Arbeiten mit XML in einem XML-Editor

Warum ein XML-Editor?

Ein XML-Dokument lässt sich natürlich in einem beliebigen Texteditor erstellen und bearbeiten. Für die Erstellung dieses Manuals wurde der Editor jEdit verwendet. Es handelt sich hierbei um einen relativ einfachen Editor, das XML erst über einige Plugins erlernen muss. Eine detaillierte Anleitung zur Arbeit mit XML in jEdit findet sich hier. jEdit reicht für den Anfang völlig aus und ist außerdem kostenlos im Internet verfügbar. Für FU-Angehörige steht der komplexere Editor oXygen zum Download zur Verfügung (Software-Seite des Zedat).

Ein XML-Editor macht aus verschiedenen Gründen Sinn. So vermag es XML-kompatible Software, XML-Strukturen zu erkennen und farblich/ durch Einschübe hervorzuheben. Das folgende Beispiel zeigt die Darstellung eines XML-kodierten Textes in jEdit:

```
</Turn>
<Turn speaker="spk1" startTime="443.67" endTime="447.191">
<Sync time="443.67"/>
que j'ai communiquée si vous voulez
<Sync time="445.221"/>
```

Ein XML-Editor kann in der Regel jedoch mehr, als nur XML-Strukturen hervorzuheben. Er kann das Dokument auch auf Wohlgeformtheit prüfen. Überschneiden sich Elemente zum Beispiel oder fehlt das Endtag eines Anfangstags, so sollte das von dem XML-Editor erkannt werden. Mächtigere Editoren können noch mehr. So ist der Editor Oxygen TEI-kompatibel und erlaubt eine Übersetzung vom XML zu einer WYSIWYG ("What you see is what you get")-Version, die die kodierten Formatangaben berücksichtigt.

Nicht nur wohlgeformt, sondern auch valide

Ein XML-Dokument ist wohlgeformt, wenn seine Syntax im XML-Sinn keine Fehler aufweist, so weit nichts Neues. Neben dem Kriterium der Wohlgeformtheit existiert jedoch auch das der Validität. Damit ein Dokument valide sein kann, muss es zunächst auch wohlgeformt sein. Die übrigen Validitätsvoraussetzungen werden in der Dokumenttypdeklaration im Kopf des Dokuments definiert:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Trans SYSTEM "trans-14.dtd">
```

Die Dokumenttypdeklaration wird durch *<!DOCTYPE* eingeleitet, worauf der Name des Wurzelelements folgt. Zuletzt erscheint die Quelle der Validitätsdefinitionen. Dies kann eine Datei, ein Pfad oder eine Internetadresse sein. Auch können die Definitionen im Dokument selbst gegeben werden. Das "Wie?" lässt sich nur sehr umfangreich beschreiben und würde in Hinblick auf den Zweck des vorliegenden Manuals zu weit gehen, weshalb es hier ausgeklammert wird.

Kurz gesagt wird in der Dokumenttyp-Deklaration beschrieben, was in dem Dokument erlaubt ist (z.B. welche Tags, welche Attribute, etc.). Sie wird von XML-kompatiblen Tools und Programmen (z.B. Transcriber) erzeugt. In aller Kürze: Ein XML-Dokument ist valide, wenn es zum einen wohlgeformt ist, zum anderen die Vorgaben der Dokumenttypdeklaration erfüllt.

Deskriptiv vs. prozedural

Speziell bei der Arbeit man mit Textdaten unterscheidet man zwischen deskriptiven und prozeduralen Markups. Prozedurale Tags sind als Hinweise zur Darstellung und Formatierung des Textes zu verstehen und werden von einigen Editoren als solche erkannt und angewendet. Von deskriptiven Auszeichnungen ist dagegen die Rede, wenn die entsprechenden Tags Textbausteine mit formatunabhäniger Information auszeichnen. Allerdings können auch deskriptive Tags indirekt auf ein Format beziehen. Wird ein Baustein z.B. als "Überschrift" markiert, so führt dies auch Informationen über ein abweichendes Format mit sich. Allerdingst weist das deskriptive Tag nicht an, den ausgezeichneten Text in ein Überschriftenformat zu bringen.

Ein Beispiel zur Unterscheidung:

Der Rohtext "zucchero ist eine Entlehnung aus dem Arabischen " lässt sich je nach Sinn und Zweck der Annotation mit unterschiedlichen deskriptiven Auszeichnungen versehen. Zum Beispiel kann die Sprache annotiert werden:

> <italienisch> zucchero </italienisch> ist eine Entlehnung aus dem Arabischen <arabisch> </arabisch>.

Der gesamte Satz lässt sich um ein <satz>-Tag erweitern (u.s.w.):

> <satz><italienisch> zucchero </italienisch> ist eine Entlehnung aus dem Arabischen <arabisch> </arabisch></satz>.

Mit diesen Tags werden Textbausteine lediglich beschrieben, sie sind deshalb deskriptiver Natur. Prozedurale Markups führen hingegen zu einer Veränderung in der Darstellung des Textes, z.B. in:

> <kursiv> zucchero </kursiv> ist eine Entlehnung aus dem Arabischen <kursiv> </kursiv>. (mit dem Ergebnis: *zucchero* ist eine Entlehnung aus dem Arabischen .)

Nun ist XML an sich eine deskriptive Auszeichnungssprache. Dies bedeutet, dass auch Tags wie *<kursiv>* zu keiner Formatierung des Textes führen und für einen einfachen Editor genauso nichtssagend sind wie *<italienisch>* und *<satz>*. Allerdings sind einige fähigere (und folglich auch weniger kostengünstige) Editoren, wie oben angedeutet, mit den Standards der TEI (Text Encoding Initiative) kompatibel. Sind Texte im TEI-Format annotiert, können diese Editoren vorab definierte prozedurale Angaben erkennen und den Text entsprechend formatieren.

Alles klar?

I. Ordnen Sie die folgenden Tags in die Kategorien "prozedural"/ "deskriptiv" zu!

<fett> ... </fett>

<italienisch>...</italienisch>

<umbruch/ >

<eigenname> ... </eigenname>

Lösung:

prozedural: <fett> ... </fett>, <umbruch/ >

deskriptiv: <italienisch>...</italienisch>, <eigenname> ... </eigenname>

Die Macht der Zeichen

Die Tag-Struktur der XML-Kodierung stützt sich auf Zeichen, die als XML-Typisch definiert sind. Tags sind durch spitze Klammern markiert, Attribute durch Anführungzeichen. Betrachten wir noch einmal den Kopf und ein beliebiges Element eines XML-Dokuments:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Trans SYSTEM "trans-14.dtd">

<Speaker id="spk2" name="speaker#2" check="no" dialect="native" accent="" scope="local"/>
```

Welche Zeichen tragen hier die XML-Struktur des Dokuments? Bereits im Kopf erkennen wir, dass *<*, *?*, *=*, *"*, *>* und */* XML-relevant sind. Im *Speaker*-Tag befindet sich außerdem noch das Zeichen */*. Wie verwendet man diese Zeichen jedoch, wenn sie nichts mit der XML-Struktur zu tun haben? Zum Beispiel wenn man direkte Rede in Anführungszeichen setzen will? Einem herkömmlichen Texteditor wäre es egal, denn für ihn sind die Daten eben nur Text. Ein XML-Editor bemüht sich hingegen um eine Interpretation dieses Texts als XML-Dokument. Er würde die direkte Rede als Attribut auffassen und eine Fehlermeldung ausgeben, da sich dieses in keinem Tag befindet. Diese Zeichen müssen in einem XML-Dokument mithilfe der Form *&Kürzel;* (was auch *&* zu einem XML-spezifischen Zeichen werden lässt) umschrieben werden. Dies ist für folgende Zeichen notwerndig:

Zeichen	Umschreibung
<	<
>	>
"	"
'	'
&	&

Und die übrigen XML-Metazeichen? Wie steht es um das Semikolon in *&Kürzel;* ? Für das Parsen der übrigen Zeichen ist entscheidend, dass sie nur in bestimmten Zeichenkombinationen vorkommen. So befindet sich das XML-relevante Fragezeichen ausschließlich direkt nach einer öffnenden spitzen Klammer oder vor einer schließenden spitzen Klammer. Das Ausrufezeichen lässt sich lediglich im Anschluss an eine öffnende spitze Klammer finden, das Gleichheitszeichen nur Anführungsstrichen vorausgehend und der Schrägstrich steht ausschließlich nach einer öffnenden spitzen Klammer oder vor einer schließenden spitzen Klammer (mit oder ohne Leerstelle). Wird eins dieser Zeichen außerhalb der beschriebenen Kombinationen verwendet, wird es als konkretes Zeichen gedeutet. Zudem ist der jeweils andere Bestandteil der Kombination mithilfe der Umschreibungen darstellbar, was zur Folge hat, dass auch die Kombinationen als konkrete Zeichenfolge im Text verwendet werden können (z.B. *<!* als *<!*>). Die Notwendigkeit der Umschreibung liegt auch dann vor, wenn die Zeichen im Inneren eines Attributnamens auftreten.

Ein XML-fähiger Editor...

- erkennt die Syntas eines XML-Dokuments und hebt sie farblich hervor.
- erkennt Fehler und gibt Fehlermeldungen aus.
- kann manchmal auch noch mehr. So können fähigere Editoren prozedurale Markups als solche erkennen und den Text entsprechend formatiert in einer WYSIWYG-Version ausgeben.

Die Dokumenttypdeklaration (DTD)

- definiert über die Regeln der Wohlgeformtheit hinaus, wie die Struktur (Attribute, Elemente, etc., Reihenfolge, Aussehen, etc.) entsprechender XML-Dokumente auszusehen hat. Während die Regeln der Wohlgeformtheit für XML-Dokumente im Allgemeinen gelten, lassen sich mithilfe der DTD weitere und für ausgewählte Dokumente geltende Regeln formulieren.
- stellt sicher, dass das XML-Dokument von bestimmten Tools interpretiert und verarbeitet werden kann.

Metazeichen

- sind Zeichen, die in der XML-Kodierung eine Funktion erfüllen.
- können paraphrasiert werden, wenn sie nicht funktional, sondern konkret interpretiert werden sollen. <, >, *&*, ″und ′werden durch *&Kürzel;* umschrieben. Die übrigen Metazeichen sind aufgrund ihrer kontextuellen Beschränkungen eindeutig und benötigen im Falle einer konkreten Interpretation keine Paraphrase.

Previous Wohlgeformtheit von XML-Dokumenten
Up Datenkodierung und XML
Ein XML-Beispiel aus der Linguistik: Transcriber Next