

# Transcriber: ... zum Kleinen

Befassen wir uns nun mit einigen Funktionen und Details, die bis jetzt noch keine Erwähnung fanden.

## Events und Comments

Transcriber bietet die Möglichkeit, Auffälligkeiten in der Transkription mithilfe von *Event*-Elementen zu kennzeichnen oder zu kommentieren. Auf der Benutzeroberfläche lässt sich das entsprechende Fenster mit dem Tastaturkürzel Strg+d an der Stelle, an der das Event zu platzieren ist, öffnen:

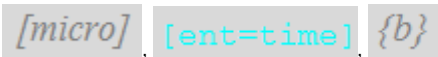
[blocked URL](#)

Zunächst lässt sich hier der *Type* des Events wählen, der dann als ein Attribut-Wert-Paar (*type=""*) des Elements *Event* im XML-Code vermerkt wird. Unter *Description* wird die Auswahl näher präzisiert. Hier kann alternativ aus einer geschlossenen Liste vorhandener Deskriptionen gewählt oder eigenständig formuliert werden. *Extent* erlaubt es schließlich zu bestimmen, auf welches Wort oder welchen Testabschnitt sich das Event bezieht.

Was in diesem Fenster nicht sichtbar wird, ist dass *Comment* gar nicht als *Type* von *Event* agiert, sondern eigentlich ein eigenes Element analog zu *Event* bildet. Ein vergleichender Blick in das XML-Dokument verdeutlicht dies:

```
<Event desc="micro" type="noise" extent="instantaneous"/>
<Event desc="time" type="entities" extent="instantaneous"/>
<Comment desc="b"/>
```

Das Element *Comment* verzichtet auf das Attribut *type* und ihm kann anders als *Event* kein *extent*-Attribut zugeordnet werden. Auf der Benutzeroberfläche lassen sich Kommentare anhand der geschweiften Klammern, die sie umfassen, erkennen, während Events in eckige Klammern gesetzt werden:



Lediglich das Event des Typus *entities* wird farblich abweichend dargestellt. Alle anderen Events und Kommentare lassen sich an der grauen Schrift erkennen.

Für die unterschiedlichen Typen eines Events stehen folgende Deskriptionen zur Verfügung:

Typ, Wert in XML	Darstellung auf der Benutzeroberfläche	Deskription
------------------	--	-------------

Noise "noise"	[b]	[r] respiration [i] inspiration [e] expiration [n] reniflement [pf] souffle
		[bb] bruit de bouche [bg] bruit de gorge [tx] toux, râcllement,éternuement [rire] rires du locuteur [sif] sifflement du locuteur
		[b] bruit indéterminé [conv] conversations de fond [pap] froissement de papiers [shh] souffle électrique [mic] bruits micro rire en fond toux en fond indicatif jingle top musique applaudissements [nontrans] segment
Pronounce "pronounce"	[pron=b]	mal prononcé lapsus [pi] inintelligible [pif] inintelligible/faible [ch] voix chuchotée
		sigle lu sigle épelé 19 cent...
Lexical "lexical"	[lex=b]	orthographe incertaine mot inconnu néologisme rupture de syntaxe

language "language"	[lang=b]	<div>Arabic</div> <div>Bulgarian</div> <div>Chinese</div> <div>Czech</div> <div>Danish</div> <div>Dutch</div> <div>English</div> <div>Finnish</div> <div>French</div> <div>German</div> <div>Greek</div> <div>Hebrew</div> <div>Hindi</div> <div>Hungarian</div> <div>Italian</div> <div>Japanese</div> <div>Korean</div> <div>Latin</div> <div>Norwegian</div> <div>Persian</div> <div>Polish</div> <div>Portuguese</div> <div>Romanian</div> <div>Russian</div> <div>Slavonic</div> <div>Spanish</div> <div>Swedish</div> <div>Turkish</div>
------------------------	----------	--

Named Entities "entities"	[ent=b]	Person pers.hum pers.anim pers.imag Organization org.pol org.edu org.non-profit org.com Geo-Soc-Pol gsp.pers gsp.org gsp.loc Location loc.geo loc.geo.line loc.addr loc.addr.post loc.addr.tel loc.addr.elec Facilities Product prod.vehicule prod.award prod.art prod.printing Time time.date time.date.abs time.date.rel time.hour Amount amount.phy amount.phy.age amount.phy.dur amount.phy.temp amount.phy.len amount.phy.wei amount.phy.spd amount.cur Unknown prod.award/pers.hum org.edu/gsp.pers
------------------------------	---------	---

Die Deskriptionen werden entweder in Form eines Buchstabenkürzel (*i* für *inspiration*, *sf* für *souffle*, etc.), als verkürzte Form (*pers* für Person), vollständig ausformuliert (*rire au fond*, *19 cent...*, etc.) oder als sonstige Zeichen (^ für *mot inconnu*, ? für *orthographe incertaine*) als Wert des entsprechenden Attributs verwendet.

Wie bereits angedeutet, kann den Events darüber hinaus ein Bezugspunkt oder Bezugsraum im Text zugeordnet werden. Dieser wird im XML-Code im Attribut-Wert-Paar *extent=""* vermerkt. Folgende Optionen stehen zur Auswahl:

Auswahl Benutzeroberfläche	Darstellung Benutzeroberfläche	Wert XML
Instantaneous event	[b]	(Attribut <i>extent</i> entfällt)
Start of event	[b-]	"begin"
End of Event	[-b]	"end"
Apply to previous word	+ [b]	"previous"
Apply to next word	[b] +	"next"
(Apply to selection)	([b-][-b])	("begin" im ersten Event-Element, "end" im zweiten)

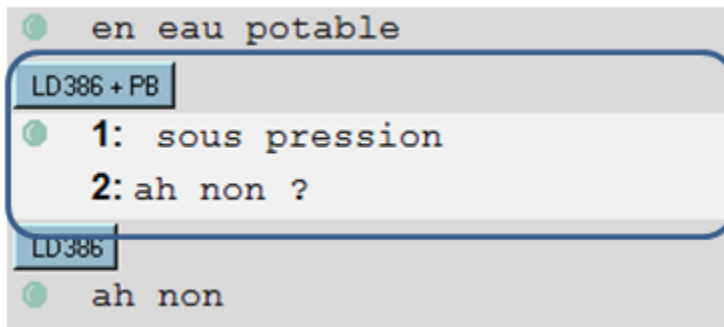
Trotz der Möglichkeit, ein Event mithilfe der Werte "*begin*" und "*end*" auf längere Textsegmente zu beziehen, sei darauf hingewiesen, dass Events punktuelle Ereignisse sind, die sich genau dort im Text situieren, wo das Event eingefügt wird. Deshalb ist es auch möglich, ein Event nur beginnen, aber nicht enden oder nur enden, aber nicht beginnen zu lassen.

## Overlapping speech

In der Regel verläuft ein Gespräch nicht immer ohne die Überlappung der einzelnen Gesprächsbeiträge. Für solche Fälle bietet Transcriber die Möglichkeit, Gesprächssegmente bei den Turnattributen (Strg+Alt+) als *overlapping speech* zwei Gesprächsteilnehmer zu kennzeichnen:

The screenshot shows the 'Edit turn attributes' dialog box. The 'Overlapping speech' checkbox is checked. Under the 'Choose first speaker' radio button, a list of speakers is shown with 'LD 386' selected. The 'External Speakers Database' is empty. Other fields include 'Name' (LD 386), 'spelling checked' and 'global name' checkboxes, 'Type' dropdown, 'Dialect' (native), 'Mode', 'Fidelity', and 'Channel' dropdowns. At the bottom are 'OK', 'Destroy', and 'Cancel' buttons.

Ist *Overlapping speech* markiert, lassen sich nach Anklicken von *Choose first speaker* und *Choose second speaker* die beteiligten Sprecher auswählen. Im Hauptfenster der Benutzeroberfläche führt dies zu folgender Darstellung:



Folgende Entsprechung lässt sich im XML-Code finden:

```

<Turn speaker="spk1 spk2" startTime="716.3" endTime="716.944">
<Sync time="716.3"/>
<Who nb="1"/>
    sous pression
<Who nb="2"/>
    ah non ?
</Turn>
<Turn speaker="spk1" startTime="716.944" endTime="717.76">
<Sync time="716.944"/>
    ah non
</Turn>

```

Die Struktur ähnelt im Ansatz der des gewohnten Turns, dem nur ein einziger Sprecher zugeordnet ist. Allerdings unterscheidet er sich von diesem in zwei Punkten. Zum einen besteht der Wert des *speaker*-Attributs nun aus zwei durch eine Leerstelle getrennten Sprecherkodierungen. Diese Nummer des Sprechers spielt bei der Reihenfolge keine Rolle. Ausschlaggebend ist, welcher Sprecher unter *Choose first speaker* und *Choose second speaker* ausgewählt wurde. Der zweite Unterschied liegt im Vorhandensein des Elements *<Who>*. *<Who>* trennt die Sprecher einer sich überlappenden Rede. Der das Attribut *nb* des ersten Sprechers den Wert "1", dem des zweiten Sprechers wird der Wert "2" zugeordnet. Wird ein neuer Breakpoint gesetzt, beginnt der Wert wieder bei "1". Dies wird fortgesetzt, bis das Ende des Turns erreicht ist.

```

<Turn speaker="spk1 spk2" startTime="716.3" endTime="716.944">
<Sync time="716.3"/>
<Who nb="1"/>
    sous pression
<Who nb="2"/>
    ah non ?
<Sync time="716.519"/>
<Who nb="1"/>
    oui
<Who nb="2"/>
    non
</Turn>

```

## Die Reihenfolge von Attributen

Wer bereits mit HTML gearbeitet hat, weiß, dass die Reihenfolge von Attributen nicht relevant ist. Ob das Attribut zur Zuweisung der Schriftfarbe oder das für die Festlegung der Schriftgröße dem jeweils anderen vorausgeht, spielt für den Browser keine Rolle. Für Transcriber verhält es sich ähnlich. Obwohl die Attribute stets in derselben Reihenfolge vom Programm gespeichert werden, vermag es sie auch dann zu interpretieren, wenn man im XML-Code die Reihenfolge manuell vertauscht. Warum sollte man sich also mit der Reihenfolge beschäftigen?

Der Sinn mag sich momentan noch nicht erschließen, er kann jedoch relevant werden, will man die Daten verwerten, z.B. mithilfe von Regulären Ausdrücken durchsuchen oder den Output von Transcriber zu einem Input für andere Programme (wie Praat) formatieren. Deshalb soll an dieser Stelle ein Überblick über die Reihenfolge der Attribute der einzelnen Elemente gegeben werden. Eingeklammerte Attribute entfallen, wenn kein Wert definiert ist.

Element	Attributreihenfolge
Comment	desc

Event	desc > type > extent
Section	type > startTime > endTime (> topic)
Speaker	id > name > check > dialect > accent > scope
Sync	time
Trans	scribe > audio_filename > version > version_data
Topic	id > desc
Turn	(speaker >) startTime > endTime (> mode) (> fidelity) (> channel)
Who	nb

In genau einem Fall weicht die Reihenfolge von der beschriebenen ab, nämlich im allerersten Turn des Episode-Elements. Hier bildet *speaker* das letzte Attribut der Reihe:

Element	Attributreihenfolge
Turn (erster Turn des Dokuments/ des <i>Episode</i> -Elements)	startTime > endTime (> mode) (> fidelity) (> channel) (> speaker)

 Unbekanntes Makro: 'scrollbar'