen, die nur bei bestimmten Rollen auftreten, sind Unterelemente des Rollen-Elements. Wie die folgende Abbildung zeigt, wird beim Übermittler ein Element für die übermittelnde Person (hier: Wirtschaftskammer als juristische Person) und ein weiteres für den Ansprechpartner aufgenommen.



### c) Personendaten in Rollen

Es gibt bei den Rollen-Elementen einige Regeln, die Adressen, Kommunikationsmittel und andere Personendaten betreffen. So zum Beispiel muss vom Antragsteller mindestens eine Adresse erfasst sein. Diese Vorschrift kommt im allgemeinen Schema der Personendaten nicht vor. Damit sie im XML-Schema der Antragsdaten trotzdem geprüft wird, erfolgt eine Erweiterung des Rollen-Elementes „Antragsteller“.

Das Element „Antragsteller“ hat daher verpflichtend ein Unterelement „PostalAddressType“. Im XML-Schema des Antrages wird somit geprüft, dass der Antragsteller eine Adresse besitzt. Analog dazu existiert ein Unterelement „AbstractAddressType“ dazu, das für Kontakte (Telefon, Fax, Email) vorgesehen ist, die Kontakte sind aber nicht verpflichtend.

### d) Personendaten im Allgemeinen

Außer bei Rollen können personenbezogene Objekte auch in anderen Objekten einbezogen werden. z.B. im Fachobjekt *Behoerde* oder im Fachobjekt *Standort* der Gewerbeanmeldung, die beide eine Adresse beinhalten.

Das Verfahren ist hier analog zu den Personendaten in Rollen. Die folgende Abbildung zeigt das Beispiel des Gewerbebestandes mit Adresse. *Standort* ist dabei der „Mantel“ um die Adresse, die den XML-Typ *PostalAddressType* hat.



```
<gwa:Standort>
  <gwa:Bezeichnung>Werkstätte</gwa:Bezeichnung>
  <gwa:Adresse>
    <pd:PostalCode>8010</pd:PostalCode>
    <pd:Municipality>Graz</pd:Municipality>
    <pd:DeliveryAddress>
      <pd:StreetName>Burggasse</pd:Stre
      <pd:BuildingNumber>2</pd:Building
    </pd:DeliveryAddress>
    <pd:Gemeindekennzahl>60101</pd:Gemeindekenn
    <pd:Gemeinde>Gemeinde Graz</pd:Gemeinde>
  </gwa:Adresse>
</gwa:Standort>
```

## (8) Entwicklung

Die generellen Geschäftsobjekte, sowohl Container- als auch Fachobjekte, werden als „strategischer“ Rahmen vorausentwickelt, in den sich dann die Modelle der einzelnen Use Cases einpassen. Es entsteht ein strategisches UML-Modell und daraus abgeleitet ein strategisches XML-Schema.

Für die bereits vorhandenen strategischen XML-Schemata im E-Government-Bereich muss das UML-Modell nachmodelliert oder, bei der Verfügbarkeit eines entsprechenden Werkzeuges, im Reverse-Engineering-Verfahren generiert werden. Es sind verschiedene solcher Werkzeuge erhältlich.

Die Konventionen bei der Ableitung von XML aus UML sind ebenfalls Bestandteil des strategischen Rahmens. Viele der tatsächlich gewählten Konventionen werden auf die Auswahl des verwendeten Werkzeuges zurückgehen.

Werden aus den UML-Definitionen, die über Reverse Engineering gewonnen wurden, wieder XML-Strukturen generiert, dann kann sich das Ergebnis vom Originalzustand unterscheiden. Das wird dann der Fall sein, wenn das originale XML-Schema nicht den Konventionen der Ableitung entspricht.

Die Geschäftsobjekte der Use Cases (z.B. Gewerbeantrag) werden grundsätzlich von „oben“ nach „unten“, von UML nach XML, erzeugt werden. So ist gewährleistet, dass diese Use Case-spezifischen Objekte in den strategischen Rahmen passen.

Auch bei den Use Cases ergeben sich jedoch bereits fertige XML-Strukturen, etwa durch Erzeugung aus einem Datenbank-Design-Werkzeug oder einem Formulargenerator. Bei Verfügbarkeit eines entsprechenden UML-XML-Werkzeuges können diese Strukturen mittels Reverse Engineering ins UML-Modell aufgenommen werden. Ist Reverse Engineering nicht möglich, so muss eine händische Nachdefinition in UML dann vorgenommen werden, wenn strategische Objekte betroffen sind. Sind die fertigen XML-Strukturen nur Use Case-bezogen, ohne Verknüpfung zu strategischen Objekten, dann wird eine Nachdokumentation empfohlen, aber nicht gefordert.