

# DAIMLERCHRYSLER

## **Die eXtensible Markup Language**

Stand der Technik, Vokabulare und aktuelle Aktivitäten

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

### I Stand der Technik

- Einführung und Herkunft
- XML-Schema
- XML und das Web

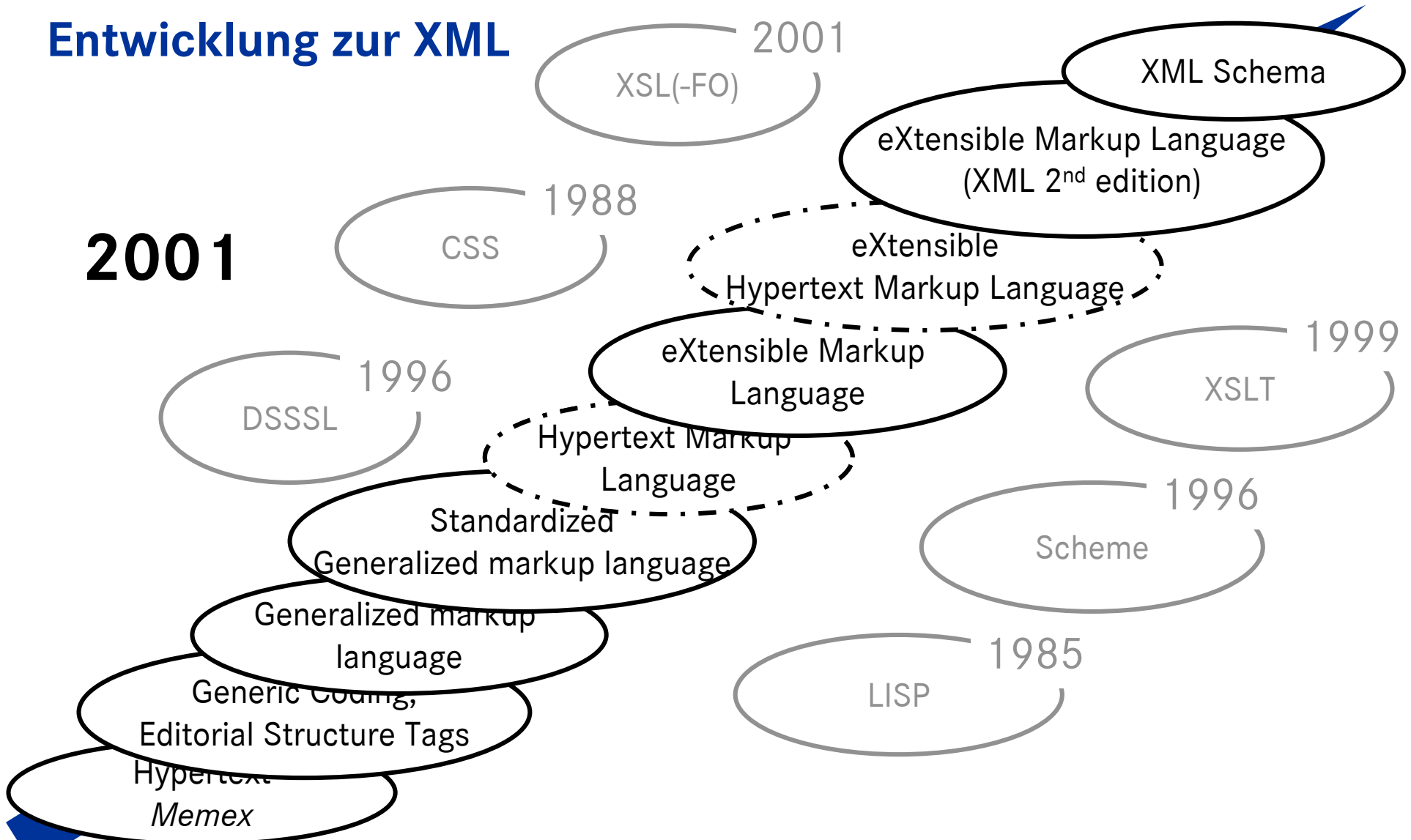
### II Vokabulare

- Architekturelle Aspekte
- Erzeugung neuer Vokabulare

### III Aktuelle Aktivitäten

- Semantic Web
- UML v2.0
- XML-basierte RPCs: SOAP/XMLP
- Anfragesprachen: XQuery

# Entwicklung zur XML

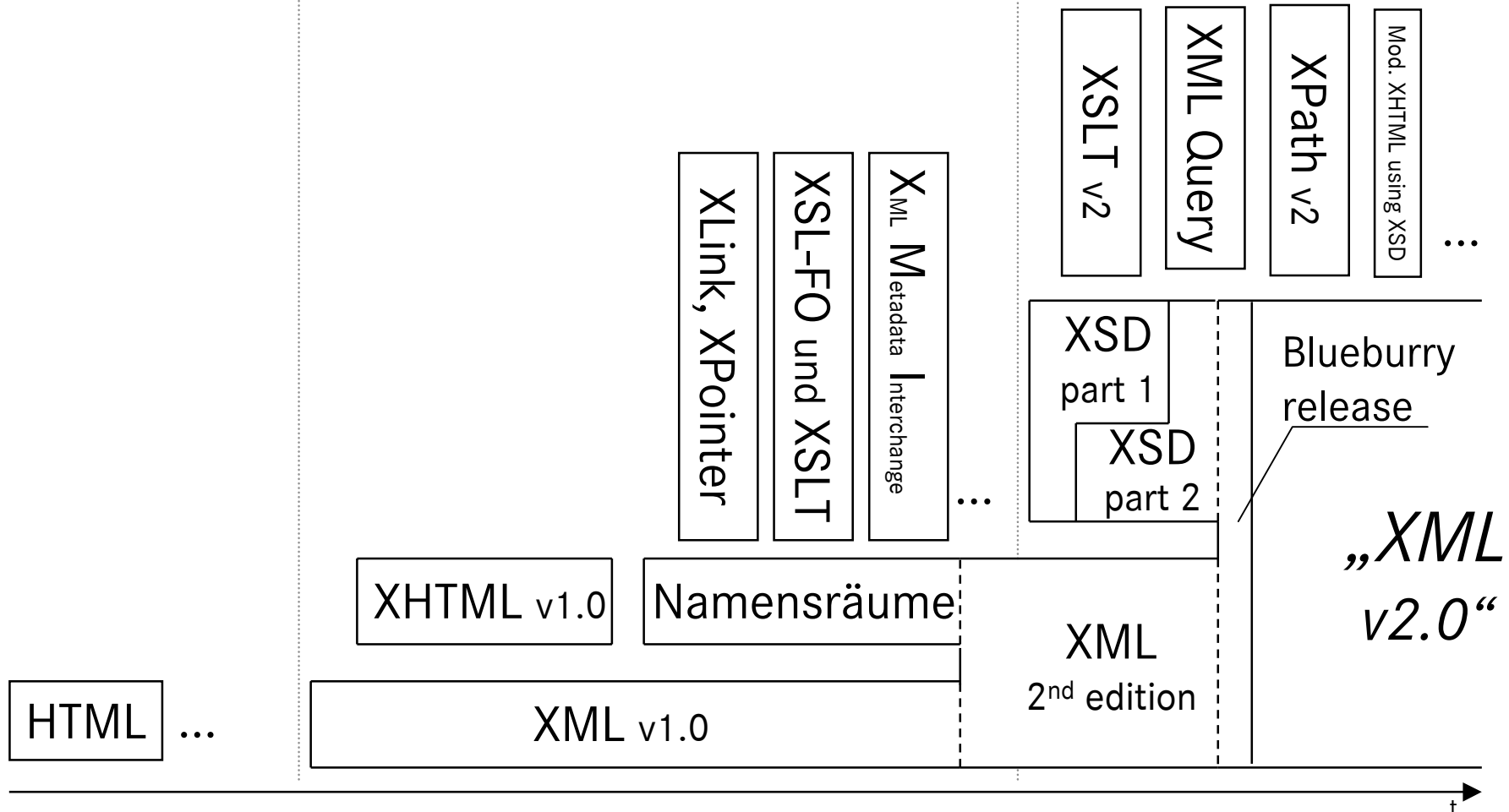


# XML-Generationen

Web-Prähistorie

1. Generation

2. Generation



## XML-Schema ... in zehn Punkten

- Erlaubt die Erstellung regulärer kontextfreier Grammatiken für XML-Vokabulare
- Part 1 beschreibt Strukturen und Inhaltseinschränkungen
- Part 2 definiert Datentypdefinition für Schema Part 1 und weitere XML-Vokabulare
- Signifikante Erweiterung der DTD-Mächtigkeit, wird diese langfristig ersetzen
- Ist eine XML-Sprache
- Integriert die wichtigsten konkurrierenden Vorgängeransätze
- Seit 2001-05-02 W3C Recommendation
- Basis aller W3C-Standards der zweiten Generation (XPath v2.0, XSLT v2.0, XHTML v2.0, XMLP, ...)
- Werkzeugunterstützung verfügbar
- Erster Schritt der Schema-Bestrebungen, weitere werden folgen ...

## XML-Schema ... ein Beispieldokument

```
<?xml version="1.0" encoding="UTF-8"?>
<Vortrag>
  <Titel>XML – Stand der Technik, Vokabulare und aktuelle Aktivitäten</Titel>
  <Veranstaltung datum="2001-06-26">
    <Name>Informatik Kolloquium der Universität der Bundeswehr</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-Schema ... Eine beschreibende Grammatik

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="Vortrag">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Titel" type="xsd:string"/>
        <xsd:element ref="Veranstaltung"/>
        <xsd:element ref="Referent"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="Veranstaltung">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="Name"/>
      </xsd:sequence>
      <xsd:attribute ref="datum"/>
    </xsd:complexType>
  </xsd:element>
  <xsd:attribute name="datum" type="xsd:date"/>

```

```

<xsd:element name="Name" type="xsd:string"/>
<xsd:element name="Referent">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Name"/>
      <xsd:element name="Firma" type="xsd:string"
        minOccurs="0"/>
      <xsd:element name="URL" type="xsd:anyURI"
        minOccurs="0"/>
      <xsd:element name="E-Mail" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:pattern value="\c{1,}@ \c{1,}.\c{2,3}"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:schema>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<Vortrag>
  <Titel>XML - Stand der Technik, Vokabulare und aktuelle Aktivitäten</Titel>
  <Veranstaltung datum="2001-06-26">
    <Name>Informatik Kolloquium der Universität der Bundeswehr</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-Schema ... Mächtigkeit

- Strukturell: Attribute und Elemente (wie in DTDs)
- Namespace-Unterstützung
- Atomare Datentypen (int, float, boolean, ...)
- Anwenderdefinierte
  - atomare skalare Datentypen (simpleType)
    - Einschränkung des Wertebereichs (Domänenrestriktion)
  - lexikalische Muster (reguläre Ausdrücke)
  - Aufzählungstypen
  - Mengentypen
  - komplexe Datentypen (complexType)
- Wiederverwendungsunterstützung
  - Vererbung: Restriktion und Erweiterung
  - Schemafragmente: Typen, Attribut- und Elementgruppen
- Substitution
- Erweiterter Schlüsselmechanismus
- NULL-Werte



# XML-Schema ... Datentypen

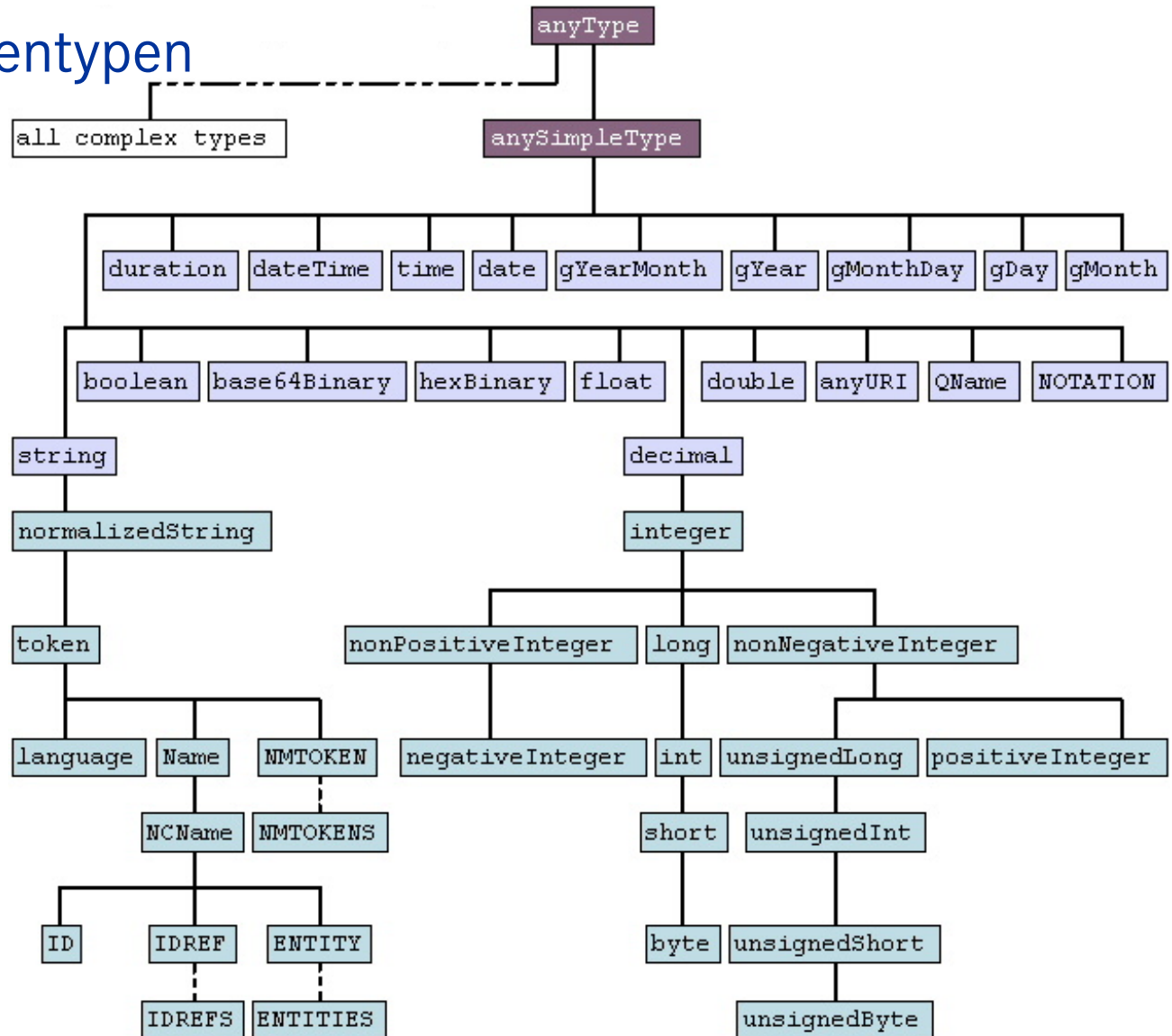
■ Ur-typ

□ Vordefinierter Primitivtyp

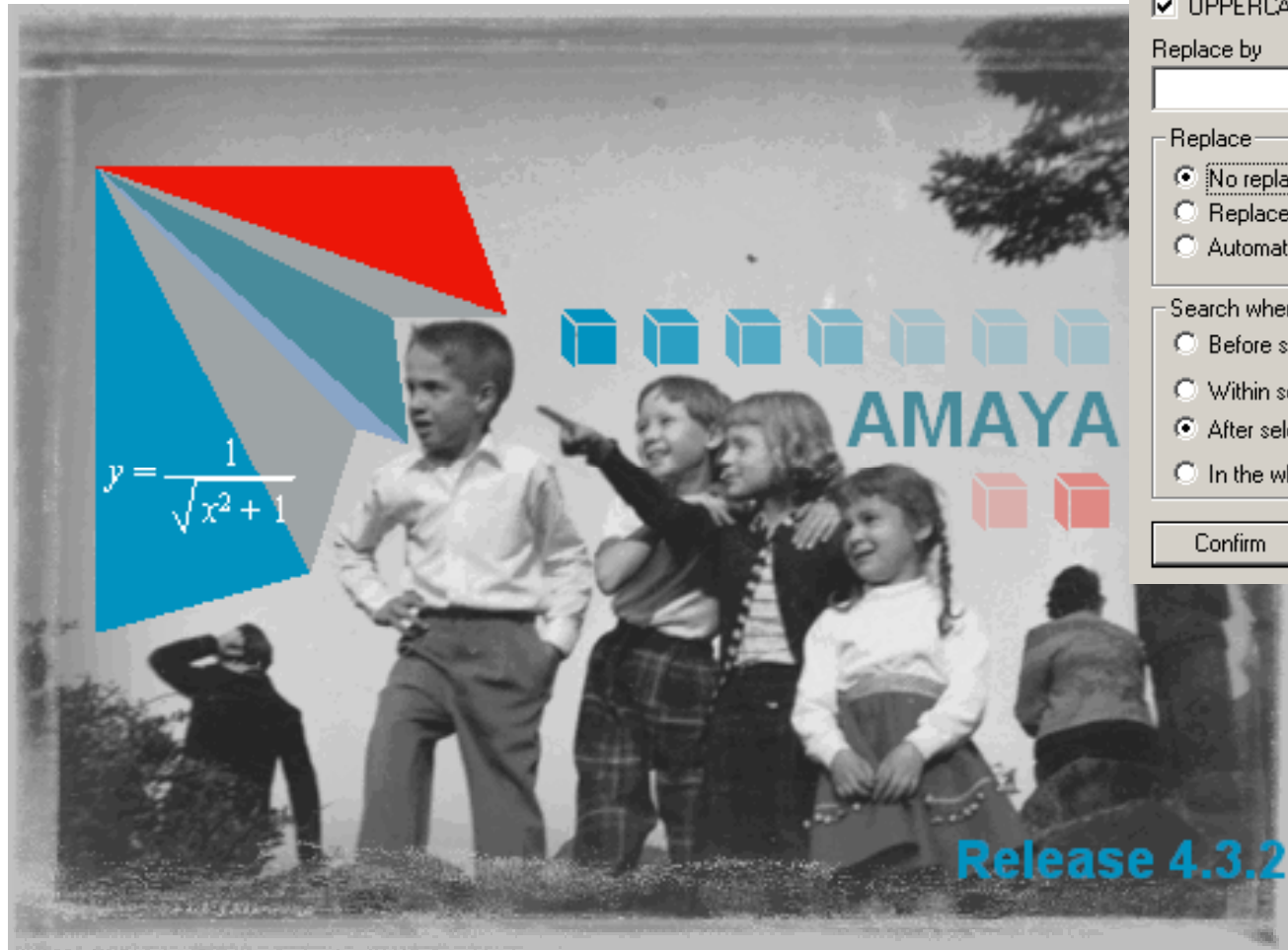
■ Vordefinierter abgeleiteter Typ

— Typeinschränkung

---- Aggregierter Typ



# XML und das Web: heute



Search for

UPPERCASE = lowercase

Replace by

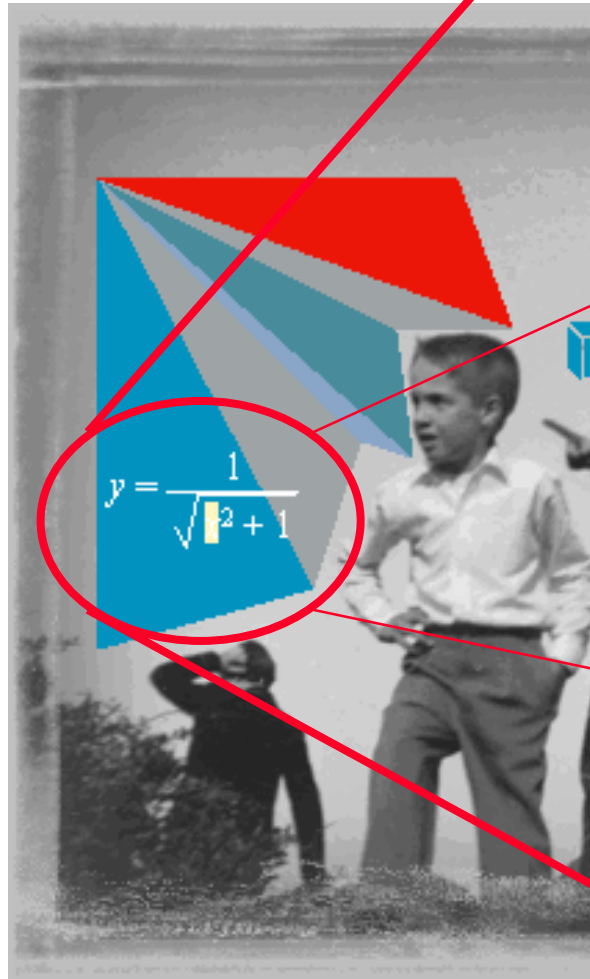
Replace

No replace  
 Replace on request  
 Automatic replace

Search where

Before selection  
 Within selection  
 After selection  
 In the whole document

## XML und das Web: heute



```

<math xmlns="http://www.w3.org/1998/Math/MathML"
  style="color: #FFFFFF; font-family: times">
  <mrow>
    <mi>y</mi>
    <mo>=</mo>
    <mfrac>
      <mn>1</mn>
      <msqrt>
        <mrow>
          <msup>
            <mi>x</mi>
            <mn>2</mn>
          </msup>
          <mo>+</mo>
          <mn>1</mn>
        </mrow>
      </msqrt>
    </mfrac>
  </mrow>
</math>

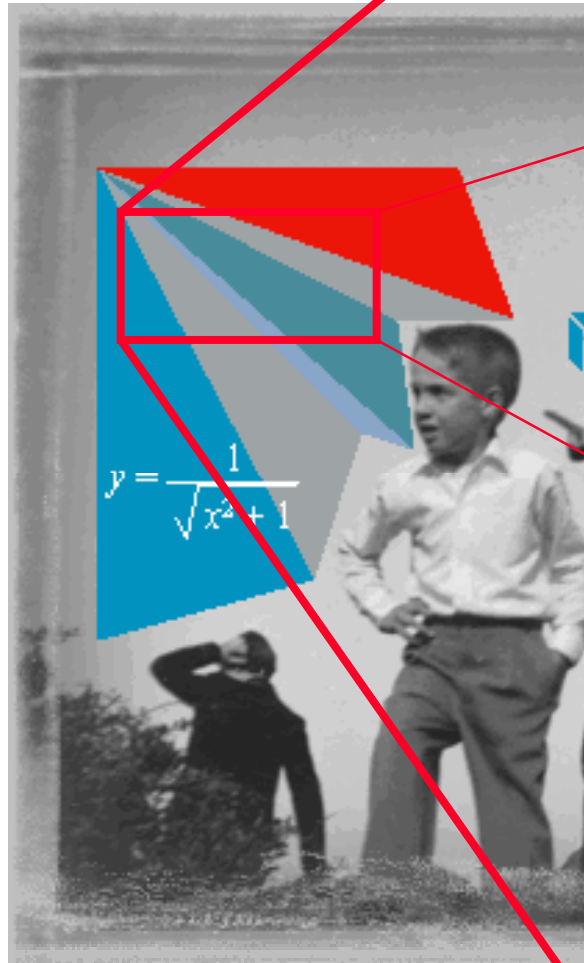
```

# XML und das Web heute

- ✓ Show buttonbar Alt v
  - ✓ Show address Alt v
  - Show map areas Alt v
  - Show targets Alt v
  - ✓ **Zoom in Alt =**
  - Zoom out Alt -
- 
- Show parsing errors
  - Show structure Alt v
  - Show alternate Alt v
  - Show links Alt v
  - Show table of contents Alt v
  - Show source Alt v



## XML und das Web: heute



```

<svg xmlns="http://www.w3.org/2000/svg" width="466"
height="349">
  <image id="backgroundImage" x="0" y="0" width="466" height="349"
    xlink:href="amaya.png">
    <desc>Leading the Web to its full potential</desc>
  </image>
  <defs>
    <symbol id="cube">
      <polygon points="0,1 5,6 5,21 0,16" stroke="1"/>
      <polygon points="1,0 16,0 21,5 6,5" stroke="1"/>
      <rect x="6" y="6" width="15" height="15" stroke="1"/>
    </symbol>
  </defs>
  <g id="cubes">
    <use x="200" y="112" xlink:href="#cube" style="fill: #0192BF"/>
    <use x="230" y="112" xlink:href="#cube" style="fill: #2B9FC3"/>
    <use x="260" y="112" xlink:href="#cube" style="fill: #53AAC5"/>
    <use x="290" y="112" xlink:href="#cube" style="fill: #7AB4C6"/>
    <use x="320" y="112" xlink:href="#cube" style="fill: #A3C0C9"/>
    <use x="350" y="112" xlink:href="#cube" style="fill: #A3C0C9"/>
    <use x="380" y="112" xlink:href="#cube" style="fill: #A3C0C9"/>
    <use x="410" y="112" xlink:href="#cube" style="fill: #EC1608"/>
    <use x="410" y="142" xlink:href="#cube" style="fill: #E2372C"/>
    <use x="350" y="172" xlink:href="#cube" style="fill: #DAAFAC"/>
    <use x="380" y="172" xlink:href="#cube" style="fill: #DF8984"/>
    <use x="410" y="172" xlink:href="#cube" style="fill: #DC5B53"/>
  </g>

```

...

# Das XML-Universum

## Sprachentwurf



- XML-Schema (XSD)
- DDML
- Schematron
- RELAX NG
- DCD

## Programmierung/ Schnittstellen

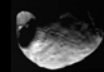


DOM

SAX

<?XML!>

## Editoren



## Anwendungen



XML-RPC

SOAP/XMLP

Werkzeugunterstützung

Protokolle

Stylesheet-, Linkingprozessoren, ...



## XML-Vokabulare

- Wichtigstes Element „von XML“:  
Vokabulare bilden Anwendung der Metasprache XML
- Syntax ist verhältnismäßig einfach zu erstellen  
→ Werkzeugunterstützung (Editoren)
- Überwiegend kostenfreie Verbreitung über das Internet
- Unterschiedlichste Herkunft  
→ Qualität, Zukunft, ...
- Integration in eigene Prozeß- und Systemlandschaft ...

## XML-Vokabulare ... und ihre Erzeugung

### ● Aktuelle Situation

- *Internet-Täglich* neue Vokabular-Definitionen
- verschiedenster Qualität ...
- Teilweise dokumentiert

### ● Existierende Lösungen

- Handcodierung (überwiegender Anteil)
- implizit algorithmisch (wenige)
- algorithmisch reproduzierbar (erste Ansätze)

### ● Lösungsansatz

- Interpretation der Vokabular- (besser: Sprach-)Erzeugung als Teil des Entwicklungsprozesses

## XML-Vokabulare ... und ihre Erzeugung

### ● Voraussetzungen

- Ein (Entwicklungs-)Prozeß muß existieren
- Modell-getriebener Entwicklungsprozeß

### ● Praktische Umsetzungen

- OMG XML Metadata Interchange's DTD Production Principles
- DaimlerChrysler Research's XML-Schema Generator

### ● Erfahrungen

- Gute Prozeßintegration (nahtlose Spracherzeugung möglich)
- Robustes Verfahren
- Technische Umsetzung hat massiven Einfluß!

## XML-Vokabulare ... und ihre Erzeugung: Herausforderungen

### **Flexibilität**

- Zukunftssicherheit des entstehenden Vokabulars  
(Adaptionsfähigkeit und verbundener Aufwand an veränderte Anforderungen)

### **Geschwindigkeit**

- Schnelle Erzeugung von XML-Vokabularen für A2A-, B2B-, B2C-Anwendungen  
(XML-Sprachen als Basis der Daten- und Prozeßintegration)

### **Kohärenz**

- Synchronisation zwischen Daten- und Prozeßstrukturen und XML-Format

### **Korrektheit**

- XML-Format soll Daten- und Prozeßstrukturen möglichst unverändert widerspiegeln

# XML-Vokabulare ... und ihre Erzeugung: Herausforderungen

## Stil

- XML-Format soll *look-and-feel* der Applikation aufweisen  
(*Look*: Visuelle Sprachaspekte, *Feel*: Technisch strukturelle Aspekte)

## Integration

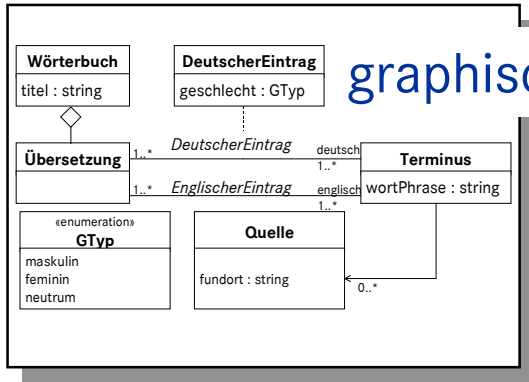
- Geringer zusätzlicher Aufwand durch XML-Vokabularerzeugung

## Wiederverwendung

- ... vorhandenen (Meta-)Wissens über Daten- und Prozeßstrukturen und Modellierung

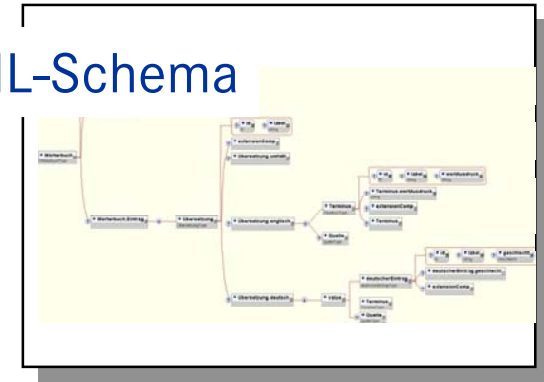


# XML-Vokabulare ... und ihre Erzeugung: technische Umsetzung



graphisches UML-Model

graphisches XML-Schema



Visualisierung



Export

```
<?xml version = '1.0' encoding = 'ISO-8859-1'?>
<XML xmi.version = '1.0' timestamp = 'Sat Jun 23 18:48:2
<XML.header>
<XML.documentation>
<XML.exporter>Unisys.JCR.1</XML.exporter>
<XML.exporterVersion>1.3.2</XML.exporterVersion>
</XML.documentation>
<XML.metamodel xmi.name = 'UML' xmi.version = '1.3'/>
</XML.header>
<XML.content>
<Model_Management.Model xmi.id = 'G.0' >
<Foundation.Core.ModelElement.name>example</Foundation.Core.ModelElement.name>
<Foundation.Core.ModelElement.visibility xmi.value = "public"/>
<Foundation.Core.ModelElement.isSpecification xmi.value = "false"/>
<Foundation.Core.GeneralizableElement.isRoot xmi.value = "false"/>
<Foundation.Core.GeneralizableElement.isLeaf xmi.value = "false"/>
<Foundation.Core.GeneralizableElement.isAbstract xmi.value = "false"/>
<Foundation.Core.Namespace.ownedElement>
<Foundation.Core.Class xmi.id = 'S.1' >
<Foundation.Core.ModelElement.name>W&#x00f6;rterbuch</Foundation.Core.ModelElement.name>
<Foundation.Core.ModelElement.visibility xmi.value = "public"/>
<Foundation.Core.ModelElement.isSpecification xmi.value = "false"/>
<Foundation.Core.GeneralizableElement.isRoot xmi.value = "true"/>
<Foundation.Core.GeneralizableElement.isLeaf xmi.value = "true"/>
...
```

XML Repräsentation

XML Schema (XSD)

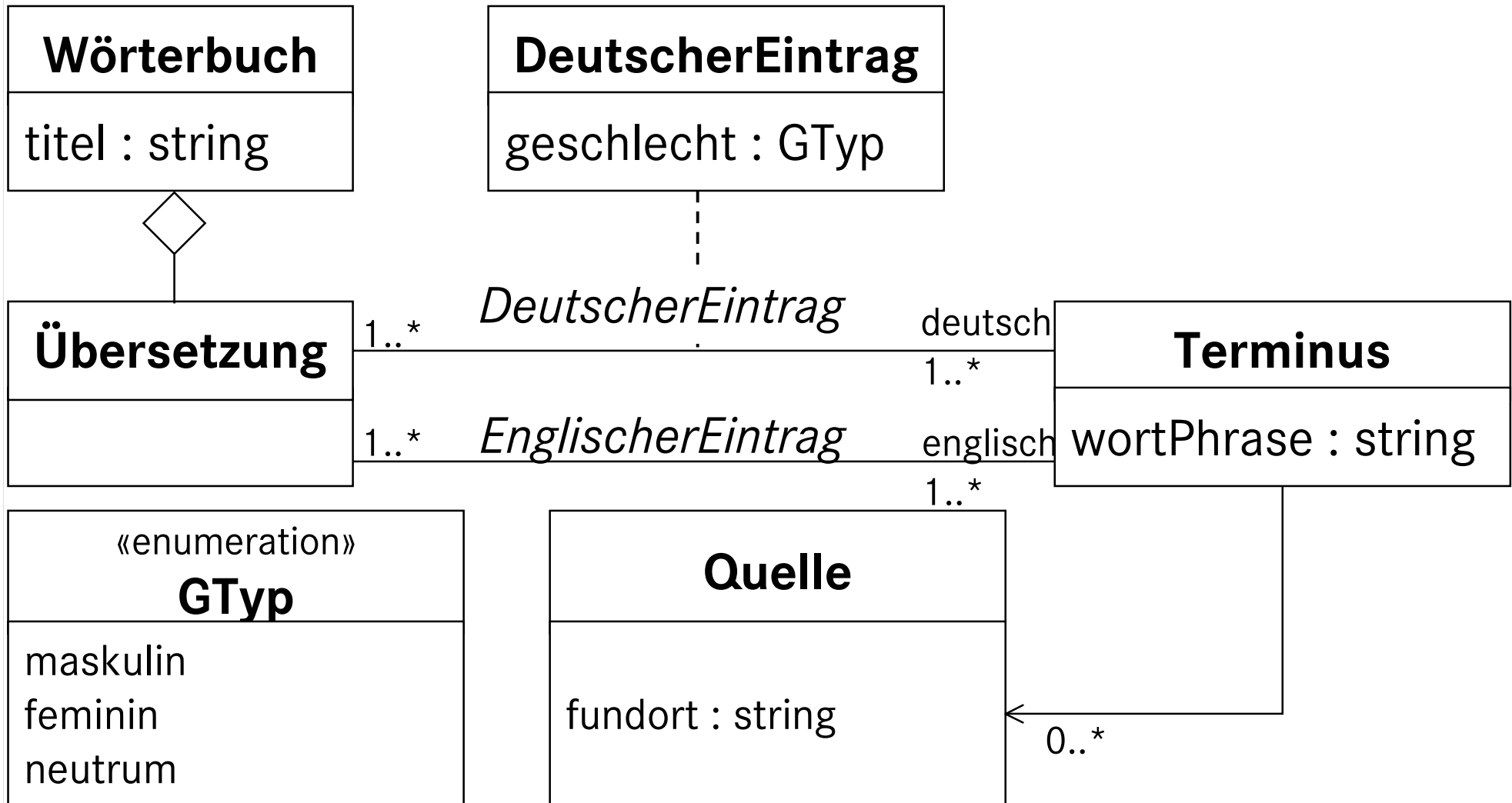
```
xOccurs = "unbounded">
ach"/>
ung"/>
<xsd:element ref = "Terminus"/>
<xsd:element ref = "Quelle"/>
</xsd:choice>
</xsd:complexType>
</xsd:element>
<xsd:element name = "Wörterbuch" type = "WörterbuchType"/>
<xsd:element name = "Übersetzung" type = "ÜbersetzungType"/>
<xsd:element name = "Terminus" type = "TerminusType"/>
<xsd:element name = "deutscherEintrag" type = "deutscherEintragType"/>
<xsd:element name = "Quelle" type = "QuelleType"/>
<xsd:complexType name = "WörterbuchType">
<xsd:all>
<xsd:element ref = "Wörterbuch.titel" minOccurs = "0"/>
<xsd:element ref = "extensionComp" minOccurs = "0"/>
<xsd:element name = "Wörterbuch.Eintrag" minOccurs = "0"/>
</xsd:complexType>
<xsd:choice maxOccurs = "unbounded">
<xsd:element ref = "Übersetzung"/>
</xsd:choice>
```



Schema-

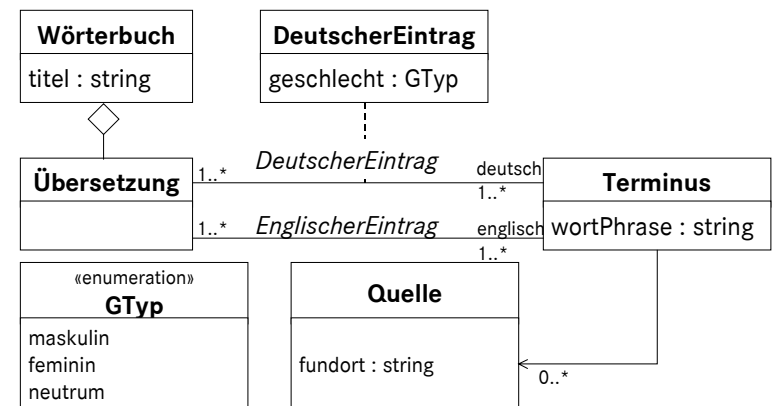
Erzeugung

## XML-Vokabulare ... und ihre Erzeugung: Am Beispiel



## XML-Vokabulare ... und ihre Erzeugung: Nutzbare UML-Konzepte

- Klassen
  - Vererbungsstrukturen
  - Assoziationsklassen
- Attribute
  - Konstanten
  - Vorgabewerte
  - Kardinalitäten (d.h. optionale und mengenwertige Attribute)
- Datentypen (skalare und komplexe)
  - vordefinierte (durch UML, MOF oder CORBA)
  - Anwender-definierte
- Assoziationen
  - Navigierbarkeit
  - Multiplizitäten/Kardinalitäten



## XML-Vokabulare ... und ihre Erzeugung: Strukturelle Aspekte

### Navigation

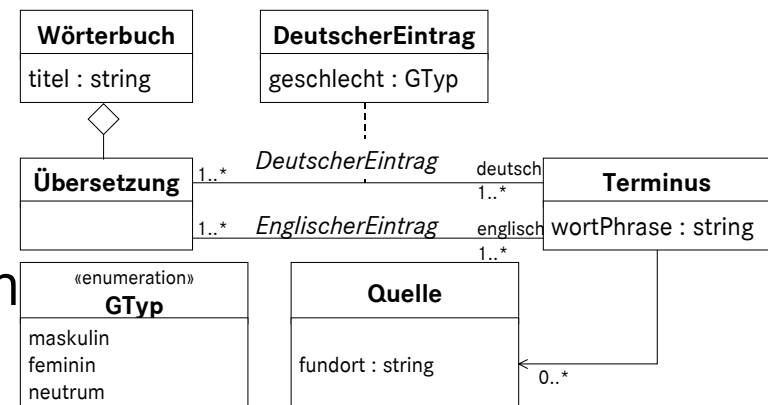
- Entlang der modellierten Assoziationspfade
- Umsetzung Netz-artiger Strukturen in XML-inhärente Baumsicht

### Kapselung und Identität

- Beachtung der Lokalität von Attributen
- Schaffung der Objektidentität in XML → Problem der Redundanz

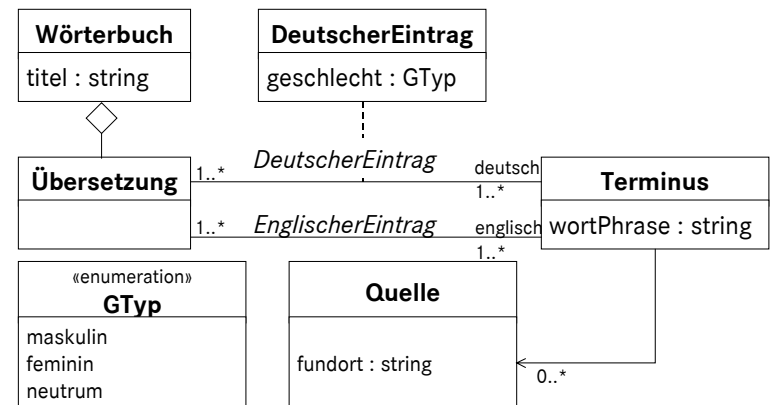
### Vererbung

- Einfachvererbung: direkte Übernahme
- Merhfachvererbung: geeignete Nachbildung



## XML-Vokabulare ... und ihre Erzeugung: Inhaltliche Aspekte

- Übersetzung der UML-/MOF-Datentypen gemäß ihrer Semantik
- Transformation strukturierter Typen in XSDs Analoga
- Erzeugung zusätzlicher Eigenschaften für zur Abbildung der Identität
- Lösung des Redundanzproblems durch Referenzierbarkeit



## XML-Vokabulare ... und ihre Erzeugung: Qualitätsaspekte

### Datentypen

- Abbildung der im Modell verwendeten (UML, CORBA, Business-spezifischen, ...) Datentypen auf die durch XML-Schema Part 2 angebotenen (Forderungen der Korrektheit und Kohärenz)

### Strukturen

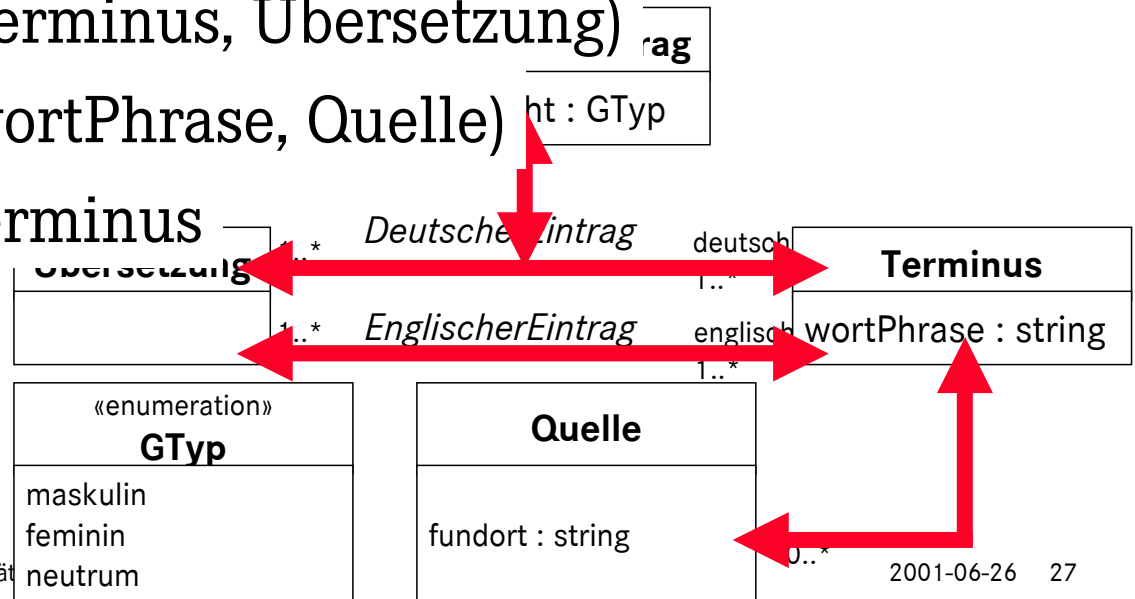
- Abbildung der modellierten (Business-)Strukturen auf die XML-inhärente Baumsicht (Forderungen der Korrektheit, Kohärenz und des Stils)

### Pragmatischer Aspekt

- Sicherheit, Reproduzierbarkeit und Zweckmäßigkeit der Umsetzung, insbesondere im Kontext existierender Systeme und Prozesse (Forderungen der Flexibilität, Geschwindigkeit, Integration und Wiederverwendung)

## XML-Vokabulare ... und ihre Erzeugung

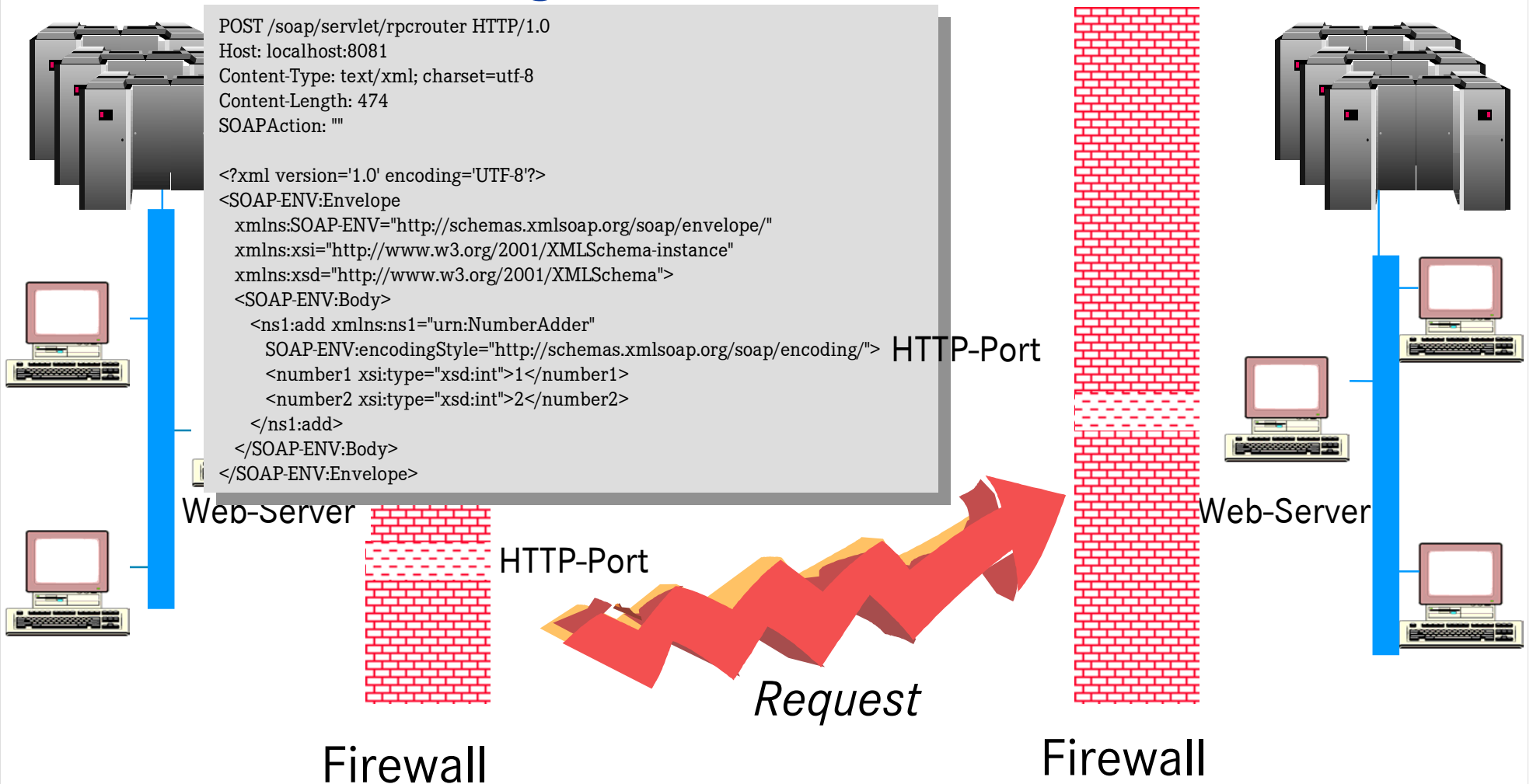
- Wörterbuch → Übersetzung, Wörterbuch
- Übersetzung → (Terminus.deutsch+, Terminus.englisch+, Wörterbuch)
- Terminus.deutsch → (DeutscherEintrag, Terminus, Übersetzung)
- DeutscherEintrag → (geschlecht, (Übersetzung, Terminus))
- Terminus.englisch → (Terminus, Übersetzung)
- Terminus → (wortPhrase, Quelle)
- Quelle → Terminus



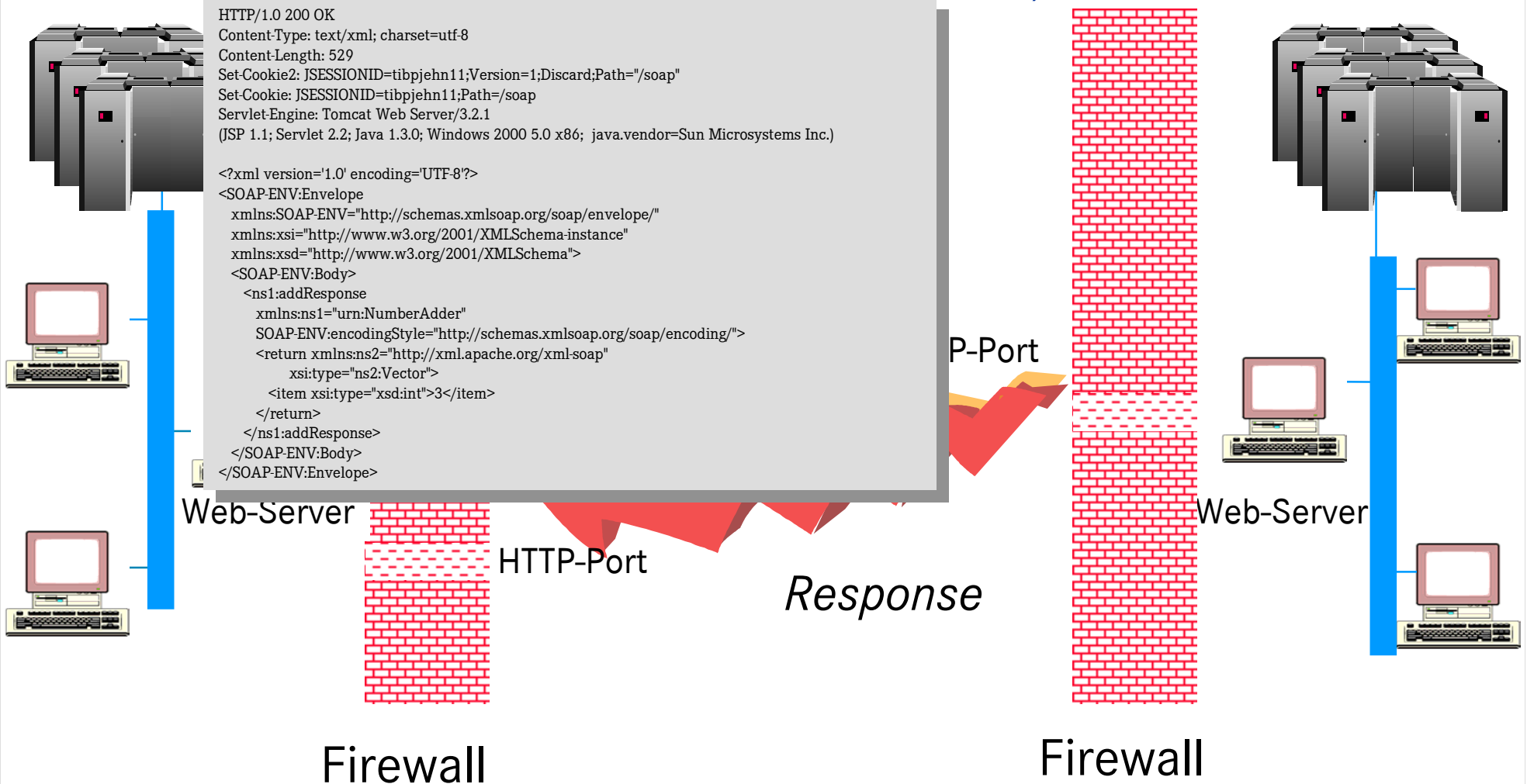
## Aktuelle Entwicklung ...

- Zusammenführung der verschiedensten W3C-Aktivitäten zur Vision des *Semantic Web*
- UML v2.0 Diagram Interchange  
Darstellung von UML-Diagrammen mit aus XMI erzeugtem SVG-Format
- Weiterentwicklung und Modularisierung von HTML
- Web Services: Grundlage XML-basierte RPCs (SOAP/XML Protocol), XML-beschriebene Dienste (Ansatz: WSDL) und Verzeichnisse (Ansatz: UDDI)
- XML als Speicherformat in Datenbanksystemen, Definition einer Anfragesprache für XML (XQuery)

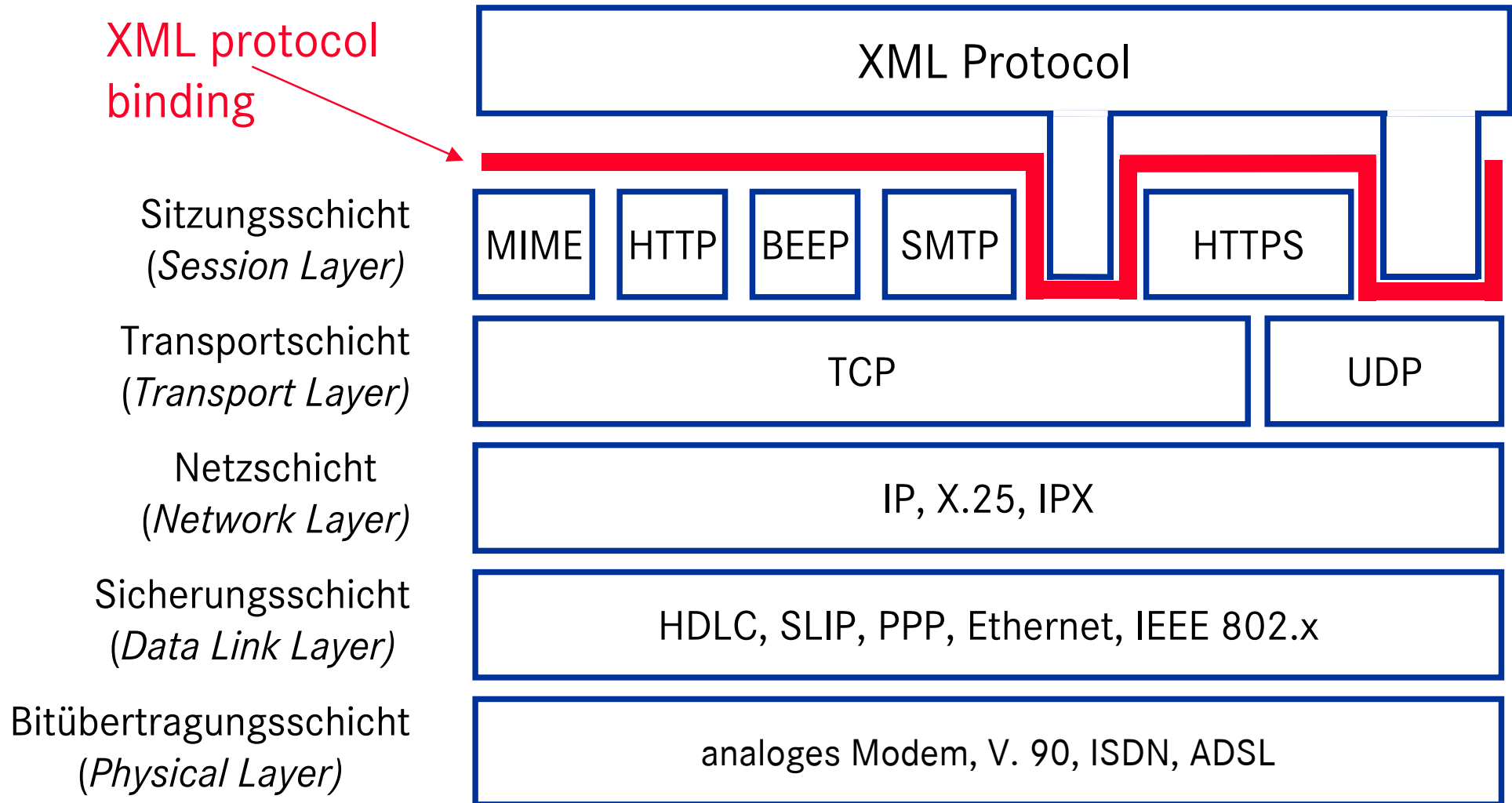
# Aktuelle Entwicklung ... XML-Protokolle (SOAP)



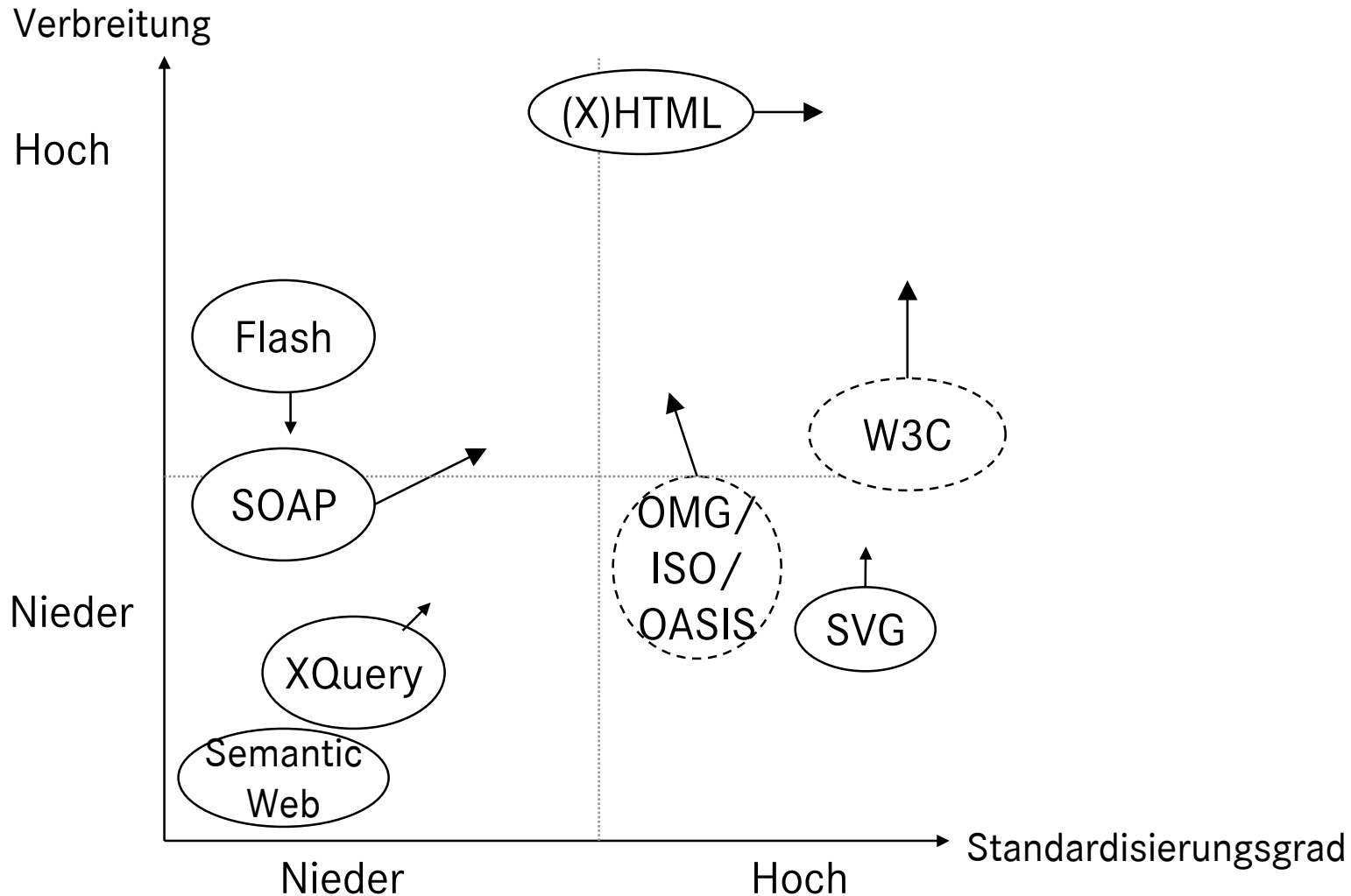
# Aktuelle Entwicklung ... XML-Protokolle (SOAP)



## Aktuelle Entwicklung ... XML-Protokolle (SOAP)

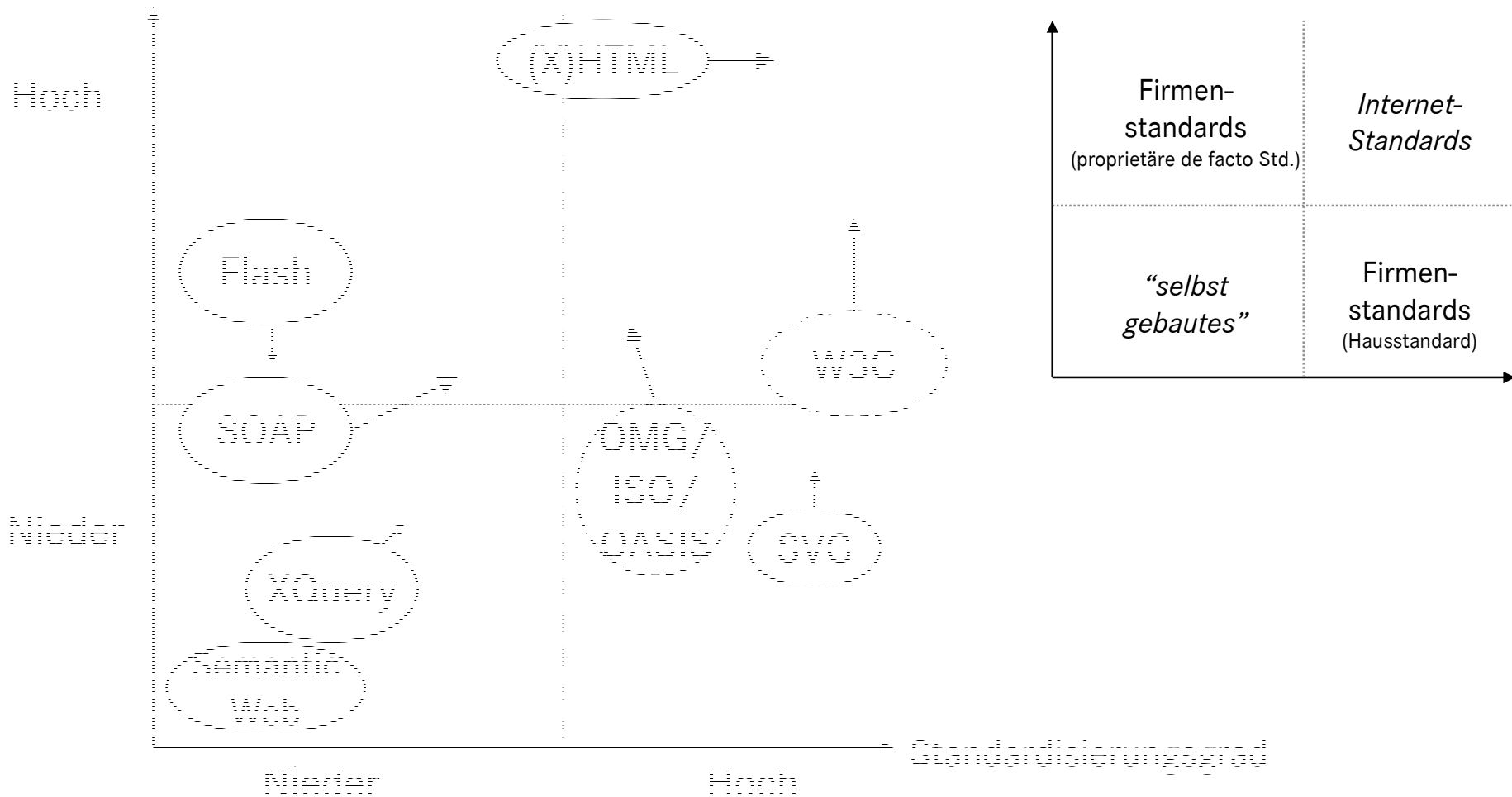


# Aktuelle Entwicklung ... und die Zukunft



# Aktuelle Entwicklung ... und die Zukunft

Verbreitung



## Referenzen

### XML Einführendes und Vertiefendes

- [www.w3.org](http://www.w3.org)
- [www.xml.com](http://www.xml.com)
- [www.xmlhack.com](http://www.xmlhack.com)
- [www.oasis-open.org/cover](http://www.oasis-open.org/cover)
- [www.mintert.com/xml/trans](http://www.mintert.com/xml/trans) (deutsche Übersetzung der XML Recommendation)
- [www.ibiblio.org/xml/index.html](http://www.ibiblio.org/xml/index.html) (Cafe con Leche)
- [www.simonstl.com](http://www.simonstl.com) (Homepage von Simon St. Laurent)
- [cm.bell-labs.com/cm/cs/who/wadler/xml](http://cm.bell-labs.com/cm/cs/who/wadler/xml) (Homepage von Phil Wadler)
- [www.cetus-links.org/oo\\_xml.html](http://www.cetus-links.org/oo_xml.html)

## Referenzen

### 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/>

### XMI @ OMG:

<http://www.omg.org/xml>

### XMI @ IBM:

<http://www.software.ibm.com/ad/features/xmi.html>

### XMI @ XML.COM:

[http://www.xml.com/xml/pub/n/New\\_XML-based\\_OMG\\_standard:\\_XMI](http://www.xml.com/xml/pub/n/New_XML-based_OMG_standard:_XMI)

### IBM's XMI-Toolkit:

<http://www.alphaworks.ibm.com/tech/xmitoolkit>

### Dieser Vortrag, seine Beispiele und vertiefende Hintergrundinformation:

<http://www.jeckle.de>