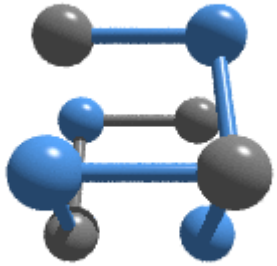


Präsenzveranstaltung zur E-Learning-Veranstaltung

Einführung in XML

Sommersemester 2009

Prof. Dr. Klaus-Peter Fähnrich
Heiko Kern

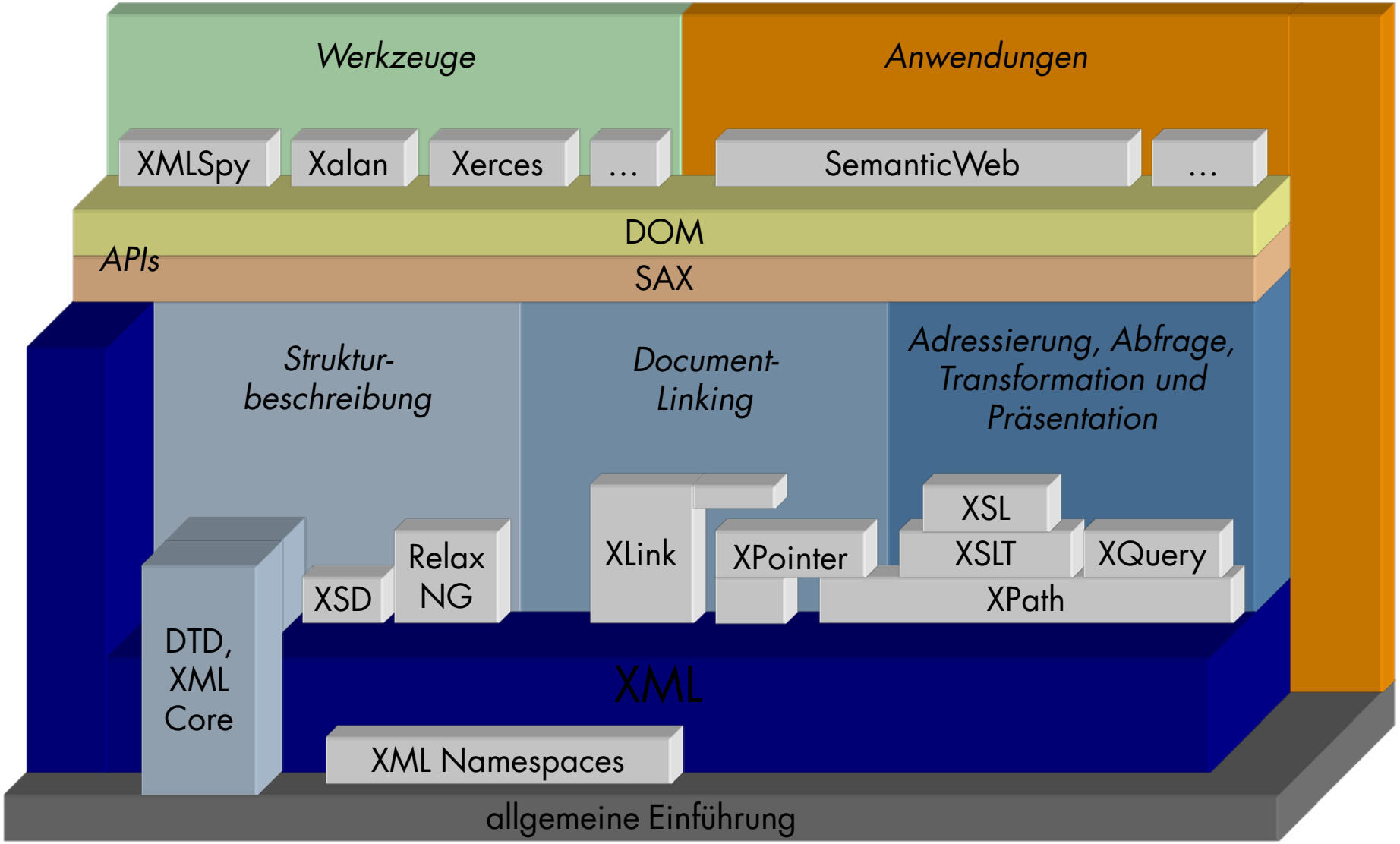
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Agenda

- **Kurzzusammenfassung der Einführung**
- Kurzzusammenfassung der Strukturbeschreibungen
- Diskussion
- Weiterer Ablauf

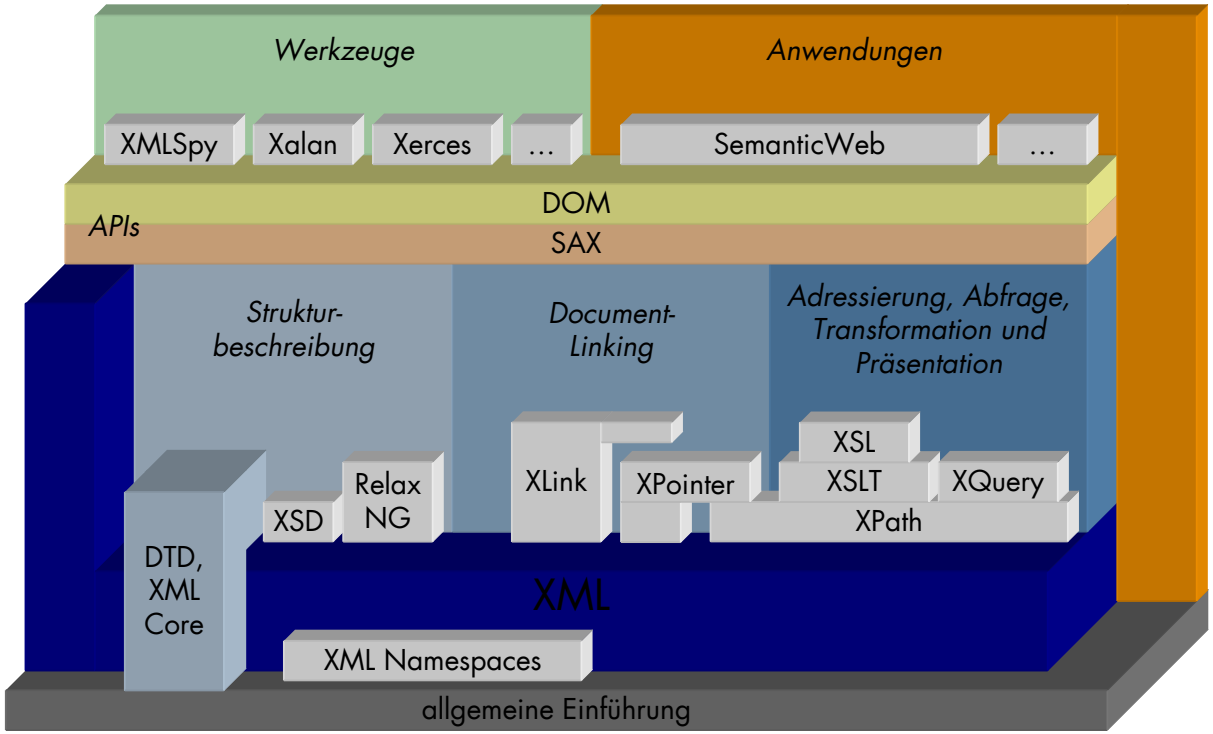
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Gliederung der Vorlesung



Einführung

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



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Motivation

- XML = Extensible Markup Language
- Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdaten
- Beispiele
  - XHTML
  - WSDL
  - SVG
  - addressDB

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.xml

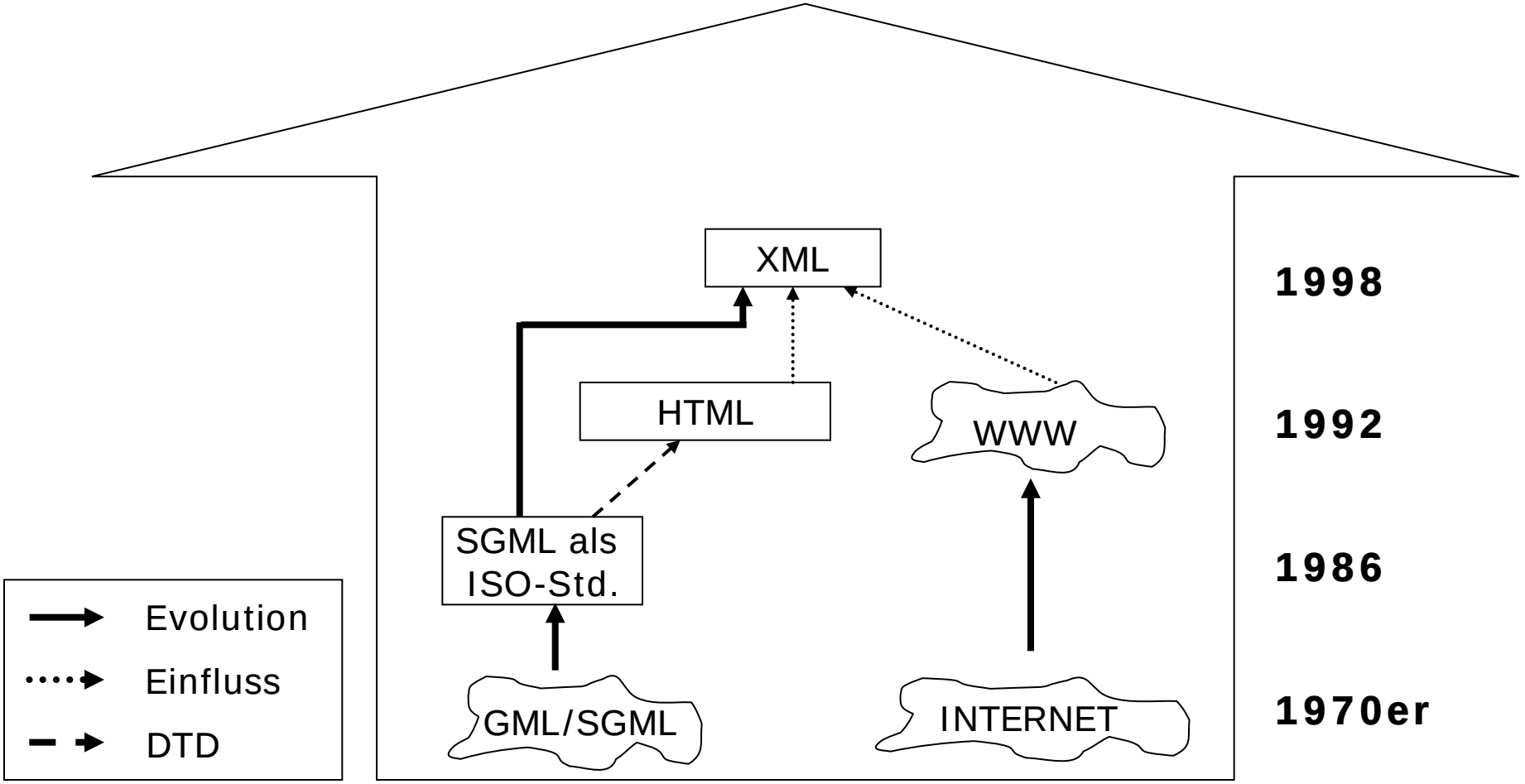
```
<Adresse>
    <Person Anrede="Frau">
        <Vorname>Eva</Vorname>
        <Name>Mustermann</Name>
    </Person>
    <Strasse Nummer="4">Beispielstrasse</Strasse>
    <Ort PLZ="54783">Musterstadt</Ort>
</Adresse>
```

instanz_von

repräsentiert

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

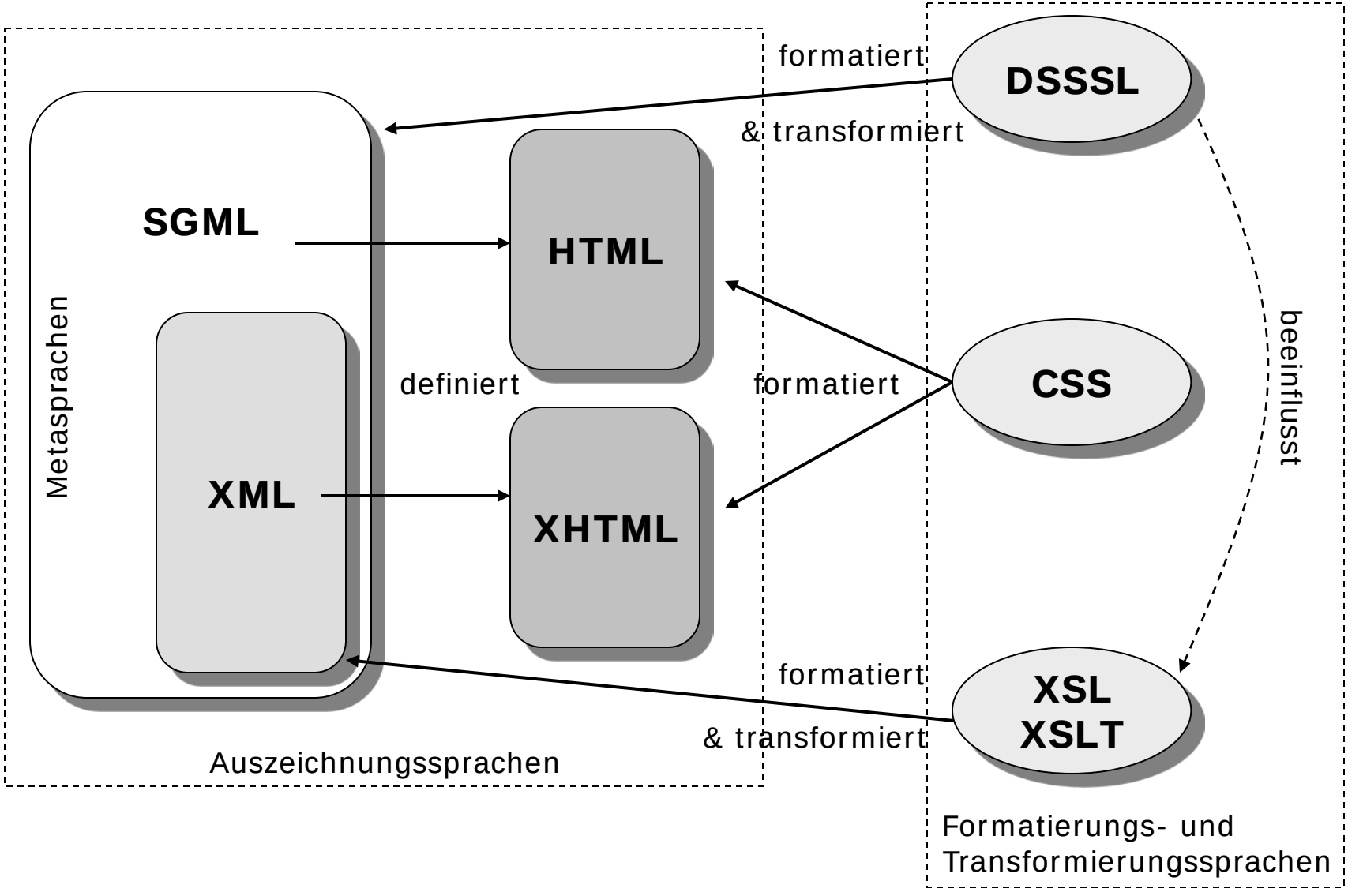
Historische Entwicklung von SGML, HTML und XML



Entwicklung von XML
Quelle: Merz

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Überblick Markup-Sprachen

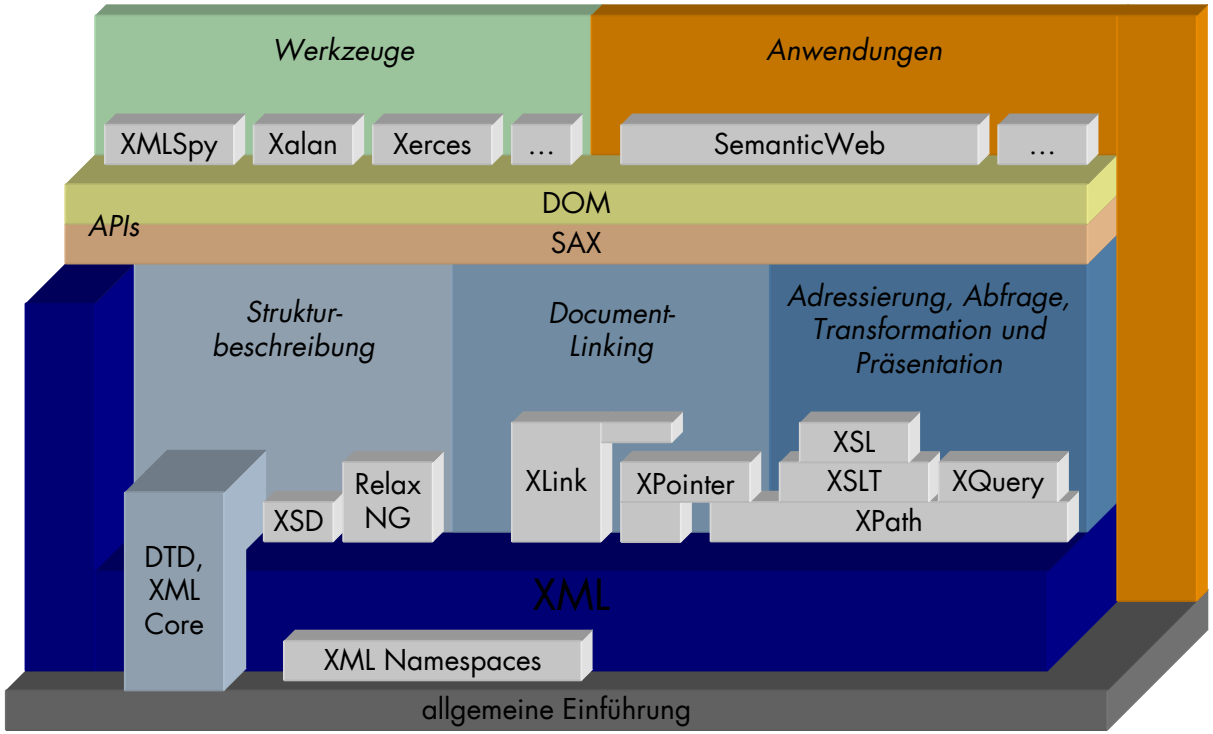


Quelle: Behme/Mintert

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

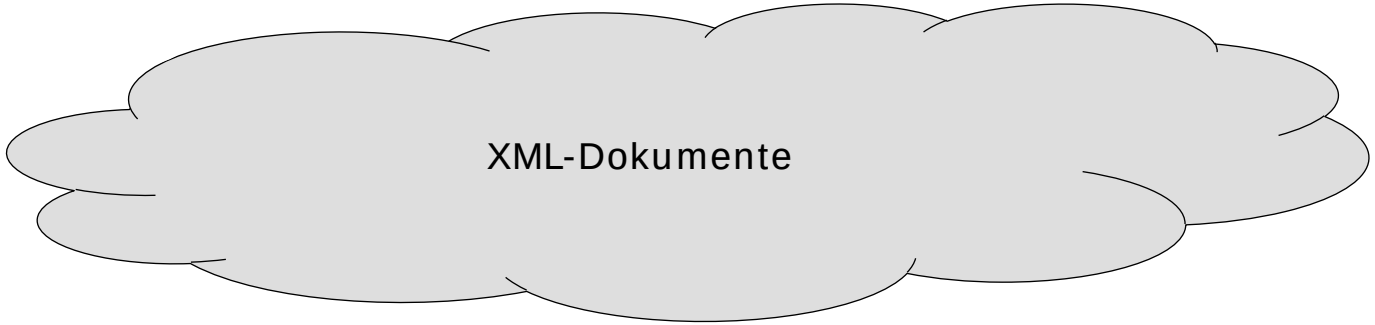
XML-Spezifikation

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



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

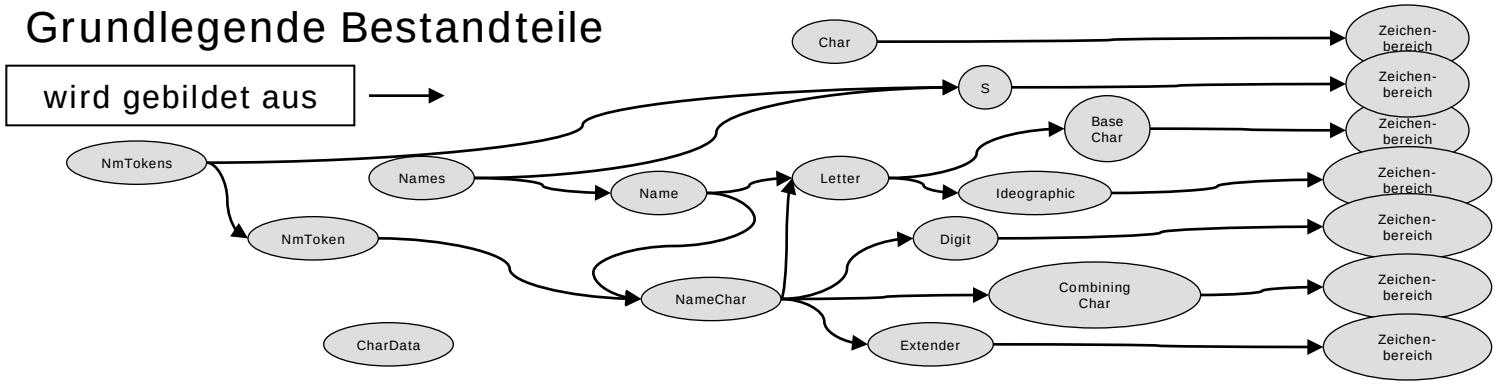
Bestandteile der XML-Grammatik



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

XML-Grammatik

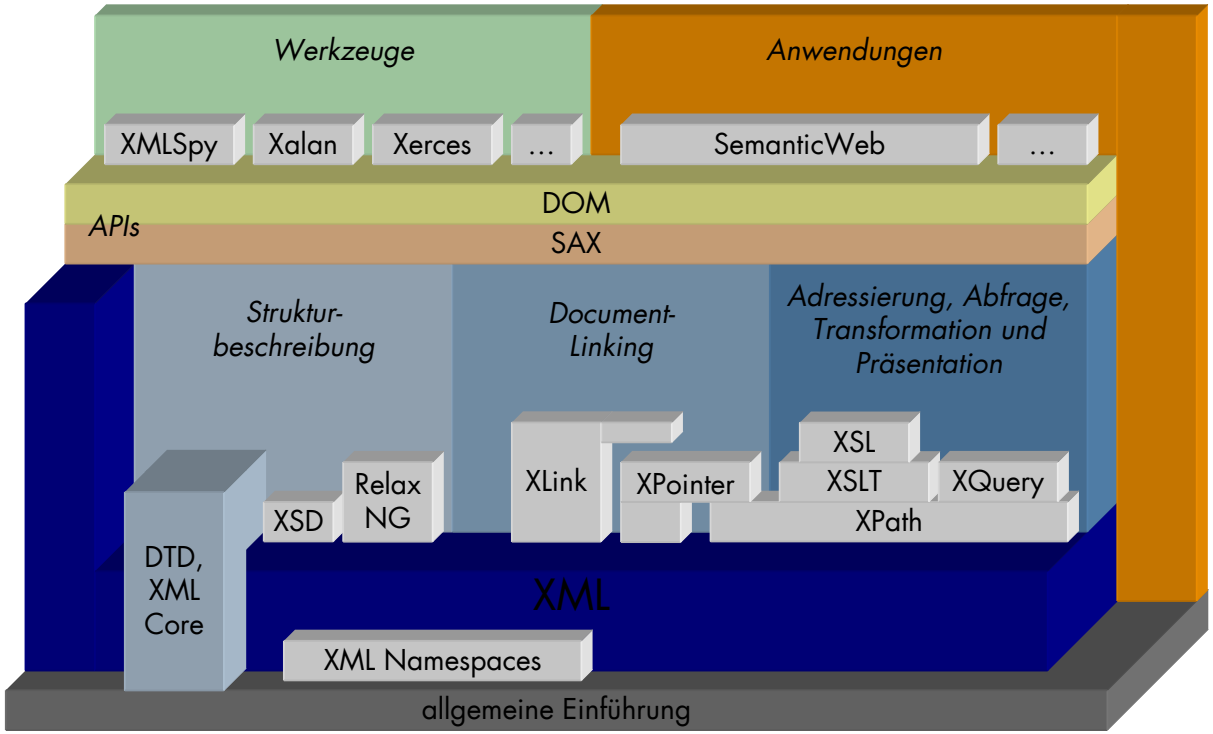
Grundlegende Bestandteile



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

XML-Namespaces

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



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

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.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Namensräume – Deklaration

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
```

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Namensräume – Verwendung generell

- 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

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Namensräume – Verwendung in Elementen

- 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>
```

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Namensräume – Verwendung in Attributen

- 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>
```

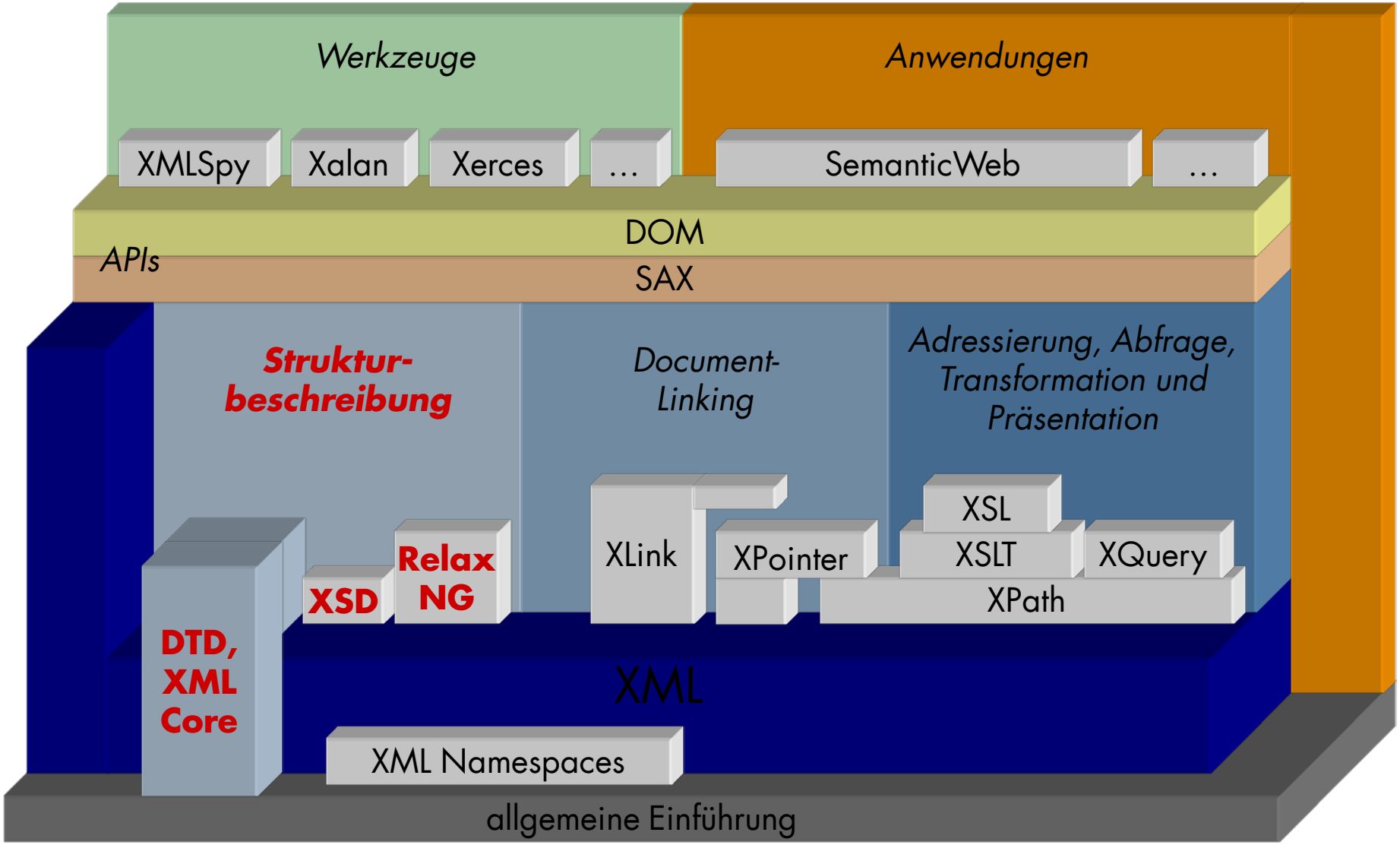
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Agenda

- Kurzzusammenfassung der Einführung
- **Kurzzusammenfassung der Strukturbeschreibungen**
- Diskussion
- Weiterer Ablauf

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

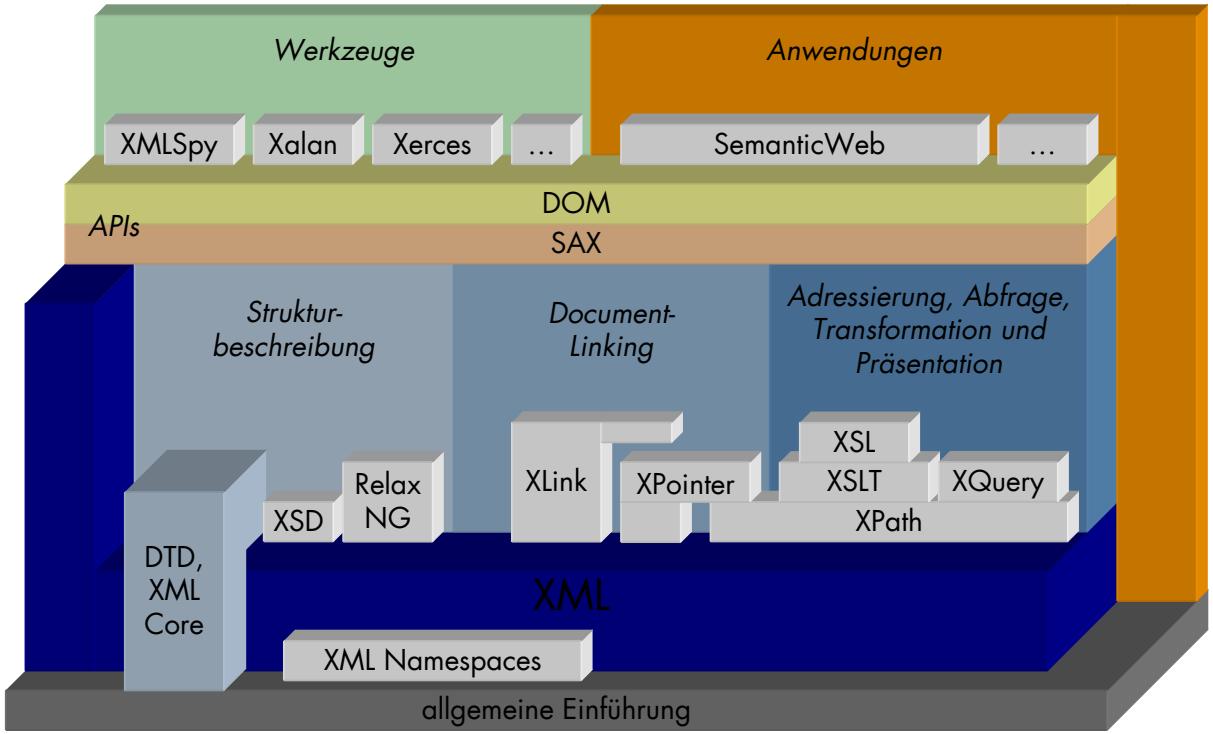
Gliederung der Vorlesung



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Strukturbeschreibung

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



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Motivation

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>
```

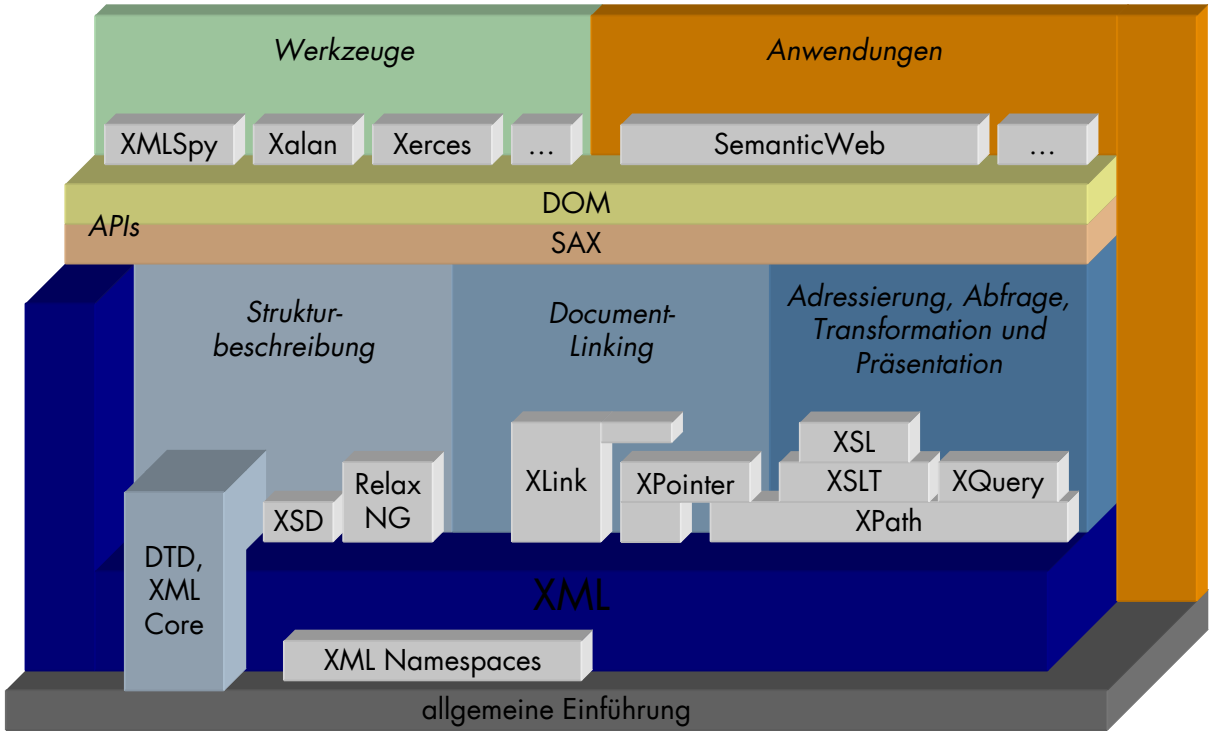
instanz_von

repräsentiert

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Strukturbeschreibung

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



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Beispiel

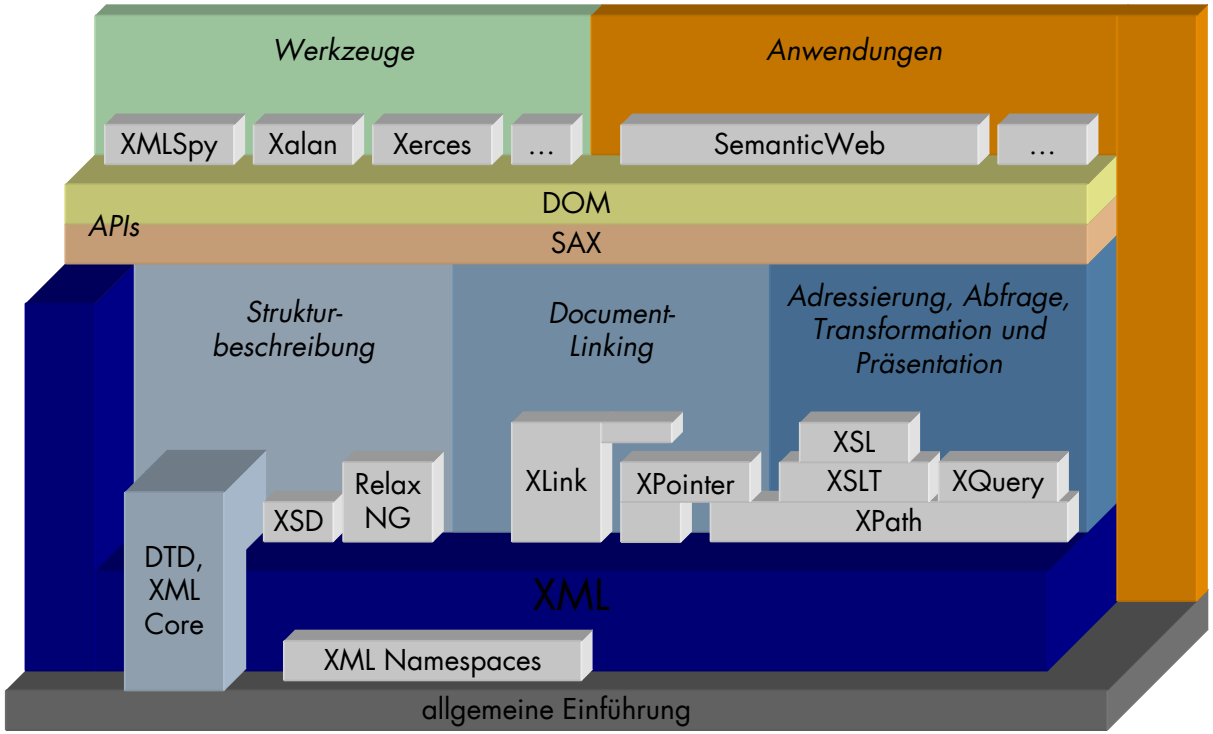
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE AdressDB SYSTEM "adressDB.dtd">
<!--Adress-Datenbank-->
<AdressDB>
    <Adresse>
        <Person Anrede="Frau">
            <Vorname>Eva</Vorname>
            <Name>Mustermann</Name>
        </Person>
        <Strasse Nummer="4">Beispielstrasse</Strasse>
        <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>
```

DTD

```
<?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>
```

2. Strukturbeschreibung

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



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

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>
    </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.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

XML Schema

```
<?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>
```

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

XML Schema

```
            <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"/>
```

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

XML Schema

```
<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>
```

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

XML Schema

```
    <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>
```

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

XML Schema

```
<?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">
        <xs:sequence>
            <xs:element name="Strasse" type="StrasseType"/>
            <xs:element name="Ort" type="OrtType"/>
        </xs:sequence>
    </xs:complexType>
    <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="StrasseType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <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:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="OrtType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="PLZ" type="xs:integer"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
```

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

"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.

AdresseType

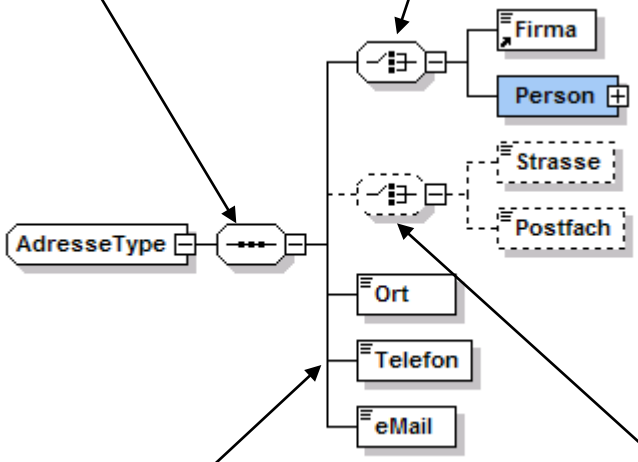
XML Schema

Besteht aus einer Sequenz

```
<xs:complexType name="AdresseT
                <xs:sequen
```

Von einer Auswahl:

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



Und drei "einfachen" Elementen:

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

weiten, jedoch optionalen Auswahl:

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

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

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="Postfach" minOccurs="0">
              <xs:simpleType>
                    <xs:restriction base="xs:integer">
```

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

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

```
<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"/>
```

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

XML Schema

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"/>
```

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"/>
```

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>
```

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

XML Schema

OrtType

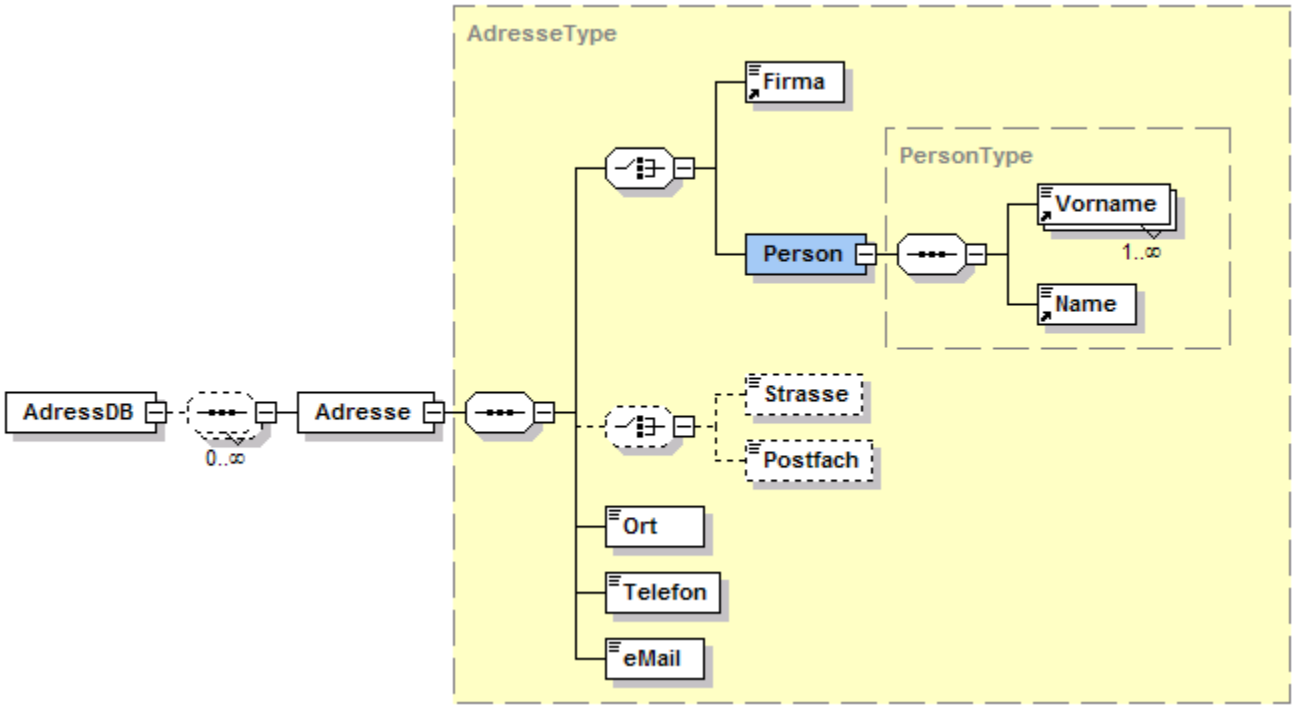
```
   <xs:complexType name="StrasseType">
      <xs:simpleContent>
         <xs:extension base="xs:string">
            <xs:attribute name="Nummer" use="required">
               <xs:simpleType>
                  <xs:restriction base="xs:string">
                     <xs:pattern value="\d{1,3}"/>
 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

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

Zu guter Letzt wird noch einmal ein einfaches Element deklariert.

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

Graphische Repräsentation des einleitenden Beispiels

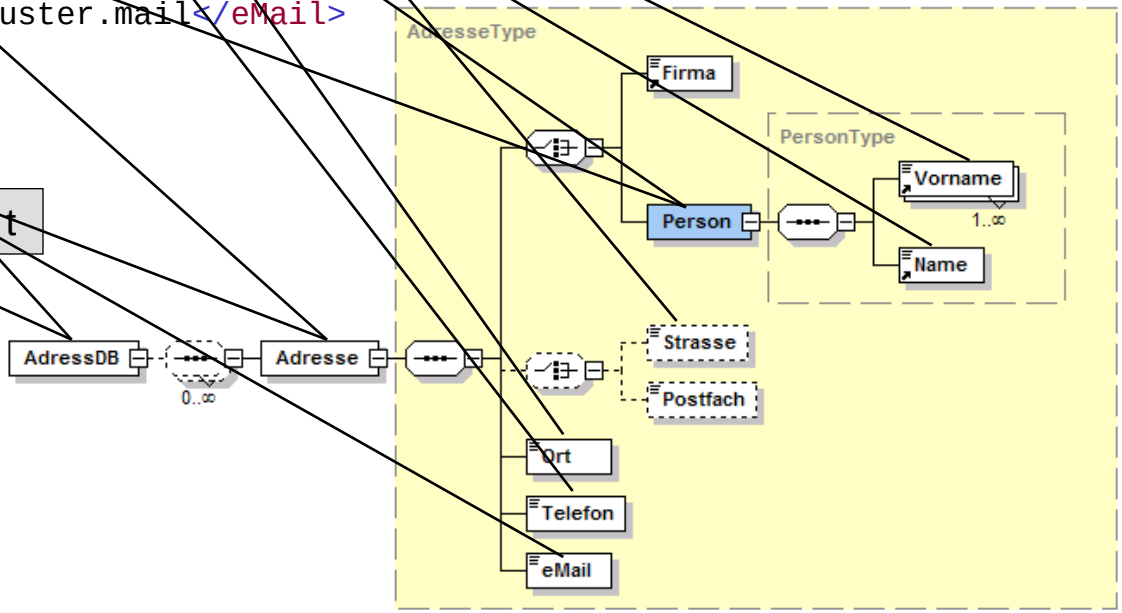


UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

XML Schema

```
<?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>
```

Wurzelelement



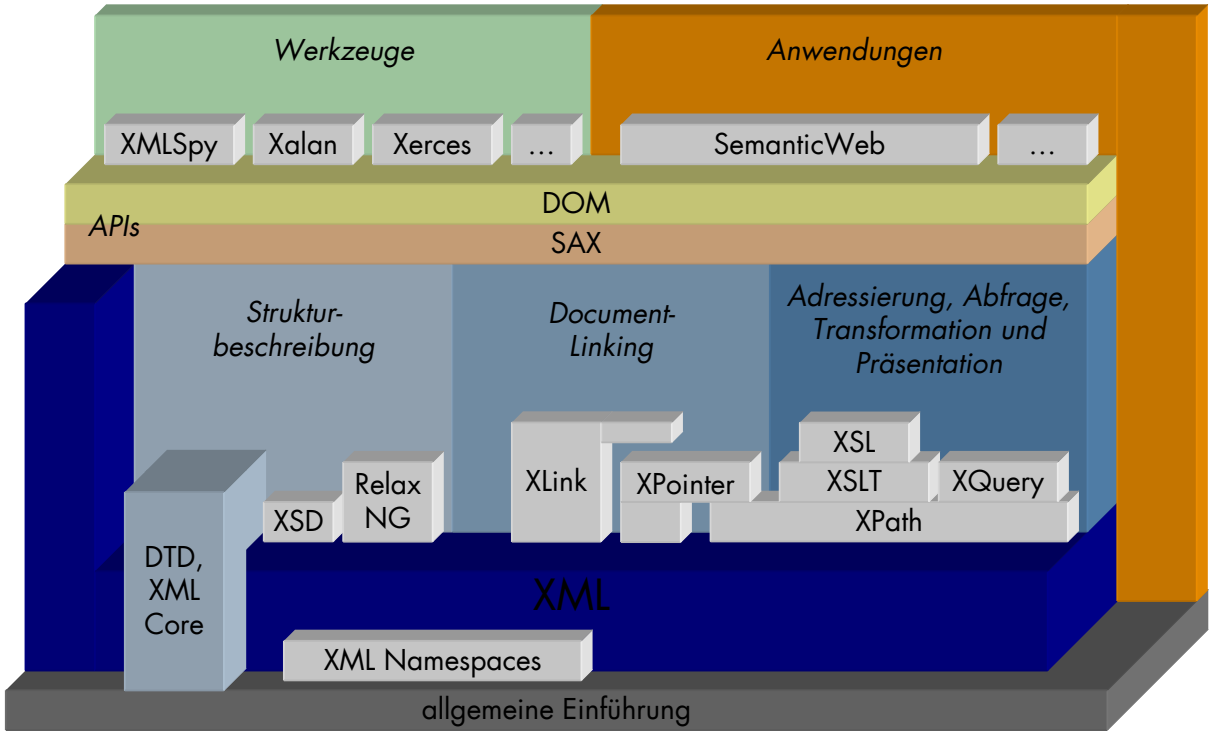
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Valides Dokument zum Schema

```
<?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>
```

2. Strukturbeschreibung

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



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

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>
```

Relax NG

```
            <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>
```

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Agenda

- Kurzzusammenfassung der Einführung
- Kurzzusammenfassung der Strukturbeschreibungen
- **Diskussion**

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Diskussion