

# DAIMLERCHRYSLER

## **XML-Schema**

### **(Neue?) Modellierungskonzepte und ihre Anwendung**

Mario Jeckle

DaimlerChrysler Forschungszentrum Ulm

[mario.jeckle@daimlerchrysler.com](mailto:mario.jeckle@daimlerchrysler.com)

[mario@jeckle.de](mailto:mario@jeckle.de)

[www.jeckle.de](http://www.jeckle.de)

## Gliederung

- XML-Strukturen: *XML Information Set*
- XML-Daten: *XML Typsystem*
- XML-Grammatiken: Historie und Hintergründe
- XML-Schema:
  - Motivation und Notwendigkeit
  - Mächtigkeit und Zielsetzung

## Die Extensible Markup Language

XML-Schema –  
(Neue?) Modellierungskonzepte und ihre Anwendung

2002-03-26

Workshop Modellierung 2002

Mario Jeckle

DaimlerChrysler Research and Technology

<http://www.jeckle.de>

[mario.jeckle@daimlerchrysler.com](mailto:mario.jeckle@daimlerchrysler.com)

## Die Extensible Markup Language

```
<?xml version="1.0" encoding="UTF-8"?>
<Vortrag>
  <Titel>XML-Schema –
    (Neue?) Modellierungskonzepte und ihre Anwendung</Titel>
  <Veranstaltung datum="2002-03-26">
    <Name>Workshop Modellierung 2002</Name>
  </Veranstaltung>
  <Referent>
    <Name>Mario Jeckle</Name>
    <Firma>DaimlerChrysler Research and Technology</Firma>
    <URL>http://www.jeckle.de</URL>
    <E-Mail>mario.jeckle@daimlerchrysler.com</E-Mail>
  </Referent>
</Vortrag>
```

## XML-Strukturvorrat (Bei Verwendung einer DTD)

- Hierarchisch Dokumentstruktur
- Verschachtelte Elemente (Festlegung des Inhaltsmodells)
- Attribute
- Freitext
- Notationen
- Entitäten (Textmakro auf Dokumentebene)
- Parametrisierte Entitäten (Textmakro auf DTD-Ebene)
- Verarbeitungsanweisungen
- Schlüssel/Referenzen
- Kommentare
- Uninterpretierte Blöcke
- Praktisch keine Unterstützung der XML-Namensräume
- Nicht als Metasprache seiner selbst geeignet

## XML-Typvorrat (Bei Verwendung einer DTD)

NMTOKEN	(NameChar)+
NMTOKENS	NMTOKEN (S NMTOKEN)*
CDATA	(Char* - (Char* ']]>' Char*))
ID	(Letter   '_'   ':') (NameChar)*
IDREF	ID
IDREFS	ID (S ID)*
ENTITY	ID
ENTITIES	ID (S ID)*
ENUMERATION	'( S? NMTOKEN (S? ' ' S? NMTOKEN)* S? )'

Hilfsdefinitionen:

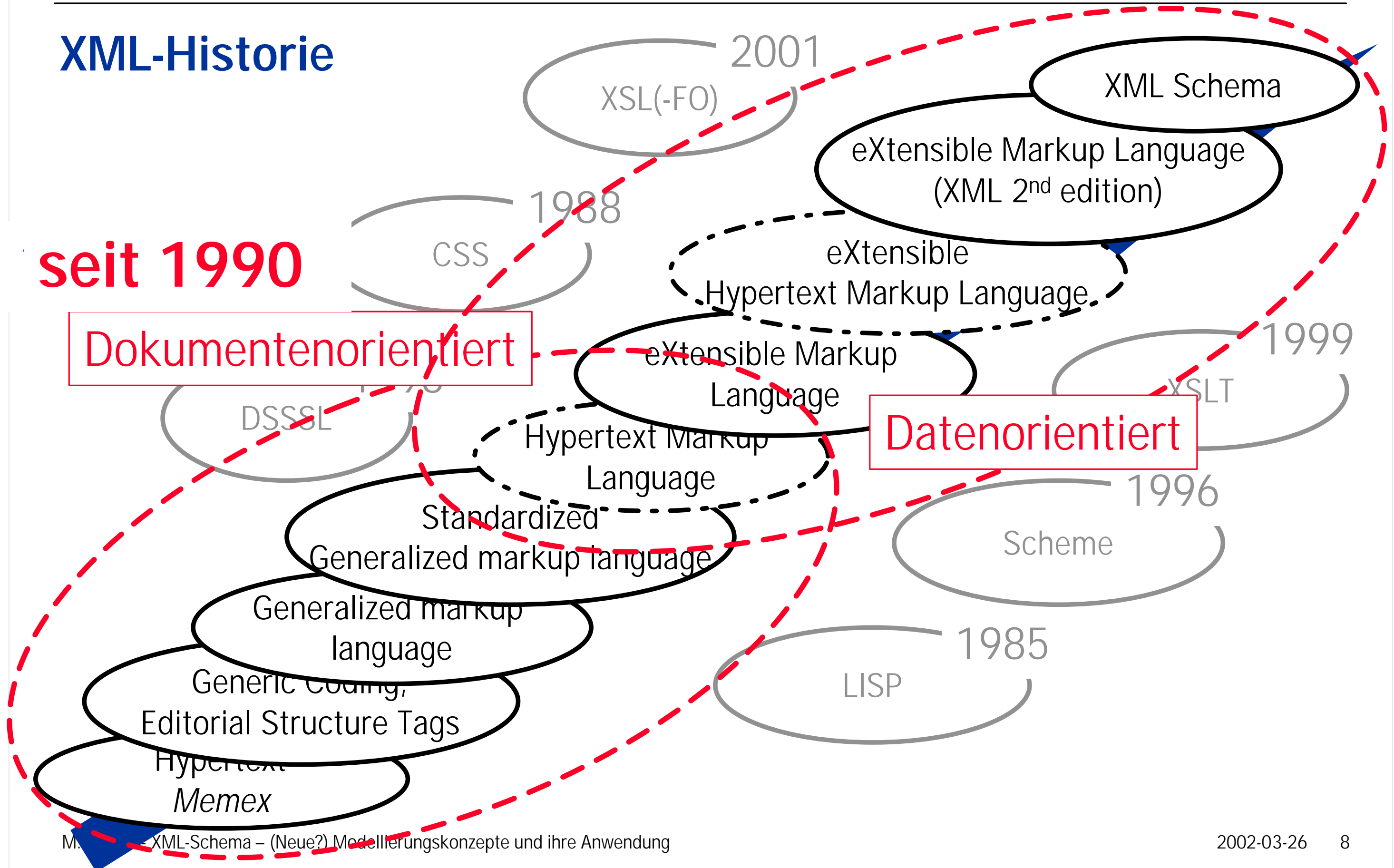
S	(#x20   #x9   #xD   #xA)+
NameChar	Letter   Digit   '.'   '-'   '_'   ':'   CombiningChar   Extender
Char	/* any Unicode character, excluding the surrogate blocks, FFFE, and FFFF. */

# XML-Historie

seit 1990

Dokumentenorientiert

Datenorientiert

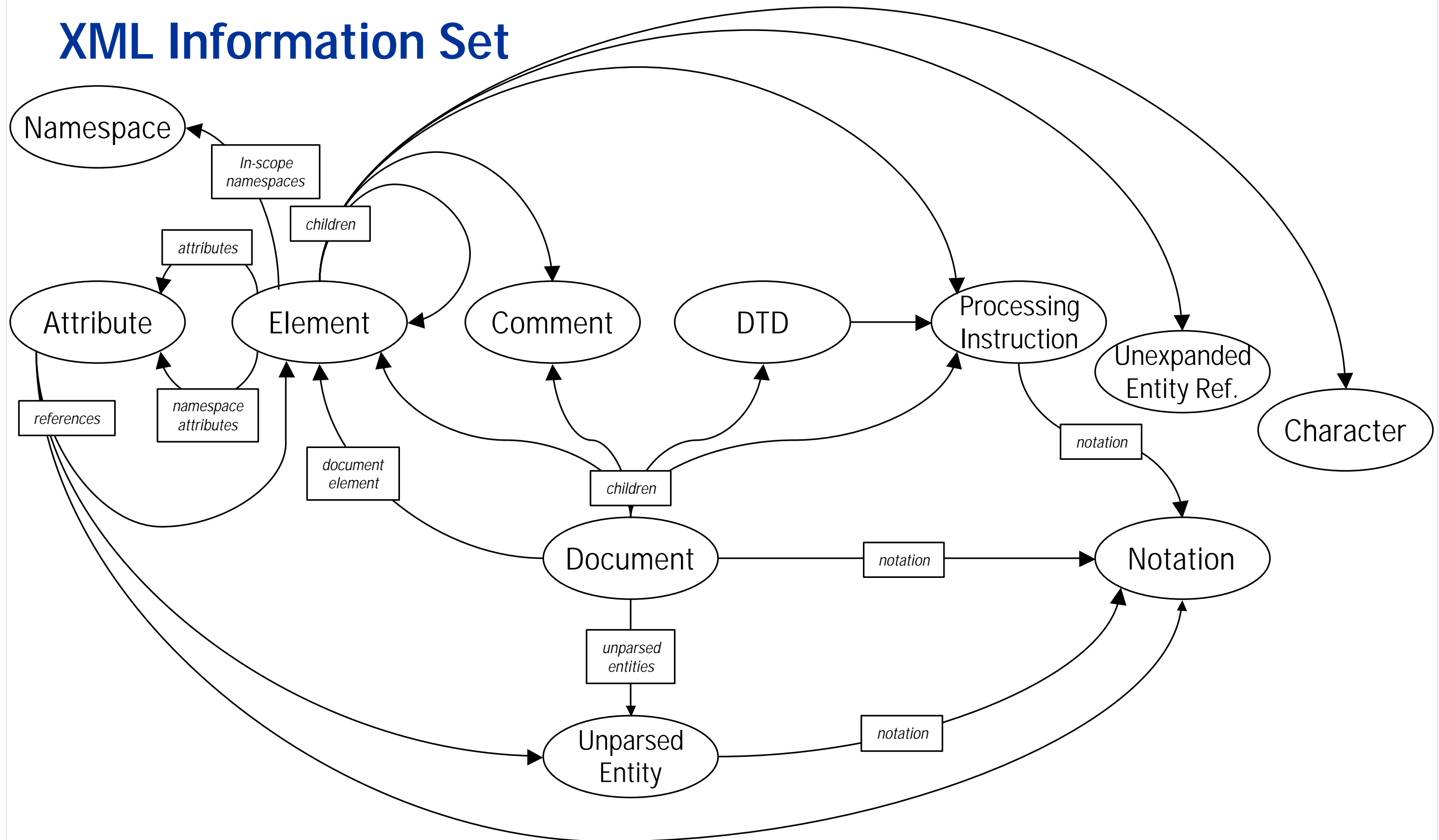


## XML Information Set

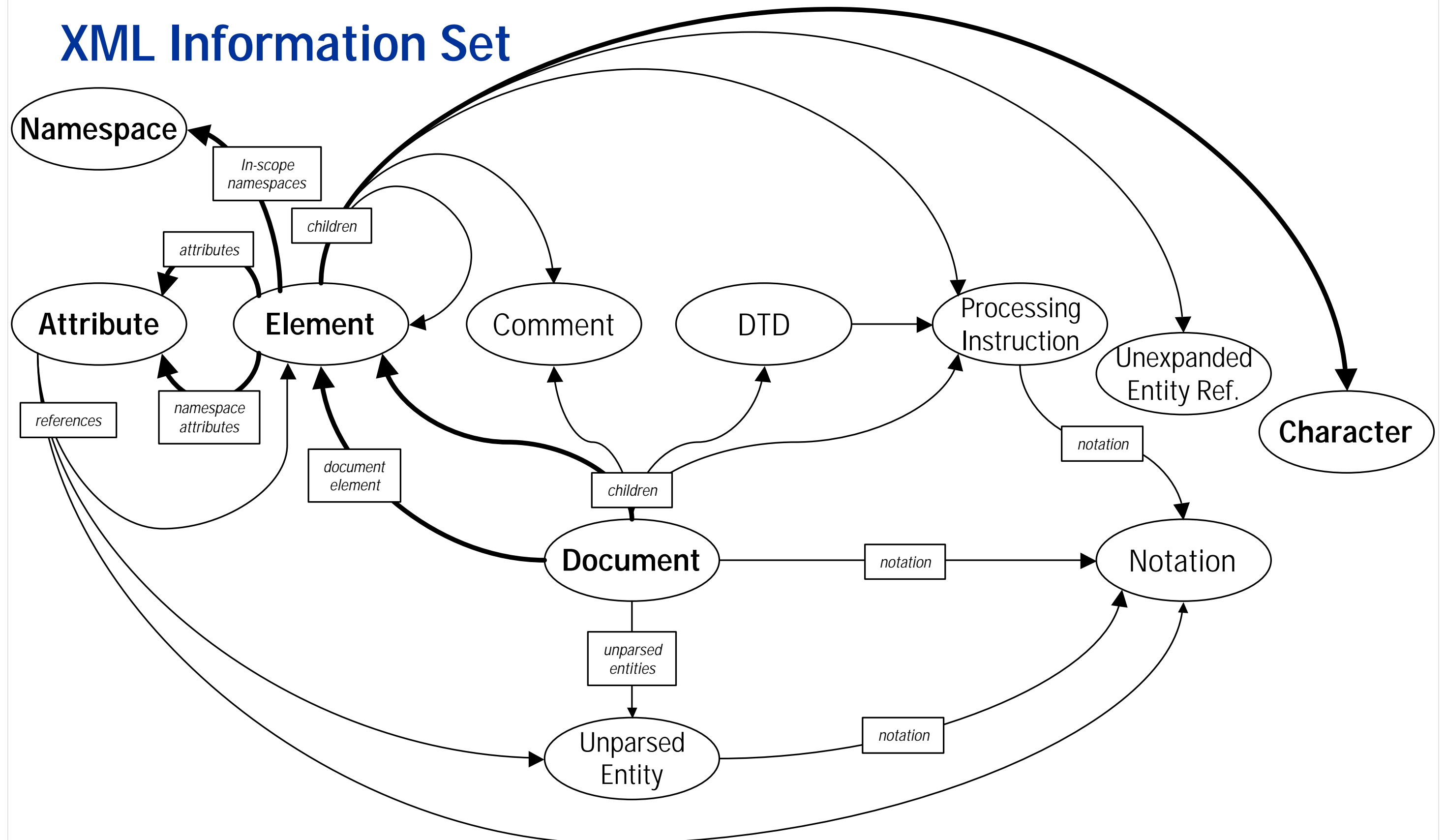
- Definiert strukturellen Aufbau einer XML-Quelle
  - ... als *abstrakte Datenmenge*
  - ... als Definitionssammlung
- Jedes wohlgeformte XML-Dokument besitzt einen InfoSet
  - ... auch wenn es nicht als Datei vorliegt
  - ... daher ist diese Spezifikation Basis weiterer W3C-Spezifikationen (XML-Schema, XMLP/SOAP, WSDL, ...)
- W3C Recommendation seit 2001-10-24
  - ... definiert keinerlei Programmierparadigmenspezifische Schnittstellen
  - ... operiert auf abstrakten Baumstrukturen
- Wird durch XML-Schemavalidierungsprozeß
  - ... auf strukturelle Konformität geprüft
  - ... mit Typinformation dekoriert



# XML Information Set



# XML Information Set

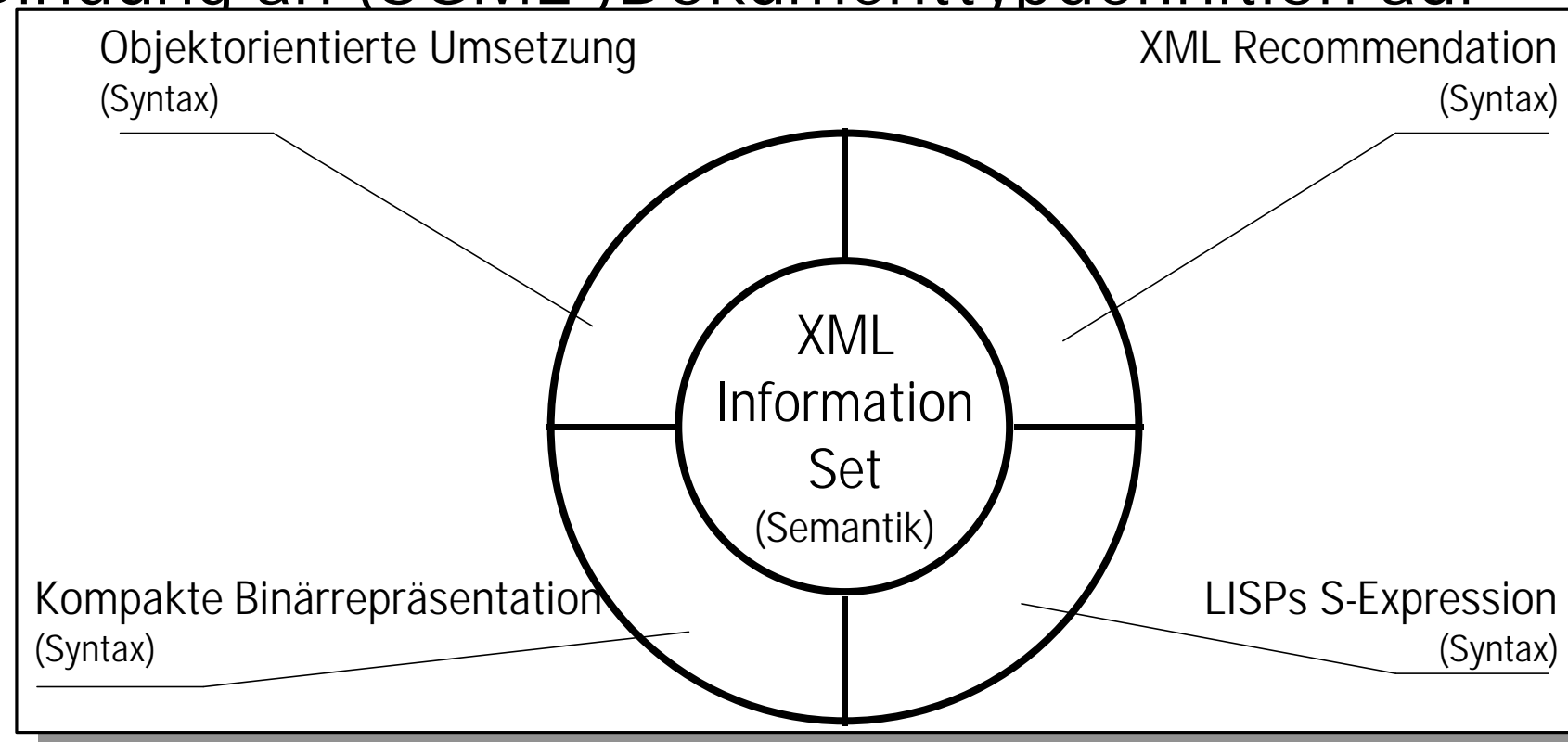


## Von DTD und InfoSet zu XML-Schema

- DTD
  - ... definiert ein nicht-XML-Vokabular zur Strukturbeschreibung
  - ... offeriert ein geschlossenes eng-limitiertes Typsystem
- InfoSet
  - ... definiert ausschließlich Struktur
  - ... trifft keine Aussage über die syntaktischen Inhaltsdarstellung
  - ... gibt Bindung an (SGML-)Dokumenttypdefinition auf

## Von DTD und InfoSet zu XML-Schema

- DTD
  - ... definiert ein nicht-XML-Vokabular zur Strukturbeschreibung
  - ... offeriert ein geschlossenes eng-limitiertes Typsystem
- InfoSet
  - ... definiert ausschließlich Struktur
  - ... trifft keine Aussage über die syntaktischen Inhaltsdarstellung
  - ... gibt Bindung an (SGML-)Dokumenttypdefinition auf



## Von DTD und InfoSet zu XML-Schema

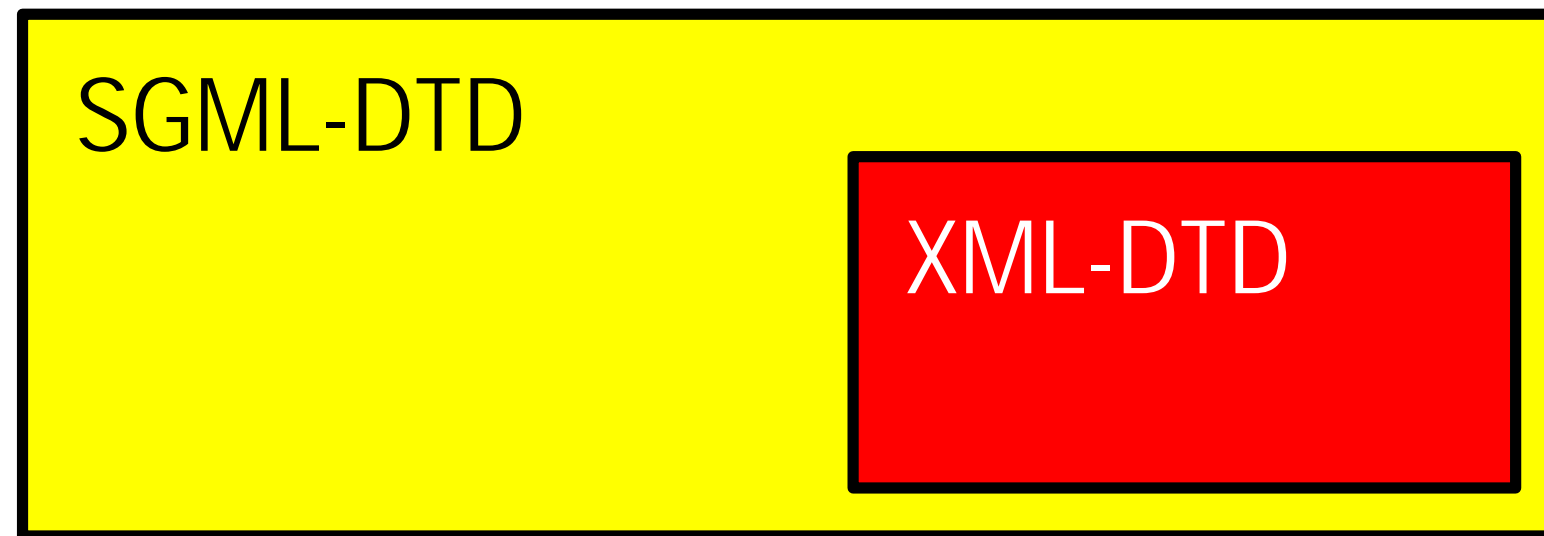
- DTD
  - ... definiert ein nicht-XML-Vokabular zur Strukturbeschreibung
  - ... offeriert ein geschlossenes eng-limitiertes Typsystem
- InfoSet
  - ... definiert ausschließlich Struktur
  - ... trifft keine Aussage über die syntaktischen Inhaltsdarstellung
  - ... gibt Bindung an (SGML-)Dokumenttypdefinition auf
- XML-Schema
  - Strukturelle Validierung
  - Typisierung
  - Erhalt und Erweiterung der DTD-Mächtigkeit

## Von DTD und InfoSet zu XML-Schema

### Erweiterungen des bestehenden (SGML-/XML-)DTD-Mechanismus

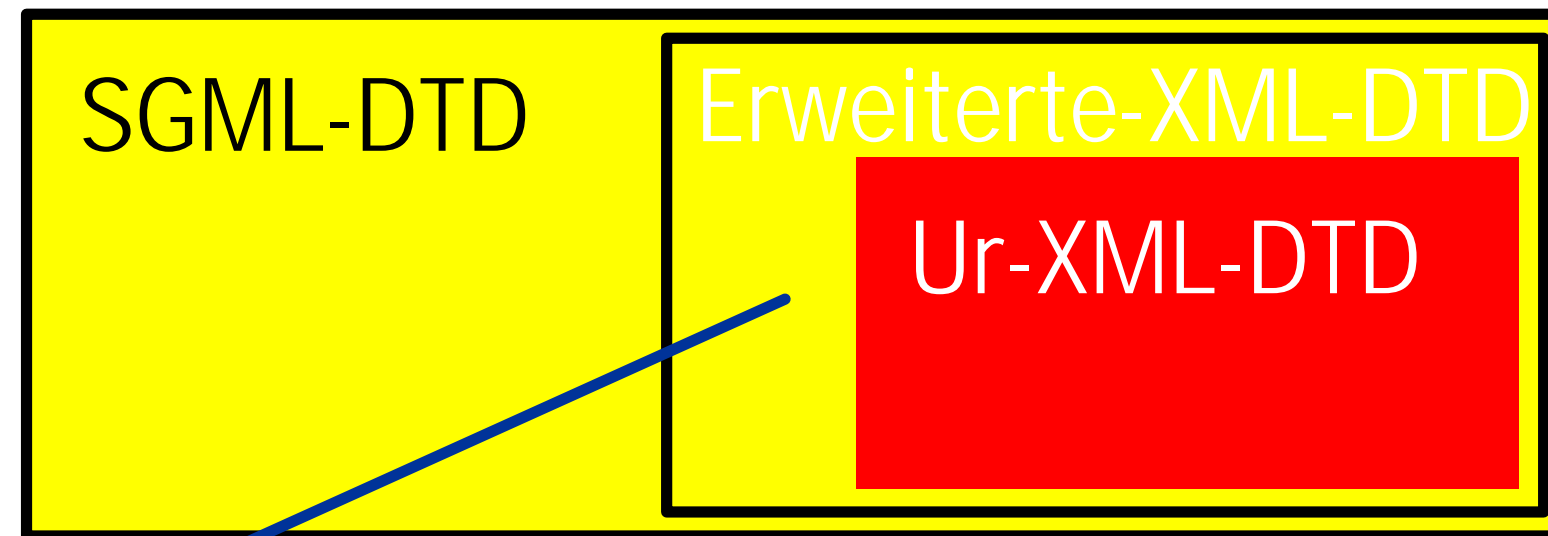
- Data Types for DTD (DT4DTD)

## Von DTD und InfoSet zu XML-Schema



XML-DTD bildet syntaktisch und hinsichtlich der Mächtigkeit eine echte Untermenge des SGML-DTD-Mechanismus

## Von DTD und InfoSet zu XML-Schema

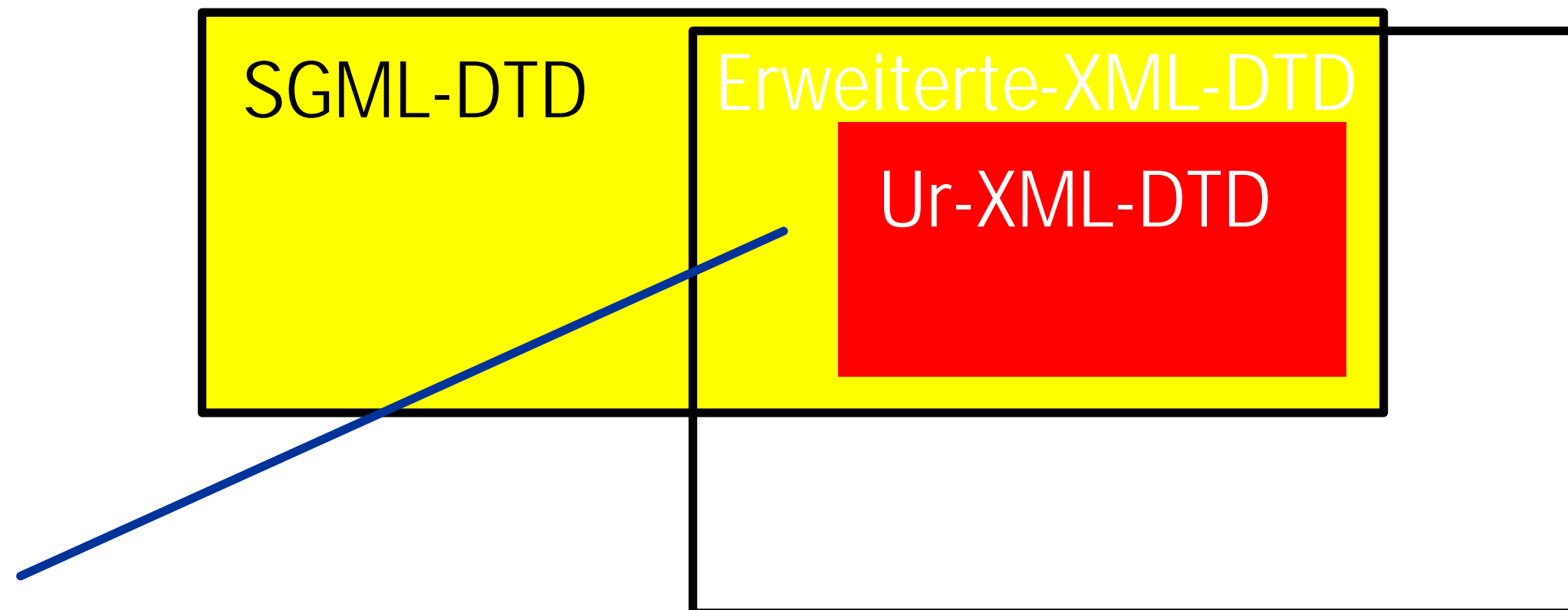


Erweiterung der XML-DTD-Mächtigkeit um weitere (bereits existierende) Primitive des SGML-DTD-Mechanismus:

- (+) (Unbestreitbarer) Mächtigkeitsgewinn
- (-) (unvermeidlicher) Komplexitätszuwachs
- (-) SGML-Mächtigkeit bildet natürliche Obergrenze



## Von DTD und InfoSet zu XML-Schema



Erweiterung der XML-DTD-Mächtigkeit um Primitive die über den SGML-DTD-Mechanismus hinausgehen:

- (+) Freiheitsgrad hinsichtlich beliebiger Erweiterungen
- (-) XML-Grammatiken keine Untermenge von SGML mehr

## Von DTD und InfoSet zu XML-Schema

### Erweiterungen des bestehenden (SGML-/XML-)DTD-Mechanismus

- Data Types for DTD (DT4DTD)

### Wissensbeschreibung

- Document Content Description for XML (DCD)  
*(RDF basierte Weiterentwicklung von XML-Data)*

### Inspiziert durch XML-API-Entwicklung

- Schema for Object oriented XML (SOX)

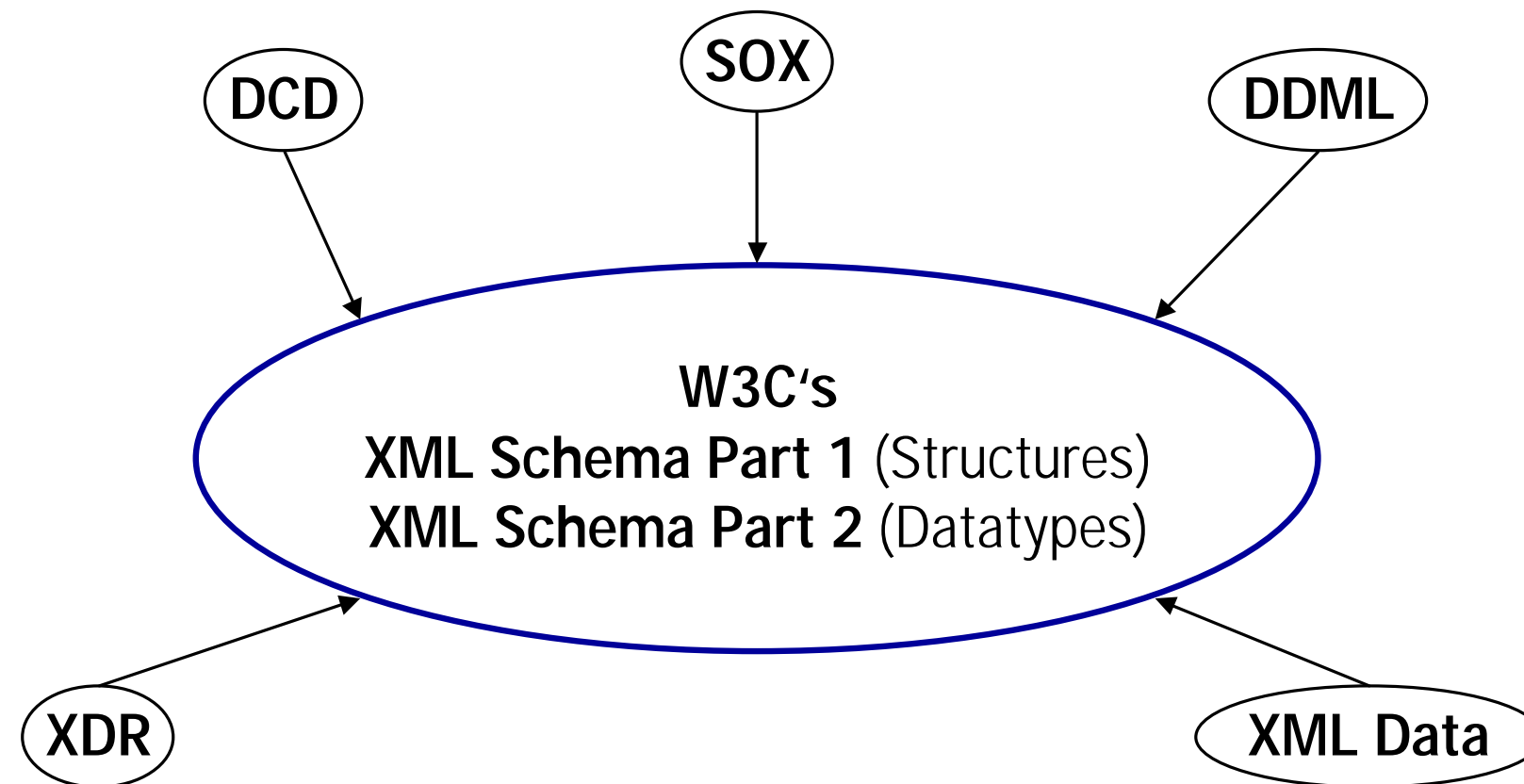
### XML-Sprachen zur Inhaltsbeschreibung

- Document Definition Markup Language/XSchema (DDML)
- Schematron (XSLT-basierte Auswertung der Dokumentstruktur)
- XML-Data/XML-Data Reduced (XDR) *(erster Ansatz noch vor Verabschiedung XML 1.0)*
- Document Structure Description (DSD)

## Von DTD und InfoSet zu XML-Schema

- DTD
  - ... definiert ein nicht-XML-Vokabular zur Strukturbeschreibung
  - ... offeriert ein geschlossenes eng-limitiertes Typsystem
- InfoSet
  - ... definiert ausschließlich Struktur
  - ... trifft keine Aussage über die syntaktischen Inhaltsdarstellung
  - ... gibt Bindung an (SGML-)Dokumenttypdefinition auf
- XML-Schema
  - Strukturelle Validierung
  - Typsystem (erweitert und erweiterbar) und –zuweisung
  - Konzeptionelle Obermenge der DTD-Mächtigkeit
  - Einige konkurrierende Vorschläge und Konzepte
  - Bekanntester Ansatz: W3Cs XML-Schema

## Von DTD und InfoSet zu XML-Schema

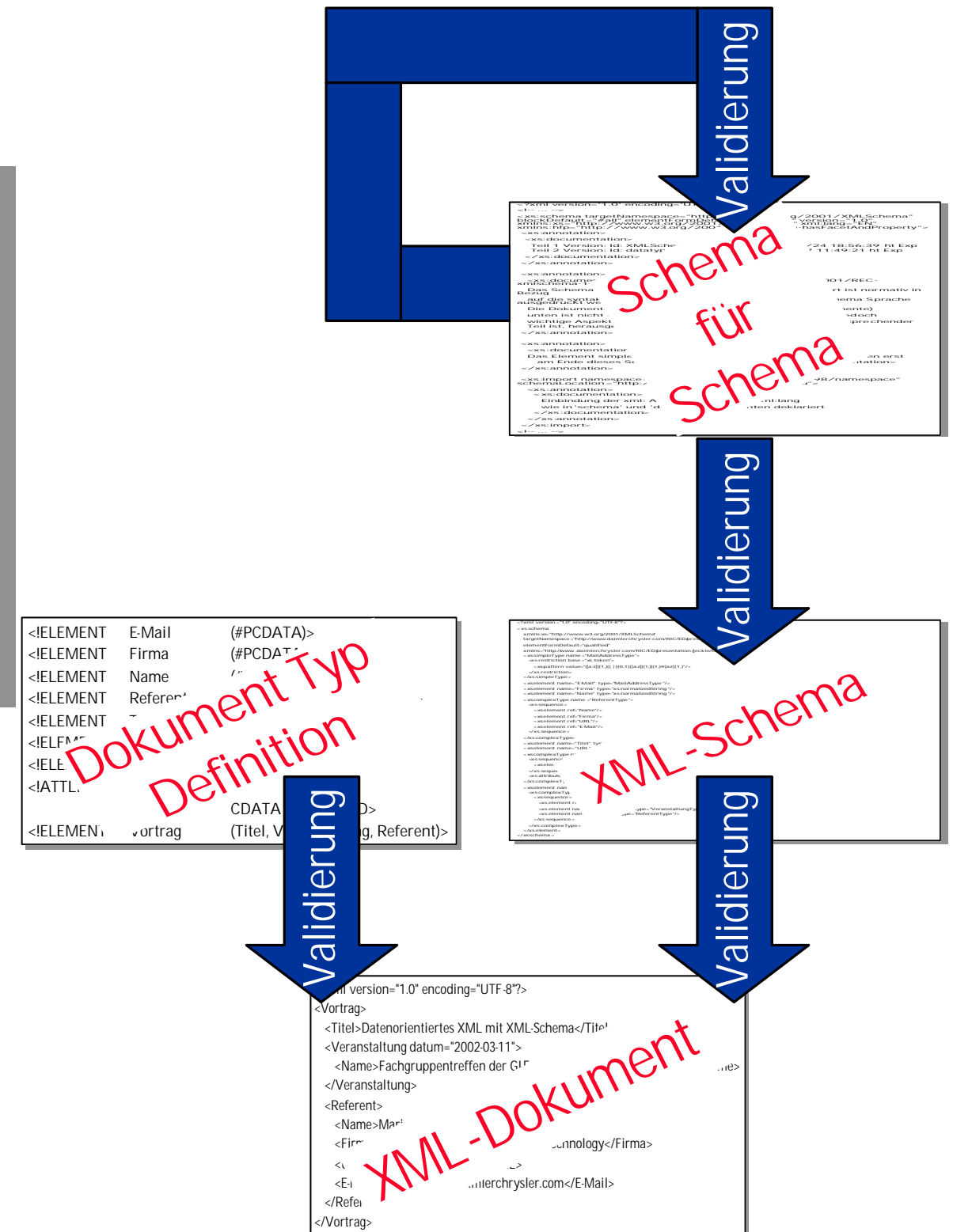


- XML-Schema
  - Strukturelle Validierung
  - Typsystem (erweitert und erweiterbar) und –zuweisung
  - Konzeptionelle Obermenge der DTD-Mächtigkeit
  - Einige konkurrierende Vorschläge und Konzepte
  - Bekanntester Ansatz: W3Cs XML-Schema

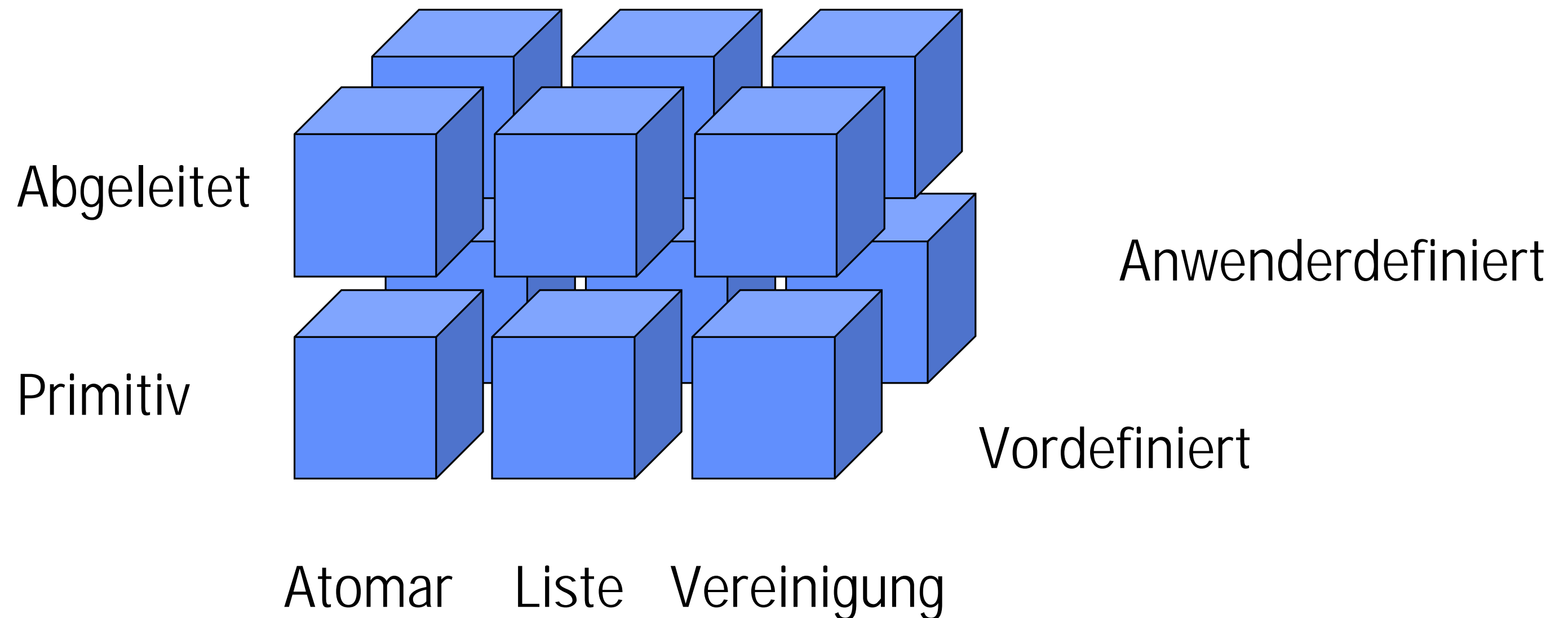
# XML-Schema: Organisatorisches

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://example.com"
  xmlns:ex="http://example.com">
...
</xs:schema>
```

- Der XML-Schema-Standard
  - ... ist strukturell selbstbeschreibend
  - ... gestattet die Definition jedes Schemas als schema-gültiger XML-Datei
  - ... unterstützt XML-Namensräume (in zweifacher Hinsicht)
  - Besteht aus zwei Teilen:
    - Strukturen
    - Datentypen



# XML-Schema: Typbegriff



# XML-Schema: Erweitertes Typsystem

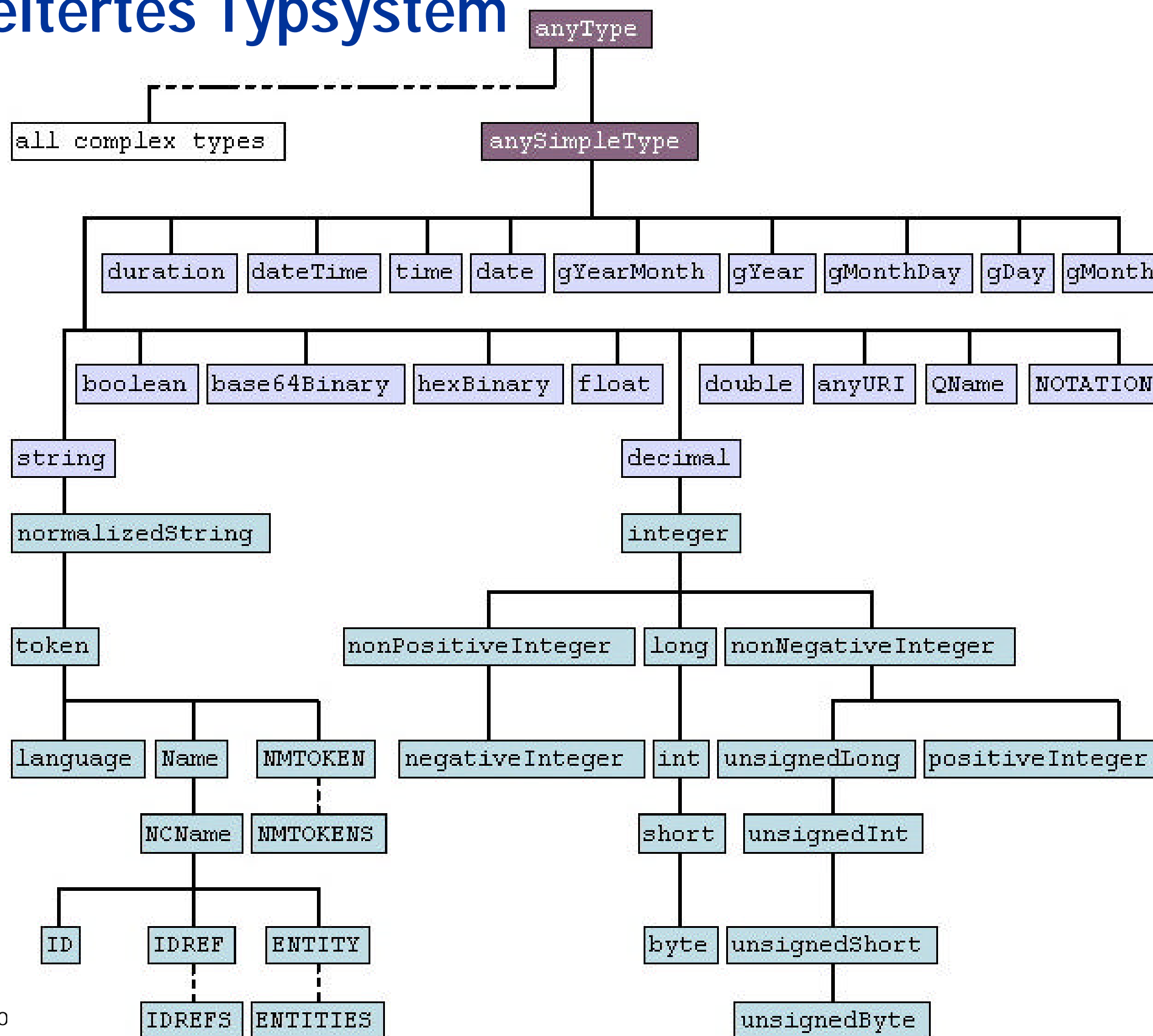
■ Ur-typ

■ Vordefinierter Primitivtyp

■ Vordefinierter abgeleiteter Typ

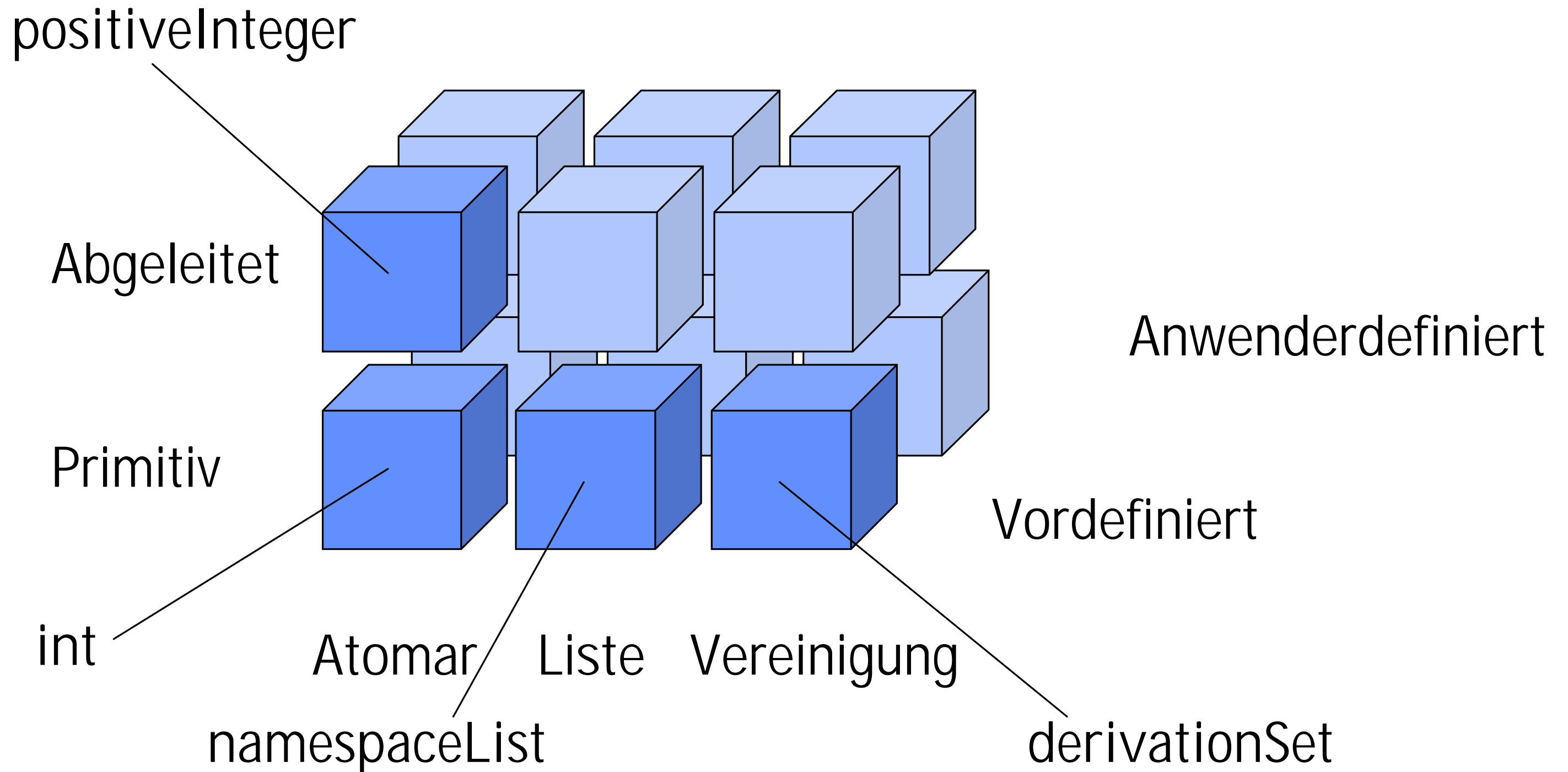
— Typeinschränkung

---- Aggregierter Typ



Aus: XML-Schema Part 2: Datatypes, W3C-Recommendation 2001-05-02, chap. 3

# XML-Schema: Typbegriff und Beispielextensionen





# XML-Schema: Erweitertes Typsystem

Typ	XML - Schema	Microsoft CLS	Java	SQL92/SQL3
$-2^7 \leq t \leq 2^7$	byte	-	byte	tinyint
$0 \leq t \leq 2^8 - 1$	unsignedByte	byte	-	
$-2^{15} \leq t \leq 2^{15} - 1$	short	short	short	smallint
$-2^{31} \leq t \leq 2^{31} - 1$	int	int	int	int, integer
$-2^{63} \leq t \leq 2^{63} - 1$	long	long	long	byte
Unicode-Zeichen(kette)	string	string	String	clob, char, character(n), varchar(n)
IEEE754	float double	float double	float double	float, real, double
{true,false,0,1}	boolean	bool	boolean	boolean
N	decimal	decimal	BigDecimal	numeric(p,s), dec, decimal(p,s)

# XML-Schema: Erweitertes Typsystem

```
<... xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ...>  
<xsi:element  
  name="URL" xsi:nillable="true"/>
```

## XML-Instanzdokument

```
<... xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ...>  
...  
<URL xsi:nil="true"/>  
...
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<Vortrag>  
  <Titel>XML-Schema –  
    (Neue?) Modellierungskonzepte und ihre Anwendung</Titel>  
  <Veranstaltung datum="2002-03-26">  
    <Name>Workshop Modellierung 2002</Name>  
  </Veranstaltung>  
  <Referent>  
    <Name>Mario Jeckle</Name>  
    <Firma>DaimlerChrysler Research and Technology</Firma>  
    <URL>http://www.jeckle.de</URL>  
    <E-Mail>mario.jeckle@daimlerchrysler.com</E-Mail>  
  </Referent>  
</Vortrag>
```

## NULL-Werte:

- Lukasiewicz-Logik für alle Skalartypen
- Analog: DBMS (SQL) und diverse Programmiersprachen

## XML-Schema: Erweiterbares Typsystem (strukturierte Typen)

```
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Vortrag">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Titel"/>
        <xs:element name="Veranstaltung">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Name"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Vortrag>
  <Titel>XML-Schema –
    (Neue?) Modellierungskonzepte und ihre Anwendung</Titel>
  <Veranstaltung datum="2002-03-26">
    <Name>Workshop Modellierung 2002</Name>
  </Veranstaltung>
  <Referent>
    <Name>Mario Jeckle</Name>
    <Firma>DaimlerChrysler Research and Technology</Firma>
    <URL>http://www.jeckle.de</URL>
    <E-Mail>mario.jeckle@daimlerchrysler.com</E-Mail>
  </Referent>
</Vortrag>
```

Titel

Veranstaltung

Name

## XML-Schema: Erweiterbares Typsystem (strukturierte Typen)

```
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Vortrag">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Titel"/>
        <xs:element name="Veranstaltung">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Name"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Vortrag>
  <Titel>XML-Schema –
    (Neue?) Modellierungskonzepte und ihre Anwendung</Titel>
  <Veranstaltung datum="2002-03-26">
    <Name>Workshop Modellierung 2002</Name>
  </Veranstaltung>
  <Referent>
    <Name>Mario Jeckle</Name>
    <Firma>DaimlerChrysler Research and Technology</Firma>
    <URL>http://www.jeckle.de</URL>
    <E-Mail>mario.jeckle@daimlerchrysler.com</E-Mail>
  </Referent>
</Vortrag>
```

### Strukturierte Typen:

- Explizieren Hierarchisierung
- Basis der Wiederverwendung
- Analogon: Javas anonyme innere Klassen

## XML-Schema: Erweiterbares Typsystem (strukturierte Typen)

```
<xs:complexType name="ReferentType">
  <xs:sequence>
    ...
  </xs:sequence>
</xs:complexType>
```

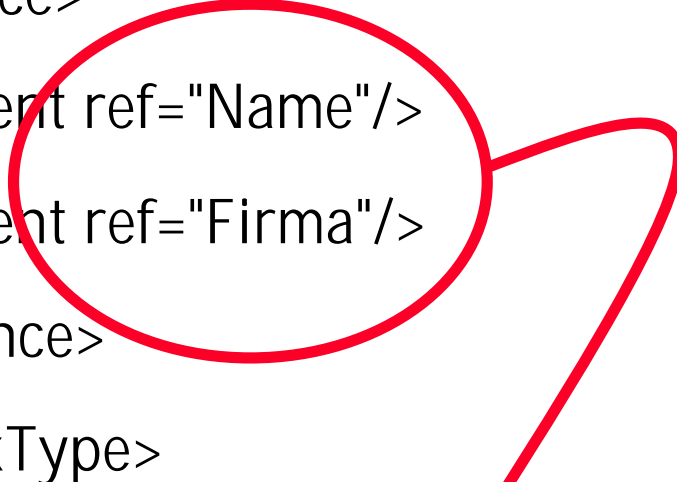
```
<?xml version="1.0" encoding="UTF-8"?>
<Vortrag>
  <Titel>XML-Schema –
    (Neue?) Modellierungskonzepte und ihre Anwendung</Titel>
  <Veranstaltung datum="2002-03-26">
    <Name>Workshop Modellierung 2002</Name>
  </Veranstaltung>
  <Referent>
    <Name>Mario Jeckle</Name>
    <Firma>DaimlerChrysler Research and Technology</Firma>
    <URL>http://www.jeckle.de</URL>
    <E-Mail>mario.jeckle@daimlerchrysler.com</E-Mail>
  </Referent>
</Vortrag>
```

Benannte strukturierte Typen:

- Verwendungsunabhängige Inhaltsmodellformulierung
- Analog: Klassen in der objektorientierten Modellierung

## XML-Schema: Erweiterbares Typsystem (strukturierte Typen)

```
<xs:complexType name="ReferentType">
  <xs:sequence>
    <xs:element ref="Name"/>
    <xs:element ref="Firma"/>
  </xs:sequence>
</xs:complexType>
```



```
<?xml version="1.0" encoding="UTF-8"?>
<Vortrag>
  <Titel>XML-Schema –
    (Neue?) Modellierungskonzepte und ihre Anwendung</Titel>
  <Veranstaltung datum="2002-03-26">
    <Name>Workshop Modellierung 2002</Name>
  </Veranstaltung>
  <Referent>
    <Name>Mario Jeckle</Name>
    <Firma>DaimlerChrysler Research and Technology</Firma>
    <URL>http://www.jeckle.de</URL>
    <E-Mail>mario.jeckle@daimlerchrysler.com</E-Mail>
  </Referent>
</Vortrag>
```

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

Benannte strukturierte Typen:

- Verwendungsunabhängige Inhaltsmodellformulierung
- Analog: Klassen in der objektorientierten Modellierung

Elementreferenzen:

- Trennung zwischen definierendem und nutzendem Auftreten



## XML-Schema: Erweiterbares Typsystem (strukturierte Typen)

```
<xs:complexType name="ReferentTypeExt">
  <xs:complexContent>
    <xs:extension base="ReferentType">
      <xs:sequence>
        <xs:element
          name="E-Mail-Adresse"
          type="MailAddressType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<Vortrag>
  <Titel>XML-Schema –
    (Neue?) Modellierungskonzepte und ihre Anwendung</Titel>
  <Veranstaltung datum="2002-03-26">
    <Name>Workshop Modellierung 2002</Name>
  </Veranstaltung>
  <Referent>
    <Name>Mario Jeckle</Name>
    <Firma>DaimlerChrysler Research and Technology</Firma>
    <URL>http://www.jeckle.de</URL>
    <E-Mail>mario.jeckle@daimlerchrysler.com</E-Mail>
  </Referent>
</Vortrag>
```

Erweiterung strukturierter Typen:

- Ausschließlich durch Extension
- Substitutionssemantik unberücksichtigt
- Analog: (hinsichtlich Erweiterung) objektorientierter Vererbung

## XML-Schema: Erweiterbares Typsystem (skalare Typen)

Aggregation existierender Typen:

```
<xs:simpleType name="WarenkorbElemente">  
  <xs:list itemType="xs:token"/>  
</xs:simpleType>
```

Typkomposition:

```
<xsd:simpleType name="termin">  
  <xsd:union memberTypes="xs:date  
    NamenDerWochentage"/>  
</xsd:simpleType>
```

- Von existierenden Typen abhängige Neudefinition
- Implizite Substituierbarkeit gemäß XML-Schema-Typsystem bleibt erhalten
- Analog: Aggregation und Vereinigung in bekannten Programmiersprachen



## XML-Schema: Erweiterbares Typsystem (skalare Typen)

### Vollständige Enumeration:

```
<xs:simpleType name="ampelfarben">
  <xs:restriction base="xs:string">
    <xs:enumeration value="rot"/> ...
  </xs:restriction>
</xs:simpleType>
```

### Domänenrestriktion:

```
<xsd:simpleType name="Postleitzahl">
  <xsd:restriction base="xs:string">
    <xsd:length value="5"/>
  </xsd:restriction>
</xsd:simpleType>
```

### Reguläre Ausdrücke:

```
<xs:simpleType
  name="MailAddressType">
  <xs:restriction base="xs:string">
    <xs:pattern
      value="([a-z]){1,}((.|-){0,1}([a-z]){1,}){1,}@([a-z]){1,}"/>
    </xs:restriction>
  </xs:simpleType>
```

- Beschränkung existierender Typen
  - ... hinsichtlich der zugelassenen Extensionen
  - ... hinsichtlich eines Merkmals
- Analog:
  - Domänenbildung in DBMS
  - Reguläre Ausdrücke (z.B. Perl)

## Ist XML-Schema „objektorientiertes XML“?

- XML-Schema vereinigt viele Merkmale verfügbarer OO-Sprachen
- XML-Schema ist jedoch keine Datenmodellierungssprache
- ... ebenso wenig eine Implementierungssprache!
- Sinnvoll: Prozeßeinsatz von XML-Schema *gemeinsam* mit OO-Modellierungssprache
- Einsatzszenario: XML-Schema zur Darstellung der textbasierten Ein- und Ausgabeformate, evtl. auch zur Persistenz-“Modellierung“
- Notwendig: Abgestimmtes Vorgehen um Redundanz zu vermeiden und Zeitaufwände zu minimieren
- Standardisiert:  
Transformationsalgorithmus UML (v1.x) → DTD und XML-Schema