DAIMLERCHRYSLER

W3C's XML Schema

Die Sprache der Sprachen

Mario Jeckle
DaimlerChrysler Forschungszentrum Ulm
mario.jeckle@daimlerchrysler.com
mario@jeckle.de
www.jeckle.de

Gliederung

I Von Dokumenten zu Daten ...

Die Document Type Definition und ihre Eignung zur Darstellung datenorientierter XML-Strukturen

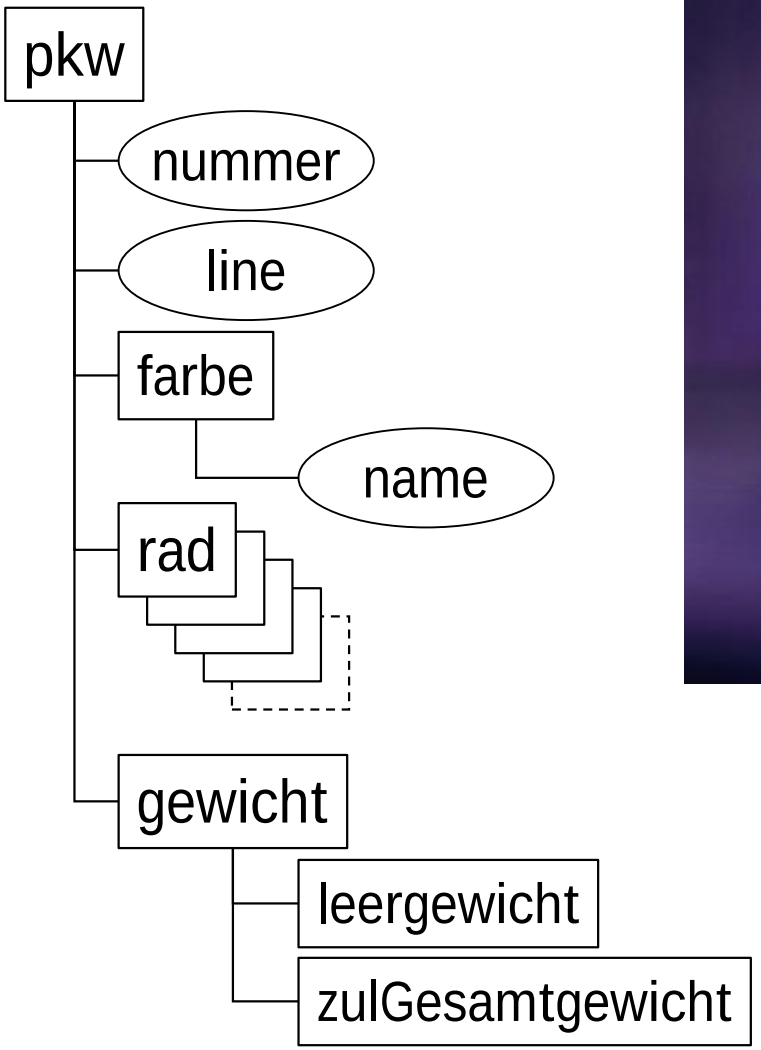
II Notwendigkeit und Anforderungen an einen XML-Schemamechanismus

- Hintegründe
- Ansätze
- Standards

III W3C's XML Schema im Einsatz ...

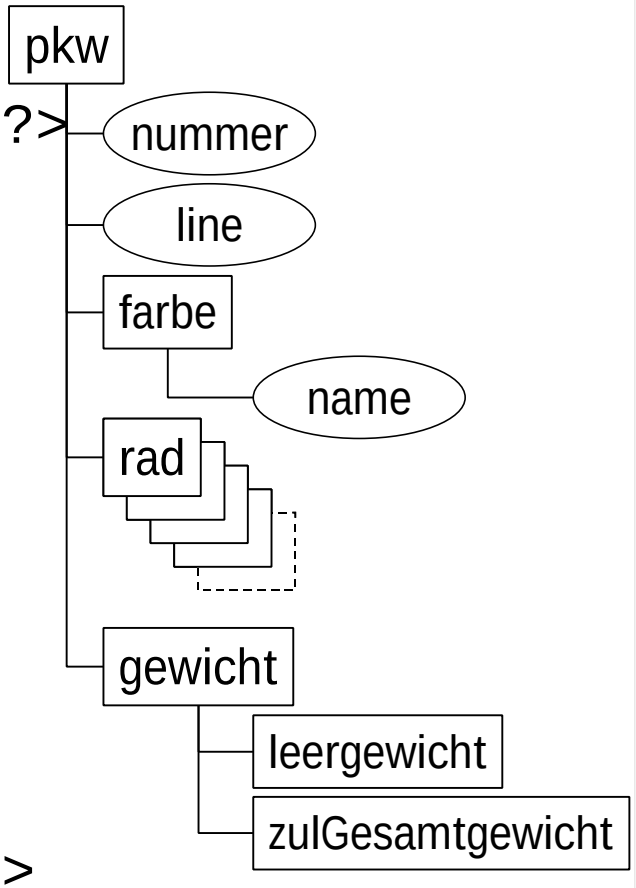
- durchgängiges Beispiel
- praktische Erfahrungen
- Hinweise zum Einsatz

Eine XML-Struktur



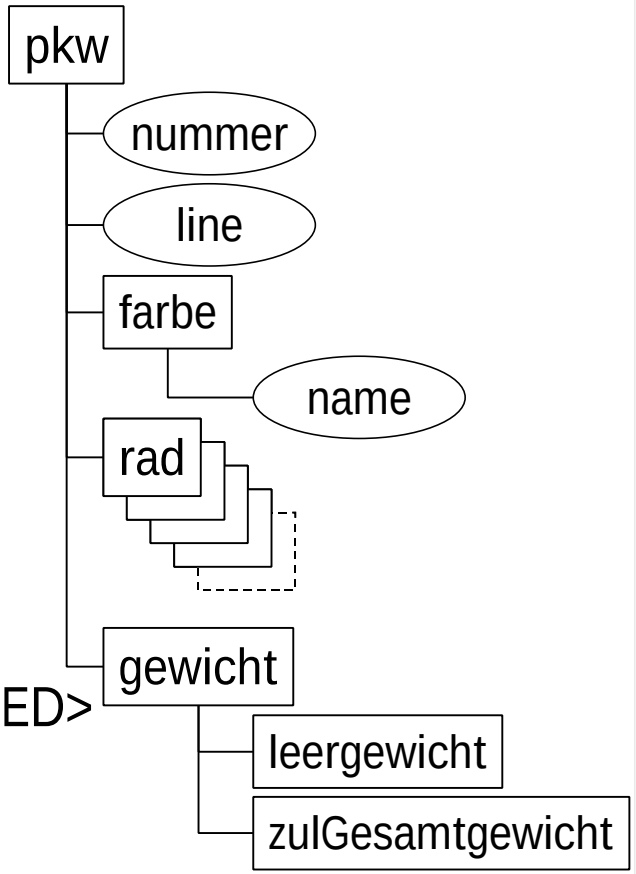
Ein XML-Dokument

```
<?xml version="1.0" encoding="UTF-8"?>
<pkw xmlns = "schema:www.daimlerchrysler.com"
     nummer = "S-NE 4229"
     line = "avantgarde">
   <farbe name="amethystviolett"/>
   <rad> ... </rad>
   <rad> ... </rad>
   <rad> ... </rad>
   <rad> ... </rad>
   <gewicht>
      <leergewicht>1485</leergewicht>
      <zulGesamtgewicht>1935</zulGesamtgewicht>
   </gewicht>
</pkw>
```



Die Document Type Definition (DTD)

```
<!ELEMENT    pkw    (farbe, rad+, gewicht)>
<!ATTLIST    pkw
             nummer      CDATA #REQUIRED
             line        CDATA   #REQUIRED>
<!ELEMENT    rad         ANY>
<!ELEMENT    farbe       EMPTY>
<!ATTLIST    farbe       name CDATA #REQUIRED>
<!ELEMENT    gewicht
                         (leergewicht, zulGesamtgewicht)>
<!ELEMENT    leergewicht (#PCDATA)>
<!ELEMENT    zulGesamtgewicht (#PCDATA)>
```



Die Document Type Definition (DTD)

```
<!NOTATION non-empty SYSTEM
  "javascript:currentNode.value.length()>0;">
<!ELEMENT pkw (farbe, rad, rad, rad, rad, rad?, gewicht)>
<!ATTLIST pkw
      nummer NOTATION (non-empty) "non-empty"
      line CDATA #REQUIRED>
<!ELEMENT rad ANY>
<!ELEMENT farbe EMPTY>
<!ATTLIST farbe  name
         (Dunkelbau|Firnweiß|Magmarot|...)
    #REQUIRED >
<!ELEMENT gewicht
       (leergewicht, zulGesamtgewicht)>
<!ELEMENT leergewicht (#PCDATA)>
<!ELEMENT zulGesamtgewicht (#PCDATA)>
```

DTD-Charakteristika

- Festlegung der Dokumentstruktur
  - Elemente
  - Attribute
  - Notationen
  - Entitäten (Textmakro auf Dokumentebene)
  - Parametrische Entitäten (Textmakro auf DTD-Ebene)
- Festlegung des Inhaltsmodells
  - Datentypen
    - Zeichenketten-artig
    - Vorgabewerte
    - Auswahltypen
  - Schlüssel / Referenzen

DTD-Charakteristika

- Streng hierarchische Sichtweise
- *Die-ganze-Welt-ist-ein-Dokument*-Sicht
  (von SGML übernommen)
  - ... mit entsprechenden Auswirkungen auf Referenzen
- Keine (praktisch verwendbare) Namespace Unterstützung
- DTD ist keine XML-Sprache
  - => zusätzliche Werkzeuge notwendig
- es existiert keine DTD für DTDs
  - => Grammatik kann nicht validiert werden
- Angebotene Datentypen unzureichend
- Angebotene Strukturprimitiven unzureichend
- => Notwendige Konstrukte zum Ausdruck mächtigerer Semantik müssen aufwendig und proprietär realisiert werden

Erweiterungsforderungen an die DTD

Strukturell ...

- Namespace Integration (und damit XML 2nd edition)
- Steuerung der Auftretensreihenfolge
  (Beschränkung und kontrollierte Freigabe)
- Vererbung (Kopier- und Substitutionssemantik)
- Wiederverwendungsunterstützung
- Praktisch einsetzbarer Referenzierungsmechanismus

Inhaltlich ...

- Primitive Datentypen (int, float, boolean, ...)
- Binärstrukturen
- Komplexe Datentypen (date, Elemente, Strukturen, ...)
- Eigendefinierte lexikalische Datentypen
- Konsistenzsichernde Einschränkungen

Erweiterungsforderungen an die DTD

Datentypen ...

- Validierung
- Vereinfachte Applikationsprogrammierung
- Separiertes Typkonzept

Namensräume ...

- Keine Unterstützung in DTDs
- Notwendig für integrierte Anwendung verschiedener XML-Vokabulare

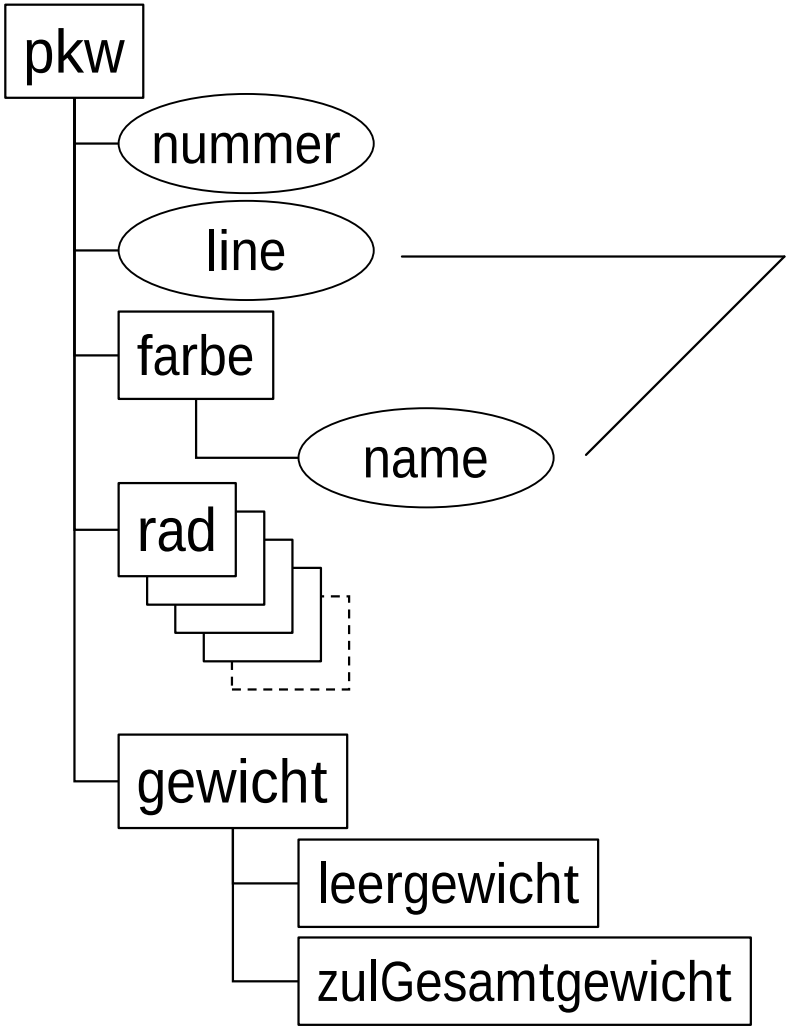
Syntax ...

- DTD-Syntax ist nicht XML
- Keine *Meta-DTD*

Gesteigerte Ausdrucksmächtigkeit ...

- Strukturen die nicht durch DTD-Mechanismen ausdrückbar sind

Erweiterungsforderungen am Beispiel



- Selbstdefinierte Typen (reguläre Ausdrücke)
- Kontextbezogene Einschränkung
- Auswahl aus vorgegebener Liste (selbst definierter Aufzählungstyp)
- Beschränkung der Auftretenshäufigkeit (Kardinalität)
- Einschränkung des zulässigen Wertebereichs (Domänenrestriktion)
- Beliebige Reihenfolge

Lösungsalternativen

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*)

Inspiriert durch XML-API-Entwicklung

- Schema for Object oriented XML (SOX)

XML-Sprachen zur Inhaltsbeschreibung

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

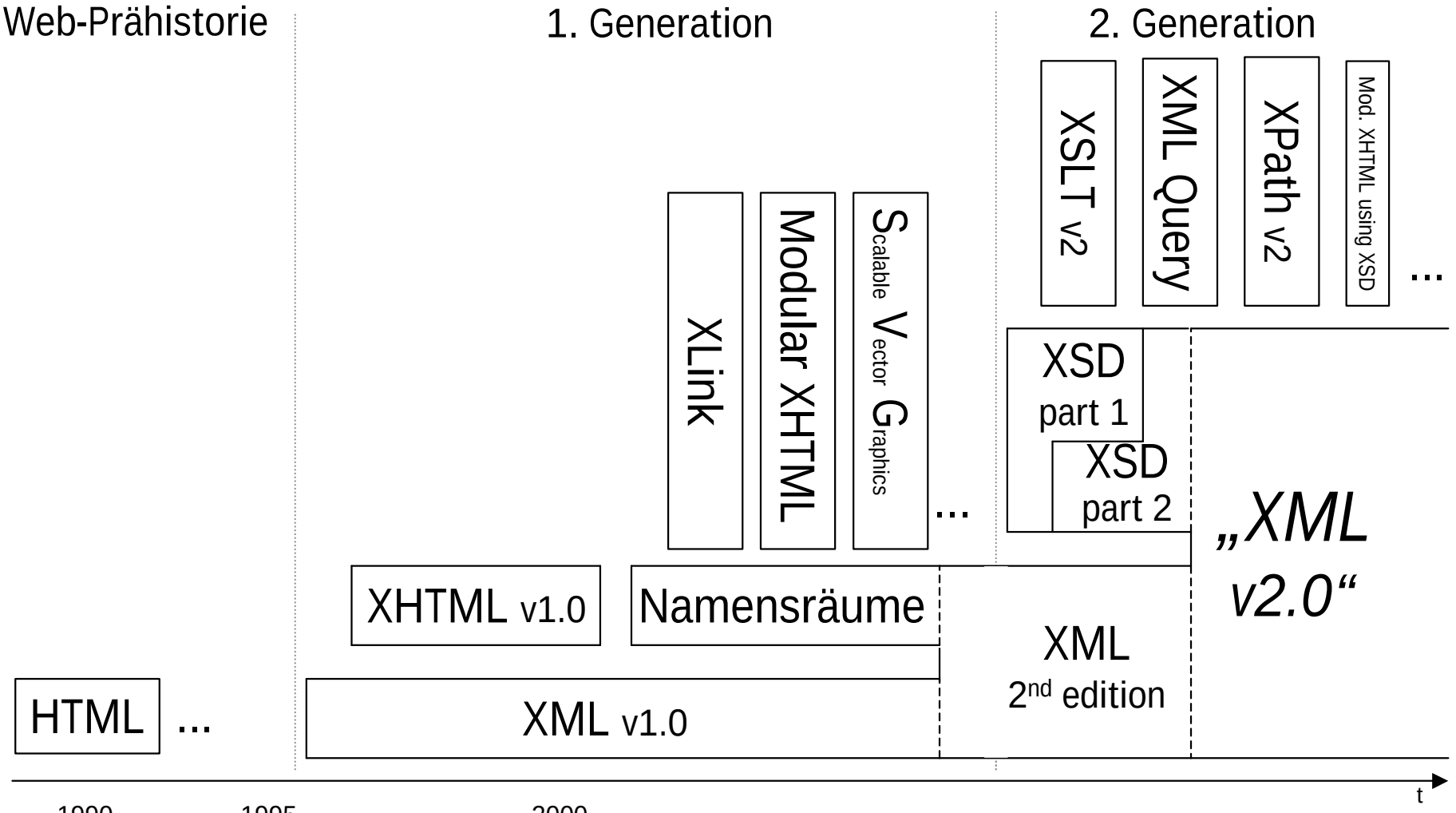
XML-Sprachen zur Inhaltsbeschreibung

Zwei konkurrierende Philosophien:

- **Regelbasiert**
  - Definiert Validierungsregeln
  - Bekannteste Implementierung: *Schematron*

- **Reguläre kontextfreie Grammatik**
  - Definiert alle gültigen Dokumentstrukturen explizit
  - Bekannt aus Programmiersprachen
  - Implementierungen: *XSD, XML-Data, XDR, DSD, ...*
  - Stand 1998/9: Viele verschiedene (konkurrierende) Sprachen
    - Zunehmend kaum noch erkennbare inhaltliche Unterschiede
  - Ab Mai 1999: W3C Arbeitsgruppe entwickelt eigenen Ansatz
    - Trennung zwischen *Struktur* und *Datentypen*
    - Berücksichtigt (nahezu) alle bedeutenden Vorgänger (explizit: DCD, DDML, SOX, XDR, XML-Data)

W3C's XML Schema -- ein Standard der 2. Generation

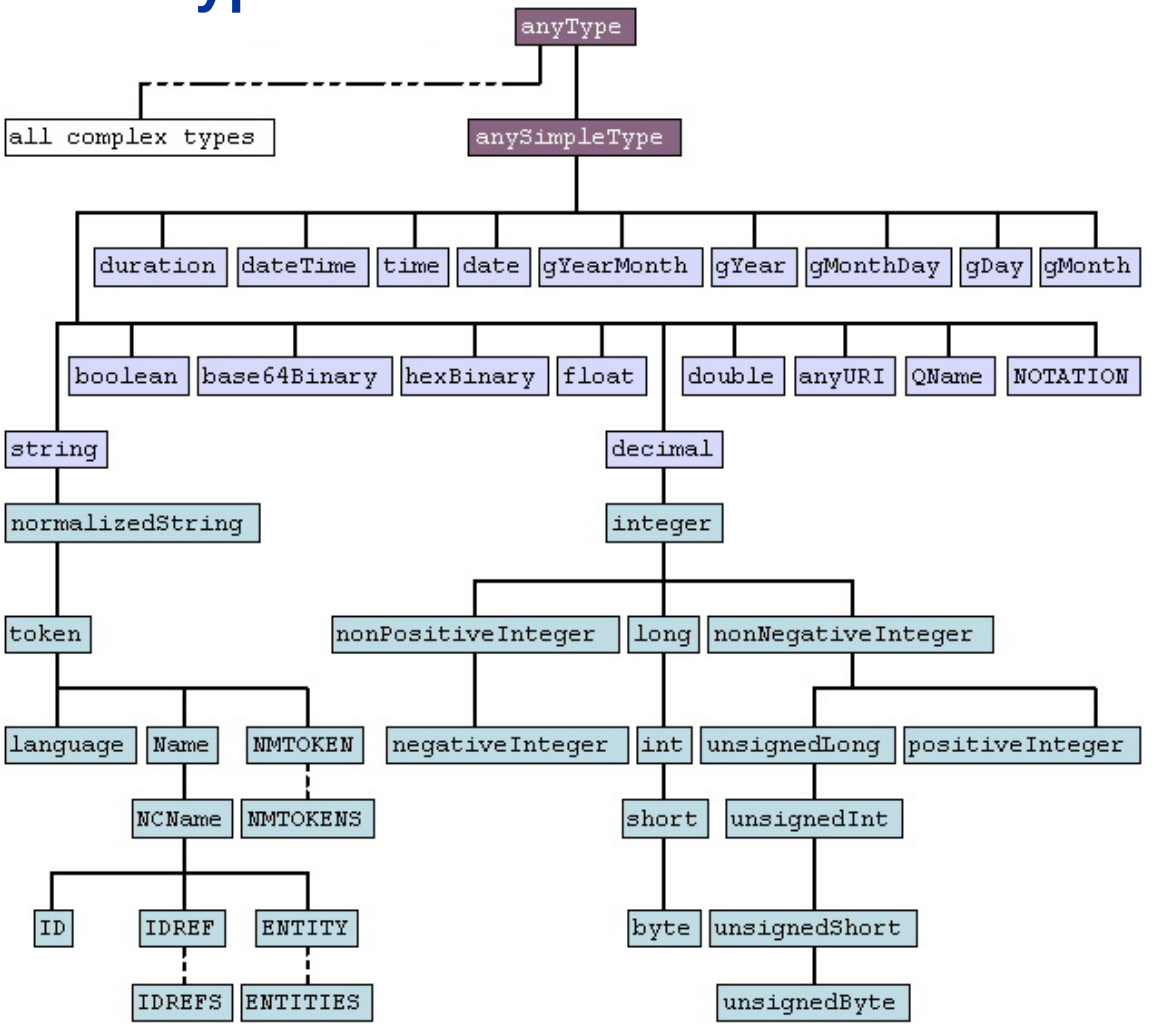


XML Schema: Mächtigkeit

- Attribute und Elemente (wie in DTDs)
- Namensraum-Unterstützung
- Atomare Datentypen (int, float, boolean, ...)
- Anwenderdefinierte
  - atomare Datentypen
    - Einschränkung des Wertebereichs (Domänenrestriktion)
  - lexikalische Muster (reguläre Ausdrücke)
  - Aufzählungstypen
  - Mengentypen
  - komplexe Datentypen (complexType)
- Vererbung
  - Restriktion und Erweiterung
- Substitution
- Erweiterter Schlüsselmechanismus
- NULL-Werte

XML Schema Part 2: Datentypen

- Ur-typ
- Vordefinierter Primitivtyp
- Vordefinierter abgeleiteter Typ
- Typeinschränkung
- Aggregierter Typ



XML Schema Part 2: DTD-Typen

ID	test, XYZ	XSD-Darstellung des DTD-Typen ID. Zugelassen sind alle Ausprägungen der Namespaceproduktion 4 (NCName). ID ist eine einschränkende Spezialisierung des Typs NCName
IDREF	test, XYZ	XSD-Darstellung des DTD-Typen IDREF. Zugelassen sind alle Ausprägungen der Namespaceproduktion 4 (NCName). IDREF ist eine einschränkende Spezialisierung des Typs NCName
IDREFS	test1 test2 test4, test3 test5	XSD-Darstellung des DTD-Typen IDREFS. Zugelassen sind Listen aus white space separierten Ausprägungen der Namespaceproduktion 4 (NCName). IDREFS ist eine nichtleere Aufzählung von IDREF-Ausprägungen
ENTITY		XSD-Darstellung des DTD-Typen ENTITY. Zugelassen sind alle Satzformen die der Produktion NCName der XML-Namensräume entsprechen und als ungeparste Entität definiert sind. ENTITY ist eine einschränkende Spezialisierung des Typs NCName
ENTITIES		XSD-Darstellung des DTD-Typen ENTITIES. Zugelassen sind Listen aus white space separierten Ausprägungen des Typs ENTITY. ENTITIES ist eine nichtleere Aufzählung von ENTITY-Ausprägungen
NOTATION		XSD-Darstellung des DTD-Typen NOTATION. Zur Verwendung dieses Typs in einem Schema muß eine Ableitung von NOTATION durch den Anwender definiert werden.
NMTOKEN	US, Deutschland	XSD-Darstellung des DTD-Typen NMTOKEN. Ausprägungen dieses Typs müssen konform zur Produktion 7 der XML-Spezifikation sein. NMTOKEN ist eine einschränkende Spezialisierung des Typs token
NMTOKENS	US UK Aus, Ger	XSD-Darstellung des DTD-Typen NMTOKENS. Zugelassen sind Listen aus white space separierten Ausprägungen des Typs NMTOKEN. NMTOKENS ist eine nichtleere Aufzählung von NMTOKEN-Ausprägungen

XML Schema Part 2: Text-artige Datentypen

string	Hello
World	Jedes beliebige Unicode Symbol gemäß XML-Syntaxproduktion 2
normalizedString	&#20Hello&#20;World	Jedes beliebige Unicode Symbol außer Zeilenvorschub, Wagenrücklauf und Tabulatoren. normalizedString ist eine einschränkende Spezialisierung des Typs string
token	Hello World	Jeder normalizedString, unter Weglassung führender, abschließender und mehrfacher Leerzeichen (#x20), sowie Zeilenvorschüben (#xA) und Tabulatoren (#x9). token ist eine einschränkende Spezialisierung des Typs normalizedString
Name	aName, _helloWorld, :notThatGood	XML Name gemäß Syntaxproduktion 5. Name ist eine einschränkende Spezialisierung des Typs token
QName	xsd:element, xsl:template, foo	Durch Namensraumpräfix qualifizierter Name gemäß Produktion 6 der XML Namespace Recommendation
NCName	aName, _anotherName, X	Name, der keinen Doppelpunkt enthält (non colonized name), gemäß Produktion 4 der XML Namespace Recommendation

XML Schema Part 2: numerische Datentypen

decimal	-1.23, 12678967.543233, +100000.00, 210	Wertebereich: $i*10^{-n}$, mit i, n aus integer, n>=0 Ein Prozessor muß mindestens 18 Dezimalstellen unterstützen
long	-1, 0, 12678967543233, +100000	Wertebereich: 2^{63} <= long <= 2^{63}-1 long ist eine einschränkende Spezialisierung des Typs integer
int	-1, 0, 126789675, +100000	Wertebereich: -2^{31} <= int <= 2^{31}-1 int ist eine einschränkende Spezialisierung des Typs long
byte	-1, 42	Wertebereich: -2^{7} <= byte <= 2^{7}-1 byte ist eine einschränkende Spezialisierung des Typs short
integer	-1, 0, 12678967543233, +100000	Wertebereich: entspricht der mathematischen Menge der ganzen Zahlen ($\mathbb{Z}$) integer ist eine einschränkende Spezialisierung des Typs decimal

XML Schema Part 2: numerische Datentypen

nonNegativeInteger	1, 0, 12678967543233, +100000	Wertebereich: 0 <= nonNegativeInteger nonNegativeInteger ist eine einschränkende Spezialisierung des Typs integer
positiveInteger	1, 12678967543233, +100000	Wertebereich: entspricht der mathematischen Menge der natürlichen Zahlen ($\mathbb{N}$) positiveInteger ist eine einschränkende Spezialisierung des Typs nonNegativeInteger
negativeInteger	-1, -12678967543233, -100000	Wertebereich: {..., -2, -1}, die unendliche Menge der negativen Zahlen negativeInteger ist eine einschränkende Spezialisierung des Typs nonPositiveInteger
unsignedLong	0, 12678967543233, 100000	Wertebereich: 0 <= unsignedLong <= 2^{64}-1 unsignedLong ist eine einschränkende Spezialisierung des Typs nonNegativeInteger
unsignedInt	0, 1267896754, 100000	Wertebereich: 0 <= unsignedInt <= 2^{32}-1 unsignedInt ist eine einschränkende Spezialisierung des Typs unsignedLong
unsignedShort	0, 12678, 10000	Wertebereich: 0 <= unsignedShort <= 2^{16}-1 unsignedShort ist eine einschränkende Spezialisierung des Typs unsignedInt
unsignedByte	0, 126, 100	Wertebereich: 0 <= unsignedByte <= 2^{8}-1 unsignedByte ist eine einschränkende Spezialisierung des Typs unsignedShort

XML Schema Part 2: kalendarische Typen

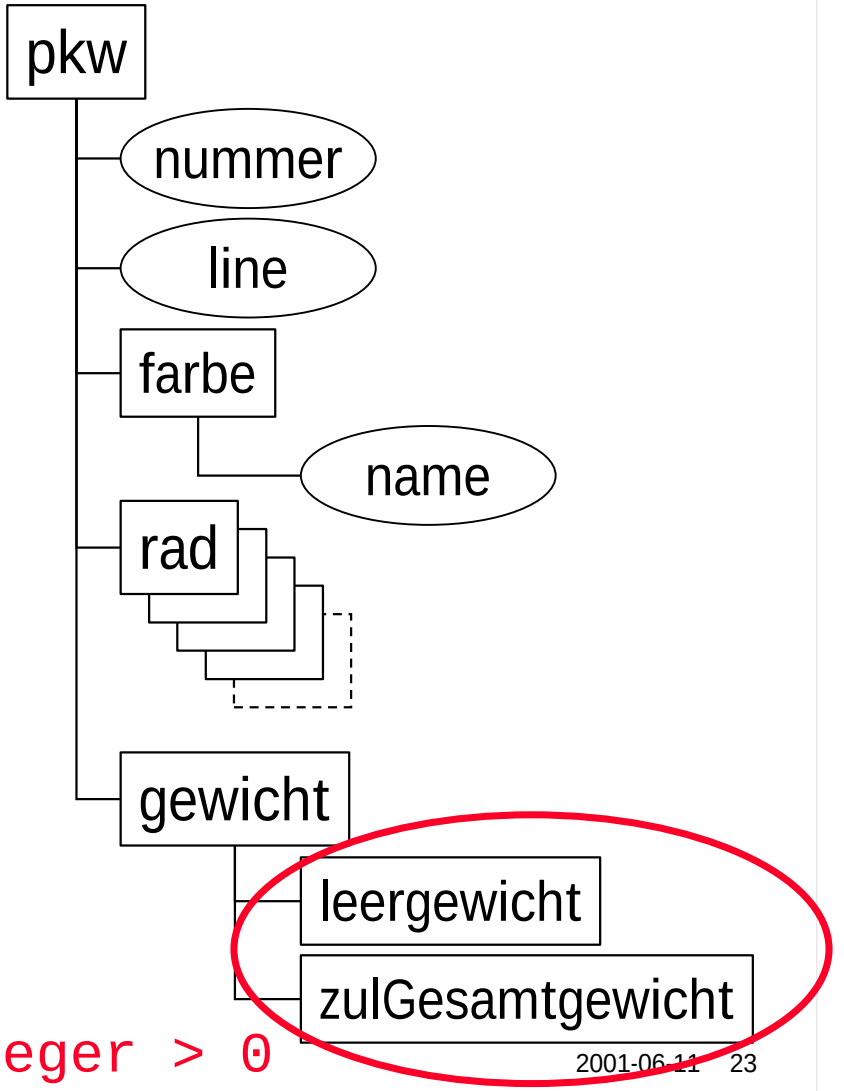
time	13:20:00-05:00, 13:20:00.000	Uhrzeit, die täglich wiederkehrt, ausgedrückt im Format gemäß ISO 8601
date	2001-06-11, 20010611	Datumsformat: CCYY-MM-DD, gemäß ISO 8601
dateTime	2001-06-11T14:00:00.000+02:00	Zeitpunkt, ausgedrückt durch Datum und Uhrzeit; beide gemäß ISO 8601 codiert.
gDay	---05, ---31	Darstellung eines wiederkehrenden Tages eines Monats gemäß ISO 8601
gMonth	--03--, --12--	Monatsformat: --MM-- gemäß ISO 8601
gYear	1999, 2000	Darstellung von Jahren des gregorianischen Kalenders gemäß ISO 8601
gYearMonth	2001-05	Darstellung eines Monats eines bestimmten Jahres des gregorianischen Kalenders gemäß ISO 8601
duration	P1Y2M3DT10H30M12.3S Zeitraum von eim Jahr, zwei Monaten, drei Tagen, zehn Stunden, 30 Minuten und 12,3 Sekunden	Nach Größe (Signifikanz) geordnete Koordinate im sechs-dimensionalen Raum aus Jahr, Monat, Tag, Stunde, Minute und Sekunde. Formatdefinition laut ISO 8601

XML Schema Part 2:

float double	-1E4, 1267.43233E12, 12.78e-2, 12, INF	32-Bit-Zahl mit einfacher Genauigkeit gemäß IEEE 754-1985. Wertebereich: $m * 2^e$, wobei m und e integer-Elemente mit $m <= 2^{24}$, und $-149 <= e < 104$ sind.
boolean	true, false, 1, 0	Unterstützung der klassischen zweiwertigen Logik
anyURI	http://www.jeckle.de foo:bar/baz	Jede gemäß IETF RFC 2396 bzw. IETF RFC 2732 gültige URI
language	en-GB, en, de-de	Sprachcodierung gemäß IETF RFC 1766 und XML Recommendation language identification. Die Identifikationsnamen werden durch ISO 639 sowie ISO 3166 definiert. language ist eine einschränkende Spezialisierung des Typs token
base64Binary	aGVsbG8gd29ybGQh	Base64-Darstellung eines beliebigen Binär-interpretierten Inhaltes gemäß IETF RFC 2045. Datenvolumen erhöht sich um Faktor 1,36
hexBinary	0FB7	Hexadezimale Darstellung beliebiger Binär-interpretierter Inhalte. Datenvolumen erhöht sich um Faktor 2
anyType	1, 2.3, aGVsb, 06b8f45, test&#20;für&#20;anyType &#0A; <sentence>the quick brown <animal>fox</animal>...	Allgemeinster Datentyp. Konzeptionell bildet er die Vereinigung aller angebotenen XSD-Typen.

XML Schema Part 1: Strukturen -- Elemente

```
<xsd:element name = "gewicht">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element
        name="leergewicht"
        type = "xsd:positiveInteger"/>
          <xsd:element
        name="zulGesamtgewicht"
        type="xsd:positiveInteger"/>
          </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```



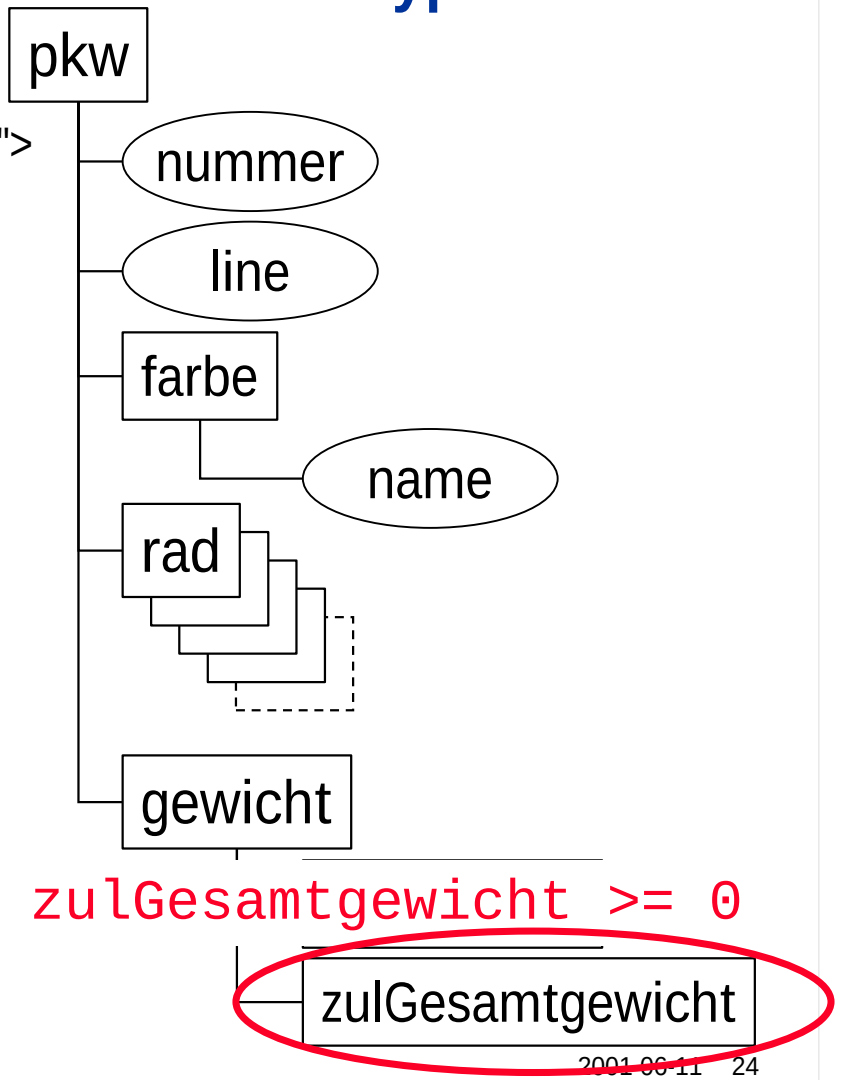
XML Schema Part 2: Anwenderdefinierte Datentypen

Definition:

```
<simpleType name="pkwZulGesamtgewichtType">
  <restriction base="decimal">
    <maxInclusive value="7500"/>
    <minInclusive value="0"/>
  </restriction>
</simpleType>
```

Verwendung:

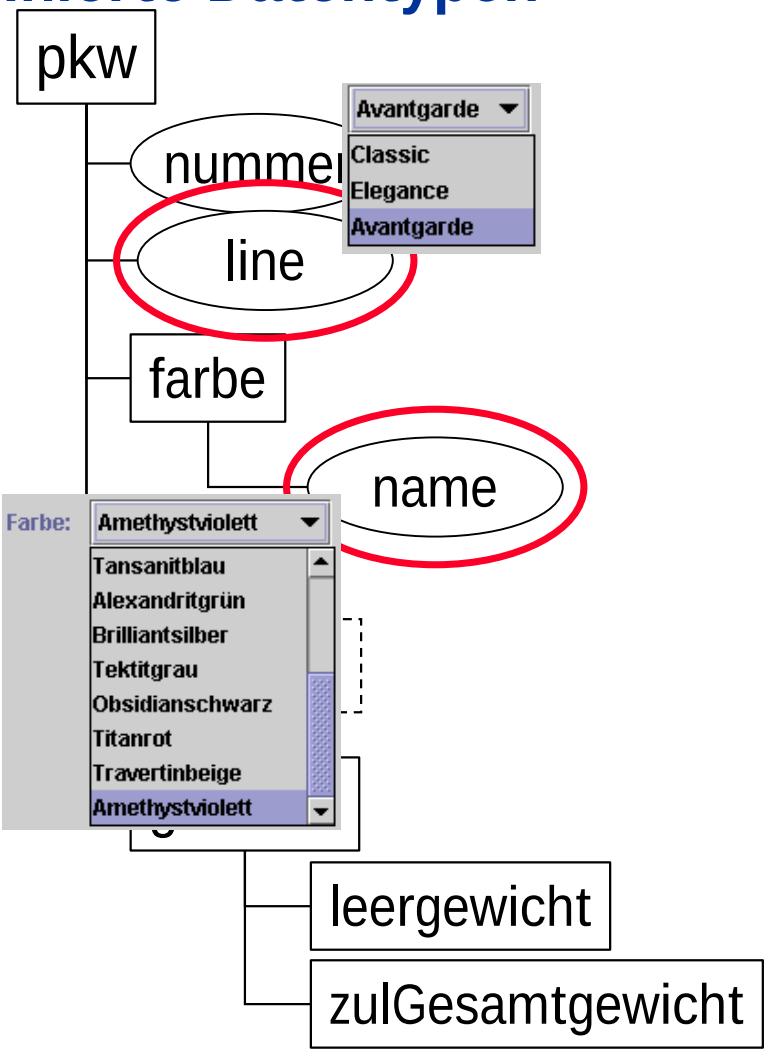
```
<element
  name = "leergewicht"
  type = "xsd:positiveInteger"/>
<element
  name = "zulGesamtgewicht"
  type = "dcx:pkwZulGesamtgewichtType"/>
```



XML Schema Part 2: Anwenderdefinierte Datentypen

Aufzählungstypen

```
<xsd:element name = "farbe">
  <xsd:complexType>
    <xsd:attribute
      name="name"
      use="required">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
<xsd:enumeration value = "Dunkelblau"/>
<xsd:enumeration value = "Firnweiß"/>
...
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:attribute>
  </xsd:complexType>
</xsd:element>
```



XML Schema Part 2: Anwenderdefinierte Datentypen

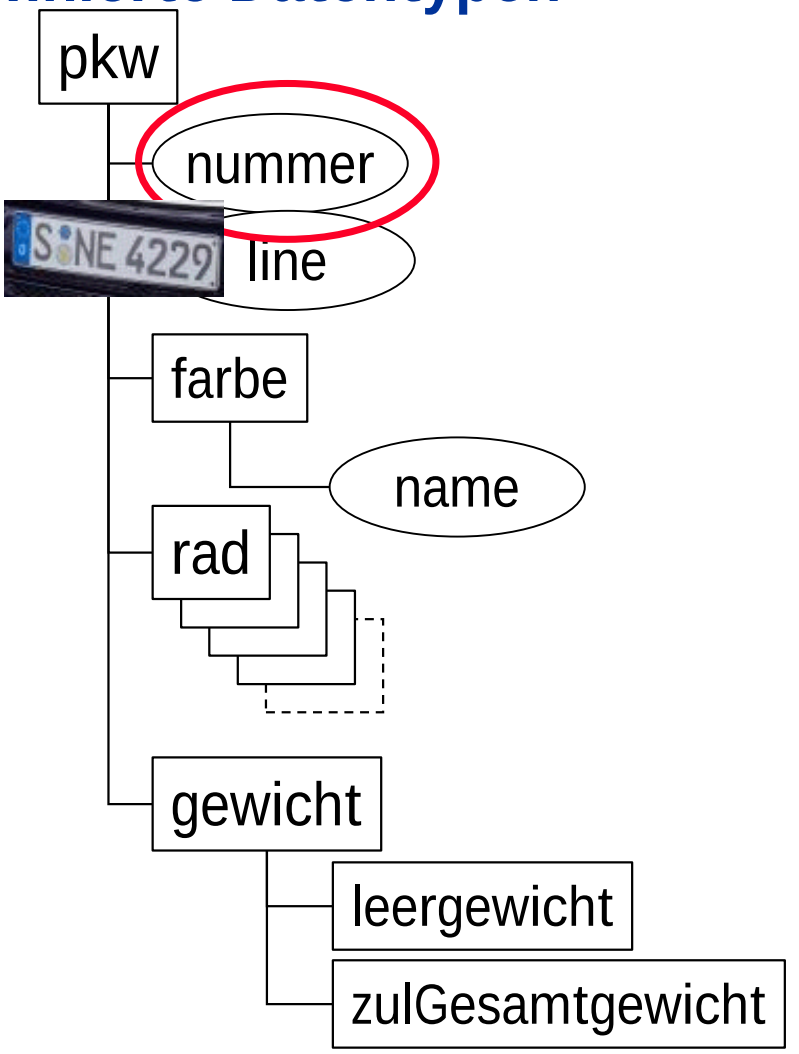
Aufzählungstypen

Definition:

```
<xsd:simpleType name = "autoNummerType">
  <xsd:restriction base = "xsd:string">
    <xsd:pattern
    value="[A-Z]{1,3}-[A-Z]{1,2} \d{1,4}"/>
  </xsd:restriction>
</xsd:simpleType>
```

Verwendung:

```
<xsd:element
  name = "nummer"
  type = "autoNummerType"/>
```



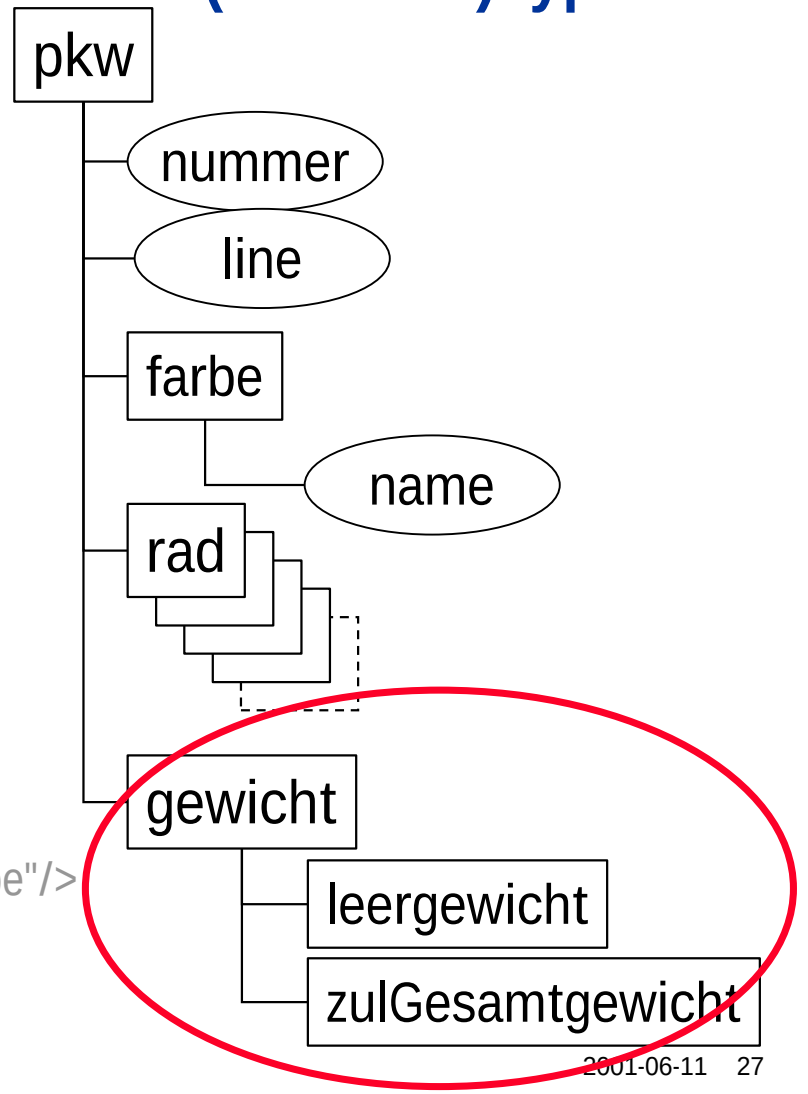
XML Schema Part 1: Anwenderdefinierte (Element-)Typen

Definition:

```
<complexType name="gewichtType">
  <all>
    <element
     name = "leergewicht"
     type = "xsd:positiveInteger"/>
    <element
     name = "zulGesamtgewicht"
     type = "dcx:pkwZulGesamtgewichtType"/>
  </all>
</complexType>
```

Verwendung:

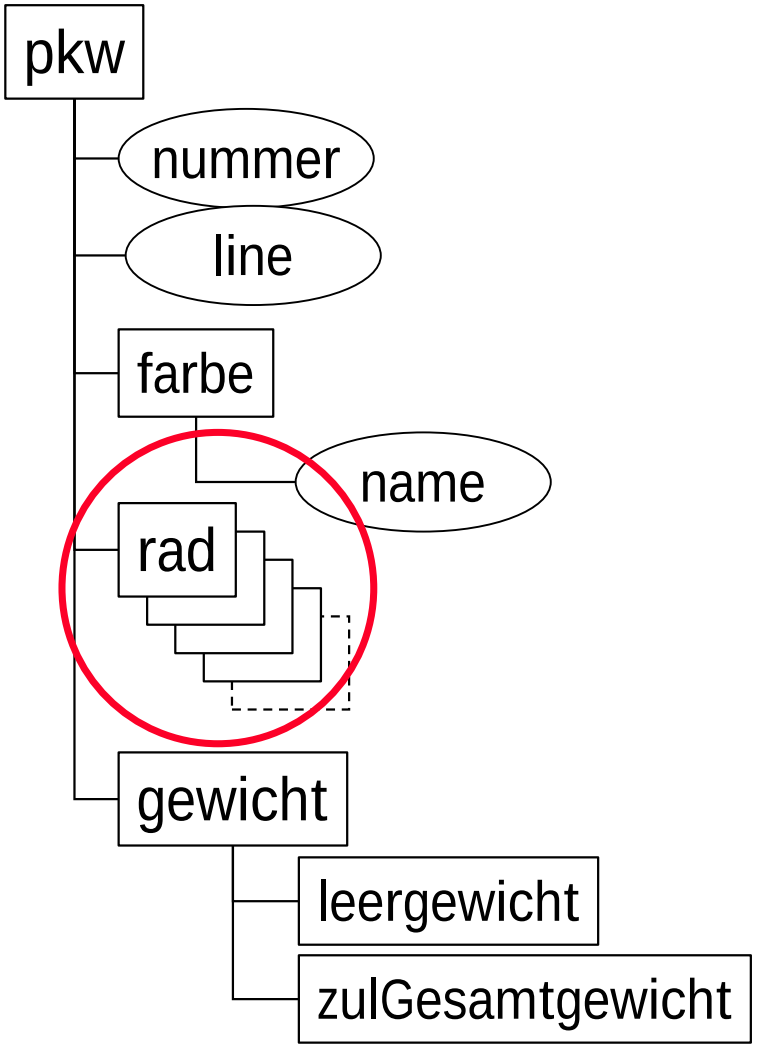
```
<element
 name = "nummer" type = "dcx:autoNummerType"/>
<element
 name = "gewicht" type = "dcx:gewichtType"/>
```



XML Schema Part 1: Strukturen

Steuerung der Auftrittshäufigkeit

```
<xsd:element name = "pkw">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element
       name = "rad"
       minOccurs = "4"
       maxOccurs = "5"/>
...
</xsd:sequence>
```

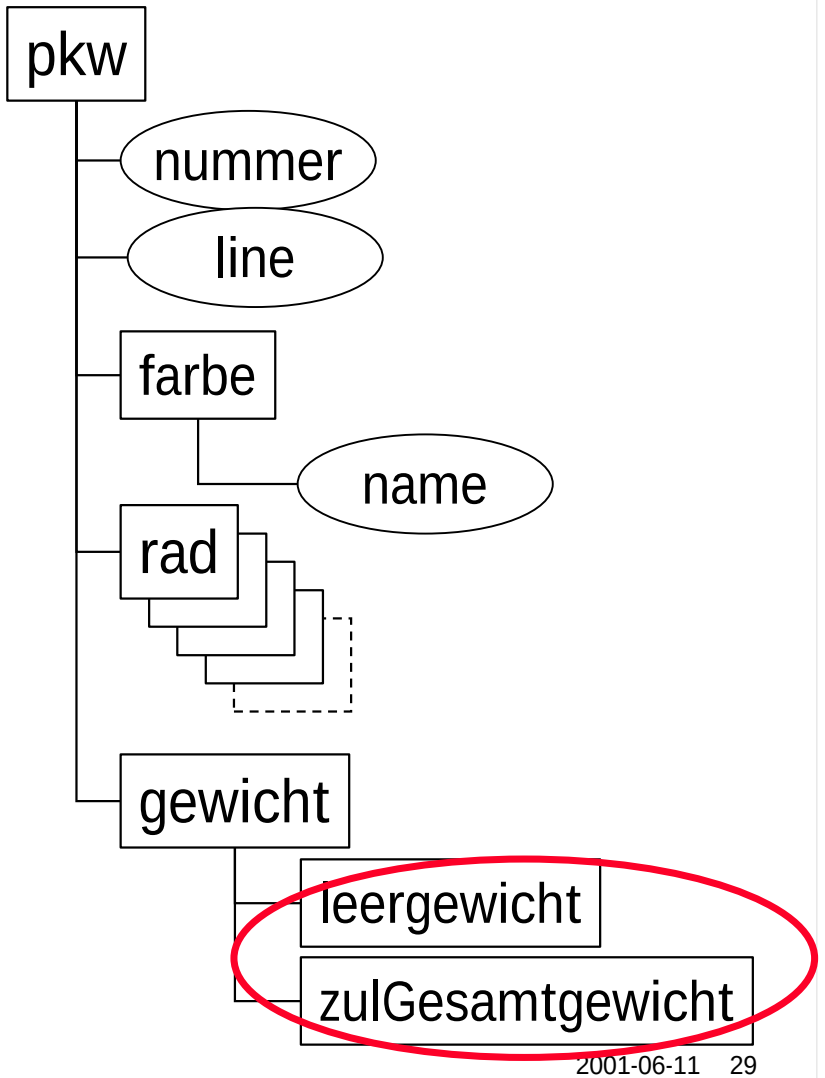


XML Schema Part 1: Strukturen

Steuerung der Auftrittsreihenfolge

```
<xsd:element name="gewicht">
  <xsd:complexType>
    <xsd:all>
      <xsd:element
        name = "leergewicht"
        type = "xsd:positiveInteger"/>
            <xsd:element
        name = "zulGesamtgewicht"
        type = "xsd:positiveInteger"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

Alle Elemente einer *all*-Gruppe können in beliebiger Reihenfolge auftreten.



XML Schema: Namensraumintegration

```
<?xml version = "1.0" encoding = "UTF-8"?>
<pkw
  xmlns = "schema:www.daimlerchrysler.com"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "schemas.dcx.com?name=example?type=XSD"
  nummer = "S-NE 4229"
  line = "Avantgarde">
    <rad/>
    <rad/>
    <rad> ... </rad>
    <rad> ... </rad>
    <farbe name="Amethystviolett"/>
    <gewicht>
      <leergewicht>1485</leergewicht>
      <zulGesamtgewicht>1935</zulGesamtgewicht>
   </gewicht>
</pkw>
```

XML Schema: Namensraumintegration

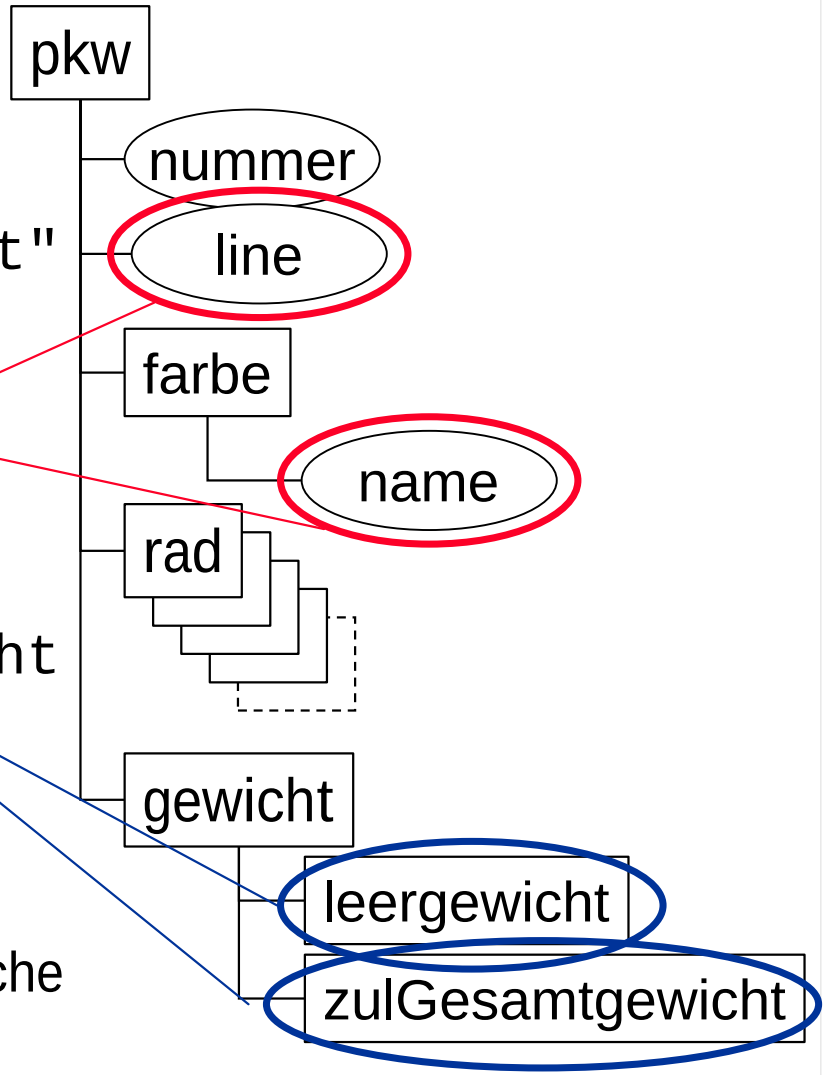
```
<schema xmlns = "http://www.w3.org/2001/XMLSchema"
          targetNamespace = "schema:www.daimlerchrysler.com"
          xmlns:dcx = "schema:www.daimlerchrysler.com"
          xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
          elementFormDefault = "qualified">
  <simpleType name="autoNummerType">...</simpleTyple>
  <element name = "nummer"
          type = "dcx:autoNummerType"/>
  <element name = "gewicht">
    <complexType>
      <all>
        <element name="leergewicht"
          type="xsd:positiveInteger"/>
        <element name="zulGesamtgewicht"
          type="dcx:pkwZulGesamtgewichtType"/>
      </all>
    </complexType>
  </element>
...</schema>
```

Grenzen von XML Schema

Konsistenzregeln

```
farbe/@name = "Amethystviolett"
=>
line = "Avantgarde"
```

```
leergewicht <= zulGesamtgewicht
```



XSD erlaubt als kontextfreie reguläre Sprache
keine inhaltsabhängigen Konsistenzregeln

Grenzen von XML Schema

... und ihre Überwindung

farbe/@name = "Amethystviolett"
=>
line = "Avantgarde"

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<schema xmlns='http://www.ascc.net/xml/schematron'>
  <ns prefix='dcx' uri='schema:www.daimlerchrysler.com'/>
  <pattern name = 'Farbe Test'>
    <rule context = "dcx:pkw">
      <assert test = "@line ='Avantgarde' and
       (//dcx:farbe/@name='Amethystviolett'
       or //dcx:farbe/@name='Dunkelblau'
       or //dcx:farbe/@name='Firnweiß'
       ...
       )">Farbe Amethystviolett ist nur
          für Line Avantgarde zugelassen</assert>
     </rule>
  </pattern>
</schema>
```

Beispiel in der Schemasprache *Schematron*

Grenzen von XML Schema

... und ihre Überwindung

leergewicht <= zulGesamtgewicht

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<schema xmlns='http://www.ascc.net/xml/schematron'>
  <ns prefix='dcx' uri='schema:www.daimlerchrysler.com'/>
  <pattern name = 'Gewicht Test'>
    <rule context = "dcx:gewicht">
      <assert test = "leergewicht &gt; zulGesamtgewicht">
      Leergewicht übersteigt zulässiges Gesamtgewicht
      </assert>
    </rule>
  </pattern>
</schema>
```

Beispiel in der Schemasprache *Schematron*

Referenzen

W3C's XML-Schema:

- Part 0: Primer: http://www.w3.org/TR/xmlschema-0/
- Part 1: Structures: http://www.w3.org/TR/xmlschema-1/
- Part 2: Datatypes: http://www.w3.org/TR/xmlschema-2/

Einige Schema-Ansätze:

- DT4DTD: http://www.w3.org/TR/dt4dtd
- DCD: http://www.w3.org/TR/NOTE-dcd
- SOX: http://www.w3.org/TR/NOTE-SOX/
- DDML: http://www.w3.org/TR/NOTE-ddml
- XML-Data: http://www.w3.org/TR/1998/NOTE-XML-data/
- XDR: http://www.ltg.ed.ac.uk/~ht/XMLData-Reduced.htm
- DSD: http://www.brics.dk/DSD/
- Schematron: http://www.ascc.net/xml/resource/schematron/schematron.html