



---

# Projektspezifische Datenschemata erstellen

XML Schema, ROMA/ODD

Martina Semlak

Zentrum für Informationsmodellierung, Universität Graz

[martina.semlak@uni-graz.at](mailto:martina.semlak@uni-graz.at)



---

# Agenda

- Was ist ein Schema?
- Der Modellbegriff
- Einführung in XML Schema
  - Übungsblock I
- TEI Schema mit Roma
  - ODD Spezifikation
  - Übungsblock II

# Wozu brauche ich ein Schema?

- In TEI gibt es verschiedene Möglichkeiten, um Personennamen auszuzeichnen

```
<persName ref="#AugHipp">  
  Augustinus  
</persName>
```

```
<name ref="#AugHipp" type="person">  
  Augustinus  
</name>
```

```
<rs key="AugHipp" type="person">  
  Augustinus  
</rs>
```



# Vom Modell zur Instanz

- Das Modell (Schema/Dokumentgrammatik) definiert Regeln für die **Struktur** und den **logischen Aufbau** eines XML-Dokuments:
  - Definition von Elementvorrat und deren Beziehung zueinander
  - Hierarchie
  - Eigenschaften (Attribute) von Elementen
  - Inhaltsmodelle / Datentypen
  - Quantifizierung
- Das XML-Dokument ist eine **Instanz** dieses Modells
- Folgt das Dokument seinem angegebenen Schema, wird es als **valide** bezeichnet.
- Das Modell kann in unterschiedlichen Schemasprachen ausgedrückt werden
  - XML Schema, Document Type Definition , Relax NG,...



---

# XML Schema

- ... seit 2001 eine Empfehlung des World Wide Web Consortiums (W3C)
- ... zur Beschreibung der Syntax von XML-Dokumenten
- ... wird selbst in XML notiert
- ... Dateiendung **.xsd**
  
- Unterstützung von
  - einfachen und komplexen Datentypen
  - Quantifizierung der Elemente
  - Ableitung und Vererbung von Typen
  - Namensräumen



# XML Schema: Wurzelement

- `<xs:schema>` Das Wurzelement von XML Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

Schemadefinition

```
  <xs:element name="persName">
```

```
    <xs:complexType>
```

```
      ....
```

```
    </xs:complexType>
```

```
  </xs:element>
```

```
</xs:schema>
```



# XML Schema: Das Typenkonzept

- XML Schema unterstützt 2 Arten von Inhaltsmodellen:

- Einfache Datentypen

```
<xs:element name="forename" type="xs:string" />
```

- besteht aus einfacher Zeichenkette des vordefinierten Typs xs:string
- Attribute werden auch über einfache Datentypen definiert

- Komplexe Datentypen

```
<xs:element name="persName">  
  <xs:complexType>  
    ...  
  </xs:complexType>  
</xs:element>
```

- Elementschachtelungen werden über komplexe Datentypen definiert
- Ein Element kann nur Attribute enthalten, wenn es als komplexer Datentyp definiert wurde



# XML Schema: Einfache Typen

- Einfache Datentypen: Element

```
<xs:element name="forename" type="xs:string" />  
<xs:element name="age" type="xs:integer" />
```

Schemadefinition

```
<forename>Max</forename>  
<age>34</age>
```

XML-Instanz

- **<xs:element>** Definition eines Elements
- **@name** Elementname
- **@type** Datentyp des Inhaltsmodells



# XML Schema: Einfache Typen

- Definition eines Elements mit Inhaltsrestriktion

```
<xs:element name="gender">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="female" />
      <xs:enumeration value="male" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Schemadefinition

```
<gender>female</gender>
<gender>male</gender>
```

XML-Instanz

- **<xs:simpleType>** Definition eines einfachen Datentyps
- **<xs:restriction>** Inhaltsbeschränkung
- **<xs:enumeration>** Aufzählung möglicher Werte



# XML Schema: Einfache Typen

- Einfache Datentypen: Attribut

```
<xs:attribute name="born" type="xs:date" use="required" />
```

Schemadefinition

```
<person born="1973-09-22">
```

```
...
```

```
</person>
```

Schemadefinition

- **<xs:attribute>** Definition eines Attributs
- **@name** Name des Attributs
- **@type** Datentyp
- **@use** Verwendung des Attributs obligatorisch (**required**) oder optional (**optional**)



# XML Schema: Einfache Typen

- Attribut mit Inhaltsrestriktion
- Die Attribute werden auf eine Liste möglicher Werte beschränkt

```
<xs:attribute name="lang" default="de">  
  <xs:simpleType>  
    <xs:restriction base="xs:string">  
      <xs:enumeration value="de" />  
      <xs:enumeration value="en" />  
      <xs:enumeration value="it" />  
      <xs:enumeration value="fr" />  
    </xs:restriction>  
  </xs:simpleType>  
</xs:attribute>
```

Schemadefinition

```
<div lang="en">  
  Hello World!  
</div>
```

XML-Instanz



# XML Schema: Einfache Datentypen

- XML Schema stellt bereits eine Reihe an einfachen Datentypen bereit (Auswahl):
  - **xs:string** beliebige Zeichenkette (Unicode)
  - **xs:decimal** Dezimalziffern (0-9), Dezimalpunkt und Vorzeichen sind erlaubt
  - **xs:integer** positive oder negative Ganzzahl
  - **xs:boolean** Wahrheitswerte (true/false bzw. 0/1)
  - **xs:date** Datierung (JJJJ-MM-TT)
  - **xs:gYear** Jahresangabe (JJJJ)
  - **xs:time** Zeitangabe (hh:mm:ss)
  - **ID** Eindeutiger Bezeichner
  - **NMTOKEN** Kann alphanumerische Zeichen und Punktationszeichen enthalten, aber keine Leerzeichen

# XML Schema: Komplexe Typen

- Elementreihenfolge definieren

```
<xs:element name="persName">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="forename"> ... </xs:element>  
      <xs:element name="surname"> ... </xs:element>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

Schemadefinition

```
<persName>  
  <forename>Hannah</forename>  
  <surname>Müller</surname>  
</persName>
```

XML-Instanz

- **<xs:complexType>** Das Element ist vom Typ complexType, sobald es Kindelemente enthält bzw. Attribute für das Element definiert werden
- **<xs:sequence>** Vorgabe einer Reihe von Kindelementen, dabei muss die Reihenfolge eingehalten werden



# XML Schema: Komplexe Typen

- Alternativen definieren

```
<xs:element name="phone">  
  <xs:complexType>  
    <xs:choice>  
      <xs:element name="office" />  
      <xs:element name="private" />  
    </xs:choice>  
  </xs:complexType>  
</xs:element>
```

Schemadefinition

```
<phone>  
  <office>0316 3802292</office>  
</phone>
```

XML-Instanz

- **<xs:choice>** Ein Element aus der Auflistung muss vorkommen



## XML Schema: Quantifizierung

- Mit den Attributen **minOccurs** / **maxOccurs** wird angegeben, wie häufig ein Element vorkommen kann bzw. muß.
- Wird keine Angabe gemacht, gilt automatisch der Defaultwert = 1
- Angabe durch natürliche Zahlen (1, 7, 10) oder unendlich (unbounded)

0 – 1:

```
<xs:element name="placeName" type="xs:string" minOccurs="0" />
```

1 – 3:

```
<xs:element name="placeName" type="xs:string" maxOccurs="3" />
```

0 – unendlich:

```
<xs:element name="placeName" type="xs:string" minOccurs="0" maxOccurs="unbounded" />
```

# XML Schema: Ein Beispiel

```
<?xml version="1.0" encoding="UTF-8">  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">  
  <xs:element name="person">  
    <xs:complexType>  
      <xs:sequence>  
        <xs:element name="persName" type="personType" />  
      </xs:sequence>  
    </xs:complexType>  
    <xs:attribute name="xml:id" type="ID" />  
  </xs:element>  
  
  <xs:complexType type="personType">  
    <xs:sequence>  
      <xs:element name="forename" type="xs:string" />  
      <xs:element name="surname" type="xs:string" />  
    </xs:sequence>  
  </xs:complexType>  
</xs:schema>
```

Schemadefinition

```
<person id="XS1">  
  <persName>  
    <forename></forename>  
    <surname></surname>  
  </persName>  
</person>
```

XML-Instanz





# XML Schema: Schema erstellen + zuweisen

- Erstellung einer neuen Schemadatei in Oxygen
  - Datei > Neue Datei... > Neues Dokument > XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <!-- Definition von Elementen und Attributen -->
</xs:schema>
```

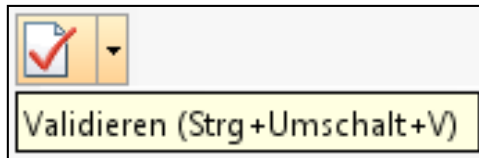
- Ein neues XML Dokument öffnen
  - Datei > Neue Datei... > XML-Dokument
  - Dokument > Schema > Schema zuweisen

```
<?xml version="1.0" encoding="UTF-8"?>
<person xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="person.xsd">
  <!-- Kindelemente -->
</person>
```



# XML Schema: Validierung

- Oxygen



- W3C-Validator for XML Schema (<http://www.w3.org/2001/03/webdata/xsv>)
- XML-Validator (<http://xmlvalidator.new-studio.org>)

**XML Validator Online** BETA

**XML Validation**

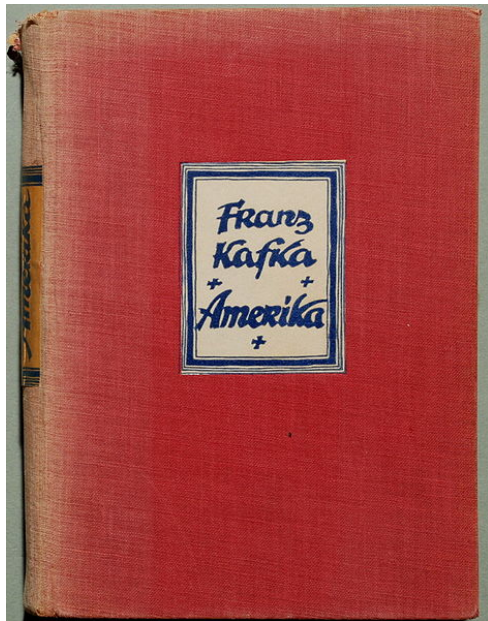
XPath  Schema  DTD

Charset: UTF-8 (Unicode, worldwide) ▼

Choose XML file to validate:  ex-01-refSchema.xml

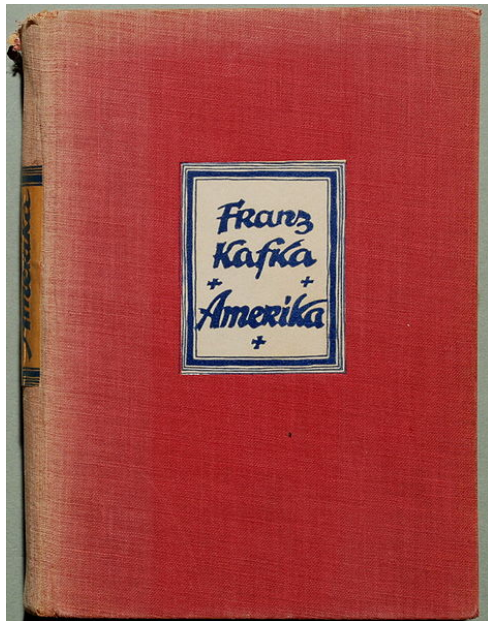
XML Schema:  ex-01.xsd

# XML-Schema: Übung



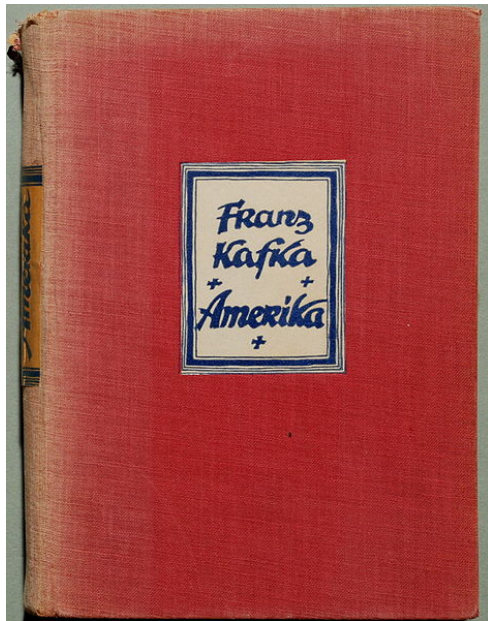
- Öffnen Sie in Oxygen eine neue Datei vom Dokumenttyp XML Schema (Datei > Neue Datei... > Neues Dokument > XML Schema)
- Erstellen Sie ein XML Schema für eine Literaturliste

# XML-Schema: Übung



- Definieren Sie das Wurzelement **listBibl**
- listBibl kann 1 bis unendlich viele Kindelemente **bibl** enthalten
- Das Element bibl wird durch das verpflichtende Attribut **@type** näher spezifiziert. Fügen Sie für das Attribut eine vordefinierte Werteliste ein z.B. Roman, Gedichtband usw.
- Kindelemente von bibl sind:
  - title** (einfach, string, 1)
  - author** (komplex, surname/forename, 1-unendlich)
  - date** (einfach, gYear, 1)
  - publisher** (einfach, string, 1)
  - pubPlace** (einfach, string, 1-3)
  - edition** (einfach, positiveInteger, 0-1)
- Kindelement von author
  - surname** (einfach, string, 1)
  - forename** (einfach, string, 1-2)
- Öffnen Sie ein neues XML Dokument, weisen Sie das XML Schema zu und befüllen Sie die Struktur
- Musterbeispiel: books.xml / books.xsd

# XML-Schema: Übung



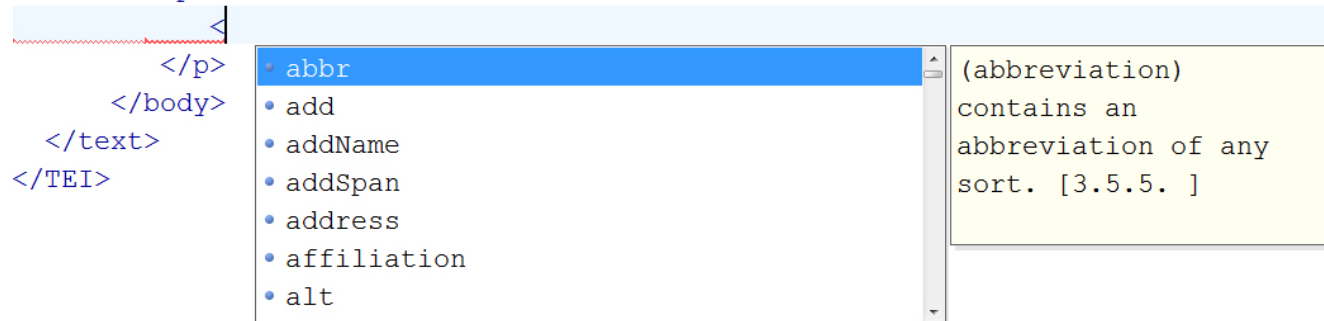
- Die Struktur des XML-Dokuments könnte z.B. so aussehen:

```
<?xml version="1.0" encoding="UTF-8">
<listBibl>
  <bibl type="">
    <title></title>
    <author>
      <forename></forename>
      <surname></surname>
    </author>
    <date></date>
    <publisher></publisher>
    <pubPlace></pubPlace>
    <edition></edition>
  </bibl>
</listBibl>
```

# Anpassung des TEI Schemas mit Roma

- TEI All in Oxygen ermöglicht eine große Auswahl an Elementen

```
<text>
  <body>
    <p>
      </p>
    </body>
  </text>
</TEI>
```



(abbreviation)  
contains an  
abbreviation of any  
sort. [3.5.5. ]

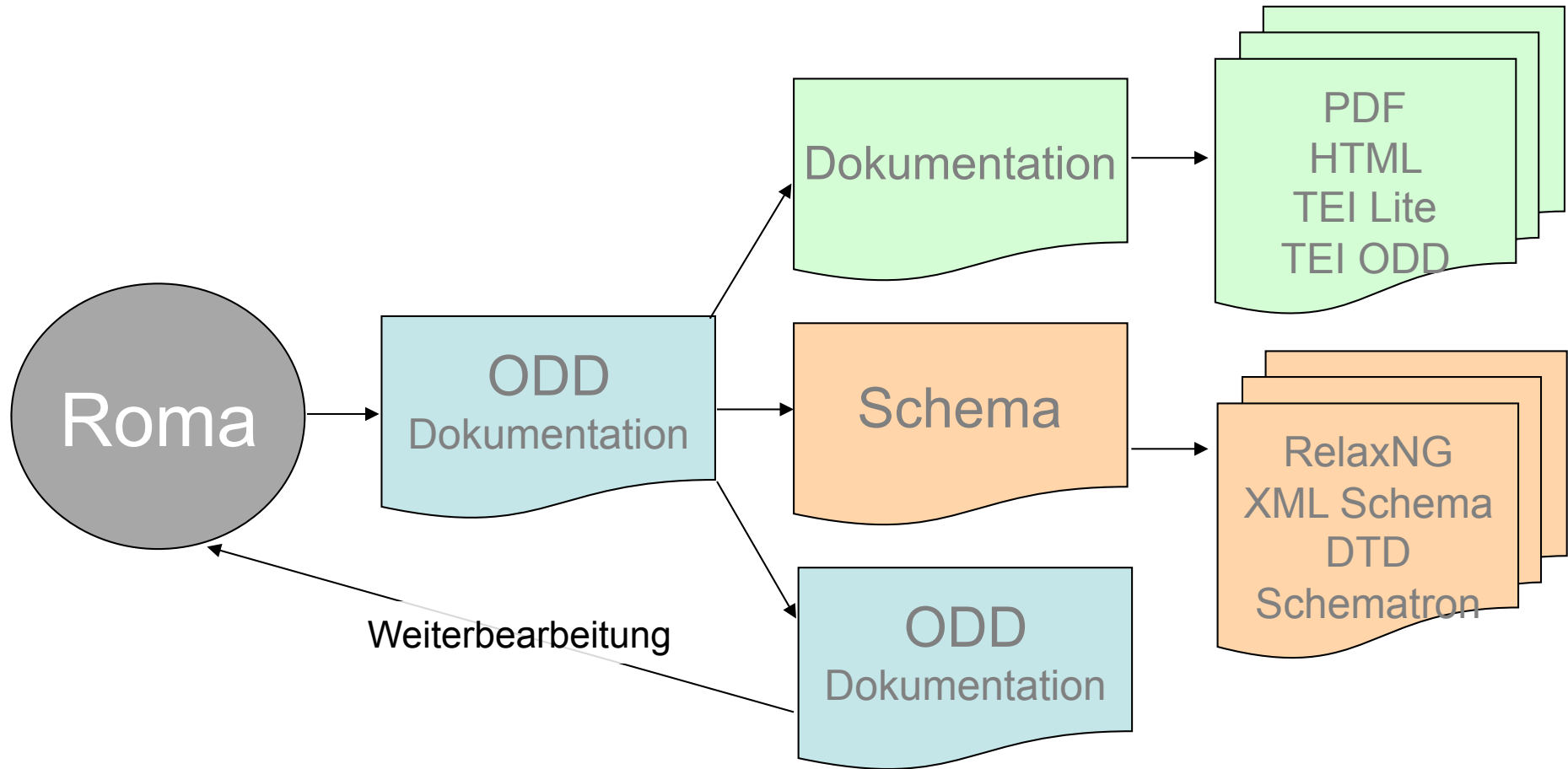
- Die TEI stellt aber auch vordefinierte Schemata zur Verfügung (auch in Oxygen implementiert)
  - TEI Absolutely Bare
  - TEI Lite
  - TEI for Manuscript Description
  - TEI with SVG
  - <http://www.tei-c.org/Guidelines/Customization/>



# Anpassung des TEI Schemas mit Roma

- Erstellung eines projektspezifischen Schemas mit Roma
- Entfernung von Module und Elemente, die nicht benötigt werden
- Je schlanker das Schema, desto konsistenter sind die Daten
- Attributwerte modifizieren
- Veränderung von Klassen
- Neue Elemente hinzufügen
- Roma als
  - Webbasiertes Interface
    - <http://www.tei-c.org/Roma/>
  - Command-line script
    - a) Internetzugang oder b) lokale Kopie der TEI XSL Stylesheets oder c) lokale TEI eXist Datenbank erforderlich

# Anpassung des TEI Schemas mit Roma







## Wiederholung

- Die TEI ist in Modulen organisiert
- Jedes Modul enthält Elementspezifikationen
- Elemente werden durch Attribute näher spezifiziert
- TEI Schemaspezifikation durch Auswahl von Modulen und Elementen sowie deren inhaltliche Anpassung
  - Welches Element gehört zu welchem Modul?
  - Welche Modul- und Attributklasse(n) gibt es?
    - Appendix A - D
    - <http://www.tei-c.org/release/doc/tei-p5-doc/en/html/index.html>

# Roma: New

## TEI Roma: generating customizations for the TEI

TEI Roma is a tool for working with TEI customizations. A TEI customization is a document from which you can generate a schema defining which elements and attributes from the TEI system you want to use, along with customized HTML or PDF documentation of it. The schema generated can be expressed in any of DTD, RELAXNG W3C Schema or Schematron languages.

You can make or modify your TEI customization in several different ways:

- Build up: create a new customization by adding elements and modules to the smallest recommended schema
- Reduce: create a new customization by removing elements and modules from the largest possible schema
- Create a new customization starting from a template
- Use or modify an existing TEI-defined customization
- Upload a customization

Community-maintained customizations can be downloaded from [the TEI website](#)

### Start

A TEI customization is informally referred to as an ODD (for "One Document Does it all")

Roma was written by Arno Mittelbach and is maintained by Sebastian Rahtz. Sanity check written by Ioan Bernevig. Please direct queries to the [TEI @ Oxford](#) project.



---

## Roma: New

- Vordefinierte Möglichkeiten für den Start
  - Build up**: Elemente und Module zum Minimalschema hinzufügen
  - Reduce**: Elemente und Module aus dem Maximalschema entfernen
  - Templates**: Anpassung ausgehend von einem vordefinierten Template
  - Upload**: Eine von ihnen gespeicherte Anpassung erneut öffnen



# Roma: Customize



## Roma: generating validators for the TEI

You are currently working on **My TEI Extension**

### Set your parameters

[New](#) [Customize](#) [Language](#) [Modules](#) [Add Elements](#) [Change Classes](#) [Schema](#) [Documentation](#) [Save Customization](#) [Sanity Checker](#)

#### Set your parameters

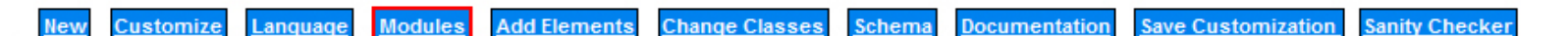
<b>Title</b>	<input type="text" value="My TEI Extension"/>
<b>Filename</b>	<input type="text" value="myTEI"/>
<b>Namespace for new elements</b>	<input type="text" value="http://www.example.org/ns/nonTEI"/>
<b>Prefix for TEI pattern names in schema</b>	<input type="text" value="tei_"/>
<b>Language</b>	<input checked="" type="radio"/> English, <input type="radio"/> Deutsch, <input type="radio"/> Italiano, <input type="radio"/> Español, <input type="radio"/> Français, <input type="radio"/> Portugues, <input type="radio"/> Russian, <input type="radio"/> Svenska, <input type="radio"/> 日本語, <input type="radio"/> 中文
<b>Author name</b>	<input type="text" value="generated by Roma 4.10"/>
<b>Description</b>	<input type="text" value="My TEI Customization starts with modules tei, core, textstructure and header"/>

[Save](#)

Änderung der Metadaten:  
Titel, Dateiname, Autorname oder  
Beschreibung

# Roma: Modules

## Modules



**List of TEI Modules**

	Module name	A short description	Changes
add	<a href="#">analysis</a>	Simple analytic mechanisms	
add	<a href="#">certainty</a>	Certainty and uncertainty	
add	<a href="#">core</a>	Elements common to all TEI documents	
add	<a href="#">corpus</a>	Corpus texts	
add	<a href="#">dictionaries</a>	Dictionaries	
add	<a href="#">drama</a>	Performance texts	
add	<a href="#">figures</a>	Tables, formulæ, notated music, and figures	
add	<a href="#">gaiji</a>	Character and glyph documentation	
add	<a href="#">header</a>	The TEI Header	

Liste aller TEI-Module

**List of selected Modules**

tei

remove [core](#)

remove [analysis](#)

remove [certainty](#)

remove [corpus](#)

remove [dictionaries](#)

remove [drama](#)

remove [figures](#)

remove [gaiji](#)







remove [header](#)

Liste der Ausgewählten Module

- TEI Elemente werden in Modulen hinsichtlich semantischer bzw. struktureller Gemeinsamkeiten gruppiert



# Roma: List of elements

List of elements in module:core						
	Include	Exclude	Name		Description	Attributes
<a href="#">abbr</a>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="text" value="abbr"/>		(abbreviation) contains an abbreviation of any sort.	<a href="#">Change attributes</a>
<a href="#">add</a>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="text" value="add"/>		(addition) contains letters, words, or phrases inserted in the text by an author, scribe, annotator, or corrector.	<a href="#">Change attributes</a>
<a href="#">addrLine</a>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="text" value="addrLine"/>		(address line) contains one line of a postal address.	<a href="#">Change attributes</a>
<a href="#">address</a>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="text" value="address"/>		contains a postal address, for example of a publisher, an organization, or an individual.	<a href="#">Change attributes</a>
<a href="#">analytic</a>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="text" value="analytic"/>		(analytic level) contains bibliographic elements describing an item (e.g. an article or poem) published within a monograph or journal and not as an independent publication.	<a href="#">Change attributes</a>
<a href="#">author</a>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="text" value="author"/>		in a bibliographic reference, contains the name(s) of an author, personal or corporate, of a work; for example in the same form as that provided by a recognized bibliographic name authority.	<a href="#">Change attributes</a>

- Jedes einzelne Element kann in das Schema aufgenommen oder ausgeschlossen werden
- Nicht vergessen: Am Ende der Liste “Save” klicken!

# Roma: Schema



Time to give you a schema

New Customize Language Modules Add Elements Change Classes **Schema** Documentation Save Custom

Creating a schema

Which format do you prefer?

Generate

We use RELAX NG

- RELAX NG schema (XML syntax)
- RELAX NG schema (compact syntax)
- RELAX NG schema (XML syntax)
- ISO Schematron
- Schematron
- W3C schema (in ZIP archive)
- DTD

[http://www.w3.org/wiki/XML\\_Schema\\_Language\\_comparison](http://www.w3.org/wiki/XML_Schema_Language_comparison)

- Für die Schemagenerierung stehen unterschiedliche Formate zur Verfügung



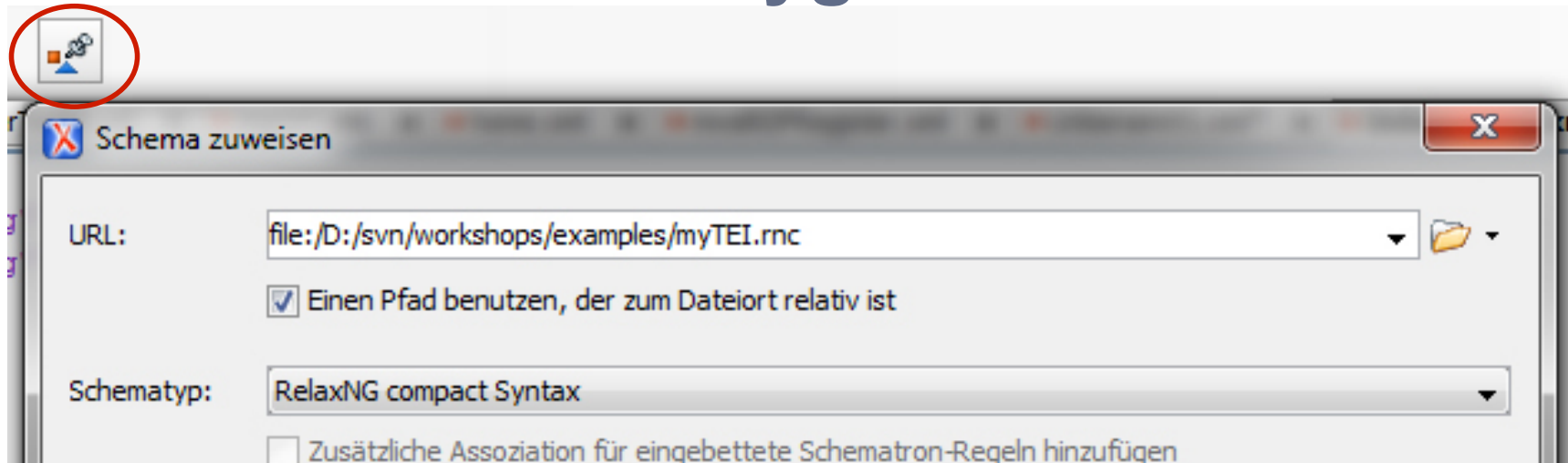
# Roma: Save Customization

[Schema](#)[Documentation](#)[Save Customization](#)[Sanity Checker](#)

- Über den Karteireiter “Save Customization” kann die Anpassung gespeichert und zu einem späteren Zeitpunkt weiterbearbeitet werden.
- Der Download wird automatisch gestartet.
- Die TEI verwendet das Metaformat ODD für die Generierung des Schemas
- Das generierte Dokument ist ein valides TEI Dokument, das spezielle Elemente des “tagdocs” – Modul verwendet.
- Um das Dokument zu einem späteren Zeitpunkt zu modifizieren, klicken Sie auf der Einstiegsseite “Upload a customization” und öffnen Sie das Dokumentationsfile.



## Roma: Schema in Oxygen zuweisen




- Einem TEI-Dokument ein Schema zuweisen: Über das Icon “Schema zuweisen...” in der Toolbar oder über das Menü Dokument > Schema > Schema zuweisen...
- Oxygen fügt die Verknüpfung zum Schema nach der XML Deklaration in das TEI Dokument ein:

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-model href="myTEI.rnc" type="application/xml"  
  schematypes="http://relaxng.org/ns/structure/1.0"?>
```



## Roma: Change attributes

- Attributwerte bearbeiten
- Über die Auswahlliste (rechts) wird ein Modul ausgewählt
- Durch Klick auf “Change attributes” in der Liste der Elemente gelangt man in eine Auflistung aller für dieses Element zulässigen Attribute

div     (text division) contains a subdivision of the front, body, or back of a text. [Change attributes](#)

- Die Attribute können ebenfalls über **include/exclude** übernommen oder ausgeschlossen werden
- Durch Klick auf das Attribut gelangt man zu einer Eingabemaske, die die Modifikation der Attributseinstellung erlaubt



# Roma: Change attributes

**Add a new attribute**

**Attribute name** type

**Class name**

**Is it optional?**  yes  
 no

**Contents** Text

**Default value**

**Closed list?**  yes  
 no

**List of values**

**Description**  
allows the encoder to classify the abbreviation  
convenient  
typology.

**Save**

- Ist das Attribut optional oder obligat?
- Datentyp für den Attributwert festlegen
- Definition eines Defaultwerts
- Ist die Liste auf die vorgeschlagenen Werte beschränkt?
- Liste der zulässigen Werte; durch Komma getrennt
- Beschreibung der Modifizierung
- SAVE!!!**



## ODD

- ODD = „**O**ne **d**ocument **d**oes it all“
- Ein Metaformat zur Generierung multipler Outputs
- Ein TEI Dokument, das Schemaspezifikationen enthält
- Eine Schemaspezifikation besteht aus:
  - einer formalen Dokumentation von verwendeten Modulen, Elementen und Attributen sowie deren Werteinschränkungen
  - einer deskriptiven Dokumentation – dazu werden Elemente aus dem TEI Modul 22 „Documentation Elements“ verwendet  
[<http://www.tei-c.org/release/doc/tei-p5-doc/en/html/TD.html>]



# ODD Spezifikation

```
1 <?xml version="1.0"?>
2 <TEI xmlns="http://www.tei-c.org/ns/1.0" xml:lang="en">
3   <teiHeader>
4     <fileDesc>
5       <titleStmt>
6         <title>IDE Summer School 2013</title>
7         <author>Martina Semlak</author>
8       </titleStmt>
9       <publicationStmt>
10        <p>for use by whoever wants it</p>
11      </publicationStmt>
12      <notesStmt>
13        <note type="ns">http://www.example.org/ns/nonTEI</note>
14      </notesStmt>
15      <sourceDesc>
16        <p>created on Monday 26th August 2013 09:37:34 AM</p>
17      </sourceDesc>
18    </fileDesc>
19  </teiHeader>
20  <text>
21    <front>
22      <divGen type="toc"/>
23    </front>
24    <body>
25      <p>My TEI Customization s
26      <schemaSpec ident="myTEI"
27        <moduleRef key="core" e
28        <moduleRef key="tei" ex
29        <moduleRef key="header" except="" />
30        <moduleRef key="textstructure" except="" />
31      </schemaSpec>
32    </body>
33  </text>
34 </TEI>
```

```
<schemaSpec xml:lang="en" prefix="tei_" docLang="en" ident="ideSchool2013">
  <moduleRef except="" key="core" />
  <moduleRef except="" key="tei" />
  <moduleRef except="" key="header" />
  <moduleRef except="" key="textstructure" />
</schemaSpec>
```

ODD



## ODD Spezifikation

```
<schemaSpec xml:lang="en" prefix="tei_" docLang="en" ident="ideSchool2013">  
  <moduleRef except="" key="core" />  
  <moduleRef except="" key="tei" />  
  <moduleRef except="" key="header" />  
  <moduleRef except="" key="textstructure" />  
</schemaSpec>
```

ODD

- `<schemaSpec>` Elternelement für die Schemaspezifikation
- `@docLang` Dokumentationssprache
- `@ident` Dateiname
  
- `<moduleRef>` Einbindung von Modulen in das Datenmodell
- `@key` Name des TEI Moduls
- `@url` Externes Schema einbinden – zur Verwendung eines anderen XML-Standards wie z.B. SVG
- `@except` Ausschluss von Elementen



## ODD: Element spezifizieren / Attribut löschen

```
<elementSpec ident="div" mode="change" module="textstructure">  
  <attList>  
    <attDef ident="decls" mode="delete" />  
    <attDef ident="met" mode="delete" />  
    <attDef ident="synch" mode="delete" />  
  </attList>  
</elementSpec>
```

ODD

- `<elementSpec>` Element näher spezifizieren
- `@ident` Name des Elements, das modifiziert wird
- `@mode` Aktion: delete | change | add | replace
- `@module` Modulname, aus dem das Element stammt
  
- `<attList>` Liste von Attributen
- `<attDef>` Attributdefinition
- `@ident` Name des Attributs, das modifiziert wird
- `@mode` Aktion: delete | change | add | replace



## ODD: Liste von Attributwerten

```
<elementSpec id="div" mode="change" module="textstructure">  
  <attList>  
    <attDef id="type" mode="change" usage="req">  
      <valList type="closed" mode="replace">  
        <valltem id="Abstract" />  
        <valltem id="Inhaltsverzeichnis" />  
      </valList>  
    </attDef>  
  </attList>  
</elementSpec>
```

ODD

- @usage Die Verwendung des Attributs ist obligatorisch oder fakultativ: req | opt
- <valList> Elternelement der Werteliste
- @type Offene oder geschlossene Listen: open | closed
- <valltem> Liste der zulässigen Attributwerte





## ODD: Dokumentieren

`<p>`Das Element `<gi>div</gi>` darf die Attribute `<att>type</att>` enthalten. Dieses kann die Werte `<val>Abstract</val>`, `<val>Inhaltsverzeichnis</val>` oder `<val>Kapitel</val>` aus der vordefinierten Liste annehmen. Die Verwendung des Moduls `<tag>moduleRef key="tei"</tag>` ist obligat.`</p>`

- `<gi>` Element
- `<att>` Attribut
- `<val>` Attributwert
- `<tag>` Tag
  
- Modul 22 „Documentation Elements“ für weitere Dokumentationselemente [<http://www.tei-c.org/release/doc/tei-p5-doc/en/html/TD.html>]



## Roma/ODD: Übung

- Öffnen Sie Roma (<http://www.tei-c.org/Roma/>)
- Wählen Sie ein neues Anpassungsszenario (z.B. build)
- Ändern Sie den Titel, den Dateinamen und den Autornamen
- Sehen Sie in der TEI nach, welche Module und Elemente Sie benötigen, um eine Personenliste auszuzeichnen (persList)
- Fügen Sie Module hinzu, die Sie dafür benötigen
- Entfernen Sie aus den Modulen Elemente, die Sie nicht benötigen
- Speichern Sie das Schema (Relax-NG XML Syntax) und das ODD-Dokument
- Öffnen Sie ein neues TEI-Dokument und verknüpfen Sie es mit dem Schema. Fügen Sie einen Personenliste ein und Validieren Sie das Dokument (Dokument > Validieren > Validieren)



## Roma/ODD: Übung

- Ändern Sie nun die Attributwerte des Elements <persName>
  - Schließen Sie bis auf @role und @n alle Attribute des Elements <persName> aus
  - Geben sie für @role eine geschlossene Werteliste an
    - z.B. Maler, Schriftsteller, Schauspieler o.Ä.
  - Ändern Sie für @n den Datentyp, der als Wert nur mehr Zahlen zulässt
- Speichern Sie das Schema erneut
- Gehen Sie zurück zu Oxygen und validieren Sie das TEI Dokument erneut (Dokument > Validieren > Validieren (gecached))
- Oxygen stellt nun eine Liste der möglichen Werte für das Attribut @role im Element <persName> bereit. Das Attribut @n darf nur noch Zahlen enthalten. Testen Sie es aus!
- Öffnen Sie das ODD-Dokument in Oxygen und fügen Sie eine kurze Dokumentation über Ihre Modellierungsentscheidungen hinzu
- Öffnen Sie das modifizierte ODD-Dokument erneut in Roma und speichern Sie ihre Dokumentation als PDF



---

## Links

- Schema
  - w3schools.com  
[<http://www.w3schools.com/schema>]
  - W3C XML Schema Specification  
Einführung: [<http://www.edition-w3.de/TR/2001/REC-xmlschema-0-20010502/>]  
Strukturen: [<http://www.edition-w3.de/TR/2001/REC-xmlschema-1-20010502/>]  
Datentypen: [<http://www.edition-w3.de/TR/2001/REC-xmlschema-2-20010502/>]
- Roma/ODD
  - Customizing the TEI with Roma  
[[http://www.tei-c.org/Guidelines/Customization/use\\_roma.xml](http://www.tei-c.org/Guidelines/Customization/use_roma.xml)]
  - Getting Started with P5 ODDs  
[<http://www.tei-c.org/Guidelines/Customization/odds.xml>]



Danke für Ihre Aufmerksamkeit!

Thanks@

IDE and DHOxSS for sharing presentation materials