XML and Semantic Web Technologies

Prof. Dr. Dr. Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute of Economics and Information Systems
& Institute of Computer Science
University of Hildesheim
http://www.ismll.uni-hildesheim.de

XML is . . .

- . . . the extensible markup language.
- . . . facilitates the separation of content from presentation.
- . . . (from a perspective of HTML) allowing the definition of own tags.
- . . . (from a perspective of SGML) a subset of SGML.
- . . . a W3C recommendation since 1998.

XML allows the definition of own tags

```
1 <html>
2  <head><link rel="stylesheet" href="books.css"
3             type="text/css"></head>
4  <body>
5    <p>
6      <span class="author">Shelly Powers</span>:
7      <span class="title">Pratical RDF</span>, 2003.
8    <p>
9      <span class="author">Eric T. Ray</span>:
10     <span class="title">Learning XML</span>, 2003.
11   </body>
12 </html>
```

Figure 1: Markup of elements in HTML with class attribute.

```
1 <page>
2  <p>
3    <author>Shelly Powers</author>:
4    <title>Pratical RDF</title>, 2003.</p>
5  <p>
6    <author>Eric T. Ray</author>:
7    <title>>Learning XML</title>, 2003.</p>
8 </page>
```

Figure 2: Markup of elements in XML with custom tags.

```
1 .author { color: red; }
2 .title { font-style: italic; }
```

Figure 3: CSS stylesheet to render HTML document.

Figure 4: Rendered HTML document.

Markup

Markup is text that is added to the data of a document in order to convey information about it.

I would not sell Attila for $1,000,000, says John.

Figure 5: Sample document.

a) <sentence><subclause><subject>I </subject> <predicate>would not sell </predicate> <object>Attila</object> for $1,000,000</subclause>, <predicate>says</predicate> <subject>John</subject>.</sentence>

b) \person[ref="John"]{I} would not sell \dog{Attila} for $1,000,000, says \person{John}.

c) I would not sell Attila for $1,000,000, says John.

Figure 6: Different kinds of markup of a text:
a) markup of syntactic structures (XML syntax),
b) markup of entities (LaTeX syntax),
c) markup of rendering attributes (XML syntax).

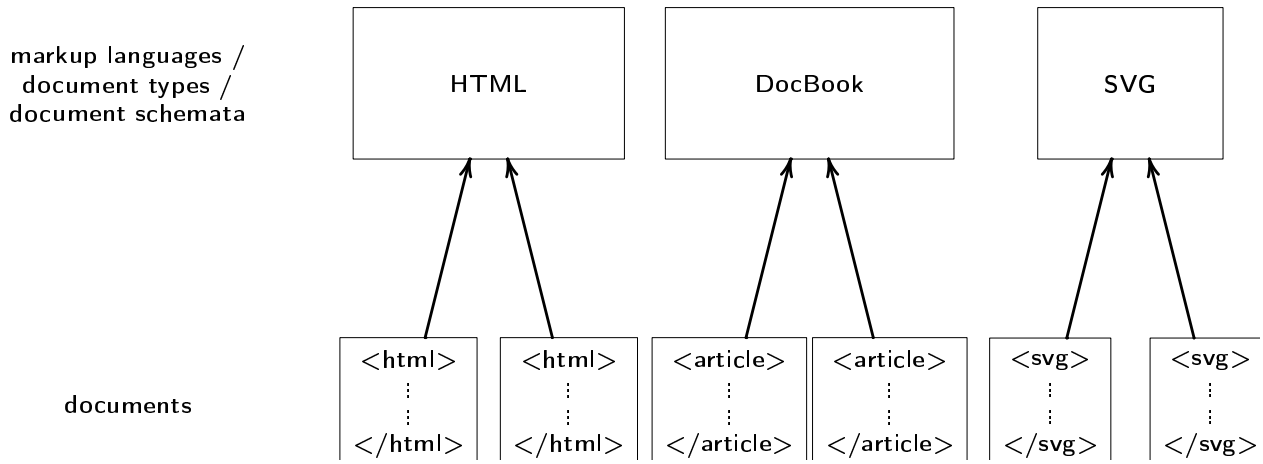

Figure 7: Documents can be described by markup languages.

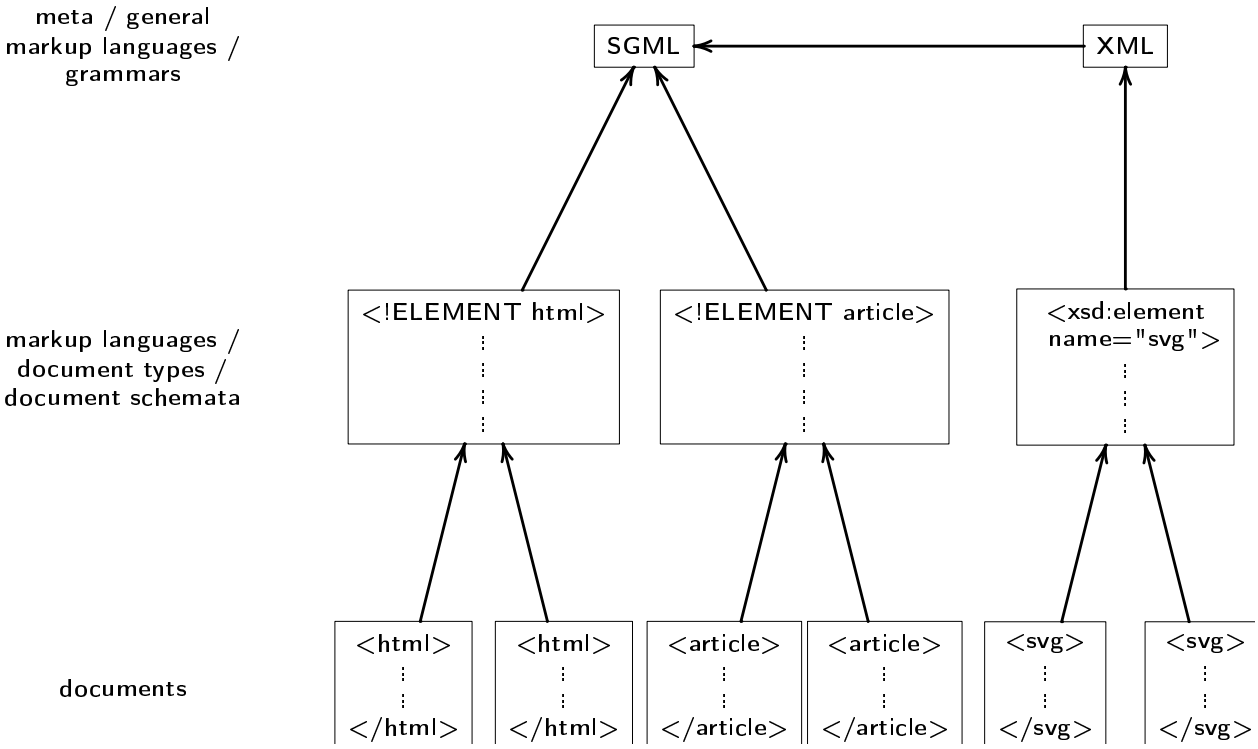

Figure 8: Markup languages can be described by meta markup languages.

From a Historical Perspective [Cou02]

1967 Tunnicliffe on **the separation of the information content of documents from their format** (invention of **generic coding**).

late '60s Rice on a universal catalog of parameterized 'editorial structure' tags.

late '60s GenCode Project (Scharpf, GCA).

1969 Generalized Markup Language (GML; Goldfarb, Mosher, Lorie; IBM).

1978 Foundation of a committee on Information Processing by ANSI.

1980 First draft, **1986** publication of SGML standard (ANSI/ISO).

1990 HTML 1 (Berners-Lee, CERN).

1994 Foundation of **World Wide Web Consortium (W3C)**.

1995 HTML 2, **1997** HTML 3.2 recommendation (W3C).

1996 First draft, **1998** publication of **XML** recommendation (W3C).

XML Applications: XHTML

```
<!DOCTYPE HTML PUBLIC
 "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
 <head>
  <title>Hello, world!</title>
 </head>
 <body>
  <h1>Hello, world!</h1>
  <p><a href=http://www.w3c.org>W3C</a>.
  <p>Another paragraph.
  <hr>
 </body>
</html>
```

Figure 9: HTML document.

```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC
 "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
 <head>
  <title>Hello, world!</title>
 </head>
 <body>
  <h1>Hello, world!</h1>
  <p><a href="http://www.w3c.org">W3C</a>.</p>
  <p>Another paragraph.</p>
  <hr/>
 </body>
</html>
```

Figure 10: Corresponding XHTML document.

XML Applications: SVG

```
<?xml version="1.0"?>
<svg xmlns="http://www.w3.org/2000/svg">
 <g style="fill-opacity:0.7; stroke:black; stroke-width:0.1cm;">
   <circle cx="6cm" cy="2cm" r="100" style="fill:red;"
            transform="translate(0,50)" />
   <circle cx="6cm" cy="2cm" r="100" style="fill:blue;"
            transform="translate(70,150)" />
   <circle cx="6cm" cy="2cm" r="100" style="fill:green;"
            transform="translate(-70,150)"/>
 </g>
</svg>
```

Figure 11: SVG sample document.

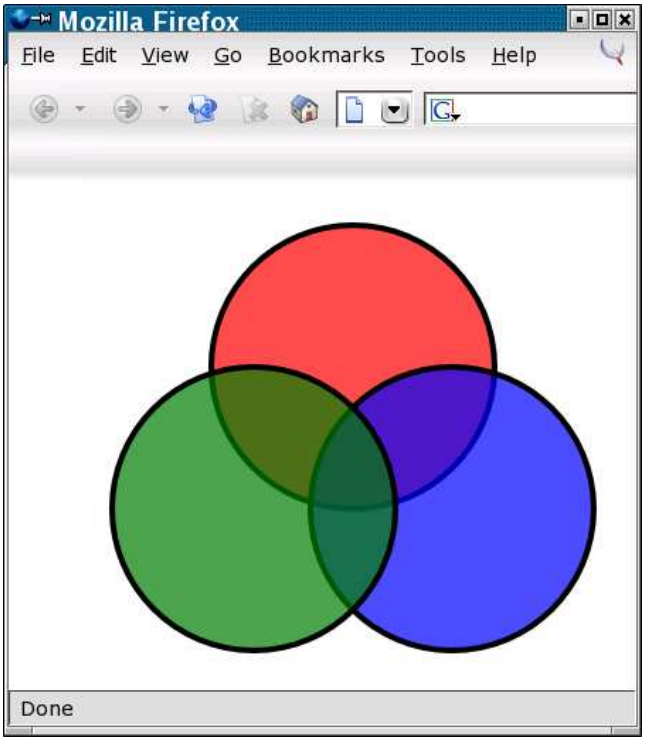

Figure 12: Rendered SVG sample document.

XML Applications: XML User-Interface Language (XUL)

XUL is implemented in Mozilla and allows to build user interfaces from descriptions in XML documents.

```
1 <?xml version="1.0"?>
2 <?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
3
4 <window id="example-window" title="xul-buttons"
5       xmlns:html="http://www.w3.org/1999/xhtml"
6       xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul">
7  <button label="Normal"/>
8  <button label="Disabled" disabled="true"/>
9  <button label="Default" default="true"/>
10 </window>
```

Figure 13: XUL sample document.

Figure 14: Rendered XUL sample document.

XML Applications: MusicXML

MusicXML is a industry standard for the markup of sheet music (v1.0, http://www.musicxml.org/)

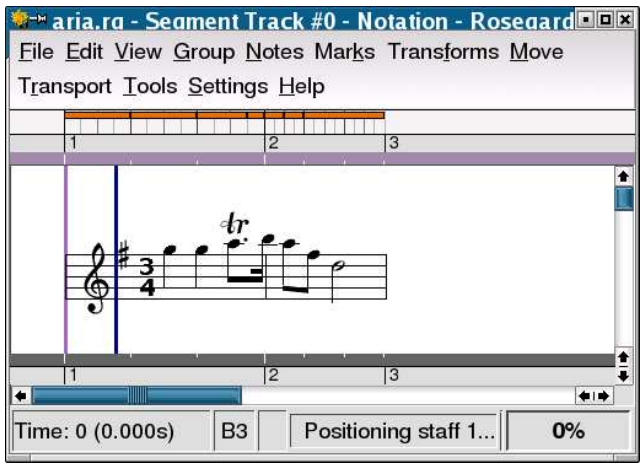

Figure 15: Rendering of the sample MusicXML document by Rosegarden.

```
1  <?xml version="1.0"?>
2  <!DOCTYPE score-partwise PUBLIC
3    "-//Recordare//DTD MusicXML 0.6 Partwise//EN"
4    "http://www.musicxml.org/dtds/partwise.dtd">
5  <score-partwise>[...]
6  <part id="0">
7    <measure number="1">
8    <attributes>
9      <divisions>960</divisions>
10     <time><beats>3</beats><beat-type>4</beat-type></time>
11     <clef><sign>G</sign><line>2</line></clef>
12     <key><fifths>1</fifths><mode>major</mode></key>
13   </attributes>
14 <note>
15   <pitch><step>G</step><octave>5</octave></pitch>
16   <duration>960</duration>
17   <voice>1</voice><type>quarter</type>
18 </note>
19 <note>
20   <pitch><step>G</step><octave>5</octave></pitch>
21   <duration>960</duration>
22   <voice>1</voice><type>quarter</type>
23 </note>[...]
24 </part>
25 </score-partwise>
```

Figure 16: Sample MusicXML document.

The Current Web

Resources:

- identified by URI's,
- untyped

Links:

- non-descriptive

Semantics has to be gleaned from content, e.g., context around a link anchor.

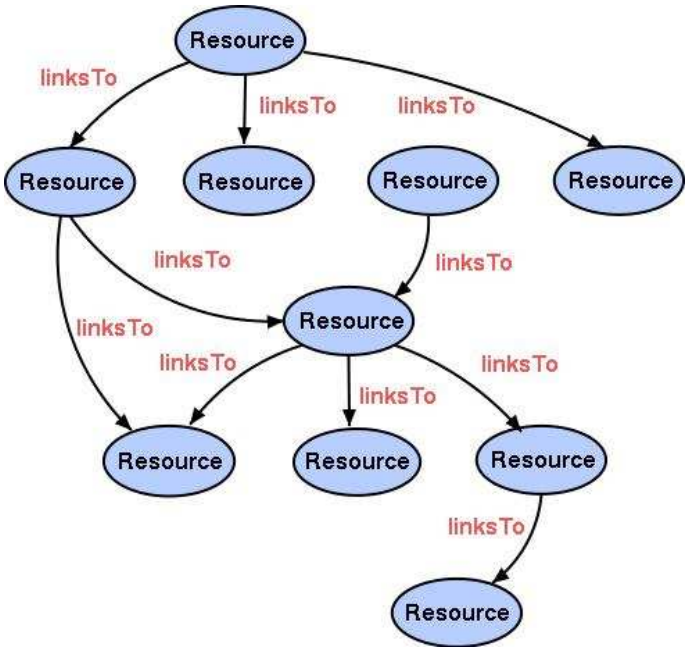

Figure 17: A sample of the current net [Mil04].

The Semantic Web

Resources:

- Globally Identified by URI's or Locally scoped (Blank)
- Extensible
- Relational

Links:

- Identified by URI's
- Extensible
- Relational

Semantics can be inferred from types of resources and links and known relations between resource / links of specific types.

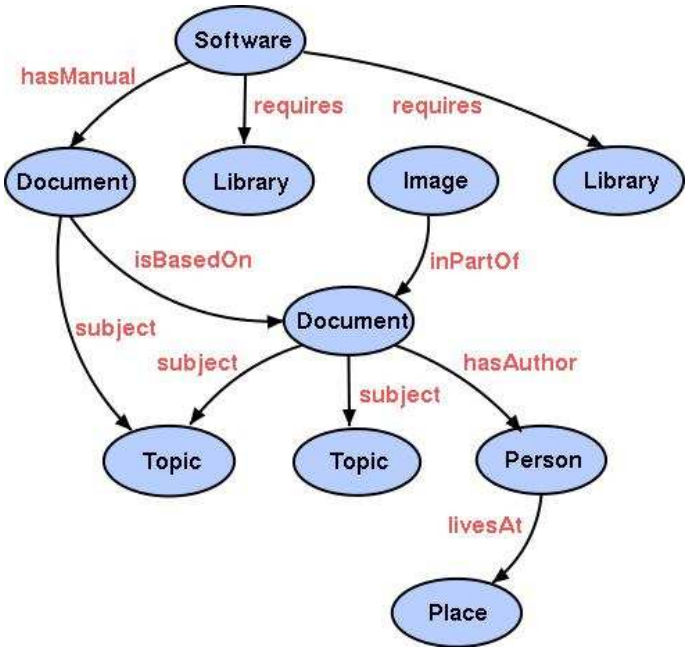

Figure 18: The same sample as semantic web [Mil04].

Semantic Web Applications

Figure 19: Looking for "'Gold Rush"' in Google.

Semantic Web Applications

Figure 20: "'Gold Rush"' on IMDB.

Semantic Web Applications

Semantic Web technologies typicaly are used for

- information retrieval
- information extraction
- information integration

You can think of Semantic Web as a hybridization of

- XML technologies (data representation) and
- logics (inference)

XML

- Syntax.
- Document Tree Model.

```
1 <?xml version="1.0"?>
2 <!DOCTYPE html PUBLIC
3   "-//W3C//DTD XHTML 1.0 Transitional//EN"
4   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
5 <html>
6   <head>
7     <title>Hello, world!</title>
8   </head>
9   <body>
10    <h1>Hello, world!</h1>
11    <p><a href="http://www.w3c.org">W3C</a>.</p>
12    <p>Another paragraph.</p>
13    <hr/>
14  </body>
15 </html>
```

Figure 21: XML sample file (here: XHTML).

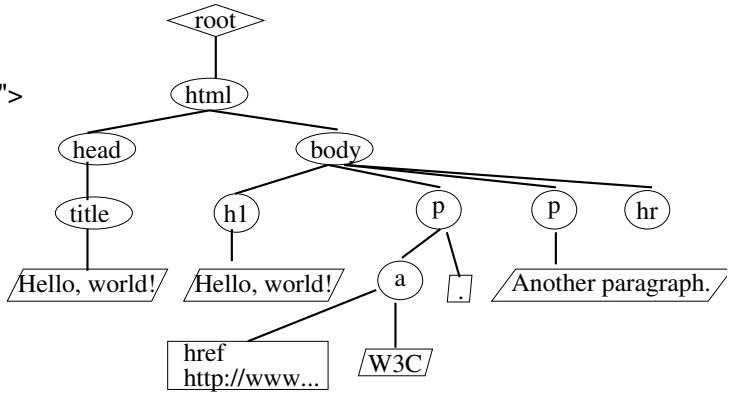

Figure 22: Corresponding document tree.

Document Type Definitions (DTDs)

DTDs describe the syntax of SGML or XML documents.

```
<!ELEMENT books (book*)>
<!ELEMENT book (author+, title, year)>
<!ELEMENT author (fn, sn)>
<!ELEMENT fn (#PCDATA)>
<!ELEMENT sn (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ATTLIST year
 edition CDATA #IMPLIED>
```

Figure 24: A simple DTD for a book list.

```
<?xml version="1.0"?>
<!DOCTYPE books SYSTEM "books.dtd">
<books>
 <book>
  <author><fn>Rainer</fn><sn>Eckstein</sn></author>
  <author><fn>Silke</fn><sn>Eckstein</sn></author>
  <title>XML und Datenmodellierung</title>
  <year>2004</year>
 </book>
 <book>
  <author><fn>Erik T.</fn><sn>Ray</sn></author>
  <title>Learning XML</title>
  <year edition="2">2003</year>
 </book>
</books>
```

Figure 23: A valid books-document.

```
<?xml version="1.0"?>
<!DOCTYPE books SYSTEM "books.dtd">
<books>
 <book lang="english">
  <author><sn>Ray</sn><fn>Erik T.</fn></author>
  <title>Learning XML</title>
  <publisher>O'Reilly</publisher>
 </book>
</books>
```

Figure 25: An invalid books-document.

XML Schema

XML Schema is a more powerful, XML specific alternative for specifying the syntax of XML documents, that replaces DTDs.

```
<?xml version="1.0"?>
<!DOCTYPE books SYSTEM "books.dtd">
<books>
 <book>
  <author><fn>Rainer</fn><sn>Eckstein</sn></author>
  <author><fn>Silke</fn><sn>Eckstein</sn></author>
  <title>XML und Datenmodellierung</title>
  <year>2004</year>
 </book>
 <book>
  <author><fn>Erik T.</fn><sn>Ray</sn></author>
  <title>Learning XML</title>
  <year edition="2">2003</year>
 </book>
</books>
```

Figure 26:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSche
<xsd:element name="books">
 <xsd:complexType><xsd:sequence>
  <xsd:element name="book" maxOccurs="unbounded">
   <xsd:complexType><xsd:sequence>
    <xsd:element name="author" maxOccurs="unbounded"
     <xsd:complexType><xsd:sequence>
      <xsd:element name="fn"/>
      <xsd:element name="sn"/>
     </xsd:sequence></xsd:complexType>
    </xsd:element>
   <xsd:element name="title"/>
   <xsd:element name="year">
    <xsd:complexType><xsd:simpleContent>
     <xsd:extension base="xsd:string">
      <xsd:attribute name="edition" use="optional"
                     type="xsd:integer"/>
     </xsd:extension>
    </xsd:simpleContent></xsd:complexType>
   </xsd:element>
  </xsd:sequence></xsd:complexType>
 </xsd:element>
 </xsd:sequence></xsd:complexType>
</xsd:element>
</xsd:schema>
```

Figure 27:

XML Linking

- XML namespaces,
- XPath,
- XLink,
- XPointer

//p/text()

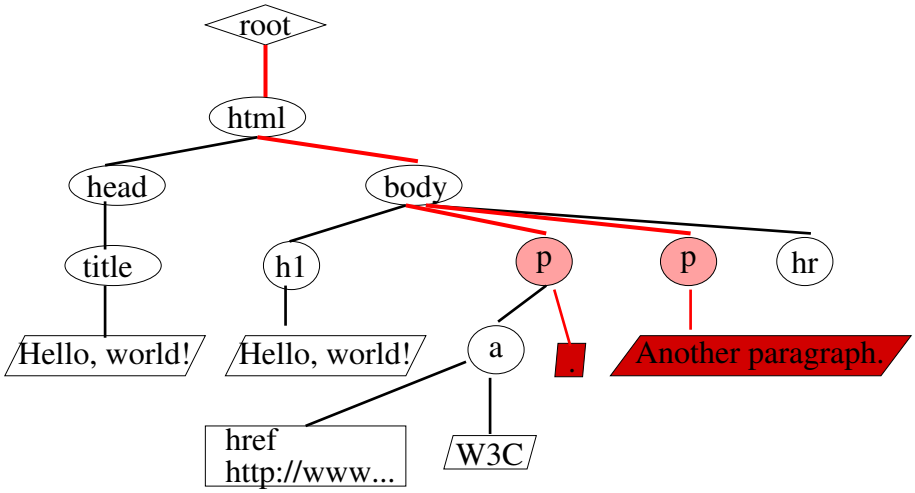

Figure 28: Selected nodes by XPath expression `//p/text()`.

XML Stylesheet Language (XSLT)

XSLT stylesheets are used to transform XML documents into another (XML) representation.

Most frequent application is transformation to HTML (rendering).

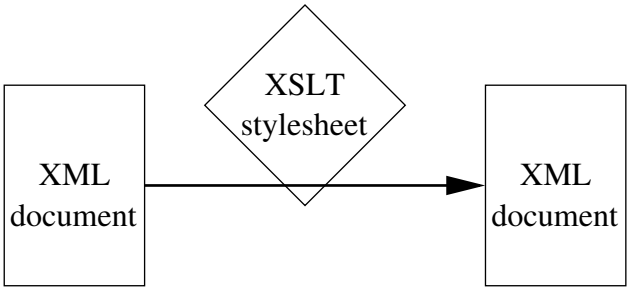

Figure 29: XSLT stylesheets transform XML documents.

XML Stylesheet Language (XSLT)

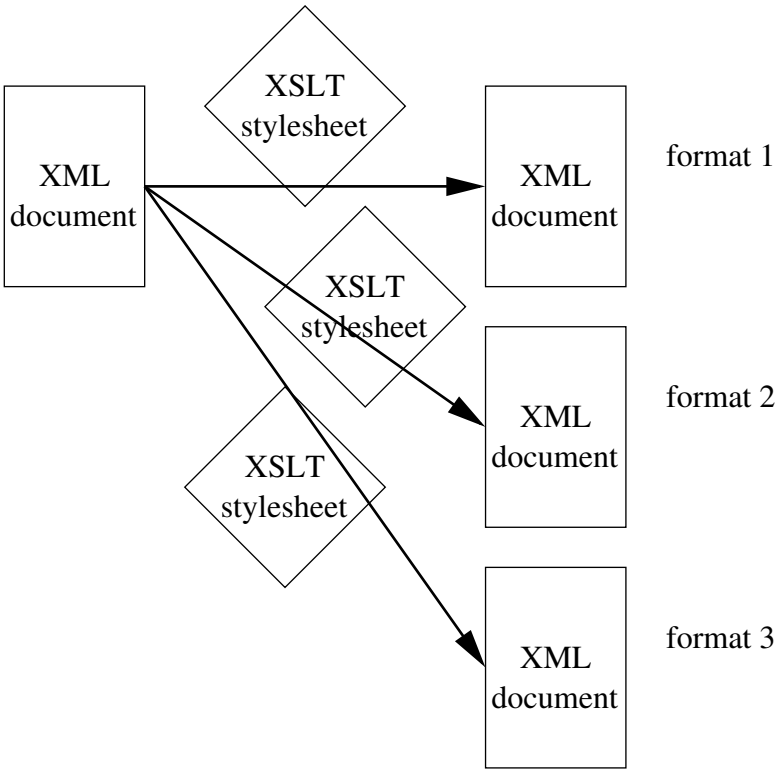

Figure 30: Different XSLT stylesheets transform to different target XML documents.

XML Stylesheet Language (XSLT)

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                version="1.0">
 <xsl:output method="html"/>

 <xsl:template match="/">
   <html><body bgcolor="#FFFFFF"><ol>
     <xsl:apply-templates/>
     </ol></body></html>
 </xsl:template>

 <xsl:template match="book">
   <li>
     <xsl:apply-templates select="author[1]/fn"/><xsl:text> </xsl:text>
     <xsl:apply-templates select="author[1]/sn"/>
     <xsl:for-each select="author[position()>1]">
       <xsl:text>, </xsl:text>
       <xsl:apply-templates select="fn"/><xsl:text> </xsl:text>
       <xsl:apply-templates select="sn"/>
     </xsl:for-each>
     <xsl:text>: </xsl:text>
     <em><xsl:apply-templates select="title"/></em><xsl:text>, </xsl:text>
     <xsl:apply-templates select="year"/><xsl:text>.</xsl:text>
   </li>
 </xsl:template>
</xsl:stylesheet>
```

Figure 31: A sample XSLT stylesheet for rendering books-documents.

XML Query Language (XQuery)

XQuery is a SQL-like query language for XML documents.

```
<?xml version="1.0"?>
<!DOCTYPE books SYSTEM "books.dtd">
<books>
  <book>
    <author><fn>Erik T.</fn><sn>Ray</sn></author>
    <title>Learning XML</title>
    <year edition="2">2003</year>
  </book>
  <book>
    <author><fn>Rainer</fn><sn>Eckstein</sn></author>
    <author><fn>Silke</fn><sn>Eckstein</sn></author>
    <title>XML und Datenmodellierung</title>
    <year>2004</year>
  </book>
  <book>
    <author><fn>Martin</fn><sn>Schader</sn></author>
    <author><fn>Lars</fn><sn>Schmidt-Thieme</sn></author>
    <title>Java</title>
    <year>2003</year>
  </book>
</books>
```

A sample books document.

```
xquery version "1.0";

declare namespace f=
  "http://cgnm.informatik.uni-freiburg.de/xml";

declare function f:authors-sortstring($author) as xs:string {
  string-join(
    for $a in $author
      return concat($a/sn, " ", $a/fn),
    " ")
};

<books> {
  for $t in //*/title
    let $r := $t/..
    where contains($t, 'XML')
    order by f:authors-sortstring($r/author), $r/year
    return $r
} </books>
```

Querying a books-document by XQuery.

Resource Description Framework (RDF)

RDF allows the "description of resources" via triples (subject, object, predicate).

RDF has a graphical and a XML representation.

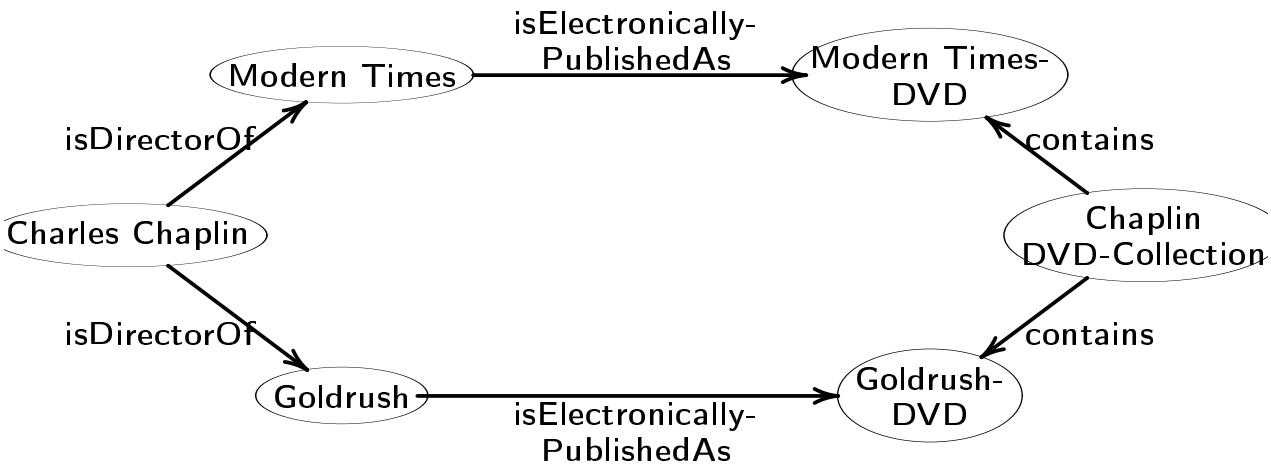

Figure 32: Sample semantic network.

RDF Schema

RDF Schema has specific constructs for expressing classes and properties.

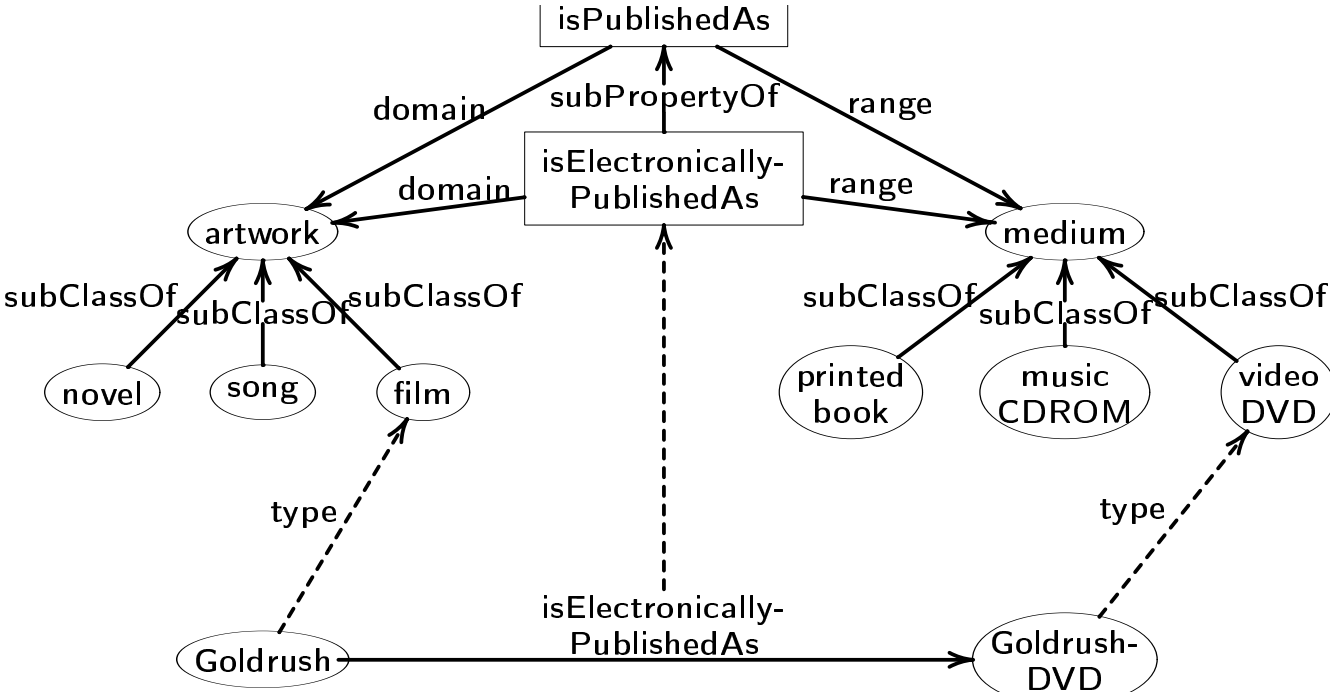

Figure 33: RDF Schema description of classes and properties in the sample semantic network.

Web Ontology Language (OWL)

OWL adds more expressive modelling constructs, e.g., to express, that the range of a given predicate depends on the subject.

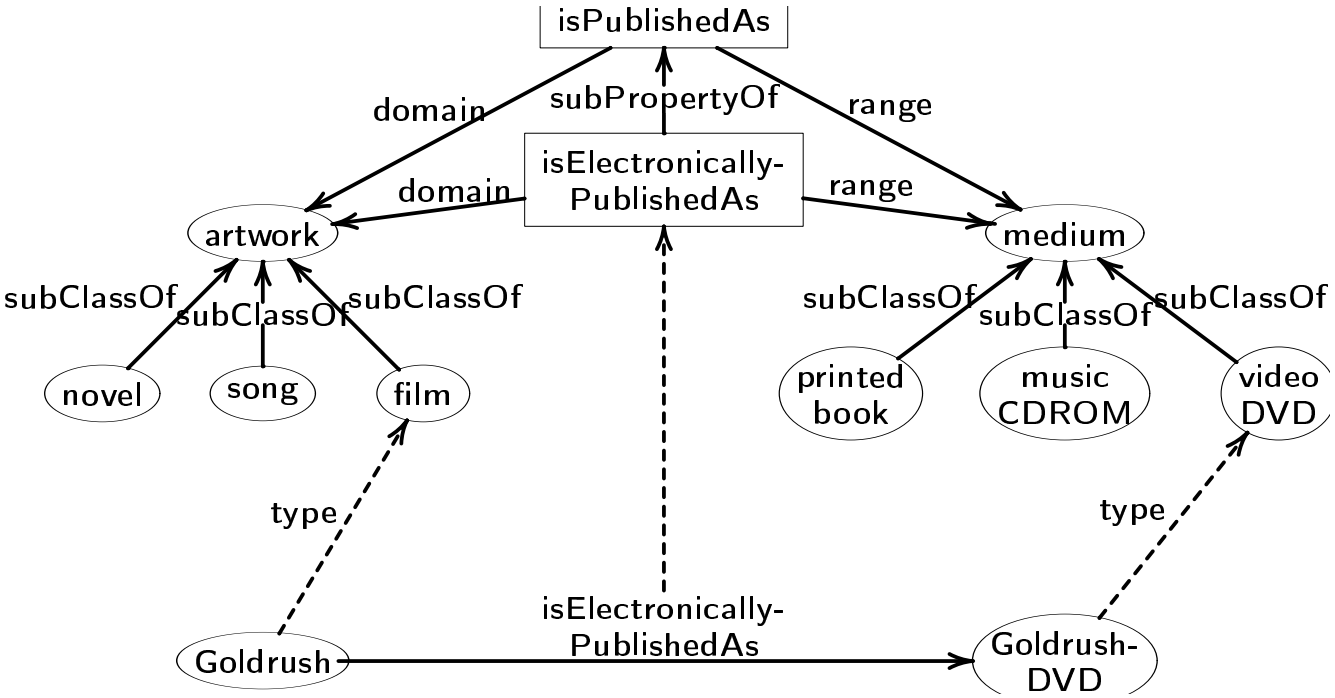

Figure 34: With RDF Schema one cannot model, that films always are electronically published on Video-DVDs.

Semantic Web Layer Cake

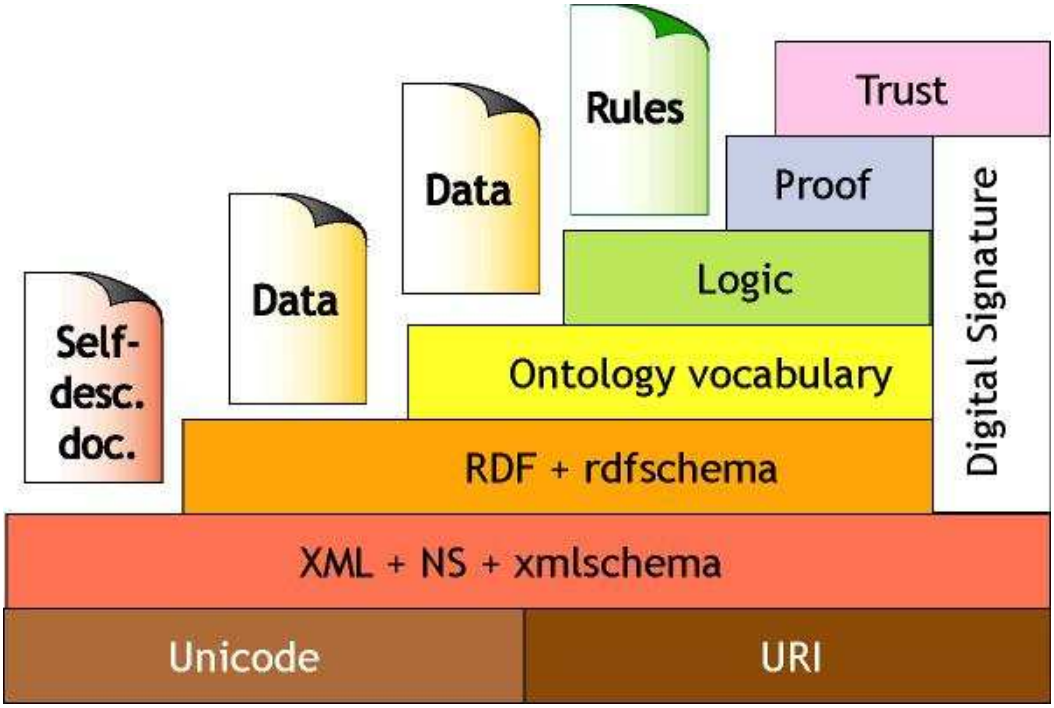

Figure 35: Semantic Web Layers (Berners-Lee).

What the course eventually will cover

If we have time:

- RDF rule and query languages
- RDF inferencing
- More practical examples
- . . .

What the course (probably) will not cover

- XML and Databases
- APIs for programming with XML as DOM, SAX, etc.
- Extensive descriptions of complex XML applications (e.g., XML based markup languages) as SVG, XForms, etc.
- Detailed instructions for the usage of tools.
- "Process models" and best practices.

But upcoming winter term we will offer

Praktikum on XML and Semantic Web Technologies

Some books

- Rainer Eckstein and Silke Eckstein. *XML und Datenmodellierung*. dpunkt.verlag, 2003
- Charles F. Goldfarb and Paul Prescod. *XML Handbook*. Prentice Hall PTR, 5th edition, 2003
- Eric T. Ray. *Learning XML*. O'Reilly, 2003
- Howard Katz, editor. *XQuery from the experts: a guide to the W3C XML query language*. Addison-Wesley, Boston, 2004
- Shelly Powers. *Practical RDF*. O'Reilly, 2002
- Grigoris Antoniou and Frank Van Harmelen. *A Semantic Web Primer*. MIT Press, 2004
- W3C recommendations at http://www.w3.org.

Some First XML Software

- XML Processors / Parsers:
  - Apache Xerxes (http://xml.apache.org/xerces2-j/index.html). v2.9.0: XML 1.1; Namespaces 1.1, XML Schema 1.0.
- XSLT Processor:
  - Apache Xalan (http://xml.apache.org/xalan-j/index.html). v2.7.0: XSLT 1.0, XPath 1.0.
- XQuery & XSLT Processor:
  - Saxon (http://saxon.sourceforge.net; Michael H. Kay). v8.9: XSLT 2.0, XPath 2.0; XQuery 1.0.

Exercises and tutorials

- There will be a weekly sheet with two exercises handed out **each Thursday** in the lecture.
  1st sheet will be handed out this Thur. 12.4.
- Solutions to the exercises can be submitted until **every next Wednesday 1 pm** in the letter box
  1st sheet is due Wed. 18.4. 1 pm.
- Exercises will be corrected by your tutor.
- Tutorials **each Thursday 11-12** immediately after the lecture, 1st tutorial at Thur. 19.4.

Exam and credit points

- There will be an exam at end of term (2h, 4 problems).
- You can get up to 10% of the points as bonus points from the tutorial.
- The course gives 7 credit points.
- The course can be used in IMIT-Module BW2 Business Intelligence.

References

[AH04] Grigoris Antoniou and Frank Van Harmelen. *A Semantic Web Primer*. MIT Press, 2004.

[Cou02] Didier Courtaud. From gencode to xml : an history of markup languages, 2002.

[EE03] Rainer Eckstein and Silke Eckstein. *XML und Datenmodellierung*. dpunkt.verlag, 2003.

[GP03] Charles F. Goldfarb and Paul Prescod. *XML Handbook*. Prentice Hall PTR, 5th edition, 2003.

[Kat04] Howard Katz, editor. *XQuery from the experts: a guide to the W3C XML query language*. Addison-Wesley, Boston, 2004.

[Mil04] Eric Miller. Weaving meaning: An overview of the semantic web, 2004.

[Pow02] Shelly Powers. *Practical RDF*. O'Reilly, 2002.

[Ray03] Eric T. Ray. *Learning XML*. O'Reilly, 2003.