



Überblick zu XML

Stephan Karrer

Markierungssprachen (Markup Language)

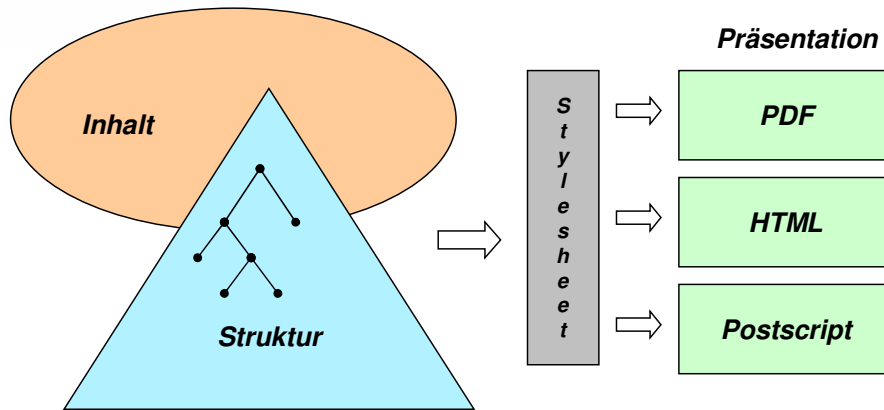
- Spezielle Markierungssprachen, z.B. RTF oder HTML
 - Für bestimmte Anwendungen/Geräte entworfen
 - Menge an Tags vorgegeben und nicht beliebig erweiterbar
 - Der Schwerpunkt liegt meist auf der Präsentation
 - Wenig portabel, da meist nur von bestimmten Anwendungen interpretierbar

- Allgemeine Markierungssprachen, z.B. SGML und XML
 - Markierung beschreibt Struktur bzw. Bedeutung der einzelnen Elemente
 - Legt nicht die Interpretation (Verwendung) und die Präsentation fest
 - Metasprachen, d.h. mit ihrer Hilfe werden geeignete Markierungen definiert

Basis-Technologie XML

- XML ist heute der Oberbegriff für eine ganze Familie von Technologien und Standards zum Strukturieren von Informationen, Formatieren von Dokumenten bis hin zur Datenfilterung
- XML (eXtensible Markup Language) ist im engeren Sinne ein Regelwerk, um Auszeichnungssprachen (Markup Languages) zu erstellen
- Grundgedanke: Trennung von Inhalt, Struktur und Präsentation
- Entwurfsziele:
 - SGML-kompatibel
 - XML-Dokumente sollen leicht erstellbar und lesbar sein
 - möglichst universell einsetzbar

Inhalt – Struktur - Darstellung



Beispiel für ein XML-Dokument: „Notiz“

Mögliche Beschreibung einer Notiz

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Quelle: XML-Tutorial auf www.w3schools.com

XML-Dokument mit zugehöriger Grammatik in Form einer DTD (Document Type Definition)

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to      (#PCDATA)>
  <!ELEMENT from      (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body      (#PCDATA)>
]>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend</body>
</note>
```

Quelle: XML-Tutorial auf www.w3schools.com

DTD: Basis

Aus Sicht einer DTD bestehen XML-Dokumente aus folgenden Basiseinheiten:

- Elemente: z.B. `<body>some text</body>`
- Attributes: z.B. `<person number="5677" />`
- Entities: z.B. `<author>&writer;©right;</author>`
(Abkürzungen für Zeichenketten bzw. Sonderzeichen)
- PCDATA: zu parsender Text
- CDATA: nicht zu parsender Text

Beispiel:

```
<!ENTITY % block "absatz | listing | abbildung | kasten">
<!ELEMENT buch (kapitel+)>
<!ELEMENT kapitel (ueberschrift, (%block;)*, abschnitt+)>
<!ELEMENT abschnitt (ueberschrift, (%block;)*, unterabschnitt*)>
<!ELEMENT unterabschnitt (ueberschrift, (%block;)+)>
```

Wohlgeformte und gültige (valid) XML-Dokumente

- Ein wohlgeformtes XML-Dokument hat korrekte XML-Syntax
 - XML-Dokumente müssen ein Wurzel-Element (root) haben
 - XML-Elemente müssen ein abschließendes Tag haben
 - Bei XML-Tags wird Groß/Kleinschreibung unterschieden
 - XML-Elemente müssen sauber verschachtelt sein
 - XML-Attributwerte stehen stets in Anführungszeichen
- Ein wohlgeformtes XML-Dokument ist gültig, wenn es einer Grammatik entspricht
- Eine Software, die ein XML-Dokument liest und auf Gültigkeit prüft, nennt man einen (validierenden) Parser
- Eine Vielzahl von Werkzeugen, insbesondere Browser, enthalten einen Parser

XML – Namensräume (namespaces)

```
<?xml version="1.0"?>
<library-entry xmlns:authr="authors.dtd" xmlns:bk="books.dtd">
<bk:book>
  <bk:title>XML Sample</bk:title>
  <bk:pages>210</bk:pages>
  <bk:isbn>1-868640-34-2</bk:isbn>
  <authr:author>
    <authr:firstname>John</authr:firstname>
    <authr:lastname>Smith</authr:lastname>
    <authr:title>Mr</authr:title>
  </authr:author>
</bk:book>
</library-entry>
```

Namespaces

Verhinderung von Namenskonflikten

XML: von der DTD zum Schema

Nachteile des DTD-Mechanismus:

- Unzureichende Datentypunterstützung
- Unzureichende Strukturierungsunterstützung
- Keine Unterstützung von Namensräumen
- ...

Dies führte zur Entwicklung von

XML Schema Description Language (XSD)

- Fremde XML-Schemata können eingebunden werden
- XML-Schemata sind reicher (ausdrucksstärker) als DTDs
- XML-Schemata unterstützen Datentypen
- XML-Schemata unterstützen Namensräume
- XML-Schemata verwenden selbst XML

XML Schema: Beispiel

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace=http://www.w3schools.com xmlns=http://www.w3schools.com
  elementFormDefault="qualified">

  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<?xml version="1.0"?>
<note xmlns="http://www.w3schools.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3schools.com note.xsd">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

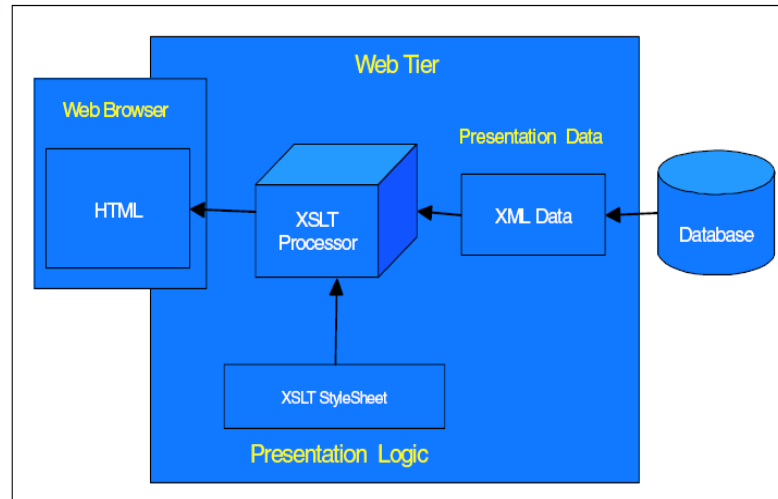
Quelle: XML-Tutorial auf www.w3schools.com

XML Schemata – erweiterte Möglichkeiten

```
...  
<xs:simpleType name="inttype">  
  <xs:restriction base="xs:positiveInteger"/>  
</xs:simpleType>  
  
<xs:simpleType name="orderidtype">  
  <xs:restriction base="xs:string">  
    <xs:pattern value="[0-9]{6}"/>  
  </xs:restriction>  
</xs:simpleType>  
  
<xs:complexType name="shiptotype">  
  <xs:sequence>  
    <xs:element name="name" type="stringtype"/>  
    <xs:element name="address" type="stringtype"/>  
    <xs:element name="city" type="stringtype"/>  
    <xs:element name="country" type="stringtype"/>  
  </xs:sequence>  
</xs:complexType>  
  
...
```

- **Einfache Datentypen**
(string, integer, boolean, ...)
- **Komplexe Datentypen**
- **Einschränkung/Erweiterung
vorhandener Datentypen**

Nutzung von XSTL (Extensible Stylesheet Language Transformations)



Quelle: IBM

XSLT - Beispiel

```
<?xml version="1.0"
      encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/
XSL/Transform">

<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collection</h2>
    <table border="1">
      <tr bgcolor="#9acd32">
        <th align="left">Title</th>
        <th align="left">Artist</th>
      </tr>
      <xsl:for-each
        select="catalog/cd">
        <tr>
          <td><xsl:value-of
            select="title"/></td>
          <td><xsl:value-of
            select="artist"/></td>
        </tr>
      </xsl:for-each>
    </table>
  </body>
</html>
</xsl:template>

</xsl:stylesheet>
```

```
<?xml version="1.0"
      encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl"
      href="cdcatalog.xsl"?>

<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  .
  .
  .
</catalog>
```

Quelle: XML-Tutorial auf www.w3schools.com

XSLT Beispiel: Darstellung des Ergebnisses

Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

http://www.w3schools.com/xsl/cdcatalog_with_xsl.x

Links anpassen Kostenlose Hotmail Windows Windows Media

Extensible... XML in 10 ... Extensible... IBM Redb... Cover Pag..

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore
Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr Hook
Maggie May	Rod Stewart
Romanza	Andrea Bocelli
When a man loves a woman	Percy Sledge
Black angel	Savage Rose
1999 Grammy Nominees	Many

Verweise in XML

XLink (XML Linking Language)

- Erzeugung von Hyperlinks in XML-Dokumenten
- Jedes Element eines XML-Dokuments kann sich als Link verhalten
- Es gibt einfache (simple) Links und erweiterte Links
- Links können außerhalb der verknüpften Dokumente definiert werden

```
<book title="XQuery Kick Start">
  <description
    xlink:type="simple"
    xlink:href="http://book.com/images/XQuery.gif"
    xlink:show="new">
    XQuery Kick Start delivers a concise ...
  </description>
</book>
```

XPointer (XML Pointer Language)

- Definition von Links, die auf spezifische Teile des XML-Dokuments verweisen
- XPointer verwendet XPath-Ausdrücke zur Navigation im XML-Dokument

Navigation in XML-Dokumenten: XPath

- XPath ist die Syntax um Teile des XML-Dokuments zuzugreifen
- XPath verwendet Pfadausdrücke um im XML-Dokument zu navigieren
- XPath enthält Bibliotheken von Standardfunktionen
- XPath wird in XSLT und XQuery benutzt

```
/child::book/child::copies
```

```
/child::book[position()-1]
```

```
/book/author[@type='old']
```

Abfrage von XML-Daten: XQuery

- XQuery ist die Abfragesprache für XML Daten
- XQuery für XML ist vergleichbar mit SQL für relationale Datenbanken
- XQuery basiert auf XPath-Ausdrücken
- XQuery wird durch die relevanten DBMS unterstützt (IBM, Oracle, Microsoft, etc.)

```
FOR $b IN document("bib.xml")//book
WHERE $b/publisher = "Morgan Kaufmann"
AND $b/year = "1998"
RETURN $b/title

FOR $sp IN document("sp.xml")//sp_tuple,
  $p IN document("p.xml")//p_tuple
  [pno = $sp/pno],
  $s IN document("s.xml")//s_tuple
  [sno = $sp/sno]
RETURN
  <sp_pair>
    $s/sname ,
    $p/descrip
  </sp_pair> SORTBY (sname, descrip)
```

Anwendung: Konfigurationsdaten

Deployment-Deskriptor
einer EJB

```
<entity>
  <ejb-name>Account</ejb-name>
  <local-home>itso.bank5.cmp.AccountLocalHome</local-home>
  <local>itso.bank5.cmp.AccountLocal</local>
  <ejb-class>itso.bank5.cmp.AccountBean</ejb-class>
  <persistence-type>Container</persistence-type>
  <prim-key-class>java.lang.String</prim-key-class>
  <reentrant>False</reentrant>
  <cmp-version>2.x</cmp-version>
  <abstract-schema-name>Account</abstract-schema-name>
  <cmp-field id="CMPAttribute_1035310323213">
    <field-name>accountID</field-name>
  </cmp-field>
  <cmp-field id="CMPAttribute_1035309444670">
    <field-name>balance</field-name>
  </cmp-field>
  .....
  <primkey-field>accountID</primkey-field>
  <ejb-local-ref id="EJBLocalRef_1035426516547">
    <ejb-ref-name>ejb/TransRecord</ejb-ref-name>
    <ejb-ref-type>Entity</ejb-ref-type>
    <local-home>itso.bank5.cmp.TransRecordLocalHome</local-home>
    <local>itso.bank5.cmp.TransRecordLocal</local>
  </ejb-local-ref>
  .....
  <query>
    <description>Retrieve gold accounts ....</description>
    <query-method>
      <method-name>findGoldAccounts</method-name>
      <method-params> <method-param>java.math.BigDecimal</method-param>
    </method-params>
    </query-method>
    <ejb-ql>select object(o) from Account o where o.balance > ?1
    </ejb-ql>
  </query>
</entity>
```

Anwendung: SOAP

SOAP Document-Stil

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap envelope">
  <env:Header>
    <m:reservation xmlns:m="http://travelcompany.example.org/reservation"
      env:role="http://www.w3.org/2003/05/soap envelope/role/next"
      env:mustUnderstand="true">
      <m:reference>uuid:093a2da1 045 79f b5d 0ff98fe8j7d</m:reference>
      <m:dateAndTime>2001 11 29T13:20:00.000 0500</m:dateAndTime>
    </m:reservation>
  </env:Header>
  <env:Body>
    <p:departure xmlns:p="http://travelcompany.example.org/reservation/travel">
      <p:departing>New York</p:departing>
      <p:arriving>Los Angeles</p:arriving>
      <p:departureDate>2001 12 14</p:departureDate>
    </p:departure>
    <q:lodging
      xmlns:q="http://travelcompany.example.org/reservation/hotels">
      <q:preference>none</q:preference>
    </q:lodging>
  </env:Body>
</env:Envelope>
```

Quelle: W3C

Anwendung: BPEL Business Process Execution Language

```
<?xml version="1.0" encoding="UTF-8"?>
<bpws:process
xmlns:bpws="http://schemas.xmlsoap.org/ws/2004/03/businessprocess/"
xmlns:ns="http://control-flow/sequenceArtifacts"
xmlns:ns0="http://control-flow/sequenceInterface"
xmlns:wpc="http://www.ibm.com/xmlns/prod/websphere/businessprocess/
6.0.0/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
expressionLanguage="http://www.ibm.com/xmlns/prod/websphere/businessproces
s/"
expression-lang/java/6.0.0/" name="sequence"
suppressJoinFailure="yes" targetNamespace="http://control-flow"
wpc:displayName="sequence" wpc:executionMode="microflow" wpc:id="1">
<bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="sequence.wsdl" namespace="http://controlflow/
sequenceInterface"/>
<bpws:import importType="http://schemas.xmlsoap.org/wsdl/"
location="sequenceArtifacts.wsdl" namespace="http://controlflow/
sequenceArtifacts"/>
...
```

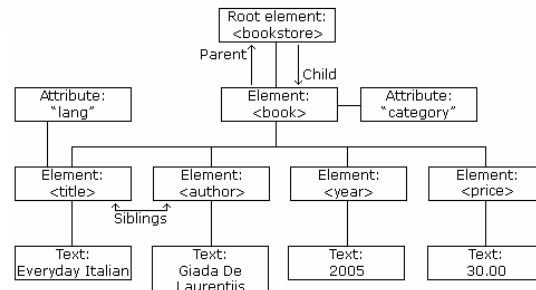
Weitere Anwendungen

- Textdarstellung
 - XSL-FO (Textformatierung)
 - DocBook
 - XHTML (XML-konformes HTML)
 - TEI (Text Encoding Initiative)
 - NITF (News Industry Text Format)
 - ...
- Grafik
 - SVG (Vektorgrafik)
 - X3D (3D-Modellierung)
 - ...
- Geodaten
 - GML (Geography Markup Language)
 - GPX Exchange Format
 - ...
- Multimedia
 - SMIL
 - MPEG-7
 - ...
- Sicherheit
 - SAML (Sicherheitsinformationen)
 - XML Signature (digitale Signaturen))
 - ...
- Semantisches Web
 - RDF (Resource Description Framework)
 - ...
- Industriezweigen/Konsortien

Werkzeuge

- XML-Editoren
- Parser
- XSLT-Prozessoren
- XML-Datenbanken
- ...

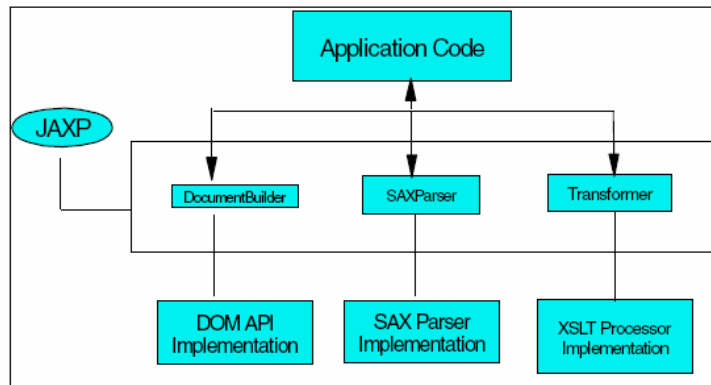
XML-Programmierung: Document Object Model



"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."

Quelle: W3C

Nutzung des Java API for XML Processing (JAXP)



Quelle: IBM

Bespiel für die Nutzung von JAXP

Example 3-6 Sample XML to HTML transformation code

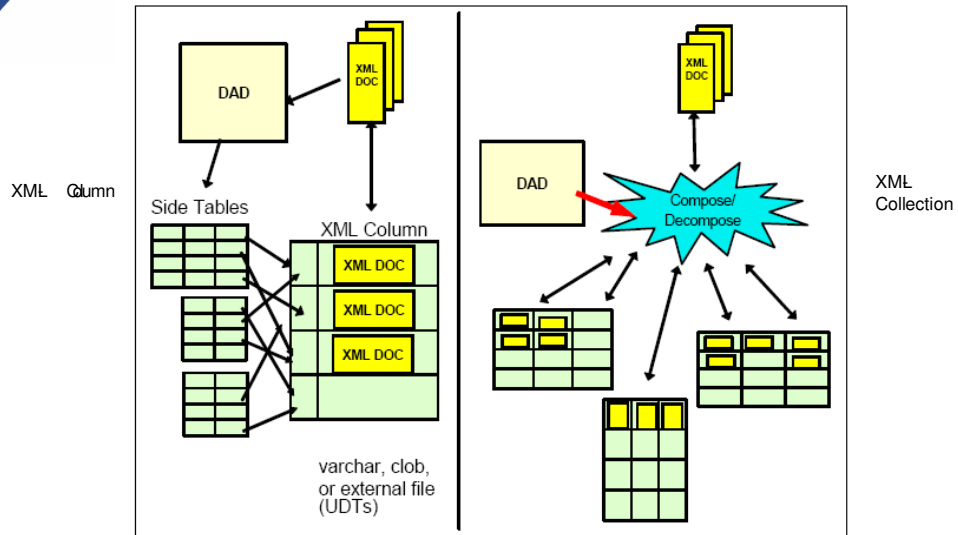
```
File xml = new File("fileName.xml");
File xslt = new File("fileName.xsl");
File html = new File("fileName.html");

javax.xml.transform.Source xmlSource =
    new javax.xml.transform.stream.StreamSource(xml);
javax.xml.transform.Source xsltSource =
    new javax.xml.transform.stream.StreamSource(xslt);
javax.xml.transform.Result result =
    new javax.xml.transform.stream.StreamResult(html);

//create an instance of TransformerFactory
javax.xml.transform.TransformerFactory transFact =
    javax.xml.transform.TransformerFactory.newInstance();
javax.xml.transform.Transformer trans =
    transFact.newTransformer(xsltSource);
trans.transform(xmlSource, result);
```

Quelle: IBM

XML-Verarbeitung mit DB2: XML-Extender



Diskussion

- Vorteile/Nachteile
- Best Practices