



Sonderzeichen, TEI und Unicode

O. Duntze



Behandlung von Sonderzeichen

- Bei Transkriptionen älterer und/oder handschriftlicher Texte häufig Sonderzeichen
- Inzwischen zahlreiche Sonderzeichen im Unicode-Standard definiert
- Außerdem bietet die TEI im gaiji-Modul Elemente an, durch die Sonderzeichen definiert, beschrieben und in der Transkription eingesetzt werden können



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 Standards der 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
- Codepoints werden hexadezimal angegeben: z.B. U+0041 = A



Warum Unicode?

- Unicode-Standard wird ständig durch das „Unicode Consortium“ kontinuierlich weiterentwickelt
- Aktuelle Version ist Unicode 6.2 (Sept. 2012)
- 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 ð Ð
- Satzzeichen „ “ ? ! ,
- Whitespace .. " ´
- Combining Diacritical Marks: á ä t' ù ã ě
- Vorkombinierte Zeichen
- Symbole 🚩 ♞ ♂ ♀ Σ
- Steuerzeichen Wagenrücklauf, EOF
- ...

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

- Lateinische Schriften und Symbole
- Lautschriften
- Andere europäische Schriften
- Nahost- und Südwestasiatische Schriften
- Afrikanische Schriften
- Südasiatische Schriften
- Südostasiatische Schriften
- Ostasiatische Schriften
- CJK-Ideogramme
- Kanadische Silben
- Symbole
- Diakritika
- UTF-16-Surrogates und privater Nutzungsbereich
- Verschiedene Zeichen
- Nicht belegte Codebereiche

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/charts/
- Datenbank unter www.decodeunicode.org
- Aktueller: www.fileformat.info/
- Oder www.isthisthingon.org/unicode/index.php
(The UniSearcher)



Was ist ein "Encoding"

- Unicode ist lediglich ein abstrakter Standard, der jedem vorhandenen Zeichen eine Nummer (den codepoint) zuweist
- Codepoints werden hexadezimal angegeben (U+1F46 usw.)
- Die Codierung (encoding) legt fest, in welcher Form die codepoints in einer Datei gespeichert werden
 - UTF8, UTF16 usw. sind **nicht identisch mit Unicode**, sondern Standards, wie Unicode-Zeichen gespeichert werden
- UTF-16 -> jedes Zeichen wird mit 2 Byte gespeichert (entspricht dem Codepoint)
 - Big-Endian (höherwertige Bits zuerst), Little-Endian (niedrigere Bits zuerst)
- UTF-8 -> Häufige Zeichen (lateinisches Alphabet) werden in 1 Byte gespeichert, seltenere in 2 oder 3 Byte
 - D.h. bei einem Text, der nur aus lateinischen Buchstaben ohne Umlaute besteht, ist eine UTF-8-codierte Datei nur halb so groß, wie eine UTF-16

Codierung in XML angeben

- Encoding wird in der XML-Declaration angegeben
 - `<?xml version="1.0" encoding="UTF-8" ?>`
- Sollte immer (!), auch in XSLT-Skripten usw. angegeben werden
- BOM = Byte Order Mark
 - Markierung zu Beginn einer Datei (2-4 Byte), die das encoding angibt
 - Z.B.
 - UTF-8: EF BB BF
 - UTF-16 (BE) FE FF
 - UTF-16 (LE) FF FE
 - Führt oft zu Problemen, wenn ein XML-Editor automatisch eine BOM schreibt, ein anderer sie aber ignoriert. Dann tauchen vor der XML-Deklaration komische Zeichen auf: `ï»¿`, `þÿ`, `ÿþ`
 - Lösung: Einstellungen der XML-Editoren prüfen, Zeichen löschen

Kodierung von Unicode in XML-Dateien

- Entweder Zeichen direkt einfügen, z.B. mit Oxygen:
 - α intuitiv lesbar, wird aber – je nach Zeichensatz – nicht angezeigt
außerdem Verwechslungen bei 'ähnlichen' Zeichen möglich

- Oder mit Zeichenentitäten:
 - Hexadezimal: `ͤ` gut, entspricht dem Codepoint
 - Dezimal: `α` bitte nicht!
 - Benannt: `α` oft besser lesbar, aber die Entitätennamen müssen vorab definiert werden (geht nur per DTD)



Kombinierende diakritische Zeichen

- Z.B. übergestelltes ^u (codepoint U+0367) = oͧ -> ö vs. o□
- Generelles Problem:
 - Ungewöhnliche Zeichen werden nur mit entsprechenden Zeichensätzen und entsprechender Software ordentlich angezeigt
 - Empfehlenswerte Schriften u.a. Arial Unicode MS, Junicode (<http://junicode.sourceforge.net/>), Code2000
 - MUFI (Medieval Unicode Font Initiative):
<http://gandalf.aksis.uib.no/mufi/>
 - Weniger empfehlenswert: Mediaevum
- Z.T. gibt es mehrere Wege, ein und dasselbe Zeichen darzustellen:
 - Ä kann als A (U+0041) und Trema " ('COMBINING DIAERESIS' = U+0308) oder als vorkombiniertes Ä ('LATIN CAPITAL LETTER A WITH DIAERESIS' = U+00C4) kodiert werden
 - Kann durch "Unicode Normalization Forms" gelöst werden (auch automatisiert)
 - <http://www.unicode.org/reports/tr15/>



Was tun, wenn Unicode nicht weiterhilft?

- A)
 - ohne weitere Deklaration die Private Use Areas (PUA) verwenden (U+E000-F8FF und die kompletten Unicode-Planes 16 u. 17)
 - den entsprechenden Zeichensatz mitliefern
 - hoffen, dass schon alles klappen wird

- B)
 - TEI bietet mit den Elementen <char>, <glyph> und <g> eine flexible Methode zur Definition von Sonderzeichen und ggf. deren Umsetzung an
 - Character -> ein bestimmter „Buchstabe“ (z.B. ein A)
 - Glyph -> eine bestimmte Ausführung eines Buchstabens („langes s“, „rundes r“)
 - char und glyph können z.B. per XSLT auch in Sonderzeichen aus der PUA umgewandelt werden!
 - Problem: Element g nicht in sourceDoc verfügbar -> TEI-Unsinn



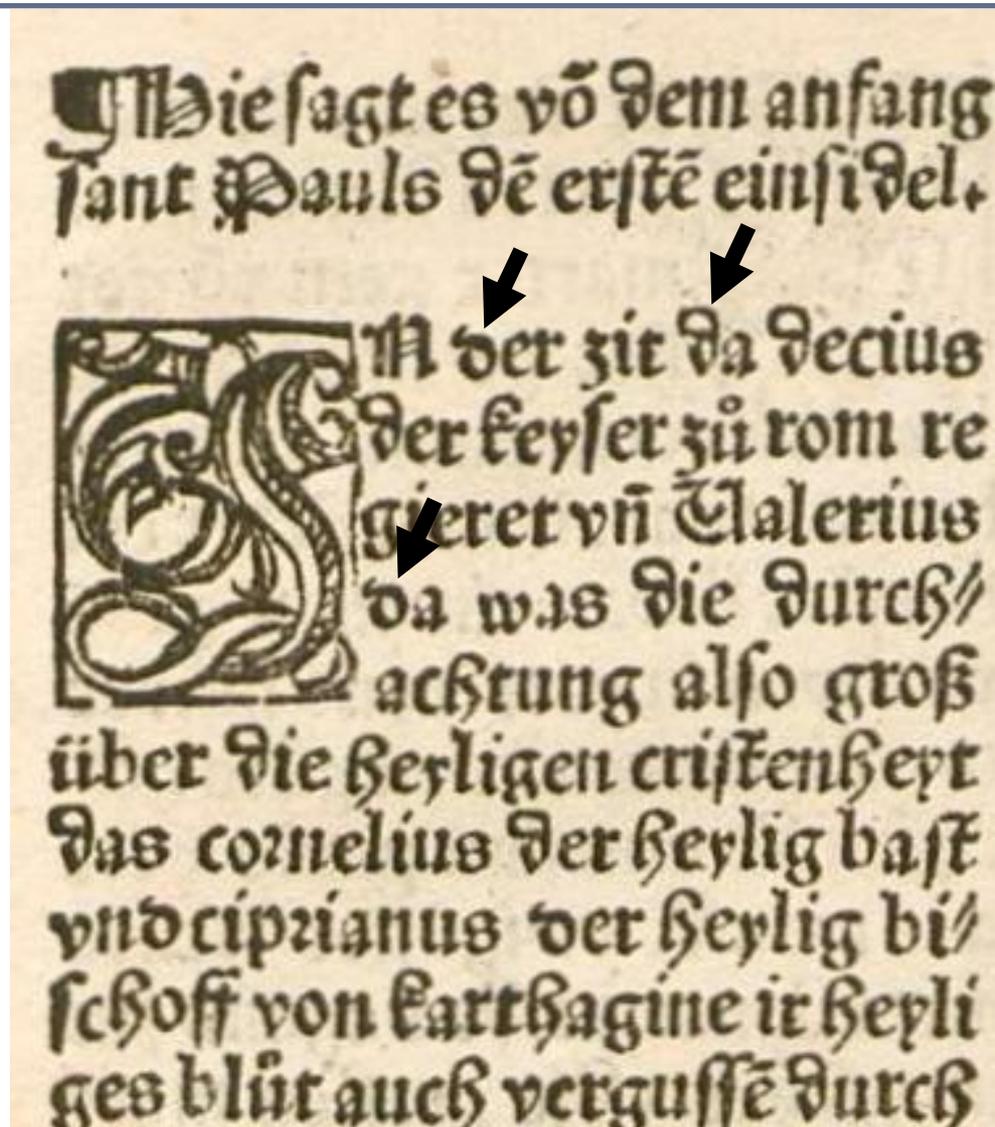
Das Element `<charDecl>`

- Teil von `teiHeader/encodingDesc`
- Enthält `<char>`- und `<glyph>`-Elemente
- Darin u.a.:
 - `<charName>` bzw. `<glyphName>`
 - `<charProp>`
 - `<desc>`
 - `<mapping>`
 - `<figure>`

Ein Beispiel

```

<encodingDesc>
...
<charDecl>
  <glyph xml:id="variantd">
    <glyphName>VARIANT OF LATIN
    SMALL LETTER D</glyphName>
    <desc>rundes kleines d</desc>
    <mapping
    type="standardized">d</mapping>
    <figure>
      <graphic url="variant-
      d.jpg"/>
    </figure>
    <note>nur am Wortanfang
    verwendet, selten</note>
  </glyph>
</charDecl>
...
</encodingDesc>
  
```



Ein Beispiel

```

<TEI>
...
<text>
  <body>
    <p>
      ...
      <lb n="3" />Jn
      <g ref="#variantd">d</g>er
      zit da decius
    </p>
  </body>
</text>
</TEI>
  
```

Beispiel: 007 gaiji.xml

Beispieltransformation: 007 gaiji2html.xsl





Übung

- Ermitteln Sie über die Seite <http://www.fileformat.info/> den Codepoint für die Abkürzung "per" (Unicode-Name: LATIN SMALL LETTER P WITH STROKE THROUGH DESCENDER)
- Finden Sie den entsprechenden Codechart unter www.unicode.org/charts/
- Codieren Sie das Zeichen als Entität: `&#xXXXX;`
- Probieren sie aus, ob ihr PC das darstellen kann
- Ggf. Junicode installieren (<http://junicode.sourceforge.net/>)
- Codieren sie eine charDecl in der das Zeichen definiert wird und korrekt als Unicode-Zeichen aufgelöst wird (oder verändern sie das Beispiel "007 gaiji.xml" dementsprechend)

Fertig

