Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

i|d|e

ZIM

KARL-FRANZENS-UNIVERSITÄT GRAZ
UNIVERSITY OF GRAZ

UNI
GRAZ

Texttransformation mit XSLT

Ulrike Henny, ulrike.henny@uni-wuerzburg.de

8. September 2016

Zentrum für Informationsmodellierung
Austrian Centre for Digital Humanities
Elisabethstraße 59/III, SR 81.31

forschungsschwerpunkt
kultur- und de
utungsgeschi
chte europas

MASTERSTUDIEN
GEISTESWISSENSCHAFTEN

Programm

- Was ist XSLT? Wozu braucht man XSLT? Wie funktioniert XSLT?
- XSLT in oXygen
- XSLT-Prozessoren
- HTML & CSS
- Wichtige Elemente
- Übung
- Anhang: Tipps & Tricks

Programm

- Was ist XSLT? Wozu braucht man XSLT? Wie funktioniert XSLT?
- XSLT in oXygen
- XSLT-Prozessoren
- HTML & CSS
- Wichtige Elemente
- Übung
- Anhang: Tipps & Tricks

Was ist XSLT?

- Eine Sprache, um Text/XML-Dokumente in verschiedene Zielformate zu transformieren
- W3C-Standard seit 1999, aktuell: XSLT 3.0
- *Extensible Stylesheet Language Transformation*
- XSL ist auch XML

Wozu braucht man XSLT?

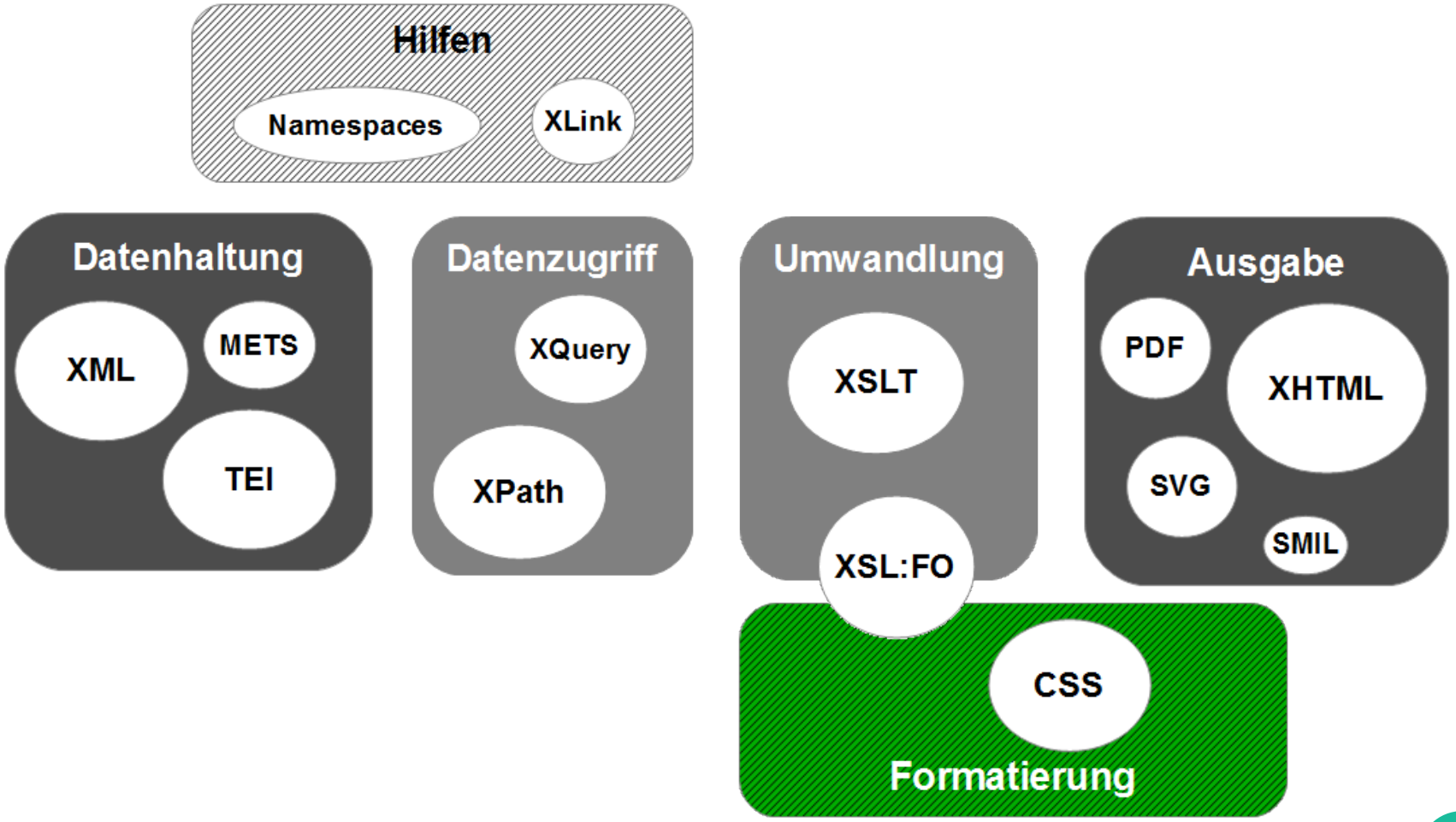
- Verwalten von Dokumenten
  - Aufräumen von Daten
  - Anreicherung mit neuen Daten
  - Zusammenführen mehrerer Dokumente
  - Aufspalten in einzelne Dokumente

- Umwandlung für den Datenaustausch

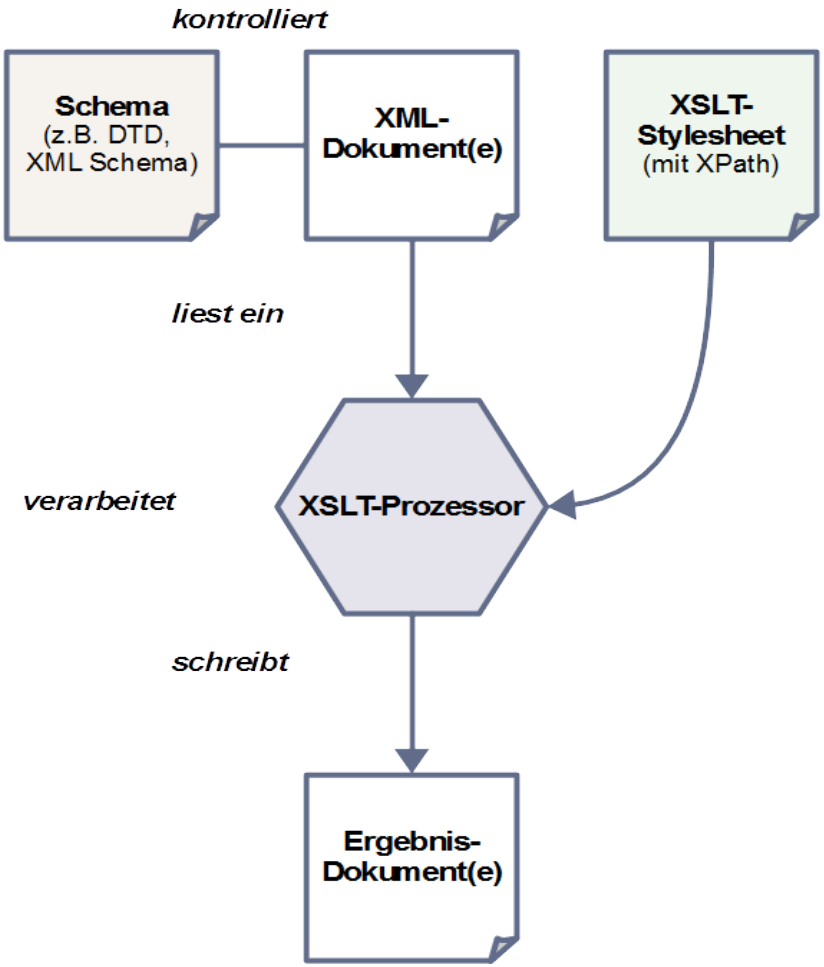
Wozu braucht man XSLT?

- Aufbereitung zur Darstellung und Publikation
  - Auswahl treffen
  - Organisieren
  - Schön machen
  - Funktional machen
  - Ergänzen

Was ist XSLT? XSLT im Kontext



Wie funktioniert XSLT?



Wie funktioniert XSLT?

- Transformationsszenario:
  - Eingabedokument(e) → XSLT-Stylesheet → Zieldokument(e)
  - Eingabebaum → Transformationsbaum → Ausgabebaum
  - Verschiedene Zielformate
    - XML → XSLT → **XML**
    - XML → XSLT → **XHTML**
    - XML → XSLT → **Text**

Wie funktioniert XSLT?

- Transformation von Dokumenten mit XSLT = Umformung ihrer Struktur
- Schablonen / Vorlagen / Templates
  - XPath zum Zugriff auf den Eingabebaum
  - Kopieren und Erstellen von Elementen, Attributen und Text, der ausgegeben werden soll
- Weitere XSLT-Elemente, mit denen die Datenauswahl und -ausgabe gesteuert wird

Wie funktioniert XSLT?

Pseudo-XSLT:

Ich bin ein XML-Dokument

Ich bin ein XSLT-Stylesheet

Ich bin eine Schablone.
Ich passe auf ein Muster.

Ich tue etwas: Ich hole etwas aus dem Ausgangsdokument, ich schreibe etwas in das Zieldokument

Wie funktioniert XSLT?

Beispiel – XSLT-Dokument:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="2.0">
    <xsl:template match="/">
        <h1>
            <xsl:value-of select="//titel" />
        </h1>
    </xsl:template>
</xsl:stylesheet>
```

Wie funktioniert XSLT?

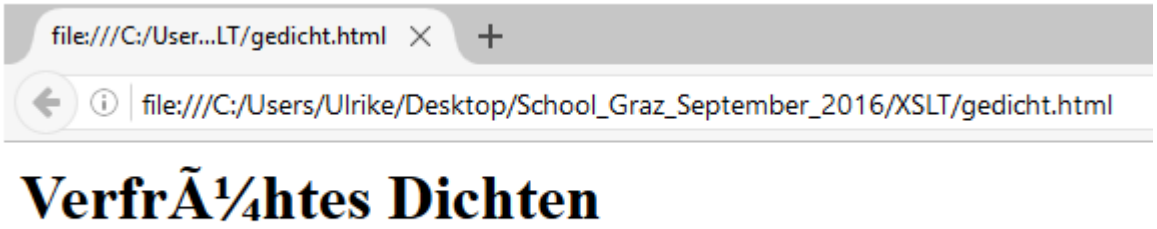
Beispiel – XML mit Verarbeitungsanweisung:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="gedicht.xsl"?>
<gedicht>
   <titel>Verfrühtes Dichten</titel>
   <lg>
      <l>Ja, Reden heilt! Solang’ ich selber spreche,</l>
      <l>Bin ich behütet, letzten Ernst zu hören.</l>
      <l>Lass’ ich mich sinken, tief mein Innres stören:</l>
      <l>Getrost! mich zieht ein Wort zur Oberfläche.</l>
   </lg>
</gedicht>
```

Wie funktioniert XSLT?

Beispiel – Ergebnis:

```
<?xml version="1.0" encoding="UTF-8"?>
<h1>Verfrühtes Dichten</h1>
```



Programm

- Was ist XSLT? Wozu braucht man XSLT? Wie funktioniert XSLT?
- XSLT in oXygen
- XSLT-Prozessoren
- HTML & CSS
- Wichtige Elemente
- Übung
- Anhang: Tipps & Tricks

XSLT in oXygen

- Neue XSLT-Datei anlegen:
  - Im Menü:
    Datei → Neue Datei → XSLT-Stylesheet → Erstellen 
- XML-Dokument mit XSLT-Stylesheet verknüpfen:
  - XML-Dokument öffnen
  - Menü: Dokument → XML-Dokument → XSLT/CSS-Stylesheet zuordnen 

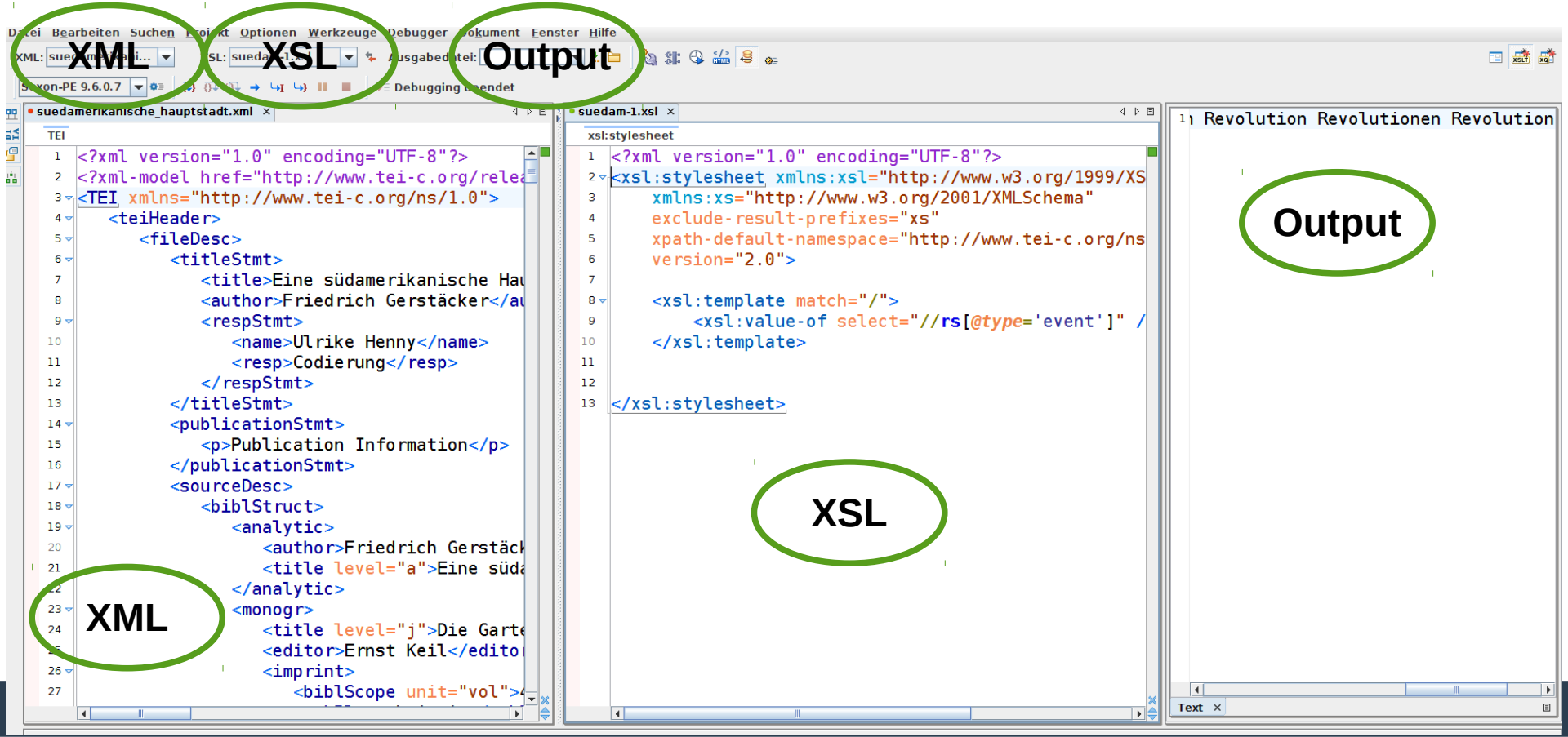
XSLT in oXygen

- Transformation ausführen:
  - Das XML-Dokument öffnen, das die Transformationsanweisung enthält + 
  - Im Menü:
    Dokument → Transformation → Transformation-Szenarios anwenden

XSLT in oXygen

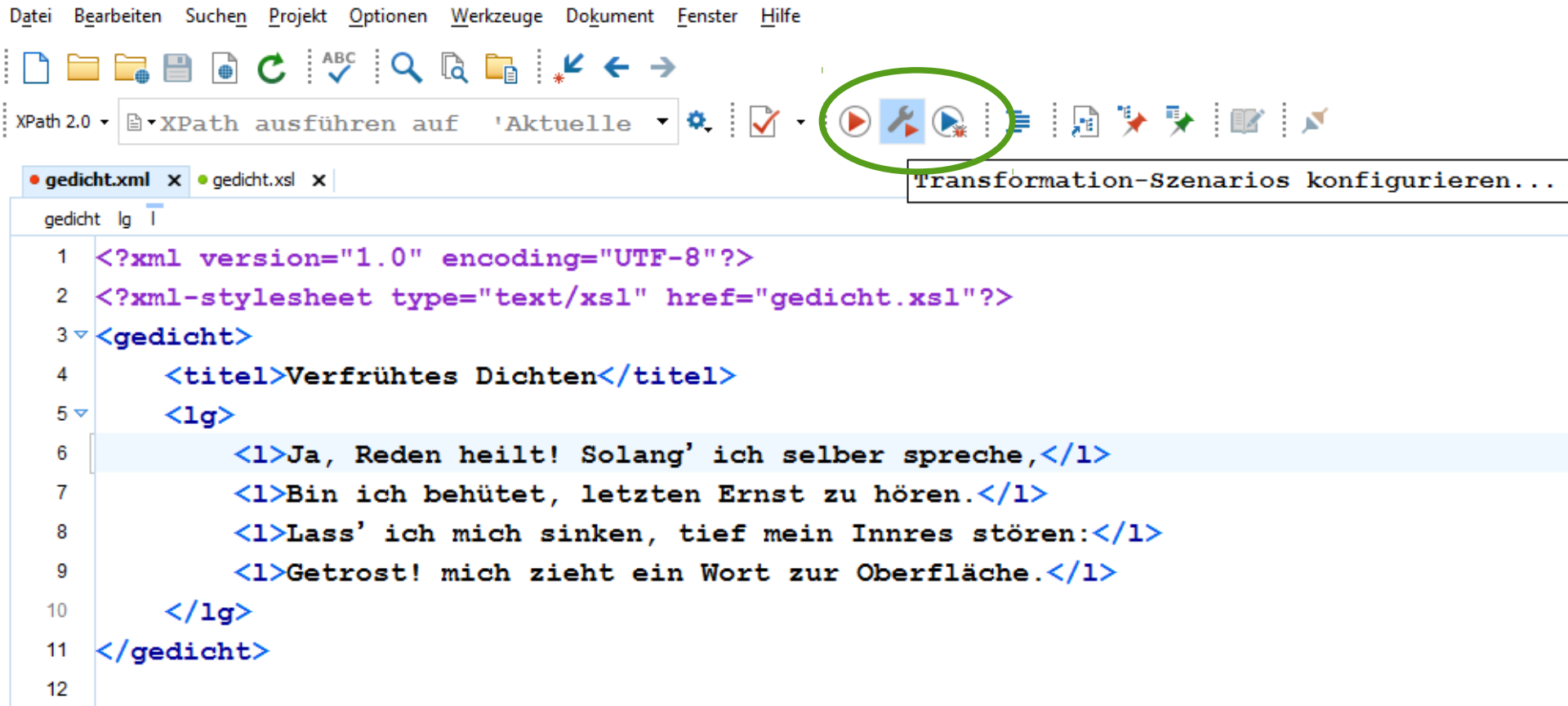
XSLT-Debugger-Ansicht:

Fenster → Perspektive öffnen → XSLT-Debugger



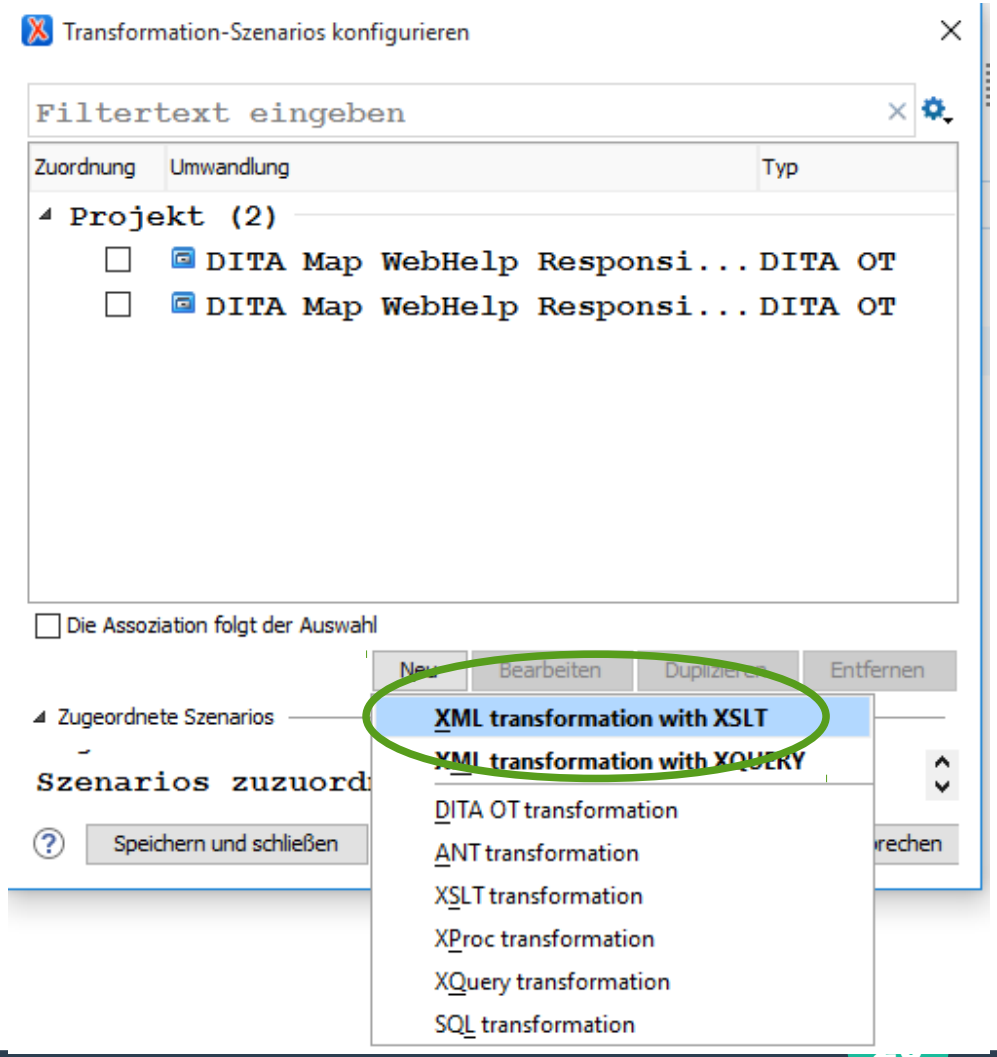
XSLT in oXygen

Transformation-Szenarios einrichten (Schritt 1):
Dokument → Transformation → Transformation-Szenarios konfigurieren...



XSLT in oXygen

Transformation-Szenarios Einrichten (Schritt 2):



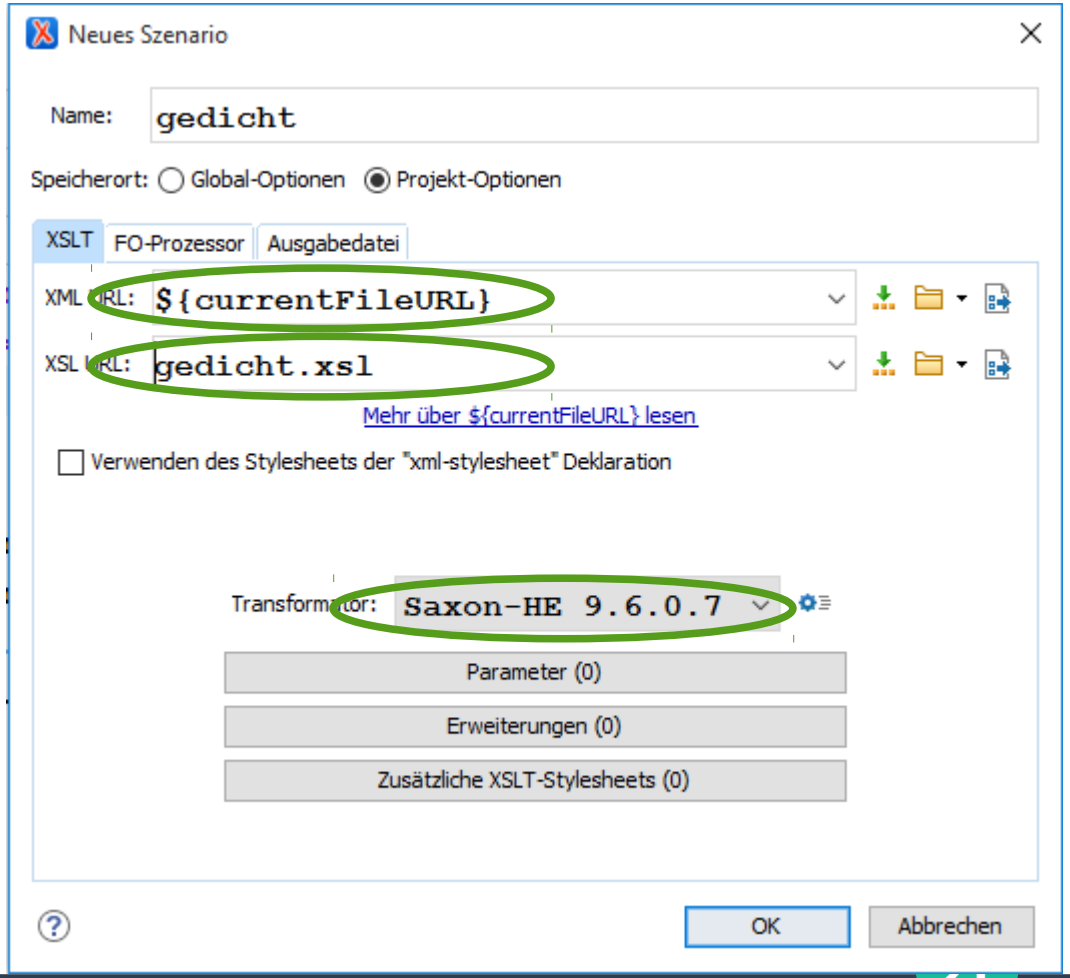
XSLT in oXygen

Transformation-Szenarios einrichten (Schritt 3):

XML-Datei:

XSLT-Datei:

XSLT-Prozessor:



XSLT in oXygen

Transformation-Szenarios einrichten (Schritt 4):

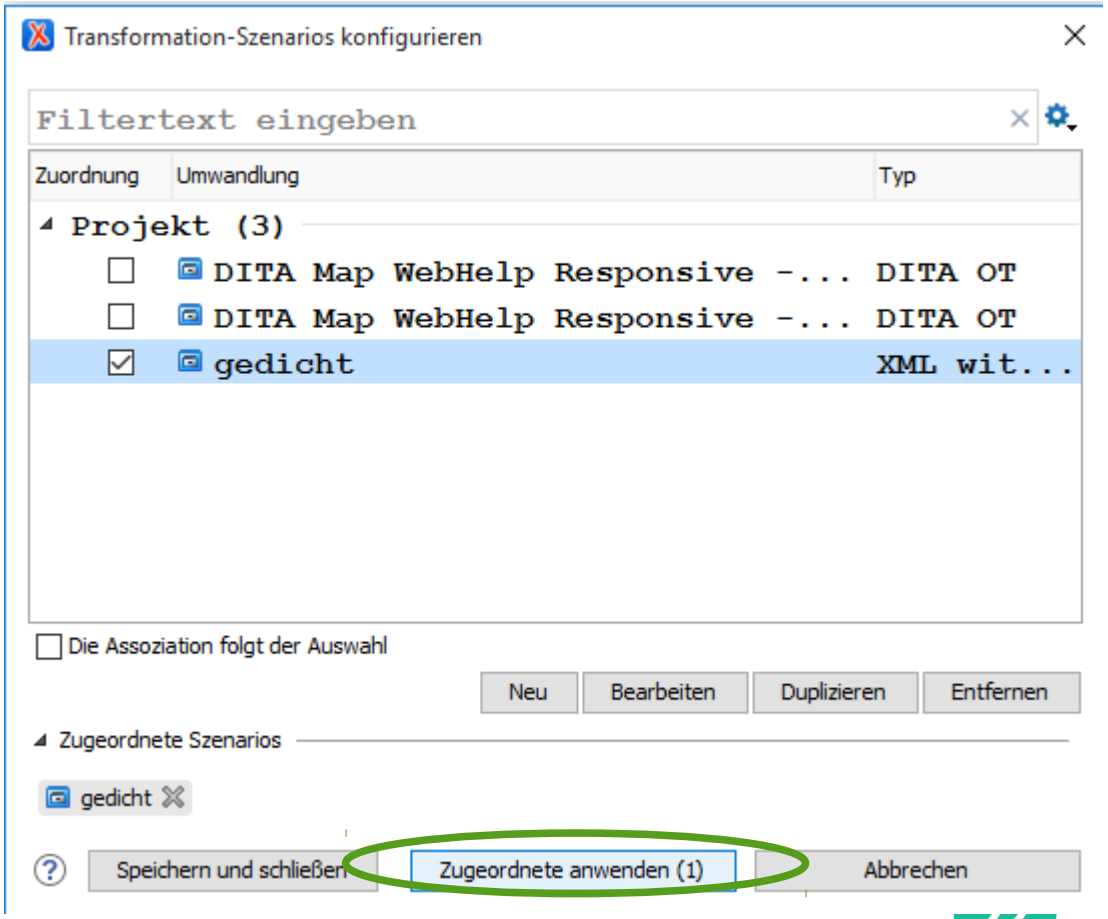
„Zugeordnete anwenden“

oder

„Speichern und schließen“

+

aus der XML-Datei heraus



Übung – erste Transformation

- Öffnen Sie die Datei gedicht.xml in oXygen
- Schreiben Sie ein XSLT-Stylesheet
  - Neue XSLT-Datei erstellen
  - Wurzel-Template anlegen
    - Darin: HTML-Überschriften-Element erstellen
      - Darin: Text des Gedichttitels ausgeben lassen
- Weisen Sie das XSLT-Stylesheet der XML-Datei zu
- Lassen Sie die Transformation laufen
- Schauen Sie sich das Ergebnis im Browser an!

Programm

- Was ist XSLT? Wozu braucht man XSLT? Wie funktioniert XSLT?
- XSLT in oXygen
- XSLT-Prozessoren
- HTML & CSS
- Wichtige Elemente
- Übung
- Anhang: Tipps & Tricks

XSLT-Prozessoren

- XSLT-Prozessor: eingebunden in Webbrowser / Webserver / Software oder eigenständig
- Xalan: aus dem Apache XML Project, Open Source, für XPath / XSLT 1.0, http://xml.apache.org/xalan-j/
- Saxon: Michael Kay, z.T. frei verfügbar, bis Xpath / XSLT 3.0, http://saxon.sourceforge.net

XSLT-Prozessoren

Saxon

- Download:
  https://sourceforge.net/projects/saxon/files/
- Voraussetzung: Java Runtime Environment
  http://www.java.com/de/download
- Beispielaufruf auf der Kommandozeile

  java -jar saxon9he.jar beispiel.xml beispiel.xsl > ergebnis.xml

Programm

- Was ist XSLT? Wozu braucht man XSLT? Wie funktioniert XSLT?
- XSLT in oXygen
- XSLT-Prozessoren
- HTML & CSS
- Wichtige Elemente
- Übung
- Anhang: Tipps & Tricks

HTML & CSS: Beispiel

```
<html xmlns="http://www.w3.org/1999/xhtml">
   <head><title>Erste HTML-Seite</title></head>
   <body>
      <div>
         <h1 style="color: red;">Erste HTML-Seite</h1>
         <h2 style="background-color: blue;">
               Mit einem Untertitel</h2>
         <p title="Ich bin ein Tooltip!">
            Und einem Absatz mit Text.
         </p>
      </div>
   </body>
</html>
```

HTML & CSS: Beispiel

XHTML-Tagset siehe z. B.
http://www.webreference.com/xml/reference/xhtml.html

HTML & CSS: Beispiel

Wie wird nun aus XML mit Hilfe von XSLT HTML?

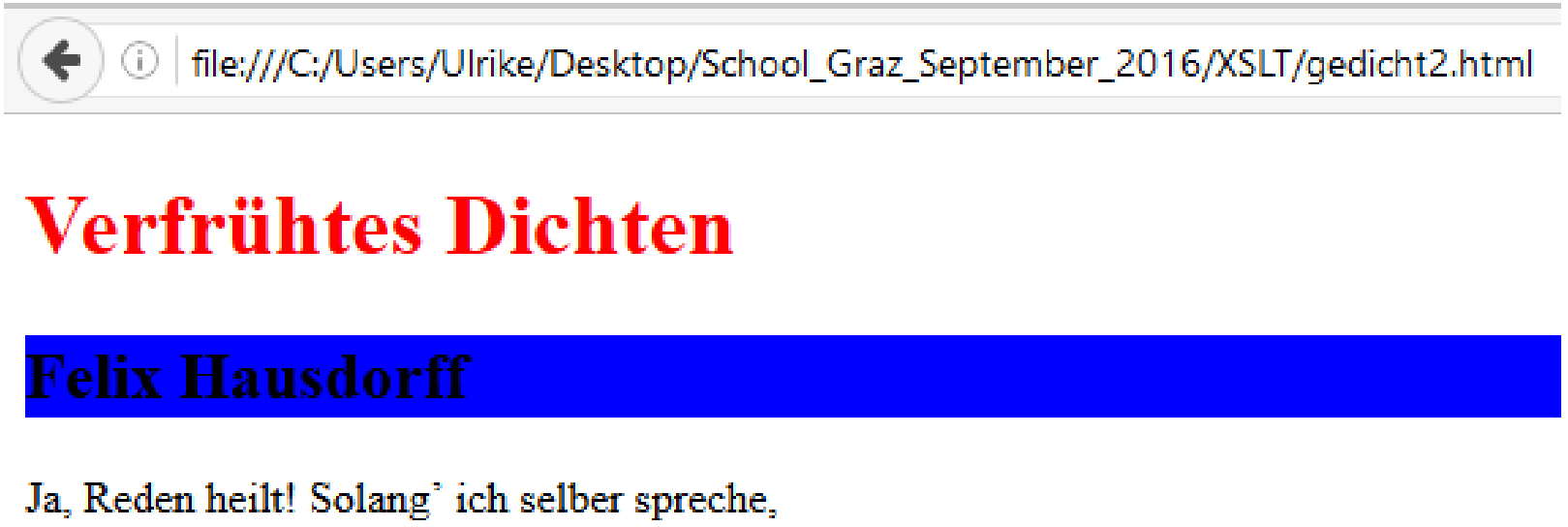
- Wir schreiben das Grundgerüst für eine HTML-Datei in unser Wurzeltemplate
- Wenn wir in den Templates neue Elemente erstellen, sind diese alle HTML-Elemente wie , <p>, <a>, etc.
- In der HTML-Struktur können weitere XSLT-Anweisungen stehen, z. B. <xsl:value-of>

Das könnte dann so aussehen: ...

HTML & CSS: Beispiel

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">
  <xsl:output method="xhtml" />
  <xsl:template match="/">
    <html xmlns="http://www.w3.org/1999/xhtml">
      <head>
        <meta charset="UTF-8"/>
        <title>Gedicht</title>
      </head>
      <body>
        <div>
          <h1 style="color: red;"><xsl:value-of select="//titel" /></h1>
          <h2 style="background-color: blue;"><xsl:value-of select="//@autor"/></h2>
          <p title="1. Vers"><xsl:value-of select="(//l)[1]" /></p>
        </div>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

HTML & CSS: Beispiel



Programm

- Was ist XSLT? Wozu braucht man XSLT? Wie funktioniert XSLT?
- XSLT in oXygen
- XSLT-Prozessoren
- HTML & CSS
- Wichtige Elemente
- Übung
- Anhang: Tipps & Tricks

Wichtige Elemente

- Templates
- Wertausgabe
- Elemente und Attribute
- Schleifen und Sortieren
- Bedingungen

Wichtige Elemente

Zum Mitmachen: briefcorpus.xml

Wichtige Elemente: Templates

<xsl:template match="XPath">

Eine Schablone, die alles verarbeitet, was auf das XPath-Muster passt.

<xsl:apply-templates select="XPath"/>

Wird innerhalb von einer Schablone notiert. Sagt: Für alle hier noch kommenden Eingabedaten sollen noch andere Schablonen berücksichtigt werden, wenn welche passen.
@select: optional; für welche Elemente genau soll dies gelten?

Wichtige Elemente: Templates

Beispiele:

1) `<xsl:template match="/">...</xsl:template>`

2)
```
<xsl:template match="postkarte">
  Postkarte:
  <xsl:apply-templates/>
</xsl:template>
```

3)
```
<xsl:template match="postkarte">
  <xsl:apply-templates select="adresse|stempel"/>
</xsl:template>
```

Wichtige Elemente: Wertausgabe

<xsl:value-of select="XPath" />

Schreibt den Wert (= den Textinhalt) von bestimmten Teilen des Eingabedokuments (bestimmt durch @select) in das Ergebnisdokument

Kurzsyntax innerhalb von Attributkonstruktionen:
<element attribut="{XPath-Ausdruck}"/>

Wichtige Elemente: Wertausgabe

Beispiele:

1)
```
<xsl:template match="name">
    <xsl:value-of select="nachname" />,
    <xsl:value-of select="vorname"/>
</xsl:template>
```

2)
```
<xsl:template match="graphic">
    <img src="{@url}" />
</xsl:template>
```

Wichtige Elemente: Elemente & Attribute

<xsl:element name="string">...</xsl:element>
<xsl:attribute name="string">...</xsl:attribute>

Konstruiert ein Element (bzw. Attribut) im Ergebnisbaum. Elemente (und Attribute) können aber auch direkt in das Zieldokument geschrieben werden (in der Praxis oft):
<elementname attributname="attributwert"/>

<xsl:text> … hier steht Text … </xsl:text>
Schreibt Zeichendaten in das Zieldokument.

Wichtige Elemente: Elemente & Attribute

Beispiel:

```
<xsl:template match="text[@type='zeitungsartikel']">
  <xsl:element name="div">
      <xsl:attribute name="type">
          <xsl:text>Zeitungsartikel</xsl:text>
      </xsl:attribute>
      <xsl:text>Kassen streiten mit Ärzten</xsl:text>
  </xsl:element>
</xsl:template>
```

Wichtige Elemente: Elemente & Attribute

Beispiel:

```
<xsl:template match="text[@type='zeitungsartikel']">
    <div type="zeitungsartikel">
        Kassen streiten mit Ärzten…
    </div>
</xsl:template>
```

Wichtige Elemente: Schleifen & Sortieren

<xsl:for-each select="XPath" />

Mit jedem Element, das vom XPath-Ausdruck zurückgegeben wird, soll etwas getan werden

<xsl:sort select="XPath-Ausdruck" data-type="text | number" order="ascending | descending" case-order="upper-first | lower-first" lang="de | en | ..." />

Sortiert die Knotenmenge, die mit @select ausgewählt wird
Kann z. B. vorkommen in: <xsl:for-each>
Wird als leeres Element direkt nach dem öffnenden Tag von z.B. for-each notiert

Wichtige Elemente: Schleifen & Sortieren

Beispiel:

```
<xsl:template match="/">
  <xsl:for-each select="//stadt/einwohnerzahl">
     <xsl:sort data-type="number"
                    order="descending"/>
     <xsl:value-of select="."/>
     (<xsl:value-of select="parent::stadt"/>)
     <xsl:if test="position() != last()">,</xsl:if>
  </xsl:for-each>
</xsl:template>
```

Wichtige Elemente: Bedingungen

<xsl:if test="XPath">...</xsl:if>

Bedingung: die in <xsl:if> enthaltenen Anweisungen werden nur ausgeführt, wenn der Ausdruck in @test wahr ist

Wichtige Elemente: Bedingungen

Beispiel:

```
<xsl:template match="/">
    Absatzprüfung:
    <xsl:for-each select="p">
      <xsl:if test="normalize-space(.) = ''">
          Absatz Nr. <xsl:value-of select="@n"/> ist leer
      </xsl:if>
      <xsl:if test="position() != last()">;</xsl:if>
    </xsl:for-each>
</xsl:template>
```

Programm

- Was ist XSLT? Wozu braucht man XSLT? Wie funktioniert XSLT?
- XSLT in oXygen
- XSLT-Prozessoren
- HTML & CSS
- Wichtige Elemente
- Übung
- Anhang: Tipps & Tricks

Übung

- Erstellen Sie ein eigenes XSLT-Stylesheet.
- Wenden Sie die vorgestellten XSLT-Elemente auf Material an, dass Sie bereits in XML codiert haben.
- Bitte melden, wenn Hilfe gebraucht wird!

Programm

- Was ist XSLT? Wozu braucht man XSLT? Wie funktioniert XSLT?
- XSLT in oXygen
- XSLT-Prozessoren
- HTML & CSS
- Wichtige Elemente
- Übung
- Anhang: Tipps & Tricks

Anhang: Tipps & Tricks

Namensraum:

- manchmal wird im Zieldokument nichts ausgegeben, weil ein Namensraum vergessen wurde
- wenn das Eingabedokument zu einem bestimmten Namensraum gehört, muss das XSLT das wissen!
- mit dem Attribut *xpath-default-namespace* kann man direkt im XSL-Wurzelelement (<xsl:stylesheet>) notieren, was der Standardnamensraum sein soll
- dann werden die XPath-Ausdrücke z. B. auch TEI finden finden

Anhang: Tipps & Tricks

Beispiel:

- Das Ausgangsdokument ist ein TEI-Dokument:

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
   <teiHeader>
      <fileDesc> ...
```

- Das XSLT soll nun auch TEI finden:

```
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="2.0"
xpath-default-namespace="http://www.tei-
        c.org/ns/1.0"> ...
```

Anhang: Tipps & Tricks

Verstecktes Text-Template:

- wenn man nichts anderes angibt, dann wird das XSLT-Stylesheet immer den ganzen Text des Eingabedokuments automatisch ausgeben
- diese Voreinstellung kann man überschreiben:

```
<xsl:template match="text()"></xsl:template>
```

- ein Template, dass auf "text()" matcht, aber keinen Inhalt hat, also nichts tut, verhindert die standardmäßige Textausgabe
- dann wird nur noch da Text ausgegeben, wo es explizit gesagt wird, z.B. mit <xsl:value-of>

Referenzen

- W3schools XSLT Tutorial.
  http://www.w3schools.com/xsl/
- W3schools XSLT Elements Reference.
  http://www.w3schools.com/xsl/xsl_w3celementref.asp
- W3schools XSLT, XPath, and XQuery Functions.
  http://www.w3schools.com/xsl/xsl_functions.asp
- W3C Candidate Recommendation „XSL Transformations (XSLT) Version 3.0“, 19. November 2015.
  https://www.w3.org/TR/xslt-30/

Vielen Dank!