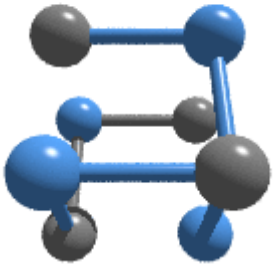


Präsenzveranstaltung zur E-Learning-Veranstaltung

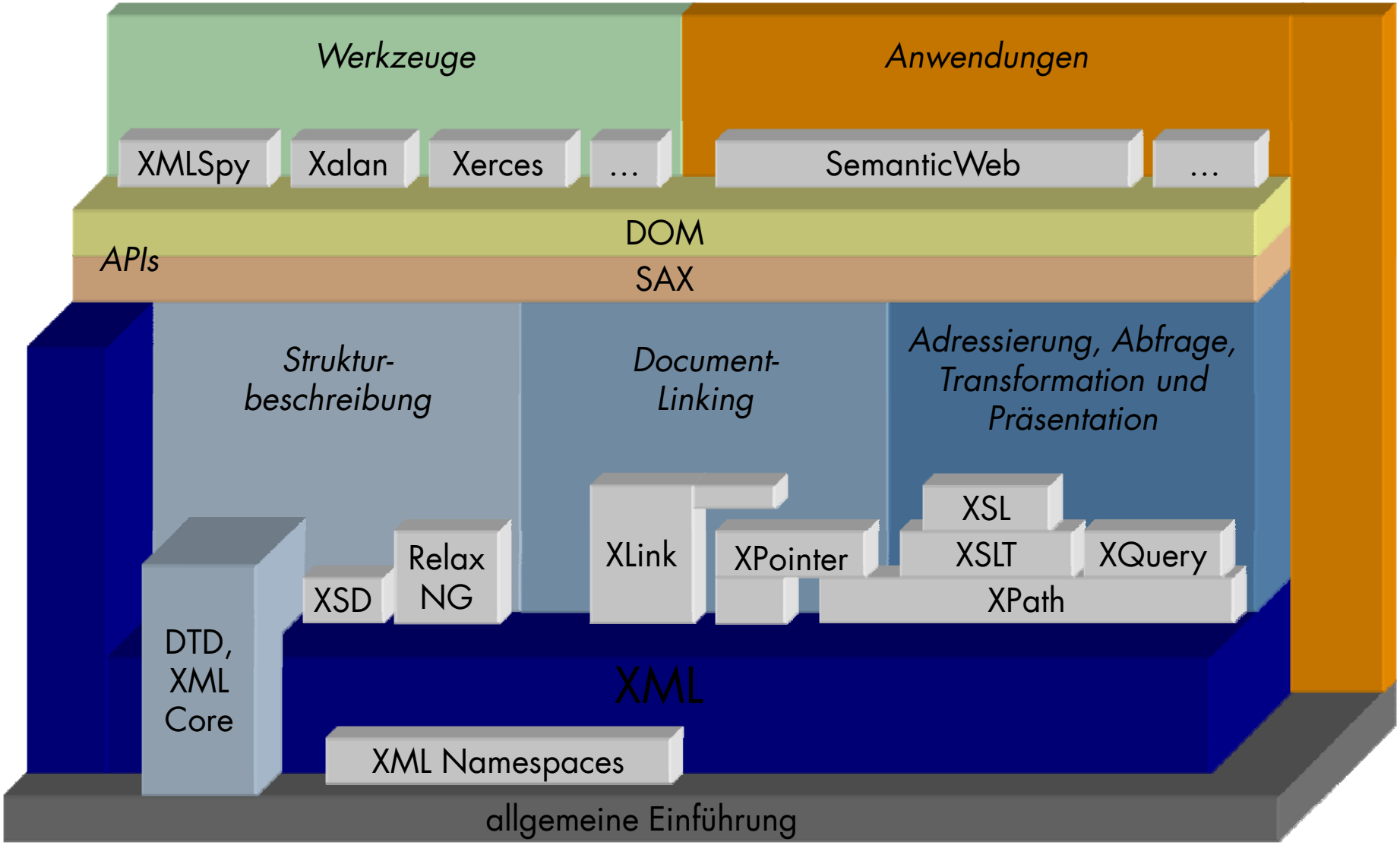
Einführung in XML

Sommersemester 2008

Prof. Dr. Klaus-Peter Fähnrich
Dr. Maik Thränert

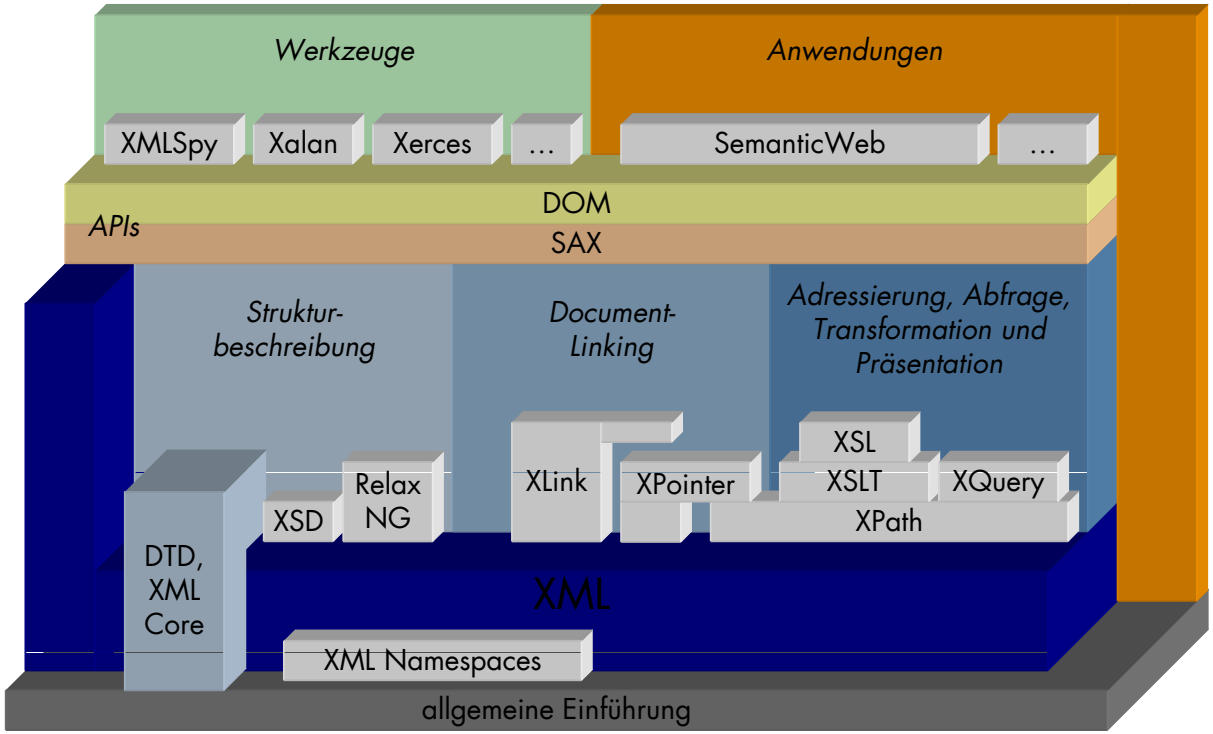
Gliederung der Vorlesung

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



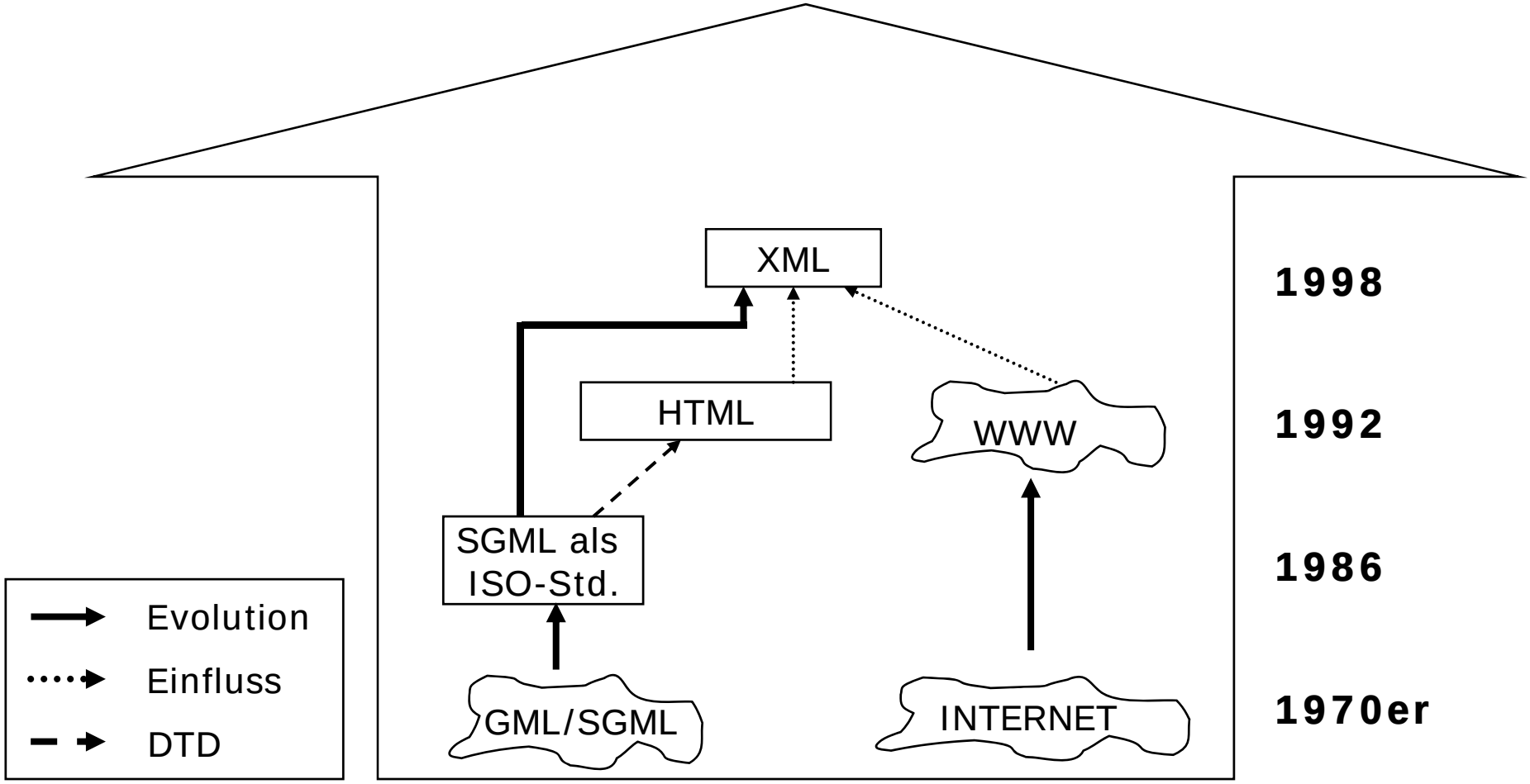
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

- Inhaltsübersicht
  - Allgemeine Einführung
  - XML-Spezifikation
  - XML-Namespaces



2. Hist. Entwicklung von SGML, HTML und XML

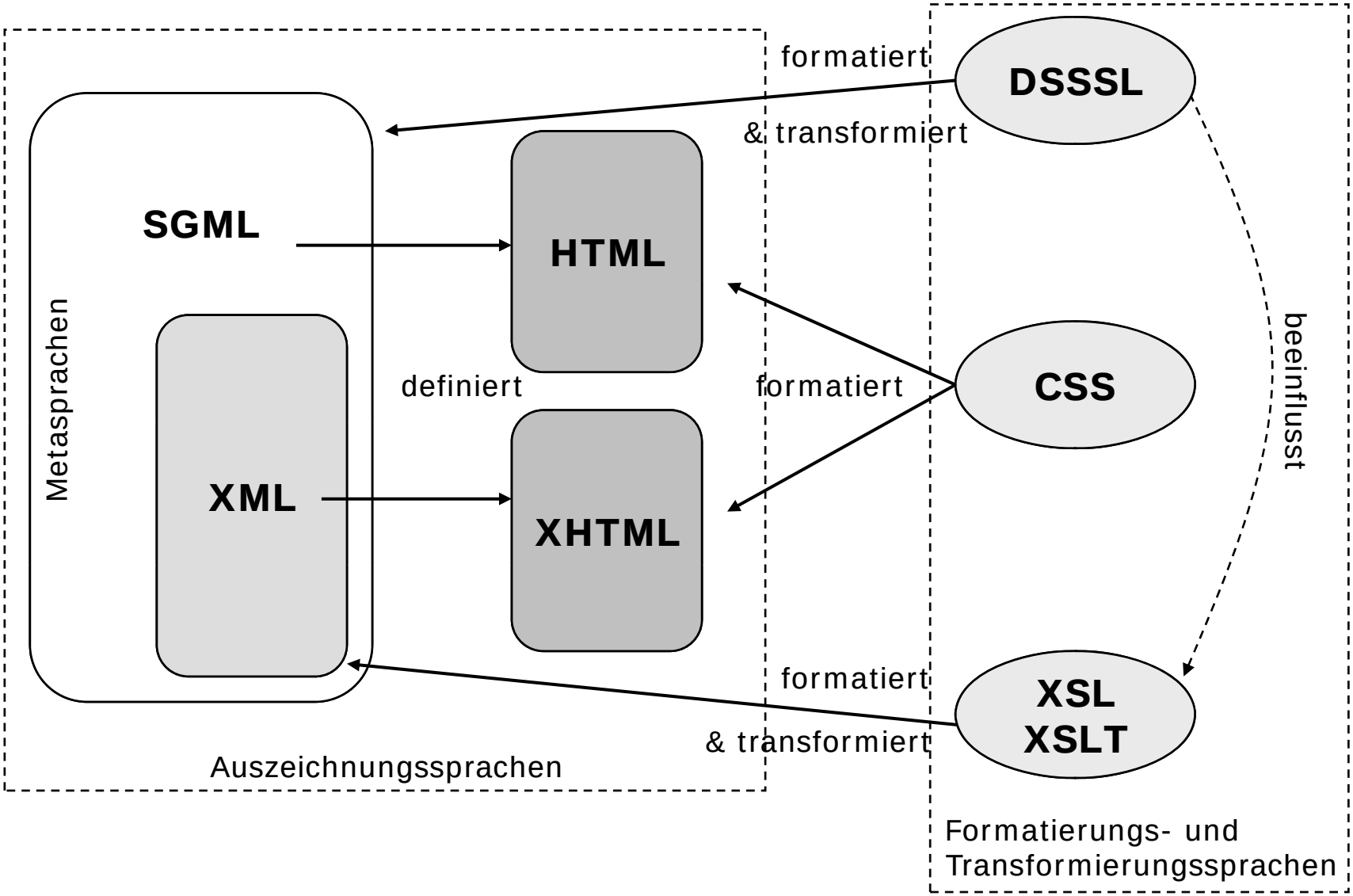
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Entwicklung von XML
Quelle: Merz

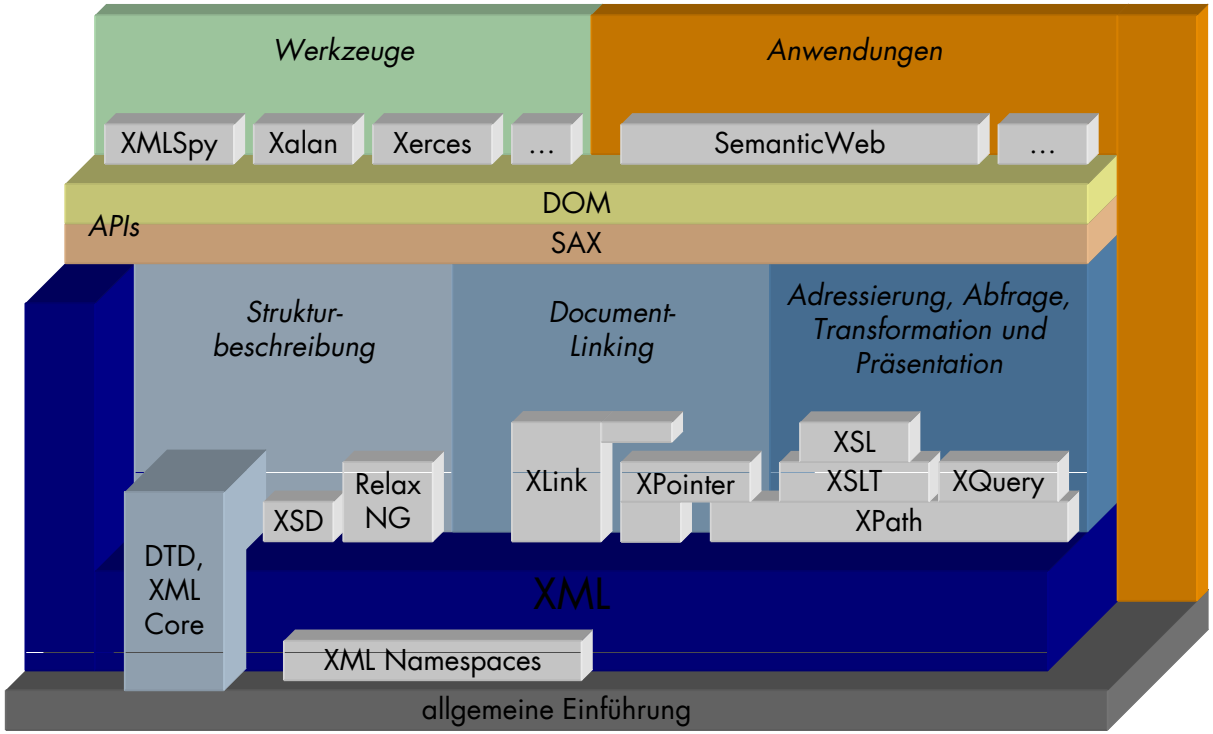
Überblick Markup-Sprachen

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Quelle: Behme/Mintert

- Inhaltsübersicht
  - Allgemeine Einführung
  - XML-Spezifikation
  - XML-Namespaces



3. Bestandteile der XML-Grammatik

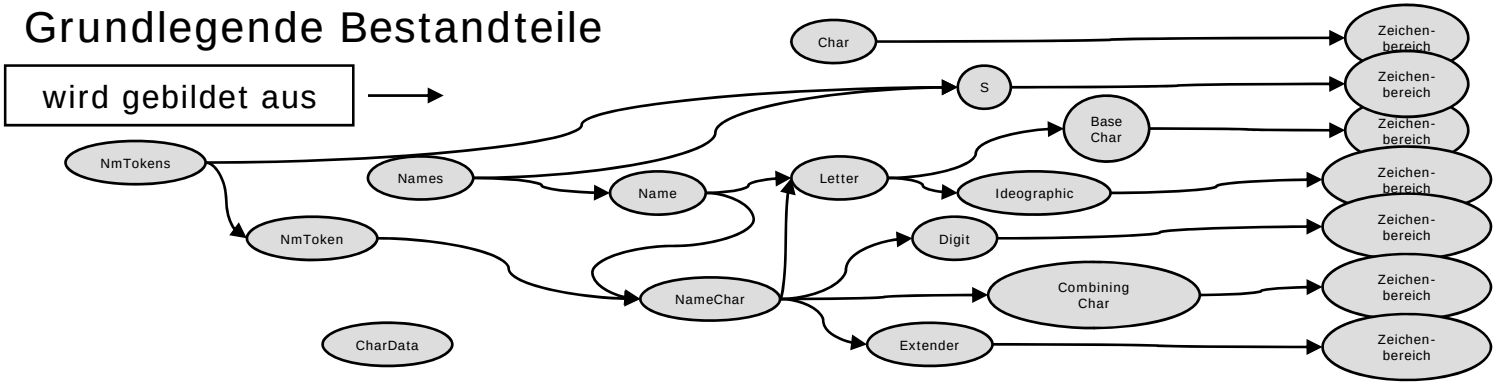
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

XML-Dokumente

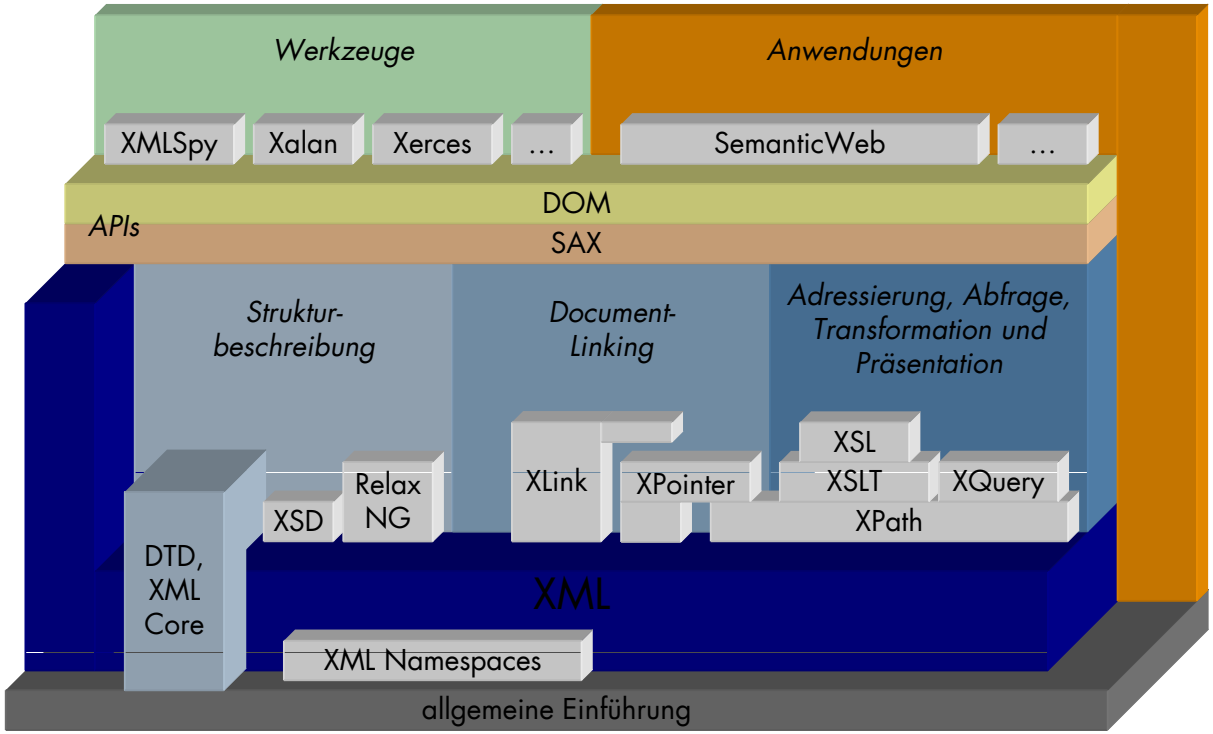
Logische Bestandteile		Physische Bestandteile	
Dokument:	DTD:	Dokument:	DTD:
Comment, PIs, CDATA, Elemente, Attribute	Comment, <!ELEMENT> <!ATTLIST>	Allgemeine Entities	Parameter-Entities, Entity-Deklaration

XML-Grammatik

Grundlegende Bestandteile



- Inhaltsübersicht
  - Allgemeine Einführung
  - XML-Spezifikation
  - XML-Namespaces



7. Namensräume – Begriff

Ein **XML-Namensraum** ist eine Sammlung von Namen, die durch eine URI (Uniform Ressource Identifier)-Referenz identifiziert wird.

Die URI dient nur der Eindeutigkeit des Dokumentherstellers, der Prozessor sucht dort keineswegs nach einem Namensraum.

Durch die Zuordnung von Element- und Attributnamen können Informationseinheiten eindeutig identifiziert und Namenskonflikte vermieden werden.

Damit können gleichlautende Namen (ggf. mit unterschiedlicher Bedeutung) in verschiedenen Vokabularen verwendet werden, ohne dass Kollisionen befürchtet werden müssten.

7. Namensräume – Deklaration

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Bevor in einem XML-Dokument ein Namensraum verwendet werden kann, muss er mittels reservierter Attribute (`xmlns:`) deklariert werden.

Zur Vereinfachung werden Namensräumen meist Präfixe (`NCName`) (beliebig, aber nicht `xml` oder `xmlns`) zugeordnet. Die Gültigkeit des Namensraums bezieht sich auf das Element und seine Kinder, wenn nicht für diese ein anderer Namensraum zugewiesen wurde. Soll der Namensraum im gesamten Dokument gültig sein, muss er in der Wurzel deklariert werden.

```
[1] NSAttName ::= PrefixedAttName | DefaultAttName
[2] PrefixedAttName ::= 'xmlns:' NCName
[3] DefaultAttName ::= 'xmlns'
[4] NCName ::= (Letter | '_') (NCNameChar)*
[5] NCNameChar ::= Letter | Digit | '.' | '-' | '_'|
                   CombiningChar | Extender
```

7. Namensräume – Verwendung generell

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

- Namen aus der XML-Spezifikation (Produktion `Name`) können qualifiziert werden
- Ein **qualifizierter Name** ist ein Name für ein Element oder Attribut, der zugleich seinen Namensraum beinhalten kann. Der Doppelpunkt teilt zwischen Namespace-Präfix und lokalem Namen auf.

```
[6] QName ::= (Prefix ':')? LocalPart
[7] Prefix        ::=    NCName
[8] LocalPart     ::=    NCName
```

- der Namespace-Präfix wurde bei der Namensraum-Deklaration deklariert und damit an eine URI gebunden

7. Namensräume – Verwendung in Elementen

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

- aus der XML-Spezifikation:

```
element ::= EmptyElemTag | STag content ETag
```

- Erweiterung aus der Namespace-Spezifikation:

```
[9]  STag      ::=    '<' QName (S Attribute)* S? '>'
[10] ETag    ::=     '</' QName S? '>'
[11] EmptyElemTag    ::=     '<' QName (S Attribute)* S? '/>'
```

- Beispiel

```
<x xmlns:edi='http://ecommerce.org/schema'>
<!-- the 'price' element's namespace is
http://ecommerce.org/schema -->
  <edi:price units='Euro'>32.18</edi:price>
</x>
```

7. Namensräume – Verwendung in Attributen

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

- aus der XML-Spezifikation:

```
[41] Attribute ::= Name Eq AttValue
```

- Erweiterung aus der Namespace-Spezifikation:

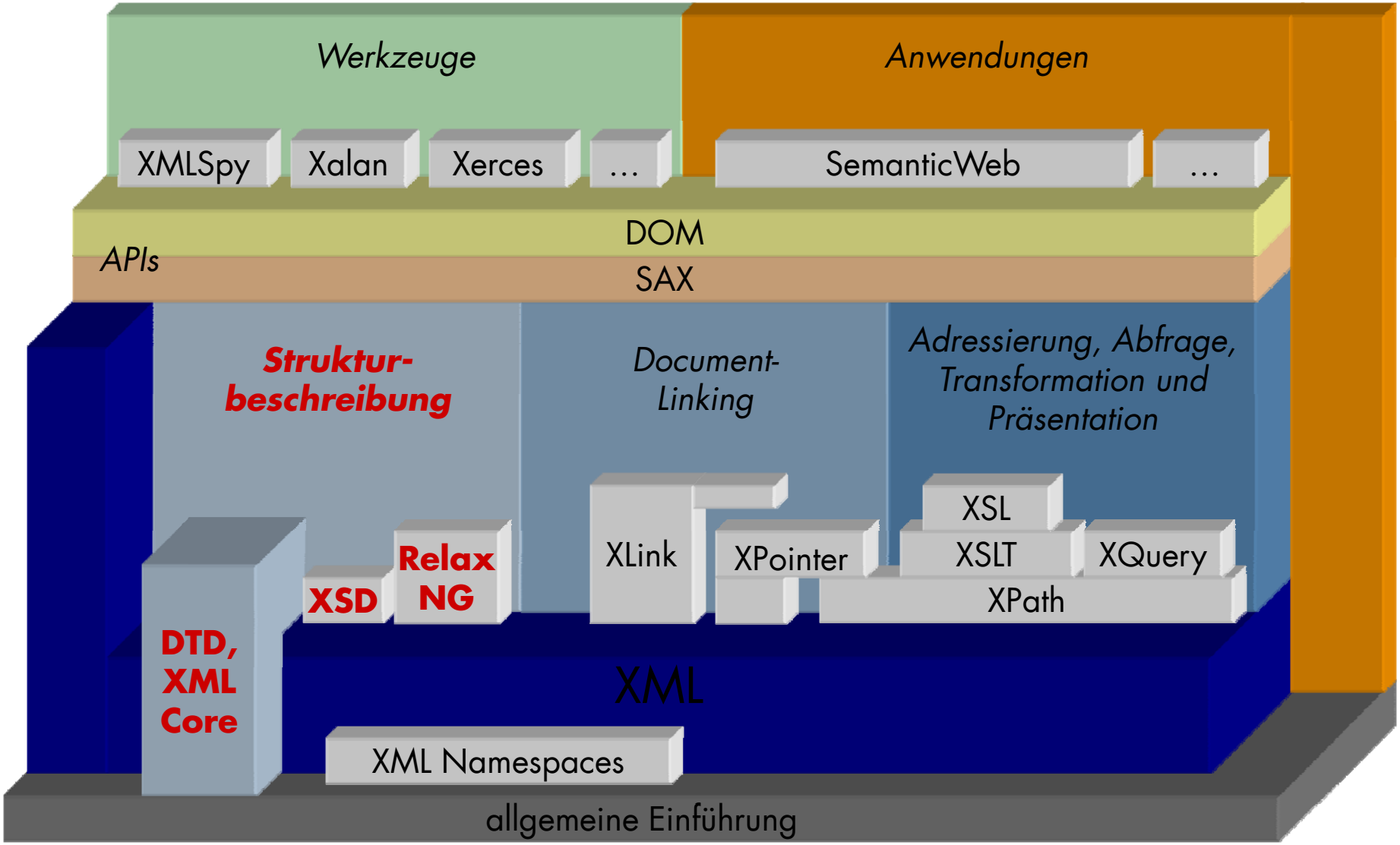
```
[12] Attribute ::= NSAttName Eq AttValue |
                   QName Eq AttValue
```

- Beispiel

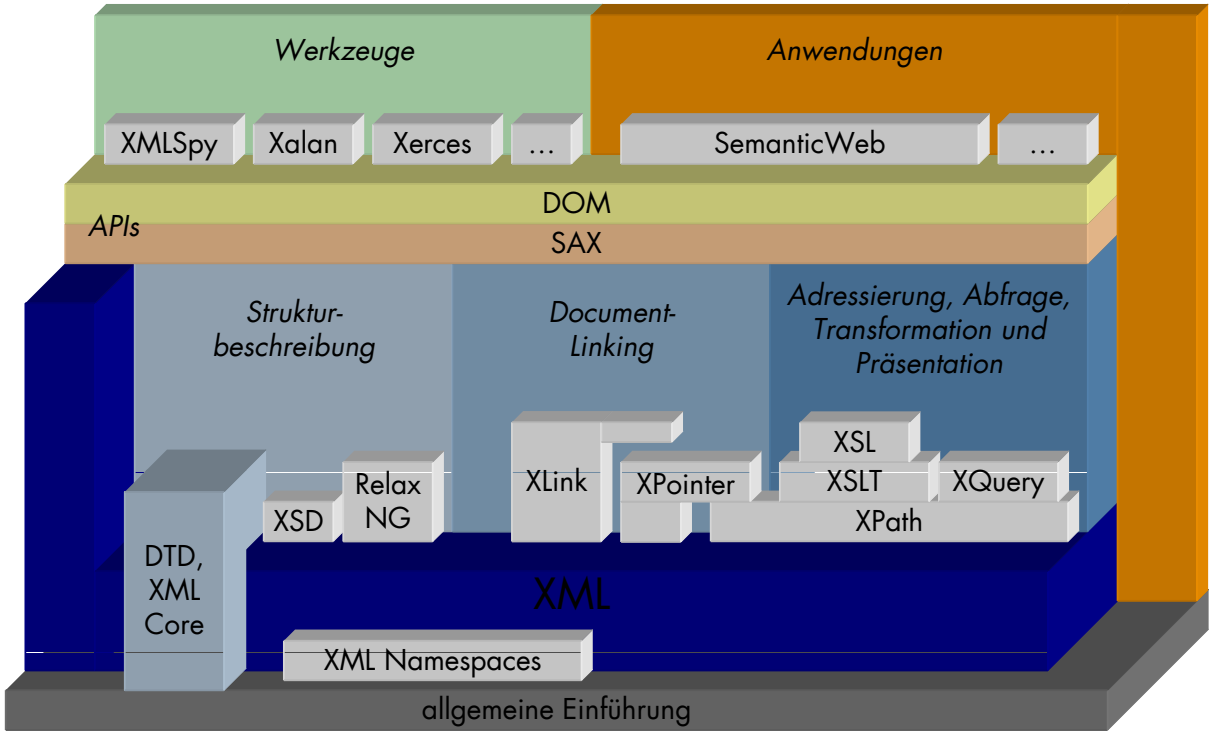
```
<x xmlns:edi='http://ecommerce.org/schema'>
    <!-- the 'taxClass' attribute's namespace is
    http://ecommerce.org/schema -->
    <lineItem edi:taxClass="exempt">Baby food</lineItem>
</x>
```

Gliederung der Vorlesung

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



- Inhaltsübersicht
  - Motivation
  - DTD
  - XML Schema
  - Relax NG



Motivation

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

addressDB.dtd

```
<!ENTITY % anr "(Herr|Frau) #REQUIRED">
<!ELEMENT Adresse (Person, (Strasse)?, Ort)>
<!ELEMENT Person (Vorname+, Name)>
<!ATTLIST Person Anrede %anr;>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Vorname (#PCDATA)>
<!ELEMENT Strasse (#PCDATA)>
<!ATTLIST Strasse Nummer CDATA #IMPLIED>
<!ELEMENT Ort (#PCDATA)>
<!ATTLIST Ort PLZ CDATA #REQUIRED>
```

addressDB.xsd

```
<xs:attribute name="PLZ" use="required">
    <xs:simpleType>
        <xs:restriction base="xs:integer">
            <xs:minInclusive value="00000"/>
            <xs:maxInclusive value="99999"/>
            <xs:totalDigits value="5"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
```

addressDB.rng

```
<rng:choice>
    <rng:element name="Strasse">
        <rng:attribute name="Nummer"/>
        <rng:text/>
    </rng:element>
    <rng:element name="Postfach">
        <rng:text/>
    </rng:element>
</rng:choice>
```

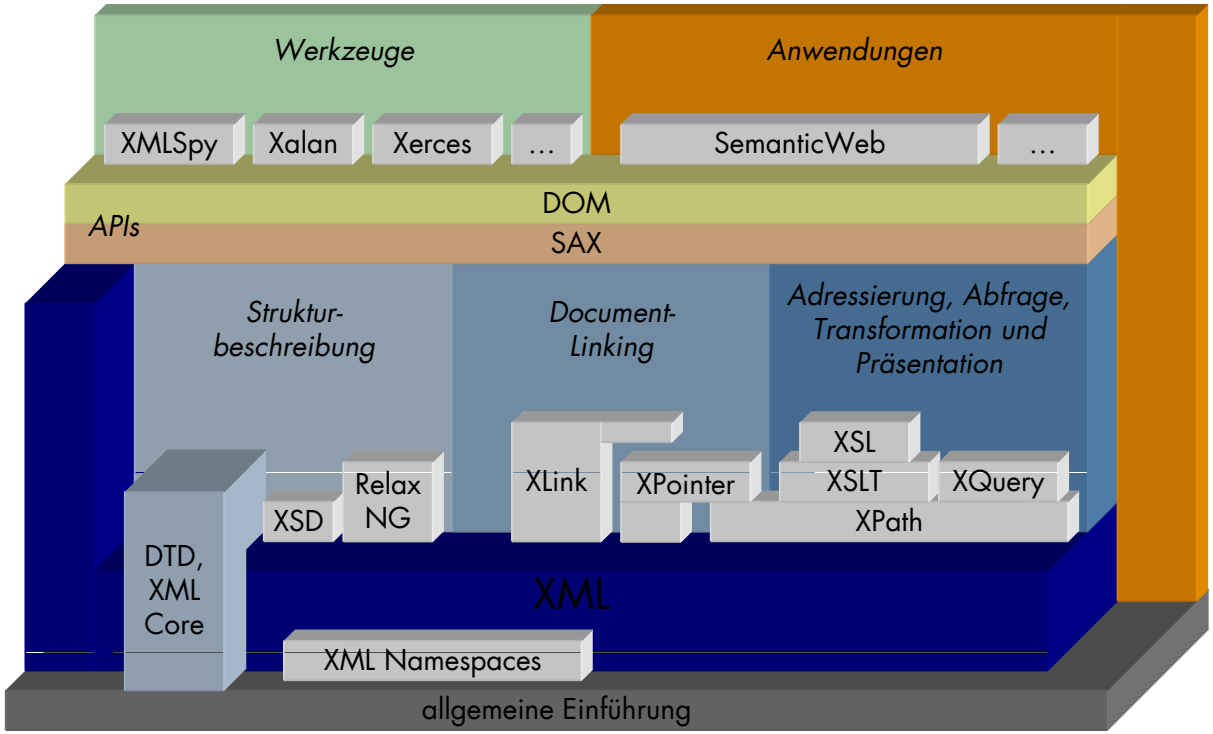
addressDB.xml

```
<Adresse>
    <Person Anrede="Frau">
        <Vorname>Eva</Vorname>
        <Name>Mustermann</Name>
    </Person>
    <Strasse Nummer="4">Beispielstrasse</Strasse>
    <Ort PLZ="54783">Musterstadt</Ort>
</Adresse>
```

← instanz_von

<- - - - - repräsentiert

- Inhaltsübersicht
  - Motivation
  - DTD
  - XML Schema
  - Relax NG

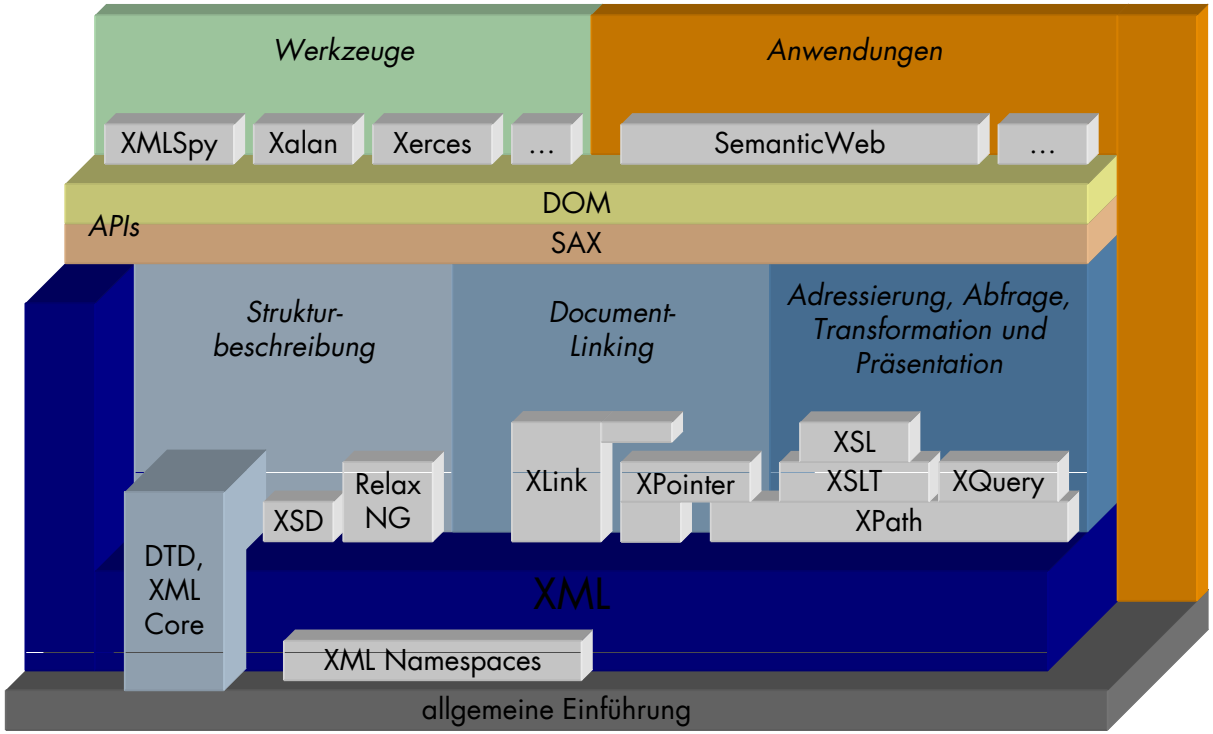


Beispiel


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE AdressDB SYSTEM "adressDB.dtd">
<!--Adress-Datenbank-->
<AdressDB>
    <Adresse>
          <Person Anrede="Frau">
                   <Vorname>Eva</Vorname>
                   <Name>Mustermann</Name>
          </Person>
          <Strasse Nummer="4">Beispielstrasse</Strasse>
          <Ort PLZ="54783">Musterstadt</Ort>
    </Adresse>
    <Adresse>
          <Firma>Universität Leipzig</Firma>
          <Strasse Nummer="10-11">Augustusplatz</Strasse>
          <Ort PLZ="04109">Leipzig</Ort>
    </Adresse>
    <Adresse>
          <Person Anrede="Herr">
                   <Vorname>Hans</Vorname>
                   <Name>Meier</Name>
          </Person>
          <Postfach>123456</Postfach>
          <Ort PLZ="80939"/>
    </Adresse>
</AdressDB>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY % anr "(Herr|Frau) #REQUIRED">
<!ENTITY uni_l "Universität Leipzig">
<!ELEMENT AdressDB (Adresse)*>
<!ELEMENT Adresse ((Firma | Person), (Strasse | Postfach)?,
   Ort)>
<!ELEMENT Firma (#PCDATA)>
<!ELEMENT Person (Vorname+, Name)>
<!ATTLIST Person Anrede %anr;>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Vorname (#PCDATA)>
<!ELEMENT Strasse (#PCDATA)>
<!ATTLIST Strasse Nummer CDATA #IMPLIED>
<!ELEMENT Postfach (#PCDATA)>
<!ELEMENT Ort (#PCDATA)>
<!ATTLIST Ort PLZ CDATA #REQUIRED>
```

- Inhaltsübersicht
  - Motivation
  - DTD
  - XML Schema
  - Relax NG



Beispiel


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE AdressDB SYSTEM "adressDB.dtd">
<AdressDB>
    <Adresse>
        <Person Anrede="Frau">
            <Vorname>Eva</Vorname>
            <Name>Mustermann</Name>
        </Person>
        <Strasse Nummer="4">Beispielstrasse</Strasse>
        <Ort PLZ="547">Musterstadt</Ort>
        <Telefon>00499991234567</Telefon>
        <eMail>eva.mustermann</eMail>
    </Adresse>
    <Adresse>
        <Firma>Universität Leipzig</Firma>
        <Strasse Nummer="10-11">Augustusplatz</Strasse>
        <Ort PLZ="04109">Leipzig</Ort>
        <Telefon>034197108</Telefon>
        <eMail>www.uni-leipzig.de</eMail>
    </Adresse>
    <Adresse>
        <Person Anrede="Herr">
            <Vorname>Hans</Vorname>
            <Name>Meier</Name>
        </Person>
        <Postfach>123456</Postfach>
        <Ort PLZ="D-80950"/>
        <Telefon>0899874563</Telefon>
        <eMail>hans(at)hans.meier.de</eMail>
    </Adresse>
</AdressDB>
```

Dieses XML-Dokument ist nach unserer DTD gültig. Aber wie jeder sieht, sind nicht alle Werte sinnvoll oder aber einige Werte kann man in verschiedenen Arten angeben.

Die Werte für eine Telefonnummer kann man in vielen Formen angeben.
Mit oder ohne (internationaler) Vorwahl, mit oder ohne Trennung der Nebenstellen usw.

Viele andere Elemente müssen ebenso einem bestimmten Format entsprechen, um sie elektronisch weiter zu verarbeiten, z. B. PLZ, eMail-Adressen usw.

Beispiel

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE AdressDB SYSTEM "adressDB.dtd">
<AdressDB>
    <Adresse>
            <Person Anrede="Frau">
                        <Vorname>Eva</Vorname>
                        <Name>Mustermann</Name>
            </Person>
    </Adres
    <Adress
    </Adres
    <Adress
            <Ort PLZ="D-80950"/>
            <Telefon>0899874563</Telefon>
            <eMail>hans(at)hans.meier.de</eMail>
    </Adresse>
</AdressDB>
```

Dieses XML-Dokument ist nach unserer DTD gültig. Aber wie jeder sieht, sind nicht alle Werte sinnvoll oder aber einige Werte kann man in verschiedenen Arten angeben.

All diese Vorgaben lassen sich mit DTDs nicht realisieren. Aus diesem Grund wurden verschiedene XML-Schemabeschreibungs-sprachen entwickelt (z. B. XDR von Microsoft). Zur Recommendation beim W3C hat es jedoch nur XML Schema gebracht.

Eine weitere standardisierte Schemasprache ist Relax NG von OASIS.

ne
kann man in
ngeben.
nternationaler)
er ohne
ebenstellen usw.

mente müssen
estimmten
Format entsprechen, um sie elektronisch weiter zu verarbeiten, z. B. PLZ, eMail-Adressen usw.

XML Schema

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
   <xs:element name="AdressDB">
      <xs:complexType>
         <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="Adresse" type="AdresseType"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:complexType name="AdresseType">
      <xs:sequence>
         <xs:choice>
            <xs:element ref="Firma"/>
            <xs:element name="Person" type="PersonType"/>
         </xs:choice>
         <xs:choice minOccurs="0">
            <xs:element name="Strasse" minOccurs="0">
               <xs:complexType>
                  <xs:simpleContent>
                     <xs:restriction base="StrasseType">
                        <xs:attribute name="Nummer">
                           <xs:simpleType>
                              <xs:restriction base="xs:string">
                                 <xs:pattern value="\d{1,3}[a-z]?"/>
                                 <xs:pattern value="\d{1,3}[a-z]?-\d{1,3}[a-z]?"/>
                              </xs:restriction>
                           </xs:simpleType>
                        </xs:attribute>
                     </xs:restriction>
                  </xs:simpleContent>
               </xs:complexType>
            </xs:element>
```

```
            <xs:element name="Postfach" minOccurs="0">
               <xs:simpleType>
                  <xs:restriction base="xs:integer">
                     <xs:totalDigits value="10"/>
                  </xs:restriction>
               </xs:simpleType>
            </xs:element>
         </xs:choice>
         <xs:element name="Ort" type="OrtType"/>
         <xs:element name="Telefon">
            <xs:simpleType>
               <xs:restriction base="xs:string">
                  <xs:pattern value="\+\d{1,3}-\(0\)\d{2,6}-\d{3,9}"/>
                  <xs:pattern value="\(0\d{2,6}\)\d{3,9}"/>
                  <xs:pattern value="\(0\d{2,6}\)\d{2,6}-\d{1,5}"/>
               </xs:restriction>
            </xs:simpleType>
         </xs:element>
         <xs:element name="eMail">
            <xs:simpleType>
               <xs:restriction base="xs:string">
                  <xs:pattern value=".+@.+\..{2,}"/>
               </xs:restriction>
            </xs:simpleType>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
   <xs:element name="Firma" type="xs:string"/>
   <xs:element name="Name" type="xs:string"/>
```

```
<xs:complexType name="OrtType">
   <xs:simpleContent>
      <xs:extension base="xs:string">
         <xs:attribute name="PLZ" use="required">
            <xs:simpleType>
               <xs:restriction base="xs:integer">
                  <xs:minInclusive value="00000"/>
                  <xs:maxInclusive value="99999"/>
                  <xs:totalDigits value="5"/>
               </xs:restriction>
            </xs:simpleType>
         </xs:attribute>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
<xs:complexType name="PersonType">
   <xs:sequence>
      <xs:element ref="Vorname" maxOccurs="unbounded"/>
      <xs:element ref="Name"/>
   </xs:sequence>
   <xs:attribute name="Anrede" use="required">
      <xs:simpleType>
         <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="Herr"/>
            <xs:enumeration value="Frau"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:attribute>
</xs:complexType>
```

```
   <xs:complexType name="StrasseType">
      <xs:simpleContent>
         <xs:extension base="xs:string">
            <xs:attribute name="Nummer" use="required">
               <xs:simpleType>
                  <xs:restriction base="xs:string">
                     <xs:pattern value="\d{1,3}"/>
                  </xs:restriction>
               </xs:simpleType>
            </xs:attribute>
         </xs:extension>
      </xs:simpleContent>
   </xs:complexType>
   <xs:element name="Vorname" type="xs:string"/>
</xs:schema>
```

XML Schema

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://bis.uni-leipzig.de/Adresses" xmlns="http://bis.uni-
leipzig.de/Adresses" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
   <xs:element name="AdressDB">
      <xs:complexType>
         <xs:sequence minOc
            <xs:element nam
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:complexType name="AdresseType">
      <xs:sequence>
         <xs:choice>
            <xs:element ref="Firma"/>
            <xs:element name="Person" type="PersonType"/>
         </xs:choice>
         <xs:choice minOccurs="0">
            <xs:element name="Strasse" minOccurs="0">
               <xs:complexType>
                  <xs:simpleContent>
                     <xs:restriction base="StrasseType">
                        <xs:attribute name="Nummer">
                           <xs:simpleType>
                              <xs:restriction base="xs:string">
                                 <xs:pattern value="\d{1,3}[a-z]?"/>
                                 <xs:pattern value="\d{1,3}[a-z]?-\d{1,3}[a-z]?"/>
                              </xs:restriction>
                           </xs:simpleType>
                        </xs:attribute>
                     </xs:restriction>
                  </xs:simpleContent>
               </xs:complexType>
            </xs:element>
```

```
<?xml version="1.0" encoding="UTF-8"?>
```

Zuerst geben wir an, dass es sich bei dem nachfolgenden Schema um ein XML-Dokument handelt.

XML Schema

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://bis.uni-leipzig.de/Adresses" xmlns="http://bis.uni-
leipzig.de/Adresses" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
   <xs:element name="AdressDB">
      <xs:complexType>
```

```
<xs:schema targetNamespace="http://bis.uni-leipzig.de/Adresses"
xmlns="http://bis.uni-leipzig.de/Adresses"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
```

Dann folgt unser Dokumentelement "xs:schema", das die gesamte Schemadeklaration enthält. Der Zielnamensraum entspricht dem Standardnamensraum. `elementFormDefault="qualified"` gibt an, dass die Elemente standardmäßig "namensraumqualifiziert" sind.

```
         <xs:complexType>
            <xs:simpleContent>
               <xs:restriction base="StrasseType">
                  <xs:attribute name="Nummer">
                     <xs:simpleType>
                        <xs:restriction base="xs:string">
                           <xs:pattern value="\d{1,3}[a-z]?"/>
                           <xs:pattern value="\d{1,3}[a-z]?-\d{1,3}[a-z]?"/>
                        </xs:restriction>
                     </xs:simpleType>
                  </xs:attribute>
               </xs:restriction>
            </xs:simpleContent>
         </xs:complexType>
      </xs:element>
```

XML Schema

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://bis.uni-leipzig.de/Adresses" xmlns="http://bis.uni-
leipzig.de/Adresses" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
   <xs:element name="AdressDB">
      <xs:complexType>
         <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="Adresse" type="AdresseType"/>
         </xs:sequence>
      </xs:com
   </xs:elemen
   <xs:complex
      <xs:sequ
         <xs:c
            <
            <
         </xs
         <xs:choice minOccurs="0">
            <xs:element name="Strasse" minOccurs="0">
               <xs:complexType>
                  <xs:simpleContent>
                     <xs:restriction base="StrasseType">
                        <xs:attribute name="Nummer">
                           <xs:simpleType>
                              <xs:restriction base="xs:string">
                                 <xs:pattern value="\d{1,3}[a-z]?"/>
                                 <xs:pattern value="\d{1,3}[a-z]?-\d{1,3}[a-z]?"/>
                              </xs:restriction>
                           </xs:simpleType>
                        </xs:attribute>
                     </xs:restriction>
                  </xs:simpleContent>
               </xs:complexType>
            </xs:element>
```

```
<xs:element name="AdressDB">
```

Mit der ersten Elementdeklaration deklarieren wir das Dokumentelement des späteren Instanzdokumentes. Es wird alle anderen Elemente enthalten und trägt den Namen "AdressDB".

XML Schema

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://bis.uni-leipzig.de/Adresses" xmlns="http://bis.uni-
leipzig.de/Adresses" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
   <xs:element name="AdressDB">
      <xs:complexType>
         <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="Adresse" type="AdresseType"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:com
      <xs:
```

```
<xs:complexType>
         <xs:sequence minOccurs="0" maxOccurs="unbounded">
                  <xs:element name="Adresse" type="AdresseType"/>
```

"AdressDB" hat als Inhaltsmodell den komplexen Typ, da es der Container für die einzelnen Adresse-Elemente ist. Das Adresse-Element hat den Typ "AdresseType", welcher ein selbst definierter komplexer Datentyp ist.

```
            <xs:complexType>
               <xs:simpleContent>
                  <xs:restriction base="StrasseType">
                     <xs:attribute name="Nummer">
                        <xs:simpleType>
                           <xs:restriction base="xs:string">
                              <xs:pattern value="\d{1,3}[a-z]?"/>
                              <xs:pattern value="\d{1,3}[a-z]?-\d{1,3}[a-z]?"/>
                           </xs:restriction>
                        </xs:simpleType>
                     </xs:attribute>
                  </xs:restriction>
               </xs:simpleContent>
            </xs:complexType>
         </xs:element>
```

XML Schema

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://bis.uni-leipzig.de/Adresses" xmlns="http://bis.uni-
leip                                                                                  ualified">
    <xs:complexType name="AdresseType">
```

Im nächsten Schritt muss der Datentyp "AdresseType" definiert werden, da er dem Adresse-Element zugewiesen wird. In unserem Fall wird dieser Typ global definiert.

```
    </xs:element>
    <xs:complexType name="AdresseType">
        <xs:sequence>
            <xs:choice>
                <xs:element ref="Firma"/>
                <xs:element name="Person" type="PersonType"/>
            </xs:choice>
            <xs:choice minOccurs="0">
                <xs:element name="Strasse" minOccurs="0">
                    <xs:complexType>
                        <xs:simpleContent>
                            <xs:restriction base="StrasseType">
                                <xs:attribute name="Nummer">
                                    <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                            <xs:pattern value="\d{1,3}[a-z]?"/>
                                            <xs:pattern value="\d{1,3}[a-z]?-\d{1,3}[a-z]?"/>
                                        </xs:restriction>
                                    </xs:simpleType>
                                </xs:attribute>
                            </xs:restriction>
                        </xs:simpleContent>
                    </xs:complexType>
                </xs:element>
```

AdresseType

XML Schema

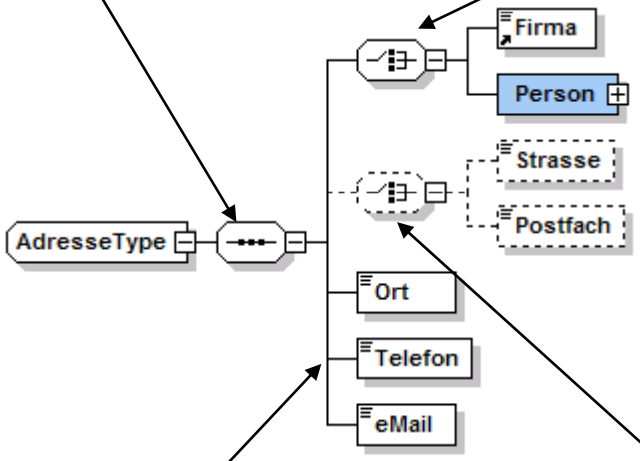
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Besteht aus einer Sequenz:

```
<xs:complexType name="AdresseType">
                    <xs:sequence>
```

Von einer Auswahl:

```
<xs:choice>
          <xs:element ref="Firma"/>
          <xs:element name="Person"
                    type="PersonType"/>
</xs:choice>
```



Einer zweiten, jedoch optionalen Auswahl:

```
<xs:choice minOccurs="0">
          <xs:element name="Strasse" minOccurs="0">
```

Und drei "einfachen" Elementen:

```
<xs:element name="Ort" type="OrtType"/>
<xs:element name="Telefon">...</xs:element>
<xs:element name="eMail">...</xs:element>
```

XML Schema

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

AdresseType

```
        <xs:element name="Postfach" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:integer">
              <xs:totalDigits value="10"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:choice>
```

```
<xs:element name="Postfach" minOccurs="0">
          <xs:simpleType>
                    <xs:restriction base="xs:integer">
                              <xs:totalDigits value="10"/>
                    </xs:restriction>
          </xs:simpleType>
</xs:element>
```

Der Typ für das optionale Element Postfach wird als anonyme Typdefinition von Integer abgeleitet.

```
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value=".+@.+\..{2,}"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Firma" type="xs:string"/>
  <xs:element name="Name" type="xs:string"/>
```

XML Schema

AdresseType

```
        <xs:element name="Postfach" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:integer">
              <xs:totalDigits value="10"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:choice>
      <xs:element name="Ort" type="OrtType"/>
      <xs:element name="Telefon">
        <xs:simpleType>
```

```
<xs:element name="Ort" type="OrtType"/>
```

Ort bekommt wiederum einen globalen Typen "OrtType" zugewiesen, welcher später definiert wird.

```
        </xs:simpleType>
      </xs:element>
      <xs:element name="eMail">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value=".+@.+\..{2,}"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Firma" type="xs:string"/>
  <xs:element name="Name" type="xs:string"/>
```

XML Schema

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

AdresseType

```
<xs:element name="Postfach" minOccurs="0">
   <xs:simpleType>
      <xs:restriction base="xs:integer">
         <xs:totalDigits value="10"/>
      </xs:restriction>
   </xs:simpleType>
</xs:element>
</xs:choice>
<xs:element name="Ort" type="OrtType"/>
<xs:element name="Telefon">
   <xs:simpleType>
      <xs:restriction base="xs:string">
         <xs:pattern value="\+\d{1,3}-\(0\)\d{2,6}-\d{3,9}"/>
         <xs:pattern value="\(0\d{2,6}\)\d{3,9}"/>
         <xs:pattern value="\(0\d{2,6}\)\d{2,6}-\d{1,5}"/>
```

```
<xs:element name="Telefon">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:pattern value="\+\d{1,3}-\(0\)\d{2,6}-\d{3,9}"/>
            <xs:pattern value="\(0\d{2,6}\)\d{3,9}"/>
            <xs:pattern value="\(0\d{2,6}\)\d{2,6}-\d{1,5}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

Für die Telefonnummer greifen wir wieder zur anonymen Typdefinition, welche mittels regulärer Ausdrücke von "xs:string" abgeleitet wird.

```
</xs
<xs
<xs:element name="Name" type="xs:string"/>
```

XML Schema

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

OrtType

```
<xs:complexType name="OrtType">
   <xs:simpleContent>
      <xs:extension base="xs:string">
         <xs:attribute name="PLZ" use="required">
            <xs:simpleType>
               <xs:restriction base="xs:integer">
                  <xs:minInclusive value="00000"/>
                  <xs:maxInclusive value="99999"/>
                  <xs:totalDigits value="5"/>
               </xs:restriction>
            </xs:simpleType>
         </xs:attribute>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
<xs:complexType name="PersonType">
   <xs:sequence>
      <xs:element ref="Vorname" maxOccurs="unbounded"/>
```

Im oberen Abschnitt wird unser OrtType definiert, als Element vom Typ "xs:string" mit einem Attribut für die Postleitzahl.

```
      <xs:simpleType>
         <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="Herr"/>
            <xs:enumeration value="Frau"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:attribute>
</xs:complexType>
```

XML Schema

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

```
<xs:complexType name="OrtType">
   <xs:simpleContent>
      <xs:extension base="xs:string">
         <xs:attribute name="PLZ" use="required">
            <xs:simpleType>
               <xs:restriction base="xs:integer">
```

Im unteren Abschnitt finden wir die Definition des PersonTypes, welcher im AdressType zur Deklaration des Person-Elementes Verwendung findet.

```
         </xs:attribute>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

PersonType

```
<xs:complexType name="PersonType">
   <xs:sequence>
      <xs:element ref="Vorname" maxOccurs="unbounded"/>
      <xs:element ref="Name"/>
   </xs:sequence>
   <xs:attribute name="Anrede" use="required">
      <xs:simpleType>
         <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="Herr"/>
            <xs:enumeration value="Frau"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:attribute>
</xs:complexType>
```

XML Schema

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

OrtType

```
  <xs:complexType name="StrasseType">
     <xs:simpleContent>
        <xs:extension base="xs:string">
           <xs:attribute name="Nummer" use="required">
              <xs:simpleType>
                 <xs:restriction base="xs:string">
                    <xs:pattern value="\d{1,3}"/>
                 </xs:restriction>
              </xs:simpleType>
           </xs:attribute>
        </xs:extension>
     </xs:simpleContent>
  </xs:complexType>
  <xs:element name="Vorname" type="xs:string"/>
</xs:schema>
```

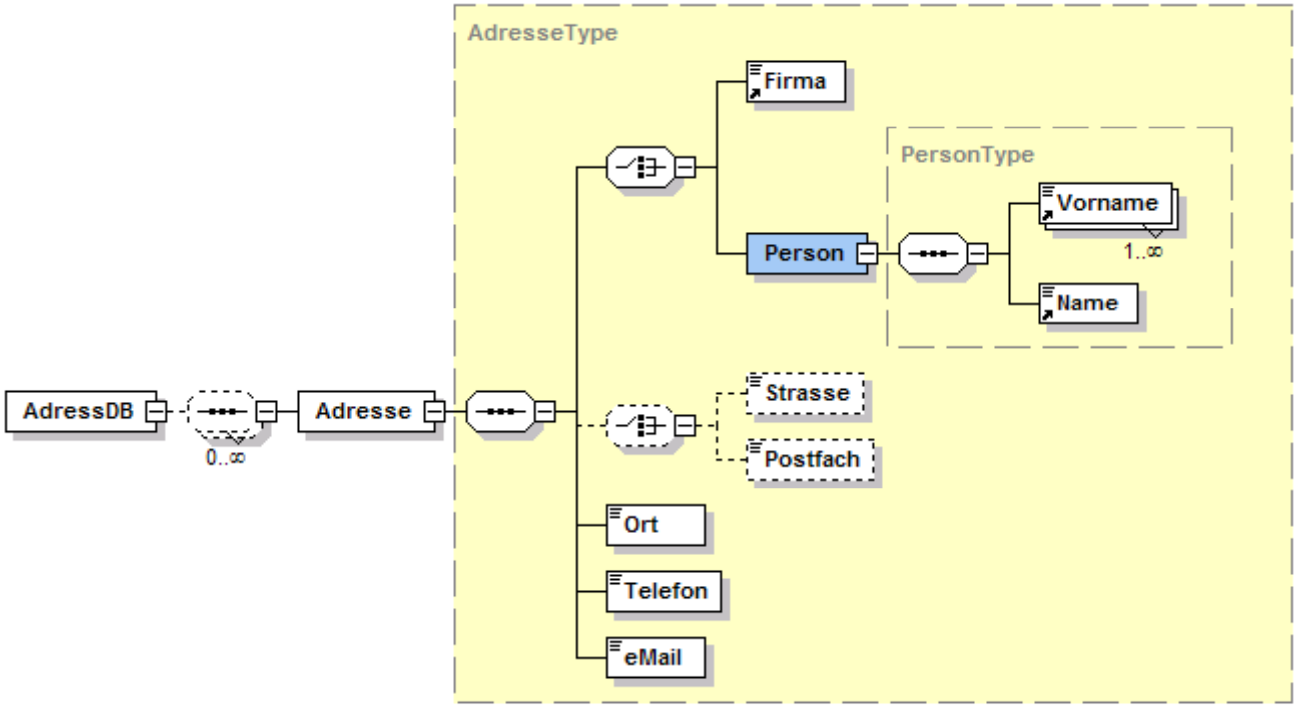
Als Vorletztes wird analog zum OrtType noch ein StrasseType definiert, wiederum mit einem Element und zugehörigem Attribut.

```
   <xs:complexType name="StrasseType">
      <xs:simpleContent>
         <xs:extension base="xs:string">
            <xs:attribute name="Nummer" use="required">
               <xs:simpleType>
                  <xs:restriction base="xs:string">
                     <xs:pattern value="\d{1,3}"/>
                  </xs:restriction>
               </xs:simpleType>
            </xs:attribute>
         </xs:extension>
      </xs:simpleContent>
   </xs:complexType>
   <xs:element name="Vorname" type="xs:string"/>
</xs:schema>
```

```
<xs:element name="Vorname" type="xs:string"/>
```

Zu guter Letzt wird noch einmal ein einfaches Element deklariert.

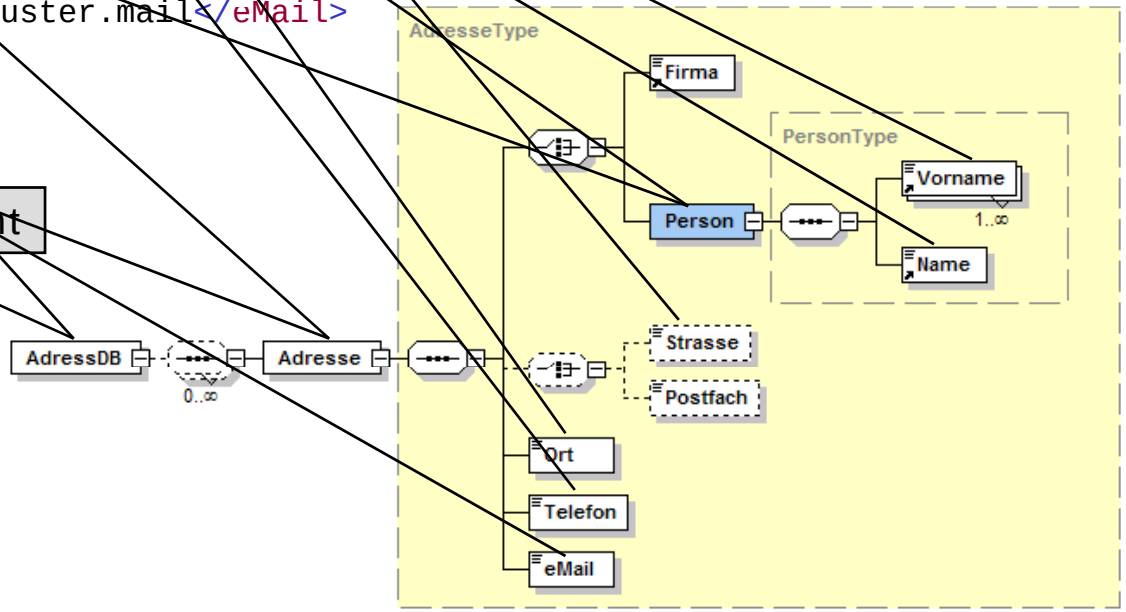
Graphische Repräsentation des einleitenden Beispiels



XML Schema

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

```
<?xml version="1.0" encoding="UTF-8"?>
<AdressDB xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://bis.uni-leipzig.de/Adresses adressdb.xsd"
  xmlns="http://bis.uni-leipzig.de/Adresses">
  <Adresse>
      <Person Anrede="Frau">
            <Vorname>Eva</Vorname>
            <Name>Mustermann</Name>
      </Person>
      <Strasse Nummer="4">Beispielstrasse</Strasse>
      <Ort PLZ="54783">Musterstadt</Ort>
      <Telefon>+49-(0)999-1234567</Telefon>
      <eMail>eva.mustermann@muster.mail</eMail>
  </Adresse>
</AdressDB>
```

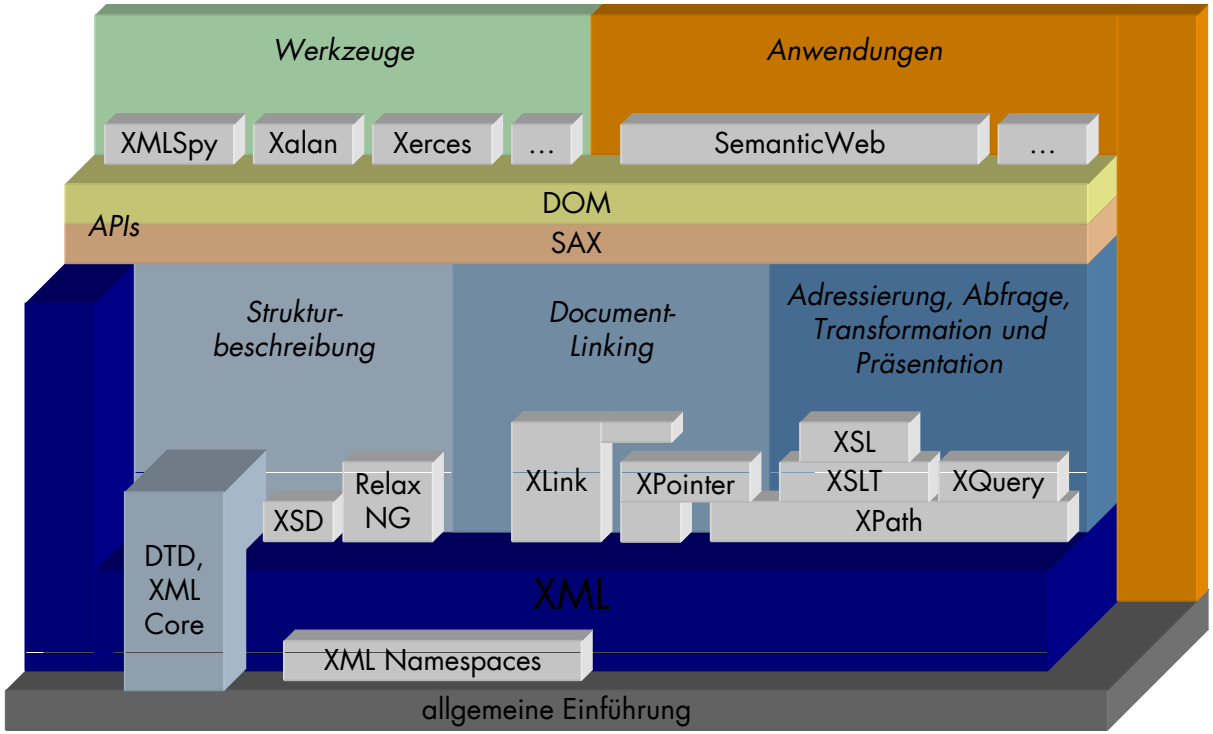


Valides Dokument zum Schema

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

```
<?xml version="1.0" encoding="UTF-8"?>
<AdressDB  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="addressDB.xsd">
    <Adresse>
           <Person Anrede–"Frau">
                      <Vorname>Eva</Vorname>
                      <Name>Mustermann</Name>
           </Person>
           <Strasse Nummer="4">Beispielstrasse</Strasse>
           <Ort PLZ="54783">Musterstadt</Ort>
           <Telefon>+49-(0)999-1234567</Telefon>
           <eMail>eva.mustermann@muster.mail</eMail>
    </Adresse>
    <Adresse>
           <Firma>Universität Leipzig</Firma>
           <Strasse Nummer="10-11">Augustusplatz</Strasse>
           <Ort PLZ="04109">Leipzig</Ort>
           <Telefon>(0341)97-108</Telefon>
           <eMail>oeffentlichkeitsarbeit@uni-leipzig.de</eMail>
    </Adresse>
    <Adresse>
           <Person Anrede="Herr">
                      <Vorname>Hans</Vorname>
                      <Name>Meier</Name>
           </Person>
           <Postfach>123456</Postfach>
           <Ort PLZ="80939"/>
           <Telefon>(089)9874563</Telefon>
           <eMail>hans@hans.meier.de</eMail>
    </Adresse>
</AdressDB>
```

- Inhaltsübersicht
  - Motivation
  - DTD
  - XML Schema
  - Relax NG



```
<element xmlns="http://relaxng.org/ns/structure/1.0"
   name="AdressDB">
   <oneOrMore>
      <element name="Adresse">
         <choice>
            <element name="Person">
               <attribute name="Anrede"/>
               <element name="Vorname">
                  <text/>
               </element>
               <element name="Name">
                  <text/>
               </element>
            </element>
            <element name="Firma">
               <text/>
            </element>
         </choice>
```

```
            <choice>
                <element name="Strasse">
                    <attribute name="Nummer"/>
                    <text/>
                </element>
                <element name="Postfach">
                    <text/>
                </element>
            </choice>
            <element name="Ort">
                <attribute name="PLZ"/>
                <text/>
            </element>
        </element>
    </oneOrMore>
</element>
```