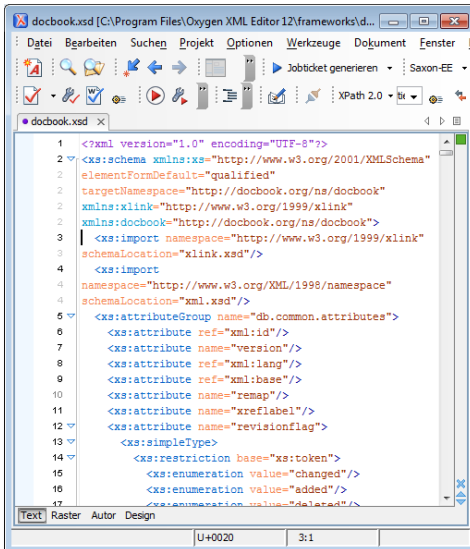


XML-Dokumente besser prüfen mit ISO-Schematron

Prof. Dr. Marko Hedler
Hochschule der Medien, Stuttgart

Grammatikbasierte Sprachen

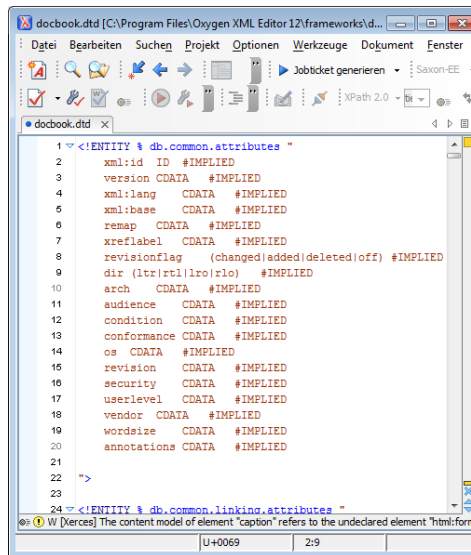
■ XML-Schema 1.0



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
3   elementFormDefault="qualified"
4   targetNamespace="http://docbook.org/ns/docbook"
5   xmlns:xlink="http://www.w3.org/1999/xlink"
6   xmlns:docbook="http://docbook.org/ns/docbook"
7   <xs:import namespace="http://www.w3.org/1999/xlink"
8     schemaLocation="xlink.xsd"/>
9   <xs:import
10     namespace="http://www.w3.org/XML/1998/namespace"
11     schemaLocation="xml.xsd"/>
12   <xs:attributeGroup name="db.common.attributes">
13     <xs:attribute ref="xml:id"/>
14     <xs:attribute name="version"/>
15     <xs:attribute ref="xml:lang"/>
16     <xs:attribute ref="xml:base"/>
17     <xs:attribute name="remap"/>
18     <xs:attribute name="xreflabel"/>
19     <xs:attribute name="revisionflag"/>
20     <xs:simpleType>
21       <xs:restriction base="xs:token">
22         <xs:enumeration value="changed"/>
23         <xs:enumeration value="added"/>
24         <xs:enumeration value="deleted"/>
25       </xs:restriction>
26     </xs:simpleType>
27   </xs:attributeGroup>
28 </xs:schema>
  
```

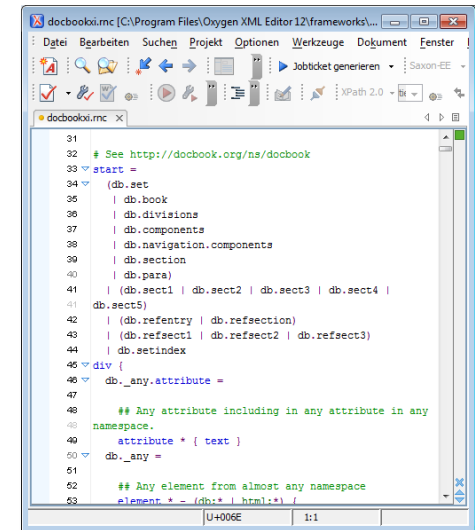
■ DTD



```

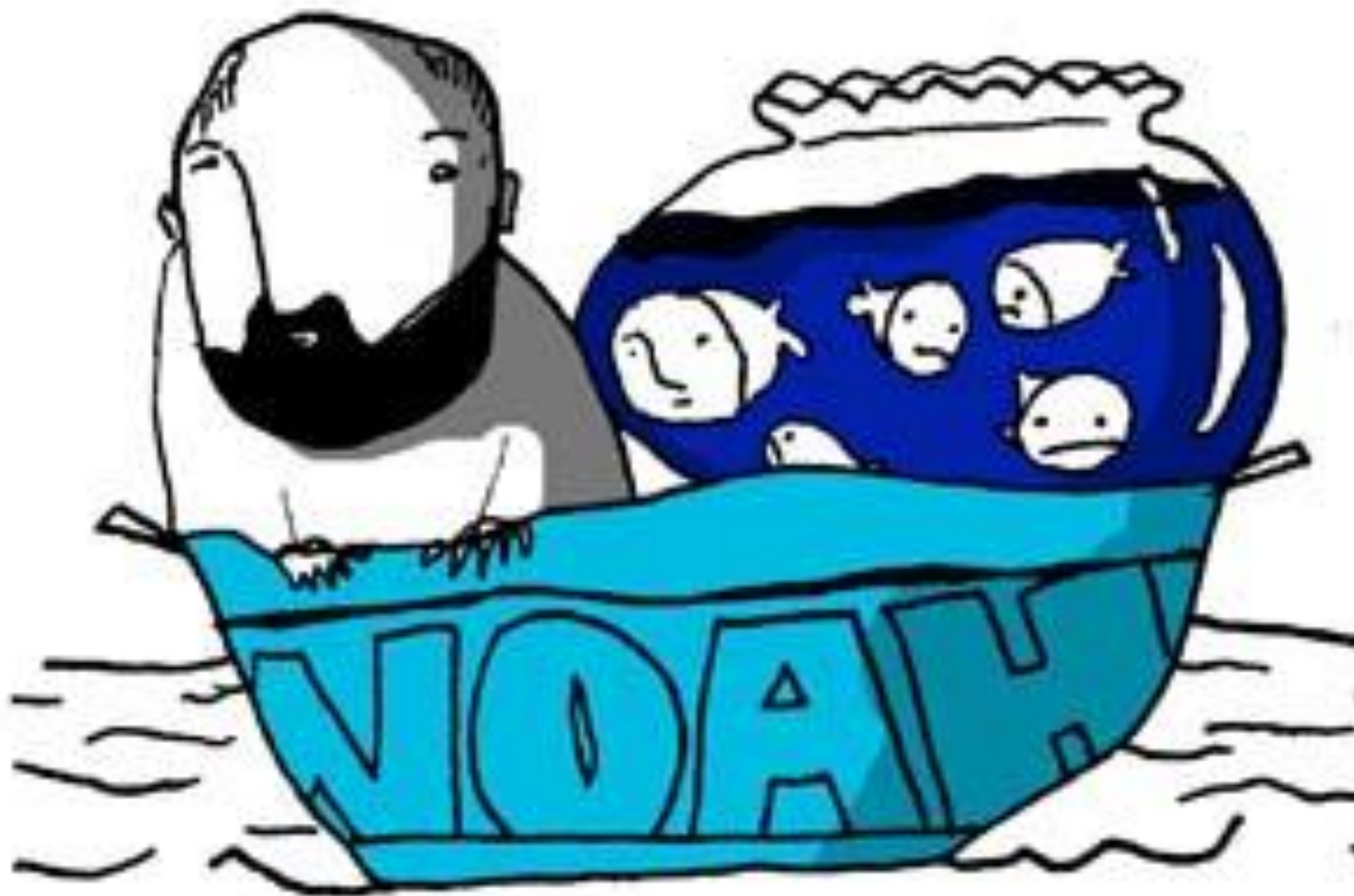
1 <!ENTITY % db.common.attributes "
2   xml:id ID #IMPLIED
3   version CDATA #IMPLIED
4   xml:lang CDATA #IMPLIED
5   xml:base CDATA #IMPLIED
6   remap CDATA #IMPLIED
7   xreflabel CDATA #IMPLIED
8   revisionflag (changed|added|deleted|off) #IMPLIED
9   dir (ltr|rtl|lro|lrl) #IMPLIED
10  arch CDATA #IMPLIED
11  audience CDATA #IMPLIED
12  condition CDATA #IMPLIED
13  conformance CDATA #IMPLIED
14  os CDATA #IMPLIED
15  revision CDATA #IMPLIED
16  security CDATA #IMPLIED
17  userlevel CDATA #IMPLIED
18  vendor CDATA #IMPLIED
19  wordsize CDATA #IMPLIED
20  annotations CDATA #IMPLIED
21 >"
22
23
24 <!ENTITY % db.common.links "
  
```

■ RelaxNG



```

31
32 # See http://docbook.org/ns/docbook
33 start =
34 (db.set
35 | db.book
36 | db.divisions
37 | db.components
38 | db.navigation.components
39 | db.section
40 | db.para
41 | (db.sect1 | db.sect2 | db.sect3 | db.sect4 |
42 | db.sect5)
43 | (db.refentry | db.refsection)
44 | (db.refsect1 | db.refsect2 | db.refsect3)
45 | db.setindex
46 )
47
48 div {
49   db._any.attribute =
50   ## Any attribute including in any attribute in any
51   namespace.
52   attribute * { text }
53   db._any =
54   ## Any element from almost any namespace
55   element * - (db.* | html:*) /
  
```

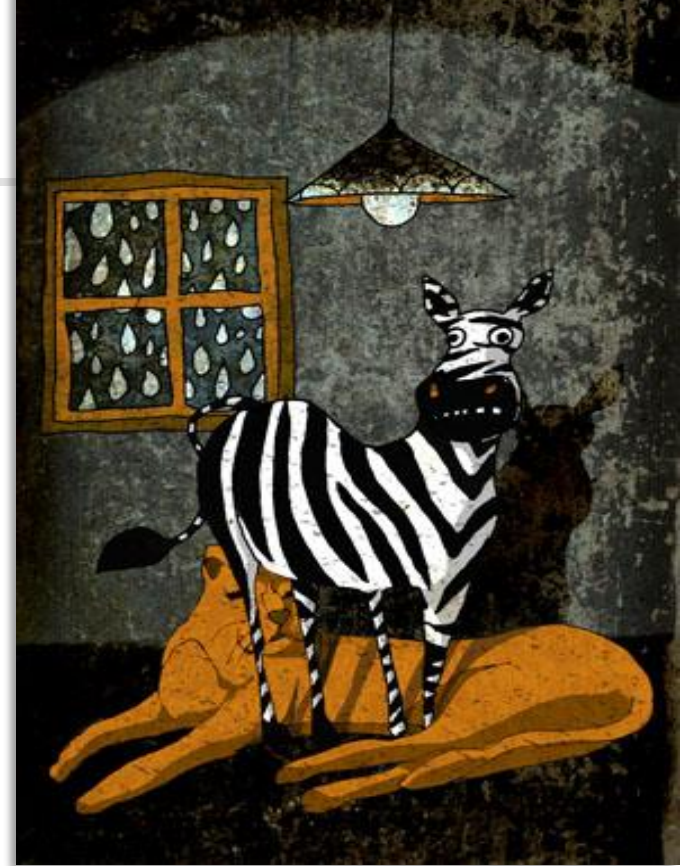




Die Arche Noah

```
<arche>
  <zimmer>
    <tier fleischfresser="nein">
      <art>Zebra</art>
    </tier>
    <tier fleischfresser="ja">
      <art>Gepard</art>
    </tier>
  </zimmer>
</arche>
```

XML



```
<xs:element name="zimmer">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="2"
        maxOccurs="unbounded" ref="tier"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

XSD 1.0

Bedingung ist abbildbar durch
XPath!

Formulierung von Bedingungen in XPATH

Arche-Noah-Business-Rule:

Gibt es Fleischfresser und Vegetarier in einem Zimmer?

```
tier[@fleischfresser='ja'] and  
tier[@fleischfresser='nein']
```

Möglichkeiten von XPATH:


- Zugriff auf alle Knoten des XML-Dokuments
- Verschiedene Knotenoperationen
- Einsatz der XPATH-Funktionsbibliothek

XPATH in Schematron einbetten!

Was ist Schematron?

- Teil des DSDL (ISO 19757)
- Hosting Sprache in XML formuliert

```

<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://purl.oclc.org/dsdl/schematron"
  queryBinding="[query language]


```

XPATH in Schematron einbetten

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://purl.oclc.org/dsdl/schematron"
  queryBinding="xslt2">
  <pattern>
    <rule context="zimmer">
      <assert test="not(
        tier[@fleischfresser='ja'] and
        tier[@fleischfresser='nein']
      )"/>
    </pattern>
  </schema>
```

Schematron-Validierungsfehler

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <?oxygen SCHSchema="arche.sch"?>
3  <arche xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:noNamespaceSchemaLocation="arche.xsd">
5     <zimmer>
6         <tier fleischfresser="nein">
7             <art>Zebra</art>
8         </tier>
9         <tier fleischfresser="ja">
10            <art>Gepard</art>
11        </tier>
12    </zimmer>
13 </arche>
  
```

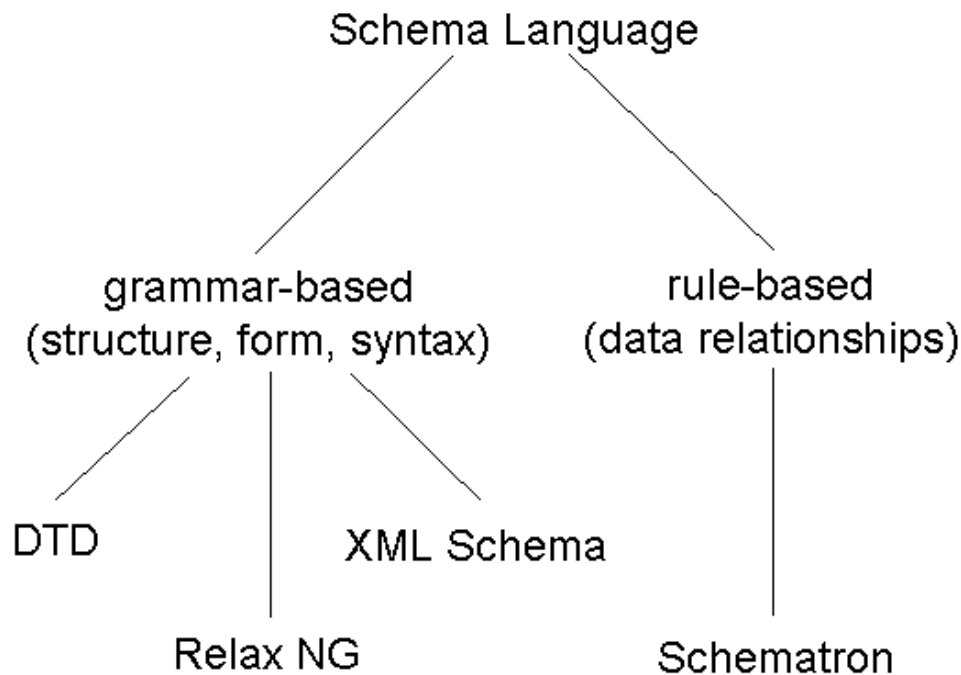
Allgemeine Prüfmöglichkeiten mit Schematron

Schematron prüft nicht aufgrund einer Grammatik, sondern aufgrund von Regeln (Regelbasierte Sprache).

Roger Costello teilt die Regeln dabei in drei Bereiche ein:

- Co-constraints
 - Regeln aufgrund der Abhängigkeit von Inhalten
 - Bsp: Wenn das Attribut X den Wert Y hat, dann....
- Cardinality checking
 - Prüfung aufgrund des Auftretens von Daten, Elementen, Attributen,...
 - Bsp: Wenn eine Processing-Instruction vorkommt, dann...
- Algorithmic checking
 - Prüfung aufgrund von Berechnungen
 - Bsp: Der Rechnungsbetrag muss der Summe aller Einzelbeträge entsprechen.

Schematron: Ein Ersatz für XML Schema?



-> Schematron als Ergänzung zu grammatikbasierten Sprachen

Grafik von Roger Costello

Entnommen aus <http://www.xfront.com/schematron/Two-types-of-XML-Schema-Language.html>

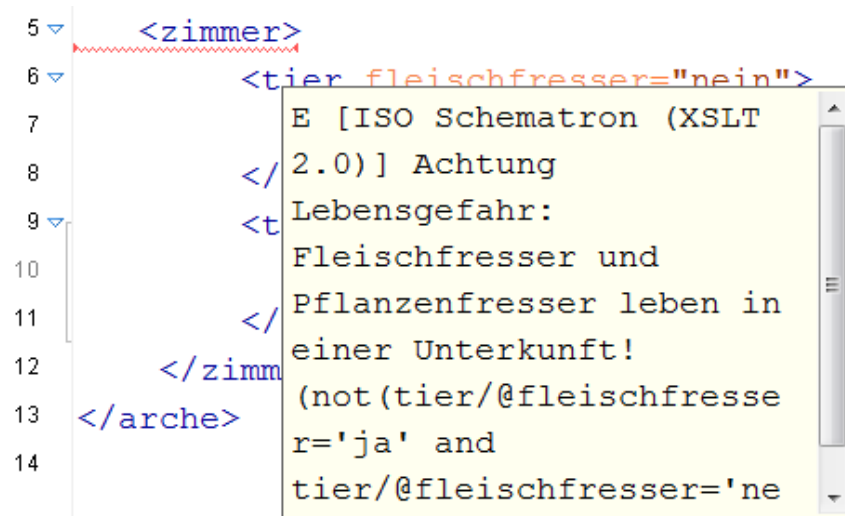
Was kann Schematron noch?

Benutzerdefinierte Fehlermeldungen

```
<assert test="[pattern]">
```

**Achtung Lebensgefahr: Fleischfresser und
Pflanzenfresser leben in einer Unterkunft!**

```
</assert>
```



```

5 <zimmer>
6   <tier fleischfresser="nein">
7     E [ISO Schematron (XSLT
8     2.0)] Achtung
9     Lebensgefahr:
10    Fleischfresser und
11    Pflanzenfresser leben in
12    einer Unterkunft!
13    (not (tier/@fleischfresse
14    r='ja' and
    tier/@fleischfresser='ne
  
```

Fehlerkategorien (Rollen)

```
<assert test="[pattern]" role="warning">  
    Alle Tiere sind Fleischfresser!  
</assert>
```

Zugriff auf externe Dokumente

```
<assert test="art = doc('tierarten.xml')//art" >  
    Hier gibt es blinde Passagiere!  
</assert>
```


Phasen

ID

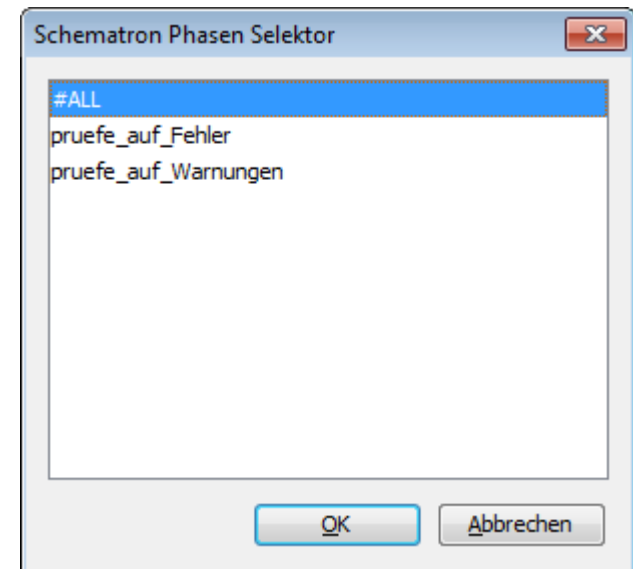
```
<phase id="pruefe_auf_Fehler">
  <active pattern="p1"></active>
</phase>
```

```
<phase id="pruefe_auf_Warnungen">
  <active pattern="p2"></active>
</phase>
```

IDREF

```
<pattern id="p1">
  ...
</pattern>
```

```
<pattern id="p2">
  ...
</pattern>
```



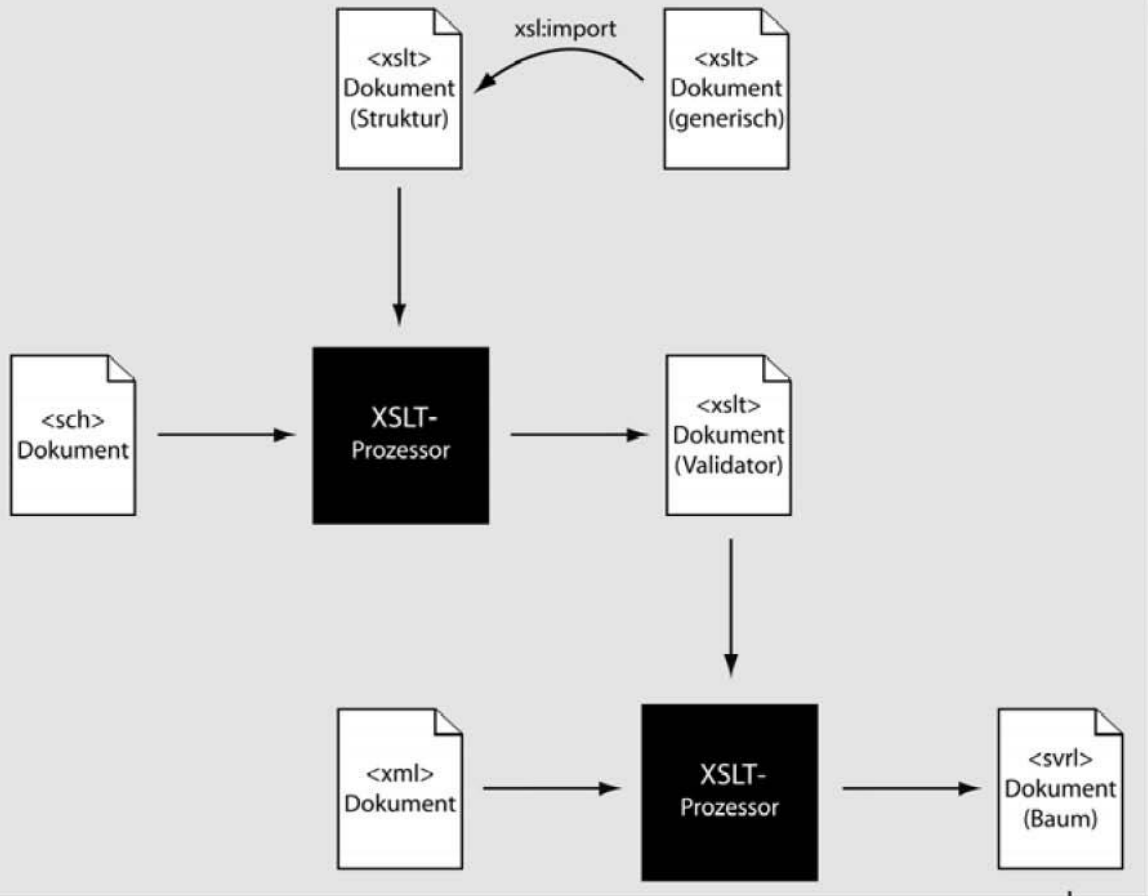
Wie nutzt man Schematron?

Wie nutzt man Schematron

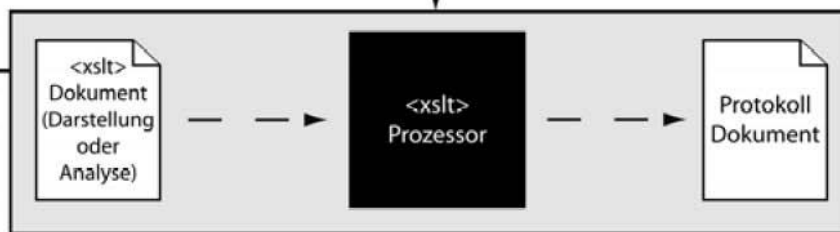
- Szenario 1: Im Editor

Editoren die Schematron unterstützen:
Oxygen, XMLMind,

- Szenario 2: Black-Box (Referenzimplementierung mit XSLT)
- Szenario 3: Black-Box eingebettet in XProc



Sequenz von weiteren
Transformationen (skalierbar)



Beispiele aus dem Publishing!

Fußnoten

XML

```

<para>
In order to view this text in a browser, you have to start the
browser.
  <footnote>
    <para>browser: Tool to view (X)HTML.
      <footnote>
        <para>(X)HTML: XML Hypertext Markup Language</para>
      </footnote>
    </para>
  </footnote>
</para>

```

SCH

```

<rule context="footnote">
  <assert test="not(../footnote)">
    ...
  </assert>
</rule>

```

PI's

SCH

```
<pattern id="PIs">  
  <rule context="processing-instruction(">  
    <assert test="starts-with(name(), 'hdm') ">  
      Diese PI stammt nicht aus der HdM.  
      Bitte dokumentieren Sie Ihre Funktion.  
      Name der PI: <value-of select="name()" />  
    </assert>  
  </rule>  
</pattern>
```

Verschachtelungstiefe von Listeneinträgen

SCH

```

<pattern id="Listen">
  <rule context="listitem">
    report test="count(ancestor::listitem) > 2">
      Eine Liste darf nicht mehr als drei Ebenen haben.
      Wir sind in der
      <value-of select="count(ancestor::listitem)+1"/>
      . Ebene.
    </report>
  </rule>
</pattern>
  
```


Verwendung von Zeichen

Beispiel:

```
<para>Bitte tragen Sie hier ihren
  Namen ein: _____</para>
```

```
<pattern id="Zeichen" role="warning">
  <rule context="*[text()]">
    <report test="contains(text(),'____')">
      Mehr als drei Unterstriche hintereinander.
    </report>
  </rule>
</pattern>
```

SCH

Lang-Attribut bei XHTML

```
<!DOCTYPE html SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
      lang="it" xml:lang="de">

...
</html>
```

SCH

```
<rule context="html:html">
  <assert test="@lang"> Das Attribut "lang" fehlt. </assert>
  <assert test="@xml:lang"> Das Attribut "xml:lang" fehlt.</assert>
  <assert test="@lang = @xml:lang">
    Die Werte der Attribute "lang" und "xml:lang" sollten
    identisch sein.</assert>
</rule>
```

Schematron bei großen Grammatiken:

Wer benutzt Schematron?

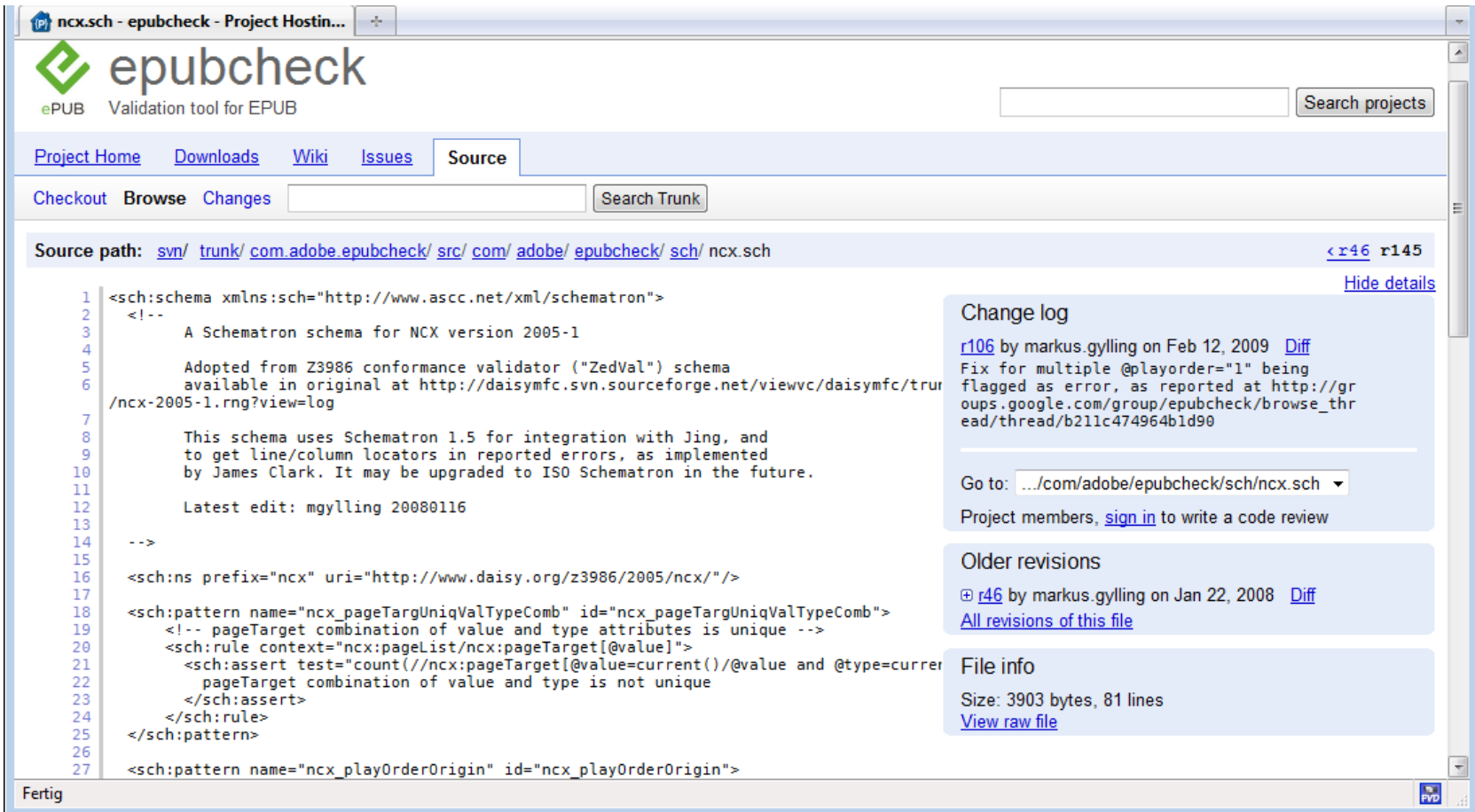
- DocBook
- DITA
- TEI
- EPUB
- ...?

Schematron in DocBook



The screenshot shows a web browser window with the title "DocBook Release". The main content area has a teal header with a duck logo and the text "DocBook.org" and "The Source for Documentation". Below the header is a navigation menu with buttons for "Home", "OASIS TC", "Documentation", "Schemas", "Modules", "Namespaces", "Help!", and "Wiki". A secondary navigation bar contains links for "Overview", "V5.x", "V4.x", "Archives", and "DocBook V5.0 Schematron Rules". The main heading is "DocBook V5.0 Schematron Rules". Below this, it says "There is a [specification](#) for this release." followed by a bulleted list of links: "Parent Directory" and "docbook.sch". At the bottom of the content area, it says "Apache/2 Server at www.docbook.org Port 80". The browser's search bar at the bottom contains the text "schematron" and the word "Fertig" is visible on the right.

Schematron im EPUB-Check



ncx.sch - epubcheck - Project Hostin...

epubcheck
ePUB Validation tool for EPUB

Project Home Downloads Wiki Issues **Source**

Checkout Browse Changes Search Trunk

Source path: [svn/](#) [trunk/](#) [com.adobe.epubcheck/](#) [src/](#) [com/](#) [adobe/](#) [epubcheck/](#) [sch/](#) ncx.sch [< r46 r145](#) [Hide details](#)

```

1 <sch:schema xmlns:sch="http://www.ascc.net/xml/schematron">
2 <!--
3     A Schematron schema for NCX version 2005-1
4
5     Adopted from Z3986 conformance validator ("ZedVal") schema
6     available in original at http://daisymfc.svn.sourceforge.net/viewvc/daisymfc/trunk
7     /ncx-2005-1.rng?view=log
8
9     This schema uses Schematron 1.5 for integration with Jing, and
10    to get line/column locators in reported errors, as implemented
11    by James Clark. It may be upgraded to ISO Schematron in the future.
12
13    Latest edit: mgylling 20080116
14
15 -->
16 <sch:ns prefix="ncx" uri="http://www.daisy.org/z3986/2005/ncx/" />
17
18 <sch:pattern name="ncx_pageTargUniqValTypeComb" id="ncx_pageTargUniqValTypeComb">
19 <!-- pageTarget combination of value and type attributes is unique -->
20 <sch:rule context="ncx:pageList/ncx:pageTarget[@value]">
21 <sch:assert test="count(//ncx:pageTarget[@value=current()/@value and @type=curren
22     pageTarget combination of value and type is not unique
23 </sch:assert>
24 </sch:rule>
25 </sch:pattern>
26
27 <sch:pattern name="ncx_playOrderOrigin" id="ncx_playOrderOrigin">

```

Change log
[r106](#) by markus.gylling on Feb 12, 2009 [Diff](#)
 Fix for multiple @playorder="1" being flagged as error, as reported at http://groups.google.com/group/epubcheck/browse_thread/thread/b211c474964bd90

Go to:
 Project members, [sign in](#) to write a code review

Older revisions
[r46](#) by markus.gylling on Jan 22, 2008 [Diff](#)
[All revisions of this file](#)

File info
 Size: 3903 bytes, 81 lines
[View raw file](#)

Fertig

Erweiterungen mit Java (nur mit Saxon)

```

<pattern>
  <rule context="bild">
    <let name="path" value="concat($myDir, '\', @src)"/>
    <let name="file" value="f:new($path)"/>
    <assert test="f:exists($file)">Bild
      <value-of select="tokenize($path, '\\')[last()]" />
      existiert nicht
    </assert>
  </rule>
</pattern>

```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <?oxygen SCHSchema="schematron/bild.sch"?>
3
4  <collection>
5    <bild src="pics\bild.jpg"/>
6    <bild src="pics\bild2.png"/>
7    <bild src="pics\bild3.png"/>
8  </collection>
9

```

E [ISO Schematron (XSLT 2.0)]
 Bild bild.jpg existiert nicht
 (f:exists(\$file)) [assert]

Weiterführende Links

- <http://www.schematron.com/>
- [http://standards.iso.org/ittf/PubliclyAvailableStandards/c040833_ISO_IEC_19757-3_2006\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/c040833_ISO_IEC_19757-3_2006(E).zip)
- <http://www.xfront.com/schematron/>
- <http://www.docbook.org/xml/5.0/sch/>
- <https://github.com/jelovirt/dita-schematron/downloads>
- <http://oval.mitre.org/language/about/validating.html>
- EPUB-Check: <http://tinyurl.com/6eelnhy>
- <http://www.w3.org/TR/xmlschema11-1/#cAssertions>
- Hedler, Montero, Kutscherauer: „Schematron – Effiziente Business Rules für XML-Dokumente“, Dpunkt-Verlag, erscheint 06/2011

Prof. Dr. Marko Hedler
Hochschule der Medien

Nobelstraße 10
70569 Stuttgart

Tel. 0711 8923 2141

hedler@hdm-stuttgart.de
www.hdm-stuttgart.de