

Datenkodierung in TEI XML im Rubensohn-Projekt

Daniel A. Werning (Berlin)*

1. Textdaten und Textmetadaten

Bei der editorischen Erschließung der Rubensohn-Sammlung unter der Leitung von Prof. Dr. Verena Lepper sollen Textartefakt-Daten gesammelt, im Internet öffentlich zugänglich gemacht und langfristig gespeichert werden.¹ Diese lassen sich aufgrund nicht zuletzt praktischer Überlegungen (dazu unten) in erster Näherung in drei Gruppen unterteilen: Textdaten, Textmetadaten und Textträger-Bilder. Die Textdaten bestehen aus der Dokumentation der auf den Textträgern geschriebenen Texte selbst, entweder in Transliteration oder in Transkription, ergänzt um Zusatzinformationen zu einzelnen Textteilen, wie die Auszeichnung von Emendationen oder die Verschlagwortung von Ortsnamen u.a.m. Bei den Textmetadaten handelt es sich um Informationen über den Textträger oder den Text selbst als Ganzes(!), z.B. die Textträger-Größe, seine Katalogisierung, das Textthema, den Textschreiber, u.a.m.

Mit dem Ziel der Nachhaltigkeit ist dazu ein Datenformat zu wählen, das langfristig speicherbar, möglichst eindeutig interpretierbar und zugleich möglichst einfach wieder verwendbar ist. Ein Datenkodierungsformat, das diesen Ansprüchen Genüge leistet, ist ein Format, das die Richtlinien der *eXtensible Markup Language* (XML)² befolgt. Da die XML-Richtlinien aber, vereinfacht formuliert, nur den generellen Strukturplan für die Kodierung festlegen, nicht jedoch ihre spezifische Ausgestaltung (dazu unten mehr), ist darüber hinaus ein ganz spezifisches XML-konformes Format zu wählen bzw. zu definieren. Hier kommen grundsätzlich zwei Optionen in Betracht: a) die Definition eines eigenen, projektspezifischen (XML-konformen) Formats oder b) die Verwendung eines bereits von anderen Projekten definierten (XML-konformen) Formats, möglichst eines Formats, das weit verbreitet ist oder zumindest das Potential hat, ein weit verbreitetes Format zu werden. Wünschenswert ist darüber hinaus die gemeinsame Speicherung von zusammengehörigen Textdaten und Textmetadaten in ein und demselben Datensatz. Bereits verwendete und veröffentlichte Kodierungsformate, die diese Bedingungen größtenteils zu erfüllen versprechen, sind z.B. die Formatempfehlungen des *Text Encoding Initiative* Konsortiums³ (kurz: TEI-Format). Im Folgenden werden Vor- und Nachteile dieser Kodierungsoptionen genauer besprochen.

2. XML-Format und Unicode

Das kurz so genannte XML-Format setzt sich in vielen Bereichen der Informationstechnologie zunehmend durch.⁴ Die XML-Format-Richtlinien legen dabei unter anderem insbesondere fest, dass Kategorie//Datum-Paare, wie z.B.

INVENTARNUMMER—„**P. 4302**“ (Textmetadatum)
oder „**Otto Rubensohn**“—NAME (z.B. bei einer Textstellen-Verschlagwortung)
wie folgt kodiert werden sollen:

`<inventarnummer>P. 4302</inventarnummer>`

bzw. `<name>Otto Rubensohn</name>`.

Man spricht hier von ‘Elementen’ mit einem ‘Elementtyp’-Namen, z.B. ‘`inventarnummer`’, und einem ‘Elementinhalt’, z.B. „**P. 4302**“.⁵ Die in spitzen Klammern eingeschlossenen, umklammernden Elementtypnamen, hier `<inventarnummer>...</inventarnummer>` und

* Ich danke an dieser Stelle insbesondere Marius Gerhardt und Paul Bartels (Berliner Papyrusdatenbank, Staatliche Museen zu Berlin – Preußischer Kulturbesitz), Michael Marx und Tobias J. Jocham (Corpus Coranicum, Berlin-Brandenburgische Akademie der Wissenschaften), Alexander Czmiel (Telota, Berlin-Brandenburgische Akademie der Wissenschaften) sowie Tonio Sebastian Richter und Katrin John (Database and Dictionary of Greek Loanwords in Coptic) für ihre Gesprächsbereitschaft und Anregungen.

`<name>...</name>` werden als *markup tags* (dtsch. etwa ‘Auszeichnungsetikett’), kurz Tags, bezeichnet. Optional können solche Tags jeweils noch eine Art von Unterinformation, sog. ‘Attribute’⁶, enthalten, wie z.B. die Zuordnung eines Namens zu einer eindeutigen Personen-Identifikationsnummer in folgendem Beispiel:

```
<name id="#Pers012">Otto Rubensohn</name>
```

Die Zeichenketten, die die Informationen bilden, sollen dabei vollständig im international standardisierten Zeichenformat des sog. *Universal Character Set* kodiert werden, dessen Implementierung unter der Bezeichnung ‘Unicode’ bekannt ist.⁷ Im Unicode-Standard ist (perspektivisch) jedem auf der Welt verwendeten Schriftzeichen eine eindeutige Nummer zugeordnet, die üblicherweise als ‘U+’ plus Nummer im Hexadezimalsystem angegeben wird. So entspricht z.B. der moderne LATIN SMALL LETTER A dem Kode U+0061, das LATIN SMALL LETTER EGYPTOLOGICAL ALEF dem Kode U+A723.⁸ Dabei legt der Standard im Prinzip nur die graphematisch distinktiven Merkmale der Zeichen fest, die genaue graphemische Umsetzung kann der Zeichensatzhersteller in Grenzen modifizieren. So können verschiedene Unicode-konforme Zeichensätze das LATIN SMALL LETTER A beispielsweise in die Graphen a, a, *a* oder **a** umsetzen oder das LATIN SMALL LETTER EGYPTOLOGICAL ALEF in die Graphen ʒ, ʒ oder ʒ — allein die Zuordnung von Kodenummer und dem spezifischen Graphem muss gewährleistet sein. Da Unicode-kodierte Texte einzelne Zeichen somit international standardisiert und zugleich unabhängig von einem spezifischen existierenden Zeichensatz kodieren, bleiben sie perspektivisch langfristig eindeutig dekodierbar. Computer, auf denen ein Zeichensatz installiert ist, der a) dem Unicode-Standard folgt und b) das betreffende Zeichen auch wirklich implementiert hat – nicht alle Unicode-konformen Zeichensätze beinhalten auch notwendigerweise alle im Standard definierten Zeichen – können die entsprechend kodierten Zeichen dann auch korrekt z.B. als ʒ darstellen. Existierende, kostenfreie Zeichensätze, welche die für ägyptologische Belange benötigten Sonderzeichen auch beinhalten, sind z.B. *New Athena Unicode* und *Charis SIL*,⁹ für koptologische Belange ist der Zeichensatz *Antinoou* gut geeignet¹⁰.

3. Spezifische XML-konforme Formate und die Formatempfehlungen der *Text Encoding Initiative* und *EpiDoc*

Als genereller Kodierungsstandard definiert der XML-Standard aber nur den generellen Aufbau einer XML-konformen Datei und insbesondere nicht die Kategorien und Kategorienbezeichnungen selbst. So könnte die Information „Otto Rubensohn“ prinzipiell XML-konform ausgezeichnet werden als z.B.

```
<name>Otto Rubensohn</name>
```

oder `<personname>Otto Rubensohn</personname>`

oder `<personal_name>Otto Rubensohn</personal_name>`

oder `<vorname>Otto</vorname> <nachname>Rubensohn</nachname>`

oder, hierarchisch verschachtelt,

```
<personname>
```

```
  <vorname>Otto</vorname> <nachname>Rubensohn</nachname>
```

```
</personname>
```

oder ähnlich. Alle diese Varianten stellen gültige XML-Kodierungen dar.

Will man Informationen unter verschiedenen Beteiligten effizient austauschen, empfiehlt es sich daher, nicht bloß grundsätzlich auf die Kodierung in irgendeinem beliebigen XML-konformen Format zu setzen, sondern sich auf spezifische gemeinsame Elementtypnamen, z.B. `persName`, zu einigen, sowie auf eine gemeinsame Struktur von Element-Tags, z.B. `<persName><forename>...</forename> <surname>...</surname></persName>`.

Eine solche Benennungs- und Struktur-Empfehlung zur Kodierung von Textdaten und Textmetadaten wird vom *Text Encoding Initiative* Konsortium herausgegeben und weiter entwickelt (kurz: TEI-Format).¹¹ Auch die TEI-Formatempfehlungen stellen den Kodierenden teilweise aber noch mehrere Möglichkeiten der Kodierung bestimmter Informationen zur Wahl. Darauf aufbauend diskutiert und empfiehlt eine Gruppe unter der Bezeichnung *EpiDoc* spezifische TEI-konforme Kodierungsweisen für die elektronische Textdokumentation.¹² Anders als die XML sind die TEI- und *EpiDoc*-Formatempfehlungen aber bislang nicht zu einem offiziellen Standard erhoben worden. Es zeichnet sich aber ab, dass zunehmend mehr Projekte diesen Empfehlung folgen¹³, diese unterstützen (z.B. *Papyri.info*; die *Scalable Architecture for Digital Editions* der BBAW)¹⁴ bzw. vorhaben, ihre Datenbestände entsprechend umzukodieren (*Corpus Coranicum*, *Thesaurus Linguae Aegyptiae*)¹⁵. Wir setzen darauf, dass das TEI-Format in Zukunft immer stärker Verwendung findet und sich damit langfristig Vorteile für die dauerhafte Verstehbarkeit und Wiederverwendbarkeit der entsprechend kodierten Daten ergeben.

4. Vor- und Nachteile der Verwendung des TEI-Formats

Vorteile der Bereitstellung der Rubensohn-Daten im TEI-Format wurden bereits oben angesprochen, darunter insbesondere die langfristige, eindeutige Lesbarkeit (XML-konform, Unicode), die gemeinsame Speicherung von Textdaten und Textmetadaten in einem gemeinsamen Datensatz und die nachhaltigen Zukunftsperspektiven, die die Verwendung und somit auch Unterstützung eines aufstrebenden Kodierungsstandards in Aussicht stellen.

Weitere Vorteile ergeben sich aus den Auszeichnungsmöglichkeiten, die das TEI-Format für den Text selbst zur Verfügung stellt. So lassen sich im Text vorkommende Wörter und Phrasen leicht und ggf. normiert verschlagworten, z.B.

**tw = n ir ʿnh m-b}h pr-ʿ <persName key="Ptolemaios III.">Ptrwmys</persName>
s} Ptrwmys irm }rsn**

‘Wir machen einen Eid vor dem König Ptolemaios, Sohn des Ptolemaios und der Arsinoe’.

Auch lassen sich sowohl antike als auch moderne Texteingriffe und -kommentierungen, wie z.B. Emendationen oder Zerstörungsmarkierungen, expliziter und somit eindeutiger auszuzeichnen, als es die teils fachspezifisch-idiosynkratischen Editionsmarkierungen vermögen. So können beispielsweise die geschweiften Klammern in Transkriptionen hieroglyphisch-ägyptischer Texte bekanntlich verschiedene Bedeutung haben:

„Geschweifte Klammern umschließen im Original versehentlich [z.B. *jmn{.t}*, D.W.] **oder** aus bloßer eugraphischer Tradition [z.B. *zw{r}j*, D.W.] zuviel Geschriebenes“.¹⁶

Diese Praxis dürfte für fachfremde LeserInnen kaum ohne zusätzliche, an anderer Stelle abgelegte Erläuterungen durchschaubar sein. Umgekehrt können nach dem in der altphilologischen Papyrologie verbreiteten sog. Leidener Klammersystem¹⁷ ausgezeichnete Texte Markierungen enthalten, die auf dem Hintergrund der ägyptologischen Praxis zu Missverständnissen führen könnten (z.B. Punkte unter uneindeutig identifizierbaren Zeichen). Das TEI-Format bietet hier die Möglichkeit, solche Auszeichnungen expliziter zu gestalten und genauer zu unterscheiden, z.B.

– Fälschlich zuviel Geschriebenes:

Traditionell: *jmn{.t}*

TEI/EpiDoc: **jmn<surplus reason="Fehler">t</surplus>**

– Traditionelle, ‘überflüssige’ Schreibung:

Traditionell: *zw{r}j*

Vorschlag in TEI: **zw<surplus reason="Traditionsschreibung">r</surplus>j**

– Fälschlich ausgelassen:

Traditionell: *jmn*<.w>

TEI/EpiDoc: **jmn**<supplied reason="Auslassung">.w</supplied>

– Idealisiert rekonstruierte Morphologie:

Traditionell: *sdm.t(w)=f*

Vorschlag in TEI: **sdm.t**<supplied reason="Idealisierung">w</supplied> = f

– Ersetzungsfehler:

Traditionell: *jmn.{t}<w>*

TEI/EpiDoc: **jmn.**<choice><sic>t</sic><corr>w</corr></choice>

– Auslassung (bestimmter Länge):

Traditionell: *s3 [///] irm 3rsn*

TEI/EpiDoc: **s3** <gap reason="Zerstörung" quantity="4" unit="Quadrate" /> **irm**
3rsn

– Hinzufügung über der Zeile:

Leidener Kl.-syst.: ἀνοκ ᾽παπας ᾽ † μαρτηρος

TEI/EpiDoc: ἀνοκ <add place="supralinear">παπας</add> † μαρτηρος

Mittels einer sog. XSL-Transformation (kurz für *eXtensible Stylesheet Language Transformation*)¹⁸ können so kodierte Daten dann zum Schluss optional wieder in vergleichsweise gewohnter Weise am Computer dargestellt werden, z.B. als „**jmn{.t}**“ bzw. „**s3 [...]** **irm 3rsn**“. Wenn gewünscht, könnte man dem Leser/der Leserin sogar mehrere Darstellungsformen zur Auswahl anbieten, die verschiedenen Lesegewohnheiten Rechnung tragen, z.B. „**s3 [...4...]** **irm 3rsn**“ oder „**s3 [///...///]** **irm 3rsn**“. Im Hintergrund stehen aber jeweils die präziseren Informationen im TEI-Format.

Den Vorteilen der Nutzung des TEI-Formats stehen allerdings auch einige Nachteile gegenüber, was bestimmte Informationskodierungen betrifft.

So ist die Kodierung von editorischen Textmarkierungen im TEI-Format mit einem größeren Aufwand verbunden als die Kodierung mit den traditionellen Klammersystemen (siehe oben). Zwar könnten einige Tags halbautomatisch mittels eines Skripts aus einem im traditionellen Klammersystem kodierten Text generiert werden (z.B. <note>-Tags aus Fußnoten, <supplied>-Tags aus runden Klammern), in bestimmten Fällen sind die Kodierenden aber angehalten, händisch mehr Informationen zu kodieren als traditionell üblich (z.B. Unterscheidung der Verwendungen der geschweiften Klammern, genaue Zerstörungslängen, u.a.m.).

Und was die Textmetadaten anbetrifft, so beinhaltet die TEI-Formatempfehlung nur suboptimale Mittel z.B. Materialfarben-Informationen zu kodieren, b) Örtlichkeiten unterhalb einer Siedlungsebene, z.B. „Haus Nr. 22“, prägnant auszuzeichnen oder c) Informationen zum Manuskript-Schreiber genauso detailliert zu kodieren, wie solche zum Text-Autor. Lösungen sind in allen diesen und weiteren ‘Problemfällen’ jeweils möglich, führen aber im Einzelfall zu vergleichsweise unintuitiven Kodierungsweisen.

Das Rubensohn-Projekt hat sich nichtsdestotrotz gegen die Definition eines eigenen, möglicherweise fachintern intuitiveren, aber idiosynkratischen (XML-konformen) Formats und für Bereitstellung der Daten im TEI-Format entschieden, wobei im Falle der Textmetadaten prozessbedingt sowieso parallel XML-konforme Datensätze anfallen (siehe dazu im Folgenden), in denen z.B. die Materialfarbe dann intuitiver kodiert sind.

5. Konsequenzen der Verwendung des TEI-Formats

Die bewusste Entscheidung für das Zielformat TEI hat nun aber auch vereinzelt Konsequenzen für die Art und Weise der Datenerhebung. Im Folgenden seien drei Beispiele genannt:

Das TEI-Format legt es nahe, eine Ortsangabe nicht bloß als `<location>`, sondern spezifisch als `<settlement>`, `<region>` oder `<country>` auszuzeichnen. Da dieses Verfahren auch zusätzliche Vorteile für die Auswertung der entsprechenden Datensätze verspricht, hat sich das Projekt entschieden, die Informationen zu Fundort, Erwerbort und Abfassungsort jeweils nicht nur als eine unspezifische ÖRTLICHKEIT-Information zu erheben, die je nach Wissensstand wahlweise aus einer Siedlungsbezeichnung, aus einer Bezirksbezeichnung oder nur aus einem Ländernamen besteht, sondern es werden pro Örtlichkeit jeweils alle drei Informationen, d.h. sowohl SIEDLUNG, als auch BEZIRK und LAND, erhoben, die dann später auf die entsprechenden Tags abgebildet werden.

In einigen wenigen Fällen werden mögliche Werte für bestimmte Informationen durch das TEI-Format in gewisser Weise vorbestimmt. Beispielsweise liegt es nahe, den Text-Erhaltungsumfang im Attribut `defective="..."` des Tags `<mslitem>` (*manuscript item*) abzulegen. Das TEI-Format legt aber fest, dass hier nicht etwa ein beliebiger Fließtext mit ungefähren Anteilsangaben, z.B. "ca. 20%", angegeben werden kann, sondern nur entweder `<mslitem defective="true">` oder `<mslitem defective="false">`. Daher wird der TEXTERHALTUNGSUMFANG vorausseilend nicht als Freitext-Information erhoben, sondern nur als „vollständig“ oder „unvollständig“.

Einfachere bzw. andere Lösungsmöglichkeiten zur Kodierung entsprechender Informationen im TEI-Format wären jeweils möglich. Da der Entschluss für das TEI-Format ja aber mit dem Ziel der Datenportierbarkeit über Projekte hinweg erfolgt ist, ist es folgerichtig, in solchen Fällen keine idiosynkratischen Lösungen zu suchen, sondern der von der *Text Encoding Initiative* vorgesehenen Kodierungslogik so weit es inhaltlich vertretbar ist zu folgen – auch wenn dies in Einzelfällen zu kodierungspraktischem Mehraufwand (sowohl `<settlement>`, als auch `<region>` und `<country>`; Explizierung der `<surplus>`-Tags) oder inhaltliche Restriktion (`defective="true"` / `defective="false"`) impliziert.

6. Praktische Umsetzung

Aus praktischen Erwägungen erfolgt die Datensammlung und -kodierung zunächst in drei getrennten Schritten:

(1a) Die Textmetadaten werden mit Hilfe einer projektspezifischen *FileMaker*-Datenbank gesammelt. *FileMaker* lässt es zu, die Datenbank als Sammlung (hierarchisch flacher) XML-konformer Datensätze zu exportieren. Die fertigen projektspezifischen XML-Datensätze werden dann mittels einer *eXtensible Stylesheet Language Transformation* (XSLT)¹⁹ automatisch in einen TEI-Datensatz-Header übersetzt. Die Datenbank und ein entsprechendes XSLT-Skript wurden bereits von Daniel Werning entwickelt.

(1b) Die Bilddateien der Textträger der Rubensohn-Sammlung werden voraussichtlich nach einem im Rahmen des *Berliner Papyrusdatenbank*-Projekts der Stiftung Preußischer Kulturbesitz etablierten Verfahren gewonnen.²⁰

(1c) Die Textdaten selbst sollen aus praktisch-ökonomischen Erwägungen zunächst weitgehend traditionell in Textverarbeitungsprogrammen erfasst werden. Hieroglyphisch-Ägyptische Texte werden dabei nicht transliteriert, sondern in ägyptologischer Transkription wiedergegeben. Im Zuge der Kodierung können einige der Tags gleich von dem/der Kodierenden eingegeben werden, z.B. `<persName>`, andere können aber später halbautomatisch mit Hilfe eines Skripts aus gewohnten Kodierungsweisen heraus generiert werden (z.B. `<note>`, `<supplied>`, `<choice>`).

(2) Eine Referenz auf die in (1b) gewonnenen Bilddateien werden sekundär in die Metatextdatenbank (1a) eingepflegt. Die in (1c) generierten TEI-konformen Textkodierungen werden mit den in (1a) generierten TEI-Headern kombiniert.

(3) Schlussendlich werden die TEI-Datensätze im Internet durchsuchbar bereitgestellt, wobei die optische Ausgabe mittels eines noch zu programmierenden XSLT-Skripts an gewohnte Lesegewohnheiten angepasst werden wird.

Literaturhinweise

EpiDoc: Epigraphic Documents in TEI XML <<http://epidoc.sourceforge.net/>>, hrsg. von der EpiDoc Community (Zugriff: 27. Okt.–10. Nov. 2012).

EpiDoc Guidelines, Version 8-beta (Update: 21. Jan. 2013) <<http://www.stoa.org/epidoc/gl/>>, [hrsg. vom Stoa Consortium] (Zugriff: 27. Febr. 2013).

Lepper, Verena M. 2012. Die ägyptische und orientalische „Rubensohn-Bibliothek“ von Elephantine. 4000 Jahre Kulturgeschichte einer altägyptischen Insel, in: Verena M. Lepper (Hrsg.), *Forschung in der Papyrussammlung. Eine Festgabe für das Neue Museum*, Ägyptische und Orientalische Papyri und Handschriften des Ägyptischen Museums und Papyrussammlung Berlin 1, Berlin: Akademie Verlag, S. 497–508.

TEI P5: Guidelines for Electronic Text Encoding and Interchange, Version 2.2.0 (Update: 25. Okt. 2012) / Version 2.3.0 (Update: 17. Jan. 2013) <<http://www.tei-c.org/Guidelines/P5/>>, hrsg. vom TEI Consortium (Zugriff: 27. Okt. 2012–10. Nov. 2012, 27. Jan. 2013).

Vonhoegen, Helmut. ⁶2011. *Einstieg in XML. Grundlagen, Praxis, Referenz*, 6., aktualisierte und verb. Auflage, Bonn: Galileo.

Weitere Literaturangaben sind in verschiedenen Endnoten genannt.

1 Lepper (2012).

2 ‘Extensible Markup Language (XML)’ <<http://www.w3.org/XML/>>, hrsg. vom World Wide Web Consortium (W3C). Eine gute Einführung in XML bietet Vonhoegen (⁶2011).

3 ‘TEI Guidelines [for Electronic Text Encoding and Interchange]’ <<http://www.tei-c.org/Guidelines/>>, hrsg. vom Text Encoding Initiative Consortium.

4 Vonhoegen (⁶2011: Kap. 1.5 und 13).

5 Vonhoegen (⁶2011: Kap. 2.1.8 und 2.1.11).

6 Vonhoegen (⁶2011: Kap. 2.1.13 und 2.3).

7 Vonhoegen (⁶2011: Kap. 2.1.5 und 2.5.2); ‘The Unicode Standard’ <<http://www.unicode.org/standard/standard.html>>, hrsg. vom Unicode Consortium.

8 Siehe ‘Code Charts’ <<http://www.unicode.org/charts/>> oder ‘Unicode’ <<http://www.fileformat.info/info/unicode/>>.

9 Einen Überblick über die für ägyptologische Belange benötigten Sonderzeichen gibt Werning, ‘Egyptological transliteration in Unicode’ <http://www.archaeologie.hu-berlin.de/aegy_anoa/utilities/index_html/egyptological_transliteration_unicode> (Januar 2013) und ders. in der Tabelle ‘Transkriptionssonderzeichen in Unicode’ in: Wikipedia (<http://de.wikipedia.org/wiki/%C3%84gyptische_Hieroglyphen#In_der_elektronischen_Datenverarbeitung>). New Athena Unicode <<http://apagreekkeys.org/NAUdownload.html>>, hrsg. von der American Philological Association; Charis SIL <<http://scripts.sil.org/CharisSILfont>>, hrsg. vom SIL International.

10 ‘Antinoou: A standard font for Coptic’ <<http://www.evertype.com/fonts/coptic/>>.

11 Siehe En. 3.

12 ‘EpiDoc Guidelines’ <<http://www.stoa.org/epidoc/gl/>>, hrsg. vom Stoa Consortium (Zugriff: 27. Febr. 2013, version 8-beta).

13 Siehe ‘Projects Using the TEI’ <<http://www.tei-c.org/Activities/Projects/>>; zu EpiDoc siehe <<http://www.stoa.org/epidoc/gl/dev/app-bibliography.html>> und <<http://wiki.digitalclassicist.org/Category:EpiDoc>>.

14 ‘Papyri.info’ <<http://www.papyri.info/>>; SADE Documentation <<http://www.bbaw.de/telota/software/sade/documentation/documentation>>.

15 ‘Corpus Coranicum’ <<http://koran.bbaw.de/>> (Michael Marx und Tobias Jocham, pers. Mitteilung), ‘Thesaurus Linguae Aegyptiae’ <<http://aew.bbaw.de/tla/>> (Stephan Seidlmayer, pers. Mitteilung).

16 Wolfgang Schenkel. ⁵2012. *Tübinger Einführung in die klassisch-ägyptische Sprache und Schrift*, 5., [grüne], veränderte Ausgabe, Tübingen: [Wolfgang Schenkel], Kap. 2.5 (Hervorhebung D.W.).

17 Siehe Hans Krummrey & Silvio Panciera. 1980. Criteri di edizione e segni diacritici, in: *Miscellanea*, Tituli 2, Rom: Ed. di Storia e Letteratura, S. 205–215 (*non vidi*); Wikipedia, ‘Leiden Conventions’ <http://en.wikipedia.org/wiki/Leiden_Conventions> (letzter Zugriff: 27. Febr. 2013)

18 Vonhoegen (⁶2011: Kap. 7).

19 Siehe En. 18.

20 ‘Berliner Papyrusdatenbank’ <<http://smb.museum/berlpap/>>.