Ausgewählte Probleme der TEI-Kodierung

IDE Spring School 2010

„Digitale Editionen – Methoden und Technologien für Fortgeschrittene“

Spezialprobleme der TEI-Kodierung

- Kodierung von Sonderzeichen mit Unicode und TEI
- Kritische Apparate
- Kodierung von Briefen

Spezialprobleme der TEI-Kodierung

- **Kodierung von Sonderzeichen mit Unicode und TEI**
- Kritische Apparate
- Kodierung von Briefen

Was ist Unicode?

- „Internationaler Standard, in dem langfristig für jedes sinntragende Schriftzeichen oder Textelement aller bekannten Schriftkulturen und Zeichensysteme ein digitaler Code festgelegt wird“ (http://de.wikipedia.org/wiki/Unicode)
- Bzw. festgelegt werden soll. (OD)

Warum Unicode?

- Ältere Zeichencodierungen konnten lediglich 128 (ASCII, 7 bit) oder 256 (z.B. ISO-8859, 8 bit) Zeichen codieren
- Folge: für unterschiedliche Schriftsysteme mussten verschiedene Zeichencodierungen entwickelt werden und ggf. angegeben werden, in welcher Zeichencodierung eine Datei gespeichert ist (z.B. ISO-8859-1, ISO-8859-5 usw.)

Warum Unicode?

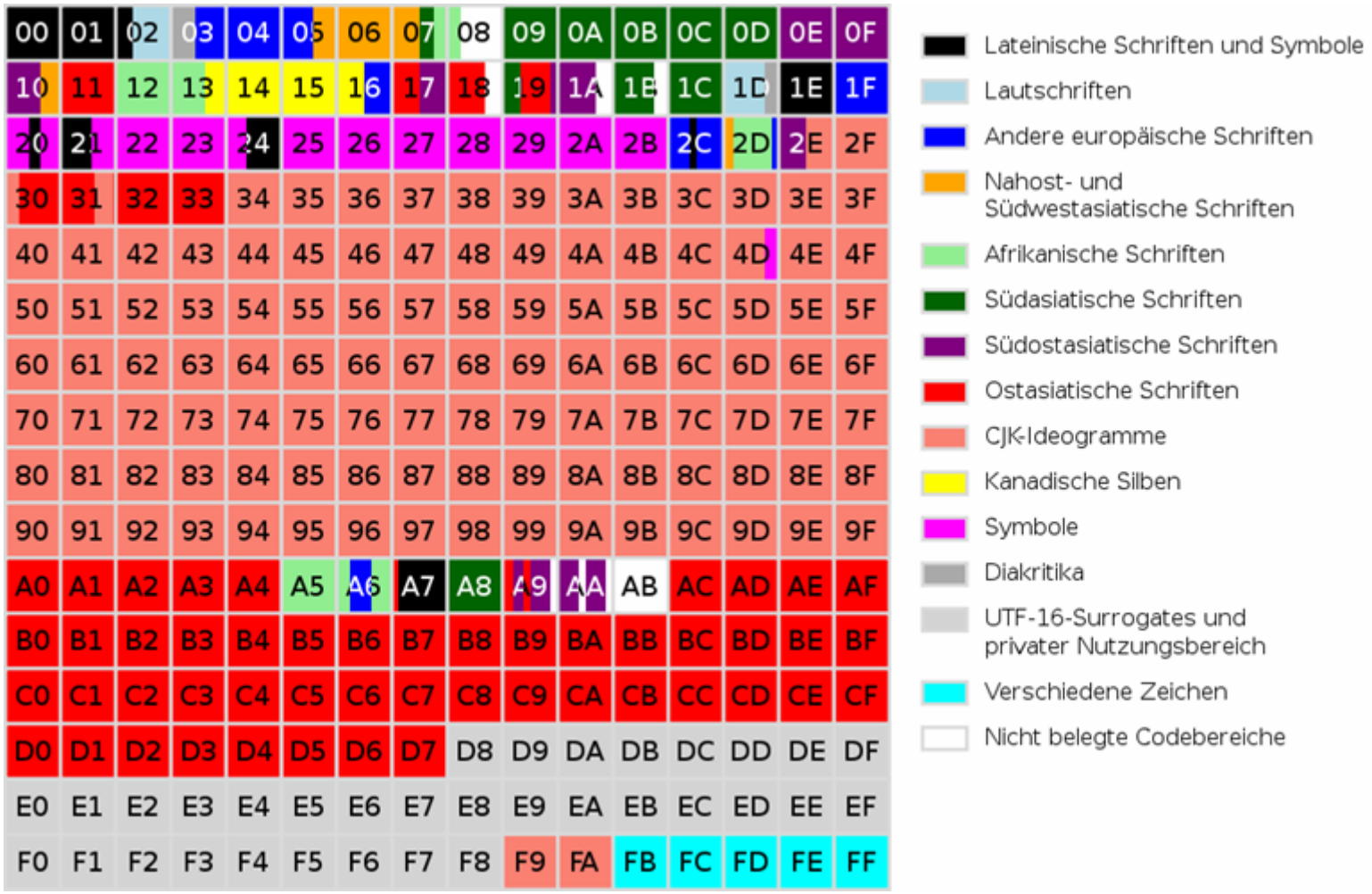
- Unicode soll die verschiedenen miteinander inkompatiblen Zeichenkodierungen ersetzen
- In Unicode 1.0 sollten alle Schriftzeichen der Welt durch 65.536 (2^{16}) sog. „codepoints“ repräsentiert werden
- Inzwischen erweitert auf 17 Bereiche („planes“) von je 65.536 codepoints -> 1.114.112 mögliche Zeichen

Warum Unicode?

- Unicode-Standard wird ständig durch das „Unicode Consortium“ kontinuierlich weiterentwickelt
- Aktuelle Version ist Unicode 5.2 (Okt. 2009)
- Lateinisch, Griechisch, Kyrillisch, Arabisch, Hebräisch, CJK
- Aber auch so Schriften wie Balinesisch, Gotisch, Glagolitisch, Ogham, Linear B usw.
- Mehrere „Private Use Areas“ (PUA)
- Ergänzungswünsche können (und sollten) dem Unicode Consortium gemeldet werden

Was gibt es in Unicode?

- „Normale“ Schriftzeichen: a b c δ Д
- Satzzeichen „ “ ? ! ,
- Whitespace
- Combining Diacritical Marks: ¨ ˝ ◌̈
- Vorkombinierte Zeichen á ä ť ů ḳ̈ ȩ̋
- Symbole ☭ ♘ ♂ ☿ ∑
- Steuerzeichen Wagenrücklauf, EOF
- …



Quelle: http://de.wikipedia.org/wiki/Datei:Roadmap_to_Unicode_BMP_de.svg

Wie finde ich das Zeichen, das ich brauche?

- Codecharts unter www.unicode.org
- Datenbank unter www.decodeunicode.org
- Oder www.isthisthingon.org/unicode/index.php (The UniSearcher)

Kodierung von Unicode in XML-Dateien

- Entweder Zeichen direkt einfügen, z.B. mit Oxygen:
  - α (intuitiv lesbar, wird aber – je nach Zeichensatz – nicht angezeigt)
- Oder mit Zeichenentitäten:
  - Hexadezimal: `ͤ` (gut, entspricht dem Codepoint)
  - Dezimal: `α` (bitte nicht!)

Kombinierende diakritische Zeichen

- Z.B. übergestelltes ᵘ (codepoint U+0367)
- uͧ -> oᵘ
- Generelles Problem:
  - Ungewöhnliche Zeichen werden nur mit entsprechenden Zeichensätzen und entsprechender Software ordentlich angezeigt (oᵘ)
  - Empfehlenswerte Schriften u.a. Arial Unicode MS, Code2000, Schriften des GW

Was tun, wenn Unicode nicht weiterhilft?

- Möglichkeit, die Private Use Areas zu verwenden (U+E000-F8FF, Planes 16 u. 17)
- TEI bietet mit den Elementen <char>, <glyph> und <g> eine Methode zur Definition von Sonderzeichen an
- Character -> ein bestimmter „Buchstabe“ (z.B. ein A)
- Glyph -> eine bestimmte Ausführung eines Buchstabens („langes s“, „rundes r“)

Das Element <charDecl>

- Teil von teiHeader/encodingDesc
- Enthält <char>- und <glyph>-Elemente
- Darin u.a.:
  - <charName> bzw. <glyphName>
  - <charProp>
  - <desc>
  - <mapping>
  - <figure>
- Tipp: Ggf. in separate Datei auslagern und per XInclude in mehrere Dokumente einbinden

Ein Beispiel


Incipit tractatus de efficacia aque bene
dicte, per venerandū magistrū Johannē de turre cremata, sacre
theologie professorem, ordinis predicatorz, tpe concilij Basilien sis
cōpilatus, cōtra Petrū anglicū hereticorz defensorē in bohemia

Beispiel: 

```
<encodingDesc>
  <charDecl>
    <char xml:id="pstroke">
      <charName>LATIN SMALL LETTER P WITH STROKE</charName>
      <desc>Ein unten durchgestrichenes p, normalerweise als Abbreviatur für per</desc>
      <charProp>
        <unicodeName>general-category</unicodeName>
        <value>Ll</value>
      </charProp>
      <mapping type="standardized">p</mapping>
      <figure>
        <graphic url="min_per01-01.jpg"/>
      </figure>
      <note>ganz häufig verwendet</note>
    </char>
  </charDecl>
</encodingDesc>
```

```
<TEI>
…
<text<
  <body>
    <p>t<g ref="#pstroke">empor</g>e concilij Baſilienſis</p>
  </body>
</text>
</TEI>
```

gaiji_test.xml

Spezialprobleme der TEI-Kodierung

- Kodierung von Sonderzeichen mit Unicode und TEI
- **Kritische Apparate**
- Kodierung von Briefen

Kritische Apparate

- Ziel des kritischen Apparats ist es, die Varianten eines Textes in seinen verschiedenen Überlieferungszeugen zu verzeichnen
- In TEI viele Möglichkeiten, Apparate zu kodieren und mit dem Basistext zu verknüpfen
- Wichtigste Elemente:
  - <app></app>
  - <rdg></rdg>
  - <lem></lem>
  - <witness></witness>

Dis ouentürlich büch bewiset wye von einer frouwen genant Melusina die ein merfeye vnd dar zü ein geborne küngin vnd vff den berg Awalon komen wz der selbe berg lit in franckrich/ Vnd wart dise merfeye alle samstag vō dē nabel hin abe ein grosser langer wurme den sy ein halb gespenste was/ Es sint ouch von ir grosse mechtige geslechte komen von künigen fürsten groffen fryen rittern vn̄ knechten der noch kommen noch hüt by disem tage ernampte lüte künige fürsten groffen ritter vnd knecht sint/ Do by mā brüffen mag/ das dise materye durch ir experienez bewiset Das die hystorie wor vnd an ir selber also ist

Ist das der grosse natürliche/ meister/ Aristotiles sprichet an dem anefāg vnd vorrede sines ersten büchs/ Methauisice/ ein ieglich mensche begert von nature vil ezü wissen Darumb so hab ich/ Thüring von Ringoltingē von bern vß üchtlant ein zü mol selezene vnd gar wunderliche fremde hystorie funden in franczösischer sprache vn̄ welscher zungen/ Die aber

1

Dis abentheürlich Buch bewyset
vns von eyner frauwen genant·
Melusina·dye do ein Merfein·vn̄
dar zü ein geborne künigin·vnnd
auß dem berg Awalon kummen ist·der selb
berg leit in Franckreich vn̄ ward dyse Mer
fein alle samstag von dem nabel hin vnder
ein grosser langer wurm dañ sy eȳ halb ge
spenst was·Es seind auch von ir kummen
gar grosse mächtige geschlächt von küni=
gen·Fürsten·Grauen·Freyen·Ritter vnnd
knecht·der aller nach kūmen noch heüt den
tag Künig·Fürsten·Grauen·Freyen·Rit=
ter·vn̄ knecht·benant seind Dabey mā wol
brüfen vnnd versteen mag das dyse matery
durch ir expergēez beweyset das dise hysto
ry war vn̄ gerecht an ir selbs ist·

Seyt das d̕ groß natür=
lich meyster Aristotiles
spricht am anfang vnd
vorred seins ersten bü=
chs Methauistice· Ein
yegklicher mensch bege
ret von natur vil zewis
sen·Vnd darumb so ha=
be ich auch Thüringen
von Ringeltingē vnd auch von Bern auß

92 b

a 2

Einfacher Apparat

```
<app>
  <rdg wit="#BR1">
     Dis ouentúrlich buoch bewiset wye von einer Frowen ge<lb/>nannt
     Melusina ...
  </rdg>
  <rdg wit="#SK1">
     Dis ouentuorlich buoch bewiset wie von einer Frauwen die genantt
     <lb/>Melusina ...
  </rdg>
  <rdg wit="#AS1">
     DAs abenteürlich Buoch beweyset <lb/>uns von eyner frauwen
     genant <lb/>Melusina ...
  </rdg>
</app>
```

Einfacher Apparat

```
<app>
  <lem wit="#BR1">ouentúrlich</lem>
  <rdg wit="#SK1">ouentuorlich</rdg>
  <rdg wit="#AS1">abenteürlich</rdg>
</app>
```

Attribute für <rdg>

- @wit -> Textzeuge (witness)
- @type -> Klassifikation (z.B. orthografisch)
- @cause -> Grund der Variante (z.B. Abschreibfehler)
- @varSeq -> Zeigt die angenommene Reihenfolge der Änderungen an
- @hand -> Schreiber der Lesart
- @resp -> Verantwortlicher der Lesart
- @cert -> Sicherheit

Gruppierung von Lesarten

```
<app>
  <rdgGrp type="orthographic">
    <rdg wit="#BR1">ouentúrlich</lem>
    <rdg wit="#SK1">ouentuorlich</rdg>
  </rdgGrp>
  <rdg wit="#AS1">abenteürlich</rdg>
</app>
```

Die „Witness List“

- @wit-Attribut enthält Verweise, die im Header mithilfe des <listWit>-Elements aufgelöst werden müssen
  - einfache Liste mit leeren <witness>-Elementen oder
  - Beschreibung der Textzeugen, ggf. mit <bibl>- oder <msDesc>-Elementen

Die „Witness List“

Minimal:

```
<listWit>
  <witness xml:id="BR1"/>
  <witness xml:id="SK1"/>
  <witness xml:id="AS1"/>
</listWit>
```

Die „Witness List“

Etwas besser:

```
<listWit>
  <witness xml:id="BR1"><bibl>Melusine. Basel: Richel [um
    1474]</bibl></witness>
  <witness xml:id="SK1"><bibl>Melusine. Straßburg: Knoblochtzer [um
    1477]</bibl></witness>
  <witness xml:id="AS1"><bibl>Melusine. Augsburg: Schönsperger [um
    1488]</bibl></witness>
</listWit>
```

- Alle Elemente von <bibl>, <biblFull>, <biblStruct>, <msDesc> verwendbar

Verknüpfung von Text und Apparat

- Drei Methoden:
  1. „location-referenced“
  2. „double-end-point-attached“
  3. „parallel segmentation“
- Methoden 1 und 2 sowohl „inline“ als auch „external“ kodiert möglich
- Angabe der gewählten Methode im Header (encodingDesc) im Element <variantEncoding method=„location-referenced" location=„inline">

Location referenced & External

- Im <body> des Dokuments der Haupttext, an anderer Stelle oder in anderem Dokument die Varianten.
- Referenzierung durch app@loc:

```
<p n="p1">Dis ouentúrlich buoch bewiset wye von einer Frowen
    ge<lb/>nannt Melusina ...</p>
<!-- … -->
<app loc="p1">
 <lem wit="#BR1">ouentúrlich</lem>
 <rdg wit="#SK1">ouentuorlich</rdg>
 <rdg wit="#AS1">abenteürlich</rdg>
</app>
```

Location referenced & Inline

- app-Element in den Grundtext eingestreut:

```
<p n="1">
  Dis ouentúrlich
  <app loc="1">
    <rdg wit="#SK1">ouentuorlich</rdg>
    <rdg wit="#AS1">abenteürlich</rdg>
  </app>
  buoch bewiset wye von einer Frowen ge<lb/>nannt
  Melusina ...
</p>
```

Double-End-Point & External

- Im <body> des Dokuments der Haupttext, ggf. mit <anchor> segmentiert, an anderer Stelle oder in anderem Dokument die Varianten.
- Referenzierung durch app@from und app@to:

```
<p>Dis <anchor xml:id="A1"/> ouentúrlich <anchor xml:id="A2"/> buoch
    bewiset wye von einer Frowen ge<lb/>nannt Melusina ...</p>
<!-- … -->
<app from="#A1" to="#A2">
  <rdg wit="#SK1">ouentuorlich</rdg>
  <rdg wit="#AS1">abenteürlich</rdg>
</app>
```

Double-End-Point & Inline

- app-Element in den Grundtext eingestreut:

```
<p n="1">
  Dis <anchor xml:id="A1"/>ouentúrlich
  <app from="A1">
  <rdg wit="#SK1">ouentuorlich</rdg>
  <rdg wit="#AS1">abenteürlich</rdg>
  </app>
  buoch bewiset wye von einer Frowen ge<lb/>nannt Melusina ...
</p>
```

Parallel Segmentation

- Bei Abweichungen wird der „Grundtext“ als Variante im app-Element notiert
- Nur inline kodiert möglich, Untervarianten können geschachtelt werden

```
<p n="1">
  Dis
  <app>
    <lem wit="#BR1">ouentúrlich</lem>
    <rdg wit="#SK1">ouentuorlich</rdg>
    <rdg wit="#AS1">abenteürlich</rdg>
  </app>
  buoch bewiset wye von einer Frowen ge<lb/>nannt Melusina ...
</p>
```

Vor und Nachteile

- Kodierungsvariante sollte je nach Gegenstand gewählt werden
  - Referenced entspricht der klassischen Druckedition, ist relativ schnell zu erstellen, aber z.T. ungenau in der Referenzierung
  - Double-End-Point ist relativ komplex zu codieren und weiterzuverarbeiten, aber exakt und als einzige Form in der Lage, mit überlappenden Strukturen umzugehen
  - Parallel Segmentation ist leicht mit XSLT zu verarbeiten, aber unflexibel bei komplexen Veränderungen und überlappenden Strukturen

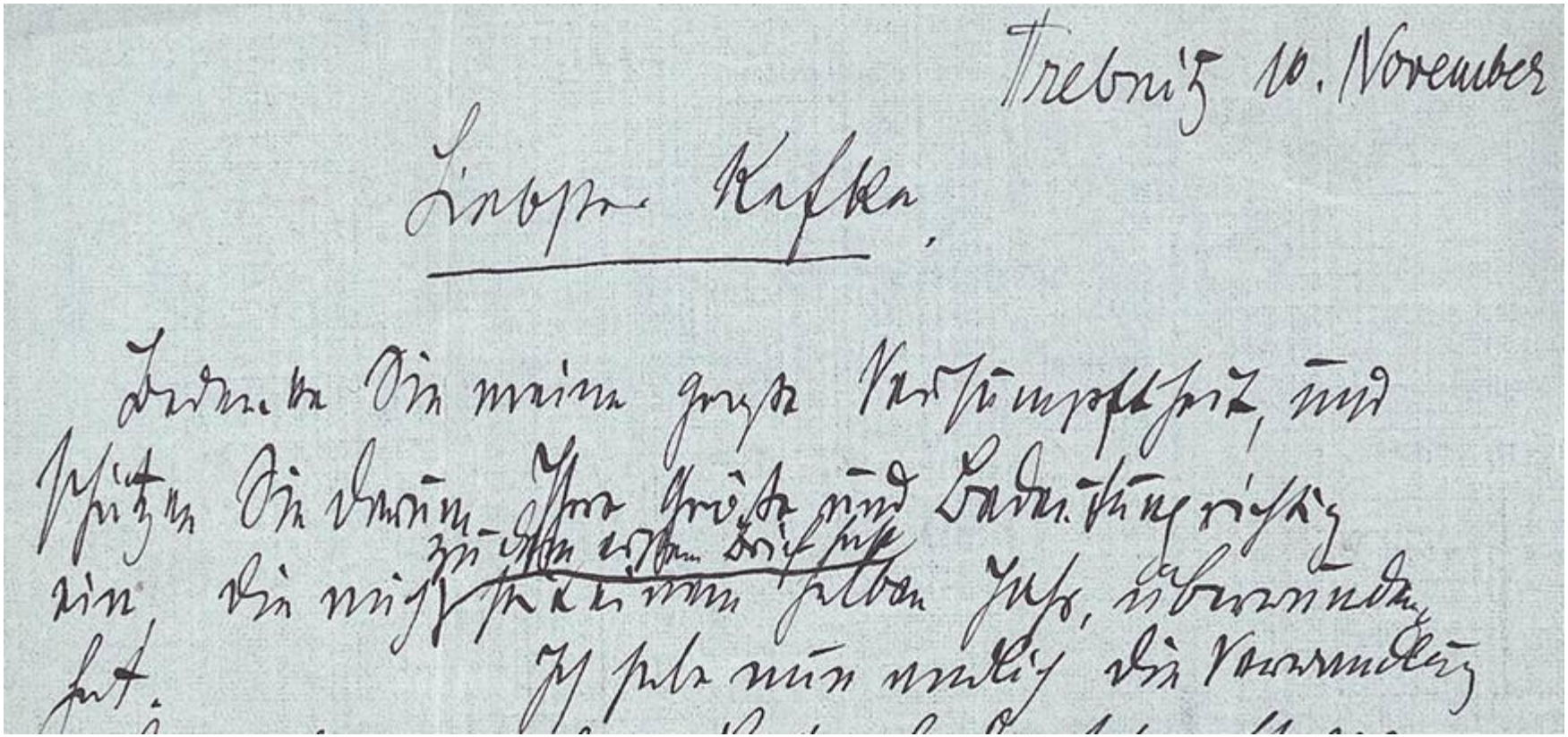
Spezialprobleme der TEI-Kodierung

- Kodierung von Sonderzeichen mit Unicode und TEI
- Kritische Apparate
- **Kodierung von Briefen**

Kodierung von Briefen

- Briefe i.d.R. als eigenes Dokument bzw. als -Element (bei Briefsammlungen) ediert
- Elemente <opener> und <closer> als Container-Elemente, z.B. für
  - <dateline>
  - <byline>
  - <salute>

Ein Beispiel



Transkription

Brief von Franz Werfel an Franz Kafka

Trebnitz 10. November

Liebster Kafka.

Bedenken Sie meine große Versumpftheit, und schätzen Sie darum Ihre Größe und Bedeutung richtig ein, die mich (zu dem ersten Brief heute) seit einem halben Jahr, überwunden hat.

Einfache Kodierung

```
<div type="letter">
  <head>Brief von Franz Werfel an Franz Kafka</head>
  <opener>
    <dateline>Trebnitz 10. November</dateline>
    <salute><hi rend="underline">Liebster Kafka.</hi></salute>
  </opener>
  <p>Bedenken Sie meine große Versumpftheit, und <lb/> schätzen Sie darum
     Ihre Größe und Bedeutung richtig <lb/> ein, die mich zu dem ersten Brief
     heute seit einem halben Jahr, überwunden <lb/> hat.</p>
  <!-- ... -->
  <closer>
    <salute>Herzlich Ihr Franz Werfel</salute>
  </closer>
</div>
```

Verfeinerung

- Wichtige Bestandteile zur Tiefenerschließung
  - Auszeichnung von Personen
  - Auszeichnung von Ortsnamen
  - Auszeichnung von Datumsangaben
  - Auszeichnung von nachträglichen Hinzufügungen
- Zentrale TEI-Kapitel
  - 13 (Names, Dates, People and Places)
  - 11 (Representation of primary sources)

Verbesserte Kodierung

```
<opener>
  <dateline>
    <placeName type="city" ref="#Treb">Trebnitz</placeName>
    <date when="1915-11-10">10. November</date>
  </dateline>
  <salute>Liebster <persName ref="#Kaf">Kafka</persName>.</salute>
</opener>
<p>Bedenken Sie meine große Versumpftheit, und schätzen Sie darum Ihre
Größe und Bedeutung richtig ein, die mich <add hand="#Werf">zu dem ersten
Brief heute</add> seit einem halben Jahr, überwunden hat.</p>
<closer><salute>Herzlich Ihr
  <persName ref="#Werf"><forename>Franz</forename>
  <surname>Werfel</surname></persName></salute>
</closer>
```

Verbesserte Kodierung

```
<opener>
  <dateline>
    <placeName type="city" ref="#Treb">Trebnitz</placeName>
    <date when="1915-11-10">10. November</date>
  </dateline>
 <salute>Liebster <persName ref="#Kaf">Kafka</persName>.</salute>
</opener>
<p>Bedenken Sie meine große Versumpftheit, und schätzen Sie darum Ihre
Größe und Bedeutung richtig ein, die mich <add hand="#Werf">zu dem ersten
Brief heute</add> seit einem halben Jahr, überwunden hat.</p>
<closer><salute>Herzlich Ihr
  <persName ref="#Werf"><forename>Franz</forename>
  <surname>Werfel</surname></persName></salute>
</closer>
```

Auflösung von Referenzen

- Möglichkeit, in Listen (listPerson, listPlace) alle Namen zu sammeln und mit Zusatzinformationen zu versehen
- Wenn die Namen mit @xml:id versehen sind, besteht die Möglichkeit auf sie zu referenzieren
- Unterschiedliche Namensformen („Lieber Kafka“, „Hallo Franz“ „Franzl“ usw.) können dadurch abgefangen werden
- Eröffnet gute Möglichkeiten für Register und Verknüpfungen

Beispiele

```
<listPerson>
  <person xml:id="Kaf">
    <persName>
      <surname>Kafka</surname>
      <forename>Franz</forename>
    </persName>
    <birth when="1883-07-03"/>
    <death when="1924-06-03"/>
  </person>
  <person xml:id="Werf">
    <persName>
      <surname>Werfel</surname>
      <forename>Franz</forename>
    </persName>
    <birth when="1890-09-10"/>
    <death when="1945-08-26"/>
  </person>
</listPerson>
```

```
<listPlace>
  <place xml:id="Treb">
    <placeName>Trzebnica</placeName>
    <placeName
      type="german">Trebnitz</placeName>
    <location>
      <geo>51° 18′ N, 17° 4′ O</geo>
    </location>
  </place>
</listPlace>
```

<persName> vs. <name> vs. <rs>

- Kodierung durch persName (bzw. placeName usw.) wird in den TEI-Richtlinien empfohlen
- Alternativen sind die Kodierung mit <name> und/oder <rs>
- <name> als Element für „Nomen oder Nominalphrasen“
- <rs> (referencing string) ist noch allgemeiner gehalten, jede Art von string, die auf etwas verweist

Beispiele

- <name type="person" ref="#Kaf">Kafka</name>
- <name type="place" ref="#Treb">Trebnitz</name>

- Bedenken
  <rs type="person" ref="#Kaf">Sie</rs>
  meine große Versumpftheit und schätzen Sie darum Ihre Größe und Bedeutung richtig ein, die
  <rs type="person" ref="#Werf">mich</rs>
  zu dem ersten Brief heute seit einem halben Jahr, überwunden hat.

Vor- und Nachteile

- <name> und <rs> sind flexibler
- <persName> etc. ist genauer und bietet bessere Möglichkeiten der „Tiefenkodierung“ (z.B. können noch Anreden, Titel, Namenszusätze usw. kodiert werden)
- Auszeichnung mit <persName> etc. ist etwas leichter mit XSLT zu verarbeiten

Pause!