

DAIMLERCHRYSLER

Entwurf von XML-Sprachen mit XML-Schema

Mario Jeckle

DaimlerChrysler Forschungszentrum Ulm

mario.jeckle@daimlerchrysler.com

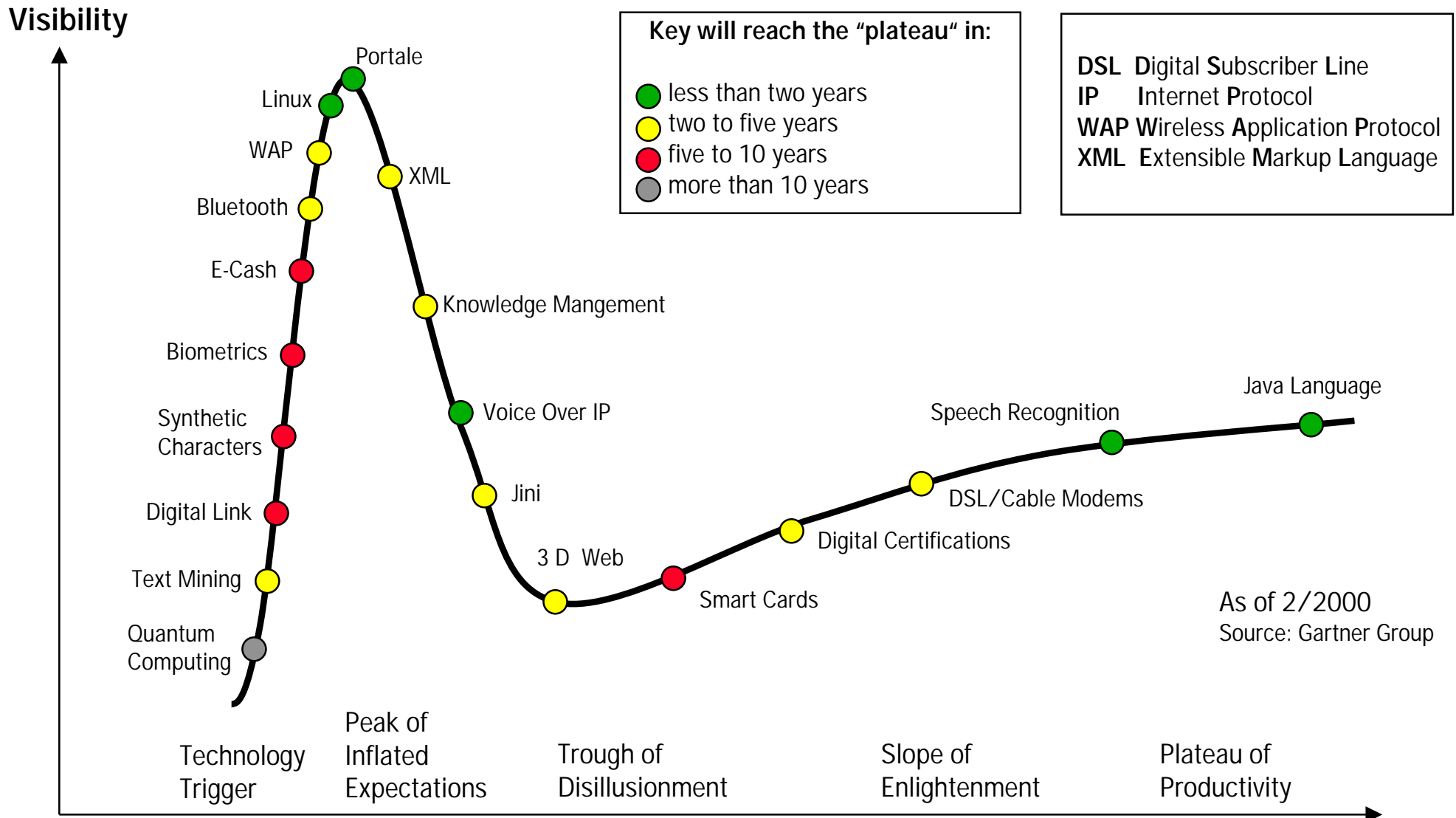
mario@jeckle.de

www.jeckle.de

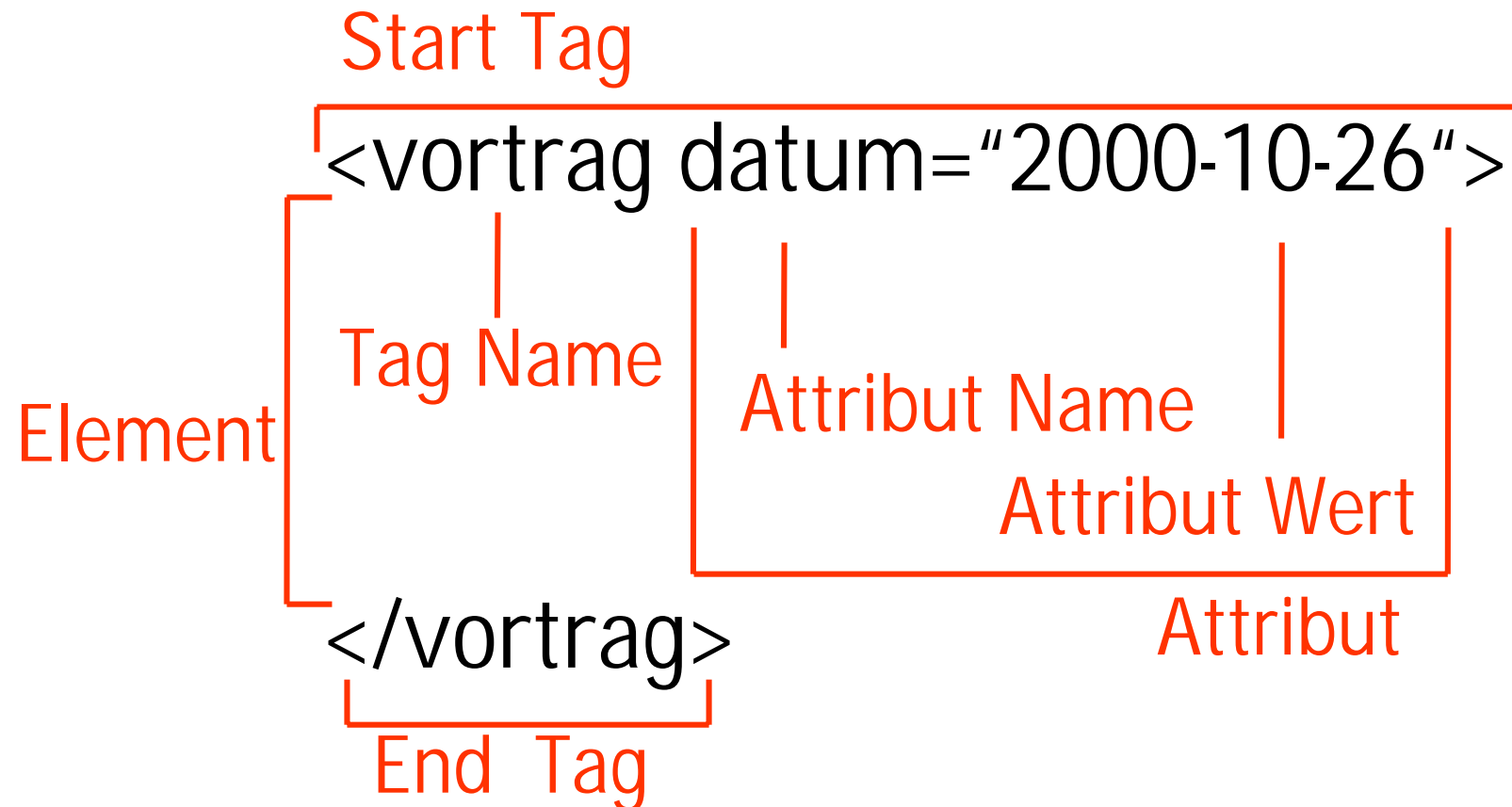
Entwurf von XML-Sprachen mit XML-Schema

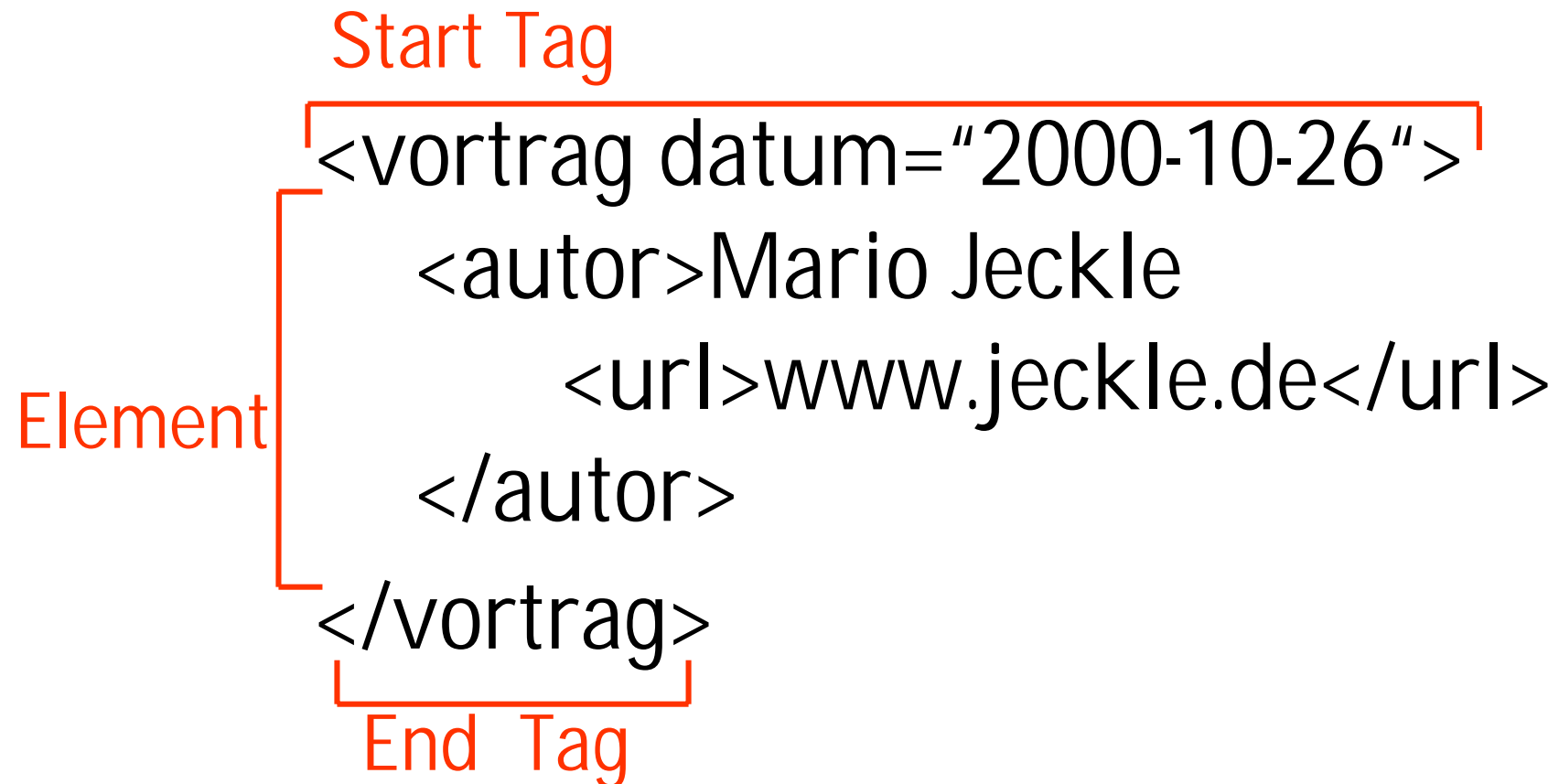
- XML-Sprachen
 - Begriffsdefinition
 - Praktische Bedeutung, existierende XML-Sprachen
- Die *Document Type Definition* (DTD)
 - Beispiel
 - Möglichkeiten und Grenzen
- Die XML-Schemasprachen
 - Alternativen und aktueller Entwicklungsstand
 - W3C's *XML Schema Description Language* (XSD)
 - Beispiel
- Praktische Einsetzbarkeit, verfügbare Werkzeuge

Gartner's Hype Life Cycle Model

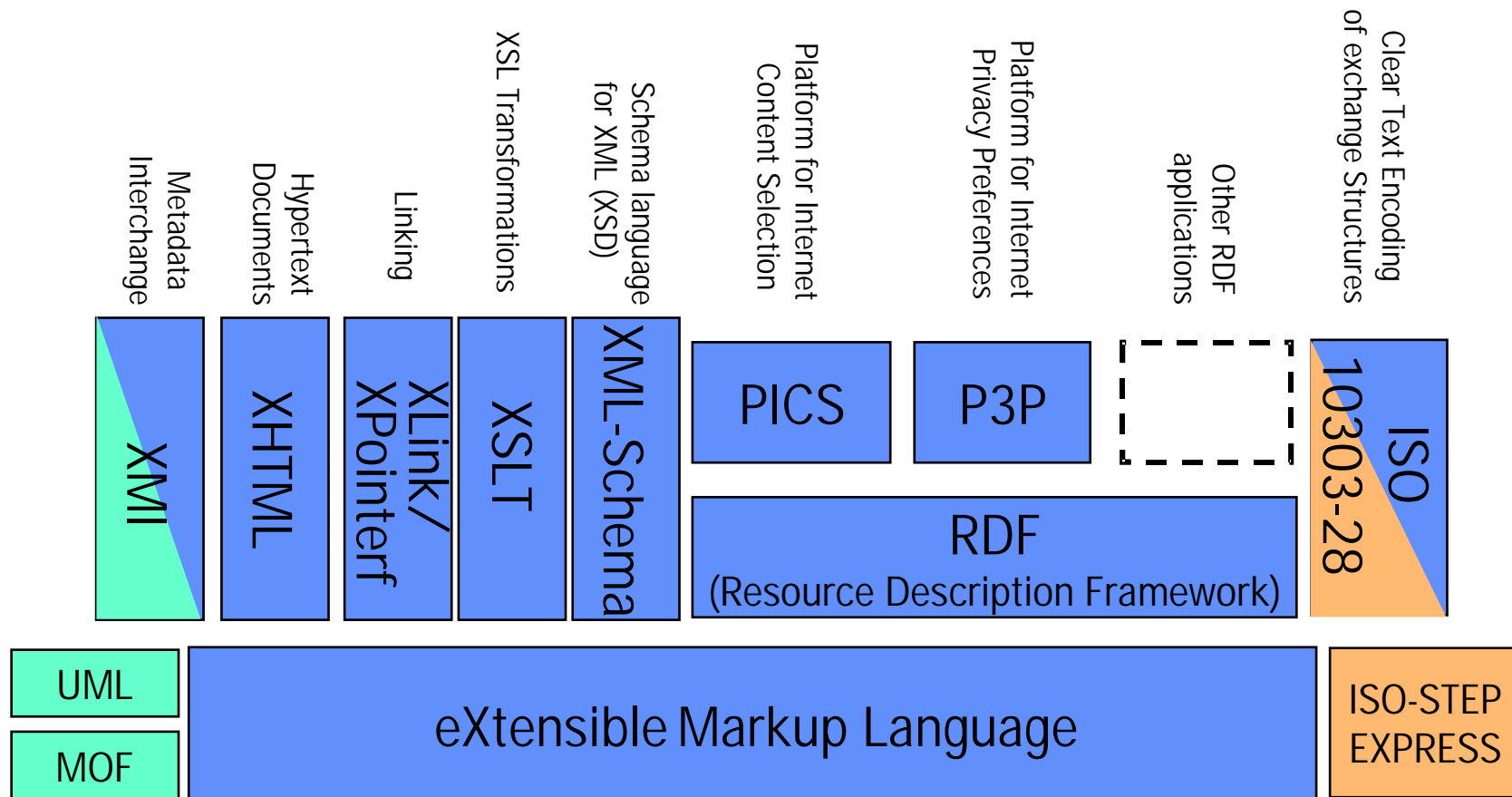


XML-Strukturprimitive





XML-Sprachen



Die XML ist eine Sprache zur Sprachdefinition (eine *Metasprache*). Mit ihr definierte Sprachen bilden die XML-Sprachfamilie.

XML-Sprachdesign mit *Document Type Definitions*

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xml:lang="en" lang="en">
  <head>
    <title>MyPage</title>
  </head>
  <body>
    content
  </body>
</html>
```

XML Dokument

Beschreibt Struktur
und erlaubte Inhalte

```
<!ENTITY % HTMLat1 PUBLIC
  "-//W3C//ENTITIES Latin 1 for XHTML//EN"
  "xhtml-lat1.ent">
%HTMLat1;

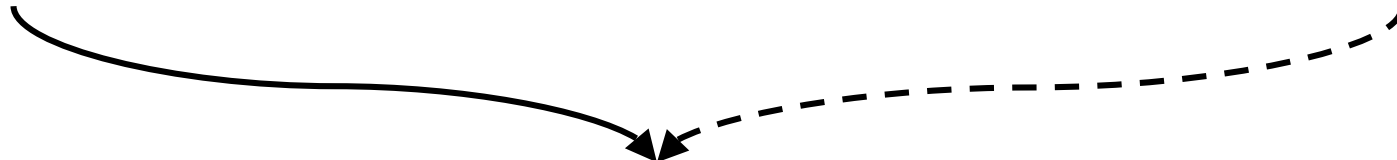
<!ENTITY % HTMLsymbol PUBLIC
  "-//W3C//ENTITIES Symbols for XHTML//EN"
  "xhtml-symbol.ent">
%HTMLsymbol;

<!ENTITY % HTMLspecial PUBLIC
  "-//W3C//ENTITIES Special for XHTML//EN"
  "xhtml-special.ent">
%HTMLspecial;

<!ENTITY % ContentType "CDATA">
  <!-- media type, as per [RFC2045] -->

<!ENTITY % ContentTypes "CDATA">
  <!-- comma-separated list of media types, as per [RFC2045] -->
```

DTD des Dokuments



Validierende generische Parser prüfen Konformität der Dokumente zu einer DTD

Entwurf von XML Sprachen – XML Schema

Der DTD-Mechanismus in XML v1.0

- Dokument ist streng hierarchisch gegliedert
 - *ELEMENTs* als innere Knoten
 - *ATTLISTs* zur Attributierung der Knoten
 - Keine Datentypen (abgesehen von CHAR-Data)
 - Rudimentärer Referenzierungsmechanismus (ID, IDREF(S))
 - Selektionstyp (enum)
 - Vorgabewerte
 - DTD ist nicht XML
- => Notwendige Konstrukte zum Ausdruck mächtigerer Semantik müssen aufwendig und proprietär realisiert werden

Entwurf von XML Sprachen – XML Schema

Motivation Schema-Mechanismen für XML

Gemeinsames Vokabular

- ermöglicht Kommunikation und Interaktion auf Basis einheitlich definierter Begriffe.
Daher ist neben der strukturellen Definition eine Semantikfestlegung unabdingbar

Formale Beschreibung

- Grundvoraussetzung maschineller Verarbeitung
- Idealerweise (vergleichsweise) einfache Verarbeitung
(schlanke, eindeutige Definitionen, möglichst kontextfreie oder reguläre Sprache)

Austauschbasis

- Explizite Strukturdefinition ist Grundvoraussetzung des Informationsaustauschs.

Entwurf von XML Sprachen – XML Schema

Anforderungen an einen XML-Schemamechanismus

Structural schemas

- Mächtigkeit analog des bestehenden DTD-Mechanismus um Dokumentstruktur (Reihenfolge, Auftrettsvielfachheit von Elementen und Attribute) zu beschreiben
Insbesondere sollen folgende Erweiterungen verwirklicht werden:
 - Namespace Integration
 - Definition von Einschränkungen für Elementinhalte
 - Integration Strukturschema und primitive Datentypen
 - Vererbung: DTD unterstützt nur *kind-of*-Beziehungen
 - Erweiterter Referenzierungsmechanismus (URI)

Entwurf von XML Sprachen – XML Schema

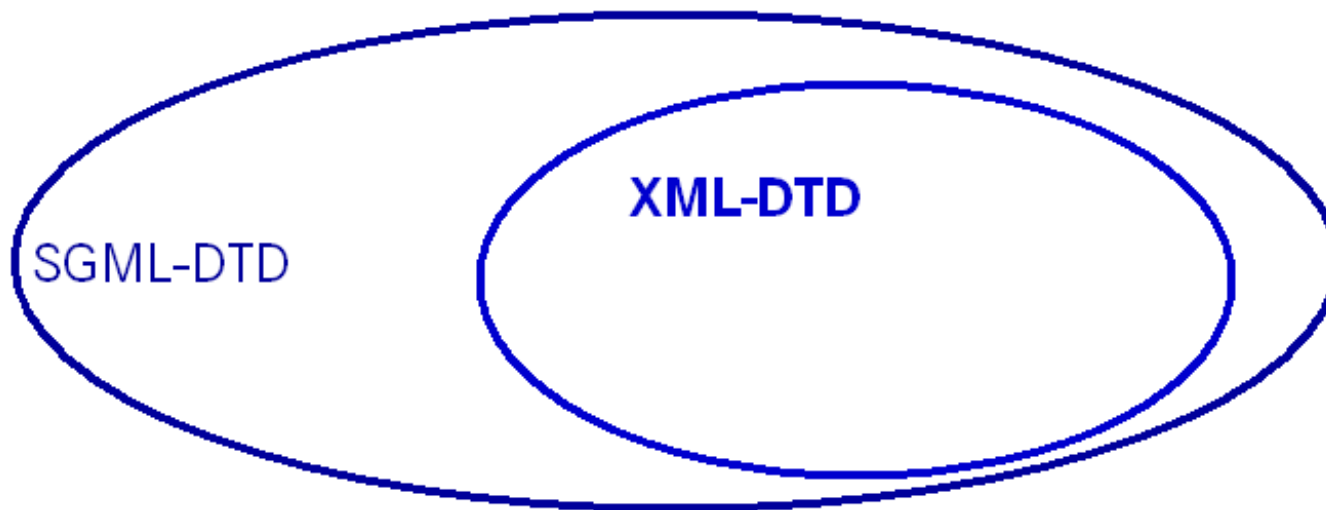
Anforderungen an einen XML-Schemamechanismus

Primitive Datentypen

- „klassische“ atomare Datentypen, ergänzt um SQL-artige, wie *integer*, *date*.
- Programmiersprachen-übliche (typischerweise Java-artige) *build-in types*
- uninterpretierte Binärstrukturen
- (durch Anwender) erweiterbares Typsystem
- lexikalische Definitionen
- Einschränkungen an Typen

Entwurf von XML Sprachen – XML Schema

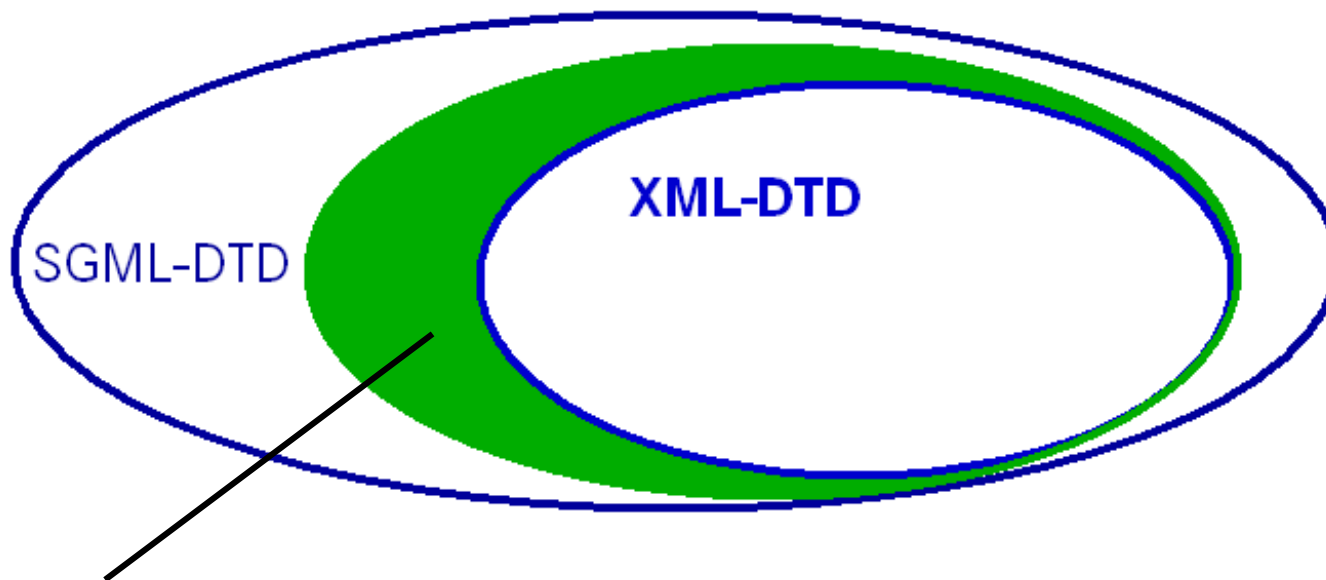
Entwicklungsoptionen ...



Ausdrucksmächtigkeit der XML-DTD bildet eine (echte) Untermenge des SGML-DTD Mechanismus.

Entwurf von XML Sprachen – XML Schema

Entwicklungsoptionen ...



Erweiterung des XML-DTD Mechanismus um weitere Elemente der SGML-DTD.

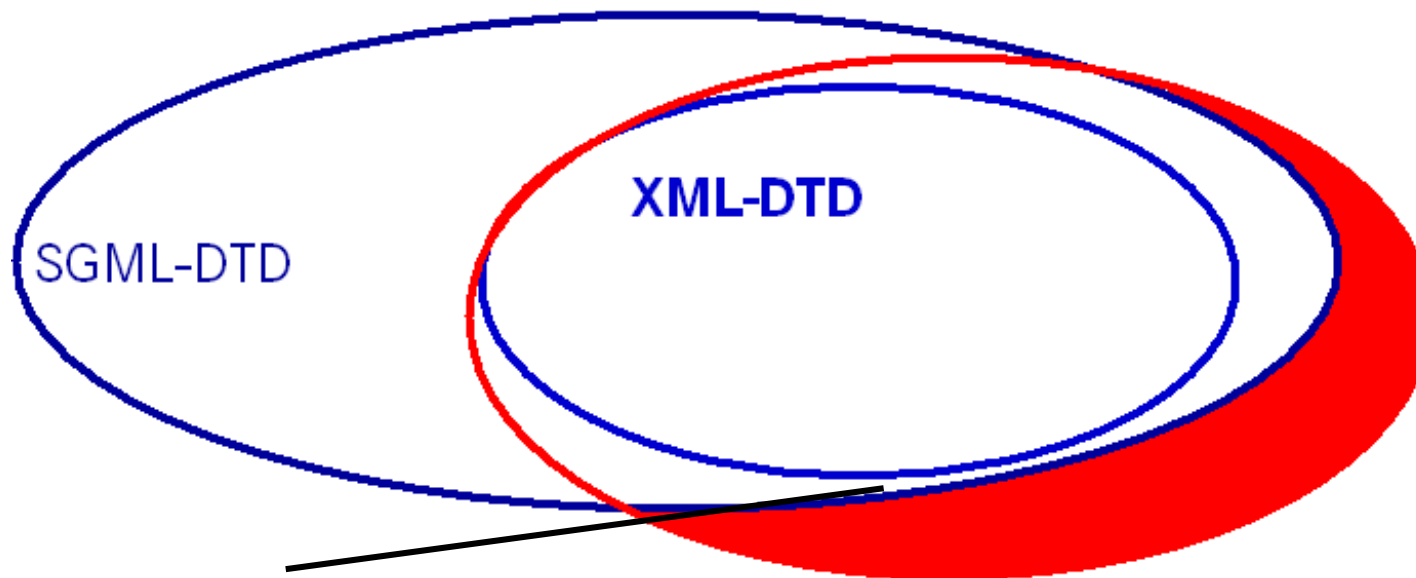
(+) Ausdrucksmächtigkeit nähert sich (wieder) der von SGML an

(-) ...die Komplexität auch

(-) Ausdrucksmächtigkeit kann die von SGML niemals übertreffen

Entwurf von XML Sprachen – XML Schema

Entwicklungsoptionen ...



Erweiterung des XML-DTD Mechanismus um Elemente, die *nicht* mit SGML-Mitteln ausdrückbar sind.

- (+) Freiheitsgrad hinsichtlich beliebiger Erweiterungen
- (-) XML-Grundforderung nach Untermengenbeziehung zu SGML entfällt
- (-) immernoch zwei verschiedene Sprachen für Inhalt und Schema

Entwurf von XML Sprachen – XML Schema

Schemadialekte

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)

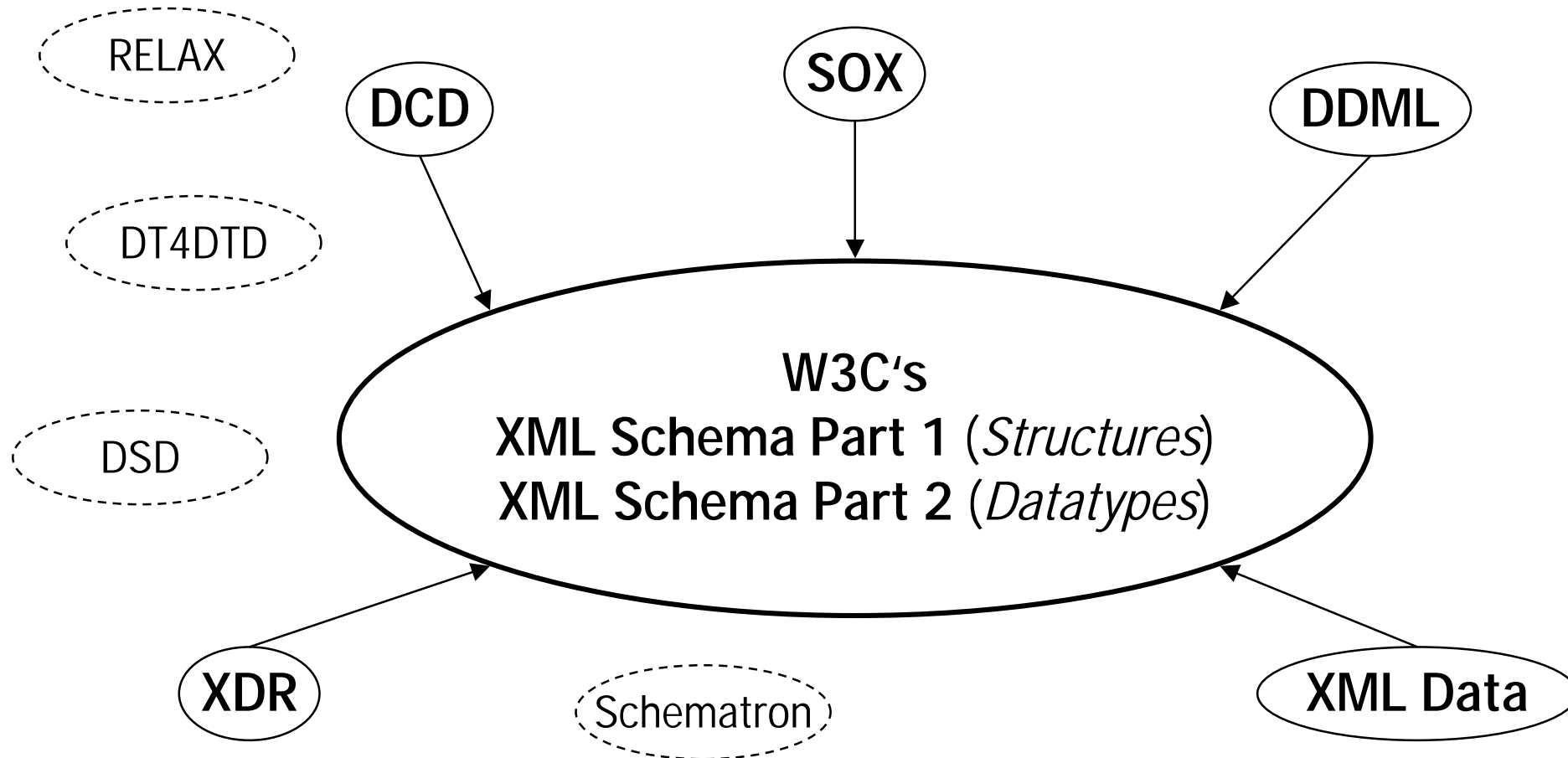
Inspiziert durch XML-API-Entwicklung

- Schema for Object oriented XML (SOX)

XML-Sprachen zur Inhaltsbeschreibung

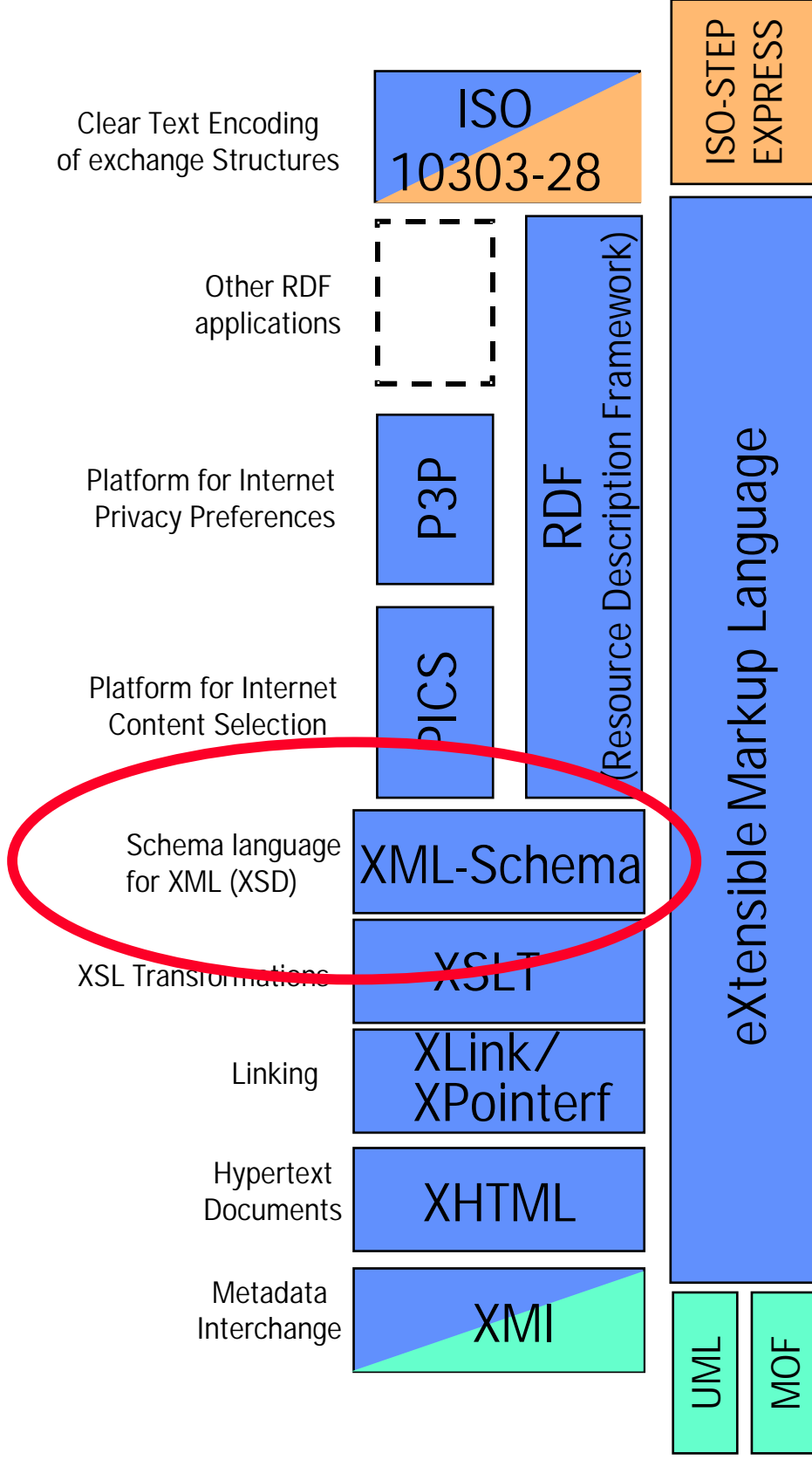
- Document Definition Markup Language/XSchema (DDML)
- Schematron (XSL-basierte Auswertung der Dokumentstruktur)
- XML-Data/XML-Data Reduced (XDR) *(erster Ansatz noch vor Verabschiedung XML 1.0)*
- Document Structure Description (DSD)
- **W3C's XML-Schema**

XML-Schema



W3C's XML-Schema konsolidiert und integriert die wesentlichen existierenden Ansätze und entwickelt sie fort.

Einordnung der XSD in die XML-Sprachfamilie



Ein Beispiel ...

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ProjektAbwicklung SYSTEM "projekt.dtd">

<ProjektAbwicklung>
  <person persID="P100" gehaltsGrp="2">
    <vorname>Hans</vorname>
    <vorname>Georg</vorname>
    <name>Meier</name>
    <projektLeiter prjRef='F300'/>
    <projektMitarb prjRef='F310'/>
  </person>
  <projekt prjID="F300" start='1.1.2000' budget='1000'/>
  <projekt prjID="F310" budget='10000'/>
  <projekt prjID="F320" budget='25000'/>
</ProjektAbwicklung>
```

Die DTD ...

```

<!ELEMENT      ProjektAbwicklung (person+, projekt*)>
<!ELEMENT      person (vorname+, name, projektLeiter?, projektMitarb*)>
<!ATTLIST      person
persID          ID          #REQUIRED
gehaltsGrp     (1|1a|2) "1">
vorname        (#PCDATA)>
<!ELEMENT      name        (#PCDATA)>
<!ELEMENT      projektLeiter EMPTY>
<!ATTLIST      projektLeiter
prjRef         IDREF #REQUIRED>
<!ELEMENT      projektMitarb EMPTY>
<!ATTLIST      projektMitarb
prjRef         IDREF #REQUIRED>
<!ELEMENT      projekt     EMPTY>
<!ATTLIST      projekt
prjID          ID          #REQUIRED
start         CDATA #IMPLIED
budget        CDATA #REQUIRED>

```

Die DTD ... (Elemente)

```
<person persID="P100" gehaltsGrp="2">  
  <vorname>Hans</vorname>  
  <vorname>Georg</vorname>  
  <name>Meier</name>  
  <projektLeiter prjRef='F300'/>  
  <projektMitarb prjRef='F310'/>  
</person>
```

```
<!ELEMENT   person  
            (vorname+, name,  
            projektLeiter?, projektMitarb*)>  
  
<!ATTLIST  person  
            persID      ID      #REQUIRED  
            gehaltsGrp  (1|1a|2) "1">
```

Elemente können weitere Elemente beinhalten.
Die Auftrittshäufigkeit (*Kardinalität*) kann mittels symbolische Notation gesteuert werden ("+" ein oder mehrmals, "?" optional, "*" tritt garnicht, oder beliebig oft auf)

Die DTD ... (Attribute)

```
<person persID="P100" gehaltsGrp="2">  
  <vorname>Hans</vorname>  
  <vorname>Georg</vorname>  
  <name>Meier</name>  
  <projektLeiter prjRef='F300'/>  
  <projektMitarb prjRef='F310'/>  
</person>
```

```
<!ELEMENT   person  
            (vorname+, name,  
            projektLeiter?, projektMitarb*)>  
  
<!ATTLIST  person  
            persID      ID      #REQUIRED  
            gehaltsGrp (1 | 1a | 2) "1">
```

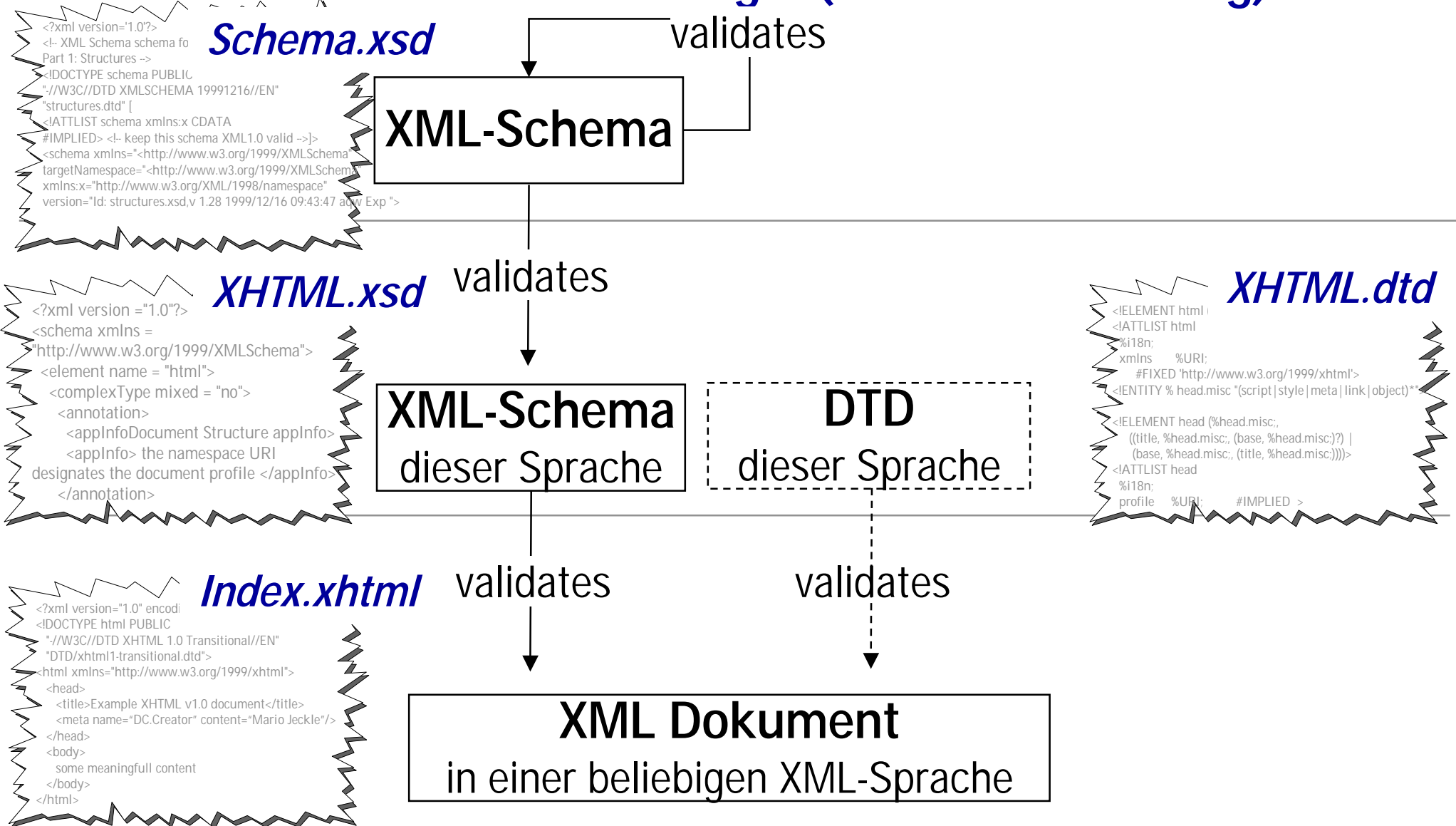
Elemente können Attribute beinhalten.
Neben identifizierenden Schlüsselattributen (*ID*) existieren Zeichenketten- und Auswahldatentypen.

Möglichkeiten und Grenzen des DTD-Mechanismus

- streng hierarchisch
- *ELEMENTs* als innere Knoten
- *ATTLISTs* zur Attributierung der Knoten
- Keine „echten“ Datentypen (abgesehen von CHAR-Data)
- Nicht erweiterbarer Typvorrat
- Rudimentärer Referenzierungsmechanismus (ID, IDREF)
- Selektionstyp
- Vorgabewerte
- DTD ist nicht XML => zusätzliche Sprache zu lernen

=> Notwendige Konstrukte zum Ausdruck mächtigerer Semantik müssen aufwendig und proprietär realisiert werden

W3C's XML Schema -- Technologie (Metamodellierung)



Von der DTD zum Schema ...

```
<!ELEMENT   person
            (vorname+, name,
            projektLeiter?, projektMitarb*)>
```

```
<element name = "person">
  <complexType mixed = "no">
    <sequence>
      <element ref = "vorname" minOccurs = "1" maxOccurs = "unbound"/>
      <element ref = "name"/>
      <element ref = "projektLeiter" minOccurs = "0" maxOccurs = "1"/>
      <element ref = "projektMitarb" minOccurs = "0" maxOccurs = "unbound"/>
    </sequence>
```

...

Von der DTD zum Schema ...

```
<!ATTLIST    person
             persID      ID      #REQUIRED
             gehaltsGrp  (1|1a|2) "1">
```

```
<attribute name = "persID" use = "required" type = "ID"/>
```

```
<attribute name = "gehaltsGrp" use = "default" value = "1">
```

```
  <simpleType base = "xsd:string">
```

```
    <xsd:enumeration value = "1"/>
```

```
    <xsd:enumeration value = "1a"/>
```

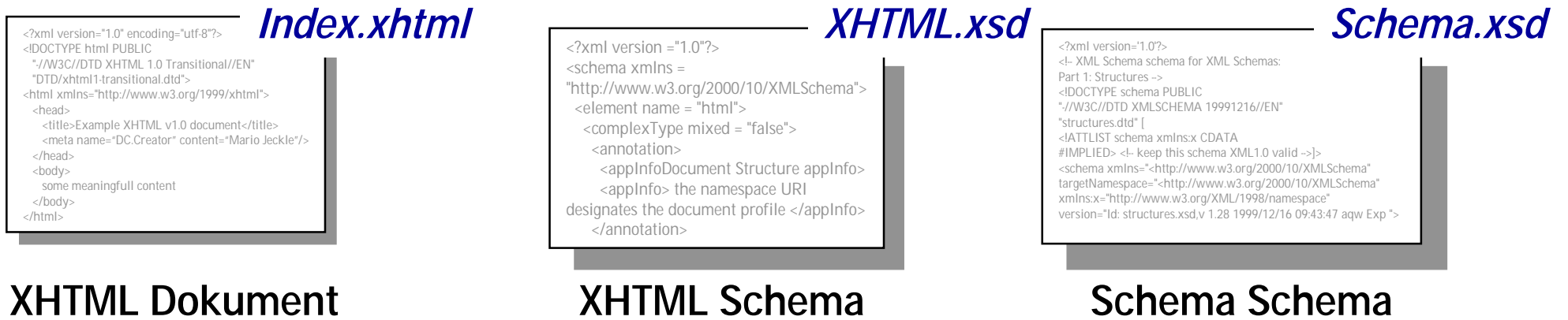
```
    <xsd:enumeration value = "2"/>
```

```
  </simpleType>
```

```
</attribute>
```

Entwurf von XML Sprachen – XML Schema

Schema validness



XML Dokument konform
zu Schemabeschreibung?

XML Dokument (Schema) konform
zu (Schema) Schemabeschreibung?

Entwurf von XML Sprachen – XML Schema

Empfehlungen zum praktischen Einsatz

- Schema-Recherche vor Eigenentwicklung!
- Information die in verschiedenen Rollen auftritt sollte (generell) als *complexType* definiert werden, um die Wiederverwendbarkeit zu erhöhen
- Wenn möglich...
 - spezialisierte Datentypen nutzen
 - eigene Datentypen ableiten
 - reguläre Ausdrücke
 - Aufzählungstypen
- Bidirektional navigierbare Beziehungen mit Kardinalitätsanteil größer Eins in beiden Richtungen (sog. *n:m-Beziehungen*) in separate Elemente aufbrechen

Zusammenfassung

- Status: *candidate recommendation* (CR 2000-10-24)
Vor der Verabschiedung als *recommendation*
- Der zukünftige Standard zur XML-Sprachdefinition
- Wert *"Schemas are coming: Start using them!"*
 - *Tim Berners-Lee, 1999-11-05*
- Umstellung von DTD-Repositories auf Schema zu beobachten
- Sprach*generierung* (XMI!) statt (manueller) Sprach*erstellung*
- XML-Schema ist keine Datenmodellierungssprache
- XML-Schema ist eine XML-Sprache
- Jedes Schema ist ein (schema) valid XML-Dokument

Ausblick: Entwurf von XML Sprachen mit XMI

Gegenwärtige Situation in der Praxis

- Verwendung von Schemaeditoren und weiteren Werkzeugen
(-> nicht integriert, keine Standardnotation, kein Entwurfsprozeß)
- Zu hohes Abstraktionsniveau der Schemasprachen
(-> mit unter Reverse-Engineering notwendig)
- Fehlende Dokumentationsintegration
(-> intransparente Semantik => Integrations- und Kopplungsproblematik)
- Statische Dokumentschemata
(-> geringer Wiederverwendungsgrad durch fehlende Modularisierung)
- **Schema allein (ohne zusätzliche deskriptive Information) nicht verwendbar**



Ausblick: Entwurf von XML Sprachen mit XMI

XMI-Ansatz: Generierung von XML-Strukturen aus UML-Modellen

- Existierendes Klassendiagramm wird unverändert zur Erzeugung von XML-DTDs (zukünftig auch Schemata) herangezogen.
 - nahtlose Prozeßintegration
 - Flexible, zeitnahe Adaptierbarkeit
 - Nachvollziehbare XML-Strukturen gleichbleibender Qualität
 - Fördert entstehen von Sprachfamilien
- XML-DTD/Schema entspricht strukturell dem Klassendiagramm, nicht jedoch semantisch.

Ausblick: Entwurf von XML Sprachen mit XMI

XML-Strukturen ...

- Widerspiegelung komplexer (Business) Strukturen
- Modellierungskultur
- Entstehung von *Dokumentfamilien* im Unternehmen
- Standardkompatibilität (semantische Interoperabilität)
- Hyperlinking über Dokumenttypgrenzen hinweg
- Applikationsseitige System- und Prozeßintegration
(*impedance mismatch*)
- Langzeitspeicherung
- Adaptierbarkeit, Erweiterbarkeit
- Kein existierender (weit anerkannter) Entwurfs-Formalismus

References

W3C's XML-Schema (2000-10-25 *candidate recommendation*):

- www.w3.org/TR/NOTE-xml-schema-req (*XML schema requirements*)
- www.w3.org/TR/xml-schema-0
- www.w3.org/TR/xml-schema-1
- www.w3.org/TR/xml-schema-2

Alternativvorschläge:

- www.w3.org/TR/dt4dtd
- www.w3.org/TR/NOTE-dcd
- www.w3.org/TR/NOTE-ddml
- www.brics.dk/DSD/
- www.w3.org/TR/NOTE-SOX/
- www.ascc.net/xml/resource/schematron/schematron.html
- www.w3.org/TR/1998/NOTE-XML-data-0105

References

Sekundärliteratur:

- www.w3.org/TR/schema-arch
- www.lindamann.com/xml/XML%20Schemas%20NG%20Guide%20HTML.htm
- xml.com/pub/2000/02/23/xmldeviant/index.html?wwwrrr_20000223.txt
- www.w3.org/TR/NOTE-xml-schema-req
- www.iso.ch/cate/d19346.html (*ISO 11404*)

Werkzeuge:

- www.alphaworks.ibm.com/formula/xml
- xml.apache.org (*Xerces*)
- www.extensibility.com (*XML Authority*)
- www.xmlspy.com

Tangierte und weiterführende XML-Literatur:

- www.w3.org/TR/REC-xml (*XML v1.0 Recommendation*)
- www.w3.org/TR/REC-xml-names
- www.w3.org/TR/xml-infoset

Dieser Vortrag und weiterführende Information:

- www.jeckle.de