

# XML im Überblick

---

**Dr. Thomas Meinike**

thomas.meinike@hs-merseburg.de  
<http://www.iks.hs-merseburg.de/~meinike/>

Hochschule Merseburg (FH)  
Fachbereich Informatik und Kommunikationssysteme  
Fachgruppe Technische Redaktion und Wissenskommunikation

Merseburg, 2004-04-29  
(erweitert im Juni 2006)



# Inhaltliche Schwerpunkte

- ⇒ Ziel des Vortrags
- ⇒ Was ist XML?
- ⇒ Syntaxregeln zur Erstellung von XML-Dokumenten
- ⇒ Aufbau von XML-Vokabularen und Validierung (DTD/Schema)
- ⇒ Formatierung von XML-Inhalten (CSS)
- ⇒ Verarbeitung von XML-Inhalten (XSL)
- ⇒ XML-Anwendungen (XHTML, MathML, RSS und SVG)
- ⇒ Ausblick zu weiteren XML-Technologien wie Web Services
- ⇒ Praktische Demonstrationen / Zusammenfassung



# Ziel des Vortrags

- ⇒ Vermittlung von Grundlagen zu XML sowie weiteren Technologien zur Anwendung und Verarbeitung von XML
- 

## Einige Thesen zum Einstieg:

- ⇒ XML ist eine Kerntechnologie des Informationszeitalters
- ⇒ XML hat grundsätzlich mit der Verarbeitung von Daten zu tun
- ⇒ XML ist nützlich zur strukturierten Ablage von Informationen aller Art (Textinhalte, Messdaten, Vektorgrafiken, usw.)
- ⇒ XML löst aber nicht alle IT-Probleme (erzeugt auch neue ;-)

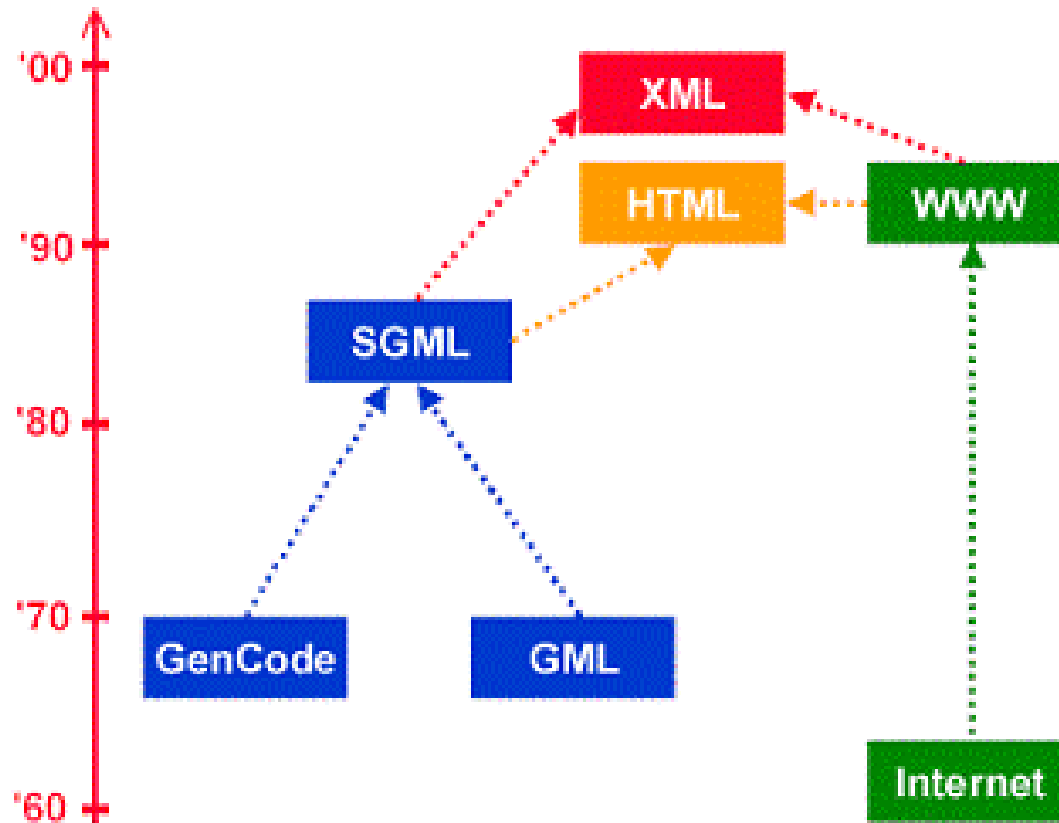


# Was ist XML? → Allgemeines

- ⇒ XML ist die **Extensible Markup Language**, in Version 1.0 1998 vom W3C spezifiziert [Version 1.1/2004].
- ⇒ XML basiert auf SGML (**Standard Generalized Markup Language**, ISO-Standard von 1986).
- ⇒ XML soll die Funktionalitäten von SGML für den Einsatz im Internet zugänglich machen, jedoch in kompakterer Form.
- ⇒ XML ist keine fertige Sprache, sondern ein Konzept für den Entwurf eigener Auszeichnungssprachen.
- ⇒ XML bildet den Kern einer „Technologie-Familie“.



# Was ist XML? → XML-Stammbaum



[Bildquelle: "XMLspektrum" 2000]

# Was ist XML? → Entwurfsziele

- ⇒ XML soll sich im Internet auf einfache Weise nutzen lassen.
- ⇒ XML soll ein breites Spektrum von Anwendungen unterstützen.
- ⇒ XML soll zu SGML kompatibel sein.
- ⇒ Es soll einfach sein, Programme zu schreiben, die XML-Dokumente verarbeiten.
- ⇒ Die Zahl optionaler Merkmale in XML soll minimal sein, idealerweise Null.
- ⇒ XML-Dokumente sollen für Menschen lesbar und angemessen verständlich sein.
- ⇒ Der XML-Entwurf soll zügig abgefasst sein.
- ⇒ Der Entwurf von XML soll formal und präzise sein.
- ⇒ XML-Dokumente sollen leicht zu erstellen sein.
- ⇒ Knappheit von XML-Markup ist von minimaler Bedeutung.

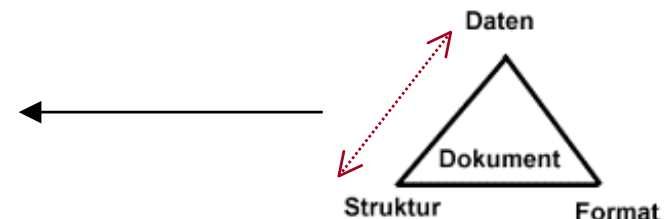


# Was ist XML? → Dokumentenaufbau

⇒ XML-Grundgerüst:

```
<?xml version="1.0"?>  
<wurzelement-name>  
  <!-- weitere Inhalte -->  
</wurzelement-name>
```

Prinzip der Trennung von:



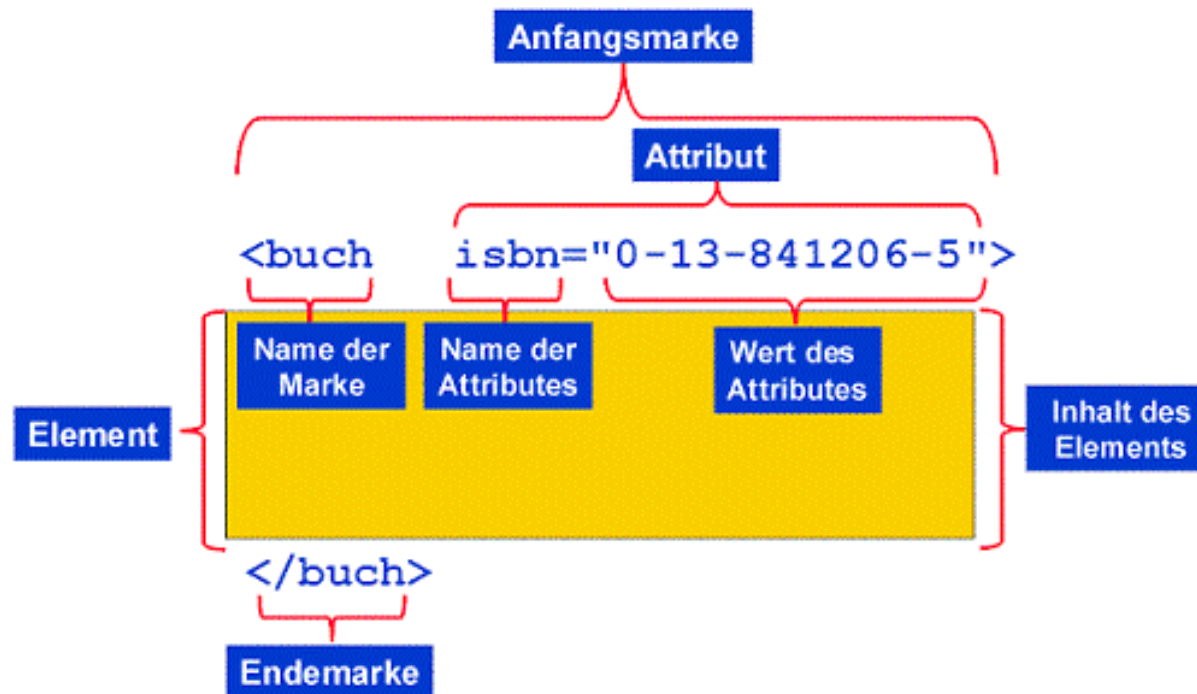
- Der Bereich vor dem Wurzelement wird als Prolog bezeichnet.
- In der ersten Zeile steht die XML-Deklaration.
- Optionale Attribute sind standalone="yes" | "no" sowie die Angabe des verwendeten Zeichensatzes mittels encoding="...", z. B.:
  - UTF-8 / UTF-16 (Unicode)
  - ISO-8859-1 (Latin 1, Westeuropa)
  - ISO-8859-2 (Latin 2, Osteuropa)
  - ...



# Was ist XML? → Dokumentenaufbau

- ⇒ XML verwendet das bereits von HTML (SGML) her bekannte Auszeichnungskonzept unter Verwendung von Elementen, Attributen und Entitys.
- ⇒ Das eigentliche Markup bilden die Tags.

## Aufbau eines Elementes



[Bildquelle: "XMLspektrum" 2000]

# Was ist XML? → Namensräume

⇒ **Ziel:** Vermeidung von Namenskonflikten und Mehrdeutigkeiten bei Bezeichnern von Elementen und Attributen

⇒ **Weg:** Bindung von Elementen an eine eindeutige URI-Referenz

⇒ **Beispiel** (Unterscheidung zwischen Buch- und Personen-Titel):

```
<buchbestand xmlns:buch="http://www.example.com/buch"
             xmlns:person="http://www.example.com/person">
```

```
  <eintrag nr="123">
    <angaben-buch>
      <buch:autor>Mustermann</buch:autor>
      <buch:titel>Mustertitel</buch:titel>
      <!-- ... -->
    </angaben-buch>
    <angaben-besitzer>
      <person:name>Musterfrau</person:name>
      <person:titel>Dr.</person:titel>
      <!-- ... -->
    </angaben-besitzer>
  </eintrag>
```

```
</buchbestand>
```

Angabe eines Standard-Namensraums:

```
<el-name xmlns="http://...">Inhalt</el-name>
```

# Was ist XML? → Beispieldokument

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<ktd-news>
```

```
<eintrag nr="1">
```

```
<titel>Semesterbeginn</titel>
```

```
<info>Das neue Semester an der FH beginnt am 29.03.04. Die  
Studierenden im Bereich KTD absolvieren zunächst eine  
thematische Blockwoche.</info>
```

```
</eintrag>
```

```
<!-- Eintrag mit nr="2" -->
```

```
<eintrag nr="3">
```

```
<titel>XML-Vortrag</titel>
```

```
<info>&quot;XML im Überblick&quot; lautet das Thema eines  
Vortrages, der am 29.04.04 um 17:00 Uhr im Hörsaal 9  
beginnt.</info>
```

```
</eintrag>
```

```
</ktd-news>
```



# Was ist XML? → Beispieldokument

Baumansicht in XML-fähigen Browsern

The image shows two browser windows side-by-side, both displaying the XML tree view of a document. The left window is Mozilla Firefox, and the right window is Microsoft Internet Explorer. Both windows show the same XML structure, which is a news document with three entries.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ktd-news>
  <eintrag nr="1">
    <titel>Semesterbeginn</titel>
    <info>Das neue Semester an der FH beginnt am 29.03.04. Die Studierenden im Bereich KTD absolvieren zunächst eine thematische Blockwoche.</info>
  </eintrag>
  <eintrag nr="2">
    <titel>Tag der offenen Tür</titel>
    <info>Am 28.04.04 ist es wieder soweit: Die FH präsentiert ihre Studiengänge mit diversen Veranstaltungen und Einblicken in den Studienalltag.</info>
  </eintrag>
  <eintrag nr="3">
    <titel>XML-Vortrag</titel>
    <info>"XML im Überblick" lautet das Thema eines Vortrages, der am 29.04.04 um 17:00 Uhr im Hörsaal 9 beginnt.</info>
  </eintrag>
</ktd-news>
```

# Syntaxregeln zur Erstellung von XML-Dokumenten

## ⇒ Namensregeln für Bezeichner:

- Buchstaben (groß/klein, Schreibweise wird unterschieden!)
- Umlaute und Sonderzeichen bei entsprechender Zeichensatzkodierung
- Ziffern 0...9
- Unterstrich, Bindestrich, Punkt
- Doppelpunkt (reserviert für Namensräume, Attribut xmlns:abc="...")
- Erstes Zeichen = Buchstabe (ggf. der Unterstrich)
- nicht xml oder XML am Anfang, reserviert für so genannte Processing Instructions (PI), z. B. `<?xml ... ?>` oder `<?xml-stylesheet ... ?>`
- Länge ist praktisch unbegrenzt



# Syntaxregeln zur Erstellung von XML-Dokumenten

## ⇒ Kriterien für die Wohlgeformtheit (well-formedness):

- Es gibt genau ein Wurzelement, welches alle Inhalte umschließt.
- Alle Elemente sind durch Anfangs- und Endtags korrekt ausgezeichnet. (leere Elemente in der Form `<el ... ></el>` oder `<el ... />`)
- Alle Elemente sind korrekt ineinander verschachtelt. (`<a><b>...</b></a>` und nicht `<a><b>...</a></b>`)
- Attributwerte stehen in Anführungszeichen (paarweise `"..."` oder `'...'`).
- Regeln für die Schreibweise werden eingehalten.
- Reservierte Zeichen sind je nach Kontext ggf. maskiert:  
`< = &lt;`; | `> = &gt;`; | `& = &amp;`; | `" = &quot;`; | `' = &apos;`;
- Ungeparste Inhalte in CDATA-Abschnitten möglich: `<![CDATA[...]]>`
- Kommentare (einzeilig/mehrzeilig): `<!-- ... -->`



# Aufbau von XML-Vokabularen und Validierung

## ⇒ Document Type Definition (DTD):

- Eine DTD enthält das Vokabular, welches in einem XML-Dokument verwendet werden darf (= "Wörterbuch" oder "Rechtschreibung").
- Benannt werden die Namen der Elemente, Attribute, Entitys und Notations. Weiterhin werden Aussagen zu Anzahl und Inhalt gemacht sowie Regeln zur Verschachtelung der Elemente festgelegt (= "Grammatik").
- DTDs werden XML-Dokumenten über die DOCTYPE-Deklaration bekannt gemacht.
- Es können interne oder externe DTDs verwendet werden.
- Dokumente, die auf der Basis einer DTD aufgebaut sind, werden auch als Instanzen dieser DTD bezeichnet.



# Aufbau von XML-Vokabularen und Validierung

## ⇒ Document Type Definition (DTD) - intern:

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<!DOCTYPE ktd-news [Definitionen ...]>  
  
<ktd-news> ... </ktd-news>
```

## ⇒ Document Type Definition (DTD) - extern:

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<!DOCTYPE ktd-news SYSTEM "ktd-news.dtd">  
  
<ktd-news> ... </ktd-news>
```

[spezielle Identifier für DTD-Referenzen: **SYSTEM** = explizite Angabe des Speicherortes, **PUBLIC** = Verweis auf öffentlichen Bezeichner]



# Aufbau von XML-Vokabularen und Validierung

⇒ DTD zum Beispiel (ktd-news.dtd):

```
<!ELEMENT ktd-news ( eintrag+ ) >
<!ELEMENT eintrag ( titel , info ) >
<!ELEMENT titel (#PCDATA) >
<!ELEMENT info (#PCDATA) >
<!ATTLIST eintrag nr CDATA #REQUIRED>
```

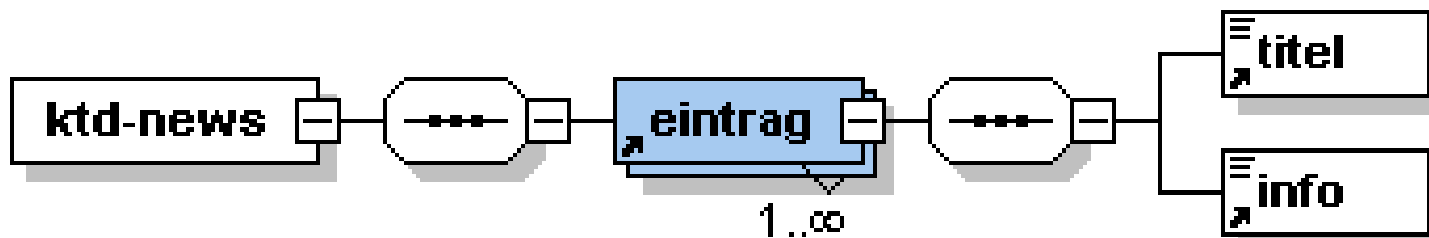
- Definiert werden Elemente und ihr Inhaltsmodell (Elemente, #PCDATA, ggf. EMPTY, ANY)
- Klammern umschließen Gruppen
- Kommata beschreiben die verbindliche Reihenfolge
- Das Zeichen | steht für ODER-Bedingungen
- Indikatoren regeln Anzahl: \* = 0...n, + = 1...n, ? = 0/1
- Attribute von Elementen werden mittels ATTLIST definiert
- Attribute können erforderlich (#REQUIRED), optional (#IMPLIED), oder feststehend (#FIXED) sein, frei wählbare Inhalte sind vom Typ CDATA
- Zeichendaten-Inhalte von Elementen werden als #PCDATA deklariert (Parsed Character DATA)



# Aufbau von XML-Vokabularen und Validierung

- ⇒ XML-Schema (XSD, W3C-Spezifikation 2001):
  - Erweitern die Möglichkeiten der DTD.
  - Sind selbst XML-Dokumente.
  - Komplexe Bedingungen sind formulierbar.
  - Datentypen lassen sich konkreter deklarieren (integer, string, date, ...) und die jeweiligen Inhalte entsprechend überprüfen.
  - Erstellung ist komplexer als bei DTDs (→ Werkzeuge wie XMLSPY).

## Schema-Struktur zum Beispiel



# Aufbau von XML-Vokabularen und Validierung

## ⇒ Schema zum Beispiel (ktd-news.xsd):

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="ktd-news">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="eintrag" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="eintrag">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="titel"/>
        <xsd:element ref="info"/>
      </xsd:sequence>
      <xsd:attribute name="nr" type="xsd:integer" use="required"/>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="info" type="xsd:string"/>

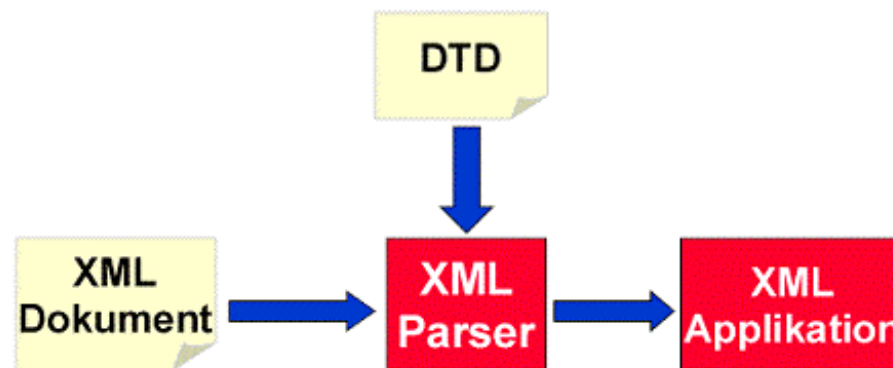
  <xsd:element name="titel" type="xsd:string"/>

</xsd:schema>
```



# Aufbau von XML-Vokabularen und Validierung

- ⇒ **Validierung gegen eine DTD bzw. ein Schema:**
- Ein XML-Parser ist eine Softwarekomponente, die XML-Dokumente einliest und den Knotenbaum analysiert sowie die Daten weiterer Software verfügbar macht: Expat, MSXML, Xerces, XT, ...
- Es existieren validierende und nichtvalidierende Parser.
- Nichtvalidierende Parser prüfen nur auf Wohlgeformtheit.



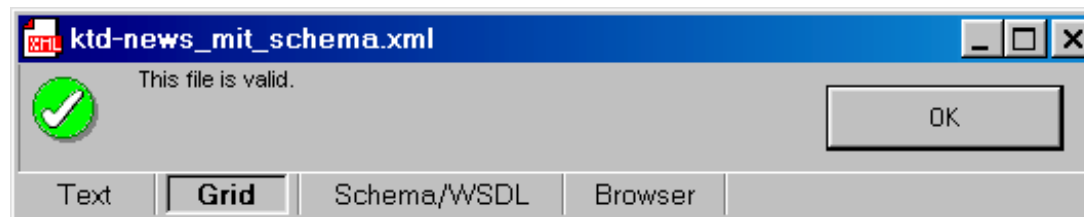
[Bildquelle: "XMLspektrum" 2000]

# Aufbau von XML-Vokabularen und Validierung

- ⇒ **Validierung gegen eine DTD bzw. ein Schema:**
- Beispiel: ktd-news.xml gegen die DTD mit XMLStarlet an der DOS-Kommandozeile:

```
D:\ktd-news\dtd>xml val -d ktd-news.dtd ktd-news.xml  
ktd-news.xml - valid
```

- Beispiel: ktd-news.xml gegen das Schema mit XMLSPY:



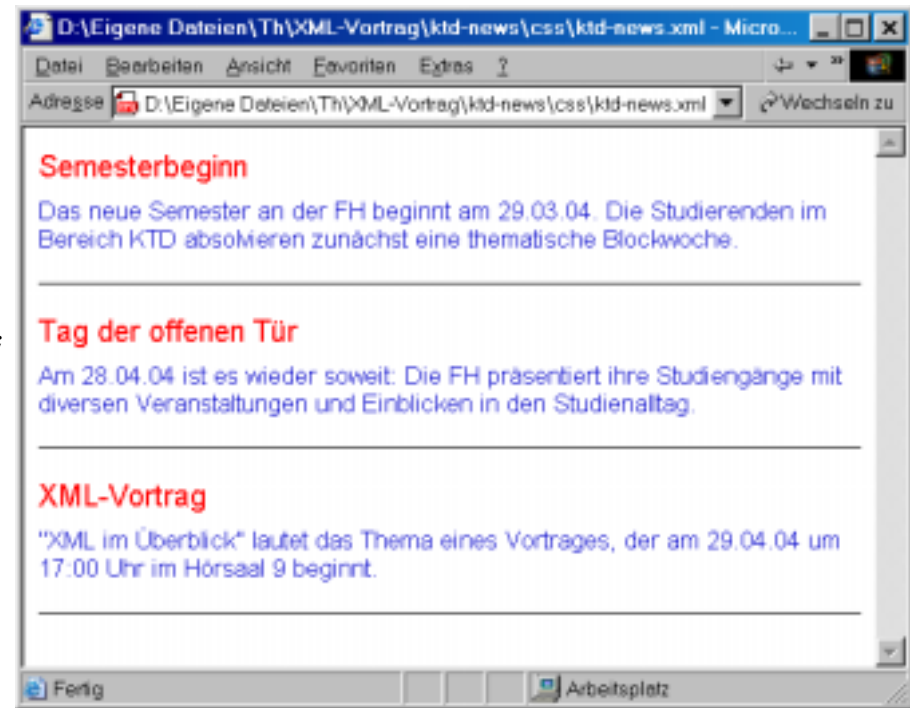
# Formatierung von XML-Inhalten (CSS)

⇒ Client-seitige Anwendung von CSS (ktd-news.css):

```
eintrag          <?xml-stylesheet version="1.0" href="ktd-news.css" type="text/css"?>
{
  display: block;
  margin-bottom: 20px;
  padding-bottom: 20px;
  border-bottom: 1px solid #000;
}

titel
{
  color: #FF0000;
  background-color: #FFFFFF;
  font-family: Arial, Helvetica, Sans-Serif;
  font-size: 120%;
  display: block;
  margin-bottom: 10px;
}

info
{
  color: #0000CC;
  background-color: #FFFFFF;
  font-family: Arial, Helvetica, sans-serif;
  font-size: 100%;
}
```



**Definiert werden Farben, Schrifteigenschaften, Rahmen und Abstände.**

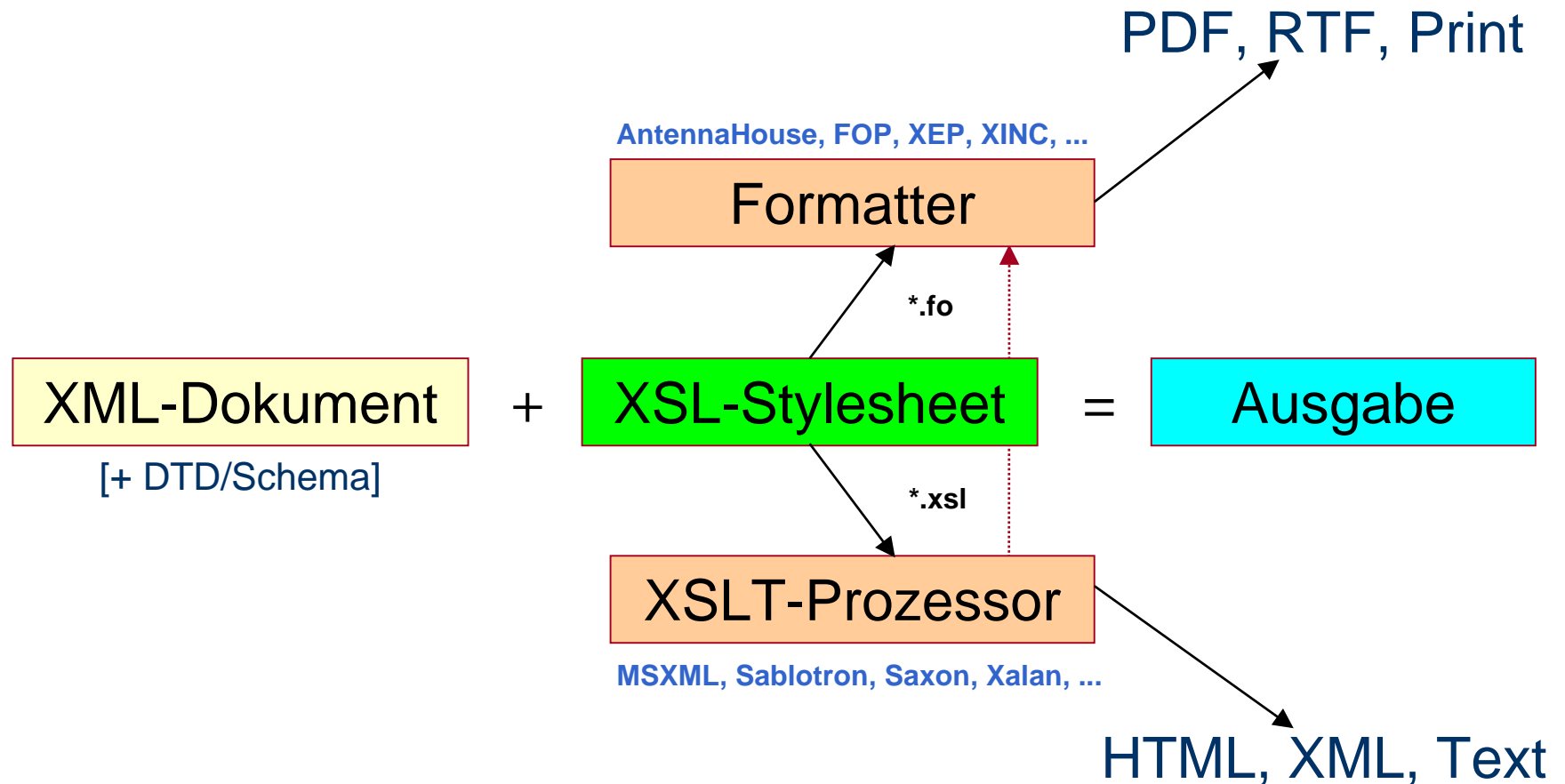
# Verarbeitung von XML-Inhalten (XSL)

- ⇒ XML-Inhalte können in unterschiedliche Zielformate umgewandelt werden, z. B. HTML, Print (PDF, RTF), Text, andere XML-Strukturen.
- ⇒ Anwendung für Single-Source- bzw. Cross-Media-Publishing.
- ⇒ XSL (Extensible Stylesheet Language) dient zur Umsetzung.
- ⇒ XSL wird unterteilt in **XSLT** (Transformations) und **XSL-FO** (Formatting Objects).
- ⇒ Zur Abfrage von XML-Inhalten wird die XML Path Language (**XPath**) verwendet.
- ⇒ Für die praktischen Realisierung wird spezielle Software benötigt: XSLT-Prozessoren bzw. Formatter.



# Verarbeitung von XML-Inhalten (XSL)

⇒ Single-Source-/Cross-Media-Publishing-Prinzip:



# Verarbeitung von XML-Inhalten (XSL)

## ⇒ 1. XSLT-Stylesheet zum Beispiel (ktd-news1.xsl):

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

<xsl:output method="html" />

<!-- Anwendung der Gesamtvorlage ab dem Wurzelknoten -->
<xsl:template match="/">

  <html><head>...</head><body><h1>KTD-News</h1>

  <!-- Jeweilige Auswahl von titel bzw. info unterhalb von ktd-news/eintrag -->
  <xsl:for-each select="ktd-news/eintrag">
    <h2><xsl:value-of select="titel" /></h2>
    <p><xsl:value-of select="info" /></p>
    <hr />
  </xsl:for-each>

  </body></html>

</xsl:template>
</xsl:stylesheet>
```



# Verarbeitung von XML-Inhalten (XSL)

## ⇒ 2. XSLT-Stylesheet zum Beispiel (ktd-news2.xsl):

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

  <xsl:output method="html" />

  <xsl:template match="ktd-news">
    <html><head>...</head><body><h1>KTD-News</h1>
      <xsl:apply-templates />
    </body></html>
  </xsl:template>

  <!-- Anwendung aller passenden Vorlagen fuer die jeweiligen Elemente -->
  <xsl:template match="eintrag"><xsl:apply-templates /><hr /></xsl:template>

  <xsl:template match="titel"><h2>[<xsl:value-of select="../@nr" />]
    <xsl:text> </xsl:text><xsl:value-of select="." /></h2>
  </xsl:template>

  <xsl:template match="info"><p><xsl:value-of select="." /></p></xsl:template>

</xsl:stylesheet>
```



# Verarbeitung von XML-Inhalten (XSL)

⇒ Ergebnis der Transformation nach HTML (Beispiel 1):

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
↵ "http://www.w3.org/TR/html401/loose.dtd">
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
5 <title>KTD-News</title>
6 <style type="text/css">
7 h1 {color: #000000; background-color: #FEFEFE;}
8 h2 {color: #FF0000; background-color: #FEFEFE;}
9 p {color: #0000CC; background-color: #FEFEFE;}
10 </style>
11 </head>
12 <body bgcolor="#FEFEFE" text="#0000CC">
13 <h1>KTD-News</h1>
14 <h2>Semesterbeginn</h2>
15 <p>Das neue Semester an der FH beginnt am 29.03.04. Die Studierenden im Bereich KTD
↵ absolvieren zunächst eine thematische Blockwoche.</p>
16 <hr>
17 <h2>Tag der offenen Tür</h2>
18 <p>Am 28.04.04 ist es wieder soweit: Die FH präsentiert ihre Studiengänge mit diversen
↵ Veranstaltungen und Einblicken in den Studienalltag.</p>
19 <hr>
20 <h2>XML-Vortrag</h2>
21 <p>"XML im Überblick" lautet das Thema eines Vortrages, der am 29.04.04 um 17:00 Uhr im
↵ Hörsaal 9 beginnt.</p>
22 <hr>
23 </body>
24 </html>
```

Aufruf des XSLT-Prozessors (XMLStarlet):

```
D:\ktd-news\xslt>xml tr ktd-news1.xsl ktd-news1.xml > ktd-news1.html
```

# Verarbeitung von XML-Inhalten (XSL)

⇒ Ergebnis der Transformation nach HTML im Browser:

**XSLT**



Beispiel 1



Beispiel 2

# Verarbeitung von XML-Inhalten (XSL)

⇒ XSL-FO-Stylesheet zum Beispiel (ktd-news.fo):

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

  <fo:layout-master-set>

    <fo:simple-page-master master-name="A4-Seite-hoch"
      page-height="29.7cm" page-width="21cm"
      margin-top="2.5cm" margin-bottom="2.5cm"
      margin-left="3cm" margin-right="2cm">

      <fo:region-body />

    </fo:simple-page-master>

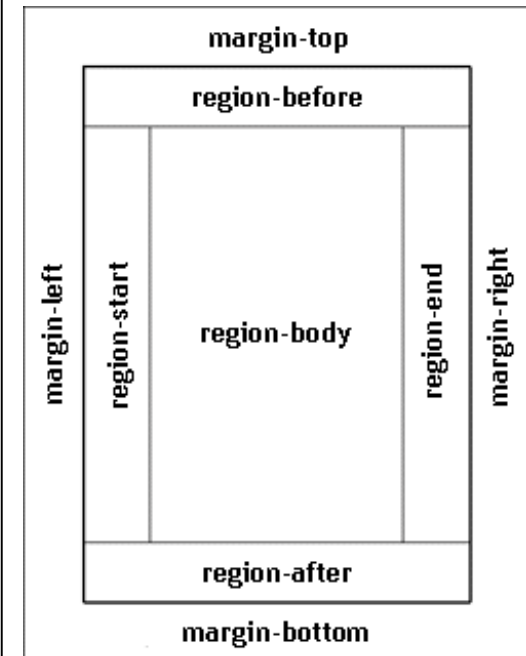
  </fo:layout-master-set>

  <fo:page-sequence master-reference="A4-Seite-hoch">
    <!-- Seiten-orientierte Ausgabe der Inhalte ... -->
  </fo:page-sequence>

</fo:root>
```

↓  
Nächste Folie

FO-Seitenaufbau



# Verarbeitung von XML-Inhalten (XSL)

⇒ XSLFO-Stylesheet zum Beispiel (ktd-news.fo):

```
<fo:page-sequence master-reference="A4-Seite-hoch">

  <fo:flow flow-name="xsl-region-body">

    <fo:block font-family="sans-serif" color="red"
      font-size="18pt" padding-after="10pt">KTD-News</fo:block>

    <fo:block font-family="sans-serif" color="blue" font-size="14pt">
      1. Semesterbeginn</fo:block>

    <fo:block font-family="serif" color="black" font-size="12pt"
      padding-before="10pt" padding-after="20pt">
      Das neue Semester an der FH beginnt am 29.03.04. Die Studierenden
      im Bereich KTD absolvieren zunächst eine thematische Blockwoche.
    </fo:block>

    <!-- weitere Inhalte -->

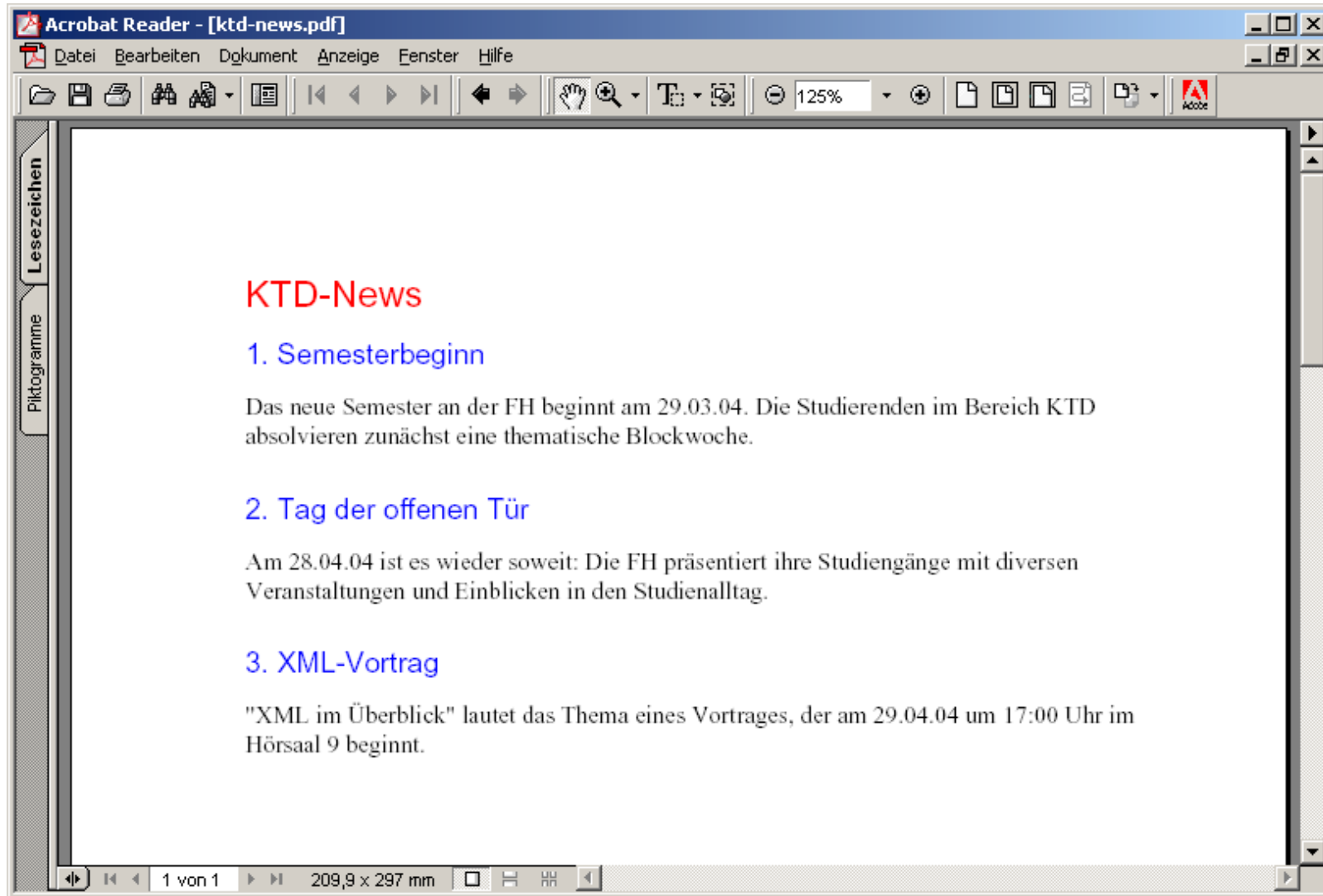
  </fo:flow>

</fo:page-sequence>
```



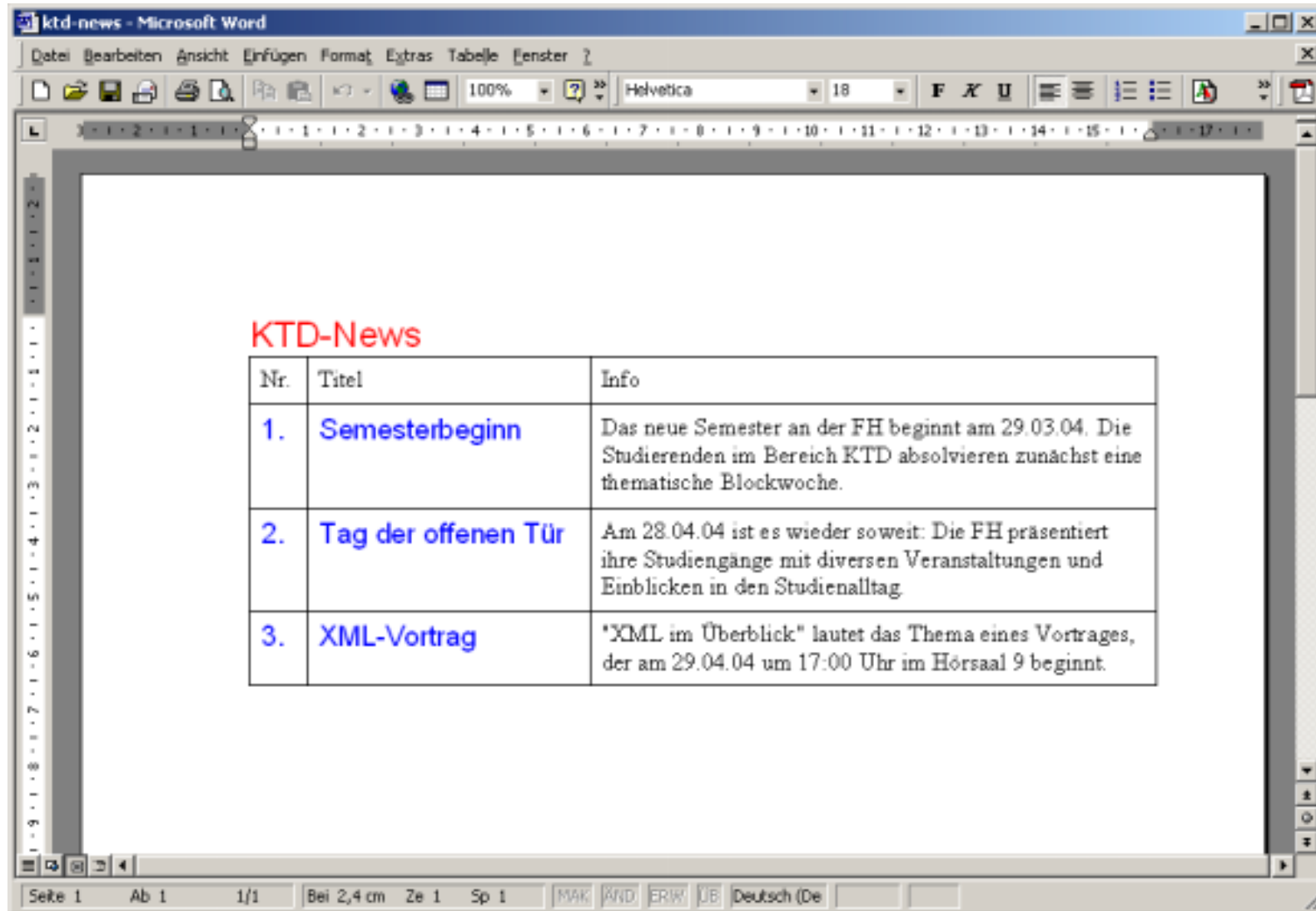
# Verarbeitung von XML-Inhalten (XSL)

⇒ PDF-Ergebnis im Acrobat Reader (Fließtext-Formatierung):



# Verarbeitung von XML-Inhalten (XSL)

⇒ RTF-Ergebnis in Word (Tabellen-Formatierung):



The screenshot shows a Microsoft Word window titled 'ktd-news - Microsoft Word'. The window displays a table with three columns: 'Nr.', 'Titel', and 'Info'. The table contains three rows of news items. The first row has '1.' in the 'Nr.' column, 'Semesterbeginn' in the 'Titel' column, and 'Das neue Semester an der FH beginnt am 29.03.04. Die Studierenden im Bereich KTD absolvieren zunächst eine thematische Blockwoche.' in the 'Info' column. The second row has '2.' in the 'Nr.' column, 'Tag der offenen Tür' in the 'Titel' column, and 'Am 28.04.04 ist es wieder soweit: Die FH präsentiert ihre Studiengänge mit diversen Veranstaltungen und Einblicken in den Studienalltag.' in the 'Info' column. The third row has '3.' in the 'Nr.' column, 'XML-Vortrag' in the 'Titel' column, and '\*XML im Überblick\*' lautet das Thema eines Vortrages, der am 29.04.04 um 17:00 Uhr im Hörsaal 9 beginnt.' in the 'Info' column. The table is titled 'KTD-News' in red text above it. The Word interface includes a menu bar, a toolbar, a status bar, and a scroll bar.

Nr.	Titel	Info
1.	Semesterbeginn	Das neue Semester an der FH beginnt am 29.03.04. Die Studierenden im Bereich KTD absolvieren zunächst eine thematische Blockwoche.
2.	Tag der offenen Tür	Am 28.04.04 ist es wieder soweit: Die FH präsentiert ihre Studiengänge mit diversen Veranstaltungen und Einblicken in den Studienalltag.
3.	XML-Vortrag	*XML im Überblick* lautet das Thema eines Vortrages, der am 29.04.04 um 17:00 Uhr im Hörsaal 9 beginnt.

# XML-Anwendungen → XHTML

- ⇒ XHTML 1.0 = Reformulierung von HTML 4(.01) in XML-Syntax.
- ⇒ Bisherige Elemente nahezu unverändert.
- ⇒ Kleinschreibung aller Element- und Attributnamen.
- ⇒ Korrekte Verschachtelung der Elemente erforderlich.
- ⇒ Attribut-Werte stehen in Anführungszeichen: `border="0"`
- ⇒ Alle Tags werden abgeschlossen / leere Elemente: `<br> → <br />`
- ⇒ Keine standalone-Attribute mehr: `checked → checked="checked"`
- ⇒ Angabe der XML-Deklaration und des jeweiligen DOCTYPEs.
- ⇒ Namespace und Sprache des Dokuments:  
`<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">`
- ⇒ Skript- und CSS-Bereiche sollten in CDATA-Abschnitten stehen.
- ⇒ XHTML 1.1 und höher strebt stärkere Modularisierung an.



# XML-Anwendungen → DocBook

- ⇒ 1991 von O'Reilly & Associates und HaL Computer Systems entwickeltes Dokumentenformat
- ⇒ Grundideen von anderen SGML-basierten Datenaustausch-Projekten übernommen
- ⇒ weitere Entwicklung durch verschiedene IT-Firmen (u. a. Novell)
- ⇒ 1998 durch OASIS neu spezifiziert, auch als XML-DTD seit DocBook 4.0 (OASIS = Organization for the Advancement of Structured Information Standards)
- ⇒ aktuell (06/2006) Version 4.5 (06/06) und Version 5.0 im Working-Draft-Stadium
- ⇒ Software-Empfehlung: XMLmind XML Editor ([xmlmind.com](http://xmlmind.com))  
XMLSpy ([altova.com](http://altova.com))



# XML-Anwendungen → DocBook

- ⇒ Basis bilden zwei Dokumentenklassen: Buch und Artikel
- ⇒ Formale Strukturelemente: Kapitel, Bereiche (Sektionen), Titel, Absätze, Listen, Tabellen, Abbildungen, Verweise, Quellcode-Blöcke, Anhänge für Glossare, Literaturangaben, ...
- ⇒ Einsatzbereich: Publikationen im Bereich der TD, speziell für Hard- und Softwaredokumentationen (z. B. Linux-Handbücher)
- ⇒ XSL-Stylesheets für unterschiedliche Ausgaben fertig verfügbar, die sich an eigene Erfordernisse anpassen lassen
- ⇒ Zur Vertiefung: Diplomarbeit von Johannes Meyknecht (KTD 00) „Cross-Media-Publishing auf der Basis von DocBook-XML“  
PDF-Version: <http://opus.fh-merseburg.de/volltexte/2005/206/>
- ⇒ Weitere Informationen unter [docbook.org](http://docbook.org) verfügbar.



# XML-Anwendungen → DocBook

## ⇒ Prinzipieller Aufbau eines Artikels:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.4//EN"
  "http://www.oasis-open.org/docbook/xml/4.4/docbookx.dtd">
<article lang="de">

  <articleinfo>
    <title>DocBook für die (Technische) Dokumentation</title>
    <subtitle>Eine kurze Einführung</subtitle>
    <date>2006-04-18</date>
    <corpauthor>Dr. Thomas Meinike</corpauthor>
    <abstract>
      <para>Es wird eine kurze Einführung in das Thema DocBook gegeben.</para>
    </abstract>
  </articleinfo>

  <section>
    <title>Was ist DocBook?</title>
    <para>DocBook ist ein Vokabular zur Strukturierung von Publikationen ...</para>
    <para>Neuer Absatz ...</para>
    <para>Neuer Absatz ...</para>
  </section>

  <!-- weitere section-Elemente -->

</article>
```



# XML-Anwendungen → DocBook

⇒ Prinzipieller Aufbau eines Buches:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.4//EN"
  "http://www.oasis-open.org/docbook/xml/4.4/docbookx.dtd">
<book lang="de">

  <bookinfo>
    <title>Buchtitel</title>
    <author><firstname>Vorname</firstname><surname>Nachname</surname></author>
    <publisher><publishername>Name des Verlages</publishername></publisher>
    <isbn>ISBN</isbn><copyright><year>Jahr</year><holder>Name</holder></copyright>
  </bookinfo>

  <toc>
    <tocchap><tocentry>...</tocentry></tocchap>
    <tocchap><tocentry>...</tocentry></tocchap>
  </toc>

  <preface>
    <title>Vorwort</title>
    <para>Warum man dieses Buch lesen sollte ...</para>
    <para>...</para>
  </preface>

  ...
```



# XML-Anwendungen → DocBook

⇒ Prinzipieller Aufbau eines Buches:

```
...
<part>
  <title>Buchteil I</title>
  <chapter>
    <title>Kapiteltitle</title>
    <section>
      <title>Titel des 1. Abschnitts</title>
      <para>Erster Absatz ...</para><para>Zweiter Absatz ...</para><para>...</para>
    </section>
    <!-- weitere Abschnitte (section) -->
  </chapter>
  <!-- weitere Kapitel (chapter) -->
</part>
<!-- weitere Buchteile (part) -->

<appendix>
  <title>Anhang ...</title>
  <para>...</para>
  <!-- hier auch spezielle Elemente wie bibliography verwendbar -->
</appendix>

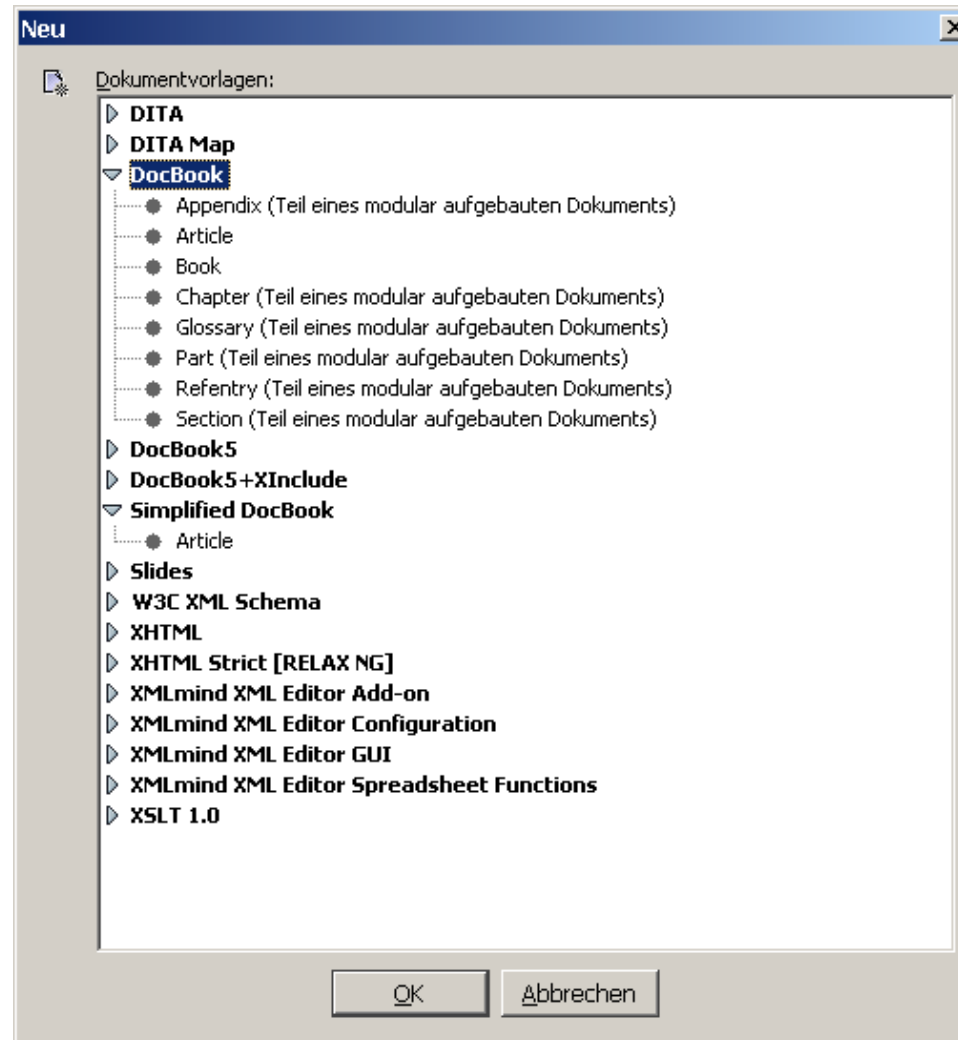
<index>
  <title>Index ...</title>
  <indexentry><primaryie>Text, Seitenzahl</primaryie></indexentry>
</index>

</book>
```



# XML-Anwendungen → DocBook

⇒ DocBook-Nutzung mit dem XMLmind XML Editor:



# XML-Anwendungen → DocBook

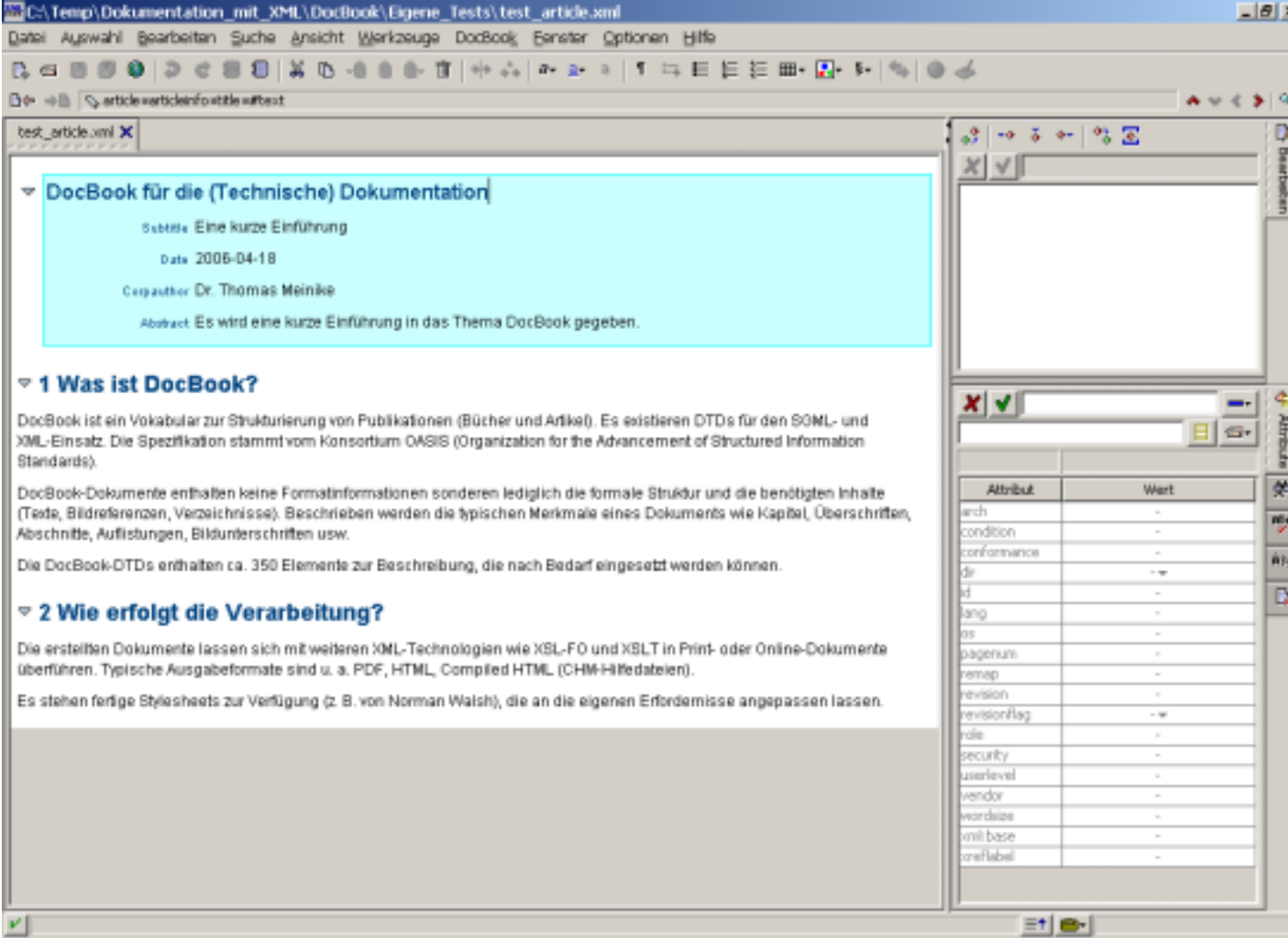
⇒ DocBook-Nutzung mit dem XMLmind XML Editor:

The screenshot displays the XMLmind XML Editor interface. The main window shows a tree view of an XML document titled 'test\_article.xml'. The root element is 'article', which contains several child elements: 'lang' (value 'de'), 'articleinfo', 'section', and 'para'. The 'articleinfo' element contains 'title' (value 'DocBook für die (Technische) Dokumentation'), 'subtle' (value 'Eine kurze Einführung'), 'date' (value '2006-04-18'), 'coauthor' (value 'Dr. Thomas Meinike'), 'abstract', and 'para'. The 'section' element contains 'title' (value 'Was ist DocBook?') and three 'para' elements. The first 'para' under 'section' contains the text: 'DocBook ist ein Vokabelar zur Strukturierung von Publikationen (Bücher und Artikel). Es existieren DTDs für den SGML- und XML-Einsatz. Die Spezifikation stammt vom Konsortium DASSIS (Organization for the Advancement of Structured Information Standard)'. The second 'para' contains: 'DocBook-Dokumente enthalten keine Formatinformationen sondern lediglich die formale Struktur und die benötigten Inhalte (Texte, Bildreferenzen, Verzeichnisse). Beschrieben werden die typischen Merkmale eines Dokuments wie Kapitel, Überschriften, Abschnitte, Aufzählungen, Bildunterschriften usw.' The third 'para' is empty. The 'attribute' table on the right side of the editor shows the following data:

Attribut	Wert
arch	-
class	-
condition	-
conformance	-
de	-w
id	-
lang	de
os	-
parentbook	-
remap	-
revision	-
revisionflag	-w
role	-
security	-
status	-
userlevel	-
vendor	-
wordsize	-
xml:base	-
xml:label	-

# XML-Anwendungen → DocBook

⇒ DocBook-Nutzung mit dem XMLmind XML Editor:



The screenshot shows the XMLmind XML Editor interface. The main window displays a DocBook document structure with the following content:

- DocBook für die (Technische) Dokumentation**
  - Subtitle: Eine kurze Einführung
  - Date: 2006-04-18
  - Coauthor: Dr. Thomas Meinke
  - Abstract: Es wird eine kurze Einführung in das Thema DocBook gegeben.
- 1 Was ist DocBook?**

DocBook ist ein Vokabular zur Strukturierung von Publikationen (Bücher und Artikel). Es existieren DTDs für den SGML- und XML-Einsatz. Die Spezifikation stammt vom Konsortium OASIS (Organization for the Advancement of Structured Information Standards).

DocBook-Dokumente enthalten keine Formatinformationen sondern lediglich die formale Struktur und die benötigten Inhalte (Texte, Bildreferenzen, Verzeichnisse). Beschrieben werden die typischen Merkmale eines Dokuments wie Kapitel, Überschriften, Abschnitte, Aufstufungen, Bildunterschriften usw.

Die DocBook-DTDs enthalten ca. 350 Elemente zur Beschreibung, die nach Bedarf eingesetzt werden können.
- 2 Wie erfolgt die Verarbeitung?**

Die erstellten Dokumente lassen sich mit weiteren XML-Technologien wie XSL-FO und XSLT in Print- oder Online-Dokumente überführen. Typische Ausgabeformate sind u. a. PDF, HTML, Compiled HTML (CHM-Hilfedateien).

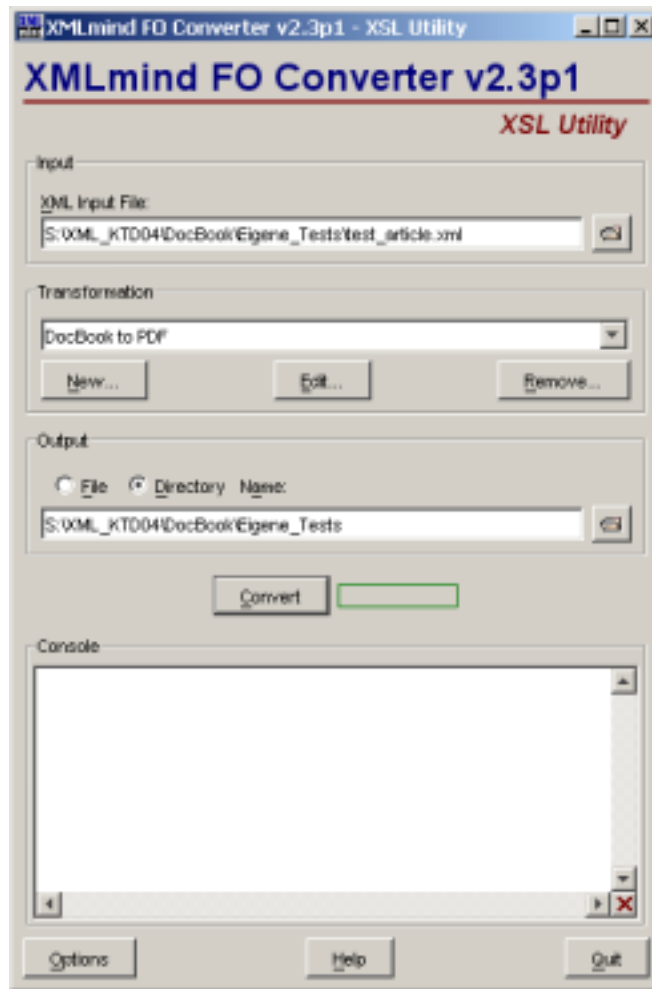
Es stehen fertige Stylesheets zur Verfügung (z. B. von Norman Walsh), die an die eigenen Erfordernisse angepasst lassen.

The right sidebar shows the 'Attributes' panel with a table of attributes and their values:

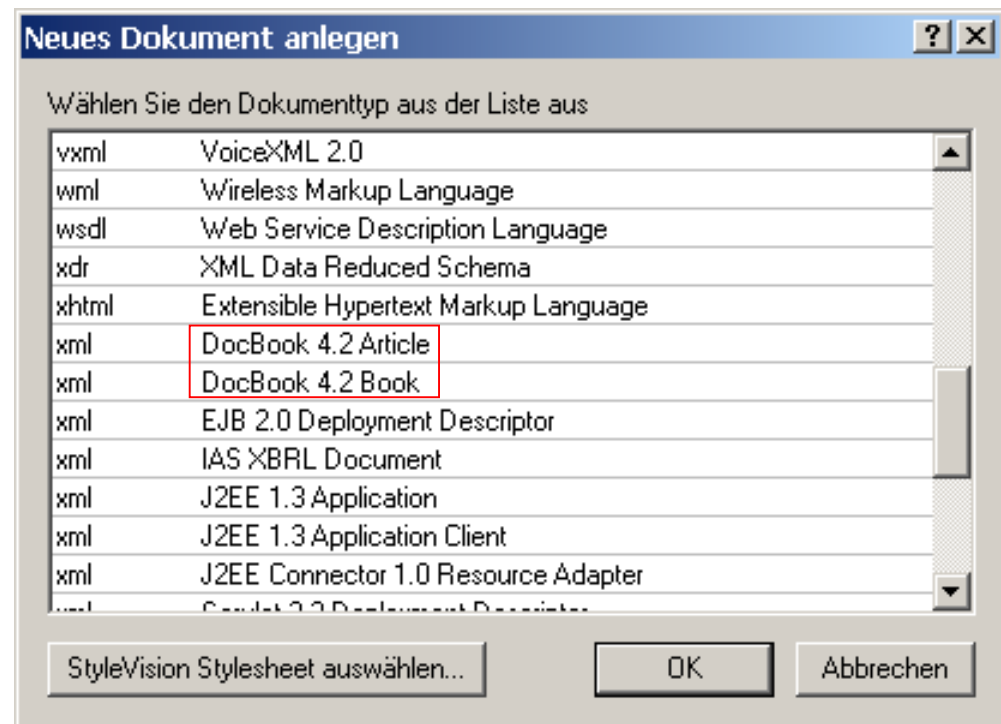
Attribut	Wert
arch	-
condition	-
conformance	-
dir	--w
id	-
lang	-
os	-
pagenum	-
remap	-
revision	-
revisionflag	--w
role	-
security	-
userlevel	-
vendor	-
wordsize	-
xml:base	-
xml:label	-

# XML-Anwendungen → DocBook

⇒ DocBook-Verarbeitung mit dem XMLmind FO Converter / XSL Utility:

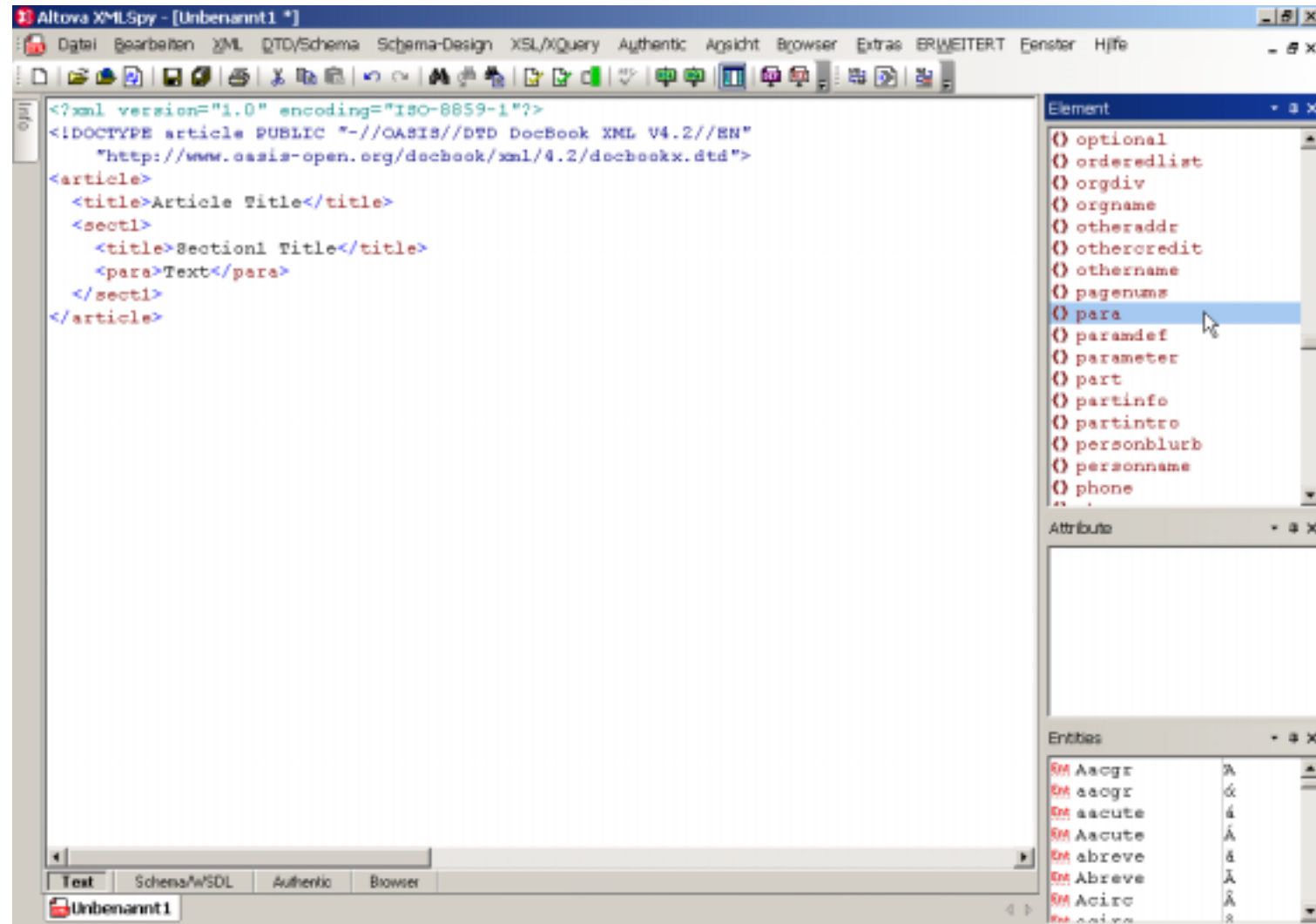


⇒ DocBook-Nutzung mit XMLSpy:



# XML-Anwendungen → DocBook

⇒ DocBook-Nutzung mit XMLSpy:



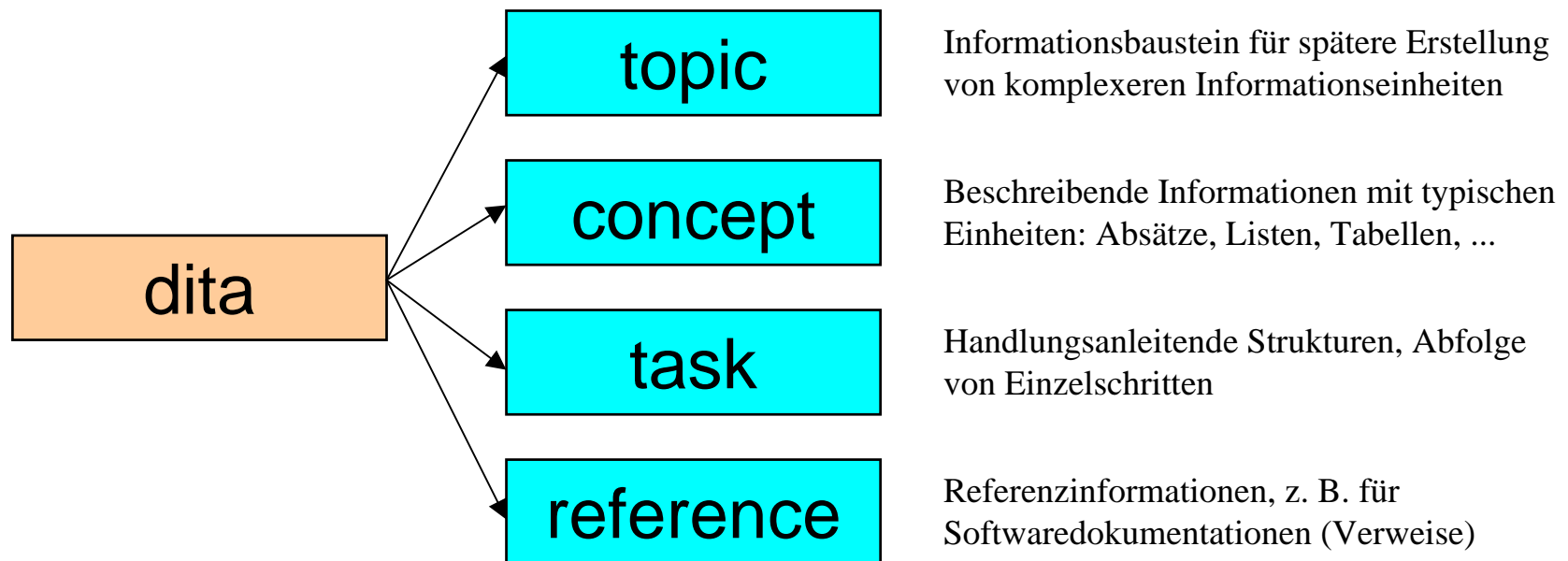
# XML-Anwendungen → DITA

- ⇒ Darwin Information Typing Architecture
- ⇒ Ursprünglich von IBM entwickelt und 2005 als OASIS-Standard spezifiziert (DITA 1.0)
- ⇒ Informationsmodell zur Strukturierung typisierter, Topic-orientierter Inhalte
- ⇒ (Charles) Darwin-Bezug: Erweiterung, Spezialisierung und Vererbung (von Strukturen und Inhalten)
- ⇒ Verwandt mit Konzepten wie InformationMapping®
- ⇒ Basis-XML-DTD → Basis-Architektur: jede Struktur muss auf die Ausgangsstruktur abbilden lassen
- ⇒ Einsatz zur Erstellung, Verteilung und Wiederverwendung von (technischen) Informationen



# XML-Anwendungen → DITA

- ⇒ Topics bilden Informationseinheiten: abgeschlossene, kontextfreie Informationen
- ⇒ Maps dienen zur Zusammenstellung der Topics zu Informationsprodukten (Handbuch, Online-Hilfe, ...)
- ⇒ DITA-Strukturelemente:



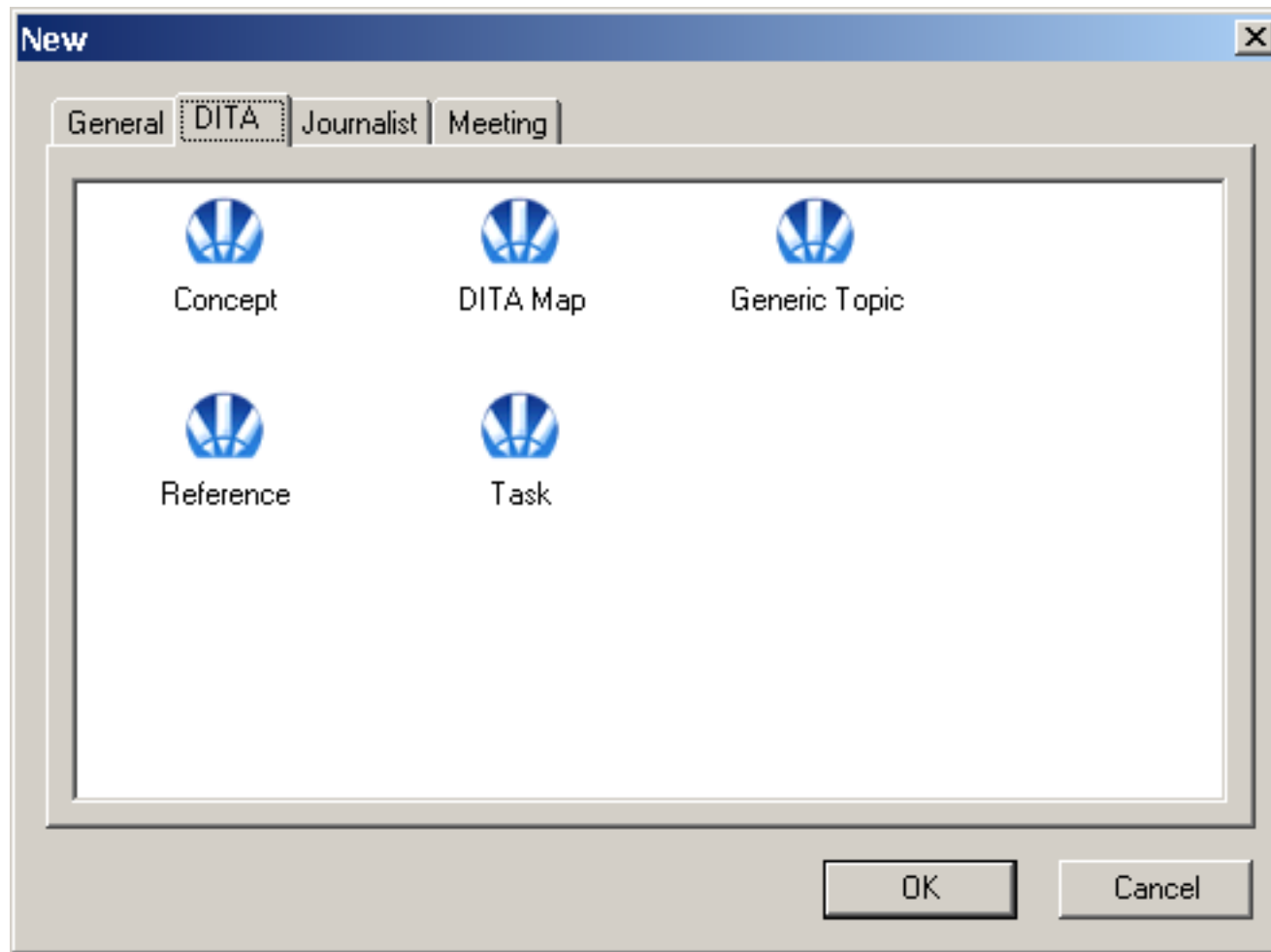
# XML-Anwendungen → DITA

- ⇒ Praktische Anwendungen bisher eher selten zu finden, mit verbesserter Software-Unterstützung wird DITA vermutlich stärker in Betracht gezogen
  
- ⇒ DITA-Konzept im (kritischen) Überblick:  
Torsten Machert (Ovidius GmbH): „DITA - eine neue Strukturierungsmethode?“  
<http://www.zindel.de/zindel/DokuForum-Daten2006/DITA.pdf>
  
- ⇒ Software mit DITA-Unterstützung:
  - XMetaL Author DITA Edition, V4.6 (blastradius.com)
  - XMLmind XML Editor + Addons (xmlmind.com)
  - XMLSpy (altova.com)
  - ...



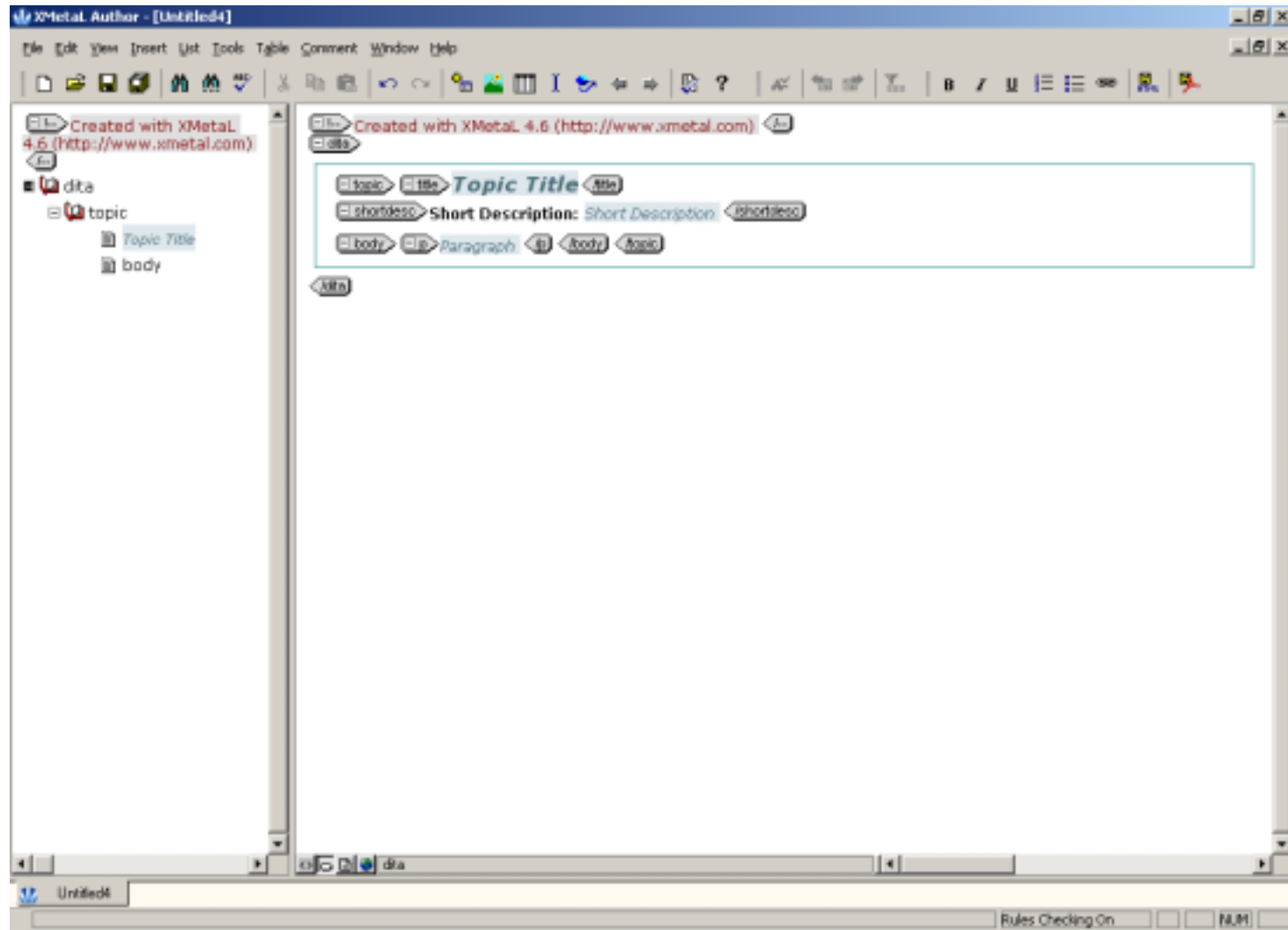
# XML-Anwendungen → DITA

⇒ DITA mit XMetaL:



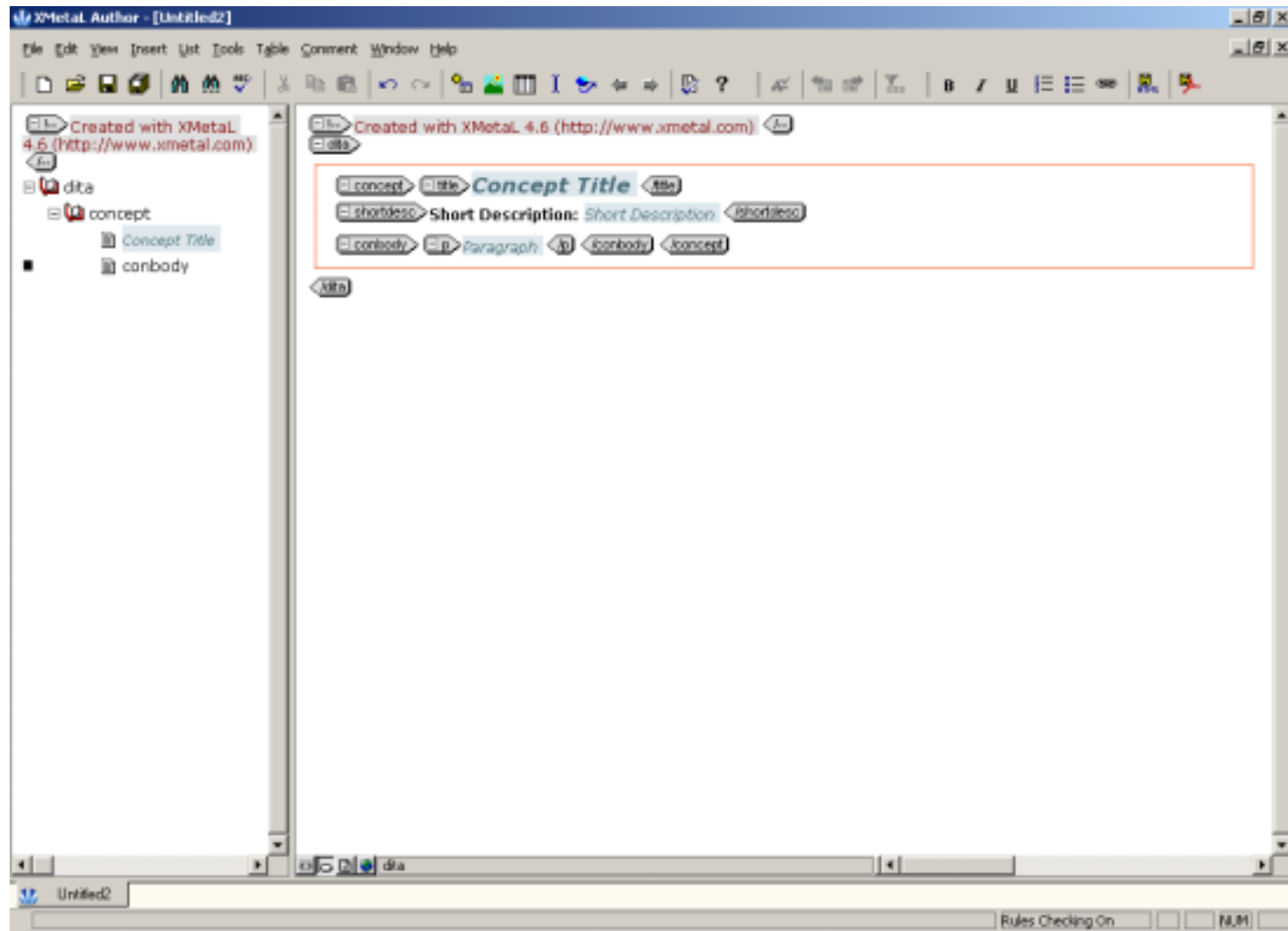
# XML-Anwendungen → DITA

⇒ DITA mit XMetaL:



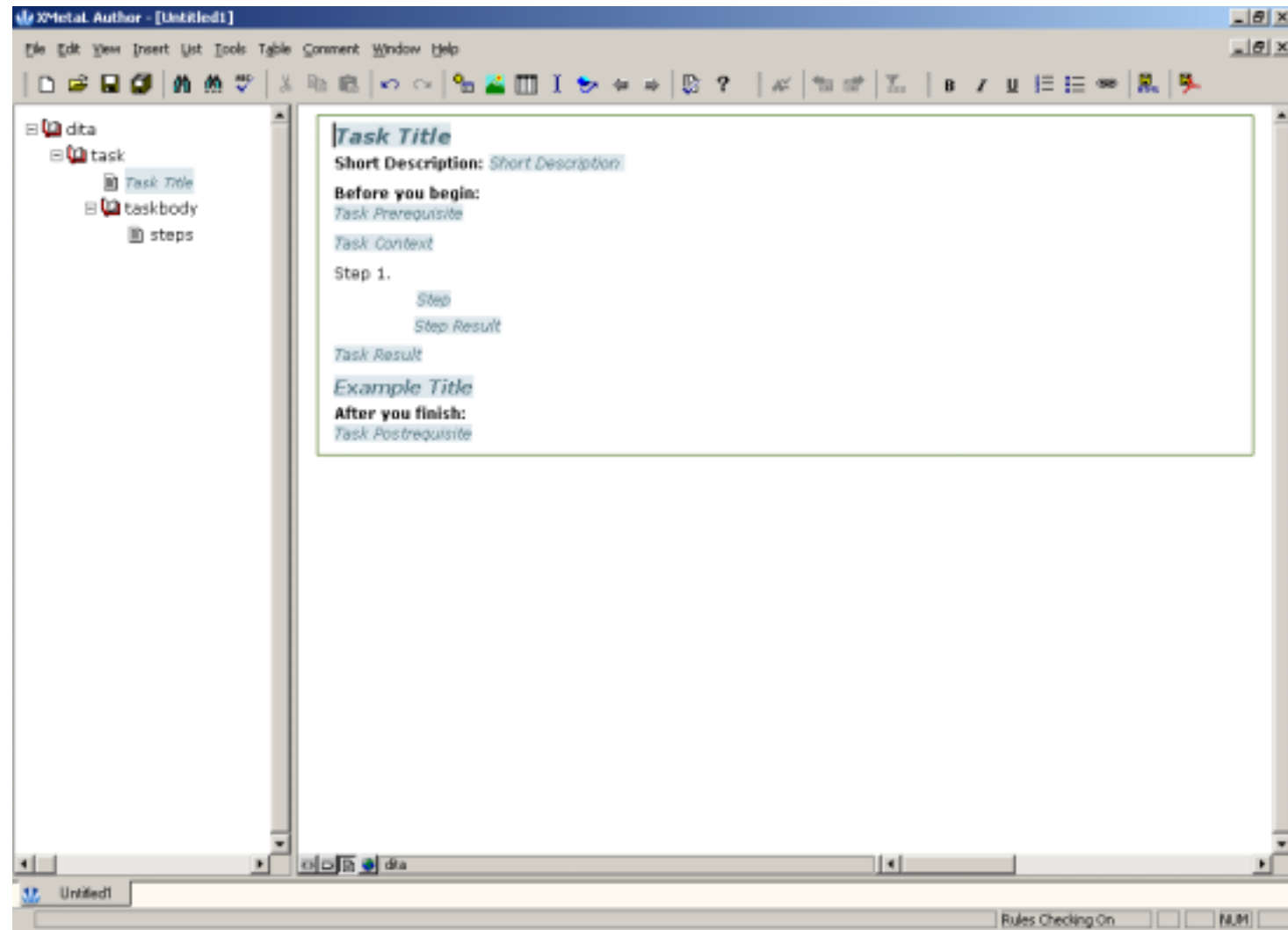
# XML-Anwendungen → DITA

⇒ DITA mit XMetaL:



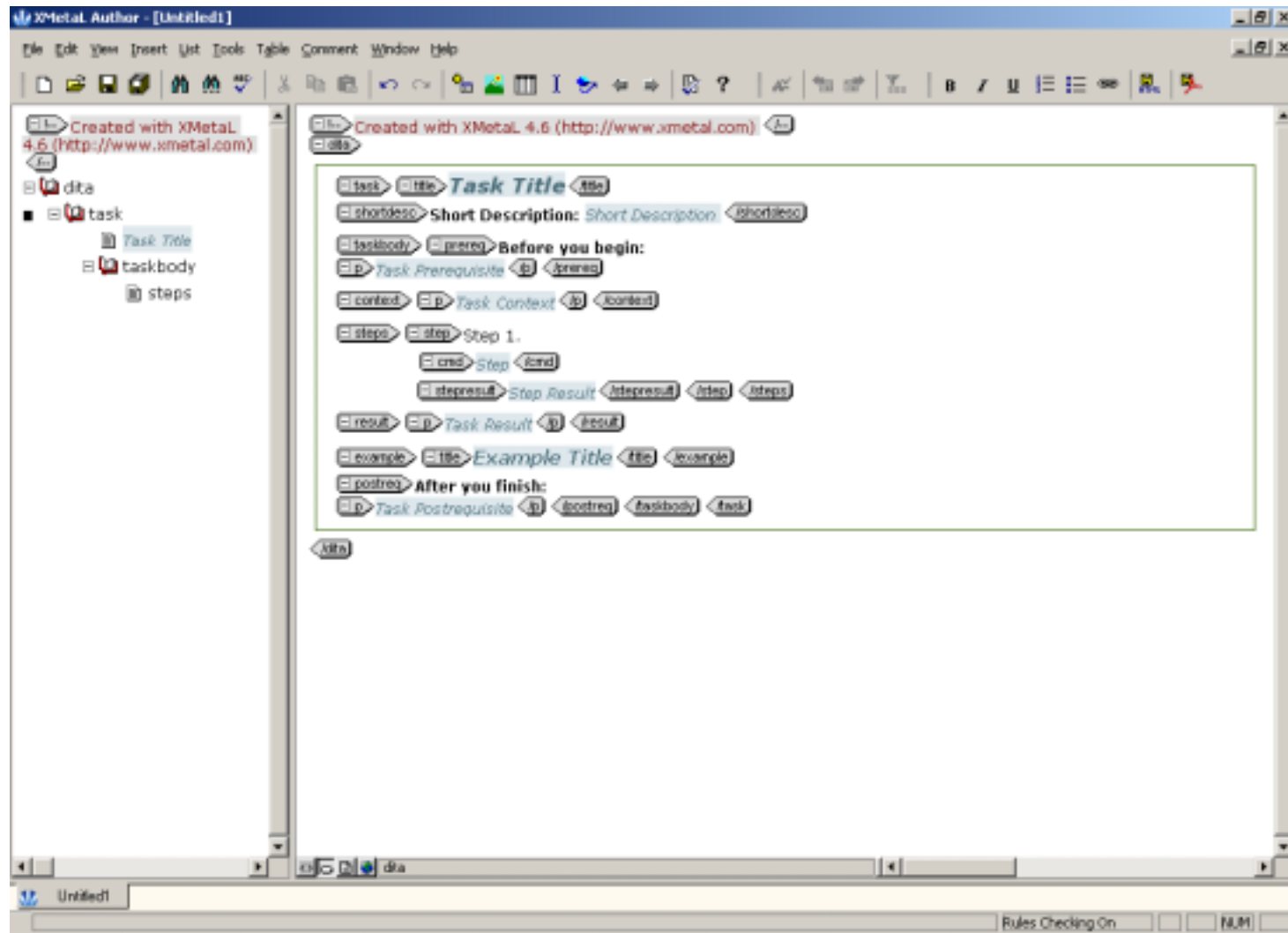
# XML-Anwendungen → DITA

⇒ DITA mit XMetaL:



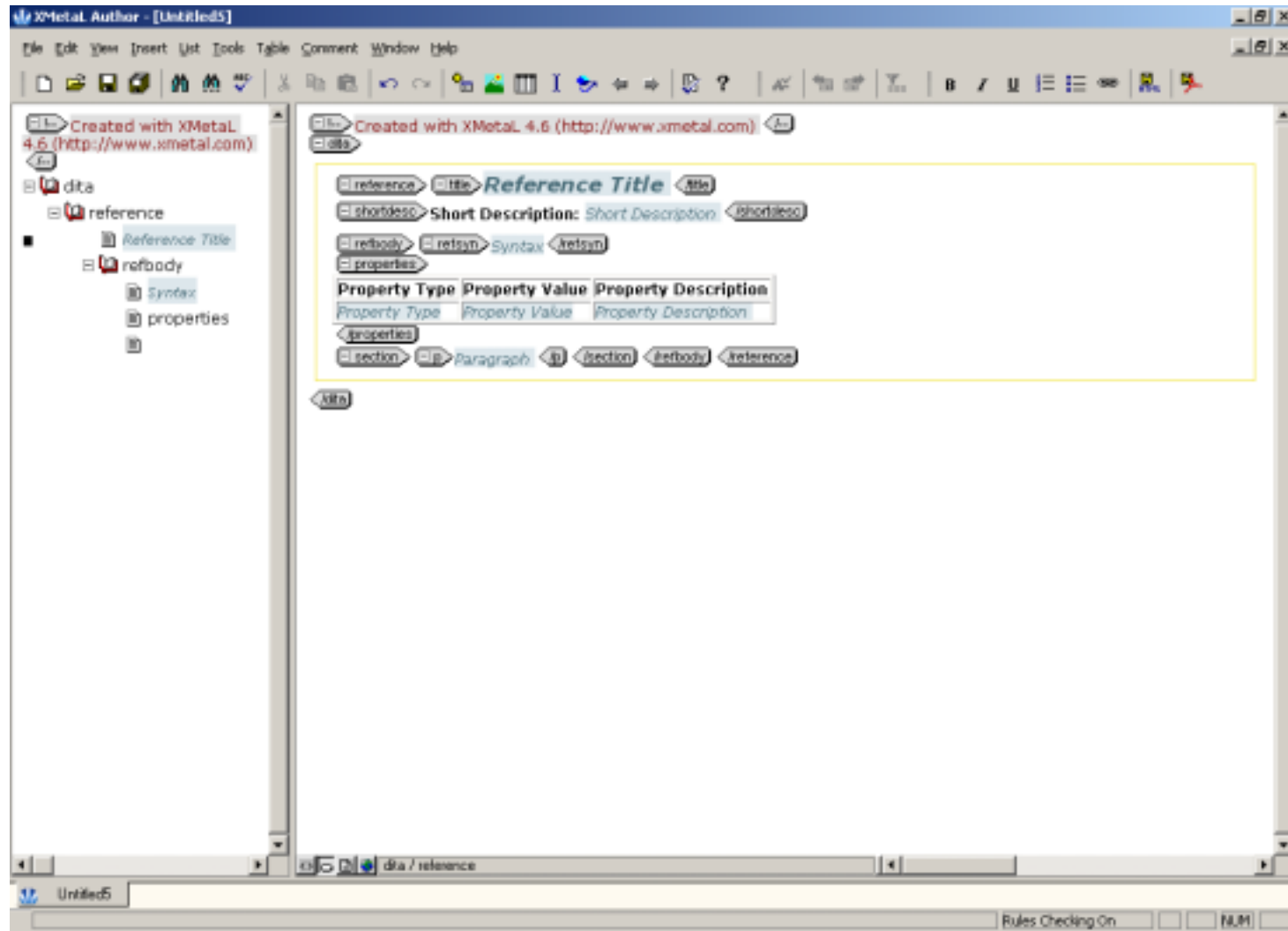
# XML-Anwendungen → DITA

⇒ DITA mit XMetaL:



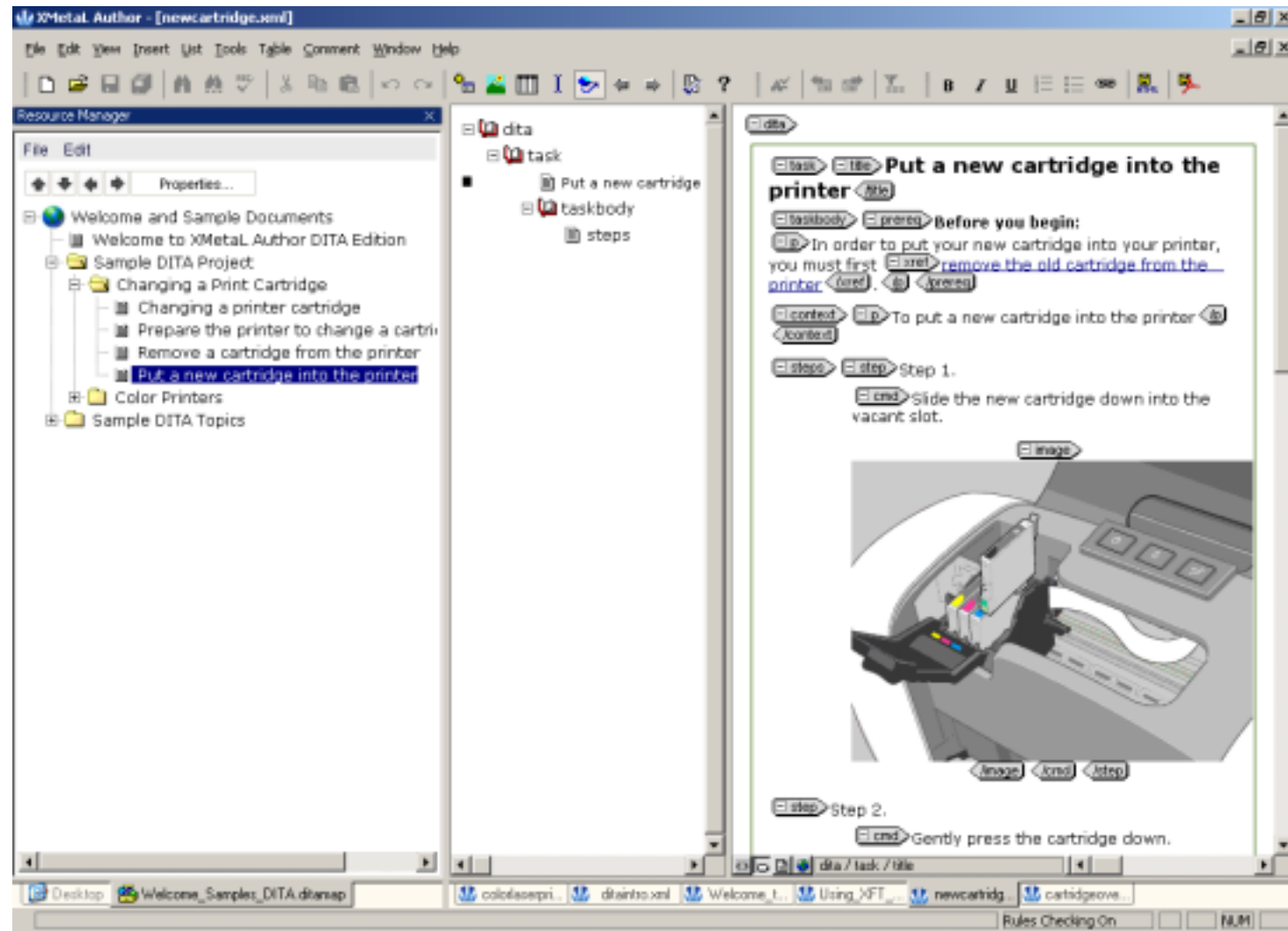
# XML-Anwendungen → DITA

⇒ DITA mit XMetaL:



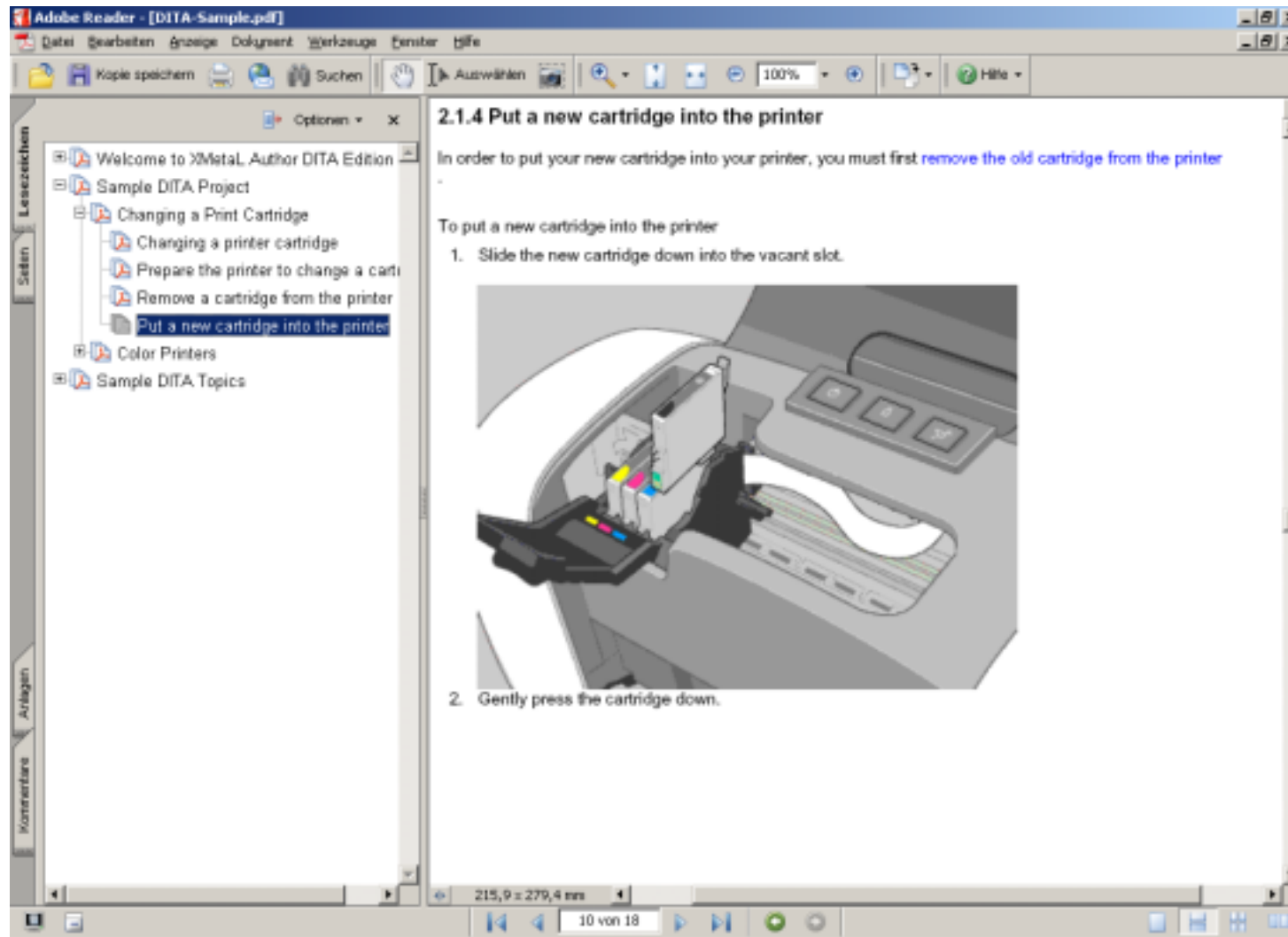
# XML-Anwendungen → DITA

⇒ DITA mit XMetaL:



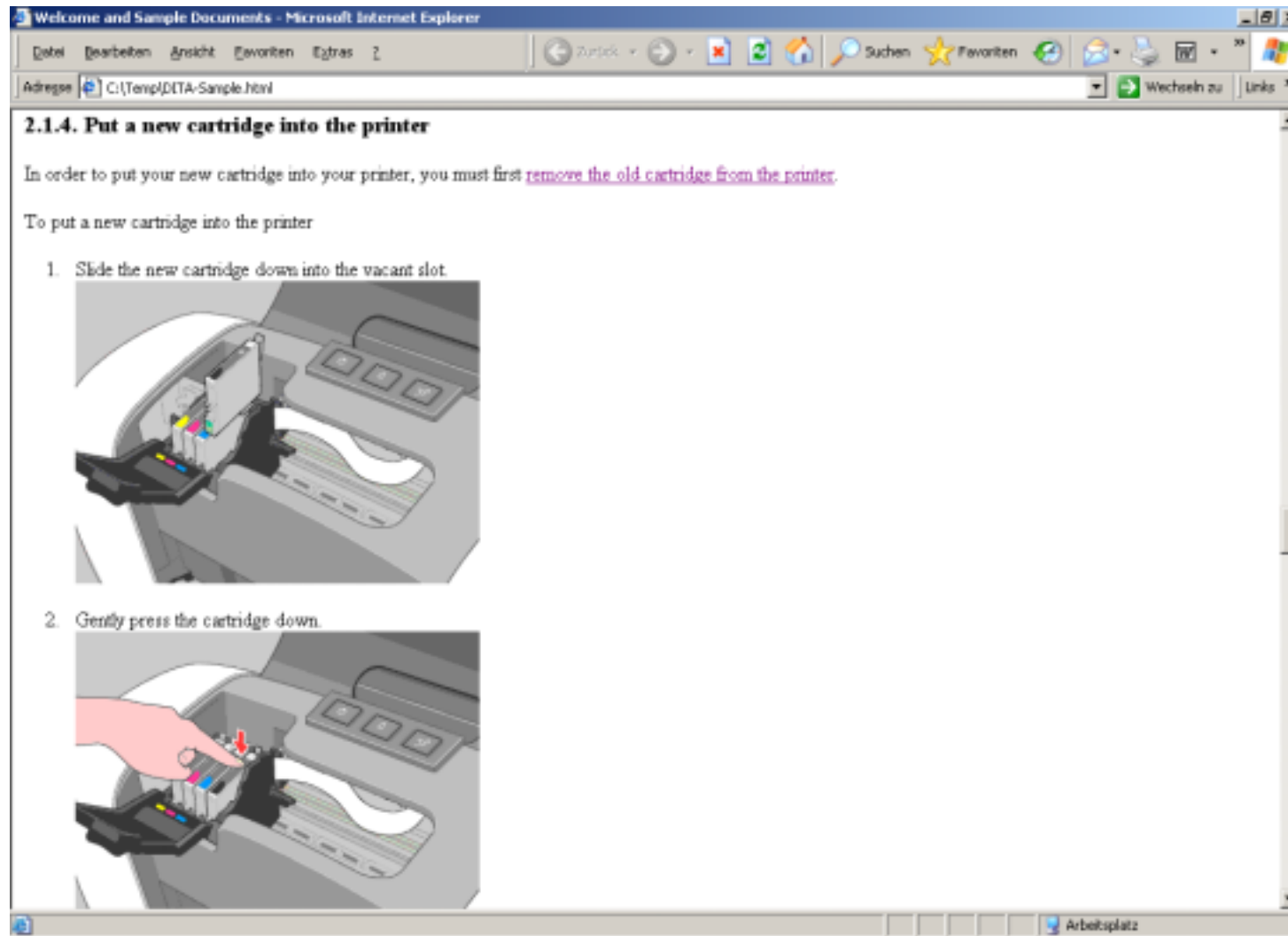
# XML-Anwendungen → DITA

⇒ DITA mit XMetaL:



# XML-Anwendungen → DITA

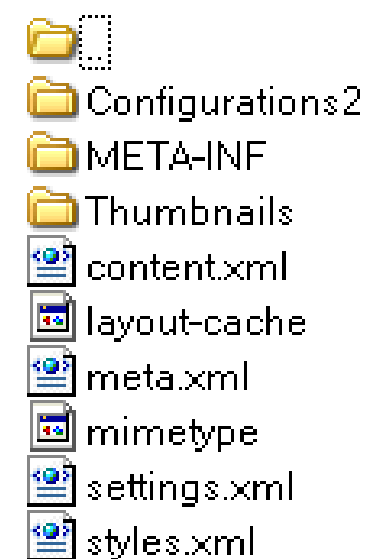
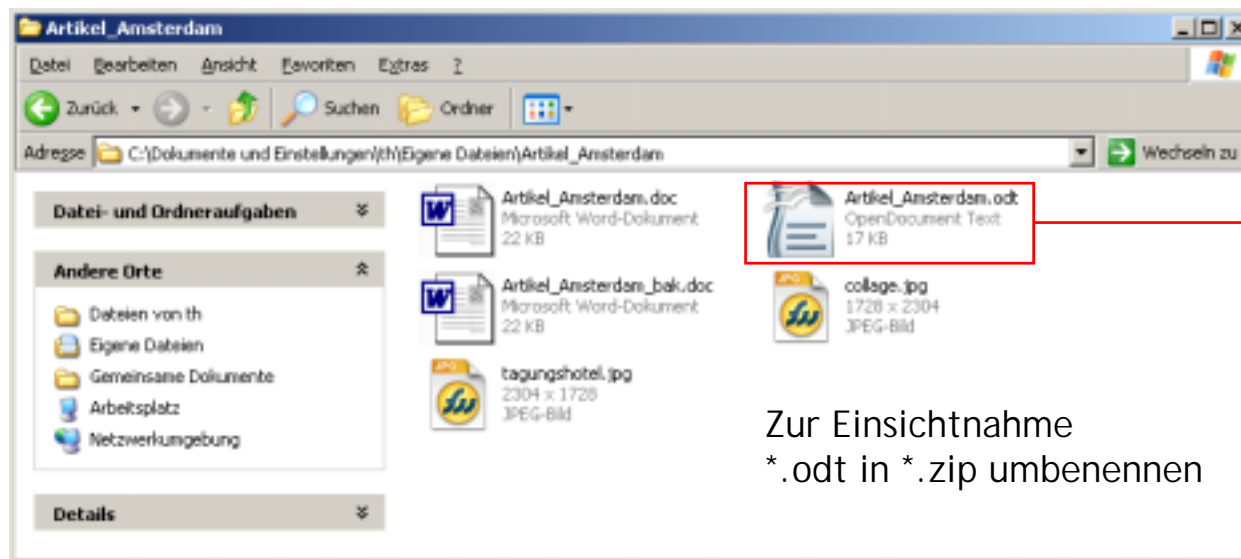
⇒ DITA mit XMetaL:



# XML-Anwendungen → Office-Formate

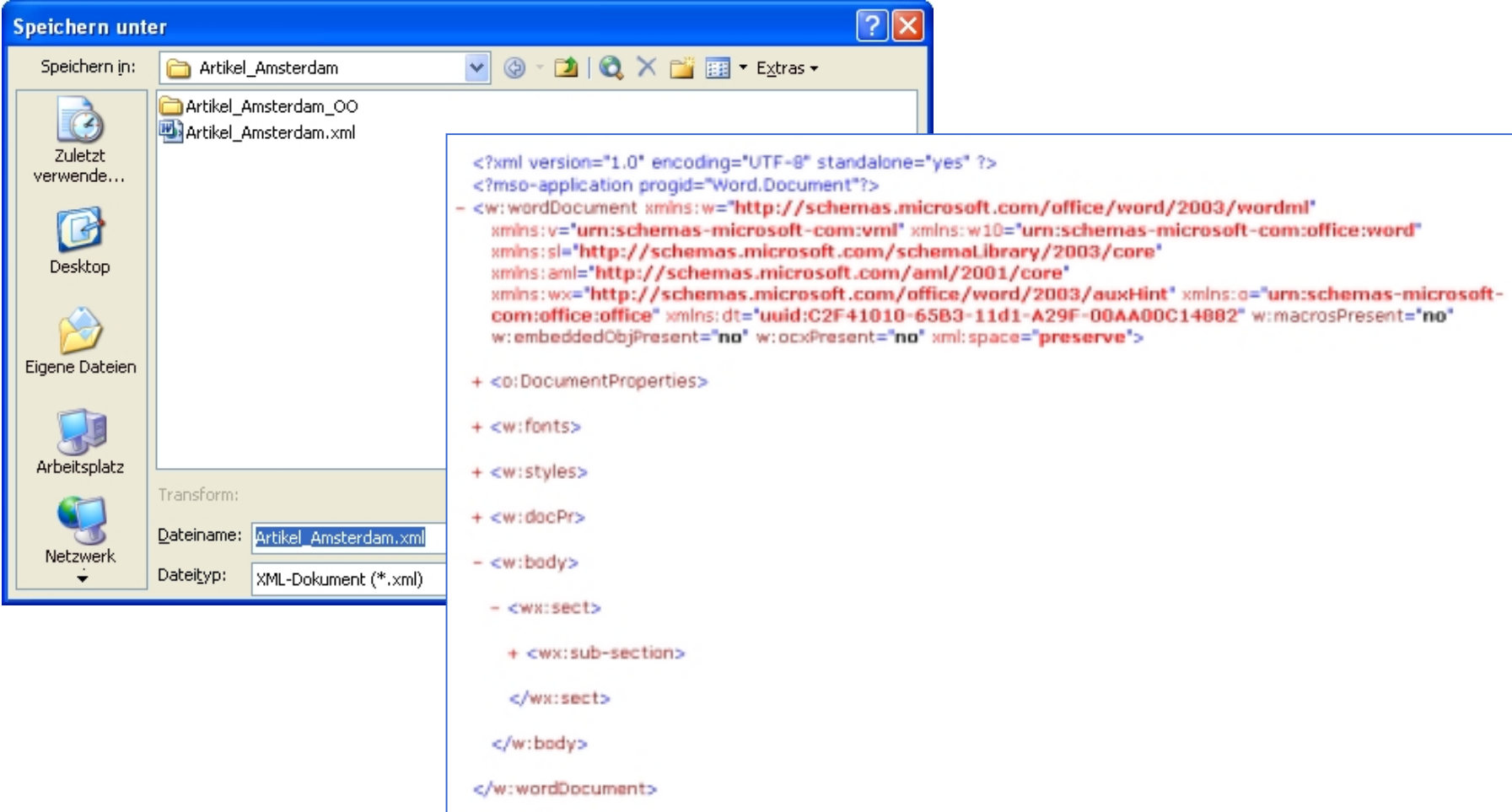
- ⇒ XML-Datenformate für Office-Anwendungen (Text, Präsentation, Tabellenkalkulation, ...) werden zunehmend etabliert
- ⇒ **OpenDocument** von OpenOffice.org (Open-Source-Office-Paket), OASIS-Standard in Version 1.0, 2005 als Norm ISO 26300 verabschiedet (Open Document Format for Office Applications)

XML-Dateien in ZIP-Archiven, u. a. content.xml, meta.xml, settings.xml, styles.xml



# XML-Anwendungen → Office-Formate

- ⇒ Microsoft verwendet eigene XML-Formate: **WordprocessingML** in Office 2003 und **Office Open XML** ab Office 2007



The screenshot shows a Windows 'Speichern unter' (Save As) dialog box. The 'Speichern in:' field is set to 'Artikel\_Amsterdam'. The file list shows 'Artikel\_Amsterdam\_OO' and 'Artikel\_Amsterdam.xml'. The 'Dateiname:' field is 'Artikel\_Amsterdam.xml' and the 'Dateityp:' is 'XML-Dokument (\*.xml)'. An inset window displays the XML code for a Word document, including the root element <w:wordDocument> and various namespace declarations.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<?mso-application progid="Word.Document"?>
- <w:wordDocument xmlns:w="http://schemas.microsoft.com/office/word/2003/wordml"
  xmlns:v="urn:schemas-microsoft-com:vml" xmlns:w10="urn:schemas-microsoft-com:office:word"
  xmlns:sl="http://schemas.microsoft.com/schemaLibrary/2003/core"
  xmlns:aml="http://schemas.microsoft.com/aml/2001/core"
  xmlns:wx="http://schemas.microsoft.com/office/word/2003/auxHint" xmlns:o="urn:schemas-microsoft-com:office:office"
  xmlns:dt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14802" w:macrosPresent="no"
  w:embeddedObjPresent="no" w:ocxPresent="no" xml:space="preserve">
+ <o:DocumentProperties>
+ <w:fonts>
+ <w:styles>
+ <w:docPr>
- <w:body>
- <wx:sect>
+ <wx:sub-section>
</wx:sect>
</w:body>
</w:wordDocument>
```

# XML-Anwendungen → RSS

- ⇒ RSS = [Rich/RDF] Site Summary oder Really Simple Syndication
- ⇒ Konzept zur Verbreitung von Nachrichten-Inhalten
- ⇒ Entwicklung durch UserLand Software (0.9x/2.0, 1.0=RDF)

## ⇒ Prinzipieller Aufbau:

```
<?xml version="1.0"?>
<rss version="2.0">
  <channel>
    <title>...</title><link>...</link><description>...</description>
    <item>
      <title>...</title><link>...</link><description>...</description>
    </item>
    <item>
      <!-- weitere Inhalte -->
    </item>
  </channel>
</rss>
```

RSS 2.0-Spezifikation:  
<http://blogs.law.harvard.edu/tech/rss>  
RSS-Quellen: <http://www.rss-scout.de>



# XML-Anwendungen → RSS

⇒ Beispiel Tagesschau-Newsfeed:

The image consists of three screenshots illustrating the RSS process:

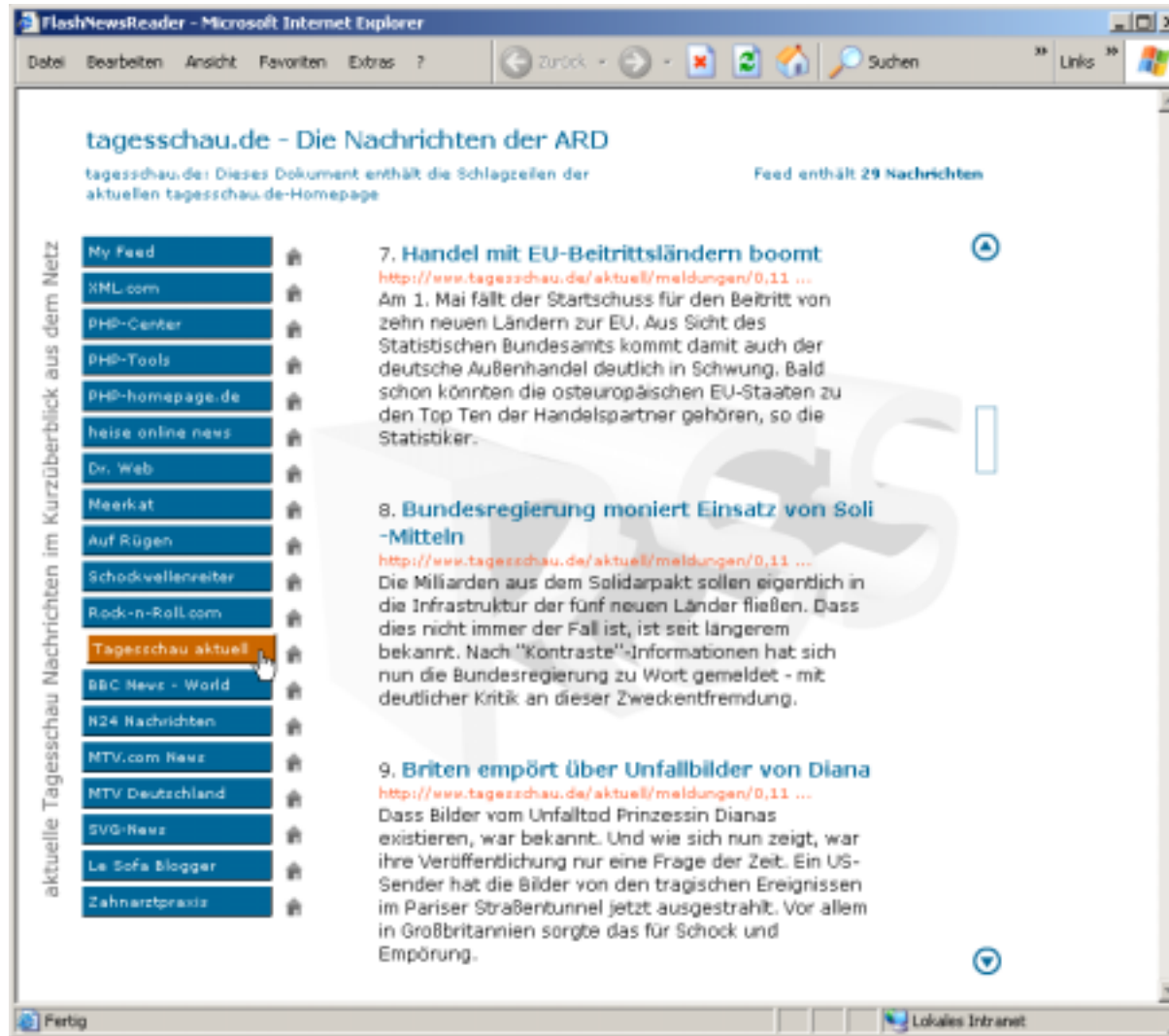
- 1) RSS-Link auf der Website:** A screenshot of the Tagesschau.de website showing the RSS link in the sidebar.
- 2) XML-Teilstrukturansicht:** A screenshot of a web browser displaying the XML feed structure for the RSS link. The XML content is as follows:

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0" ?>
  <channel>
    <title>tagesschau.de - Die Nachrichten der ARD</title>
    <link>http://www.tagesschau.de</link>
    <description>tagesschau.de - Die Nachrichten der ARD</description>
    <item>
      <title>Italien und Tschechien zittern</title>
      <link>http://sport.ard.de/wm2006/wm/news200606/22/vorbericht_italien_tschechien.html</link>
      <description>Der Verlierer der Partie Italien gegen Tschechien scheidet fast sicher aus. Deshalb mobilisiert Tschechiens Trainer alle Spieler, die halbwegs fit sind.</description>
      <guid>http://sport.ard.de/wm2006/wm/news200606/22/vorbericht_italien_tschechien.html</guid>
    </item>
  </channel>
</rss>
```
- 3) Newsreader Opera 9.0:** A screenshot of the Opera 9.0 newsreader displaying the RSS feed. The feed items are listed in a table with columns for 'Von', 'Betreff', 'Datum', and 'Größe'. The selected item is 'Italien und Tschechien zittern'.

- 1) RSS-Link auf der Website
- 2) XML-Teilstrukturansicht
- 3) Newsreader Opera 9.0

# XML-Anwendungen → RSS

⇒ Mit Flash realisierter RSS-Newsreader:



Praktischer Teil der Diplomarbeit von Ralf Hatoum (KTD 99) zum Thema:

„Content Syndication – ein moderner Weg für den Informationsaustausch mit XML“ (2004)

Ergebnisse der Arbeit:

- ✓ RSS-Grundlagen
- ✓ Erkenntnisse zur Kommunikation von Flash mit XML
- ✓ Frei konfigurierbarer Newsreader in Flash

# XML-Anwendungen → SVG

- ⇒ SVG 1.0 ist eine 2001 verabschiedete W3C-Spezifikation zur Beschreibung von 2D-Vektorgrafiken in XML-Syntax [1.1/2003, 1.2/2005].
- ⇒ Enthalten sind geometrische Primitive wie Rechteck, Kreis, Ellipse, Linie, Polylinie, Polygon sowie weitere Elemente zur Darstellung von Pfaden, Text, Hyperlinks usw.
- ⇒ SVG-Inhalte lassen sich mit CSS formatieren und durch den Einsatz von JavaScript-Routinen dynamisch verändern.
- ⇒ Techniken wie Animationen, Gradienten, Filter und Transformationen ermöglichen auch komplexe Darstellungen.

## Grundgerüst:

```
<?xml version="1.0"?>
<!DOCTYPE svg ...>
<svg xmlns="http://www.w3.org/2000/svg">
  <!-- Inhalte der Grafik -->
</svg>
```

## Einige SVG-Elemente mit Attributen:

```
<rect x="..." y="..." width="..." height="..." />
<circle cx="..." cy="..." r="..." />
<ellipse cx="..." cy="..." rx="..." ry="..." />
<line x1="..." y1="..." x2="..." y2="..." />
<polyline points="x1,y1,...,xn,yn" />
<polygon points="x1,y1,...,xn,yn" />
<text x="..." y="...">Textinhalt</text>
...
```



# XML-Anwendungen → SVG

⇒ SVG-Demonstration (10 KB Code):

Objekte und Effekte in SVG [Der rote Kreis, die Erklärungstexte und die Textlinks sind mit JavaScript-Funktionen verknüpft.]

Rechteck

Kreis

Ellipse

Polygon

Polylinie

Linie

Pfad

linearer Gradient

radialer Gradient

Opazität (Durchlässigkeit)

Spezialfilter

Muster

Text

Gruppe+Transformation

externes Bild

Animation

Textlink

normaler Fließtext

© by Dr. Thomas Meinike 2002

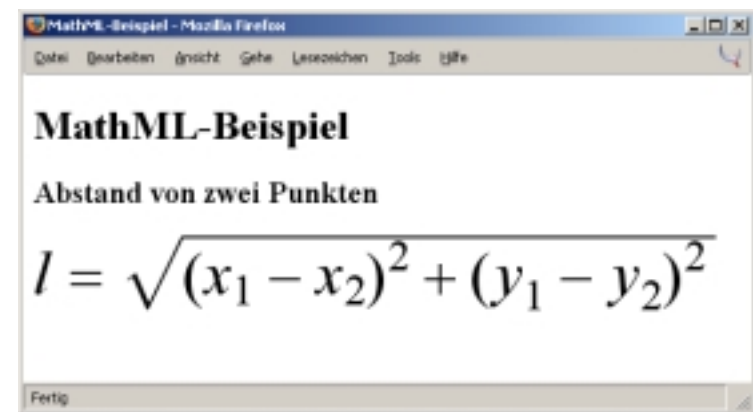
TMs10kSVGDemo.svg

# XML-Anwendungen → MathML

- ⇒ MathML = Mathematical Markup Language
- ⇒ Beschreibung von mathematischen Formeln in XML-Syntax
- ⇒ Einsatz im Bereich Web und Print (Formelsatz) → Beispiel:

```
<?xml version="1.0"?>
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mi>1</mi><mo>=</mo>
    <msqrt>
      <msup>
        <mrow>
          <mo>( </mo>
          <msub><mi>x</mi><mn>1</mn></msub>
          <mo>-</mo>
          <msub><mi>x</mi><mn>2</mn></msub>
          <mo>)</mo>
        </mrow>
        <mn>2</mn>
      </msup>
      ...
    </msqrt>
  </mrow>
</math>
```

<b>math</b>	= Wurzelement
<b>mrow</b>	= Blockbildung
<b>mi</b>	= Variable
<b>mo</b>	= Operator
<b>mn</b>	= Zahlenwert
<b>msub</b>	= tief gestellt
<b>msup</b>	= hoch gestellt
<b>msqrt</b>	= Quadratwurzel
...	



# Weitere XML-Anwendungen

- ⇒ **CML** - Chemical Markup Language (chemische Strukturen)  
→ <http://www.xml-cml.org>
- ⇒ **ebXML** - Electronic Business XML  
→ <http://www.ebxml.org>
- ⇒ **MusicXML** - Austausch von Musikdaten, Notensatz)  
→ <http://www.musicxml.org>
- ⇒ **SMIL** - Synchronized Multimedia Integration Language  
→ <http://www.w3.org/AudioVideo>
- ⇒ **VoiceXML** - für Sprachanwendungen)  
→ <http://www.voicexml.org>
- ⇒ **WML** - Wireless Markup Language (für Handys, PDAs)  
→ <http://www.wapforum.org> und <http://www.wml-tutorial.de>



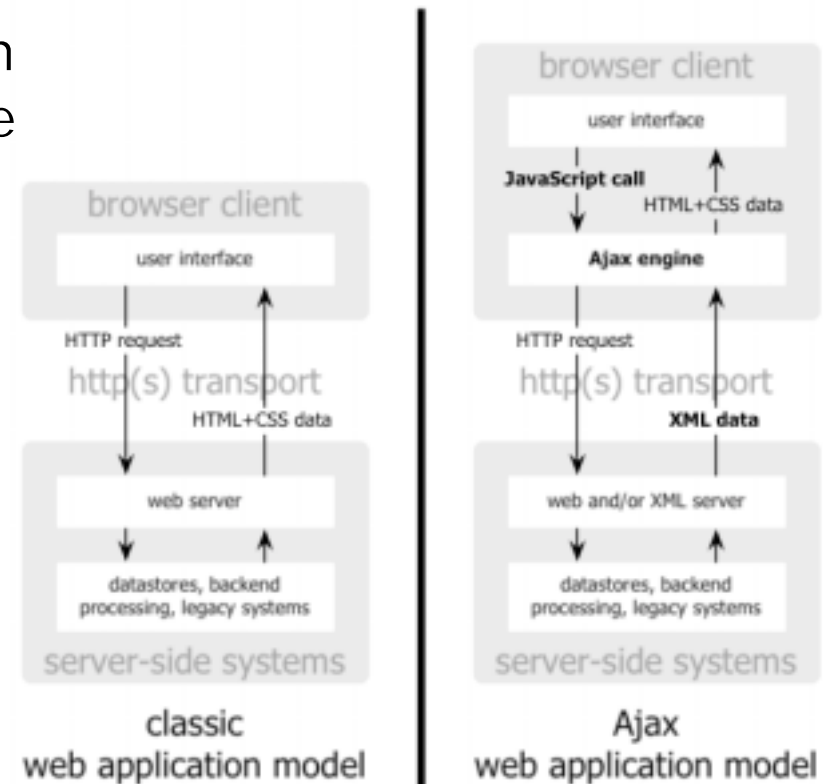
# Weitere XML-Anwendungen

- ⇒ **mumasy** – Multimediales Maschineninformationssystem  
XML-Schema bzw. DTD für typische Anwendungen im Maschinenbau,  
von einem Arbeitskreis des VDMA entwickelt  
(VDMA = Verband Deutscher Maschinen- und Anlagenbau e. V.)  
→ <http://www.mumasy.de> sowie <http://www.vdma.org>
  
- ⇒ **Spec 1000D** sowie **Spec 2000M** – AECMA-Spezifikationen für  
Beschreibung, Wartung und Instandhaltung von Technik (vorwiegend im  
militärischen Bereich eingesetzt, z. B. für Fahrzeug-Dokumentationen)  
  
Spec 2000M behandelt speziell die Erstellung von Ersatzteilkatalogen  
(AECMA = European Association of Aerospace Industries)  
→ <http://www.spec1000d.org> sowie <http://www.aecma.org>



# XML-Anwendungen → AJAX

- ⇒ Asynchronous JavaScript and XML
- ⇒ Datenübertragung zwischen Browser und Server ohne Neuladen der HTML-Seiten bei weiteren HTTP-Anfragen
- ⇒ Ziel: Desktop-ähnliche Webanwendungen mit hohem Grad an Interaktion durch die Benutzer
- ⇒ Kernkonzept des Web 2.0 (Blogs, Communities, Social Software, ...)
- ⇒ Begriff AJAX wurde von Jesse James Garrett im Februar 2005 geprägt
- ⇒ (X)HTML + JS + CSS + DOM  
XMLHttpRequest-Objekt (W3C-Draft)



[Bildquelle: <http://www.adaptivepath.com/publications/essays/archives/000385.php>]



# XML-Anwendungen → AJAX

⇒ AJAX-Beispiele:

**AJAX-Beispiel**

Daten werden dynamisch aus [adressbuch.xml](#) gelesen.

Suchbegriff (Vorname oder Nachname):

Vorname	Hans
Nachname	Mustermann
Straße	Musterstraße 1
Ort	Musterstadt
PLZ	01234
Telefon	0123-12345678
E-Mail	<a href="mailto:hans@mustermann.de">hans@mustermann.de</a>
Typ	privat

HTML + XML-Daten aus Datei

**Periodensystem der Elemente mit Online-Datenabfrage**

Hauptgruppe I II III IV V VI VII VIII  
 I II III IV V VI VII VIII  
 1 H He  
 2 Li Be B C N O F Ne  
 3 Na Mg III IV V VI VII VIII I II Al Si P S Cl Ar  
 4 K Ca Sc Ti V Cr Mn Fe Co Ni Cu Zn Ga Ge As Se Br Kr  
 5 Rb Sr Y Zr Nb Mo Tc Ru Rh Pd Ag Cd In Sn Sb Te I Xe  
 6 Cs Ba La Hf Ta W Re Os Ir Pt Au Hg Tl Pb Bi Po At Rn  
 7 Fr Ra Ac Ku Hs Sg Bh Hs Mt Uun Uuu Uub

Lanthanoide Ce Pr Nd Pm Sm Eu Gd Tb Dy Ho Er Tm Yb Lu  
 Actinoide Th Pa U Np Pu Am Cm Bk Cf Es Fm Md No Lr

Zuordnung der Farben:  
 ■ > Nichtmetalle  
 ■ > Halbmetalle  
 ■ > Hauptgruppen-Metalle  
 ■ > Nebengruppen-Metalle  
 ■ > Aktuelles Element

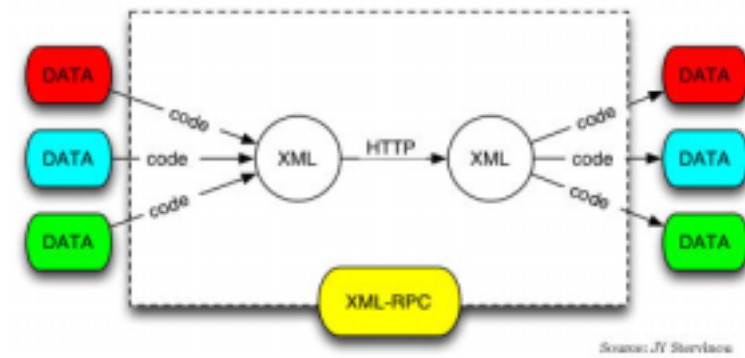
PHP-Skript zur Datenabfrage

Daten von WebserviceX.NET  
 Symbol: Ir Density: 22650 kg / m<sup>3</sup>  
 ElementName: Iridium MeltingPoint: 2683.000000 K  
 AtomicNumber: 77 BoilingPoint: 4800.000000 K  
 AtomicWeight: 192.220000 u ElectroNegativity: 1.550000  
 AtomicRadius: 1.260000 Å IonisationPotential: 8.680000 eV

SVG + XML-Daten (von Webservice)

# XML-Anwendungen → XML-RPC

- ⇒ RPC = Remote Procedure Call
- ⇒ Datenaustausch über HTTP
- ⇒ Daten in XML-Strukturen verpackt
- ⇒ Ablauf der Kommunikation:
  - Beispiel: Abfrage eines Messwertes (Temperatur)



[Bildquelle: <http://www.xmlrpc.com>]

## Anfrage (Request):

```
POST /xmlrpc HTTP 1.0
User-Agent: MyProgram
Host: myhost.example.com
Content-Type: text/xml
Content-Length: ...
```

```
<?xml version="1.0"?>
<methodCall>
  <methodName>getData</methodName>
  <params>
    <param>
      <value><string>temp</string></value>
    </param>
  </params>
</methodCall>
```

## Ergebnis (Response):

```
HTTP/1.0 200 OK
Connection: close
Content-Type: text/xml
Content-Length: ...
Date: Thu, 29 Apr 2004 12:00:00 GMT
Server: ...

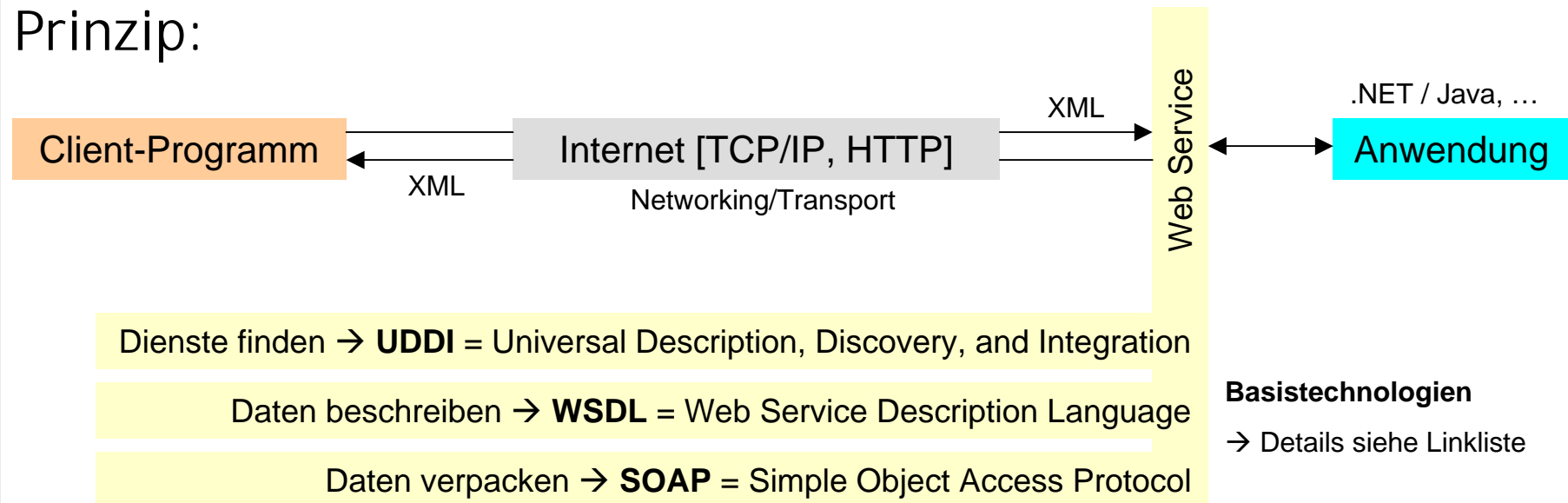
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><double>20.25</double></value>
    </param>
  </params>
</methodResponse>
```



# XML-Anwendungen → Web Services

⇒ Web Services sind Schnittstellen zu Anwendungen

⇒ Prinzip:



⇒ Beispiel - Abfrage von Wetterdaten (Global Weather):

<http://www.webservicex.net/WS/WSDetails.aspx?CATID=12&WSID=56>

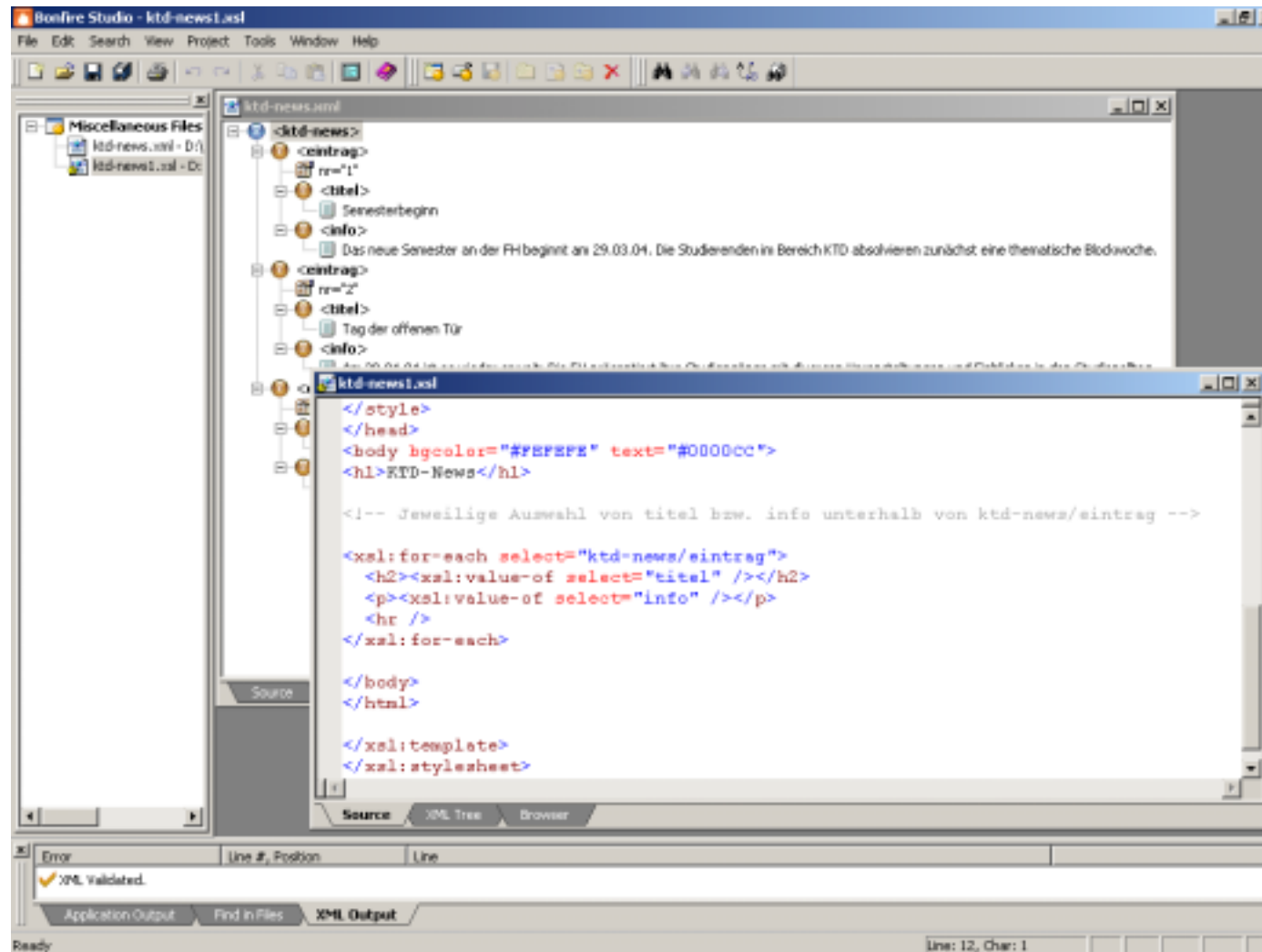
⇒ Beispiel - Abfrage von Daten zu chemischen Elementen (Periodic Table):

<http://www.webservicex.net/WS/WSDetails.aspx?CATID=7&WSID=19>



# Praktische Demonstrationen

⇒ XML-Software / Beispiele zum Vortrag



Freeware: Bonfire Studio – <http://www.nzworks.com>

# Zusammenfassung und Ausblick

- ⇒ XML trennt Struktur/Inhalt und visuelle Präsentation und ermöglicht somit eine medienneutrale Datenhaltung.
- ⇒ XML ist besonders interessant bei Workflows zur parallelen Erstellung von gedruckten Publikationen und Online-Inhalten.
- ⇒ „Fertige“ XML-Dialekte finden Anwendung in unterschiedlichen Bereichen (Web/Print, Wissenschaft/Technik, E-Business, usw.).
- ⇒ Neuere Entwicklungen nutzen XML als Transportmedium von Datenpaketen: XML-RPC, Web Services, XCP (XML Control Protocol).
- ⇒ Die Nutzung von unterschiedlichen Technologien der XML-Familie innerhalb von konkreten Projekten schafft Synergie-Effekte.
- ⇒ Marktanteil von Software mit XML-Unterstützung nimmt zu.
- ⇒ **Fazit: XML ist ein universelles Konzept zur Informationsverarbeitung!**



# Nützliche Online-Ressourcen

- ⇒ **XML 1.0:** <http://www.edition-w3c.de/TR/2000/REC-xml-20001006/>
- ⇒ **XML 1.1:** <http://www.edition-w3c.de/TR/2004/REC-xml11-20040204/>
- ⇒ **XML Schema:** <http://www.edition-w3c.de/TR/2001/REC-xmlschema-0-20010502/>
- ⇒ **Namensräume:**  
<http://www.schumacher-netz.de/TR/1999/REC-xml-names-19990114-de.html>
- ⇒ **XSLT 1.0:** <http://xml.klute-thiemann.de/w3c-de/REC-xslt-20020318/>
- ⇒ **XPath 1.0:** <http://www.obqo.de/w3c-trans/xpath-de>
- ⇒ **MathML:** <http://www.w3.org/Math/>
- ⇒ **HTML 4.01:** <http://www.edition-w3.de/TR/1999/REC-html401-19991224/>
- ⇒ **XHTML 1.0:** <http://www.w3.org/TR/xhtml1/>
- ⇒ **SVG:** <http://www.w3.org/Graphics/SVG/>
- ⇒ **SVG - Learning By Coding:** <http://svglbc.datenverdrahten.de/>
- ⇒ **Web Services (Link-Sammlung):**  
<http://www.et.fh-merseburg.de/person/meinike/PHP/webservices.htm>

