

Dissertation  
zur Erlangung des akademischen Grades  
Doctor philosophiae (Dr. phil.)

**Werkzeuge zur flachen Annotation  
von Transkriptionen gesprochener Sprache**

Eingereicht an der  
Fakultät für Linguistik und Literaturwissenschaft  
– Universität Bielefeld –

von Kai Wörner aus Wetzlar  
im Mai 2009

Gutachter: Dr. Andreas Witt  
Prof. Dr. Dieter Metzger

# Inhalt

<b>1. Einleitung</b>	<b>5</b>
Annotation Science vs. Annotation	6
<b>1.1. Begriffsdefinitionen</b>	<b>8</b>
<b>1.2. Gliederung der Arbeit</b>	<b>10</b>
<b>1.3. Anmerkungen zur Form</b>	<b>12</b>
<b>2. Modellierung gesprochener Sprache</b>	<b>13</b>
<b>2.1. Gesprochene vs. geschriebene Sprache</b>	<b>13</b>
2.1.1. Was ist gesprochene Sprache?	13
2.1.2. Besonderheiten der Mündlichkeit	18
<b>2.2. Transkription</b>	<b>31</b>
2.2.1. Anwendungsbereiche von Transkription	31
2.2.2. Grundlagen der Transkription in Gesprächsforschung und Diskursanalyse	37
<b>2.3. Problemfälle für die Modellierung von Sprachdaten</b>	<b>43</b>
2.3.1. Überlappende Hierarchien	43
2.3.2. Zeitbezug	44
2.3.3. Überlappende Annotationen	45
2.3.4. Lautung vs. Schreibung	45
2.3.5. Segmentierung	48
2.3.6. Anforderungen an das Datenmodell	54

<b>3. Existierende Datenmodelle</b>	<b>55</b>
<b>3.1. Ordered Hierarchy of Content Objects (OHCO) und hierarchische Baumstrukturen</b>	<b>55</b>
3.1.1. OHCO-Modelle	55
3.1.2. Die TEI-Guidelines	57
3.1.3. Bewertung	64
3.1.4. Exkurs: XCONCUR, XCONCUR-CL	67
3.1.5. Das NITE Object Model (NOM)	71
<b>3.2. Gerichtete, azyklische Graphen</b>	<b>80</b>
3.2.1. Annotation Graphs	82
3.2.2. Transkriptionsgraphen (EXMARaLDA)	88
3.2.3. Linguistic Annotation Framework	101
<b>3.3. Zusammenfassung</b>	<b>108</b>
3.3.1. Verarbeitbarkeit von Datenmodellen	109
3.3.2. Voraussetzungen für die Etablierung eines Standards	112
3.3.3. Auswahl eines Modells	115
<b>4. Praktische Fragen zur Annotation</b>	<b>117</b>
<b>4.1. Bestehende Annotationsmöglichkeiten des EXMARaLDA-Systems</b>	<b>117</b>
4.1.1. Annotationsspuren	117
4.1.2. Annotation von Konkordanzen	123
4.1.3. Annotation in anderen Werkzeugen	128
<b>4.2. Annotationsmöglichkeiten in existierenden Werkzeugen</b>	<b>130</b>
4.2.1. Werkzeuge zur automatischen Annotation von Texten	130
4.2.2. Werkzeuge zur manuellen Annotation schriftlicher Texte	133
4.2.3. Anwendbarkeit von Werkzeugen für Texte auf Transkriptionen	136
4.2.4. Transkriptionswerkzeuge mit Annotationsmöglichkeiten	137
<b>4.3. Desiderata für Annotationen im EXMARaLDA-System</b>	<b>141</b>

<b>5. Ein Standoff-Annotationsverfahren für Transkriptionsgraphen</b>	<b>143</b>
<b>5.1. Datenmodell und -format</b>	<b>144</b>
5.1.1. Vorüberlegungen	144
5.1.2. Standoff-Annotation mit Feature-Structures	148
5.1.3. Problemfälle	150
<b>5.2. Umsetzung</b>	<b>158</b>
5.2.1. XML-Schema	158
5.2.2. Standoff EXMARaLDA-Transcription Annotation Tool (Sextant)	160
5.2.3. Manuelle Annotation mit dem Z2-Tagger	169
5.2.4. Wünschenswerte Ergänzungen	171
<b>5.3. Zusammenfassung und Ausblick</b>	<b>176</b>
<b>6. Konklusion</b>	<b>178</b>
<b>6.1. Zusammenfassung</b>	<b>178</b>
<b>6.2. Diskussion und Ausblick</b>	<b>185</b>
<b>7. Verzeichnisse</b>	<b>188</b>
<b>7.1. Korpora</b>	<b>188</b>
<b>7.2. Bibliographie</b>	<b>191</b>
<b>7.3. Abbildungsverzeichnis</b>	<b>204</b>
<b>7.4. Index</b>	<b>207</b>

# 1. Einleitung

Diese Arbeit entstand im Rahmen meiner Tätigkeit am Sonderforschungsbereich 538 Mehrsprachigkeit in Hamburg. Dort arbeite ich im Projekt „Computergestützte Erfassungs- und Analysemethoden multilingualer Daten“.

Das Projekt strebt an, auf der Basis aktueller texttechnologischer Forschung methodische und technologische Grundlagen für den Computereinsatz in der Mehrsprachigkeitsforschung zu schaffen. Konkret bedeutet dies, in Zusammenarbeit mit den Teilprojekten des SFBs, die mit Sprachkorpora arbeiten, Software-Werkzeuge zu entwickeln und anzupassen, die der computergestützten Erfassung, Analyse und Archivierung mehrsprachiger Daten dienen.

Ein Ergebnis der Arbeit dort ist EXMARaLDA. EXMARaLDA steht für *Extensible Markup Language for Discourse Annotation*. Es wird meistens als eine Transkriptionssoftware wahrgenommen, ist aber vielmehr ein auf dem Modell der *Transkriptionsgraphen* aufbauendes System von Konzepten, Datenformaten und Werkzeugen für die computergestützte Transkription und Annotation gesprochener Sprache, zu der eben auch ein Transkriptionswerkzeug – der Partitur-Editor – gehört.

Die Werkzeuge des EXMARaLDA-Systems bieten einige Möglichkeiten, Transkriptionen gesprochener Sprache mit Annotationen zu versehen. Diese Möglichkeiten beschränken sich aber auf Segmente in Transkriptionen, die der Transkribend zuvor auf der Zeitachse selbst markiert hat. Diese Segmente – die zudem immer einem Abschnitt transkribierter Sprache eines bestimmten Sprechers zugeordnet sein müssen – lassen sich dann in einer für diesen Zweck angelegten *Annotationsspur* im Partitur-Editor eingeben.

Für viele Anwendungen ist diese Form der Annotation ausreichend. In den Rückmeldungen der Anwender der EXMARaLDA-Werkzeuge wurde aber immer wieder der Wunsch geäußert, die angefertigten Transkriptionen mit umfangreichen linguistischen Annotationen anreichern zu können, die sich auf Segmente wie Wörter oder Sätze beziehen, ohne diese zuvor mühsam in einer Partitur-

darstellung markieren zu müssen.

Für schriftsprachliche Daten existiert eine Vielzahl von Verfahren, solche Annotationen anfertigen zu können, teilweise – je nach Sprache – mittels statistischer, automatisierter Verfahren, teilweise manuell. Diese Verfahren und die entsprechenden Programme lassen sich auf die meisten Daten gesprochener Sprache nicht einfach anwenden, da diese überwiegend auf andersartigen Datenmodellen und -formaten operieren und Daten gesprochener Sprache häufig nicht in Standardorthografie vorliegen.

Aus diesen Beobachtungen heraus reifte der Entschluss, die Annotation von Transkriptionen gesprochener Sprache in den Fokus dieser Arbeit zu rücken.

## **Annotation Science vs. Annotation**

Bei der Recherche, welche Datenmodelle, Formate und Werkzeuge für die Annotation von Transkriptionen gesprochener Sprache bereits existieren, stellte sich heraus, dass das Thema Annotation von Sprachdaten zwar offenbar einen breiten Raum in der Forschung einnimmt, konkrete Anwendungen, insbesondere für Daten gesprochener Sprache, aber ausgesprochen rar sind.

Dies liegt zum einen daran, dass sich die sog. *Annotation Science* (Ide 2007) immer noch sehr an stark annotierten schriftsprachlichen Ressourcen orientiert. So tritt das bei der ISO zur Standardisierung vorgesehene, seit Jahren in der Entwicklung befindliche Linguistic Annotation Framework (LAF) (Ide and Romary 2003a) zwar mit der Erkenntnis an, es sei essenziell, dass „(...) LAF accommodate all varieties of annotation and data (...)“, es setzt aber trotzdem als Primärdatum für die *Annotation Characters* fest, „a contiguous byte sequence of a specified length“ (Ide 2007: 3) – obwohl eine solche Festlegung (wie sich in dieser Arbeit noch zeigen wird) – für Transkriptionen gesprochener Sprache nicht immer sinnvoll sein muss.

Zudem entstehen im nicht klar definierten Rahmen dieser *Annotation Science* zwar eine Vielzahl von Vorschlägen, wie Annotationen und annotierte Daten modelliert werden sollten, konkrete Implementationen und insbesondere benut-

zerfreundliche Anwendungen erwachsen aus diesen Betrachtungen aber selten. Diese Beobachtung lässt sich für die Bereiche der Gesprächsforschung, pragmatischen Linguistik und Diskursanalyse, die sich vor allem mit gesprochenen Diskursen beschäftigen, noch verstärkt machen: Hier existiert ein tatsächliches Defizit an konkreten Anwendungen, mit denen sich Transkriptionen gesprochener Sprache auf einem Niveau anfertigen, annotieren und untersuchen lassen, das dem anderer linguistischer Disziplinen entspricht.

Zwar ist es nicht die erklärte Aufgabe der Computerlinguistik und der Texttechnologie (und auch nicht dieser Arbeit), Anwendungsprogramme für Gesprächsforscher zu schreiben, es wäre aber doch wünschenswert, wenn sich Erkenntnisse dieser Forschung auch positiv auf die Forschungspraxis auswirken würden.

Die Voraussetzungen dafür will diese Arbeit schaffen. Anhand der Besonderheiten, die Daten gesprochener Sprache ausmachen, soll erörtert werden, ob und welche Verfahren aus der Domäne der geschriebenen Sprache sich sinnvoll auf die der gesprochenen Sprache übertragen lassen und welche möglicherweise benötigt werden, um Transkriptionen mit komplexerer Annotation anreichern zu können.

Dabei soll ein starker Fokus auf die praktische Anwendbarkeit der erarbeiteten Lösungen gelegt werden. Wie die Arbeit zeigen wird, resultiert das Defizit an Verfahren und Softwarelösungen zur Annotation gesprochener Sprachdaten nicht aus einem Mangel an Datenmodellen oder fehlender theoretischer Auseinandersetzung mit dem Thema Annotation, sondern vielmehr aus fehlendem Praxisbezug.

Auch aus dieser Beobachtung heraus basiert das in dieser Arbeit entwickelte Annotationsverfahren auf dem EXMARaLDA-Modell: Für dieses Modell existieren bereits Programme und umfangreiche Korpora, sodass das Verfahren direkt auf seinen praktischen Nutzen hin erprobt werden kann. So kann festgestellt werden, ob ein erfolgreicher Brückenschlag zwischen Annotation Science und Praxis gelungen ist, mit dem sich konkrete Anwendungen für die Forschungspraxis entwickeln lassen.

## 1.1. Begriffsdefinitionen

Der Titel dieser Arbeit lautet „*Werkzeuge zur flachen Annotation von Transkriptionen gesprochener Sprache*“. Die verwendeten Termini sollen gleich zu Anfang geklärt und von verwandten Konzepten separiert werden.

Der Begriff *Werkzeuge* steht nicht zufällig gleich zu Beginn des Titels. Er bezieht sich in diesem Fall nicht nur auf konkrete, ausführbare Software, sondern auch auf zugrunde liegende Modelle und Datenformate.

Die Beschreibung der tatsächlichen Softwarewerkzeuge, die implementiert wurden, um das in dieser Arbeit entwickelte Annotationsverfahren zu testen, soll keinen bedeutenden Teil der Arbeit ausmachen, es soll aber durchaus klar gemacht werden, dass die Arbeit einen starken praktischen Fokus hat. Das entwickelte Annotationsverfahren soll auch seinen Niederschlag in konkreten Anwendungen finden. Die Konzepte und Methoden, die in der Arbeit herausgearbeitet werden, müssen sich vor allem am praktischen Nutzen für die linguistische Forschung messen lassen.

*Annotation*, der zentrale Begriff der Arbeit, ist in diesem Fall die Summe aller beschreibenden oder analytischen Bezeichnungen, die sprachlichen Primärdaten hinzugefügt werden. Diese Arbeit beschäftigt sich dabei insbesondere mit flacher Annotation. Formen hierarchischer Annotation werden nur am Rande behandelt, andere denkbare Annotationsformen spielen keine Rolle.

Im Bezug auf eine klare Definition und Abgrenzung der Begriffe *Rohdaten* versus *Primärdaten* versus *Sekundärdaten* sowie *Daten* versus *Metadaten* oder auch *Daten* versus *Annotationen* gibt es unterschiedliche Auffassungen. Da der Fokus der Arbeit auf Daten gesprochener Sprache liegt, sind die Rohdaten hier in den überwiegenden Fällen (Audio- oder Video-)Aufnahmen gesprochener Sprache. Die Transkriptionen dieser Aufnahmen sind – im Kontext dieser Arbeit – die Primärdaten. Die Annotationen, um die es in dieser Arbeit gehen soll, annotieren diese Primärdaten; der Begriff *Sekundärdaten* soll aber in dieser Arbeit für Annotationen vermieden werden, da hier wieder Ambiguitäten zum Begriff der *Metadaten* entstehen können, der in dieser Arbeit für Daten zu Konstellationen und



Sprechern verwendet wird. Informationen zur Segmentierung der Primärdaten zählen hierbei mit zu den Annotationen.

*„Note that the identification of a segment as a word, sentence, noun phrase, etc. also constitutes linguistic annotation. In current practice, when it is possible to do so, segmentation and identification of the linguistic role or properties of that segment are often combined (...)“ (Ide und Romary 2003: 4)*

Es sind gute Gründe denkbar, die Ebenen zwischen den Abstraktionsstufen anders einzuziehen und zu benennen. So lassen sich die Rohdaten beispielsweise eine Ebene tiefer aufhängen – beim eigentlichen Gesprächsereignis. Die Aufnahme dieses Ereignisses ist dann bereits ein Primärdatum. Damit wird dem Umstand Rechnung getragen, dass bereits mit der Aufnahme ein analytischer Schritt vorgenommen wurde, der potenziell Information verfälscht bzw. durch die Aufnahmekonstellation eine bestimmte Interpretation des Ereignisses vornimmt. Die Tatsache, dass es für das Gesprächsereignis aber außer der Aufnahme keine andere – der Analyse zugängliche – Repräsentation gibt, begründet für diese Arbeit die Entscheidung, diese unterste Ebene wegfällen zu lassen; für eine Ebene, für die kein zugängliches Datum existiert, gibt es auch keine spezielle Benennung.

Mit *Transkriptionen gesprochener Sprache* schließlich grenzt sich der Gegenstand der Arbeit klar von anderen Sprachdaten ab, also gedruckten Dokumenten, digitalen Texten, aber auch Abschriften von historischen Originalmanuskripten, für die auch häufig der Terminus *Transkription* verwendet wird. Ebenfalls nicht Bestandteil dieser Arbeit sind phonetische Transkriptionen oder Transkriptionen, die der Abbildung und Erforschung phonetischer oder phonologischer Phänomene dienen.

Ob die gesprochenen Rohdaten manuell transkribiert wurden oder die Transkriptionen maschinell mithilfe von Spracherkennung erzeugt wurden, spielt für den Gegenstand der Arbeit keine Rolle: Die Daten, die betrachtet werden, wurden jedoch alle manuell transkribiert.

Auch der Begriff der gesprochenen Sprache bedarf einer Klärung, diese wird im Kapitel 2.1 ab Seite 13 ausführlich vorgenommen.

## 1.2. Gliederung der Arbeit

In der vorliegenden Arbeit sollen Formate, Verfahren und Werkzeuge für die Annotation von Transkriptionen gesprochener Sprache evaluiert und daraus ein praktikables Annotationsverfahren für diese Art von Daten entwickelt werden.

In „Gesprochene vs. geschriebene Sprache“ ab Seite 13 werden dazu zunächst die spezifischen Merkmale gesprochener Sprache im Vergleich zu geschriebener Sprache (*Texten*) herausgearbeitet.

In „Anwendungsbereiche von Transkription“ ab Seite 31 werden verschiedene Wissenschaftsbereiche beleuchtet, in denen Transkriptionen eine Rolle spielen. In diesem Rahmen wird die Entscheidung motiviert, warum die Methoden der Computerlinguistik mit ihren Forschungsfragen zu Spracherkennung und -synthese in dieser Arbeit keine eingehende Behandlung erfahren wird. Darüber hinaus werden einige grundlegende Gedanken zum Wesen der Transkription eingeführt.

Eine zentrale Rolle bei der Betrachtung von Transkriptionen spielen bestimmte Eigenschaften gesprochener Sprache, die eine computergestützte Modellierung von Sprachdaten erschweren. Bezug nehmend auf die zuvor eingeführten Besonderheiten gesprochener Sprache wird auf diese in „Problemfälle für die Modellierung von Sprachdaten“ ab Seite 43 näher eingegangen und daraus eine Liste von Anforderungen für ein Datenmodell erstellt.

Im Kapitel „Existierende Datenmodelle“ ab Seite 55 werden die gebräuchlichsten Datenmodelle, die beim Anfertigen von Sprachkorpora eine Rolle spielen, vorgestellt. Diese Datenmodelle sollen auf ihre Eignung für die zuvor ermittelten Anforderungen untersucht werden. Dazu soll für sie jeweils geklärt werden:

- Gibt es ein Datenformat für das Datenmodell?
- Ist das Datenformat dokumentiert und nachhaltig?
- Lässt das Datenmodell Annotation zu?
- Welche Arten der Annotation sind möglich?
- Existieren Verfahren und Werkzeuge für die Annotation?
- Existieren Verfahren zur Visualisierung der Daten und Annotationen?

- Existieren Verfahren zur Abfrage und Suche in Daten und Annotationen?

Im diesem Kapitel soll auch das EXMARaLDA-Datenmodell, das als Grundlage für die zu erarbeitende Annotationslösung dient, mit seinen existierenden Anwendungen kurz vorgestellt werden. Die bestehenden Annotationsmöglichkeiten sollen betrachtet werden und es soll auf dieser Grundlage erörtert werden, ob und wie das Datenmodell modifiziert werden muss, um darüber hinaus gehende Annotationen zu ermöglichen.

Dem pragmatischen Ansatz der Arbeit trägt das Kapitel „Praktische Fragen zur Annotation“ ab Seite 117 Rechnung: Für die möglichen Annotationstypen wird erörtert, wie sich diese möglichst benutzerfreundlich eingeben lassen. Dazu werden die Verfahren einiger Annotationswerkzeuge schlaglichtartig erläutert und Empfehlungen für die Umsetzung aufgestellt.

Dem Kapitel „Ein Standoff-Annotationsverfahren für Transkriptionsgraphen“ ab Seite 143 schließlich dienen die zuvor aufgestellten Beobachtungen und Anforderungen als Grundlage, um ein neues Annotationsverfahren für Transkriptionen gesprochener Sprache zu entwickeln. Dieses Verfahren ist in der Lage, auch komplexere Annotationen auf Transkriptionen anzuwenden, ohne dass das Datenmodell – und damit bestehende Anwendungen für dieses Datenmodell – verändert werden müssen. Neben der Erläuterung des grundsätzlichen Verfahrens werden auch zwei *proof-of-concept* Softwarekomponenten vorgestellt, die einen Beleg dafür liefern, dass sich das entwickelte Verfahren auch in der Praxis einsetzen lässt.

Die Erkenntnisse dieser Arbeit schaffen die Basis für benutzerfreundliche Annotationsanwendungen und werden hoffentlich direkt in die Weiterentwicklung des EXMARaLDA Systems fließen, sodass es für eine Vielzahl von Forschern weiter an Attraktivität gewinnt. Einige Gedanken, welche konkreten Schritte dazu notwendig wären, schließen das fünfte Kapitel ab.

## 1.3. Anmerkungen zur Form

### Korpora

Beispiele aus Transkriptionen verfügen über eine von Literaturangaben abweichende Quellenangabe. Eine Angabe wie (DiK-Korpus, 205) bezeichnet das zitierte Korpus und die Fundstelle in diesem Korpus. Eine Auflösung der Abkürzungen und eine nähere Beschreibung der zitierten Korpora und findet sich in „Korpora“ auf Seite 188.

Beispiele für sprachliche Phänomene und Transkriptionsausschnitte, die keine Angabe zur Fundstelle in einem bestimmten Korpus haben, sind vom Verfasser erdacht.

### Geschlechtersensible Sprache

In der Arbeit wird keine geschlechtersensible Sprache angewandt. Bei der Verwendung des generischen Maskulinums wie bei *Forscher* oder *Wissenschaftler* sind Forscherinnen und Wissenschaftlerinnen immer mit einbezogen.

### Implementation und Anlagen

Die Arbeit umfasst als Teil der Implementation einige Softwarewerkzeuge. Die Version der Werkzeuge zum Zeitpunkt der Abgabe der Dissertation (Mai 2009) war der Arbeit ursprünglich als CD beigelegt. Sie findet sich – mit aktuelleren Versionen und inklusive einiger Erläuterungen zu deren Benutzung – im Internet unter: <http://www.exmaralda.org/sextant>.

## 2. Modellierung gesprochener Sprache

Um sich dem Thema *Annotation von Transkriptionen gesprochener Sprache* zu nähern, müssen zunächst die Unterschiede zwischen diesen Transkriptionen und geschriebener Sprache in Form von Texten verdeutlicht werden.

### 2.1. Gesprochene vs. geschriebene Sprache

Die Unterscheidung von Transkriptionen gesprochener Sprache und schriftsprachlichen Texten muss formal bereits eine Ebene tiefer vorgenommen werden, nämlich bei der Unterscheidung zwischen *geschriebener* und *gesprochener Sprache*; bei *Transkriptionen gesprochener Sprache* handelt es sich ja bereits um eine Form der Modellierung, die vom Forscher vorgenommen wurde.

Dieser Unterscheidung liegt die grundlegende Annahme zugrunde, dass sich geschriebene und gesprochene Sprache unterscheiden. Im Folgenden sollen zunächst die unterschiedlichen Bezeichnungen, die für die relevanten Konzepte existieren, benannt und auseinanderdividiert werden. Dann sollen die wichtigsten Unterschiede anhand spezifischer Eigenschaften der *gesprochenen Sprache* benannt werden, um später herausarbeiten zu können, wie sich diese Eigenschaften auf die Modellierung und computergestützte Auswertung auswirken.

#### 2.1.1. Was ist gesprochene Sprache?

Die Begriffe *gesprochene Sprache* und *geschriebene Sprache* werden in verschiedenen Kontexten unterschiedlich verwendet. Die definitorische Unklarheit ergibt sich daraus, dass Ebenen wie die Saussursche Dichotomie *langue/parole* und die Unterscheidung zwischen einer konzeptuellen und einer medialen (Realisierungs)-Ebene nicht klar benannt und berücksichtigt werden.

Auch die Grenzen zwischen den Begriffen *gesprochene Sprache/geschriebene Sprache* und verwandten Konzepten scheinen fließend zu sein, so existieren auch die Paare *Rede/Schrift*, *Gespräch/Text*, *Mündlichkeit/Schriftlichkeit*, *Sprache der Nähe/Distanzsprache*, die sich in demselben Themenfeld befinden.

Eine Kategorisierung in Rede und Schrift lässt sich beispielsweise anhand von

intuitiven Merkmalen in einer einfachen, zweidimensionalen Tabelle abtragen:

<b>Eigenschaft</b>	<b>Rede</b>	<b>Schrift</b>
<b>Medium</b>	<i>akustisch</i>	<i>visuell</i>
<b>Zeitstruktur</b>	<i>parallele Signalströme</i>	<i>lineare Symbolfolge</i>
<b>Organisation</b>	<i>eher flach</i>	<i>stark hierarchisch</i>
<b>Einheiten</b>	<i>eher äußerungsorientiert</i>	<i>eher satzorientiert</i>
<b>Produktion</b>	<i>„fehlerhaft“</i>	<i>„editiert“</i>
<b>Rezeption</b>	<i>flüchtig</i>	<i>speichernd</i>
<b>Interaktivität</b>	<i>eher dialogisch</i>	<i>eher monologisch</i>
<b>Soziale Funktion</b>	<i>eher unverbindlich</i>	<i>eher verbindlich</i>
<b>Modalität</b>	<i>multimodal</i>	<i>unimodal</i>

Abbildung 1: Eigenschaften von Rede und Schrift.<sup>1</sup>

An dieser Tabelle lässt sich allerdings auch relativ schnell erkennen, dass mehrere der zuvor angeführten Kategorien in dieser Einteilung zusammenspielen, die für Begriffsklarheit besser getrennt betrachtet würden. Auch lassen sich einige Eigenschaften bei näherer Betrachtung nicht mehr so klar einem der beiden Pole zuordnen: Kann *Schrift* nicht auch *unverbindlich* sein? Ist die *Produktion* von *Rede* automatisch *fehlerhaft*?

Am eindeutigsten lässt sich eine Abgrenzung wohl auf der Ebene der Realisierung bzw. des eingesetzten Mediums vornehmen. Liegt ein gedruckter oder niedergeschriebener Text vor, so lässt sich von einer *Realisierung im grafischen Kode*<sup>2</sup> sprechen.

Bei Sprache, die mit dem menschlichen Sprechapparat – den Artikulationsorganen Lunge, Kehlkopf, Zunge, Gaumensegel und Lippen – produziert und mündlich geäußert, also tatsächlich *gesprochen* wird, liegt eine *Realisierung im phonischen Kode* vor.

Diese Kodes bilden eine Dichotomie: Sprache lässt sich immer eindeutig und vollkommen einem Realisierungsmodus zuordnen. Eine gesprochene Äußerung ist immer eine Realisierung im phonischen Kode, ein Zeitungsartikel auf Papier

1 Aus Gibbon (1997), erweitert von mir. Vgl. auch Gibbon (1997b: 81f).

2 Vgl. Koch und Oesterreicher (1985: 17).

eine Realisierung im grafischen Kode. Obwohl die Zuordnung zu diesen Polen einfach und eindeutig möglich ist, sind die Kodes mit Einschränkungen ineinander überführbar.

„Graphie und Phonie sind zwei auf Ausdrucks- und Inhaltsebene extrem isomorphe (ähnlich strukturierte) Kodes. Das macht auch die Gleichsetzung und Zuordnung von phonischen und graphischen Ausdrücken (und ihren Inhalten) zueinander möglich: Deshalb kann man originär schriftliche Texte auch (mit mehr oder weniger großen Einbußen!) laut sprechen und originär mündliche Texte (ebenfalls mit Verlusten) verschriftlichen und lesen.“ (Kalmbach 1996: 85)

Diese Isomorphie der Kodes liegt natürlich darin begründet, dass es sich um dasselbe Sprachsystem, mithin dieselbe Sprache im Sinne der *langue* handelt, das realisiert wird.<sup>3</sup>

Nur in den Bereich des phonischen Kodes fallen auch alle Aspekte der Phonologie und Phonetik, die einen großen Bereich der Linguistik ausmachen und für die eigene Transkriptions- und Schriftsysteme existieren. Diese sollen aber in dieser Arbeit keiner ausführlichen Betrachtung unterzogen werden.

Neben dieser strengen Dichotomie steht als zweites Unterscheidungsmerkmal nach Söll (1985: 17ff) die Ebene der *Konzeption* sprachlicher Äußerungen mit den Polen *gesprochen* und *geschrieben*, die sich nicht auf das Medium, sondern auf kommunikative Strategien beziehen.

Eigenschaften, die einen sprachlichen Ausdruck in die *Konzeptions*-Kategorie *gesprochen* sortieren würden, wären nach Koch und Oesterreicher (1985) beispielsweise *Vorläufigkeit*, *geringe Komplexität* und *geringere Elaboriertheit*, für die Kategorie *geschrieben* die jeweils gegenteiligen Eigenschaften *Endgültigkeit*, *gro-*

---

3 Aus diesem Grund ist auch die Realisierung in einer Gebärdensprache ein Sonderfall: Gebärdensprachen lassen sich nicht als phonischer Kode realisieren – die „gesprochene“, also in diesem Fall gebärdete Variante bildet einen eigenen linguistischen Kode. Dieser lässt sich zwar auch in einen phonischen oder graphischen Kode überführen, muss dafür aber in eine andere Sprache übersetzt werden. Auch bei Transkriptionen von gebärdeten Gesprächssituationen wird diese Übersetzung in graphischen Kode einer anderen Sprache (also beispielsweise DGS -> Deutsch) in den meisten Fällen vorgenommen, obwohl auch graphische Kodes für Gebärdensprachen existieren. Vgl. dazu die Webseiten zur deutschen Gebärdenschrift (<http://www.gebaerdenschrift.de>) und zu anderen Gebärdenschriftsprachen (<http://www.signwriting.org/>).

*ße Komplexität und hohe Elaboriertheit.*

An den äußeren Polen dieser beiden Achsen ergeben sich für einen gegebenen sprachlichen Ausdruck vier Zuordnungsmöglichkeiten: *gesprochen+phonischer Kode*, *gesprochen+grafischer Kode*, *geschrieben+phonischer Kode* sowie *geschrieben+grafischer Kode*. Wie Koch und Oesterreicher (1985) aber feststellen, lassen sich viele Äußerungsformen nicht eindeutig einer *Konzeptions*-Kategorie zuordnen:

*„Bei genauer Betrachtung dieser doppelten Unterscheidung stellt sich heraus, dass das Verhältnis von phonischem und graphischem Kode im Sinne einer strikten Dichotomie zu verstehen ist, während die Polarität von ‚gesprochen‘ und ‚geschrieben‘ für ein Kontinuum von Konzeptionsmöglichkeiten mit zahlreichen Abstufungen steht.“ (Koch und Oesterreicher 1985: 17)*

In diesem Kontinuum lässt sich eine Äußerungsform *vertrautes Gespräch* nahe am Pol *gesprochen*, eine Äußerungsform *Verwaltungsvorschrift* nahe am Pol *geschrieben* und eine Äußerungsform *Tagebucheintrag* zwischen den Polen und den beiden anderen Formen verorten.

Koch und Oesterreicher (1985) stellen fest, dass (mit Einschränkungen, s.o.) immer eine Übertragung dieser Äußerungsformen von einem Medium (Kode) ins andere möglich ist. Da es ihnen darum geht, Varietätenunterschiede in dem Kontinuum der Konzeption auszumachen, legen sie für ihren Untersuchungsgegenstand fest:

*„Im folgenden wird daher das konzeptionelle Kontinuum mit den Polen ‚gesprochen‘ und ‚geschrieben‘ als definitorisch für Mündlichkeit und Schriftlichkeit im Mittelpunkt unserer Überlegungen stehen.“ (Koch und Oesterreicher 1985: 19)*

Wird von *gesprochener* und *geschriebener Sprache* jenseits der Realisierungsebenen *gesprochen*, so sind wohl in den meisten Fällen diese Pole gemeint, für die häufig auch andere Begriffe wie *informell vs. formell* oder *Umgangssprache vs. Schriftsprache* gewählt werden.



Betrachtet man die Tabelle auf Seite 14 in diesem Kontext noch einmal, so stellt man fest, dass Eigenschaften, die für die Zuordnung zu den dort verwendeten Kategorien *Rede* und *Schrift* verwendet werden, in unterschiedliche Kategorien der Unterscheidung zwischen *Konzeption* und *Realisierung* fallen: Die Eigenschaft *Medium* liegt ganz klar auf der Realisierungsebene (und ist in gewissem Sinne mit dieser identisch), eine Eigenschaft wie *soziale Funktion* liegt dagegen eindeutig auf der Ebene der Konzeption – *eher unverbindlich* würde im Kontinuum der Konzeptionsebene näher beim Pol *gesprochen* oder bei der *Mündlichkeit* liegen.

Dass diese doppelte Unterscheidung überhaupt so scheinbar problemlos außer Acht gelassen werden kann, wie es in der Tabelle zu sehen ist, liegt daran, dass eine Äußerungsform, die auf der Konzeptionsebene nahe am Pol der Schriftlichkeit liegt, natürlich eine gewisse Affinität zur grafischen Realisierung besitzt. So „sind zweifelsohne die Kombinationen ‚gesprochen+phonisch‘ (...) und ‚geschrieben + graphisch‘ (...) besonders typisch.“ (*Koch und Oesterreicher 1985: 17*). Es existieren aber eben auch die beiden anderen Paare, weshalb die doppelte Unterscheidung auf jeden Fall sinnvoll ist.

Diese Arbeit beschäftigt sich mit *Transkriptionen gesprochener Sprache*. Der Begriff der *gesprochenen Sprache* bezieht sich in diesem Fall einzig auf die Ebene der Realisierung. Gesprochene Sprache ist also Sprache, die – wie oben bereits angeführt – mit dem menschlichen Sprechapparat erzeugt und mündlich geäußert wird. An welcher Stelle im Kontinuum Mündlichkeit-Schriftlichkeit der Konzeptionsebene sich die Sprache befindet, spielt für die Betrachtungen dieser Arbeit ebenfalls eine Rolle: Die Forschungsfelder, für welche die Konzepte und Lösungen dieser Arbeit relevant sind, beschäftigen sich freilich überwiegend mit Spontansprache, also frei, ungeplant und mündlich geäußelter Sprache. In der Regel finden sie sich also auf der Achse der Konzeption eher auf der Seite der Mündlichkeit. Die spezifischen Probleme, die sich bei der Modellierung gesprochener Sprache für die Verarbeitung mit dem Computer (siehe Seite 43 ff.) ergeben, verstärken sich, je näher man sich auf dieser Achse der Mündlichkeit nähert.

## 2.1.2. Besonderheiten der Mündlichkeit

Was sind nun aber die konkreten Eigenschaften, die gesprochene von geschriebener Sprache unterscheiden? Für eine derartige Untersuchung ist es wichtig, die zuvor eingeführte Unterscheidung zwischen Konzeptions- und Realisierungsebene im Kopf zu behalten. Für die Realisierungsebene ist das primäre Unterscheidungsmerkmal nun klar: (medial) gesprochene Sprache **ist** *gesprochen* und geschriebene Sprache *geschrieben*. Diese simple Unterscheidung hat Auswirkungen auch auf die Ebene der Konzeption. Exemplarisch lassen sich mehrere Eigenschaften und Phänomene benennen, die Schriftlichkeit von Mündlichkeit unterscheiden.

Es fällt dabei auf, dass einige Merkmale, die in der gesprochenen Sprache auftreten, in der geschriebenen Sprache als *fehlerhaft* (im Sinne von *ungrammatisch*) gelten würden. Die Fehlerhaftigkeit dieser Merkmale lässt sich für gesprochene Sprache aber nur dann feststellen, wenn man sie von der Folie der geschriebenen Sprache aus beurteilt. Sprachverwendungen wie Ellipsen oder Aposiopesen haben in der gesprochenen Sprache spezifische kommunikative Funktionen, die das Gelingen einer kommunikativen Handlung ermöglichen oder erleichtern. Es wäre vor dieser Feststellung also irreführend, bei solchen Phänomenen von *Fehlern* zu sprechen, wie die Begriffe es zunächst einmal suggerieren.

Für die linguistische Untersuchung von Diskursen gesprochener Sprache werden in den meisten Fällen Transkriptionen angefertigt. Transkription als wissenschaftliche Methode existiert aber auch in anderen Forschungsfeldern, mit zum Teil stark abweichenden Verfahren und Definitionen. Einige davon werden in der Arbeit kurz vorgestellt, um dann auf die speziellen Eigenschaften von Transkriptionen im Bereich der in dieser Arbeit fokussierten Forschungsbereiche einzugehen.

Auch lässt sich in den letzten Jahren die Tendenz feststellen, dass sich Merkmale der Mündlichkeit zunehmend in Genres der geschriebenen Sprache, insbesondere journalistischen Textsorten, wiederfinden. Durch den regelmäßigen Gebrauch von Konstruktionen wie den auf Seite 27 erwähnten Herausstellungen in solchen

Genres, verfestigen sich diese Merkmale auch in der schriftlichen Sprache und werden im Laufe der Zeit auch dort nicht mehr als fehlerhaft wahrgenommen. Keller (1999) bemerkt zu dieser Form des Sprachwandels:

*„We view language selectively, in a small window of time, and in this meager slice, we necessarily see a slew of mistakes and ,barbarisms.‘ But the systematic mistakes of today are very probably the new rules of tomorrow.“*

Für die gesprochene Sprache eigene Kategorien zu etablieren, ist bereits mehrfach vorgeschlagen worden. Hennig (2006: 17) bemerkt dazu:

*„Der angesichts der stark skriptizistischen Grammatiktradition verständliche Versuch, die Grammatik der gesprochenen Sprache als eigenständiges System zu beschreiben, das nur mit neuen Kategorien sinnvoll erfasst werden kann, wird der historischen Lagerung kontext- und symbolgrammatischer Strukturierungen nicht gerecht.“*

Andererseits wird durchaus die Notwendigkeit einer Grammatiktheorie für die gesprochene Sprache erkannt und diskutiert. So bemerkt z.B. Deppermann (2006: 23):

*„Zwar können aus den Bedingungen verbaler Interaktion, die für die Produktion und Rezeption gesprochener Sprache maßgeblich sind, interaktive und kognitive Determinanten gewonnen werden, die Eigenheiten der gesprochenen Sprache erklären. Was jedoch weiterhin fehlt, ist eine umfassende Theorie der Grammatik der gesprochenen Sprache und mit ihr eine konsistente Beschreibungssprache, die an die spezifische Konstitutionsweise des Gesprochenen angepasst ist.“*

### **Zeitstruktur und Organisation**

Die augenfälligsten und für die Modellierung von Sprachressourcen kritischsten Unterscheidungsmerkmale sind die *Zeitstruktur* und *Organisation* gesprochener Sprache.

Allein die Tatsache, dass eine Zeitstruktur vorliegt, stellt einen bedeutsamen Unterschied zwischen gesprochener und geschriebener Sprache dar: Die Äußerungen der gesprochenen Sprache geschehen zu einem bestimmten Zeitpunkt und in einem bestimmten Rhythmus. Sie können mit unterschiedlicher Geschwindigkeit

erfolgen, es können Pausen oder schnellere Passagen enthalten sein. Diese Merkmale sind Träger von *paraverbaler* Information, die für Forschungsbereiche wie die Gesprächsforschung wichtige Erkenntnisse liefern können. Vor allem aber kann es in der *Zeitstruktur* zu Überlappungen kommen, sofern mehrere Sprecher am Kommunikationsereignis beteiligt sind.<sup>4</sup>

[40]

	185	186	187	188	189	190	191	192
Nes [v]							schnell	
Mut [v]	(taşınmışık), ••	şu	raya.	((anl.))	Onlar	işte	ev	aldılar.
Ati [v]	((anl. ))'		((anl.))'				Sie	meint so: "Eins, zwei, (drei)".

Abbildung 2: Lineare, parallele Signalströme mit Überlappungen. (Rehbein-Korpus, 731)

Die sprachliche *Organisation* ist dabei aber flach. Die einzelnen Signalströme können zwar Hierarchien wie Sätze, Nebensätze, Einschübe und Gliedsätze aufweisen, weitere strukturierende Elemente, wie sie in geschriebener (und hier insbesondere in gedruckter) Sprache auftreten können (Kapitel, Layoutanweisungen, Absätze, Gliederungen etc.), treten dafür nicht auf. Wie sich dies auf die Modellierung der Sprachressourcen auswirkt, wird ab Seite 44 näher erläutert.

## Flüchtigkeit

Für Ehlich (1994) ist die *Flüchtigkeit* das zentrale und prototypische Merkmal für mündliche Kommunikation. Ein mündliches Gesprächsereignis ist demnach immer ein einmaliges Ereignis. Diese Einmaligkeit hat eine Reihe von Konsequenzen: Die an einer ad-hoc-Kommunikation beteiligten Sprecher haben ein eingeschränktes Wahrnehmungs- und Planungsfeld, in dem sie ihre Äußerungen formulieren müssen.

*„Spontane verbale Planung geht der Äußerungsformulierung voraus; in der Regel kann nicht mehr als eine Äußerungseinheit im Vorhinein geplant werden (psycholinguistische Faustregel).“ (Dittmar 2004: 40)*

In dieser kurzen Spanne müssen nicht nur die sachlichen, sondern auch hand-

4 Es kann auch zu Überlappungen mit nichtsprachlichen Ereignissen kommen, was in bestimmten Forschungszusammenhängen wichtig und daher für die Transkription relevant sein kann.

lungs- und diskurssteuernden Informationen erarbeitet und mitgeteilt werden – und vom Rezipienten auf der anderen Seite verarbeitet und in die eigene Diskursplanung aufgenommen werden<sup>5</sup>. Alle weiteren (zum Teil im Folgenden noch einmal aufgegriffenen) Faktoren wie spontane Rückmeldungen durch Interjektionen, Mimik etc. fließen in die spontane Äußerungsplanung ein und können die Äußerung modifizieren. Dies erfordert und bedingt eine Effizienz in der Kommunikation, die bei der Produktion von schriftlicher Sprache meistens nicht erforderlich ist.

### **Lautbildung (Phonologie und Phonetik)**

Die Erforschung der Lautlehre in den Disziplinen der Phonologie und Phonetik berührt einen Teilbereich der Sprache, der ausschließlich in der gesprochenen Sprache vorkommt. Die physiologische Ebene der Produktion und Wahrnehmung von Sprache hat keine Entsprechung in der geschriebenen Sprache. Für die perzeptive Beobachtung und Beschreibung dieser Ebene existieren aber natürlich Modelle (insbesondere in Form von phonetischen Alphabeten), um sich in der geschriebenen Sprache damit auseinandersetzen zu können und eine möglichst enge Beziehung zwischen Sprachsignal und Transkription herstellen zu können.

Da es sich bei einem Sprachsignal um eine physikalisch messbare Größe handelt, lässt sich diese natürlich auch in anderer Form festhalten als mit den Mitteln der Schriftsprache. Analysen des Sprachsignals auf bestimmte Parameter wie Amplitude über Zeit lassen sich mit computergestützten Verfahren visualisieren und weiter analysieren. Häufig werden solche Visualisierungen als Hilfsmittel bei der eigentlichen Transkription verwendet.

Für die Transkription gesprochener Sprache spielen diese Faktoren eine gewichtige Rolle: Phonetische, phonemische oder phonologische Transkriptionen stellen andere Anforderungen an Modellierung, Werkzeuge und insbesondere Transkribenten. In den Kontexten von Gesprächsforschung und Diskursanalyse wird meistens angestrebt, mit den Mitteln des Standardalphabets möglichst viel Information bezüglich der phonologischen Ebene in die Transkriptionen her-

---

5 Vgl. Rosen, Howell und Bartram (1993).

überzuretten und gleichzeitig eine einfache Lesbarkeit zu erhalten (siehe „Lautung vs. Schreibung“ ab Seite 45).

Die Eigenschaft der Sprachmelodie oder Intonation lässt sich auf diese Art und Weise allerdings nie befriedigend abbilden. Da sie aber wichtige Funktionen in Bezug auf Gesprächssteuerung und den Ausdruck von Emotionen hat, muss sich ein Forscher, der auf diese Informationen angewiesen ist, auch Gedanken darüber machen, ob das gewählte Transkriptionssystem mit dieser Art Daten umgehen kann.

### **Morphologie**

Die Morphologie spielt in den hier betrachteten Transkriptionskonventionen für die eigentliche Transkription keine Rolle. Einzig in Form einer morphologische Transliteration transkribierter Äußerungen wird sie bedeutsam, vor allem, „je größer die Distanz zwischen der Sprache der Daten und der Metasprache der Analyse ist. [...] Auf diese Weise sind, z.B. in agglutinierenden Sprachen, zumindest Einzelworte jeweils einer Kette von durch Bindestriche verbundenen Morphembezeichnungen [...] zuzuordnen“ (Rehbein et al. 2004: 61).

Um die Morphologie gesprochener Sprache erfassen zu können, müssen immer tatsächlich geäußerte Laute (als Audiomaterial) zur Verfügung stehen.<sup>6</sup>

### **Nonverbale Elemente**

Ein Merkmal, das mündliche Kommunikation auszeichnet, ist der Einsatz nonverbaler Mittel im Diskurs. Zu diesen nonverbalen Mitteln zählen alle Formen der Kommunikation, die in einem *nichtsprachlichen Modus* erfolgen. Zuvorderst sind dies natürlich diejenigen, die über den visuellen Kanal gesendet und wahrgenommen werden, insbesondere jene, die von einem Teilnehmer ausgehen, also Gestik, Kinesik, Mimik und Augenkontakt / Orientierung der beteiligten Sprecher.

Kommunikation, die über den ausschließlich sprachlichen Modus hinausgeht, wird als *multimodale Kommunikation* bezeichnet und in der Forschung ausführ-

---

6 Vgl. Bleiching et al. (1995).

lich behandelt.<sup>7</sup> Für die Modellierung, Bearbeitung und Kategorisierung multimodaler Daten existieren spezielle Werkzeuge.<sup>8</sup>

Auch andere Parameter, die mittels der menschlichen Sensorik erfahrbar sind, können für die Kommunikation eine Rolle spielen, werden aber deutlich seltener betrachtet und spielen in der linguistischen Forschung keine sichtbare Rolle. Auch nicht-personengebundene Ereignisse und Dinge können auf den Gesprächsverlauf einwirken und daher relevant für die Beschreibung einer Gesprächssituation sein.<sup>9</sup>

### **(Ab-)Brüche und Reparaturen in Äußerungen**

Durch die bereits erwähnte spontane *ad-hoc* Diskursplanung, die durch eine ganze Reihe von Faktoren auch während einer Äußerung beeinflusst wird, kommt es in der gesprochenen Sprache häufig zu Brüchen und Korrekturen innerhalb einer Äußerung.

A: (Es) war aber ja Freitag, ne?  
P: Das is Frei/ Sonnabend früh...  
A: Hm  
P: Freitag.  
Tschuldigung.  
Freitag, ja.

(DiK-Korpus, D-ANA-62)

Abbildung 3: Abbruch.

Bei Brüchen, die auf Änderungen in der Äußerungsplanung zurückzuführen sind, wird von Anakoluthen gesprochen<sup>10</sup>; diese Selbstkorrekturen haben einen Fokus und einen Skopus (Bezugsspanne).

---

7 Der Begriff der multimodalen Kommunikation wird auch (und ungleich häufiger) im Bereich der Mensch-Maschine-Kommunikation gebraucht – um diesen Bereich geht es hier nicht. Zum Stellenwert der Multimodalität vgl. Schmitt (2004).

8 Die meisten Werkzeuge, die mit Daten der multimodalen Kommunikation arbeiten, sind Transkriptionswerkzeuge für gesprochene Sprache, die die benötigten Funktionen in unterschiedlichem Maße zusätzlich anbieten. Eine Übersicht über verschiedene Werkzeuge bietet Rohlfing et al. (2006), auf einige dieser Werkzeuge wird im Kapitel „Existierende Datenmodelle“ ab Seite 57 näher eingegangen.

9 Vgl. Henne und Rehbock (2001: 79).

10 Vgl. Hennig (2006: 261f).

*„Da das Sprechen stark unter dem Einfluss der Situation und der Teilnehmer an einer Interaktion steht, sind oft Korrekturen in Bezug auf die geäußerten Gedanken notwendig, die als Folgen verschiedener kommunikativer Bedürfnisse wie (i) Neu- und Umbewertungen, (ii) Richtigstellen falscher Informationen, (iii) Überschreiben alter Informationen im Lichte neuer Erkenntnisse, (iv) Zurücknahme oder Verstärkung von Bedeutungen etc. auftreten können. [...] Von solchen spontanen Korrekturen ist die Tätigkeit des Schreibens frei.“ (Dittmar 2004: 32)*

Im Unterschied dazu stehen *Aposiopesen*, also Abbrüche, die vom Verfasser als rhetorisches Mittel beabsichtigt sind:

Floh: Und wenn ich den Stecker rauszieh, dann...  
Mer2: Ha, neler yapıyosunuz falan?  
Sin: Ha`  
Mer2: H1`  
Floh: und wenn ich den Stecker rauszieh?  
Nad: Dann krickst du Ärger.

(Rehbein-Korpus, 0406a/b19)

Abbildung 4: Aposiopese.

Davon lassen sich Reparaturen unterscheiden, die nicht mit Änderungen in der Äußerungsplanung zusammenhängen, aber auch nicht beabsichtigt sind, sondern am ehesten auf einen Mangel an Zeit zurückzuführen sind.

## **Ellipsen**

Ellipsen, also im grammatischen Sinne unvollständige Sätze, treten in der gesprochenen Sprache sehr häufig auf und liefern ein weiteres prototypisches Merkmal für die Effizienz gesprochener Sprache.<sup>11</sup> In einem Dialog wie dem Folgenden sind mehrere Beiträge keine vollständigen Sätze.

---

11 Vgl. Stein (2003: 88f).



Sey: So eine Frau und dann/ Mi/ Mitt/ nee,  
 • Montag und Donnerstag kommt immer ein Mann.  
 Mer2: Aha  
Sey: • (Thiefs) odeer • • Thorsten.  
Mer2: Thorsten.  
Sey: • Und (Thiefs).  
 Mer2: Und die helfen euch bei den Schularbeiten?  
 Sey: Ja.

(Rehbein-Korpus, 413)

Abbildung 5: Ellipsen.

Durch die räumlich-situative Erfahrung, über die die beteiligten Sprecher verfügen, können Redundanzen in der Kommunikation vermieden werden. Das Verständnis generiert sich bei Antwortellipsen wie dieser aus Konstruktionsübernahmen aus vorherigen Konstruktionen. Diese bilden eine syntaktische Grundlage, auf der die Ellipsen funktionieren können. Diese Grundlage wird verlassen, wenn ein Themenwechsel auftritt. Dann muss in der Regel mit einem vollständigen Satz eine neue Grundlage geschaffen werden.

In Dialogen sind Ellipsen eher die Regel als die Ausnahme, da das Effizienzgebot verhindert, dass überflüssige (bereits eingeführte) Informationen wiederholt werden. Würde Sprecher Sey nicht

Sey: • (Thiefs) odeer • • Thorsten.

sondern

Sey: Montags und Donnerstags kommt immer (Thiefs) odeer Thorsten.

sagen, so würde dieses Gebot klar verletzt.<sup>12</sup>

An den Phänomenen von Brüchen und Ellipsen lässt sich gut erkennen, dass gängige Grammatiktheorien geschriebener Sprache für die gesprochene Sprache ungeeignet sind:

*„Die Unzulänglichkeit gängiger Grammatiktheorien für die gesprochene Sprache beruht (unter anderem) darauf, dass sie fast immer von folgenden Prämissen ausgehen:*

---

12 Vgl. auch die Grice'schen Kommunikationsmaxime (Maxime der Quantität) (Grice 1975).

(1) *Satz-Prämisse: Vollständige syntaktische Einheiten sind Sätze, sie drücken eine Proposition aus und bestehen mindestens aus Subjekt und Prädikat.*

(2) *Formalitätsprämisse: Syntaktische Regeln sind rein formal und deshalb abstrakt und allgemein. Sie sind deduktiv und exhaustiv, d.h., sie gelten für alle Instanzen der betreffenden grammatischen Kategorie (Wortart, Satztyp etc.) bzw. syntaktischen Relation (Satzglied etc.).*

(3) *Kompositionalitätsprämisse: Phrasen- und Satzbedeutung sind kompositional: Sie ergeben sich aus der lexikalischen Bedeutung der Wörter und der syntaktischen Struktur ihrer Verknüpfung“ (Deppermann 2006: 44)*

## **Konjunktionen**

Konjunktionen finden sich sowohl in geschriebener wie auch in gesprochener Sprache. In der gesprochenen Sprache haben sich aber alternative Verwendungsweisen von Konjunktionen etabliert, insbesondere die Verwendung einer subordinierenden Konjunktion (*weil, während, obwohl* und andere) mit anschließender Verbzweitstellung (*epistemisches weil*<sup>13</sup> bei Begründung einer Erkenntnis des Sprechers oder *illokutives/sprechaktbezogenes weil* als Begründung des eigentlichen Sprechaktes) entstammt als Konstruktion klar der gesprochenen Sprache:

A: Ja, aber wenn dein Vater es dir auf Türkisch erzählt, Altan, kann ich es ja nicht verstehen, weil ich sprech ja gar kein Türkisch.  
(Rehbein-Korpus, 0192a)

Abbildung 6: „Illokutives weil“.

In der geschriebenen Sprache würde eine derartige Konstruktion als ungrammatisch wahrgenommen. In ähnlicher Weise werden in der gesprochenen Sprache Relativsätze nach dem Muster von Hauptsätzen gebildet:

A: Es gibt Tage, die machen mich ganz krank.

---

13 Vgl. Blühdorn (2008).

## Herausstellungen

(Links-, Rechts-) Herausstellungen, Rahmendurchbrechungen, Prolepsen, (Rechts-, Links-) Versetzungen, Expansionen und andere Begriffe<sup>14</sup> stehen für Phänomene, dass Satzglieder, die in der geschriebenen Sprache im Mittelfeld stehen müssten, in das Vor- oder Nachfeld der gesprochenen Äußerung rutschen:

A: Seit der Operation, oder?

P: Die ist jetzt zwei Jahre her, nich, *die Lungenoperation*.

(DiK-Korpus, D-AUF-58)

Abbildung 7: Rechts herausstellung.

In diesem Fall verlässt das Satzglied die eigentliche Satzklammer vermutlich, weil der Sprecher beim Tätigen der Äußerung bemerkt, dass zusätzliche Informationen benötigt werden, um die kommunikative Funktion erfüllen zu können.

Linksherausstellungen dagegen werden häufig verwendet, um einen kommunikativen Anschluss zu einer früheren Äußerung oder an ein früheres Thema herstellen zu können, Äußerungsteile vorwegzunehmen, mehrere Akzente zu setzen oder das Verstehen für den Partner durch Segmentierung zu erleichtern. Rechts herausstellungen können verwendet werden, um – wie oben – die Referenz sicherzustellen oder generell Informationen zu ergänzen. Oftmals werden sie interaktiv produziert, zum Beispiel wenn eine Reaktion des Gesprächspartners ausbleibt.

Die Herausstellungen erfüllen in manchen Fällen nicht die Kriterien an einen grammatisch vollständigen Satz, können aber – im Gegensatz zur Ellipse – mit dem dazugehörigen Satz meist vollständig aufgelöst werden.<sup>15</sup> Herausstellungen haben in einigen Genres der schriftlichen Sprache – insbesondere in der Presse – in den letzten Jahren stark zugenommen.

---

14 Vgl. Altmann (1981).

15 Zu Herausstellungen vgl. Auer (1991).

## Reduktionen

Im Zusammenhang mit gesprochener Sprache wird der Begriff der *Reduktion* für zweierlei verwendet: Zum einen werden Phänomene wie *Ellipsen* und *Abbrüche* auch als *Reduktion* (eines gedachten, grammatischen Satzes) bezeichnet, zum anderen auf der Ebene der Akustik die Abschwächung unbetonter Silben und Wörter in der fließenden Rede. Diese Abschwächung kann so stark sein, dass Silben und Wörter gar nicht geäußert werden, obwohl sie in der geplanten Äußerung des Sprechers wohl vorhanden waren.

P: Ich war angemeldet, ja.  
A: Aber hatten eh schon n Termin hier.

(DiK-Korpus, D-ANA-55)

Abbildung 8: Reduktion.

Auch **Klitisierungen** wie *sonen* statt *so einen* fallen in diesen Bereich, obwohl sie zumeist nicht mehr unter den Begriff der Reduktion gefasst werden.

## Modalpartikeln

Modalpartikeln treten in der gesprochenen Sprache häufiger auf und übernehmen dort Funktionen, die sie in der geschriebenen Sprache nicht haben.<sup>16</sup> Sie werden vor allem eingesetzt, um eine bestimmte Haltung eines Sprechers zu einem Sachverhalt auszudrücken, sie sind als solche also auch Träger von Information und keineswegs redundant.

Fer1: Jàa, das ist halt so'ne Tradition, ne?

(Rehbein-Korpus, 757)

A: Das war jetzt vielleicht überflüssig!

Abbildung 9: Spezielle Verwendung von Modalpartikeln.

Da es sich häufig um Formengleichheit handelt, sind sie an der Oberfläche nicht immer (und vor allem nicht automatisch) sofort auszumachen. So könnte der zweite Satz sowohl als „Möglicherweise war das jetzt überflüssig“ als auch als „Das war jetzt total überflüssig“ interpretiert werden. Die Disambiguierung

---

16 Zu Modalpartikeln vgl. Heggelund (2001).

ließe sich in diesem Fall wahrscheinlich aufgrund der Betonung des Satzes leicht vornehmen.

### **Deiktische Mittel und Handlungszusammenhang**

Deiktische Ausdrücke sind ein zentrales Element in der linguistischen Pragmatik. Sie zeigen auf Dinge, die sich im *Verweisraum* des Diskurses befinden, und zwar von der *Basis* und dem *Ausgangspunkt des Sprechers* („Hier-Jetzt-Ich-Origo“).<sup>17</sup>

Dabei können deiktische Ausdrücke auf personelle (ich, du, er), temporale (jetzt, dann, nachher) und lokale (hier, dort, drüben) Charakteristika verweisen.

Die Ausdrücke kommen natürlich auch in der geschriebenen Sprache vor, dort fehlt ihnen aber das gemeinsame Bezugssystem, das Sprecher und Hörer in einer Gesprächssituation zur Verfügung haben, die Deixis kommt also durch die bestimmte Situation zustande.

Dieser Handlungszusammenhang ist es, der für viele der Besonderheiten gesprochener Sprache bestimmend ist. So sind beispielsweise Gegenstände im Handlungszusammenhang eines Diskurses adressierbar<sup>18</sup>, beteiligte Sprecher können den Aufmerksamkeitsfokus gezielt lenken, der Sprachhandlungskontext wird intuitiv wahrgenommen. Alle hier genannten Phänomene sind beim Betrachten einer Transkription eines solchen Diskurses nicht mehr ohne Weiteres rekonstruierbar und können oftmals überhaupt nur verstanden werden, wenn entweder ausführliche Beschreibungen des Verlaufs und der Situation vorliegen oder eine Videoaufzeichnung des Diskurses vorliegt.

Dies sind nur einige Merkmale, die charakteristisch für die gesprochene Sprache sind. Diese Merkmale haben auf verschiedenen Ebenen Auswirkungen auf ihre Modellierung. Im folgenden Kapitel, das sich mit der computergestützten Transkription gesprochener Sprache als Methode beschäftigt, werden die erwähnten Merkmale im Unterkapitel „Problemfälle für die Modellierung von Sprachdaten“

---

17 Zur Deixis vgl. Bühler (1978: 177ff), Ehlich und Rehbein (1979) und Fillmore (1997).

18 Vgl. auch Fiehler et al. (2004).

ab Seite 43 noch einmal aufgegriffen und auf den Aspekt hin untersucht, inwiefern sie sich bei der Datenmodellierung für die computergestützte Verarbeitung als problematisch erweisen und wie mit diesen Problemen umgegangen werden kann.

## 2.2. Transkription

### 2.2.1. Anwendungsbereiche von Transkription

Der Begriff der Transkription wird in verschiedenen Wissenschaftsbereichen unterschiedlich verwendet. In der Genetik wird mit Transkription die Synthese von RNA anhand einer Vorlage bezeichnet – dies ist im Grunde aber der einzige Zusammenhang, in dem das Endprodukt einer Transkription keine irgendwie gear- tete Verschriftlichung ist.

Dabei muss bei dem Begriff der Transkription zusätzlich noch zwischen der Ver- wendung für den Vorgang der Transkription (auch *das Transkribieren*) und dem Endprodukt dieses Vorgangs (*die Transkription*, auch *das Transkript*)<sup>19</sup> un- terschieden werden:

„(...) *the term is just as frequently used (...) to denote the product of that act, that is, a document. Elsewhere, the term is not infrequently used to refer to the relationship between documents—one document may be said to be a transcrip- tion of another document.*“ (Huitfeldt und Sperberg-McQueen 2008: 296)

Die Vermischung der Begriffe für Vorgang und Produkt lässt sich nicht immer vollständig vermeiden, hier wird es aber in der Regel um den Vorgang des Trans- kribierens gehen, wenn von Transkription die Rede ist.

Auch für die Transkription, bei der als Endprodukt ein Dokument steht, gibt es noch eine Reihe von Bedeutungen, die im Rahmen dieser Arbeit keine Rolle spie- len: die Transkription in der Musik, also das Verschriftlichen eines Musikstückes (eines musikalischen Sachverhaltes) zumeist in Notenschrift, aber auch die in der Editionsphilologie und Literaturwissenschaft häufig angewandte Übertragung einer schriftlichen Quelle in eine andere Form.<sup>20</sup>

Aber auch die Transkription gesprochener Sprache findet in so unterschiedlichen

---

19 Schmidt (2004: 8) unterscheidet darüber hinaus die *Transkription* als „in *abstrakter* Weise die Ge- samtheit der modellierten Gesprächseinheiten und der Beziehungen zwischen ihnen, sie ist eine Menge von Daten“ von dem *Transkript* als „jede *konkrete*, visuell erfassbare, in irgendeiner Form geschlossene graphische Darstellung der modellierten Einheiten und Zusammenhänge“. Diese Trennung wird hier auch vorgenommen, allerdings wird von *Transkripten* kaum die Rede sein.

20 Vgl. Huitfeldt und Sperberg-McQueen (2008).

wissenschaftlichen und nichtwissenschaftlichen Kontexten mit so unterschiedlichen Voraussetzungen und Zielen statt, dass es wichtig ist, die für diese Arbeit relevante Form der Transkription aus dieser Obermenge herauszusortieren und zu definieren.

## **Sozialforschung**

In der qualitativen Interview- und Sozialforschung steht in den allermeisten Fällen die Rekonstruktion des gesprochenen Textes ohne phonologische oder konversationsanalytische Fragestellungen im Mittelpunkt des Interesses. Transkriptionskonventionen existieren selten, genauso wenig wie spezielle Modelle oder Datenformate für die entstehenden Transkriptionen. Transkriptionen werden hier in Standardorthographie und zumeist als einfache Textdateien angefertigt. Diese Praxis spiegelt sich auch in der Auswahl an Werkzeugen zur qualitativen Auswertung derartiger Transkriptionen wider: Populäre Programme wie MaxQDA<sup>21</sup>, Atlas.ti<sup>22</sup> oder TAMS-Analyzer<sup>23</sup> verarbeiten als Eingangsmaterial nur reine Text- oder Textverarbeitungsformate.<sup>24</sup> Auch die häufig verwendeten Transkriptionswerkzeuge wie Transana<sup>25</sup> oder f4 sind für die Produktion einfacher, linearer Textdateien konzipiert und können mit Phänomenen wie überlappenden Sprecherbeiträgen nur über Hilfskonstruktionen umgehen.

Wenn derartige Phänomene überhaupt mittranskribiert werden, so wird dafür meist ein *Transkriptionsschema* gewählt, dem kein weiteres Modell zugrunde liegt und das vollständig an der Oberfläche abbildbar ist.<sup>26</sup>

## **Sprachtechnologie**

In den Forschungszusammenhängen der *Computerlinguistik* steht in der überwiegenden Zahl der Fälle die Beschäftigung mit Methoden und Anwendungen

---

21 <http://www.maxqda.com/>

22 <http://www.atlasti.com/>

23 <http://tamsys.sourceforge.net/>

24 Eine Übersicht über die gängigsten Programme findet sich unter [http://onlineqda.hud.ac.uk/Which\\_software/reviews-of-sw.php](http://onlineqda.hud.ac.uk/Which_software/reviews-of-sw.php).

25 <http://www.transana.org/>

26 Vgl. Forsyth, Clarke und Lam (2008).



der *Sprachtechnologie* für automatische Spracherkennung und Sprachgenerierung (*Speech Recognition*<sup>27</sup> und *Speech Synthesis*<sup>28</sup>) im Vordergrund, wenn sich mit *gesprochener Sprache* beschäftigt wird.

Dort liegt zumeist ein spezieller Arbeitsablauf vor, der zwar sehr stark auf automatischen Verfahren fußt, in dem die eigentliche Transkription aber manuell erfolgt.

Die Sprachaufnahmen werden zumeist unter kontrollierten Bedingungen und für vordefinierte Szenarien erstellt. Dabei wird versucht, bestimmte Probleme wie überlappende Sprecherbeiträge von vorneherein zu vermeiden.<sup>29</sup>

Die Transkriptionen, die von diesen Sprachaufnahmen erstellt werden, sind zunächst einmal standardorthografisch, um eine eindeutige Zuordnung von sprachlichem Symbol und Aufnahme herstellen zu können. Diese Transkriptionen werden dann mit weiterer Information annotiert – insbesondere mit Verbindungen zu einem Aussprachelexikon. Entscheidend für die Nützlichkeit dieser Transkriptionen ist die Genauigkeit, mit der die Zuordnung von Zeitabschnitten in der Aufnahme mit der Transkription erfolgt; damit sich die angefertigten Transkriptionen für das Training und die Kontrolle automatischer Spracherkennung und Syntheseprogramme mittels empirischer Methoden der Phonetik eignen, muss die manuelle Transkription mit größter Sorgfalt vorgenommen und gewissenhaft kontrolliert werden.

Die Annotationen, die für Spracherkennung und -synthese relevant sind, können wesentlich komplexer werden als in den anderen erwähnten Disziplinen – insbesondere, wenn anspruchsvollere Aufgaben wie concept-to-speech oder Emotionserkennung angegangen werden.<sup>30</sup>

---

27 Vgl. Sagerer (1990).

28 Vgl. Dutoit (1997).

29 Ein Beispiel sind die „Knopfdruckdialoge“, die einen großen Teil der in Verbmobil (Wahlster 1997) verwendeten Dialoge ausmachen: Bei diesen Dialogen müssen die Gesprächsteilnehmer ihren Turn mittels eines Knopfdrucks einleiten – der jeweils andere Teilnehmer kann dann nicht sprechen und es kann zu keiner zeitlichen Überlappung kommen. Vgl. dazu Burger (1997).

30 Vgl. hierzu die Materialien auf <http://emotion-research.net/>.

Bei der Modellierung der Sprachdaten ergeben sich die Probleme eher für die multiple Annotation der transkribierten Daten. Schwierigkeiten, wie sie zum Beispiel durch überlappende Sprecherbeiträge entstehen können, spielen dagegen kaum eine Rolle, da bereits beim Design der Aufnahmeszenarien darauf geachtet wurde, diese zu vermeiden – vor allem, da sich das Audiosignal der überlappenden Sprecher nur unter erheblichem Aufwand wieder trennen lässt.

## Phonologie und Phonetik

Transkriptionen, die zur Untersuchung von Phonologie und Phonetik angefertigt werden, vereinen Merkmale der Transkriptionen aus der Sprachtechnologie und den im nächsten Abschnitt beschriebenen Transkriptionen, die sich stärker mit Aspekten des sprachlichen Handelns beschäftigen.

Die Transkription wird meistens in einem phonetischen Alphabet wie SAMPA oder IPA<sup>31</sup> angefertigt. Obwohl es die lautlichen Phänomene sind, die im Fokus der Untersuchung stehen, wird zumeist auch eine zusätzliche orthografische Transkription (als eigene Transkription oder Annotation) angefertigt, schon, um bestimmte Formen schneller auffinden zu können.

[6]

	86 87	88 89	90 91	92 93	94 95	96 97
Miguel [pho]	[ran.*je.βaə.*θo.pia.*mo.ɪo]					
Miguel [syll]	[CVC.*CV.CVG.*CV.CVG.*CV.CV]					
Miguel [orth]	alli). Han llevado ((unv.)) moto. Un tren. ((unv.)) El coche, el coche. El globo, el globo, globo. Flor.					

[7]

	98 99	100 101	102 103	104 105	106 107	108 109	110 111
Miguel [pho]	[ʔe.ʔel.*ka.jo]						
Miguel [syll]	[CV.CVC.*CV.CV]						
Miguel [orth]	(La pala).	El globo.	El globo, el globo.	(Se cayó).	((unv.))	El nene, el nene.	El nené no, no

Abbildung 10: Transkription mit Information zu Phonologie und Silbenstruktur.

Im Gegensatz zu den Transkriptionen in der Sprachtechnologie stammen die Ausgangsdaten häufiger auch aus nicht-kontrollierten Sprachsituationen. Die Transkriptionen werden ebenfalls manuell angefertigt, dabei wird häufig auf die Hilfe von auf diesen Zweck spezialisierte Transkriptionssoftware (insbesondere *Praat*, (Boersma und Weenink 2009)) zurückgegriffen, die messbare Parameter

31 Vgl. Gibbon, Mertins und Moore (2000), S. 658ff. (IPA) und S. 685ff. (SAMPA).

des akustischen Sprachsignals wie Amplitude oder Intonationsverlauf grafisch darstellen und damit die Transkriptionsarbeit erleichtern kann.

### **Gesprächsforschung, Pragmatik und Diskursanalyse**

Die wissenschaftlichen Disziplinen der Gesprächsforschung, der Pragmatik und der Diskursanalyse beschäftigen sich auf unterschiedlichen Ebenen und mit verschiedenen Zielsetzungen mit Aspekten sprachlichen Handelns.<sup>32</sup>

Für diese Aufgabe ist es, bezogen auf das Ausgangsmaterial der Analysen, in allen drei Disziplinen gewünscht, der spontanen, unbeeinflussten, „natürlich“ gesprochenen Sprache möglichst nahe zu kommen, also in dem auf Seite 16 erwähnten Kontinuum Mündlichkeit-Schriftlichkeit möglichst weit auf der Seite der Mündlichkeit zu liegen und auf der medialen Ebene *gesprochene* Sprache als Untersuchungsgegenstand zu haben.

Natürlich handelt es sich auch hier nicht um eine absolute Regel: Auch Konversation, die medial nicht gesprochen ist (z.B. Chat-Kommunikation im Internet) ist von Interesse, ebenso ist es keine Voraussetzung, dass die Kommunikation *Face-to-Face* stattfindet – auch Telefongespräche liegen durchaus im Forschungsinteresse.

*„Ko-Präsenz‘ von Sprecher(n) und Hörer(n) ist für die Ethnographie der Kommunikation und damit für die Anthropologie eine conditio sine qua non für die authentische und unmaskierte ‚face-to-face‘-Kommunikation, die Wiege und Basis der menschlichen Lebenskultur und der kulturellen Reproduktion. (...)*

*Das Axiom (...) ist (...) durch die vielen multimedialen Facetten kommunikativer Austauschmöglichkeiten eher zur Legende geworden.“ (Dittmar 2004: 35)*

Dies führt dazu, dass potentiell alle der ab Seite 18 aufgeführten „Besonderheiten der Mündlichkeit“ in den aufgenommenen Gesprächen zum Tragen kommen können. Die Gespräche, die wie die Chat-Kommunikation nicht als Audio- oder

---

32 Die Disziplinen der Gesprächsforschung, Pragmatik und Diskursanalyse gleichberechtigt im selben Abschnitt zu behandeln, würde von ihren Vertretern sicherlich kritisiert werden. Wenn es allerdings darum geht, die Anforderungen an Transkriptionssysteme, -modelle und deren Annotation für diese Disziplinen zu bestimmen, so liegen sie doch so dicht beieinander, dass sich dieser Schritt gut rechtfertigen lässt.

Videoaufnahme vorliegen, bilden insofern einen Sonderfall, als dass sie in der Regel nicht mehr zusätzlich transkribiert werden, da sie bereits in schriftlicher Form vorliegen.

Bezogen auf die Annotation der transkribierten Gespräche existieren auch einige Gemeinsamkeiten: In den drei Disziplinen ist das Festhalten möglichst vieler Aspekte des Gesprächskontextes ein gemeinsames Desideratum. So ist es nicht nur entscheidend, welcher Sprecher zu welchem Zeitpunkt welche Äußerung tätigt, es ist auch wünschenswert, möglichst viel über Ethnografie und das komplette Setting zu erfahren:

- Welche Personen sind anwesend?
- Was weiß man über diese Personen?
- Wo findet die Kommunikation statt?
- Welche nonverbalen Elemente begleiten eine Äußerung?
- Welche paraverbalen Handlungen werden getätigt?
- Wann werden wie lange Pausen gemacht?
- Wie reagieren die Anwesenden auf eine Äußerung (nonverbal)?
- Was geschieht gleichzeitig in der Umgebung?

Diese Liste ist keineswegs vollständig. Die Aspekte, die sich über den ganzen Gesprächsverlauf nicht ändern, sind eher dem Bereich der *Metadaten* zuzuordnen und werden nicht immer in der Transkription miterfasst. Andere Aspekte, insbesondere Gestik und Mimik, werden unter dem Begriff *Multimodale Annotation* oder *Transkription* geführt, weil sie das Medium der Sprache verlassen, aber trotzdem kommunikative Funktionen erfüllen.

Darüber hinaus sind die Annotationsebenen, die linguistische Kategorien betreffen und beispielsweise auch in der Sprachtechnologie verwendet werden, natürlich auch hier relevant: Annotationen zu Morphologie, Syntax, thematischer- und Informationsstruktur – sie alle können in diesen Forschungsbereichen relevant werden.

In Deutschland sind in diesen Forschungsbereichen insbesondere zwei Transkriptionssysteme verbreitet: Das *Gesprächsanalytische Transkriptionssystem* (GAT, Selting et al. 1998) und die *Halb-Interpretative Arbeits-Transkription* (HIAT, Ehlich und Rehbein 1976; Rehbein et al. 1993 und Rehbein et al. 2004).

Bereich	Vorgang	Ausgangsmaterial	Ergebnis
<b>Genetik</b>	Synthese	DNA	RNA
<b>Musik</b>	Verschriftlichung	Musikalischer Sachverhalt, Dokument in Notenschrift	Dokument in Notenschrift
<b>Soziologie, Interviewforschung</b>	Orthografische Transkription	Gesprächsaufnahmen	Textdokumente
<b>Sprachtechnologie (Spracherkennung, Sprachsynthese)</b>	Orthografische Transkription	Sprachaufnahmen aus zumeist klar definierten Szenarien	XML- und spezialisierte Formate
<b>Phonologie, Phonetik</b>	Orthografische Transkription und Phonologische Transkription	Sprachaufnahmen aus kontrollierten und natürlichen Gesprächssituationen	XML- und spezialisierte Formate
<b>Gesprächsforschung, Diskursforschung</b>	Variiert	Sprachaufnahmen (Audio, auch Video) von möglichst natürlichen Gesprächssituationen	XML- und spezialisierte Formate

Abbildung 11: Die Methode der Transkription in verschiedenen Zusammenhängen.

## 2.2.2. Grundlagen der Transkription in Gesprächsforschung und Diskursanalyse

Für die Art der Transkription, mit der sich diese Arbeit hauptsächlich befasst, und die in der obigen Aufstellung am ehesten als typisch für Gesprächsforschung und Diskursanalyse gelten kann, müssen einige theoretische Grundlagen beleuchtet werden, die auf die praktische Arbeit an und mit diesen Transkriptionen direkten Einfluss haben.

### Theoriehaftigkeit von Transkriptionen

Transkriptionen können niemals theorieneutral sein. Dittmar (2004: 45), Bezugnehmend auf Ochs (1979) sagt dazu:

*„Erkenntnisleitend für die Sprachbeschreibung ist die soziolinguistische Einsicht, dass die Vielfalt des regelhaften Sprechens in einer Kommunikationsge-*

*meinschaft nicht durch die Intuition eines linguistischen Experten, sondern nur durch sorgfältig erhobene Korpora erfasst werden kann.“*

Obwohl diese Einsicht heute immer weniger Widerspruch erfährt, so ist die zunehmende Bedeutung der korpusbasierten Methode und der Korpuslinguistik doch ein relativ junges Phänomen. Die Tatsache, dass Korpora als Basis für die Untersuchung gesprochener Sprache immer wichtiger werden, bedeutet zwangsläufig, dass im benötigten Methodenrepertoire das Transkribieren einen wichtigen Stellenwert hat, da die Transkriptionen die Arbeitsgrundlage für viele Linguisten darstellen.

Die Beschäftigung mit dem Thema der Transkription fokussiert hauptsächlich auf den interpretativen Charakter des Transkribierens und die daraus ableitbaren Aspekte wie Selektivität und Theoriehaftigkeit.

Mit Selektivität ist zunächst gemeint, dass eine Transkription niemals alle Aspekte eines beobachteten und/oder aufgezeichneten Gesprächsereignisses erfassen kann. Diese Feststellung ist sofort einleuchtend: Bereits bei der Betrachtung von Besonderheiten der gesprochenen Sprache wurde hier festgestellt, dass unterschiedliche Anforderungen an die Genauigkeit gestellt werden können, was die Abbildung von lautlichen Merkmalen in einer Transkription betrifft. Forscher, bei denen diese Merkmale nicht im Fokus des Interesses stehen, würden sie bei der Transkription einer Gesprächsaufnahme von vorneherein unberücksichtigt lassen und damit eine Selektion vornehmen.

Diese Selektion leitet direkt auf die *Theoriehaftigkeit* über: Die Kriterien, nach denen die Selektion vorgenommen wird, sind in der Regel von der Fragestellung und der Theorie bestimmt, der der Forscher nachgeht.

Die Tatsache, dass eine theoriegeleitete, selektive Transkription nur einen eingeschränkten Blick auf das Gesprächsereignis zulässt, ist so lange kein Problem, wie die Entscheidungen nachvollziehbar dokumentiert werden. Ein Forscher, der mit Daten arbeitet, die von anderen Forschern mit anderen Forschungsinteressen erhoben wurden, muss sich mit deren Entscheidungen vertraut machen können. Sonst kann es zu falschen Generalisierungen kommen, die auf dem Fehlen be-

stimmter Informationen beruhen.

Die Theoriehaftigkeit von Transkriptionen hat auch Auswirkungen auf die Wahl von Schrift-, Notations- und Transkriptionssystem: So werden Forscher, die an einer präzisen zeitlichen Einordnung der gesprochenen Einheiten interessiert sind, eine Partiturschreibweise einer Liste von Äußerungen vorziehen, Forscher, die an Phonologie interessiert sind, werden ihre Transkriptionen eher in IPA als in Standardorthografie vornehmen; im letzten Fall lässt die Auswahl einfache Rückschlüsse auf den wahrscheinlich zugrunde liegenden Interessenschwerpunkt zu, es ist aber immer vorzuziehen, diesen für die erhobenen Daten explizit zu machen.

*„Was wie (in welcher grob- oder feingranulierten Form) mit welcher Qualität und Menge von Symbolen verschriftlicht wird, ist eine Funktion der wissenschaftlichen Fragestellung.“ (Dittmar 2004: 51)*

Die Theoriehaftigkeit der Transkription wird auch von den Vertretern der verbreitetsten Transkriptionssysteme nicht grundsätzlich infrage gestellt.<sup>33</sup>

### **Prozesshaftigkeit von Transkription**

Wenn ein Transkribent mit der Erstellung einer Transkription abgeschlossen hat, so bedeutet dies nicht, dass die Transkription in diesem Moment „fertig“ ist.

Lehnen und Gulich (1997: 112) sagen dazu:

*„Dabei bleibt die Transkription des mündlichen Diskurses ein potentiell unabschließbarer Prozess. [...] Eine Transkription ist immer verfeinerbar, sie muß immer wieder ‚gegengehört‘ werden.“*

Der geschilderte Umstand wird wohl nicht umsonst als *potentiell* unabschließbar bezeichnet: In dem Moment, in dem der Ersteller entscheidet, keine weitere Arbeit in die Verfeinerung einer Transkription zu investieren, ist der Prozess für ihn sehr wohl abgeschlossen. Es bleibt aber richtig, dass jede Transkription immer weiter verfeinerbar ist, dass – auch und insbesondere durch den interpretativen Charakter der Transkription – es immer eine abweichende Meinung zu bestimm-

---

33 Vgl. auch Schmidt (2004: 17f).

ten Entscheidungen und Festlegungen in einer Transkription geben kann.

Schmidt (2007: 237) bemerkt:

*„[Das Transkriptionssystem, KW] HIAT trägt dem bereits in seinem Namen Halbinterpretative Arbeitstranskriptionen Rechnung, GAT in der Unterscheidung zwischen Basis- und Feintranskript und dem sog. Zwiebelprinzip.“*

Auch die Prozesshaftigkeit von Transkription wird also – genau wie die Theoriehaftigkeit – von den Autoren der Transkriptionskonventionen nicht infrage gestellt. Sie wirkt sich unmittelbar auf den *Workflow* aus, der sich bei der Transkription, Annotation und Zusammenstellung eines Korpus ergibt, und er hat auch Auswirkungen darauf, von welchen Voraussetzungen man bei der Modellierung dieser Ressourcen ausgehen muss: Ein Datenmodell muss demnach in der Lage sein, jederzeit Änderungen an der Transkription und an den daran vorgenommenen Annotationen zuzulassen.

Ehlich (1992: 52) bemerkt dazu:

*„Revisionsfähigkeit ist also eine wichtige Voraussetzung für ein sachangemessenes Transkriptionsverfahren.“*

Wie sich später zeigen wird, ist dies keineswegs eine triviale Anforderung.

### **Funktion und Aufgabe der Transkription**

Wird nach der Funktion der Transkription gefragt, so ist die *Verschriftlichung* von gesprochener Sprache zumeist der Ausgangspunkt der Erklärung. Obwohl der Begriff im Grunde zutreffend und als kurze Zusammenfassung der einleuchtendste ist – es wird schließlich eine bestimmte Verschriftlichung einer Aufnahme einer mündlichen Kommunikation vorgenommen – so kann durch ihn schnell der irriige Eindruck entstehen, man könne gesprochene Sprache einfach und quasi verlustfrei in geschriebene Sprache überführen. Die in Kallmeyer et al. (1974: 45) vertretene Auffassung *„Text ist die Gesamtmenge der in einer kommunikativen Interaktion auftretenden Signale.“* kann inzwischen als überholt gelten.



Für den reinen, textuellen<sup>34</sup> Inhalt wird es in Einzelfällen möglich sein, eine derartige Überführung vorzunehmen, und für einige der auf Seite 32 erläuterten Forschungszusammenhänge ist eine derartige Überführung auch oft ausreichend. Für die meisten Anwendungsfälle gilt aber, dass eine Transkription niemals alle Aspekte der Aufnahme und schon gar nicht alle Aspekte des tatsächlichen Gesprächsereignisses abbilden kann – einige Gründe dafür wurden schon im vorhergehenden Unterkapitel näher erläutert.

Die Transkription unterscheidet sich damit von Verschriftlichungsformen wie Wortprotokollen, die sich an den Rezeptionsnormen der Schriftlichkeit orientieren und die spezifischen Aspekte der Mündlichkeit nicht berücksichtigen oder absichtlich tilgen.

Schmidt (2004: 50) weist darauf hin, dass die Funktion der Schrift im Prozess der Transkription nur die eines Mittels der Modellbildung ist und nicht deren zentraler Aspekt.

Redder, die von der Transkription als „*Verschriften von authentisch Gesprochenem*“ spricht (Redder 2001: 1038), betont besonders den Aspekt der *Verdauerung*, die das Gesprochene von seiner Flüchtigkeit befreit und damit erst eine Analyse ermöglicht. Eine derartige Verdauerung wird natürlich auch bereits in dem Moment vorgenommen, in dem eine Aufnahme des Gespräches vorgenommen wird. Wichtiger ist die Feststellung, dass die Transkription als Grundlage für die Dokumentation und Untersuchung von mündlicher Kommunikation verstanden wird:

*„Anders als Schrift bzw. ein Schriftsystem ist ein Transkriptionssystem nicht für kommunikative, sondern für analytische Zwecke ausgebildet. Dementsprechend gehört das Transkribieren nicht zu den verallgemeinerten Kulturtechniken einer Sprachgemeinschaft, sondern zu den – heute weitgehend selbstverständlichen – wissenschaftlichen Arbeitstechniken empirischer Kommunikationsforschung (...)“ (Redder 2001: 1038)*

---

34 Text gehört wiederum zu den in der Linguistik am stärksten umstrittenen Begriffen (vgl. dazu Klemm (2002)). Der „textuelle Inhalt“ meint an dieser Stelle den *mündlich in Worten geäußerten* Anteil des Gespräches.

Dittmar (2004: 49f.) formuliert es so:

*„Auf der Folie der Transkription werden wissenschaftliche Aussagen über das Untersuchungsgebiet ‚sprechsprachliche Kommunikation‘ in ‚wahr‘ und ‚falsch‘ geschieden und empirische Belege von Beschreibungen und Erklärungen als Bedingung der Möglichkeit von ‚Evidenz‘ (Beweisen) verstanden. Daher wird die Dokumentation flüchtiger kommunikativer Austauschprozesse zum Zwecke wissenschaftlicher Analyse als die wesentliche Aufgabe der Transkription verstanden.“*

### **Modellhaftigkeit von Transkription**

Die angeführten Charakteristika von Transkriptionen als Forschungsgrundlage für die Beschäftigung mit gesprochener Sprache, insbesondere Selektivität und Theoriehaftigkeit, belegen den Modellcharakter von Transkription. Schmidt (2004: 49), der den Gedanken von Transkription als Modellierung ausführlich herleitet, sagt dazu:

*„Da die Begriffe Modell und Theorie (oder auch Interpretation) weitestgehend synonym gebraucht werden, ist es zunächst nur eine begriffliche Änderung, wenn hier von Transkription als Modellierung statt von Transkription als Theorie gesprochen werden soll.“*

Transkriptionen als (mithilfe des Computers erzeugte) Produkte sind also nicht einfach eine Verschriftlichung, sondern *eine mögliche Modellierung* eines Gesprächsereignisses. Dieses Modell ließe sich wieder in unterschiedlichen Weisen visualisieren – eine entsprechende Visualisierung könnte eher einer Verschriftlichung im eigentlichen Sinne nahekommen.

Bei der Erstellung eines Modells für die computergestützte Transkription gilt es, einige charakteristische Merkmale besonders zu berücksichtigen. Auf diese Faktoren soll im folgenden Kapitel genauer eingegangen werden.

## **2.3. Problemfälle für die Modellierung von Sprachdaten**

Die Suche nach dem besten Modell für die Modellierung von gesprochener Sprache ist ein immer wiederkehrendes Thema in der texttechnologischen und computerlinguistischen Forschung. Es werden in regelmäßigen Abständen neue Vorschläge für Modelle und Formate entwickelt und vorgestellt, um die klassischen Problemfälle, die bei der Modellierung derartiger Informationen auftauchen, zu behandeln.

Interessanterweise sind die Probleme, die auftreten, in ihrer Anzahl relativ überschaubar und auch die vorgeschlagenen Lösungsansätze lassen sich auf wenige generische Ansätze zurückführen, die unterschiedlich zu Ende geführt werden. Im Folgenden sollen die Probleme vorgestellt werden, um im nächsten Kapitel näher auf drei elementare Modellierungsansätze eingehen zu können.

### **2.3.1. Überlappende Hierarchien**

Obwohl einige der Probleme für die Modellierung von Sprachdaten auch bei Daten geschriebener Sprache auftreten, finden sich die schwierigsten eindeutig oder verschärft auf der Seite der gesprochenen Sprache. Ein besonders kompliziertes Problem stellt die Modellierung von überlappenden Hierarchien dar.

Überlappende Hierarchien kommen durchaus auch in schriftlichen Texten vor; schon in einfachsten, gedruckten Dokumenten überlappen sich meistens die physische Struktur (z.B. Seitenzahlen) und die linguistische Struktur (z.B. Sätze). Will man eine elektronische Edition eines solchen gedruckten Dokumentes anfertigen, so muss man sich mit dieser Form der Überlappung auseinandersetzen. Modelle, die einen Text immer als eine strikte Hierarchie von Inhaltsobjekten ansehen (gemäß der ab Seite 55 vorgestellten OHCO-These), können bereits mit diesem nicht sehr komplizierten Spezialfall nicht mehr umgehen. Derart eingeschränkte Modelle kommen deshalb in der Praxis so gut wie nicht vor. Das vermutlich verbreitetste Modell für elektronische, schriftliche Texte, die TEI-Guidelines, widmen sich den überlappenden Hierarchien an verschiedenen Stellen. In dieser Arbeit werden sie im Kapitel „Die TEI-Guidelines“ ab Seite 57 eingehend betrachtet.

Das Problem der überlappenden Hierarchien verschärft sich bei Transkriptionen gesprochener Sprache noch: Während es bei geschriebenen Texten im allgemeinen nicht zur Überlappung von tatsächlichen Primärdaten kommt<sup>35</sup>, so kommt es in Transkriptionen gesprochener Sprache häufig zu überlappenden Sprecherbeiträgen: Es finden also quasi zwei Stränge von Primärdaten gleichzeitig statt. In der Betrachtung der Besonderheiten von gesprochener Sprache ist dies unter dem Punkt „Zeitstruktur und Organisation“ auf Seite 19 angeführt, in der Abbildung auf Seite 20 ist ein entsprechendes Partiturbeispiel zu sehen.

In geschriebenen Texten lassen sich auch in hierarchischen Modellen strukturelle Überlappungen mit Hilfe bestimmter Hilfskonstruktionen abbilden – ab Seite 61 werden einige näher beschrieben. Für gesprochene Sprache, in der sich Sprecherbeiträge überlappen, sind solche Konstruktionen auch möglich und vorgeschlagen worden. Sie sind dann allerdings – wie sich zeigen wird – in mehrererlei Hinsicht sehr weit vom ursprünglichen Modell entfernt, sodass die Nachteile, die man sich damit erkaufte, die eigentlichen Vorteile des Modells (insbesondere effiziente Verarbeitbarkeit und Lesbarkeit) bereits eliminieren oder übertreffen.

### **2.3.2. Zeitbezug**

Die Modellierung der Überlappung von Sprecherbeiträgen wird häufig über einen Bezug zu einer einheitlichen Zeitachse gelöst. Der zeitliche Bezug von Gesprächsanteilen ist aber bereits ohne die Problemfälle von Überlappungen für gesprächsanalytische Untersuchungen relevant. Parameter wie Sprechgeschwindigkeit oder die Länge von (insbesondere untranskribierten) Pausen lassen sich zwar mit absoluten Angaben auch in Transkriptionen vermerken, die keinen direkten Bezug zu einer Zeitachse haben, in Transkriptionen, die mit einer Zeitachse aligniert sind, muss diese Information aber nicht erst expliziert werden.

Ein solcher Zeitachsenbezug erlaubt es auch erst, eine Synchronisation zwischen einer existierenden Aufnahme und der angefertigten Transkription herzustellen, um einfacher entsprechende Transkriptionspassagen an der Aufnahme überprü-

---

35 Ausnahmen bestätigen auch hier die Regel: In Editionen und Transliterationen in verschiedenen Schriftsystemen können sich auch in geschriebenen Texten Primärdaten überlappen.

fen zu können.

### 2.3.3. Überlappende Annotationen

Probleme mit Überlappungen von Annotationen gibt es für geschriebene und gesprochene Sprache in ähnlicher Form. So können beispielsweise bereits Überlappungen auf einer Annotationsebene auftreten, wenn diskontinuierliche Einheiten („Das *habe* ich doch *gesagt*“) vorliegen. Werden Annotationen auf mehr als einer Ebene vorgenommen, sind Überlappungen schon fast unausweichlich.

Möchte man Primärdaten und Annotationen in derselben Datei ablegen, so hat man mit denselben Problemen zu tun, die sich bei überlappenden Hierarchien innerhalb der Primärdaten auch ergeben.

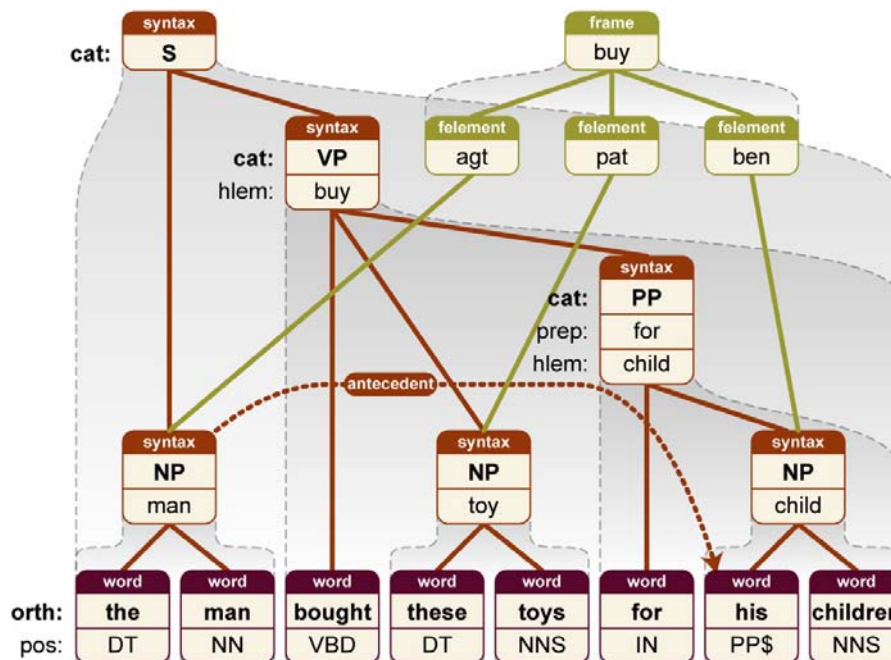


Abbildung 12: In diesem Beispiel einer NITE-Annotation bezieht sich die Frame- auf die Syntax-Annotation und führt zu Überlappungen. (aus Carletta et al. 2003: 2)

Je mehr dieser Überlappungen in Primärdaten und Annotationen auftreten, desto untauglicher werden Modelle, die solche Überlappungen im Text nur mittels Hilfskonstruktionen wie beispielsweise Milestones berücksichtigen können.

### 2.3.4. Lautung vs. Schreibung

Der Aspekt, der für die Modellierung von Sprachdaten zu einem Problem wird, ist in der Übersicht über die Besonderheiten der Mündlichkeit unter „Lautbil-

derung (Phonologie und Phonetik)“ ab Seite 21 beschrieben. Dort ist bereits erwähnt, dass sich mit Mitteln der Standardorthografie niemals eine vollständige Abbildung aller phonischen Parameter einer gesprochenen Lautfolge erreichen lässt.

Für eine einfache Übertragung von gesprochener Sprache wird im Kontext der Transkriptionskonventionen der Gesprächsanalyse<sup>36</sup> daher – wie bereits erwähnt – der Kompromiss der literarischen Umschrift gewählt. Diese verfährt zunächst nach dem phonographischen Prinzip:

*„Den Phonemen lassen sich regelhaft Segmente des Geschriebenen, nämlich Grapheme, zuordnen (phonographisches Prinzip).“ (Fabricius-Hansen et al. 2006: 63)*

Dieses Prinzip wird aber – im Kontrast zu rein orthografischen Transkriptionen<sup>37</sup>, die sich an der regelhaften Beziehung zwischen gesprochener Sprache und Schrift orientieren – eindeutig höher gewichtet als die weiteren, insbesondere als das morphologische Prinzip.

Damit wird erreicht, dass sich bestimmte phonologische Besonderheiten in der Aussprache besser darstellen lassen, als dies mit Mitteln der Standardorthografie möglich wäre, was den Vorteil mit sich bringt, dass solche Besonderheiten direkt beim Lesen einer Transkription intuitiv erfasst werden können. Dieser Vorteil wird allerdings mit zwei grundlegenden Problemen erkaufte, auf die auch Schmidt (2004: 63f) hinweist. Das erste Problem ist die mangelnde Verbindlichkeit von literarischer Umschrift:

*„Da es für die literarische Umschrift (...) kein verbindliches Regelwerk gibt, bleiben (...) Ungenauigkeiten hinsichtlich phonographischer Prinzipien bestehen.“*

Zum anderen verliert man durch die literarische Umschrift die Information, um

---

36 Hier sind insbesondere die Transkriptionskonventionen von HIAT, GAT und DIDA (vgl. <http://www.ids-mannheim.de/prag/dida/>) gemeint.

37 In den TEI-Guidelines (siehe Seite 56 ff.) wird für die Transkription gesprochener Sprache eine orthografische Transkription vorgesehen. Ebenso wie in Verbmobil werden Aussprachevarianten gesondert vermerkt (vgl. Burger und Kachelrieß (1996)).

welches lexikalische Wort in der Standardorthografie es sich handelt – und damit die Möglichkeit, auf dem Volltext einer Transkription bestimmte Wörter einfach aufzufinden.

Dieses Problem wird in den genannten Transkriptionssystemen zwar erkannt, für seine Lösung werden aber keine Vorschläge gemacht. Es würden sich verschiedene Lösungsmöglichkeiten anbieten:

Es könnte versucht werden, für die literarische Umschrift ein verbindliches Regelwerk zu erstellen. Damit würden wenigstens die Ungenauigkeiten, die bei der Anwendung phonographischer Prinzipien entstehen, eliminiert. Damit würde aber ein entscheidender Vorteil der literarischen Umschrift über Bord geworfen, nämlich der, dass man sich für deren Anwendung weitestgehend (und viel stärker als bei der Standardorthografie) auf die eigene Intuition verlassen kann.

Doch selbst mit einem entsprechenden Regelwerk ließe sich immer noch kein Lexikon erstellen, das einfach Wörter in literarischer Umschrift in Standardorthografie überführt, da auch die Segmentierung in Wörter in der literarischen Umschrift nicht unbedingt gegeben ist:

Och nee, dat jiwet ja nich.

(Demokorpus, Tropfsteinhöhle)

Eine automatische Übersetzung von „jiwet“ in „gibt es“ gelänge auch mit einem Regelwerk für die literarische Umschrift nicht.

Die erfolgversprechendste Lösung ist es, zur Transkription in literarischer Umschrift eine Transliteration (oder eine weitere Transkription) in Standardorthografie vorzunehmen und diese möglichst genau zu alignieren. So lassen sich Suchen auf der Ebene der Standardorthografie vornehmen und gleichzeitig die entsprechenden Abschnitte der Transkription in literarischer Umschrift auffinden. Dass dieser Weg nicht häufiger gegangen wird, liegt sicherlich daran, dass sich der ohnehin beträchtliche Aufwand der Transkription durch diesen zusätzlichen Arbeitsschritt noch einmal erhöht.

Ein weiterer Aspekt, der bei der Übertragung von Lautung in Schreibung zu einem Problem führt, ist der der Prosodie. Zwar lassen sich mit der gewählten lite-

rarischen Umschrift Besonderheiten der Phonologie in Grenzen erfassen, die Aspekte der Prosodie wie Akzent, Intonation, Quantität und Pausen werden durch sie aber nicht erfasst – dies ist auch insofern konsequent, da sie die Ebene des rein Sprachlichen bereits verlassen und eher auf der paraverbalen Ebene anzusiedeln sind.

Auch für diesen Aspekt werden in gesprächsanalytischen Transkriptionssystemen keine klaren Lösungsvorschläge gemacht. Es wird überwiegend vorgeschlagen, besonders auffällige Ausprägungen prosodischer Eigenschaft in einer Annotation zu vermerken – was allerdings niemals zu einer präzisen Beschreibung prosodischer Merkmale führen kann.

Dies ist aber auch im Kontext der computergestützten Arbeit nicht unbedingt nötig: Sofern digitale Aufnahmen der Gespräche, die den Transkriptionen zugrunde liegen, vorliegen, lassen sich bestimmte prosodische Eigenschaften mit Hilfe von existierender Software sehr gut automatisch erkennen und visualisieren. Sie sind dann zwar nicht Teil der Transkription und erschließen sich daher nur über Umwege für die Suche, sie lassen sich aber bei der qualitativen Auswertung von Transkriptionen jederzeit schnell zurate ziehen. Wie dies zum Beispiel aussehen kann, wird in der Beschreibung des EXMARaLDA-Systems ab Seite 96 gezeigt.

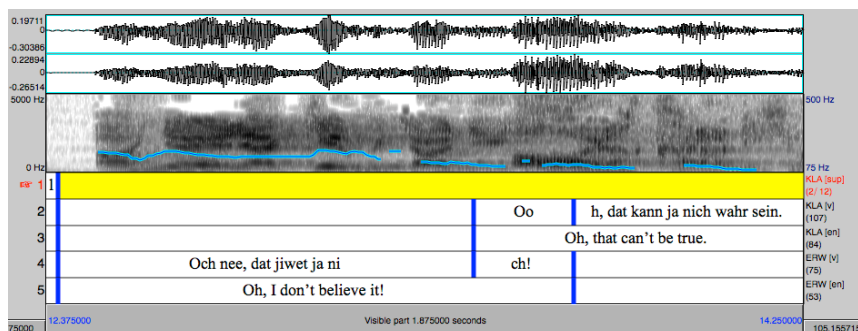


Abbildung 13: Automatische Berechnung der Intensität eines Audiosignals mit alignierter Transkription in Praat. (Demokorpus, Tropfsteinhöhle)

### 2.3.5. Segmentierung

Ein wiederkehrendes Problem, das mit Transkriptionskonventionen und der Modellierung gesprochener Sprache auftritt, ist die Frage, welche Einheiten oder



Segmente in einer Aufnahme und der dazu angefertigten Transkription existieren, welche für die Analyse gewünscht und benötigt und welche unerwünscht sind, wie und ob man sie kennzeichnet und wie man sie in das Modell integriert.

Würde man sich bei der Festlegung der Einheiten, die in einer Transkription existieren können, an der Domäne der geschriebenen Sprache orientieren, so könnte man beispielsweise auf für die Schriftsprache konstituierende Einheiten wie *Wort*, *Phrase* und *Satz* zurückgreifen.

*„Wortformen sind im laufenden Text durch Zwischenräume voneinander getrennt. Die geschriebene Sprache bietet damit ein Mittel zur einheitlichen Segmentierung von Texten in ihre Grundbausteine, wie es die gesprochene Sprache nicht kennt. Andererseits besteht damit in der Regel ein Zwang, in jedem Einzelfall kenntlich zu machen, wo eine Grenze zwischen Wortformen liegt.“*  
(Fabricius-Hansen et al. 2006: 77)

Die Tatsache, dass man in dieser Domäne so einfach auf sinnvoll nutzbare Einheiten stößt, täuscht aber darüber hinweg, dass auch diese Begriffe nicht unumstritten sind. So merkt die Dudenredaktion an,

*„(...)dass sich hinter der Bezeichnung Satz nicht ein einzelner Begriff, sondern eine ganze Familie sich überschneidender Begriffe steht.“* (Kunkel-Razum und Münzberg 1995)

Obwohl es bei Begriffen wie *Satz* oder *Wort* für die geschriebene Sprache also unterschiedliche Definitionen gibt, sind doch die Begriffe *an sich* weitestgehend etabliert. In der Orthografie werden regelhaft Satzgrenzen durch Interpunktionszeichen markiert.

Für die Domäne der gesprochenen Sprache existieren für verwandte Konzepte je nach zugrunde liegender Schule häufiger mehrere verschiedene Begriffe. Da die Grenzen von Segmenten der gesprochenen Sprache durch unterschiedliche Eigenschaften markiert werden können (Intonation, Morphologie, Syntax), orientieren sich die unterschiedlichen Betrachtungen gesprochener Sprache auch unterschiedlich, was die Segmentierung angeht. Dies erschwert einerseits die Kommunikation über den Gegenstand, auf der anderen Seite verfügen diese Be-

griffe häufig über eine bessere Trennschärfe gegenüber Begriffen wie *Satz*, die verwendet werden, obwohl ihnen unterschiedliche Bedeutungen zugeschrieben werden.

Die Einheit *Satz* wird als solche in Beschreibungen der gesprochenen Sprache beispielsweise so gut wie nicht verwendet. Im Transkriptionssystem HIAT<sup>38</sup> wird die Einheit, die einem *Satz* am nächsten kommt, *Äußerung* genannt. Diese Einheiten entstehen durch das Einfügen der aus der Schriftsprache bekannten Interpunktionszeichen, und zwar „in ihrer üblichen Weise“ (Ehlich und Switalla 1976: 23).

An welcher Stelle diese *Äußerungsendzeichen* eingefügt werden, wird durch eine Typologie der Äußerungsakte festgelegt, die bei Rehbein (1995) beschrieben ist. Vereinfachend ausgedrückt orientiert sich eine Äußerung an der sprachlichen Handlung (dem *Handlungsprozess*), die von dem sprechenden Teilnehmer vollzogen wird. Sie ist stark von den Bedingungen, die typischerweise in einer Gesprächssituation eine Rolle spielen (Ko-Präsenz, situative Kontexte, Vorwissen), abhängig.

Damit werden auch Einheiten der Transkription zu Äußerungen, die in keinem klassischen Sinne einem grammatischen Satz entsprechen – unter anderem sind natürlich auch die unter den in „Besonderheiten der Mündlichkeit“ ab Seite 18 genannten Merkmale wie abgebrochene Äußerungen und Ellipsen in diesem Sinne Äußerungen.

Um die Grenzen solcher Äußerungen feststellen zu können, spielen Merkmale der Morphosyntax und der Intonation eine Rolle. Äußerungen nach HIAT werden bei dieser Identifikation noch in verschiedene *Äußerungsmodi* eingeteilt, die anhand des vergebenen Äußerungsendzeichens identifizierbar sind: Assertionen (Punkt), Fragen (Fragezeichen), Ausrufe, Anreden und Aufforderungen (Ausrufezeichen), Abbrüche (Auslassungspunkte) und Äußerungen *ohne Modus* (hochgestellter Punkt).<sup>39</sup> Sofern Äußerungen nach den Regeln der Grammatik der ge-

---

38 Vgl. Ehlich und Rehbein (1976), Ehlich und Rehbein (1979), Ehlich und Rehbein (1979b).

39 Konkrete Anweisungen finden sich in Rehbein et al.(2004: 18ff).

schriebenen Sprache gemacht werden, entspricht auch die Segmentierung in Äußerungen derjenigen in Sätze in der Schriftsprache.

In HIAT existieren noch weitere Einheiten, in die sich eine Transkription segmentieren lässt: Unterhalb der beschriebenen Äußerungen existieren Wörter. Diese werden wie Wörter in Standardorthografie behandelt, obwohl sie – wie erwähnt – wahlweise auch in literarischer Umschrift niedergeschrieben werden können. Das bedeutet vor allem, dass Leerzeichen die Grenzen zwischen Wörtern markieren. Unter dem *Wort* existieren noch die Einheiten *Silbe* und *Phonem*, die ebenfalls den Regeln der Standardorthografie folgen. Oberhalb der Äußerung existiert die Einheit des *Turns*, der – vereinfachend ausgedrückt – die Äußerungen eines Sprechers bis zu einem Sprecherwechsel markiert.

Zu diesen Segmenten, die sich den phonologischen Bestandteilen widmen und in eine hierarchische Ordnung Turn -> Äußerung -> Wort -> Silbe -> Phonem bringen lassen, kommen noch Segmente, die bestimmte Besonderheiten eines Abschnittes markieren und sich mit den bereits genannten Segmenten potentiell überlappen können. Morphologische Segmente sind in den HIAT-Konventionen nicht vorgesehen, Informationen zur Morphologie können der Transkription in Form einer morphologischen Transliteration hinzugefügt werden.<sup>40</sup>

In den Transkriptionskonventionen zum Transkriptionssystem GAT (Selting et al. 1998) spielen prosodische Eigenschaften der gesprochenen Sprache eine größere Rolle als bei HIAT, pragmatische Eigenschaften dagegen eine geringere.

Entsprechend wird auch das Segment, das oberhalb von Wörtern angesiedelt ist, bei GAT anders definiert, nämlich als Phrasierungseinheit bzw. Intonationsphrase.<sup>41</sup> Die Grenzen einer Intonationsphrase werden – wie es ihr Name andeutet – durch prosodische Merkmale wie Tonhöhenbewegungen definiert und in der Transkription markiert.

Auch in GAT existiert eine Einheit *Turn*, deren Beschreibung einen breiteren

---

40 Vgl. Rehbein et al. (2004: 61ff).

41 In der neuen, noch unveröffentlichten Fassung der GAT-Richtlinien („GAT 2.0“) wird der Begriff *Phrasierungseinheit* von *Intonationsphrase* abgelöst.

Raum einnimmt als die der gleichnamigen Einheit bei HIAT, sowie die Einheiten Wort, Silbe und Phonem, die sich wie bei HIAT in eine Hierarchie einordnen lassen.

Darüber hinaus umfasst es eine große Anzahl Einheiten, die sich auf nicht-segmentale, prosodische Phänomene beziehen und deren Grenzen sich – wie bei HIAT – nicht unbedingt auf die Grenzen der phonologisch motivierten Segmente beziehen lassen.

So werden sich Transkriptionen nach GAT und Transkriptionen nach HIAT in den meisten Fällen in ihrer Segmentierung unterscheiden: Segmente in GAT-Transkriptionen werden sich stärker an prosodischen Merkmalen orientieren, Segmente in HIAT-Transkriptionen kommen denen der Schriftsprache deutlich näher.<sup>42</sup>

---

42 Eine weitergehende Betrachtung der Eigenschaften der HIAT und GAT-Transkriptionssysteme sowie der Transkriptionssysteme DIDA und Verbmobil findet sich in Schmidt (2004: 88ff).

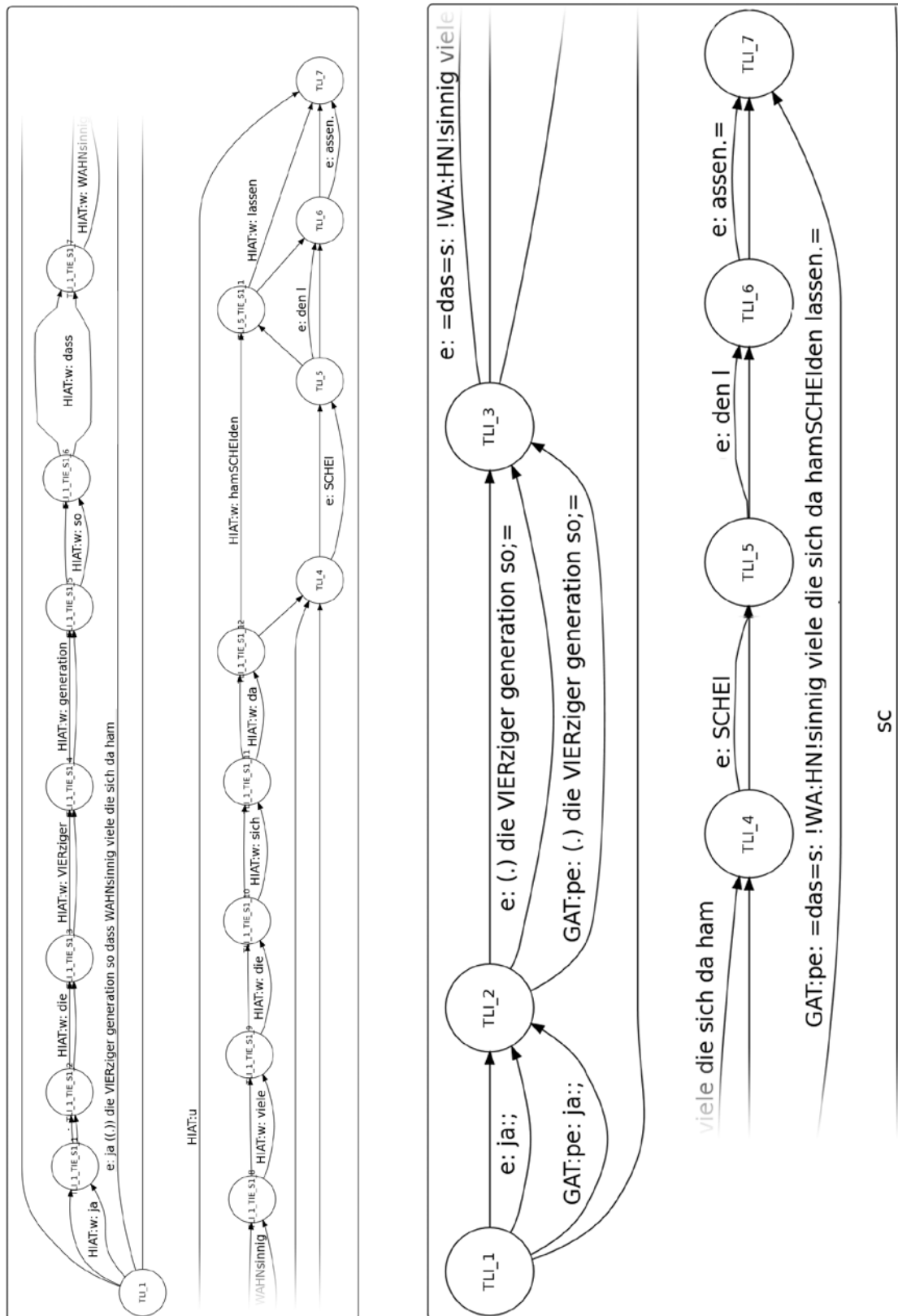


Abbildung 14: Zweimal derselbe Transkriptionsabschnitt, einmal nach HIAT- (links) und einmal nach GAT-Konventionen (rechts) segmentiert.

### 2.3.6. Anforderungen an das Datenmodell

Aus den beschriebenen Problemfällen für die Modellierung ergeben sich Anforderungen, die ein Datenmodell für Transkriptionen gesprochener Sprache erfüllen muss, um die Voraussetzungen für eine effektive computergestützte Verarbeitung bereitzustellen:

- **Zeitachse:** Für die Untersuchung gesprochener Sprache sind Informationen über die zeitliche Einordnung von Sprecherbeiträgen und anderer für die Untersuchung relevanter Ereignisse wichtig. Ein Transkriptionsmodell muss diese Informationen abbilden können.
- **Überlappung:** In gesprochener Sprache kommt es zu Überlappungen von Sprecherbeiträgen. Dadurch können mehrere, parallel verlaufende Textströme entstehen, gleichzeitig können auch noch nichtverbale Ereignisse stattfinden, die ebenfalls in der Transkription berücksichtigt werden müssen. Ein Transkriptionsmodell muss solche überlappenden, parallelen Datenströme abbilden können.
- **Segmentierung:** Ein wesentlicher Bestandteil existierender Transkriptionssysteme ist die Systematik, transkribierten Text in bestimmte Einheiten oder Segmente aufzuteilen. Ein Transkriptionsmodell muss in der Lage sein, diese Einheiten abzubilden.
- **Schriftsysteme:** Transkriptionen gesprochener Sprache müssen in vielen Fällen phonologische Besonderheiten des Gesprochenen berücksichtigen. Ein Transkriptionsmodell muss für seinen Einsatzzweck eine Möglichkeit bereitstellen, diese Besonderheiten abzubilden. Dies kann durch die Unterstützung spezieller Schriftsysteme oder die Codierung in nicht-textuellen Datensätzen ermöglicht werden.

Im nächsten Kapitel sollen zwei unterschiedliche Ansätze und einige konkrete Modelle zur Modellierung von Sprachdaten vorgestellt und daraufhin untersucht und bewertet werden, wie gut sie sich unter Berücksichtigung der bis hierher erarbeiteten Anforderungen an ein Modell für die Transkription gesprochener Sprache eignen.

## 3. Existierende Datenmodelle

Bei Datenmodellen, die zur Codierung von Sprachdaten für die linguistische Forschung verwendet werden, verläuft eine sichtbare Trennlinie zwischen Modellen, die dem OHCO-Formalismus zuzurechnen sind und solchen, die auf nicht baumförmigen Graphenstrukturen basieren.

Vereinfachend ausgedrückt werden Sprachdaten in einem OHCO-Modell als streng geordnete Hierarchien, in Graphenmodellen als Graphen mit Bezug zu einer kontinuierlichen Zeitachse repräsentiert. Beide Modelle haben ihre spezifischen Vor- und Nachteile für die Verwendung mit Daten gesprochener Sprache, auf die im Folgenden näher eingegangen werden soll.

### 3.1. Ordered Hierarchy of Content Objects (OHCO) und hierarchische Baumstrukturen

#### 3.1.1. OHCO-Modelle

Die OHCO-These besagt, dass Texte immer eine *Ordered Hierarchy of Content Objects* sind, also immer eine geordnete, hierarchische Struktur haben, in der Elternelemente ihre Kindelemente einschließen und es somit keine überlappenden Hierarchien geben kann. Diese These wurde von DeRose et al. (1990: 20f) mit einiger Vehemenz aufgestellt:

*„WE THINK that our point can scarcely be overstated. Text is an Ordered Hierarchy of Content Objects; any software application, or any set of computing practices that is based on some other model of text is inadequate for our intellectual and scholarly purposes.“*

Diese Sicht auf Texte lässt sich nur aufrechterhalten, wenn man Texte überwiegend als (gedruckte) Dokumente wahrnimmt. Eine derartig starke These aufzustellen hatte vor allem praktische Vorteile:

*„Texts modeled as OHCOs are easier to create, modify, analyze, etc. The comparative efficiency and functionality of treating texts „as if“ they were OHCOs is best explained, according to this argument, by the hypothesis that texts ‚are‘*

*OHCOs.*“ (Renear 1997)

Insbesondere die leichte und effiziente Verarbeitbarkeit von Texten, die in elektronischer Form tatsächlich nach dem OHCO-Paradigma modelliert sind, hat dabei zu dessen großer Verbreitung und Akzeptanz geführt. Die Einschränkungen, die sich dadurch ergeben, finden sich so auch in den TEI-Guidelines und allgemeiner in XML<sup>43</sup> wieder: Wohlgeformte XML-Dokumente dürfen keine überlappenden Elemente und Hierarchien enthalten.<sup>44</sup>

In der linguistischen Realität (und nicht nur dort) lässt sich die OHCO-These aber nicht aufrechterhalten. Das war (oder wurde) auch den Verfassern klar, die sich in der Diskussion um die These an deren Falsifikation rege beteiligten und im Verlauf zu dem Schluss kommen:

*„And although we have retreated from the simple OHCO thesis, we note that the spirit of the OHCO hypotheses is borne out to the extent that texts qua intellectual objects still seem to be composed of structures of meaning-related features and that, moreover, these structures are often hierarchical.“* (Renear, Mylonas und Durand 1996)

Den Umstand, dass Textstrukturen „oft hierarchisch“ sind, hatte aber auch von vorneherein niemand ernsthaft bestritten – genauso wenig wie die Tatsache, dass sich mit dem OHCO-Modell viele Bedürfnisse bei der digitalen Repräsentation von Texten befriedigen lassen.

Dass eine einfache XML-Struktur im Grundsatz zunächst natürlich auch eine *Ordered Hierarchy of Content Objects* ist, erleichtert dabei die Verarbeitung tatsächlicher OHCO-Ressourcen. Die relativ offene Struktur von XML und insbesondere die offene Semantik von Zeigerstrukturen ermöglicht aber auch die Modellierung von Ressourcen, die völlig anderen Strukturen unterliegen, was allerdings häufig zu Lasten der Verarbeitungsgeschwindigkeit geht. Auf diese Problematik wird später – insbesondere bei der Vorstellung der Graphenmodelle – noch näher eingegangen.

---

43 Vgl. Lobin (2001).

44 Vgl. Witt (2002: 41).



### 3.1.2. Die TEI-Guidelines

Die OHCO-Grundlage von Text als einer hierarchischen Struktur findet ihre bekannteste Realisierung in den Guidelines der Text Encoding Initiative (TEI, (Text Encoding Initiative Consortium 2001)), dort wird allerdings die Ausrichtung der Guidelines mit folgenden Worten beschrieben:

*„They [the guidelines, KW] provide means of representing those features of a text which need to be identified explicitly in order to facilitate processing of the text by computer programs. In particular, they specify a set of markers (or tags) which may be inserted in the electronic representation of the text, in order to mark the text structure and other textual features of interest.“*

Die Autoren grenzen das Anwendungsgebiet der Guidelines explizit nicht ein:

*„The Guidelines apply to texts in any natural language, of any date, in any literary genre or text type, without restriction on form or content.“ (Text Encoding Initiative Consortium 2001: 1)*

In der Praxis haben sich die TEI-Guidelines, insbesondere in der Version P4, vor allem in den Bereichen literaturwissenschaftlicher Philologien, im Bibliothekswesen und bei der linguistisch oder anderweitig motivierten Beschreibung schriftlicher Ressourcen durchgesetzt; für die Repräsentation gesprochener Sprache spielen sie hingegen bislang keine große Rolle.

#### **TEI und gesprochene Sprache**

Der Hauptgrund für die mangelnde Akzeptanz der TEI-Guidelines für die Erforschung gesprochener Sprache liegt darin, dass die grundlegende Struktur auch einer TEI-Transkription immer eine geordnete Hierarchie ist. Transkriptionen sind im Rahmen der TEI-Guidelines einfach eine andere Form von Text: Der dokumentenzentrierte Ansatz der TEI wird nicht aufgegeben.

Einige der Probleme, die für Daten gesprochener Sprache spezifisch sind, werden in den Guidelines (u.a. explizit im Kapitel *Transcriptions of Speech*<sup>45</sup>) angesprochen und Empfehlungen zu ihrer Behandlung gegeben:

---

45 Vgl. auch Witt (1998).

- Für Sprecherzuordnungen gibt es zusätzlich zu den Einheiten, die in allen TEI-Dokumenten vorgesehen sind, in TEI-Transkriptionen ein Element des Typs `<u>`, das für eine *utterance* steht. Dieser Begriff steht in der TEI-Definition für einen Gesprächsbeitrag eines Sprechers (der mit einem Attribut genannt wird), der durch Pausen oder Sprecherwechsel begrenzt wird und mit `<seg>`-Elementen weiter unterteilt werden kann (Johansson 1995: 87).

Dass ein derartiges Element in anderen Transkriptionssystemen häufiger als *Turn* beschrieben wird, verdeutlicht ein Problem der TEI-Guidelines nicht nur im Bezug auf Transkriptionen gesprochener Sprache: Selbst wenn sich die Guidelines als generisches System für die Transkription von Sprachdaten gebrauchen lassen, so sind Definitionen wie die für das Element `<u>` doch klar aus einer bestimmten linguistischen Perspektive heraus motiviert und stehen einer breiteren Akzeptanz damit im Wege.

- Für nonverbale Ereignisse ist ein Element `<kinesic>` vorgesehen, das mit einem `desc`-Attribut näher beschrieben werden kann. Auch für Ereignisse, die keinem Sprecher zugeordnet sind, gibt es ein Element: `<event>`.
- Für verbalisierte Ereignisse, die nicht lexikalisiert sind (wie Lachen oder Husten), ist das Element `<vocal>` vorgesehen; auch dieses Element lässt sich mittels eines `desc`-Attributes weiter spezifizieren. Sprecherpausen werden mithilfe eines dezidierten `<pause>`-Elementes transkribiert, das Längenangaben als Attribut bekommen kann.

Um bestimmte Annotationen (im Sinne der Definition dieser Arbeit) zu Merkmalen der Sprechweise einzelner Sprecher zu machen, sehen die Guidelines das Element `<shift>` vor. Dieses Element stellt einen Sonderfall dar: Es handelt sich um ein leeres Element, das nur den Beginn eines Abschnittes markiert, an dem sich eine Eigenschaft der Sprechweise eines Sprechers ändert, also einen klassischen Milestone. Die Guidelines benennen bestimmte Eigenschaften, die sich mit diesem Konstrukt annotieren lassen sollen, so z.B. Sprechgeschwindigkeit, Tonhöhe, Lautstärke und ähnliche weitere lassen sich über modifizierte DTDs ergänzen.

```

<u who="lb"><shift feature="loud" new="f"/>Elizabeth</u>
<u who="eb">Yes</u>
<u who="lb"><shift feature="loud" new="normal"/>Come and try this <pause/>
<shift feature="loud" new="ff"/>come on</u>

```

Abbildung 15: Die Veränderung der Sprechintensität des Sprechers „lb“ wird hier mit Hilfe von shift-Milestones markiert. (Beispiel aus den TEI-P5 Guidelines: <http://www.tei-c.org/release/doc/tei-p5-doc/html/ref-shift.html>)

Da die <shift>-Elemente als Milestones kein schließendes Tag besitzen, muss das Ende eines annotierten Abschnittes wieder mit einem Milestone markiert werden. Die TEI-Guidelines sehen hierfür allerdings keine Verknüpfung über eine eindeutige Id vor: ein neuer, annotierter Abschnitt zum gleichen feature beendet in diesem Fall den vorigen. Wenn der folgende Abschnitt in Bezug auf die annotierte Eigenschaft keine Besonderheit aufweist, so wird der Wert des new-Attributes leer gelassen oder auf den Wert normal gesetzt.

## Überlappende Hierarchien

Milestones finden sich auch an anderen Stellen der TEI-Guidelines, beispielsweise, um physische (Seitenzahlen, Spalten, Bände) und sprachliche Struktur (Absätze, Sätze, Wörter) eines Dokumentes, die sich auch überlappen können, darzustellen.

Die Milestones überwinden eine schwerwiegende Einschränkung des OHCO-Modells, allerdings um den Preis, das Modell gleich mit zu überwinden: Wenn ein Text eine *Ordered Hierarchy of Content Objects* ist, dann fallen Milestones als Elemente, die keinen Content haben, aus dem Modell heraus.

*„Milestones do not align with the truism that elements contain a range of text.“*  
(Witt 2004: 2).

Auf weitere Strategien, mit Überlappungen umzugehen, gehen die P4-Guidelines in einem separaten Kapitel ein (Text Encoding Initiative Consortium 2001: Abschnitt 31). Dort wird gleich zu Beginn festgestellt:

*„Non-nesting information poses fundamental problems for any XML-based encoding scheme, and it must be stated at the outset that no current solution combines all the desirable attributes of formal simplicity, capacity to represent*

*all occurring or imaginable kinds of structures, suitability for formal or mechanical validation. The representation of non-hierarchical information is thus necessarily a matter of trade offs among various sets of advantages and disadvantages“.*

Neben den bereits erwähnten Milestones werden als Möglichkeit zur Modellierung überlappender Hierarchien noch genannt:

- *Concurrent Markup of Multiple Hierarchies*: CONCUR ist ein SGML-Feature, um konkurrierende Hierarchien abbilden zu können. Da es bereits in SGML-Umgebungen – wenn überhaupt – nur unzureichend implementiert und schwierig handzuhaben war, wurde es in XML (das ja vor allem die Komplexität von SGML reduzieren sollte) nicht übernommen. Ein (Nicht-XML-) Format, das die Funktionalität von CONCUR abbildet und Brücken zur XML-Welt baut, ist in Hilbert, Schonefeld und Witt (2005) und Schonefeld und Witt (2006) als XCONCUR beschrieben. Es wird in dieser Arbeit ab Seite 67 näher beschrieben.
- *Fragmentation of Elements*: Die Unterteilung von Segmenten in kleinere Segmente, um Überlappungen zu markieren. Der offensichtliche Nachteil dieser Methode ist, dass sich die Segmente, in denen sich die neuen Untersegmente finden, durch die Fragmentierung nicht mehr eindeutig zueinander in Bezug setzen lassen und auf diese Weise Information verloren geht.
- *Reconstitution of Virtual Elements*: Diese Methode versucht den Informationsverlust, der bei der Fragmentierung entstanden ist, wieder auszugleichen. Mit einem separaten JOIN-Element, das hinter die Fragmentierung gesetzt wird, werden die zueinandergehörenden Segmente wieder miteinander in Beziehung gesetzt, die Hilfskonstruktion also durch eine weitere Hilfskonstruktion gestützt, ohne dass dabei die Nachteile einer mangelhaften Verarbeitbarkeit aufgehoben würden.
- *Multiple Encodings of the Same Information*: Für diese Methode werden die einzelnen Hierarchieebenen im Text und in den Annotationen, die sich überlappen, jeweils in separaten Dateien annotiert und das Primärdatum (der Text) dabei jedes Mal dupliziert. Diese Dateien sind immer TEI-konform, passen in

das OHCO-Modell und lassen sich effektiv verarbeiten. Auf der anderen Seite müssen diese Dateien allerdings zusammengeführt werden, um einen Blick auf die Gesamtheit der Hierarchien und Annotationen bekommen zu können, und Änderungen am Primärdatum müssen immer – mit großer Sorgfalt – in allen Dateien vorgenommen werden.

Ein Format, das mit primärdatenidentischen Annotationen arbeitet, ist *GENAU* (Witt et al. 2007 und Witt et al. 2009).

### **TEI P5 und überlappende Hierarchien**

Nach einem lange währenden Diskussionsprozess ist die nächste Version der TEI-Guidelines (P5) im Jahr 2007 veröffentlicht worden (Burnard und Bauman 2007). Da die Diskussion der Modellierung überlappender Hierarchien für die TEI von großem Interesse sein muss, beschäftigten sich auch die überarbeiteten Guidelines ausführlicher als zuvor mit diesem Thema.

In den P5-Guidelines werden die gleichen Verfahren für die Modellierung überlappender Hierarchien empfohlen wie in P4. Abweichend werden Fragmentation und Reconstruction (sinnvollerweise) zu einem Punkt zusammengefasst, in der (nicht unbedingt relevanten) Reihenfolge der Konzepte ist die Mehrfachannotation an die erste Stelle gerückt.

Ein Verfahren, das in den P4-Guidelines keine Erwähnung fand, aber bereits innerhalb der TEI diskutiert wurde, kommt in P5 hinzu: Die *Standoff-Annotation*, „*the annotation of information by pointing at it, rather than by placing XML tags within it.*“ (Burnard und Bauman 2007)

### **Standoff-Annotation**

Standoff-Annotation basiert prinzipiell darauf, Annotationen nicht in Form von *Elementen im Text*, sondern von *Verweisen auf den Text* vorzunehmen.

**(annotiertertext.xml)**

```
...
<s id="s4">
<w id="w45" lemma="immer">immer</w>
<w id="w46" lemma="unterbrechen">unterbrichst</w>
<w id="w47" lemma="du">du</w>
<w id="w48" lemma="ich">mich</w>
</s>
...
```

Abbildung 16: Ein TEI-XML-Fragment mit Inline-Annotation: Die Annotation zu Lemmata wird in den annotierten Elementen gespeichert.

**(text.xml)**

```
...
<s id="s4">
<w id="w45">immer</w>
<w id="w46">unterbrichst</w>
<w id="w47">du</w>
<w id="w48">mich</w>
</s>
...
```

**(Lemma.xml)**

```
...
<annotation xlink:href="#w45" type="lemma">immer</annotation>
<annotation xlink:href="#w46" type="lemma">unterbrechen</annotation>
<annotation xlink:href="#w47" type="lemma">du</annotation>
<annotation xlink:href="#w48" type="lemma">ich</annotation>
...
```

Abbildung 17: Das gleiche Fragment mit der gleichen Annotation in einem Standoff-Format: Die Annotation ist über Zeiger mit dem nicht-annotierten Text verknüpft.

Für diese Verweise gibt es mehrere Möglichkeiten: Annotationen können sich direkt auf Elemente des Ausgangstextes beziehen, die dazu mit eindeutigen IDs versehen sein müssen. Sie können sich aber auch auf unmarkierte Textabschnitte, die mittels Character-Offsets verlinkt werden, beziehen. Die jeweiligen Annotationsebenen können dabei im selben Dokument gehalten werden oder in einem (oder mehreren) separaten Dokumenten. Bei TEI erfolgt die eigentliche Verlinkung mittels XPointer-Verweisen (Grosso et al. 2003).

Zu den Vorteilen der Standoff-Annotation zählen insbesondere die Möglichkeiten, die sich mit einer tatsächlichen physischen Trennung von Annotiertem und Annotation ergeben: So können beispielsweise beliebig viele Annotationen zum selben Text angefertigt werden, unterschiedliche Personen können die gleichen Merkmale unterschiedlich annotieren und Annotationen können – sofern dies

sinnvoll ist – auch ohne den Ursprungstext weitergegeben werden. Andere Vorteile ergeben sich aus der Zeigerstruktur: So lassen sich beispielsweise neben überlappenden Hierarchien auch diskontinuierliche Einheiten mit Hilfe von Zeigern gut annotieren. Die grundsätzliche Trennung von Annotiertem und Annotation ermöglicht darüber hinaus, Annotation auf andere Annotationen vorzunehmen, also beispielsweise eine syntaktische auf eine Wortartenannotation.

Als Nachteil von Standoff-Annotation lässt sich insbesondere anführen, dass die verschiedenen Ebenen voneinander abhängen. Obwohl es nicht explizit erwähnt wird, treten auch hier Probleme auf, wenn sich der Ausgangstext verändert. Diese Probleme verstärken sich, wenn sich Annotationsebenen auf andere Annotationsebenen beziehen – was für Standoff-Annotation durchaus möglich und praktikabel ist – und sie verstärken sich noch weiter, wenn tatsächlich nicht nur auf Elemente verwiesen wird, sondern Character-Offsets zum Einsatz kommen.

*(text.xml)*

```
...
<s id="s4">
  <w id="w45">immer</w>
  <w id="w46">unterbrichst</w>
  <w id="w47">du</w>
  <w id="w48">blödmann</w>
  <w id="w49">mich</w>
</s>
...
```

*(Lemma.xml)*

```
...
<annotation xlink:href="#w45" type="lemma">immer</annotation>
<annotation xlink:href="#w46" type="lemma">unterbrechen</annotation>
<del><annotation xlink:href="#w47" type="lemma">du</annotation></del>
<del><annotation xlink:href="#w48" type="lemma">ich</annotation></del>
...
```

Abbildung 18: Wird in die annotierte Textdatei ein Wort eingefügt („blödmann“ mit der ID „w48“), so stimmen die Zeiger in der Annotationsdatei nicht mehr. Eine Arbeitsumgebung, die auf Standoff-Annotation baut, muss für derartige Fälle vorsorgen.

Obwohl einige verfügbare XML-Werkzeuge Standoff-Annotation bereits berücksichtigen, ist auch bei dieser Art, mit Überlappung umzugehen, die Informationsgewinnung und Verarbeitung keineswegs trivial und daher rechenintensiv: durch den Umstand, dass die Annotationen und das Annotierte permanent aus

verschiedenen Dateien gegeneinander aufgelöst werden müssen, vervielfacht sich die Verarbeitungszeit im Vergleich zu Inline-Annotationen.

### Überlappung von Sprecherbeiträgen

Auch die Überlappung von Sprecherbeiträgen wird in den P5-Guidelines explizit im Kapitel 16, „*Linking, Segmentation, and Alignment*“ behandelt.<sup>46</sup> Es werden drei Verfahren empfohlen, mit diesen Überlappungen umzugehen:

- Das Setzen von eindeutigen anchor-Milestones, die die jeweiligen Stellen, an denen die Überlappung stattfindet, markieren und einen direkten Bezug im Text zu diesen Milestones mittels eines `synch`-Attributes herstellen
- Das Setzen von Milestones an den Schnittpunkten und das anschließende Zusammenführen dieser Milestones mit Hilfe von `link`-Elementen in einer für diesen Zweck geschaffenen `linkGrp`
- Das Verweisen mittels `synch` unter bestehenden Elementen, die sich vollständig überlappen (weil dort Milestones eigentlich überflüssig sind) oder unter dafür vorgesehenen `seg`-Elementen (was allerdings nur funktioniert, wenn dadurch keine andere Überlappung entsteht)

Diese Verfahren, die im Grunde bereits relative Zeitpunkte festlegen, können noch um eine explizite Timeline erweitert werden: Die Guidelines sehen dafür ein `timeline`-Element vor, das `when`-Elemente als Zeitpunkte etabliert, die entweder ein Intervall (von einem vorhergehenden Zeitpunkt), einen absoluten Zeitpunkt oder einen un spezifizierten Zeitpunkt markieren. Diese Zeitpunkte in der Timeline können ebenfalls mit `link`-Elementen mit den oben erwähnten anchor-Zeitpunkten verknüpft werden, um eine direkte zeitliche Anordnung der Elemente ablesen zu können.

### 3.1.3. Bewertung

Wären die TEI-Guidelines als solche in der Domäne schriftlicher Texte eine bemerkenswerte Verbreitung gefunden haben, konnten sie sich für Korpora gesprochener Sprache nie richtig durchsetzen. Die Probleme, die bei der Modellierung

---

46 Zu einer ausführlicheren Diskussion von TEI-Transkriptionen vgl. Schmidt (2004: 179ff).



von Daten gesprochener Sprache entstehen können, standen bei der Ausarbeitung der ersten Versionen der Guidelines auch mit Sicherheit nicht im Fokus, vielmehr ging es darum, ein einheitliches Datenformat für hierarchisch strukturierte (und eben vor allem schriftliche) Texte zu erstellen, das den grundsätzlichen Austausch und die Archivierung solcher Daten in digitaler Form ermöglicht.

*„While famous TEI corpora such as the BNC (...) exist, TEI is much more popular and a widely adopted standard in other fields of research, e. g. in literature, in the humanities or in text archives. The TEI website lists over a hundred projects which use the TEI guidelines, and this list is far from complete. The TEI guidelines’ compliance to established indexing mechanisms used in libraries also contributes to this success.“ (Lehmborg und Wörner 2008)*

Die Verfahren, die von den P4-Guidelines zur Behandlung von typischen Problemfällen mit Daten gesprochener Sprache auftreten, wirken eher wie „Workarounds“. Insbesondere die Tatsache, dass viele der vorgeschlagenen Verfahren – zum Beispiel der Behandlung von Überlappung von Sprecherbeiträgen – zwar eingeführt werden, für deren Auswahl aber keine eindeutige Empfehlung gegeben wird<sup>47</sup>, führte sicherlich dazu, dass sich die TEI-Guidelines nicht auch als Standard für derartige Daten durchsetzen konnten.

Die P5-Guidelines sehen zusätzliche Verfahren vor, um mit den genannten Phänomenen umzugehen und bauen damit eine Brücke zu anderen Modellen, die sich in der Zeit seit Verabschiedung der P4-Guidelines entwickelt haben. Eine Festlegung, mit welchen Mitteln man *konkret* den Besonderheiten gesprochener Sprache zu Leibe rücken soll, werden aber auch hier nicht gemacht.

Es ist allerdings inzwischen durchaus ein etabliertes Verfahren geworden, basierend auf den TEI-Guidelines eigene DTDs mit eigenen Tagsets zu definieren, so dass sich TEI für konkrete Anwendungsfälle durchaus als Format für Transkriptionen gesprochener Sprache eignen könnte.

Das von der TEI zur Verfügung gestellte Werkzeug für das assistierte Erstellen

---

47 Das Vorsehen mehrerer Möglichkeiten der Codierung gleichartiger Daten und Annotationen findet sich in den TEI-Guidelines auch an anderen Stellen und hat die allgemeine Akzeptanz als Standard auch in anderen Bereichen behindert. Vgl. Lehmborg und Wörner (2008) .

von TEI-DTDs bietet zum Beispiel ein fertiges Template für die Transkription gesprochener Sprache an.<sup>48</sup> Das Problem, dass überlappende Hierarchien zu einer deutlich erschwerten Verarbeitbarkeit durch den Computer führen, stellt sich aber natürlich auch für die TEI-Guidelines.

Grundsätzlich ermöglichen die TEI-Guidelines auch die Annotation solcher Transkriptionen; alle Mittel, die in diesem Kapitel für die Behandlung von überlappenden Sprecherbeiträgen angeführt wurden, eignen sich in gleichem Maße für überlappende Annotationen – selbstverständlich würden überlappende Annotationsebenen auf überlappenden Sprecherbeiträgen die Komplexität weiter erhöhen und damit die Lesbarkeit und die Performance computergestützter Auswertung weiter verringern – eine Tatsache, die allerdings nicht den TEI-Guidelines anzulasten ist.

Aus Anwendersicht bietet die Arbeit mit den TEI-Guidelines den Vorteil, dass es zwar sehr wenige wirkliche Werkzeuge für die explizite Arbeit mit TEI-Dokumenten gibt, auf der anderen Seite jedoch eine ganze Reihe von XML-Werkzeugen auf die Arbeit mit TEI-Dokumenten vorbereitet ist.<sup>49</sup>

Diese Werkzeuge sind allerdings nicht auf die Art von TEI-Dokumenten vorbereitet, um die es in dieser Arbeit geht. Die wenigsten dieser Werkzeuge können z.B. tatsächlich mit Merkmalen wie Standoff-Annotation arbeiten oder TEI-timelines und ihre when-Points in einer Art verarbeiten, dass der Benutzer dadurch eine wirkliche Arbeitserleichterung hätte. Diese ließe sich nur schaffen, wenn eine Software für den konkreten Zweck der Transkription und Annotation von gesprochener Sprache entwickelt würde, die TEI als Datenformat zugrunde legt.

Bei Verfahren und Werkzeugen zur Visualisierung von TEI-Daten verhält es sich analog: Zwar findet man im Internet unzählige XSL-Stylesheets, die verschiedene

---

48 Das Werkzeug „Roma“ für TEI P5 (<http://www.tei-c.org/wiki/index.php/Roma>) ist Nachfolger des „Pizza Chef“ für TEI P4 (<http://www.tei-c.org.uk/pizza.html>). Beide stellen ein Template für *Speech* als Ausgangsbasis für eigene DTDs zur Verfügung.

49 Das TEI-Wiki bietet eine Übersicht über Werkzeuge, mit denen sich TEI-Dokumente be- und verarbeiten lassen (<http://www.tei-c.org/wiki/index.php/Category:Tools>).

Visualisierungen von TEI-Dokumenten liefern können. Auch diese sind aber fast immer für geschriebene Texte und nicht für Transkriptionen gesprochener Sprache vorgesehen. Eine Visualisierung einer komplexen TEI-Transkription und ihrer Annotation zu erstellen, wäre keineswegs trivial, und die bereits erwähnten Performance-Probleme kämen voll zum Tragen.

Für Abfragewerkzeuge auf TEI-Transkriptionen gilt verständlicherweise dasselbe: Es existieren keine spezialisierten Abfragewerkzeuge für Transkriptionen, die nach TEI erstellt wurden. Will man entsprechende Transkriptionen durchsuchen, so ist man auf Standardwerkzeuge wie XML-Editoren, XPATH und XQUERY-Ausdrücke und XSL-Stylesheets angewiesen – diese Mittel stehen aber grundsätzlich für alle Modelle zur Verfügung, die XML als Datenformat haben. Auch hier müsste also ein speziell auf TEI abgestimmtes Such- und Abfragewerkzeug erstellt werden, um eine echte Arbeitserleichterung zu schaffen.

Einige Transkriptionswerkzeuge (so auch der Partitur-Editor des für diese Arbeit relevanten EXMARaLDA-Systems) bieten aber beispielsweise einen Export ihrer Transkriptionsdaten in ein TEI-XML-Format an, so dass TEI im Kontext der Transkription gesprochener Sprache wegen seiner ausführlichen Dokumentation und seiner langen Geschichte durchaus als ein Format zum Austausch von Daten zwischen verschiedenen Formaten und Werkzeugen, vor allem aber auch zur Langzeitarchivierung verwendet werden kann.

### **3.1.4. Exkurs: XCONCUR, XCONCUR-CL**

Das im vorigen Kapitel bereits erwähnte XCONCUR (Schonefeld et al. 2006) nimmt ein Feature wieder auf, das bereits in SGML existierte, um überlappende Hierarchien innerhalb einer Dokumentinstanz modellieren zu können. In SGML war es vorgesehen, durch das Einschalten des CONCUR-Features (sonst verbotene) Überlappungen zu ermöglichen, indem bestimmte Ebenen des Textes identifiziert werden, die einander dann nicht mehr überlappen dürfen (siehe Sperberg-McQueen und Huitfeldt (1999)).

Das Feature war in SGML zwar vorgesehen, wurde aber in keiner SGML-Umgebung implementiert. Bei der Einführung von XML wurde das Feature fallen

gelassen, weil es dem vorrangigen Ziel, SGML leichter versteh- und handhabbar zu machen, direkt entgegenstand.

XCONCUR versucht, dieses SGML-Feature in die XML-Welt zu übertragen (Schonefeld 2005).<sup>50</sup>

Für das folgende Transkriptionsbeispiel<sup>51</sup>:

	0	1	2	3
SPK0 [v]	Hey Paul!	Would you give me		
SPK1 [v]			the hammer?	

Abbildung 19: Der Satz von Sprecher 0 wird von Sprecher 1 zu Ende geführt.

würde eine XCONCUR-Datei (auch „XMC-Datei“) so aussehen:

```
<?xconcur version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE (l1)div SYSTEM "tei/dtd/teispok2.dtd">
<!DOCTYPE (l2)text SYSTEM "tei/dtd/teiana2.dtd">
<(l1)div type="dialog" org="uniform">
  <(l2)text>
    <(l1)u who="Peter">
      <(l2)s>Hey Paul!</(l2)s>
      <(l2)s>Would you give me
    </(l1)u>
    <(l1)u who="Paul">
      the hammer?</(l2)s>
    </(l1)u>
  </(l2)text>
</(l1)div>
```

Abbildung 20: XCONCUR-Markup.

Die überlappenden Strukturen werden ermöglicht, indem in allen Tags über ein Annotationsebenen-Präfix eine eindeutige Zuordnung zu einer bestimmten Annotationsebene hergestellt wird. Für das obige Beispiel werden mit (l1) die Äußerungen der Sprecher und mit (l2) die linguistischen Segmente (hier: Text und Satz) markiert. Da der Satz „Would you give me the hammer“ von Sprecher 0 begonnen und von Sprecher 1 zu Ende gesprochen wird, überschneiden sich diese Ebenen, es entstehen zwei überlappende Baumstrukturen.

50 Für weitere Informationen vgl. <http://www.xconcur.org/>.

51 Beispiele aus Schonefeld (2007).

In XML würde eine Struktur wie `<u><s/><s></u><u></s></u>` den Wohlgeformtheits-Grundsatz verletzen. In XCONCUR müssen alle Elemente, die zu einer Annotationsebene gehören, diesem Grundsatz genügen – für die einzelnen Annotationsebenen lassen sich auch Dokumentgrammatiken in den gängigen XML-Schemasprachen erstellen und zuordnen.

Im obigen Beispiel liegt noch keine zeitliche Überlappung vor; Überlappungen dieser Art müssen, genau, wie es in XML der Fall ist, mit direkten Verweisen auf die Zeitachse realisiert werden:

```
<(11)orth id="TIE1.a1" s="TLI2" e="TLI3" value="Mira.">
```

Abbildung 21: XCONCUR-Element mit Zeitachsen-Verweisen.

Das Merkmal, das XCONCUR als Format für überlappende Hierarchien besonders interessant macht, ist die Möglichkeit, die Beziehungen von XCONCUR-Elementen aus verschiedenen Annotationsebenen mittels der Constraints-basierten Schema-Sprache XCONCUR-CL (XCONCUR Constraints Language, Schonefeld 2007) festzulegen und zu verifizieren. So lässt sich beispielsweise festlegen, dass Elemente der Annotationsebene, die Wörter aufnimmt, immer ein oder mehrere Elemente der Annotationsebene, die Silben aufnimmt, enthalten müssen.

Sind die Regeln für ein solches XCONCUR-Annotationsschema einmal festgelegt, so lässt sich nicht nur die Wohlgeformtheit der einzelnen Bäume, sondern auch die Korrektheit der Beziehungen zwischen den unterschiedlichen Annotationsebenen automatisch validieren.

Der größte Nachteil von XCONCUR und XCONCUR-CL wiegt allerdings schwer: XCONCUR ist nicht XML-kompatibel. Zwar orientiert sich das Modell stark an den Vorgaben von XML, setzt bei der Validierung von einzelnen Annotationsebenen auf XML-Dokumentgrammatiken und ist für einen Anwender, der mit XML-Technologie vertraut ist, grundsätzlich nachvollziehbar, teilt aber das Schicksal aller Modelle, die sich zwar an XML orientieren, aber eben kein XML sind: Das inzwischen stattliche Repertoire an Werkzeugen, APIs und Serverum-

gebungen lässt sich nicht unmodifiziert anwenden. Da in der Struktur von XCONCUR-Dokumenten mit der direkten Überlappung von Elementen gegen eine der zentralen Regeln von XML verstoßen werden kann, lässt sich dieses Problem auch nicht mit einfachen Mitteln umgehen.

Für die computergestützte Verarbeitung von XCONCUR-Markup existiert inzwischen eine Referenzimplementierung und eine Programmierschnittstelle. Diese XCONCUR SAX-API (Schonefeld 2008) ermöglicht – analog zum SAX-Parser für XML-Dokumente<sup>52</sup> – XCONCUR-Dokumente zu lesen und mittels Rückruffunktionen weiterzuverarbeiten. Im Unterschied zu DOM-Parsern produziert die API aber kein Abbild des Dokuments im Speicher – diese Aufgabe müsste weitergehende Software übernehmen, die auf der API aufsetzt. Die Referenzimplementierung (Schonefeld 2005) erfüllt diese Aufgabe, bietet allerdings als „proof of concept“ keine Schnittstellen zu anderen Annotationsumgebungen.

Das XCONCUR zugrunde liegende Konzept bietet unmittelbar einsichtige Vorteile, insbesondere die Möglichkeit, Mehrebenenannotation mit den zugrunde liegenden Texten über Dokumentgrammatiken zu validieren und die Annotationsmöglichkeiten mit deren Hilfe klar zu definieren.

Auf der praktischen Seite ist der Ansatz kritischer zu betrachten: Die Tatsache, dass XCONCUR den Rahmen des XML-Standards verlässt, stellt den praktischen Nutzen des Systems stark in Frage. Anwender, die Daten im XCONCUR-Format verwenden wollten, wären immer darauf angewiesen, dass speziell auf das Modell zugeschnittene Software existiert. In den Anforderungen an eine Software zur Annotation auf Basis von XCONCUR erwähnt der Autor unter anderem:

*„Weiterhin muss der Editor in der Lage sein, verschiedene Formate zu importieren und exportieren. Besonders wichtig ist das Einlesen von n primärdatenidentischen XML-Dokumenten. Beim Laden muss eine Whitespace-Normalisierung durchgeführt werden und die XML-Dateien können dann zu einem XMC-Dokument zusammengeführt werden. (...) Natürlich sollte ein XMC-Dokument auch wieder in n XML-Dokumente gespeichert werden können. So kann der Be-*

---

52 Vgl. Means und Bodie (2002).

*nutzer das Dokument bequem annotieren und nach dem Exportieren mit anderer Software weiterverarbeiten.“ (Schonefeld 2005: 54)*

In einem Workflow, der primärdatenidentische XML-Dokumente in einem Annotationswerkzeug zu einem XCONCUR-Dokument zusammenstellt und nach erfolgter Annotation zur Weiterverarbeitung wieder in XML-Dokumente zurückschreibt, bleibt der Nutzen des eigentlichen Modells und Datenformates etwas im Unklaren – die Annotationsregeln ließen sich dann im Grunde auch gleich ohne das Modell softwareseitig implementieren, ohne dafür die Behandlung des XCONCUR-Datenformates umsetzen zu müssen. Es ist nämlich wegen der fehlenden XML-Kompatibilität eher nicht damit zu rechnen, dass auf der Basis des XCONCUR-Formates eine große Anzahl von Werkzeugen zur Weiterverarbeitung der Daten, insbesondere zur Suche und Visualisierung, entwickelt werden wird.

Es bleibt abzuwarten, in welche Richtung sich XCONCUR weiterentwickelt. Mit einem intuitiv zu bedienenden Annotationswerkzeug und einer umfassenden Ausstattung an Schnittstellen zu anderen Datenformaten könnte es durchaus eine interessante Rolle spielen, die Probleme des Nicht-XML-Formates würden dadurch aber nicht aus der Welt geschafft.

### **3.1.5. Das NITE Object Model (NOM)**

#### **Geschichte**

Das NITE-Projekt blickt bereits auf eine recht lange Entwicklung zurück: Es war aus MATE (für *Multilevel Annotation Tools Engineering*) hervorgegangen, einem von der Europäischen Union geförderten Projekt, das eine Infrastruktur für die Arbeit mit Sprachkorpora entwickeln sollte. Dazu sollten von den beteiligten Partnern<sup>53</sup> Standards und Werkzeuge konzipiert und entwickelt werden.

---

53 Das MATE-Projekt wurde vom Natural Interactive Systems Laboratory (NISLab; Odense, Dänemark) geleitet. Partner waren:

- Centro Studi e Laboratori Telecomunicazioni S.p.A. (CSELT; Turin, Italien)
- Departament de Filologia Espanyola (DFE; Universitat Autònoma de Barcelona, Spanien)
- Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI; Saarbrücken, Deutschland)
- Human Communication Research Centre (HCRC; Edinburgh, England)
- Institut für Maschinelle Sprachverarbeitung (IMS; Universität Stuttgart, Deutschland)

Der Fokus lag – wie die Auswahl der Projektpartner schon deutlich macht – auf der Entwicklung von Technologie für Sprachkorpora, die im Bereich der Sprachtechnologie eingesetzt werden sollte.

Dabei orientierten sie sich an den Vorgaben der CES/XCES-Guidelines<sup>54</sup>, die entwickelt wurden, um möglichst viele existierende Annotationsschemata in einem gemeinsamen, standardisierten Austauschformat zu integrieren.<sup>55</sup> Die XCES-Guidelines, die aus den CES-Guidelines hervorgingen, sind eine sehr frühe und dabei relativ weitverbreitete Implementation von Standoff-Annotation (vgl. „Standoff-Annotation“ auf Seite 61).

Das MATE-Projekt wurde im Jahr 2000 vom NITE-Projekt abgelöst und soll deshalb hier nicht näher beschrieben werden – es soll aber immerhin noch erwähnt werden, dass für die Arbeit mit MATE-Daten eine Sammlung von Werkzeugen – die MATE-Workbench – existiert.<sup>56</sup>

Das NITE-Projekt, ebenfalls am NISLab in Dänemark angesiedelt, baute auf der Arbeit des MATE-Projektes auf und ersetzte dessen Datenformat mit dem NITE Object Model (NOM).

*„They (...) aimed at focusing on requirements for the markup and annotation of spoken corpora, including issues such as overlapping speech, non-hierarchical markup, transcriptions or the annotation of prosody or speech-acts while enabling cross-level coding, i.e. links between different levels of annotation.“  
(Lehmberg und Wörner 2008: 489)*

## Datenmodell und Format

Das NITE Object Model (Evert et al. 2003) basiert auf mehrwurzeligen Bäumen.

- 
- Istituto Linguistica Computazionale (ILC; Pisa, Italien)
  - Telefónica Investigación y Desarrollo (TID; Madrid, Spanien).

Siehe auch <http://mate.nis.sdu.dk/>.

- 54 Die CES/XCES-Guidelines werden in dieser Arbeit nicht näher betrachtet. Vgl. dazu Ide und Suderman (2008).
- 55 CES steht für „Corpus Annotation Standard“ und drückt damit bereits den Anspruch aus, einen Standard schaffen zu wollen.
- 56 Es war für die Anfertigung dieser Arbeit allerdings nicht möglich, eine funktionierende Version der Software zu finden. Beschreibungen der Software finden sich dagegen noch, beispielsweise unter <http://www.slt.atr.jp/cocosda/beijing/COCOSDA-mate-paper.htm>.



Mehrere Element-Bäume können also auf demselben Primärdatum basieren. Dieses Primärdatum kann sich zusätzlich noch an einer Timeline orientieren, muss dies aber nicht.<sup>57</sup> Die in NITE als Primärdaten bezeichneten Daten werden dabei bereits als eine Annotationsschicht betrachtet – es handelt sich um einen tokenisierten Text, auf den sich die weiteren Annotationen beziehen können. Zusätzlich können sich Annotationen, die keine Bezugspunkte in diesem Text haben, auch direkt auf die Timeline beziehen.

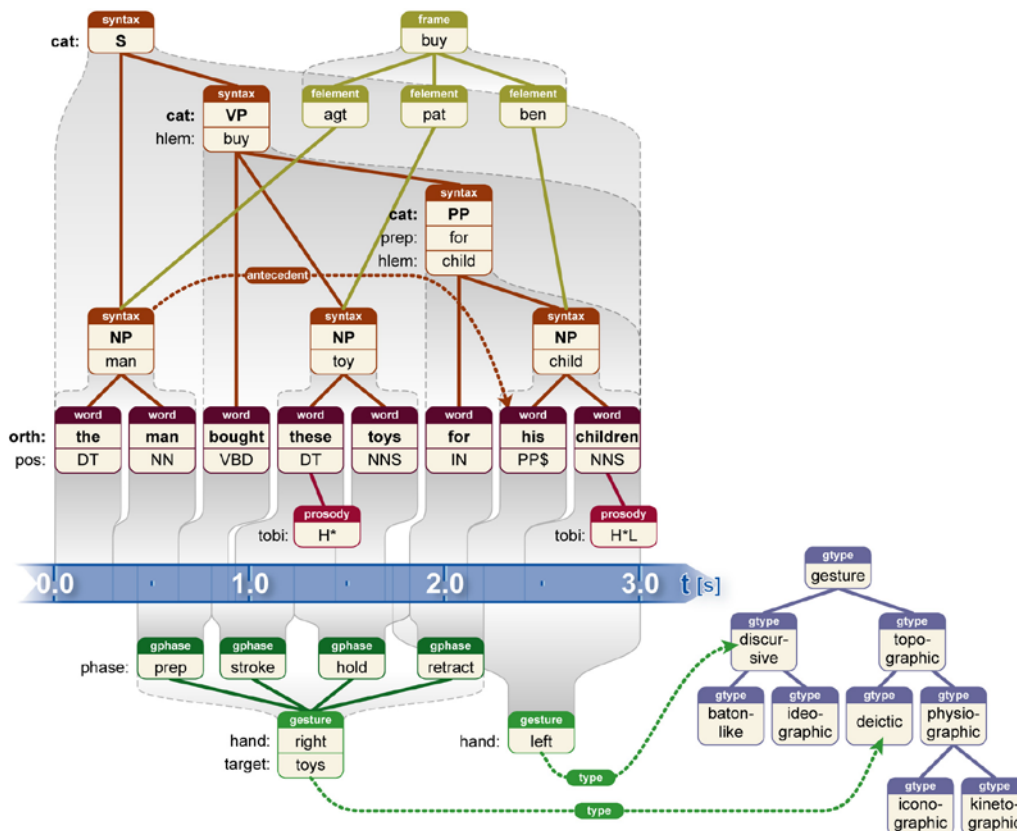


Abbildung 22: NOM-Annotationsbeispiel. (Carletta et al. 2003: 2)

In Abbildung 22 ist oberhalb der Timeline die bereits mit POS-Annotation versehene, tokenisierte „Primärdaten“-Spur zu erkennen. Auf diese Daten beziehen sich die Annotationsbäume, die sich in der Abbildung über dieser Spur (*frame*, *syntax*) und direkt darunter (*prosody*) befinden.

57 Das NITE-Format eignet sich durchaus auch für die Annotation geschriebener, linearer Texte, bei denen eine obligatorische Timeline keinen Sinn ergeben würde.

Unterhalb der Timeline befinden sich weitere Annotationen (gesture), die sich direkt auf die Timeline beziehen und deren Bezug zur orthografischen Transkription und den dazugehörigen Annotationen sich nur über Zeitpunkte auf dieser Timeline herstellen lassen.

Zusätzlich zu den mehrwurzeligen Bäumen existieren – ebenfalls in der Abbildung erkennbar – noch gerichtete Zeiger, die zwischen beliebigen Knoten der Annotation eingesetzt werden können (in der Abbildung der Zeiger antecedent, der von der Annotation der Nominalphrase auf das Wort „his“ zeigt), aber auch auf Konzepte und Ontologien zeigen können (die Zeiger, die aus der Gestenannotation auf Knoten im gtype-Baum zeigen), die außerhalb des eigentlichen Datensatzes existieren („unconstrained pointers“).

Da XML keine mehrwurzeligen Bäume zulässt, verfügt das NOM über ein eigenes Datenformat. Dieses definiert Sets von Ressourcen, aus denen sich ein NOM zusammensetzt. Diese einzelnen Ressourcen (einwurzelige Bäume für Annotationen, gerichtete Zeiger, Primärdaten) sind allerdings jeweils in XML codiert, so dass sich diese auch mit entsprechender, standardisierter XML-Software verarbeiten lassen.

## **Werkzeuge**

Die Obermenge der Werkzeuge, die zur Bearbeitung von NITE-Korpora zur Verfügung stehen, firmiert unter dem Namen Nite XML Toolkit (NXT) – aus diesem Grund wird im Zusammenhang mit diesen Daten auch oft von *NXT-Daten* gesprochen.

Wie der Name bereits suggeriert, stellt ein Teil dieser Werkzeuge eine Schnittstelle zwischen dem NOM und der Welt von XML her. Insbesondere ist im Nite XML Toolkit ein sehr umfangreiches Java-API enthalten, mit dem sich NOM-Daten in eigenen Java-Programmen auf allen Ebenen der Verarbeitung ansprechen lassen.

Die Paketstruktur des API teilt sich in die Hauptkomponenten Core, Search, GUI, Stylesheet processing, Configurable annotation tools und Datainspection – womit bereits ein Hinweis gegeben ist, welche Aufgaben sich programmtech-

nisch angehen lassen. Für den Fokus dieser Arbeit sind insbesondere die API-Teile aus dem Paket `Configurable annotation tools` interessant. Enthalten sind unter anderem Module zur Annotation von Diskursentitäten oder für die Benutzung externer Tagger. Diese bieten aber vor allem Unterstützung für die Arbeit mit den GUI-basierten Komponenten des NXT, sodass für die Programmierung eigener Komponenten eher auf die *Low-Level*-Klassen des API zurückgegriffen werden sollte.

NXT verfügt neben der Java-API auch über eine GUI-basierte Komponente, mit der sich verschiedene Schritte von Annotation und Auswertung durchführen lassen.

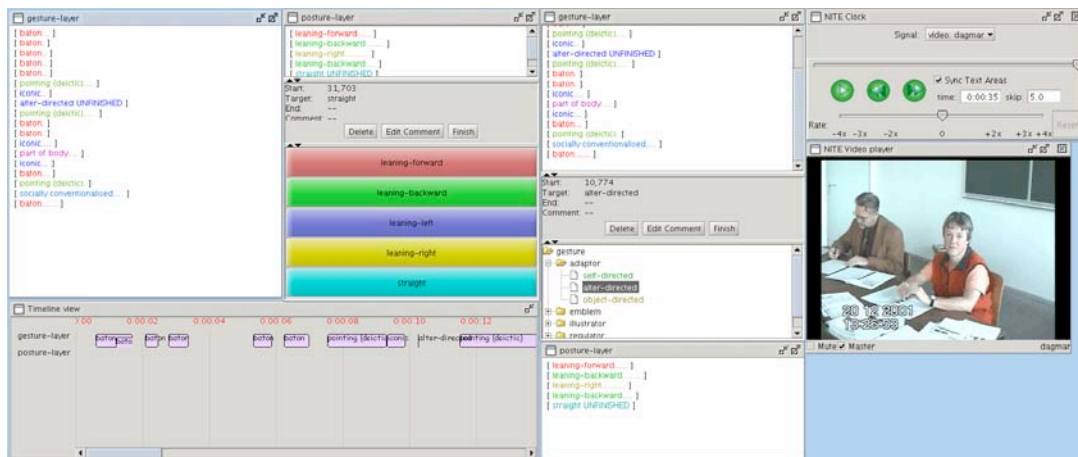


Abbildung 23: Fensteranordnung zur Gestenannotation in NXT.

Der Fokus dieser Anwendungen und der in der Distribution enthaltenen Beispiele liegt ganz klar auf der Annotation von Gesten auf unterschiedlichen Ebenen (NITE ist das Akronym für *Natural Interactivity Tools Engineering*). Es lassen sich verschiedene Anwendungsszenarien für einen Korpus-Metadatenatz anlegen und dann beispielsweise in einem Szenario per Mausklick zum laufenden Video bestimmte Gestenannotationen vornehmen. Ein anderes Szenario dient der Wiedergabe des Videos mit synchronisierter Darstellung der alignierten Annotationen. Es existieren die folgenden GUI-gestützten Werkzeuge:

Mit dem *Signal Labeller* lassen sich Labels aus einer feststehenden Liste bestimmten Abschnitten der Timeline zuordnen – also wie oben beschrieben eine Annotation direkt auf der Timeline anfertigen. Der *Discourse Entity Coder* leistet das

gleiche auf einem existierenden Text (einem NITE-Primärdatum), indem es existierende Tokens – ebenfalls aus einer feststehenden Liste – Werte zuweist. Der *Discourse Segmenter* dient dazu, existierende Texte in Segmente zu unterteilen und diesen Labels zuzuweisen. Das *non-spanning comparison display* und das *dual transcription comparison display* sind reine Darstellungskomponenten, die Annotationen bzw. Transkriptionen verschiedener Bearbeiter zum selben Primärdatum nebeneinander darstellen, um das *Inter-Annotator-Agreement* schnell erfassen zu können.

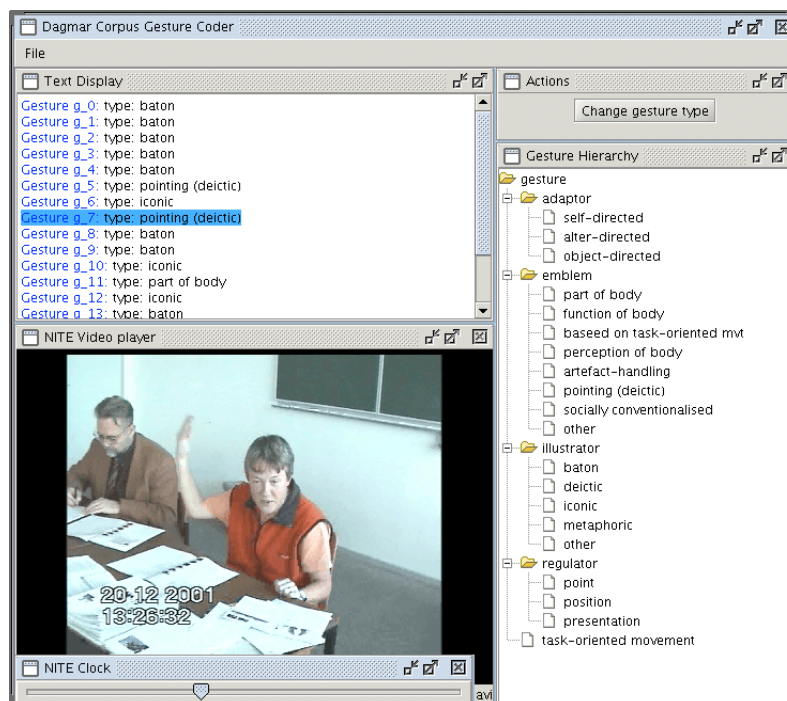


Abbildung 24: Wiedergabefenster mit Gestenannotation.

Um diese Werkzeuge benutzen zu können, müssen für ein existierendes oder zu erstellendes Korpus bestimmte Konfigurationsdateien angelegt werden, in denen die Zuordnungen zu Mediendateien, die verfügbaren Werkzeuge und anderes festgelegt werden. Für das Erstellen dieser Konfigurationsdateien existiert kein eigenes, benutzerfreundliches Programm, sodass ein Forscher ohne spezielle Kenntnisse auf Hilfe angewiesen ist.

Dabei wird ein grundsätzliches Problem der NXT-GUI-Tools deutlich: Obwohl sie alle für ihren Einsatzzweck sichtbar leistungsfähig sind und dem geübten Anwender eine hervorragende Hilfestellung für bestimmte Aufgaben geben können,

so sind sie für einen Anfänger ohne sorgfältige Einweisung und die Bereitschaft, sich auf einer sehr technischen Ebene mit dem System zu beschäftigen, nicht zu benutzen. Auch sind die Komponenten offensichtlich einerseits für die Belange bestimmter, existierender Korpora und andererseits als ein Proof-of-Concept für NOM programmiert worden, für Aufgaben wie die initiale Transkription zu einer existierenden Aufnahme existieren dagegen keine Werkzeuge.

Auch das Suchwerkzeug ist in der Lage, komplexe Abfragen über existierende NITE-Daten abzuschicken, verlangt dem Anwender aber eine eingehende Auseinandersetzung mit der Syntax der Abfragesprache ab<sup>58</sup> und bietet keine Ergebnisdarstellung, mit der sich der Kontext der Fundstelle leicht einordnen ließe.

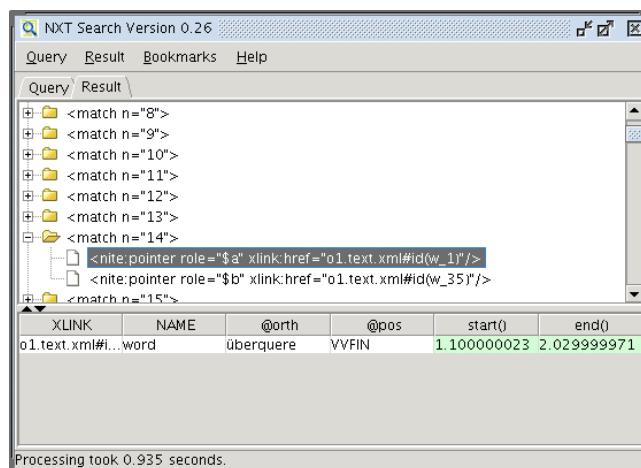


Abbildung 25: Suchergebnisse im NXT Suchfenster. Der Abschnitt auf der Timeline, auf dem sich das Suchergebnis befindet, ist zwar erkennbar, lässt sich aber nicht direkt anspringen oder auf andere Weise visualisieren.

Das NITE-Projekt lief zum Ende des Jahres 2003 aus. Der Quellcode und die Dokumentation des Datenmodells und Datenformats wurden aber in ein Sourceforge-Projekt<sup>59</sup> überführt, das auch danach noch aktiv gepflegt wurde.<sup>60</sup>

58 Dies muss nicht unbedingt als Mangel aufgefasst werden: Komplexe Daten erfordern immer komplexe Abfragen und diese lassen sich oftmals nicht vereinfachen. Es ist im Gegenteil oft zu beobachten, dass das Einführen weiterer Abstraktionsebenen zur „Vereinfachung“ nur dazu führt, dass diese Abstraktionsebenen noch zusätzlich verstanden werden müssen.

59 <http://sourceforge.net/projects/nite/>

60 Die letzten Änderungen auf der Sourceforge-Seite erfolgten im April 2008. Da die letzten (auf der NITE-Seite aufgeführten) Projekte, die NITE aktiv unterstützen, nur bis Ende 2008 finanziert wurden, wird auch die Aktivität im Sourceforge-Projekt vermutlich in Zukunft nachlassen.

## Korpora

Es existieren einige Korpora im NITE-Format, zu deren bekanntesten sicherlich das AMI-Meeting Korpus und das Switchboard-Korpus gehören.<sup>61</sup>

Eine Übersicht über weitere NITE-Korpora findet sich unter

<http://www.ltg.ed.ac.uk/NITE/inuse.html>

## Bewertung

Die Anforderungen an ein Datenmodell, die im Rahmen dieser Arbeit aufgestellt werden, erfüllt das NITE-Datenmodell nur zum Teil. Grundsätzlich ist NOM sehr flexibel, was die Möglichkeiten der Annotationen betrifft: Die Flexibilität, die sich aus der Verwendung von Standoff-Annotation ergibt, wird durch die Möglichkeit der NITE-Pointer, die Beziehungen zwischen allen Annotationsebenen sowie Konzeptbäumen außerhalb des NOM herstellen können, noch beträchtlich erweitert.

Erkauft wird diese Flexibilität allerdings damit, dass sich derartige Daten noch einmal eine Stufe schwieriger verarbeiten lassen, als dies bei Standoff-Annotationen ohnehin der Fall ist. Sollen bei Suchabfragen über NITE-Daten die NITE-Pointer mit berücksichtigt werden, so ist davon auszugehen, dass sich die Abfragezeiten noch einmal deutlich verlängern.<sup>62</sup>

NITE-Pointer können potentiell zwischen allen Ebenen der Annotation und den Primärdaten existieren. Dadurch wird es schwierig, Annotationswerkzeuge zu erstellen, die solche Pointer tatsächlich setzen können, schließlich müsste man für diesen Zweck auch Zugriff auf alle existierenden Annotationsebenen haben.

Aber auch ohne die Probleme, die durch den Einsatz der NITE-Pointer auftreten (es ist schließlich niemand gezwungen, diese auch tatsächlich zu benutzen), genügt das NITE-Modell den hier aufgestellten Forderungen an ein Datenmodell nicht. Das entscheidende Problem ist die Festlegung auf ein textuelles,

---

61 <http://corpus.amiproject.org/>  
[http://www ldc.upenn.edu/Catalog/readme\\_files/switchboard.readme.html](http://www ldc.upenn.edu/Catalog/readme_files/switchboard.readme.html)

62 Mit der Frage, ob dieser Performanceverlust in der Praxis überhaupt ein Problem darstellt, beschäftigt sich das Kapitel „Verarbeitbarkeit von Datenmodellen“ ab Seite 110.

tokenisiertes Primärdatum, das die Grundlage für diejenigen Annotationen bildet, die sich auf den Text und damit nicht direkt auf die Timeline beziehen.

In dem Moment, in dem eine orthografische Transkription eines Gesprächsereignisses angefertigt, nach Wörtern segmentiert und als Primärdatum für eine NITE-Annotation festgelegt wurde, lässt sich diese Transkription beispielsweise nicht mehr anders tokenisieren. Auch sind für ein solches Gesprächsereignis ganz andere Formen der Transkription denkbar, für die dann wiederum andere Annotationen notwendig sind.

*„(...) even NITE’s very flexible approach is not suited for all linguistic purposes. For example, it is perfectly reasonable for a linguist to annotate both an orthographic transcription and a phonetic representation of a spoken dialogue between two people. There are correspondences between these two layers, but, eventually, they are independent and have to be treated as primary data each. As structures such as these cannot be modelled by the NITE Object Model, an even more flexible approach is needed in order to provide a generic annotation format that is able to represent genuinely arbitrary linguistic data structures.“*  
(Wörner et al. 2006: 1)

Auch die Tatsache, dass die einzelnen Bäume und die Primärdaten in NOM zwar XML-Daten sind, das NOM selber dagegen den XML-Rahmen verlässt, kann nicht nur für die langfristige Verwendbarkeit derart codierter Daten ein Problem darstellen. Es ist auch aus heutiger Sicht bereits mindestens unbequem, mit Werkzeugen außerhalb des XML-Kontextes arbeiten zu müssen, wenn man die Gesamtheit eines NOM-Datensatzes bearbeiten will.

## 3.2. Gerichtete, azyklische Graphen

Auch bei Bäumen handelt es sich im Sinne der Graphentheorie um Graphen, und zwar um monohierarchische, gewurzelte Graphen mit genau einer Wurzel.

*„In the terminology of graph theory ordered hierarchies are 'ordered, rooted trees'. In linguistic theory the ancestral and ordering relations are often separately described as 'dominance' relations and 'precedence' relations.“ (Renear, Mylonas und Durand 1996: 8)*

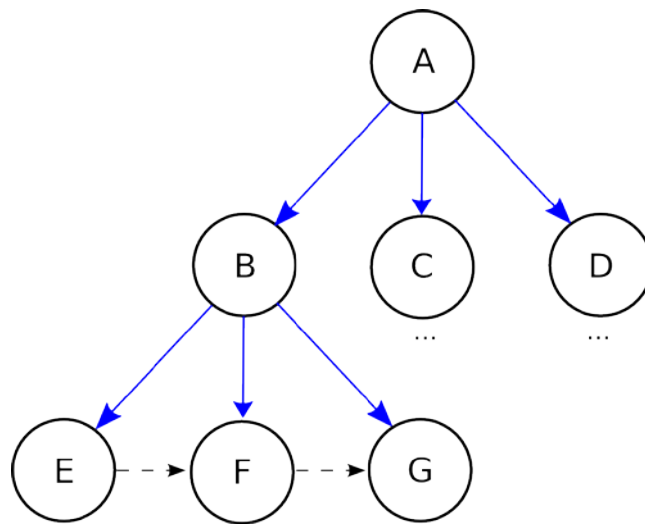


Abbildung 26: Gewurzelter Baum. In OHCO-Modellen für Texte ist die Reihenfolge der Blätter (gestrichelte Pfeile) von Bedeutung.

Der Begriff *Graphenmodelle* soll in dieser Arbeit aber für Modelle verwendet werden, die auf gerichteten, azyklischen Graphen basieren, in denen gerichtete Kanten nicht als Dominanz/Inklusion, sondern als zeitliche Abfolge interpretiert werden.

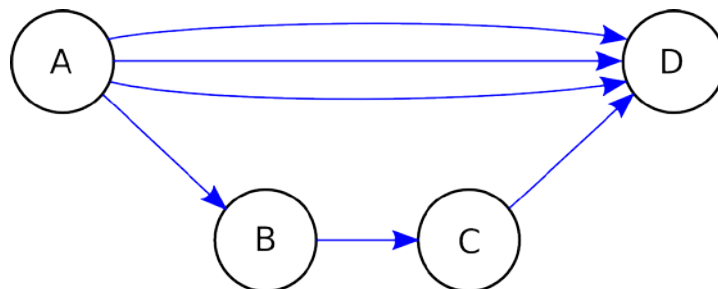


Abbildung 27: Gerichteter, azyklischer Graph.



Während diese Trennung klar zu treffen ist, existieren in der Realität der Modellierung von Sprachdaten – wie sich später zeigen wird – zwischen diesen beiden Polen einige Misch- und Übergangsformen. Die zuvor erörterten TEI-Guidelines mit ihrer Behandlung von überlappenden Hierarchien liegen in diesem Spektrum bereits nicht mehr vollständig auf einem dieser Pole, das NITE Object Model passt mit seinen unbeschränkten Zeigern und dem Zeitachsenbezug nicht mehr in das OHCO-Paradigma, modelliert die Daten aber auch nicht als gerichtete, azyklische Graphen.

Im Folgenden sollen drei grundlegende Modelle für die Modellierung gesprochener Sprachdaten vorgestellt werden, die sich eher zum Pol dieser Graphen hin orientieren:

- *Annotation Graphs* (Seite 82), die die Grundlage und gemeinsame Basis für einige existierende weitere Modelle bilden, so auch für die
- *Transkriptionsgraphen* (Seite 88) des EXMARaLDA-Modells, der Grundlage für das später erarbeitete Annotationsmodell, und das
- *Linguistic Annotation Framework* (Seite 101), ein ISO-standardisiertes Modell sowie eine existierende Implementierung (*PAULA*).

Der anwendungsorientierten Ausrichtung dieser Arbeit folgend, sollen diese Modelle nach einer kurzen Vorstellung ihrer logischen Prinzipien insbesondere unter den folgenden Gesichtspunkten betrachtet werden:

- Gibt es ein *Datenformat* für das Datenmodell?
- Sind Modell und Format dokumentiert und nachhaltig?
- Lässt das Datenmodell Annotation zu?
- Welche Arten der Annotation sind möglich?
- Existieren konkrete Verfahren (Workflows) und Werkzeuge für die Annotation, Suche und Visualisierung?
- Existieren bereits Korpora in dem Datenformat?

### 3.2.1. Annotation Graphs

#### Geschichte

Das Modell und die zugehörigen Werkzeuge zum Annotationsgraphenformalismus sind einerseits ein Ergebnis der Arbeit des LDC (Linguistic Data Consortium) an der Penn University in Philadelphia, das Korpora für verschiedene sprachtechnologische Anwendungen aufbereitet und – gegen Gebühren – vertreibt, andererseits ein Teil der Anstrengungen des ATLAS (Architecture and Tools for Linguistic Analysis Systems) Projektes<sup>63</sup>, an dem neben dem LDC noch das NIST (National Institute of Standards and Technology)<sup>64</sup> und die MITRE-Gesellschaft<sup>65</sup> beteiligt waren. Während die Arbeit innerhalb des LDC eher auf die Sprachressourcen gerichtet war, fokussierte sich die Arbeit des ATLAS-Projektes stärker auf Anwendungen zur Datenverarbeitung.

Für das Annotation Graph Toolkit existiert eine Sourceforge-Seite<sup>66</sup>, auf der die Annotation-Graph-Libraries und der Quellcode heruntergeladen werden können. Die letzte Aktualisierung erfolgte 2007.

#### Datenmodell und Format

Die Basis für den Annotationsgraphenformalismus wird in Bird and Liberman (2001) gelegt. In diesem Aufsatz wird versucht, von bestehenden Annotationsformaten für Transkriptionen gesprochener Sprache existierender Korpora (TIMIT, Partitur, CHILDES, LACITO, LDC, NIST, Switchboard, MUC-7) und Modellen, die nicht (nur) für bestimmte Korpora angelegt wurden (Emu, Festival und MATE)<sup>67</sup> auf ein gemeinsames Datenmodell zu abstrahieren, in dem sich

---

63 Die Seiten des ATLAS-Projektes existieren zum größten Teil nicht mehr. Einen Überblick gibt es unter anderem unter <http://xml.coverpages.org/atlasAnnotation.html>.

64 <http://www.nist.gov/speech/>

65 <http://www.mitre.org/>

66 <http://sourceforge.net/projects/agtk/>

67 Für nähere Informationen (alle URLs gültig Februar 2009):

**TIMIT:** <http://www ldc upenn edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1>

**Partitur:** <http://www.phonetik.uni-muenchen.de/forschung/publikationen/Granada-98-Partitur.ps>

**CHILDES:** <http://childes.psy.cmu.edu/intro/childes.pdf>

**LACITO:** <http://www ldc upenn edu/exploration/expl2000/papers/michailovsky/index.htm>

**LDC:** <http://www ldc upenn edu/Catalog/CatalogEntry.jsp?catalogId=LDC96S46>

bestehende und zukünftige Korpora abbilden lassen, und das als universelle Zwischenschicht zwischen der Anwendungsebene und der (physischen) Datenebene dienen soll.

*„Dieser Beitrag [...] konzentriert sich auf die logische Struktur linguistischer Annotationen. Wir untersuchen eine breite Auswahl existierender Formate und können zeigen, daß diesen ein gemeinsames Konzept zugrundeliegt. Dieses bildet die Grundlage für einen algebraischen Formalismus zur linguistischen Annotation, während gleichzeitig die Konsistenz zu vielen alternativen Datenstrukturen und Datenformaten erhalten bleibt.“ (Bird und Liberman 2001: 23)*

Dieser algebraische Formalismus ergibt sich aus dem beobachteten Umstand, dass in allen untersuchten Formaten und Modellen die linguistischen Annotationen über einen gemeinsamen Zeitbezug verfügen – Hierarchien werden somit explizit nicht als primäres Merkmal der Annotation angesehen.<sup>68</sup> Die eigentliche Formalisierung beschreibt dann, dass sich linguistische Annotationen immer als gerichteter, azyklischer Graph modellieren lassen.

*„A minimal formalization of this basic set of practices is a directed graph with fielded records on the arcs and optional time references on the nodes. We call these annotation graphs (AGs). We believe that this minimal formalization in fact has sufficient expressive capacity to encode, in a reasonably intuitive way, all of the kinds of linguistic annotation in use today. We also believe that this minimal formalization has good properties with respect to creation, maintenance and searching of annotation.“ (Bird und Liberman 2001: 47)*

Dabei werden alle Spezifikationen, die über die zeitliche Information hinausgehen, als Attribut-Wert-Paare an den Kanten der Graphen abgebildet. So werden beispielsweise unterschiedliche Annotationsebenen einfach dadurch gekenn-

---

**NIST:** [http://www.nist.gov/speech/tests/bnr/bnews\\_99/utf-1.0-v2.ps](http://www.nist.gov/speech/tests/bnr/bnews_99/utf-1.0-v2.ps)

**Switchboard:** [http://www ldc.upenn.edu/Catalog/readme\\_files/switchboard.readme.html](http://www ldc.upenn.edu/Catalog/readme_files/switchboard.readme.html)

**MUC-7:** <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2001T02>

**Emu:** <http://annotation.exmaralda.org/index.php/Emu>

**Festival:** <http://www.cstr.ed.ac.uk/projects/festival/>

**MATE:** Siehe die Ausführungen zu NITE auf Seite 71.

68 Hierarchien werden nicht explizit gekennzeichnet, sind aber implizit dadurch repräsentiert, dass eine Kante, die mehrere andere Kanten überspannt, die den selben Anfangs- und Endknoten haben, diese dominiert.

zeichnet, dass deren Kanten zum gleichen Typ gehören.

„(...) we impose no constraints on the structure of a layer. In fact, layers themselves will not be treated specially; a layer will be modeled as the collection of arcs having the same type.“ (Bird und Liberman 2001: 40)

Die Knoten der Annotationsgraphen können, müssen aber nicht mit einer absoluten Zeitachse verknüpft sein. Dadurch, dass die Graphen azyklisch sind, alle Kanten also in dieselbe Richtung weisen, ist immer gewährleistet, dass sich eine zeitliche Ordnung für aufeinanderfolgende Knoten feststellen lässt.<sup>69</sup>

Der Annotationsgraphenformalismus beschreibt ein abstraktes Datenmodell, das als eine *Interlingua* für verschiedene Datenformate dienen kann. Bird und Liberman (2001) beschreiben daher kein konkretes Datenformat für das Modell, sondern verweisen auf Softwareanwendungen, die von dem Modell Gebrauch machen, insbesondere auf das Annotation Graph Toolkit (AGTK) und AGLib (hier noch *AG API* genannt), die für den konkreten Austausch dieser Daten verwendet werden sollen.

## Werkzeuge

Mit dem Erscheinen des Artikels wurde auch eine erste Version einer API für die Verarbeitung von Annotationsgraphen – die AGLib – veröffentlicht. Damit lassen sich Annotationsgraphen aus eigenen Programmen erstellen und modifizieren.

Im Rahmen der Arbeiten des ATLAS-Projektes wurde aber – basierend auf dem Annotationsgraphenformalismus – auch ein auf XML basierendes Dateiformat, das *Atlas Interchange Format*, entwickelt, das die Kanten und Knoten des Annotationsgraphen einfach als eine lineare Abfolge von XML-Elementen abbildet.<sup>70</sup>

Ebenfalls in diesem Rahmen und auf der Basis von AGLib entstanden Werkzeuge (das Annotation Graph Toolkit), mit denen Endbenutzer auf der Basis von Annotationsgraphen arbeiten können sollten: *TableTrans*, *MultiTrans* und *TreeTrans*.

---

69 Eine erschöpfende Definition der formalen Eigenschaften des Annotationsgraphenformalismus findet sich – außer in Bird und Liberman (2001) – auch in Schmidt (2004: 168f).

70 <http://agtk.sourceforge.net/doc/aglib/2.0/formats.html#AG>

Die drei Werkzeuge verarbeiten alle zeitachsenbasierte Transkriptionen mit unterschiedlichem Fokus: TableTrans dient dazu, in einer tabellenartigen Darstellung Gesprächsdaten einzelner Sprecher zu erfassen, MultiTrans ist auf Gesprächssituationen mit mehreren Sprechern und potentiellen Überlappungen von Sprecherbeiträgen zugeschnitten und TreeTrans soll die Annotation hierarchischer Baumstrukturen erleichtern.

Es ist bei diesen Programmen bemerkenswert, dass sie durch die Verwendung des AGTK mit relativ wenig Code programmiert sind. Dafür sind sie aber auch – verglichen mit dem Funktionsumfang aktuell verfügbarer Software für vergleichbare Zwecke – nicht besonders leistungsfähig.



Abbildung 28: MultiTrans aus dem Annotation Graph Toolkit ist für die Transkription (genau) zweier paralleler Signalströme optimiert.

Auch die Tatsache, dass die Programme (bis auf eine Fehlerbereinigung) seit 2002 nicht mehr aktualisiert wurden, deutet darauf hin, dass sie eher als ein *proof of concept* erstellt wurden, um die Möglichkeiten der Programmierung auf der Basis des AGLib zu demonstrieren.

„This work demonstrates that the development of diverse tools and re-use of software components is greatly facilitated by a common high-level application programming interface for representing the data and managing input/output, together with a common architecture for managing the interaction of multiple components.“ (Bird et al. 2002)

So ist es mit MultiTrans zwar möglich, Transkriptionen mit zwei Sprechern zu erstellen, in denen sich Gesprächsbeiträge überlappen, es ist aber nicht möglich, diese Überlappung in grafischer Form darzustellen (wie es beispielsweise in einer Partiturdarstellung möglich ist) oder Transkriptionen mit mehr als zwei Sprechern anzulegen.

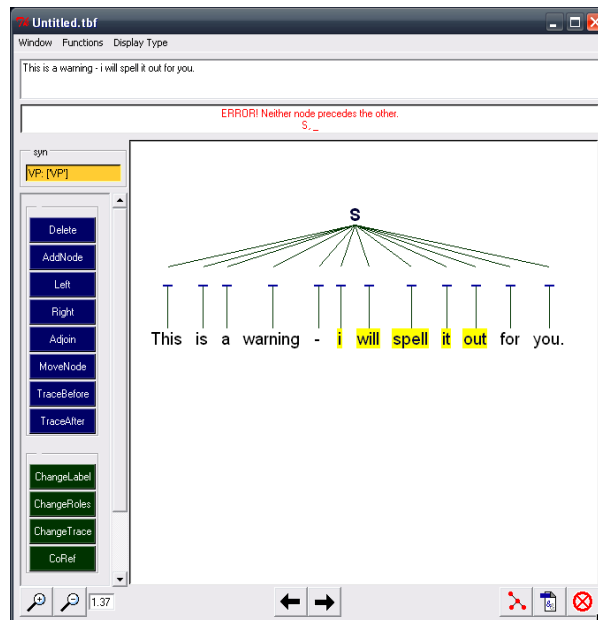


Abbildung 29: TreeTrans soll die hierarchische Annotation auf der Basis von Annotation Graphs erleichtern.

Für Visualisierung und Suche bietet die AGTK-Umgebung keine eigenen Anwendungen. Bird und Liberman (2001) verweisen lediglich darauf, dass auf gleicher Modellbasis aufbauende Programme relativ einfach so erweitert werden können, dass sie auch Annotationsgraphen verarbeiten können.<sup>71</sup>

71 Hier sind der DGA Transcriber (<http://trans.sourceforge.net/>) und der ISIP Transcriber (<http://annotation.exmaralda.org/index.php/ISIP>) erwähnt, die aber beide bislang nicht nativ mit Annotationsgraphen umgehen können. Für den DGA Transcriber ist angekündigt, dass er – auf Annotationsgraphen basierend – von Grund auf neu programmiert werden und im zweiten Quartal 2009 erscheinen soll. Zu Beginn des zweiten Quartals 2010 ist dies noch nicht geschehen.

## Bewertung

Der herausragende Verdienst des Annotationsgraphenformalismus und hier insbesondere des Bird und Liberman (2001) - Aufsatzes besteht darin, strukturelle und systematische Gemeinsamkeiten von Transkriptions- und Annotationsmodellen identifiziert und in einem formalen Rahmen beschrieben zu haben, auf den sich Forscher beziehen können, wenn sie Interoperabilität für ihre Daten erreichen wollen.<sup>72</sup>

Dabei kann das AG/AIF-Format als eine Interlingua auf dem Weg von einem Format in ein anderes dienen. Ohne den Schritt über ein drittes Format kann auch das AGLib bei einer solchen Übersetzung gewinnbringend eingesetzt werden. In der Praxis ist es sogar oftmals ausreichend, den Annotationsgraphenformalismus nur zu *denken*, um mit dieser gemeinsamen Basis eine direkte Konvertierung von einem in ein anderes Format programmieren zu können, sofern sich diese Formate mit Annotationsgraphen beschreiben lassen.

Das Modell selbst ist einerseits einfach beschreibbar und verzichtet vollständig auf Elemente, die auf einer bestimmten linguistischen Theorie basieren, ist andererseits mächtig genug, auch komplexe Transkriptions- und Annotationsszenarien bewältigen zu können. Erkauft wird diese Flexibilität – wieder einmal – mit einer möglicherweise umständlichen computergestützten Verarbeitbarkeit.

Dabei ist allerdings zu bedenken, dass dieser häufig gebrachte Einwand in vielen Kontexten – insbesondere in den Kontexten, die mit dieser Arbeit beleuchtet werden sollen – nicht die wichtige Rolle spielt, die man aus informatischer Perspektive möglicherweise annehmen würde. So ist es bei geringen Korpusgrößen und den Arbeitsabläufen, wie sie in der Gesprächs- und Konversationsanalyse gängig sind, oft nicht entscheidend, ob eine Suchanfrage in einer oder in zehn Sekunden abgearbeitet wird.<sup>73</sup>

Auch ist der Umstand, dass andere Datenmodelle (insbesondere Baumstrukturu-

---

72 Vgl. dazu Schmidt et al. (2008), in dem ein Austauschformat für multimodale Annotationen auf der Basis von Annotationsgraphen beschrieben wird.

73 Vgl. dazu auch „Verarbeitbarkeit von Datenmodellen“ ab Seite 112.

ren) von informatischer Seite besser erforscht sind und daher für ihre Verarbeitung bessere Optimierungen existieren, kein Grund, andere Datenmodelle nicht einzusetzen, wenn diese sich grundsätzlich besser für die Modellierung eines bestimmten Datentyps eignen.

Im Bezug auf die Eignung von Annotationsgraphen für die Verwendung in gesprächsanalytischen Transkriptionssystemen kritisiert Schmidt (2004: 177), dass die in Bird und Liberman (2001) vorgeschlagenen Lösungen zur Abbildung von Interpunktion und Sprecherzuweisung unbefriedigend sind. Es scheint sich hierbei allerdings nicht um ein systematisches Problem von Annotationsgraphen zu handeln, das sich nicht innerhalb des Formalismus lösen ließe – es ist eher davon auszugehen, dass die Autoren dieses Detailproblem für ihre Aufstellung außer Acht gelassen haben.

### **3.2.2. Transkriptionsgraphen (EXMARaLDA)**

#### **Geschichte**

Die Transkriptionsgraphen, die in Schmidt (2004) eingeführt und ausführlich beschrieben werden, sollen den Annotationsgraphenformalismus auf eine Weise modifizieren, dass er sich besser für die direkte Anwendung im Rahmen der gesprächsanalytischen Forschung und insbesondere für die Entwicklung entsprechender Transkriptionssoftware eignet.

Auf der Basis der Transkriptionsgraphen entstand im institutionellen Rahmen des Sonderforschungsbereichs 538 Mehrsprachigkeit in Hamburg das EXMARaLDA-System (*Extensible Markup Language for Discourse Annotation*). Das System besteht zunächst aus einem Datenmodell und einem dazugehörigen, XML-basierten Datenformat. Auf dieser Basis entstand dann ein Werkzeug zur Transkription in Partiturschreibweise (Partitur-Editor), später ein Suchwerkzeug (EXAKT) und ein Programm zur Verwaltung von Korpora und Metadaten (Coma), das allerdings mit dem Datenmodell nur mittelbar in Zusammenhang steht.

Seit Projektbeginn sind um EXMARaLDA noch eine Reihe weiterer Programme und Methoden entstanden. Einen genauen Überblick erhält man auf der Projekt-



Webseite.<sup>74</sup> Da das Forschungsprojekt, in dem EXMARaLDA entstanden ist, noch existiert (die Laufzeit geht bis Mitte 2011), wird die Architektur und die Softwareumgebung immer noch aktiv gepflegt. Das Verfahren zur Annotation von Daten gesprochener Sprache, das in dieser Arbeit entwickelt wird, geht von EXMARaLDA und den zugrunde liegenden Transkriptionsgraphen aus.

## Datenmodell und Datenformat

Schmidt (2004) analysiert die Annotationsgraphen hinsichtlich ihrer Eignung als Modell für die computergestützte Transkription gesprochener Sprache für sprachanalytische Zwecke. Dabei stellt er fest, dass sich das Modell grundsätzlich eignet (siehe auch das vorige Kapitel), für den speziellen Einsatzzweck aber Modifikationen vorgenommen werden sollten, und zwar in der Form von *zusätzlichen, strukturellen Beschränkungen* und *semantischen Spezifikationen*.

Die Modifikationen umfassen insbesondere:<sup>75</sup>

- Transkriptionsgraphen müssen *zusammenhängend* sein und dürfen nur *jeweils einen* Anfangs- und Endpunkt haben (Annotationsgraphen dürfen auch Teilgraphen umfassen, die nicht mit anderen Teilgraphen verbunden sind).
- Kanten müssen einem Sprecher und einer Kategorie (siehe dazu weiter unten) zugeordnet sein (dies ist natürlich auch in Annotationsgraphen möglich und nützlich, aber nicht explizit ausformuliert).
- Es gibt eine Unterscheidung zwischen Kanten, die *selbstständige Einheiten* darstellen und solchen, die *Merkmale dieser selbstständigen Kanten* sind und damit quasi von ersteren abhängig sind (in AG sind alle Kanten zunächst gleichwertig).
- Die unterschiedlichen Kantenkategorien sind typisiert.
- Zeitwerte werden auf rationale, positive Zahlen und Elementen auf Unicode-Strings festgelegt.

---

74 <http://exmaralda.org>

75 Vgl. dazu Schmidt (2004: 221ff).

Die Einschränkung auf zusammenhängende Graphen dürfte in der Praxis nicht als solche wahrgenommen werden: Die Möglichkeit unzusammenhängender Teilgraphen ist – genau wie die Möglichkeit mehrerer unabhängiger Zeitachsen – nur für seltene Spezialfälle erforderlich, schränkt aber die Operationalisierbarkeit der Annotationsgraphen in einem beträchtlichen Maße ein. Für die Praxis bedeutet sie insbesondere, dass auch Pausen in Transkriptionsgraphen als Kanten modelliert werden müssen.

Die weitere Spezifizierung der Kanten im Annotationsgraphen schränkt deren Einsatzmöglichkeiten ein, erleichtert aber gleichzeitig deren Kontrolle. Entscheidend ist, dass durch diese Spezifizierung erst eine systematische Visualisierung der Graphen ermöglicht wird. Um aus dem reinen Graphenmodell etablierte und potentielle neue Visualisierungen erstellen zu können, ist es unabdingbar, dass Eigenschaften wie Sprecherzuordnung, Funktion, Abhängigkeit und weitere mehr den Kanten des Graphen eindeutig zugeordnet sind.

Auch die Festlegung von Zeitfunktionen auf positive Zahlen sowie die Festlegung auf das Unicode-Zeicheninventar ist eine sinnvolle Spezifizierung, es ist aber davon auszugehen, dass auch die meisten Annotationsgraphen diesen Anforderungen entsprechen.

Das Transkriptionsgraphenmodell unterscheidet vier grundsätzliche Kantentypen:

- *Transkriptionskanten* („Typ T“) repräsentieren die tatsächlichen, gesprochenen Einheiten der Transkription.
- *Kanten des Typs „Anti-T“* verhalten sich wie Kanten des Typs „T“, enthalten jedoch nicht-phonologische Einheiten wie Pausen, Lachen oder ähnliches.
- *Deskriptionskanten* („Typ D“) beschreiben Ereignisse außerhalb der Redebeiträge, die nicht als segmentale Einheiten (wie Wörter oder Sätze) aufgefasst werden, aber trotzdem einem Sprecher zuzuordnen sind (also beispielsweise Gesten, nicht aber ein vorbeifahrendes Auto).
- *Annotationskanten* („Typ A“) repräsentieren Annotationen zu einer anderen Kante, sie müssen also immer eine Bezugskante haben, auf die sich die Annota-

tion bezieht.

- *Segmentkanten* („Typ S“) schließlich sind spezielle Annotationskanten. Sie dienen dazu, andere Kanten per Annotation zusammenzufassen – so können sie beispielsweise ein Wort, das sich auf zwei Transkriptionskanten verteilt, als solches markieren. Das eigentliche Wort bleibt dabei implizit, wird also in der Segmentkante nicht noch einmal angegeben.

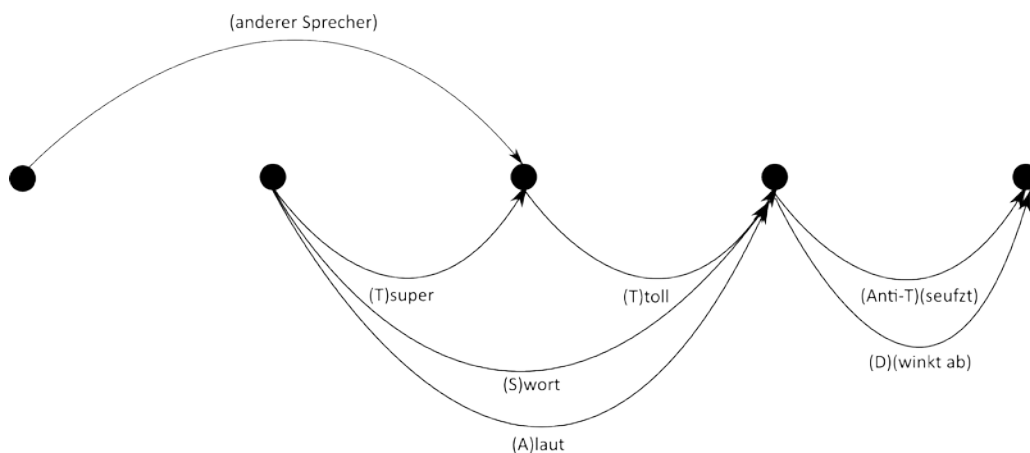


Abbildung 30: Ein Transkriptionsgraph mit allen Kantentypen: Die S-Typ-Kante fügt die T-Typ-Kanten zum Wort „supertoll“ zusammen. Die A-Typ-Kante annotiert ebenfalls die beiden T-Typ-Kanten (und könnte nicht ohne diese stehen), die D-Typ-Kante beschreibt eine Geste und annotiert einen nicht-phonologischen Laut, der mit einer Anti-T-Typ-Kante beschrieben ist.

Das Transkriptionsgraphenmodell unterscheidet darüber hinaus Kanten danach, ob sie sich kombinieren, segmentieren, vereinen oder spalten lassen. *Kombinier- und Segmentierbarkeit* beziehen sich dabei auf Kanten wie die im Beispiel gezeigten Typ-T-Kanten: Diese lassen sich zu einer Kante  $(\tau)_{\text{supertoll}}$  kombinieren, diese Kante dann wiederum in zwei Kanten  $(\tau)_{\text{super}}$  und  $(\tau)_{\text{toll}}$  segmentieren.<sup>76</sup> Verein- und Spaltbarkeit bezieht sich auf Kanten wie die Annotationskante  $(A)_{\text{laut}}$ , die sich auf die beiden T-Typ-Kanten bezieht. Diese ließe sich in zwei Kanten  $(A)_{\text{laut}}$  spalten und wieder zu einer vereinen. Die semantische Interpretation des Graphen bleibt dabei die gleiche, es ändert sich lediglich die Präzision der zeitlichen Zuordnungen.

76 Natürlich ließe sie sich auch in andere Kanten, z.B. su-per-toll, spalten.

## Segmentierbarkeit

Für das Transkriptionsgraphenmodell existieren drei konkrete, XML-basierte Datenformate, von denen zwei in der praktischen Arbeit eine größere Rolle spielen:

In der *EXMARaLDA-Basistranskription* werden sogenannte *Eventgrenzen* in der Transkription als Knoten markiert. Solche Eventgrenzen werden bei der Transkription immer an den Stellen eingefügt, an denen sich etwas in der Gesprächskonstellation ändert, also beispielsweise ein Sprecherwechsel stattfindet, eine längere Pause in einem Sprecherbeitrag gemacht wird, eine Überlappung beginnt oder endet oder ein anderes Ereignis eintritt.

	0 [00.0]	1 [01.2]	2 [02.1]	3 [03.0]
<b>Herrner [v]</b>		Stimmt ja gar	nicht!	
<b>Moos [v]</b>	Immer unterbrichst	Du mich!		

Abbildung 31: Eventgrenzen erscheinen in der Partiturdarstellung als horizontale (Spalten-) Linien.



Abbildung 32: Die Eventgrenzen werden in der Basistranskription zu Knoten des Transkriptionsgraphen.

In XML wird zunächst eine *common-timeline* mit optionalen absoluten Zeitpunkten angegeben. Darauf folgen die einzelnen Spuren (*tier*), so wie sie im Partitureditor angelegt wurden, und die auf diesen Spuren liegenden Kanten des Graphen mit (in der Timeline definiertem) Start- und Endpunkt als *events*.

```
<common-timeline>
  <tli id="T1" time="0.0"/>
  <tli id="T3" time="1.25"/>
  <tli id="T2" time="2.14"/>
  <tli id="T4" time="3.0"/>
</common-timeline>
<tier id="TIE0" speaker="SPK0" category="v" type="t"
  display-name="Herrner [v]">
  <event start="T3" end="T2">Stimmt ja gar</event>
  <event start="T2" end="T4">nicht!</event>
</tier>
<tier id="TIE1" speaker="SPK1" category="v" type="t"
  display-name="Moos [v]">
  <event start="T1" end="T3">Immer unterbrichst </event>
```

```

    <event start="T3" end="T2">Du mich!</event>
  </tier>

```

Abbildung 33: XML-Darstellung (ohne Metadaten) des Beispiels.

Gesprächsanalytische Transkriptionssysteme definieren aber – wie in Kapitel „Segmentierung“ ab Seite 48 erwähnt – Segmentierungseinheiten, die von den Events der EXMARaLDA-Basistranskription erheblich abweichen. Sofern die Segmentierungseinheiten eindeutig regelhaft in den Transkriptionskonventionen festgehalten sind, kann der EXMARaLDA Partitur-Editor auf dieser Basis automatisch von Basistranskriptionen in *segmentierte Transkriptionen* konvertieren.

Wird das beschriebene Beispiel auf der Basis von HIAT-Konventionen segmentiert, so werden Wörter, Äußerungen und Sprecherbeiträge segmentiert. Dies geschieht auf der Grundlage von Regelmäßigkeiten, die die Transkriptionskonvention definiert. So werden gewisse Interpunktionszeichen als Begrenzer von Äußerungen, Leerzeichen als Wortgrenzen und Sprecherwechsel als Begrenzer von Sprecherbeiträgen erkannt und als Markup-Tags expliziert.

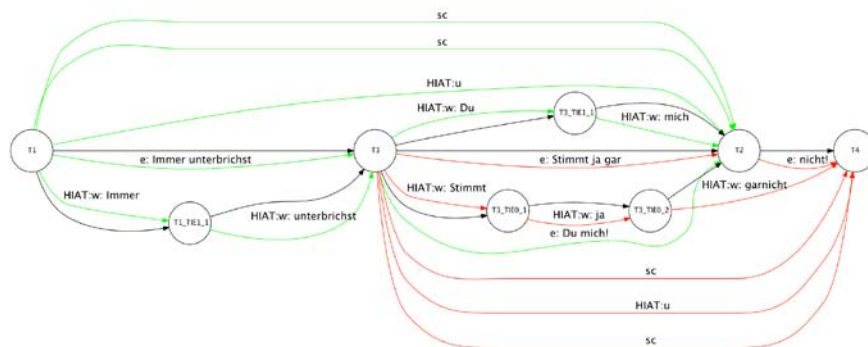


Abbildung 34: Transkriptionsgraph einer segmentierten Transkription nach HIAT.

Durch die Segmentierung entstehen – wie in der Abbildung zu erkennen – Teilgraphen, die nicht mehr auf der common-timeline liegen und deren absolute Zeitpunkte nicht bekannt sind – auch in Annotationsgraphen ist dies möglich, dort werden Graphen in Annotationsgraphen (ohne absolute Zeitinformationen), teilweise verankerte Graphen, bei denen Start- und Endpunkte auf eine absolute Zeitachse verweisen, sowie vollständig verankerte Graphen, bei denen dies für alle Knoten des Graphen gilt, unterschieden.

Auch lässt sich die zeitliche Anordnung von Knoten in unterschiedlichen Teilgraphen nicht eindeutig bestimmen. In der Abbildung liegt der Knoten T<sub>3</sub>\_TIE<sub>1</sub>\_1 (zwischen *Du* und *mich*) zwischen den beiden Knoten T<sub>3</sub>\_TIE<sub>0</sub>\_1 und T<sub>3</sub>\_TIE<sub>0</sub>\_2 – die tatsächliche zeitliche Abfolge lässt sich aber nicht bestimmen. Die Reihenfolge ließe sich nur durch manuelles Zuordnen von absoluten Zeitwerten (durch Abhören) rekonstruieren, was in der Praxis in der Regel nicht möglich ist.

Im XML der segmentierten Transkriptionen bleibt die common-timeline erhalten – zum einen, um aus einer segmentierten wieder eine Basistranskription rekonstruieren zu können, zum anderen fallen Eventgrenzen häufig mit Segmentgrenzen zusammen und können für diesen Zweck „weiterverwendet“ werden.

```
<common-timeline>
  <tli id="T1" time="0.0"/>
  <tli id="T3" time="1.25"/>
  <tli id="T2" time="2.14"/>
  <tli id="T4" time="3.0"/>
</common-timeline>
<segmented-tier id="TIE0" speaker="SPK0" category="v" type="t" display-
name="Herrner [v]">
  <timeline-fork start="T3" end="T2">
    <tli id="T3.TIE0.1"/>
    <tli id="T3.TIE0.2"/>
  </timeline-fork>
  <segmentation name="SpeakerContribution_Utterance_Word" tierref="TIE0">
    <ts n="sc" id="Seg_0" s="T3" e="T4">
      <ts n="HIAT:u" id="Seg_2" s="T3" e="T4">
        <ts n="HIAT:w" id="Seg_4" s="T3" e="T3.TIE0.1">Stimmt</ts>
        <nts n="HIAT:ip" id="Seg_5"><![CDATA[ ]]></nts>
        <ts n="HIAT:w" id="Seg_7" s="T3.TIE0.1" e="T3.TIE0.2">ja</ts>
        <nts n="HIAT:ip" id="Seg_8"><![CDATA[ ]]></nts>
        <ts n="HIAT:w" id="Seg_10" s="T3.TIE0.2" e="T4">garnicht</ts>
        <nts n="HIAT:ip" id="Seg_11">!</nts>
      </ts>
    </ts>
  </segmentation>
  <segmentation name="SpeakerContribution_Event" tierref="TIE0">
    <ts n="sc" id="Seg_12" s="T3" e="T4">
      <ts n="e" id="Seg_14" s="T3" e="T2">Stimmt ja gar</ts>
      <ts n="e" id="Seg_16" s="T2" e="T4">nicht!</ts>
    </ts>
  </segmentation>
</segmented-tier>
```

Abbildung 35: Segmentierte Transkription mit Teilgraphen.

Für die neu entstandenen Teilgraphen müssen jeweils eigene Verzweigungen der Zeitachse (*timeline-forks*) erstellt werden, da – wie erwähnt – keine zeitliche Einordnung in die *common-timeline* vorgenommen werden kann. Erkennbar ist auch, dass – für den Fall von HIAT – eine Hierarchie Sprecherbeiträge > Äußerung > Wort vorgenommen wird, sodass eine Sprecherbeitrags-Kante (*sc* = *speaker contribution*) immer eine oder mehrere Äußerungskanten (*u* = *utterance*) und diese wiederum eine oder mehrere Wortkanten (*w* = *word*) überspannt.

Am Ende einer segmentierten Transkription findet sich noch eine *conversion-info*, die das benötigte Übersetzungsvokabular expliziert, um eine segmentierte Transkription wieder in eine Basistranskription übersetzen zu können:

```
<conversion-info>
  <basic-transcription-conversion-info>
    <conversion-timeline>
      <conversion-tli id="T1"/>
      <conversion-tli id="T3"/>
      <conversion-tli id="T2"/>
      <conversion-tli id="T4"/>
    </conversion-timeline>
    <conversion-tier
      segmented-tier-id="TIE0"
      name="SpeakerContribution_Event"
      category="v"
      display-name="Herrner [v]"
      type="t"/>
    <conversion-tier
      segmented-tier-id="TIE1"
      name="SpeakerContribution_Event" category="v"
      display-name="Moos [v]" type="t"/>
    </basic-transcription-conversion-info>
  </conversion-info>
```

Abbildung 36: „Conversion Info“ zur Rekonstruktion einer Basistranskription.

In diesem Fall wird dort angegeben, dass der *SpeakerContribution\_Event*-Graph wieder in eine Typ-T-Spur in der Basistranskription überführt werden soll.

Datenmodell und Datenformat sind in mehreren Publikationen sowie Dokumentgrammatiken ausführlich beschrieben.<sup>77</sup>

---

77 Neben Schmidt (2004) finden sich weitere Publikationen unter <http://www.exmaralda.org/publikationen.html>.

## Werkzeuge

Die zentrale Komponente des EXMARaLDA-Systems ist der bereits erwähnte Partitur-Editor.

Der Partitur-Editor dient in erster Linie dem Anfertigen von Transkriptionen zu Gesprächsereignissen mit mehreren, natürlichen Sprechern, von denen es idealerweise digitale Aufzeichnungen gibt. Seit der Version 1.4 erlaubt das Programm auch die Darstellung der Wellenform einer digitalen Audioaufnahme, aus der man zu transkribierende Zeitabschnitte direkt auswählen kann, ein Verfahren, das in anderen Transkriptionswerkzeugen etabliert ist und den Prozess des Transkribierens stark vereinfacht.

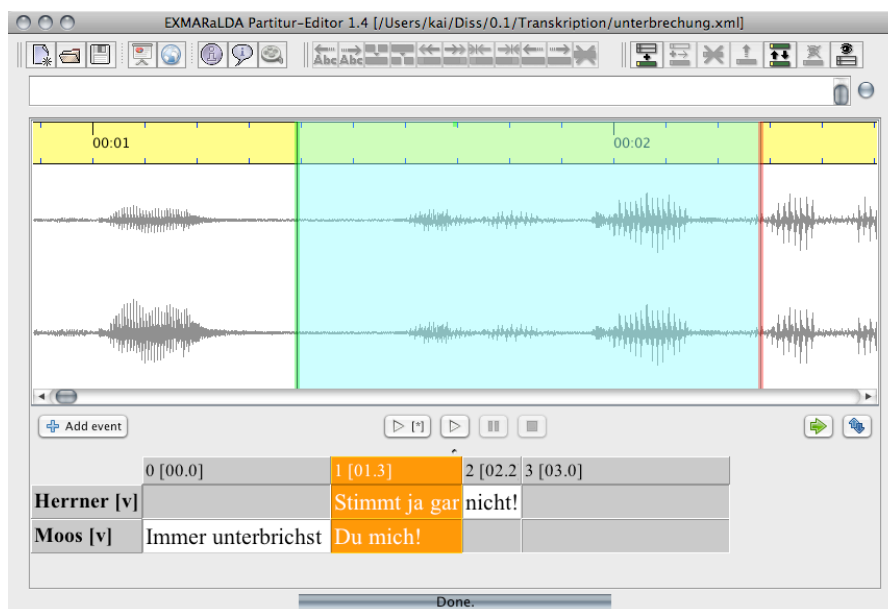


Abbildung 37: Partitur-Editor mit Wellenform-Display.

Zusätzlich kann der Partitur-Editor noch als ein Konvertierungswerkzeug zwischen verschiedenen Datenformaten genutzt werden, die auch mit interlinearem Text arbeiten und die sich als eine Form von Annotationsgraphen modellieren lassen.





Abbildung 38: Der Partitur-Editor dient auch als Schnittstelle, um zwischen verschiedenen Transkriptions- und Annotationsformaten zu übersetzen. Links sind Import-Formate, rechts Ausgabeformate aus den entsprechenden Programmkomponenten abgebildet.

Dabei kann der Partitur-Editor die Dateiformate der gängigsten, noch aktiv gepflegten Transkriptionswerkzeuge direkt öffnen und die dadurch entstandenen EXMARaLDA-Basistranskriptionen in einigen dieser Formate wieder speichern.

Die bei der Modellbeschreibung erwähnten Annotationskanten finden im Partitur-Editor ihre Entsprechung in Annotationsspuren, die Sprechern zugeordnet werden können. Näheres zur Annotation im Partitur-Editor findet sich auch unter „Bestehende Annotationsmöglichkeiten des EXMARaLDA-Systems“ ab Seite 117.

	9	40	41
<b>T [transcription]</b>	nark	gibst	insgesamt
<b>T [turn]</b>			
<b>T [sentence/MS]</b>			
<b>T [field/MS]</b>			
<b>T [sentence/KS]</b>			
<b>T [field/KS]</b>		VK3-1	NF3-1
<b>T [function]</b>	OBJEKT	PRÄDIKAT	ADVERBIAL
<b>T [Phrase]</b>	NP	VP	ADVP
<b>T [POS]</b>	NN	VVFIN	ADV

Abbildung 39: Potentiell beliebig viele Annotationsspuren erlauben die Annotation bestehender Segmente direkt im Partitur-Editor.

Das EXMARaLDA-Format besitzt einen Transkriptionskopf, in dem sich Metadaten zu Sprechern und Konstellationen ablegen lassen. Wenn diese Metadaten über Transkriptionsgrenzen hinweg in einen größeren Zusammenhang gebracht werden sollen, so lassen sie sich mit Hilfe des EXMARaLDA Corpus-Managers (Coma) zu Korpora zusammenfassen, für die dann transkriptionsübergreifend Metadaten erfasst werden können.

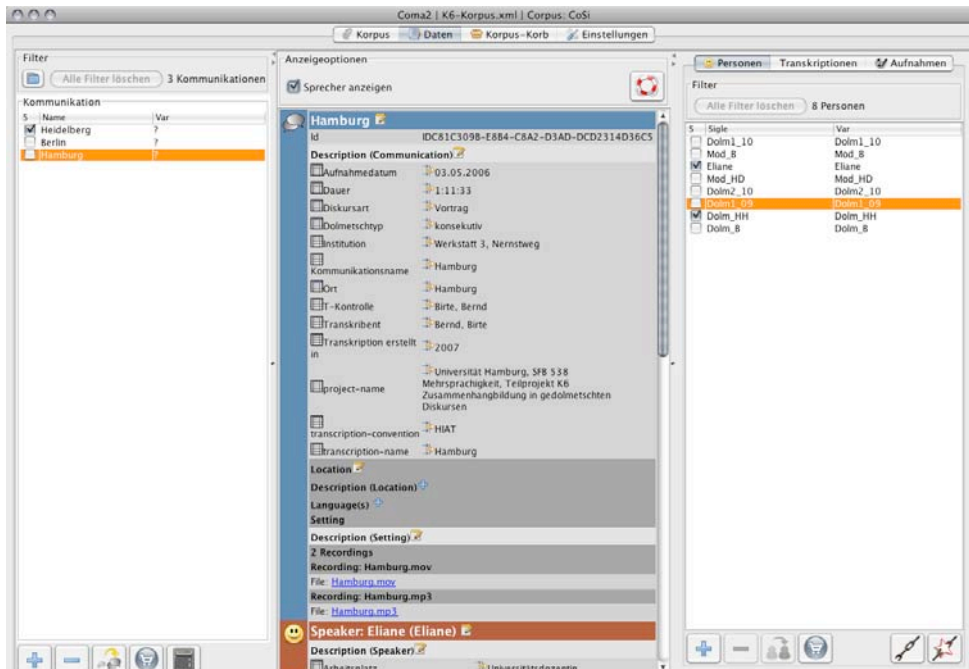


Abbildung 40: Im Corpus Manager lassen sich Metadaten zu Kommunikationsereignissen und Sprechern miteinander verknüpfen, so dass diese nicht mehrfach eingegeben werden müssen (CoSi-Korpus).

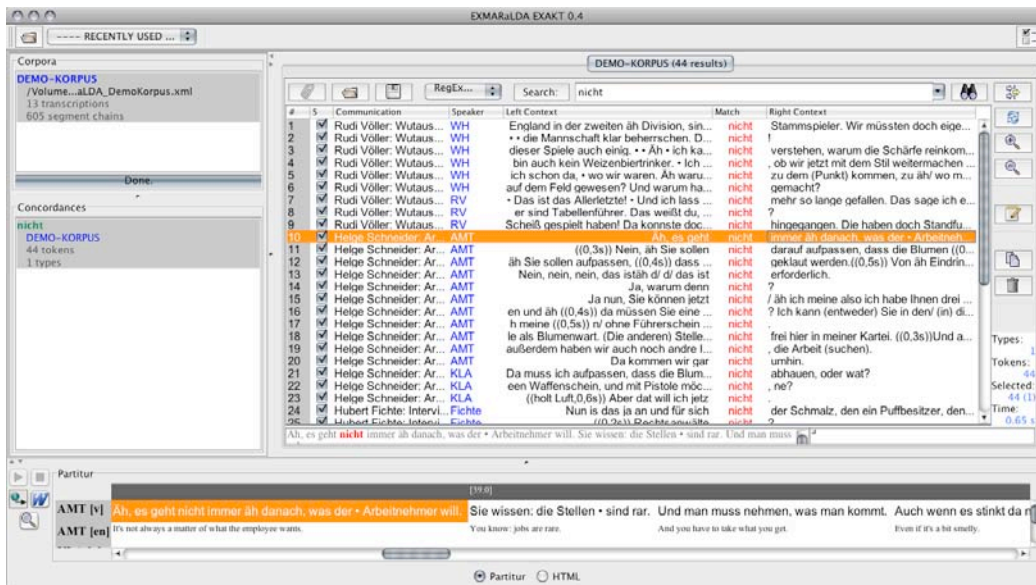


Abbildung 41: Das EXMARaLDA Analyse- und Konkordanz-Tool (EXAKT) durchsucht ganze Korpora und präsentiert die Suchergebnisse zunächst in einer einfachen Keyword-in-Context-Konkordanz (Demokorpus).

Die so entstehenden Korpora lassen sich dann im *EXMARaLDA Analyse- und Konkordanz-Tool (EXAKT)* durchsuchen.

Dabei lassen sich mit Hilfe von regulären Ausdrücken, XPath-Ausdrücken oder XSL-Stylesheets alle Spuren in EXMARaLDA-Transkriptionen durchsuchen. Die Suchergebnisse lassen sich durch Informationen aus den gesammelten Metada-

ten ergänzen, auf verschiedene Arten visualisieren und mit weiteren Informationen anreichern (näheres dazu findet sich ebenfalls unter „Bestehende Annotationsmöglichkeiten des EXMARaLDA-Systems“ ab Seite 117).

## **Korpora**

Es ist insbesondere der freien Verfügbarkeit der Werkzeuge für alle gängigen Betriebssysteme zu verdanken, dass eine Reihe von Korpora im EXMARaLDA-Datenformat existiert. Im Rahmen einer Initiative, die am SFB 538 Mehrsprachigkeit entstandenen Korpora nachhaltig aufzubereiten, wurden zusätzlich viele existierende Korpora in „alten“ Datenformaten<sup>78</sup> in das EXMARaLDA-Format überführt (Schmidt und Bennöhr 2008).<sup>79</sup>

Zusätzlich entstehen zur Zeit zwei größere Referenzkorpora auf Basis der EXMARaLDA-Korpustechnologie: das *Forschungs- und Lehrkorpus gesprochenes Deutsch* am IDS Mannheim<sup>80</sup> sowie ein Referenzkorpus des gesprochenen Türkisch (ODT-STD) an der technischen Universität (METU) Ankara<sup>81</sup>.

## **Bewertung**

Die Modifikationen, die das Transkriptionsgraphenmodell an den Annotationsgraphen vornimmt, sind für den Zweck, zu dem sie eingeführt wurden, sinnvoll und erweisen sich in der Praxis als nützlich. In Form des EXMARaLDA-Formates hat das Modell seine Tauglichkeit für die Modellierung gesprochener Sprachdaten unter Beweis gestellt.

Ohne positive Merkmale von Modell und Format abzusprechen, kann man EXMARaLDA aber auch als ein gutes Beispiel dafür sehen, dass Akzeptanz und Verbreitung eines Datenmodells in einem ganz großen Maße von der Verfügbarkeit

---

78 Es wurden dabei Daten berücksichtigt, für die es keine weiterhin aktualisierten Anwendungen mehr gibt oder die in schlecht bzw. gar nicht dokumentierten Formaten vorlagen (Daten aus Datenbank Anwendungen wie dBase oder 4th Dimension und Transkriptionen aus Programmen wie HIAT Dos und SyncWriter).

79 Eine Übersicht über die Korpora des SFB 538 findet sich unter <http://corpora.exmaralda.org/sfbkorpora.html>.

80 <http://agd.ids-mannheim.de/html/folk.shtml>.

81 [http://www.fedu.metu.edu.tr/site/eng/linkler/pub\\_detail.aspx?ID=630](http://www.fedu.metu.edu.tr/site/eng/linkler/pub_detail.aspx?ID=630).

brauchbarer Software abhängen. Das frühe Erscheinen des Partitur-Editors und seine kontinuierliche Weiterentwicklung, die sich vor allem an den Belangen der Benutzer orientierte, hat zu weiter Verbreitung der EXMARaLDA-Werkzeuge und damit des Datenformats beigetragen.

EXMARaLDA-Transkriptionen werden dabei auch über ihren ursprünglichen Einsatzzweck – die Transkription im gesprächsanalytischen Forschungszusammenhang – hinaus angewendet, und zwar vor allem deshalb, weil sich mithilfe des Partitur-Editors intuitiv Daten eingeben lassen, die dann in einem gut dokumentierten und flexiblen Datenformat weiterverarbeitet werden können.<sup>82</sup>

Eine Einschränkung, die für die Annotationsgraphen vorgenommen wurde, wirkt im Bezug auf Annotationen Probleme auf: Annotationskanten müssen sich immer auf eine oder mehrere Kanten eines bestimmten Sprechers beziehen. Dieser Umstand verhindert es beispielsweise, situationsspezifische Ereignisse zu annotieren, die nichts mit einem beteiligten Sprecher zu tun haben.

Ein weiteres Problem zeigt sich auf der Werkzeugebene: Der Partitur-Editor bearbeitet Basistranskriptionen und stellt diese mit ihren Eventgrenzen in Partiturschreibweise dar. Diese – manuell angelegten – Events lassen sich auch im Partitur-Editor mit Annotationen versehen. Die *segmentierten* Transkriptionen jedoch, die zusätzliche, automatisch generierte Information über – durch die Transkriptionskonventionen festgelegte – sprachliche Segmente enthalten, lassen sich im Partitur-Editor nicht bearbeiten.

Die durch Segmentierungsalgorithmen entstandenen Segmente stehen also für die Annotation nicht zur Verfügung. Für gesprächsanalytische Belange mag das in vielen Bereichen ausreichend sein, aber Annotationsformen wie beispielsweise manuelles POS-Tagging, das auf eine Unterteilung in annotierbare Tokens angewiesen ist, lassen sich auf diese Weise nicht im Partitur-Editor anfertigen.

Dies führt dazu, dass Transkribenten, die derartige Annotationen von Hand durchführen wollen, im Partitur-Editor an den Wortgrenzen häufig Eventgrenzen

---

82 Ein Beispiel (Fehlerannotation in geschriebenen Texten) findet sich wiederum unter „Bestehende Annotationsmöglichkeiten des EXMARaLDA-Systems“ ab Seite 117.

setzen. Dies ermöglicht zwar, direkt im Programm Wörter zu annotieren, läuft aber dem Konzept des Partitur-Editors insofern entgegen, als Ereignisgrenzen nun nicht mehr durch die zeitliche, sondern auch durch die sprachliche Struktur des Diskurses motiviert werden.

### 3.2.3. Linguistic Annotation Framework

#### Geschichte

Die ISO (International Organization for Standardization)<sup>83</sup> betreut ein „Technisches Komitee“, das sich mit der Standardisierung von Sprachressourcen beschäftigt (*Terminology and other language and content resources*, TC37). Die Organisation verfügt über eine ausgewiesene Autorität im Bezug auf Standards, schon deshalb ist es interessant, wenn sie sich mit Sprachressourcen beschäftigt.

Das Komitee ist in weitere *Special Committees* unterteilt, von denen sich eines mit *Language Resource Management* befasst und in dessen Rahmen die Standardisierung von Sprachressourcen in Bezug auf linguistische Annotation vorangetrieben werden soll. Konkret befassen sich die fünf beteiligten Arbeitsgruppen mit morphosyntaktischer und syntaktischer Annotation, der Annotation von Feature Structures und mit Datenkategorien.

Das Komitee verfolgt dabei den Ansatz, generalisierte (Meta-) Kategorien und Standards zu definieren, unter denen sich möglichst viele, spezialisierte Annotationsschemata subsumieren lassen.

Das abstrakte, generalisierte Datenmodell und -format soll dabei das *Linguistic Annotation Framework* (ISO/TC37/SC4 2008) sein, das eine entsprechend prominente Rolle in der Arbeit des Komitees spielt.

*„The core of this work is the definition of a Linguistic Annotation Framework (LAF) that is defined broadly enough to accommodate all types of linguistic annotations and provides means to represent precise and potentially complex linguistic information.“ (ISO/TC37/SC4 2008: 5)*

---

83 <http://www.iso.org>

Die Arbeiten am LAF reichen bereits bis in das Jahr 2001 zurück<sup>84</sup>, dennoch existieren bisher nur Entwürfe für einen endgültigen Standard, was selbst im Rahmen eines langwierigen ISO-Zertifizierungsprozesses ein ausgesprochen langer Zeitraum ist. Über die Gründe für diese – für die Akzeptanz des Entwurfs eher abträgliche – lange Zeitspanne lässt sich nur spekulieren – einige Gedanken dazu finden sich unter „Voraussetzungen für die Etablierung eines Standards“ ab Seite 112.

## Datenmodell und Datenformat

Für das LAF-Datenmodell werden unter anderem folgende Prinzipien aufgestellt:

- *„The data model and document form are distinct but mappable to one another*
- *The data model is parsimonious, general, and formally precise*
- *The data model is built around a clear separation of structure and content“ (ISO/TC37/SC4 2008)*

Es soll also – wie die Annotationsgraphen – ein vollständig generisches Modell darstellen. Für Annotationsgraphen werden zwar in der Modellbeschreibung keine Angaben zur Dokumentform gemacht, den weiteren Prinzipien – insbesondere, ein Modell mit klarer Trennung von Inhalt und Form zu sein – entsprechen sie aber auch.

Das Modell selbst besteht aus

*„(...) (1) a referential structure for associating stand-off annotations with primary data, instantiated as a directed graph; and (2) a feature structure representation for annotation content.“ (ISO/TC37/SC4 2008)*

Bis hierher unterscheidet sich das Modell wieder kaum von Annotationsgraphen. Neu ist, dass die Struktur der eigentlichen Annotationen auf *feature structures* festgelegt ist. Solche *feature structures*, die insbesondere aus der HPSG (Pollard

---

84 Teile des LAF beziehen sich auf CES und XCES (<http://www.cs.vassar.edu/CES/> und Lehmborg und Wörner (2008)), an denen sogar bereits seit 1993 gearbeitet wurde.

und Sag 1994)<sup>85</sup> bekannt sind, modellieren Informationen als Merkmal/Attribut-Wert-Paare, deren Werte entweder atomar oder wiederum Merkmal/Attribut-Wert-Paare sein können, wodurch feature structures zu komplexen Baumgraphen werden können.

```
<fs>
  <f name=category>noun</f>
  <f name=wordForm>Kind</f>
  <f name=proper>minus</f>
  <f name=agreement>
    <fs>
      <f name=gender>neuter</f>
      <f name=number>singular</f>
      <f name=case>nominative</f>
    </fs>
  </f>
</fs>
```

Abbildung 42: Beispiel für feature structures (hier in TEI-Schreibweise). Das Feature „agreement“ nimmt als Wert eine weitere feature structure mit atomaren Werten an. Für das Generic Modeling Tool (GMT) des LAF sind für feature structures die Tags <struct> und <feat> reserviert.<sup>86</sup>

Die Kategorien, die für die Beschreibung dieser *features* zur Verfügung stehen, werden in einem Data Category Register<sup>87</sup> festgehalten, das vom gleichen ISO-Komitee erstellt wird.

Ein weiterer Unterschied besteht in der Definition von primary data und der expliziten Behandlung von Segmentierung: Obwohl an mehreren Stellen auf die Zeitachse referenziert wird, wird als Primärdatum für die Annotation ein „stream of character data“ ohne jedes Markup angenommen, für den eine *base segmentation* definiert wird:

*„A base segmentation of primary data that defines edges between virtual nodes located between each „character“ in the primary data.“ (ISO/TC37/SC4 2008)*

Diese Segmentierung kann also Basis für Annotation und/oder für weitere Segmentierungen (beispielsweise Wörter/Tokens, Sätze, Utterances etc.) dienen. Auf diesen Segmentierungen können dann wiederum verschiedenste Annotationen

---

85 Siehe auch: <http://hpsg.stanford.edu/ideas.html>.

86 Zu feature structures in TEI vgl.: <http://www.tei-c.org/release/doc/tei-p5-doc/html/FS.html> und Langendoen und Simons (1995).

87 Zum Data Category Register (DCR) vgl.: <http://www.tc37sc4.org/WG1/doc/wg1N26.pdf>.

vorgenommen werden, die dann für genau diese Segmentierung, nicht aber für andere Segmentierungen gültig sind. Ähnlich wie bei der Beschreibung von Kanten in Transkriptionsgraphen wird für LAF unterschieden, ob Segmente andere Segmente einschließen können (wie Sätze und Wörter) oder ob sie diskontinuierlich, also miteinander verlinkte kontinuierliche Segmente sind. Die Segmentierungen werden dabei bereits klar als eine Form der Annotation bezeichnet.

Interessanterweise kommt eine absolute Zeitachse in dem Modell nur am Rande vor: Segmente, die annotiert werden können, können zusätzlich mit `startsAt` und `endsAt`-Attributen versehen werden, die den Bezug zu einer solchen Zeitachse herstellen.

Auf Annotationsgraphen wird direkt eingegangen und gezeigt, wie sich diese im LAF-Format abbilden lassen. Dass bei der Entwicklung des LAF die gesprochene Sprache offenbar nicht im Vordergrund steht, zeigt sich auch daran, dass hierarchische Strukturen einen zentralen Stellenwert für die Verwaltung linguistischer Daten haben. Bei der Auseinandersetzung mit Annotationsgraphen heißt es etwa:

*„AG is not well-suited to hierarchically organized annotations, and special mechanisms are required to deal with them.“ (Ide und Romary 2002: 147)*

Dies ist zwar richtig, umgekehrt müssen aber für LAF auch *special mechanisms* eingesetzt werden, um mit Problemen wie zeitlicher Timeline-Orientierung und Überlappung umgehen zu können.

Eine Grundkonzept des LAF besteht darin, Benutzern die Wahl ihres Annotationsformats freizustellen, solange es sich ohne Verluste in das *LAF Dump Format*<sup>88</sup> konvertieren lässt.

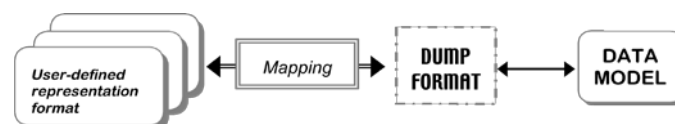


Abbildung 43: Die Architektur des LAF. (Ide und Romary 2003b: 3)

88 In einigen LAF-Publikationen wird dieses Format auch als „Pivot-Format“ bezeichnet, in den ersten Publikationen wird der Vorläufer noch als *Generic Mapping Tool* bezeichnet.



Bei dem LAF Dump-Format handelt es sich um ein Stand-Off-Format, bei dem die Annotationen auf Primärdaten zeigen, die in LAF explizit als „read only“, also schreibgeschützt definiert werden, um eine Reihe von Problemen, die im Zusammenhang mit Stand-Off-Formaten entstehen können, von vorneherein auszuschließen.

Das Dump-Format ist dabei eine von vielen möglichen Formen der Serialisierung von Daten, die im LAF-Datenmodell abgebildet werden können. Eine weitere Form ist das ISO-GrAF-Format (Ide und Suderman 2007), das es ermöglicht, die Stand-Off-Annotationen des LAF-Dump-Formates in einem einzigen Graph (und einer Datei) zusammenzuführen.

### **Werkzeuge und Daten**

Ausgewiesene Werkzeuge zur Erstellung von Ressourcen, die dem LAF-Modell folgen, existieren meines Wissens nicht. Aus dem ISO-Komitee gibt es zwar ausführliche Beschreibungen für das LAF-Dump-Format sowie für ISO-GrAF, aber keine öffentlich verfügbaren Dokumentgrammatiken oder Werkzeuge, mit denen sich Daten in kompatiblen Formaten überführen lassen.

Tatsächlich ist das Modell aber flexibel genug, um beispielsweise die anderen in dieser Arbeit beschriebenen Datenmodelle abbilden und damit im LAF-Dump-Format serialisieren zu können.

Das *Potsdamer Austauschformat für linguistische Annotationen* („PAULA“, Kepa et al. 2007) nimmt die Idee des LAF auf:

*„Paula is an XML-based standoff format for linguistic annotations, inspired by the ‚dump format‘ of the Linguistic Annotation Framework.“ (Rodriguez et al. 2007: 152)*

Das PAULA-Format verwendet – analog zum LAF-Dump-Format – mehrfache Stand-Off-Annotationsmechanismen: Auf der ersten Annotationsebene wird die Segmentierung auf dem Primärdatum bestimmt (das bei PAULA immer eine einfache Zeichenkette ist). Auf diese Ebene – die sogenannten *markables* – beziehen sich dann weitere Annotationen (*feat*-Elemente), auf die sich wiederum weitere

Annotationen beziehen können. Diese *feat*-Elemente sind aber – im Gegensatz zu LAF – nicht als feature structures, sondern als einfache Listen modelliert.

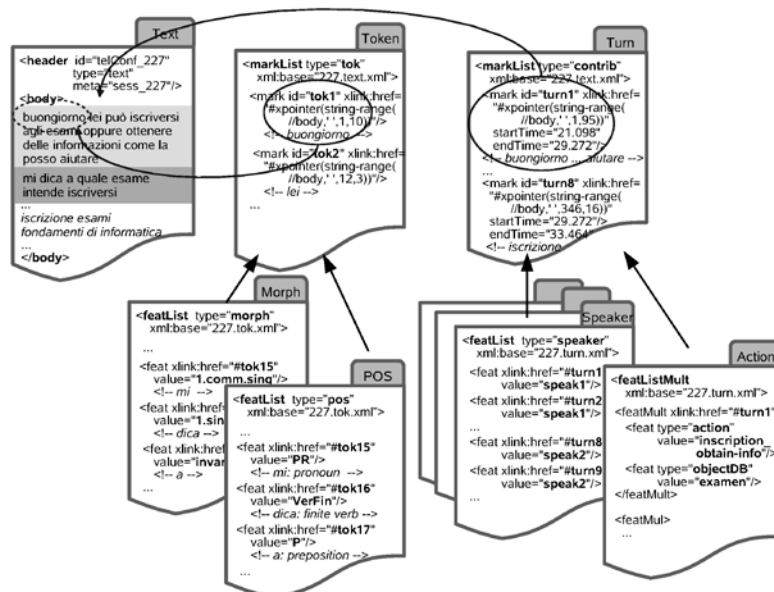


Abbildung 44: Beispiel einer PAULA-Annotation. (aus Rodriguez et al. (2007))

Analog zur Idee, die hinter dem LAF-Dump-Format steht, wurde und wird Paula auch vor allem als ein Zwischenformat eingesetzt, um annotierte Daten in die linguistische Datenbank ANNIS (Dipper, Götze und Skopeteas 2004) zu überführen. Dazu wurden Korpora, die in verschiedenen Werkzeugen in deren eigenen Formaten erstellt wurden (unter anderem im *EXMARALDA Partitur-Editor*, in *MMAX* (Müller und Strube 2006) und in *annotate* (Plaehn 1998) in das PAULA-Format überführt, um so in die Datenbank importiert werden zu können. Entsprechend existieren auch keine wirklichen PAULA-Korpora<sup>89</sup> – ebenso wenig, wie Korpora im LAF-Dump-Format zur Verfügung stehen.

## Bewertung

Ähnlich wie der Annotationsgraphenformalismus soll auch das LAF eher dem Zweck dienen, heterogene linguistische Daten in einem einheitlichen Modell abbilden zu können. Im Gegensatz zu Annotationsgraphen stellt LAF allerdings wieder baumartige Strukturen in den Vordergrund. Dies ist wieder klar dem Fo-

89 Natürlich ließen sich die PAULA-Daten wieder aus der ANNIS-Datenbank generieren, für diese gäbe es aber keine konkrete Anwendung.

kus auf einen bestimmten Datentyp geschuldet: Während Annotationsgraphen die Transkription gesprochener Sprache und die Orientierung der Daten an einer Zeitachse im Auge haben, konzentriert sich das LAF stärker darauf, beliebig komplexe Annotationen auf textuelle Inhalte abbilden zu können.

Das in der LAF-Beschreibung vorgebrachte Argument, dass sich Annotationsgraphen auch im LAF-Modell abbilden lassen, lenkt davon ab, dass das Annotationsgraphenmodell sich stärker an Modellen existierender Werkzeuge für die Transkription gesprochener Sprache orientiert. Viele Transkriptionsdaten liegen bereits implizit im Annotationsgraphenmodell vor, ohne dass sie explizit das Annotationsgraphenformat nutzen. Eine Konvertierung solcher Daten in das tatsächliche Format ist deutlich leichter zu bewerkstelligen als in das Dump-Format des LAF.

Das LAF-Modell ist, was die Komplexität der Annotationen auf den markierten Segmenten betrifft, tatsächlich flexibel genug, die meisten, wenn nicht alle Fälle von Annotation abdecken zu können, die in der linguistischen und texttechnologischen Forschung auftreten. Dazu, wie es um die Verarbeitbarkeit dieser Daten bestellt ist, kann hier keine Aussage gemacht werden, da tatsächliche Anwendungsszenarien fehlen. Die Feststellung

*„Graph theory provides a well-understood model for representing objects that can be viewed as a connected set of more elementary sub-objects, together with a wealth of graph-analytic algorithms for information extraction and analysis.“  
(Ide und Suderman 2007)*

ist zwar richtig, bedeutet aber nicht, dass dadurch bereits Verfahren zur Verfügung stünden, mit denen sich ein LAF-Graph auch effizient nach linguistischen Fragestellungen analysieren lässt.

### 3.3. Zusammenfassung

Die Betrachtung der verschiedenen Sprachdatenmodelle hat gezeigt, dass diese unter sehr unterschiedlichen Voraussetzungen entwickelt wurden:

Sie unterscheiden sich einerseits in Bezug auf die Annahme, welche Struktur Sprachdaten zugrunde liegt. Sowohl das NITE Object Model, die TEI, als auch das Linguistic Annotation Framework gehen davon aus, dass sich Sprachdaten eher in hierarchischen Strukturen modellieren lassen. Der Annotationsgraphenformalismus und die daraus abgeleiteten Transkriptionsgraphen hingegen stellen die zeitliche Struktur der Sprachdaten in den Vordergrund und modellieren sie als gerichtete Graphen mit Bezug zu einer (im Falle von Annotationsgraphen auch potenziell mehrerer) Zeitachse(n).

Diese Annahmen kommen insbesondere daher, dass tatsächlich unterschiedliche Typen von Sprachdaten im Fokus der Entwickler liegen: Der *Annotationsgraphenformalismus* und die Transkriptionsgraphen orientieren sich an Transkriptionen gesprochener Sprache mit den bereits erwähnten Besonderheiten. Das Linguistic Annotation Framework (auch wenn es gesprochene Sprache explizit mitbehandelt) und insbesondere die TEI-Guidelines orientieren sich stärker an reich annotierten Texten und ihrer inhaltlichen Struktur. Das NITE Object Model stellt noch am ehesten einen Mischfall dar und ist besonders für multimodale Annotationen geeignet.

Andererseits wurden die Modelle für unterschiedliche Einsatzzwecke erstellt. Das NITE Object Model, die TEI-Guidelines sowie die Transkriptionsgraphen wurden alle für konkrete Anwendungsfälle erdacht; dabei war im Falle der TEI-Guidelines die Lücke, die es zu füllen galt, sicherlich am größten. Das LAF und die Annotationsgraphen dagegen entstanden eher aus einem theoretischen Interesse, beide mit dem Anspruch, eher ein Modell für den universellen Austausch von Sprachdaten als für einen konkreten Anwendungsfall zu erstellen.

Eine Frage, die bei der Beurteilung von Modellen und Formaten – auch hier – immer wieder zur Diskussion gestellt wird, ist die nach der Performanz, mit der diese sich in der Praxis verarbeiten lassen. Dazu sollen hier – mit Fokus auf das

Forschungsfeld der Diskursanalyse und Gesprächsforschung – einige kurze Überlegungen einfließen.

### **3.3.1. Verarbeitbarkeit von Datenmodellen**

Die Diskussion über die praktische Einsatzfähigkeit von Datenmodellen wird häufig von der Frage dominiert, ob die computergestützte Verarbeitung die nötige Performanz erbringt. Dabei sind es immer wieder zwei Faktoren, die in die Bewertung dieser Frage einbezogen werden:

- Modelle, die hierarchische Strukturen aufweisen, werden gemeinhin als besser verarbeitbar<sup>90</sup> angesehen als Modelle, die auf gerichteten, azyklischen Graphen basieren.
- Modelle, die ihre Annotationen mit Zeigern auf die annotierten Bestandteile lösen (Standoff-Annotation) werden als schlechter verarbeitbar angesehen als solche, die die Annotationen bei den annotierten Bestandteilen halten (Inline-Annotation). Werden die Annotationen dazu noch in separaten Dateien gehalten, so sinkt der Grad der Verarbeitbarkeit noch weiter.

Für beide Faktoren ist anzunehmen, dass sie sich zum einen auf reine Erfahrungswerte stützen und dass sie zum anderen kaum in größerem Maßstab empirisch belegt sind: Die Modelle, die auf gerichteten, azyklischen Graphen basieren, werden nicht in der nötigen Breite praktisch eingesetzt, als dass sich ein derartiges Urteil rechtfertigen ließe und auch die meisten der hierarchischen Modelle haben durchaus mit Performanzproblemen zu kämpfen.

Ein klarer Vorteil für die hierarchischen Modelle steht außer Frage: Durch ihre Nähe zu XML (in den meisten Fällen sind die dazugehörigen Dateiformate in XML realisiert) lassen sie sich effizient mit etablierten Verfahren aus dem XML-Umfeld bearbeiten. Techniken wie XPath (Clark und De Rose 1999) und XQuery (Boag et al. 2007) erleichtern die Navigation in derart strukturierten Daten und stehen in etablierten Werkzeugen zur Verfügung.

---

90 „Besser verarbeitbar“ bedeutet in diesem Fall „mit besserer Performanz“ verarbeitbar: Gleiche Aufgabenstellungen lassen sich mit computergestützten Methoden in kürzerer Zeit abarbeiten.

Zwar basieren meistens auch die Datenformate für die Graphenmodelle auf XML, sie folgen aber keiner OHCO-Struktur (siehe auch Seite 55ff.), so dass sich Techniken wie XPath zur Navigation in den Daten nicht gut eignen.

Ein echter Performancegewinn wird durch die XML-Nähe aber nicht erzielt. Obwohl XML-Techniken bereits einen langen Weg zurückgelegt haben, so sind sie doch immer noch vergleichsweise behäbig (native XML-Datenbanken sind immer erst dann schnell, wenn ihre interne Datenrepräsentation wieder auf relationalen Strukturen beruht, die Verarbeitung von XML-Stylesheets ist nur dann schnell, wenn keine Verweise aufgelöst werden müssen etc. (Genevès 2004)).

Für beide Modelltypen gilt, dass sie sich in dem Moment besser verarbeiten lassen, in dem sie für den konkreten Anwendungsfall in optimierte Formen projiziert werden. Beide werden für diesen Zweck häufig in ihre Annotationsebenen zerlegt und linearisiert, um dann mit Mitteln relationaler Datenbanken bearbeitet werden zu können. Diese Möglichkeit existiert aber für beide Modelltypen, sodass sich daraus kein direkter Vorteil für einen der beiden ableiten lässt.

Mit Stand-Off-Annotation handelt man sich dagegen zwangsläufig Performanz-Defizite ein. Die Verweise von Annotation zu Annotiertem müssen immer aufgelöst werden und dieser Vorgang ist immer zeitaufwendig. Sind die Annotationen in separaten Dateien gespeichert, so kommt die Zeit für die Behandlung von Dateiverweisen noch dazu. Sobald die Annotationen aber ebenfalls linearisiert und parallel zu den annotierten Daten gehalten werden, fällt dieses Problem auch hier weg.

Daten jeder Art in relationale Datenbanken zu überführen ist aber in vielen Fällen keine praktikable Lösung: Bereits die Linearisierung ist nur von Spezialisten zu bewerkstelligen, auch muss eine technische Infrastruktur (Datenbankserver, Clientanwendungen) zur Verfügung stehen. Der finanzielle und personelle Aufwand kommt so eher für Projekte infrage, die sich mit der langfristigen Verwaltung großer Datenmengen und Korpora beschäftigen.

Man sollte sich aber fragen, ob die mangelnde Performanz im hier behandelten Bereich der Gesprächsanalyse und der Diskursforschung ein Problem darstellt,

das die Verwendung bestimmter Modelle grundsätzlich in Frage stellt. Diese Forschungsbereiche arbeiten stärker *Corpus Driven* als *Corpus Based* (Tognini-Bonelli 2001: 65ff). Sie benutzen Korpora also nicht vorwiegend dazu, Hypothesen an ihren Daten zu überprüfen, sondern um aus der Arbeit mit diesen Daten neue Hypothesen zu entwickeln.

Relevant werden Performanzfragen vor allem, wenn Gesprächsdaten für die Analyse durchsucht werden sollen. Die Ergebnisse einer solchen Suche werden in den betrachteten Bereichen meistens nicht maschinell weiterverarbeitet, sondern zumeist manuell analysiert oder für weitere Analyseschritte aufbereitet (Quantifizierung, Auswahl von Beispielen). Sollte sich herausstellen, dass eine Suchanfrage nicht die gewünschten Ergebnisse hervorgebracht hat (weil sie fehlerhaft formuliert war), so muss sie in modifizierter Form wiederholt werden.

Mit dem Suchwerkzeug EXAKT (Schmidt 2008) für das EXMARaLDA-System konnten einige Erfahrungen in Bezug auf den Zeitaufwand bei der Suche auf Korpora von Transkriptionen gesprochener Sprache gesammelt werden:

Die Suche nach dem regulären Ausdruck

`[Dd]er|[Dd]ie|[Dd]as`

auf einem mehrsprachigen Korpus (Rehbein-Korpus) mit 754.224 Wörtern<sup>91</sup> braucht beispielsweise etwas über 1 Minute, um die Suchergebnisse anschließend in einer Keyword-In-Context-Konkordanz auszugeben und dabei die Verknüpfungen in die durchsuchten Transkriptionen aufrechtzuerhalten.<sup>92</sup>

Die Performanz bleibt auf gleichem Niveau, wenn Annotationen auf EXMARaLDA-Annotationsspuren (siehe Seite 98ff.) gesucht werden. Sollte zukünftig Stand-Off-Annotation zu den Transkriptionen hinzukommen, wird sich die Performanz eher wieder verschlechtern.

Der ermittelte Wert ließe sich durchaus noch optimieren, indem man das Korpus auf einer schneller angebundenen Festplatte speichert. Auch EXAKT ist bislang

---

91 Das Korpus besteht aus 836 Transkriptionen deutsch-türkischer Diskurse.

92 Der Versuch wurde auf einem Dual-Core-iMac aus dem Jahre 2007 mit dem Korpus auf einer externen Festplatte durchgeführt.

überhaupt nicht auf Performanz optimiert – alleine die Nutzung der Multi-Threading-Möglichkeiten aktueller Betriebssysteme würde schon zu einer spürbar besseren Performanz führen.

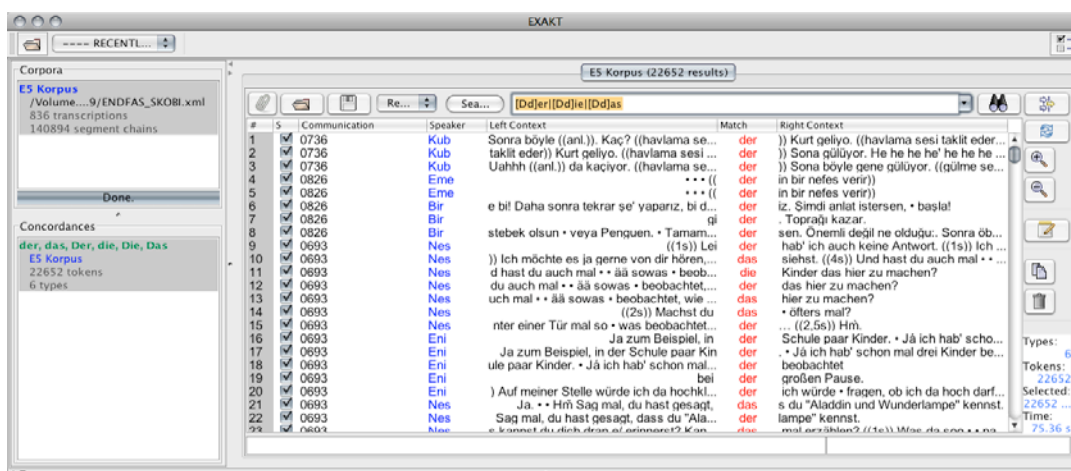


Abbildung 45: Korpusuche mit EXAKT. (Rehbein-Korpus)

Es hat sich gezeigt, dass eine solche Zeitspanne von den Forschern nicht als unzumutbar lange wahrgenommen wird, insbesondere, da derartige Suchabfragen nur den ersten Schritt und einen verschwindend kleinen Teil der Analysearbeit ausmachen.<sup>93</sup>

Insgesamt lässt sich konstatieren, dass die Vorteile des Datenformats zum einen für den Anwender (insbesondere durch intuitiv benutzbare Software), zum anderen im Bezug auf das Modell (durch die grundlegende Kompatibilität zu anderen Werkzeugumgebungen) die vermeintlichen Performancedefizite mehr als ausgleichen, zumal sich mit optimierten Verfahren der Datenhaltung und optimierten Werkzeugen noch deutliche Verbesserungen erzielen ließen und der Beweis einer besseren Performanz anderer Datenmodelle noch aussteht.

### 3.3.2. Voraussetzungen für die Etablierung eines Standards

Sowohl der Annotationsgraphenformalismus als auch das Linguistic Annotation Framework haben den Anspruch, einen Standard für die Modellierung von

93 In diesem speziellen Fall ist allerdings auch anzumerken, dass vorher überhaupt keine ähnliche Möglichkeit bestand, das Korpus als ganzes zu durchsuchen, und eine entsprechende Aufgabe „von Hand“ hätte durchgeführt werden müssen.



Sprachdaten für die Forschung zu definieren. Dafür, dass beide Modelle in der texttechnologischen Forschung auf breiter Basis diskutiert werden (so auch hier), ist es bemerkenswert, dass sich kaum wahrnehmbare Anwendungen oder Daten (Korpora) finden, welche die Tauglichkeit der Modelle für die Praxis unter Beweis stellen würden.

Das ebenfalls hier besprochene Transkriptionsgraphenmodell, das den Annotationsgraphenformalismus für einen bestimmten Anwendungszweck einschränkt, liefert dagegen die Implementation in Form von EXMARaLDA und den dazugehörigen Werkzeugen gleich mit. Der Partitur-Editor als zentrales dieser Werkzeuge ist mittlerweile schon über fünf Jahre erhältlich, und in dieser Zeit sind bereits einige umfangreiche Korpora gesprochener Sprache aus verschiedenen Bereichen der linguistischen Forschung in diesem Format entstanden, die teilweise öffentlich verfügbar sind.<sup>94</sup>

Diese Beispiele werfen ein Schlaglicht auf die Art und Weise, wie sich Standards etablieren – und wie nicht. Offensichtlich ist ein gewisses Momentum nötig, um eine Akzeptanz für einen Standard zu erreichen, und ein großer Teil dieses Momentums ergibt sich aus der tatsächlichen Anwendbarkeit der standardisierten Methode.

Das Datenformat der Childes-Datenbank<sup>95</sup> beispielsweise kann durchaus als ein Standardformat für Spracherwerbsdaten von Kindern angesehen werden, weil durch die schiere Menge an Daten, die in diesem Format vorliegen, niemand umhin kommt, sich mit dem Format auseinanderzusetzen, um die Daten benutzen zu können. Dabei ist nicht davon auszugehen, dass das zugrunde liegende Datenmodell entwickelt wurde, um einen (*den*) Standard für Gesprächsdaten zu etablieren.

So stehen sich zwei Arten von Standards gegenüber, die sehr gut die unterschiedlichen Bedeutungen des Wortes Standard repräsentieren: Auf der einen Seite Standards, die sich ungeplant durch *Etablierung* entwickeln (*Bottom-Up*), auf der

---

94 <http://corpora.exmaralda.org>

95 <http://childes.psy.cmu.edu>

anderen Seite diejenigen, die durch *technische Normierung* in einem Regelwerk festgelegt werden (*Top-Down*).

Beide haben ihre Vorzüge: Die durch Anwendung etablierten Standards haben ihre praktische Anwendbarkeit bereits bewiesen – sie sind unmittelbar nützlich. Durch Normierung entstandene Standards verfügen dagegen idealerweise über ein umfangreiches, von Spezialisten diskutiertes Regelwerk, das eindeutige und verbindliche Aussagen zu allen relevanten Bestandteilen des Standards macht. Dieser Standardisierungsprozess kann auch so weit führen, dass es zu einer endgültigen Verabschiedung gar nicht erst kommt: Der Umstand, dass es nach ungefähr zehn Jahren Arbeit am Linguistic Annotation Framework immer noch keinen verabschiedeten Standard gibt, kann als ein gutes Beispiel dafür dienen.<sup>96</sup>

Es gibt durchaus Beispiele, bei denen der *Top-Down*-Ansatz einer Standardisierung durch Normierung funktioniert: Die TEI-Guidelines (siehe „Die TEI-Guidelines“ ab Seite 57) konnten sich als ein Standard etablieren, ohne dass der Beweis ihrer Nützlichkeit vorher durch spezielle Anwendungen oder größere Sammlungen von Daten erbracht worden wäre. Tatsächlich war zur Zeit, als die TEI ins Leben gerufen wurde, der Bedarf an Vereinheitlichung so groß, dass eine Anstrengung mehrerer betroffener Projekte eigentlich nur ein Erfolg werden konnte:

*„When the Text Encoding Initiative (TEI) was originally established, scholarly projects and libraries attempting to take advantage of digital technology seemed to be faced with an overwhelming obstacle to creating sustainable and shareable archives and tools: the proliferating systems for representing textual material. These systems seemed almost always to be incompatible, often poorly designed, and multiplying at nearly the same rapid rate as the electronic text projects themselves.“ (Text Encoding Initiative 2009)*

Obwohl sicher nicht alle relevanten Aspekte von Anfang an in den TEI-Guidelines optimal gelöst worden waren, so war alleine die Tatsache, dass es nun

---

96 Eine engültige Standardisierung bei der ISO hätte indes nicht nur Vorteile: So werden die Dokumente, die den Standard beschreiben, mit dem Zeitpunkt der Standardisierung kostenpflichtig und damit insbesondere für die Forschung eher weniger attraktiv.

einen gemeinsamen Bezugspunkt gab, Anlass genug, diese Guidelines auch einzusetzen. Zum Erfolg der TEI-Guidelines trägt nicht zuletzt auch der Umstand bei, dass das Konsortium inzwischen über 20 Jahre mit einer gewissen personellen Kontinuität existiert – auch dies ist bei vielen Standardisierungsbemühungen ein kritischer Faktor.

So scheinen zwei Szenarien wünschenswert: Ein Standard, der durch Normierung entstanden ist und mit Methoden und Werkzeugen für seine Etablierung unterfüttert wird, oder ein Standard, der sich aus einem etablierten Modell ergibt und in die „höheren Weihen“ einer verbindlichen Norm gebracht wird. Beides ist aus meiner Sicht für Daten gesprochener Sprache in naher Zukunft nicht zu erwarten.

Für die Betrachtungsweise dieser Arbeit ist die Art eines *etablierten Standards* vorzuziehen: So sind das EXMARaLDA-Format und die zugrunde liegenden Transkriptionsgraphen zwar sicherlich (noch) kein Standard, sie haben aber ihre unmittelbare Nützlichkeit bereits unter Beweis gestellt und sind dabei sowohl formell als auch anwendungsbezogen gut dokumentiert. Sie lassen sich sowohl im Modell des Linguistic Annotation Framework als auch in Annotationsgraphen abbilden. Solange jedoch sonst niemand für den Austausch von Gesprächsdaten den Weg über diese Modelle geht, gibt es keinen Grund, diese Option auch tatsächlich zu nutzen.

### **3.3.3. Auswahl eines Modells**

Was die Anforderungen aus dem Kapitel „Anforderungen an das Datenmodell“ ab Seite 54 angeht, so werden diese grundsätzlich von mehreren Modellen erfüllt: Der Zeitachsenbezug lässt sich – mit Einschränkungen – mit allen vorgestellten Modellen herstellen, Überlappungen lassen sich im Grundsatz ebenso mit allen Modellen realisieren. Durch die durchgängige Verwendung von XML (in XCONCUR mit Einschränkungen) und der sich daraus ergebenden Verwendung von Unicode sollten auch alle Modelle den unterschiedlichen Schriftsystemen gewachsen sein. Bei der Segmentierung sind die Einschränkungen sowohl bei TEI, dem NITE Object Model und LAF allerdings nicht hinnehmbar, so dass nur

Annotations- und Transkriptionsgraphen übrig bleiben.

Die Grundlage für das Annotationsverfahren, das in dieser Arbeit entwickelt und ab Seite 143 vorgestellt wird, bilden die Transkriptionsgraphen und das darauf basierende EXMARaLDA-Format. Den Ausschlag dafür geben insbesondere praktische Erwägungen: Weder für den Annotationsgraphenformalismus noch für das Linguistic Annotation Framework existieren Anwendungen zur computergestützten Transkription, die sich um Annotationskomponenten erweitern ließen.

*„(...)jedes noch so umfassende und theoretisch gut begründete Konzept, selbst wenn es im engagierten und andauernden Austausch zwischen Gesprächsforschern und Informatikern entwickelt wurde, ist wenig wert, wenn nicht auch die Mittel zu seiner Umsetzung und, was letztendlich genauso wichtig ist, die Mittel zu seiner Pflege und Anpassung an die sich rasant wandelnde Computerlandschaft sichergestellt sind.“ (Schmidt 2002: 247)*

Auch stehen keine Korpora in Formaten bereit, die direkt aus diesen Modellen abgeleitet wären. Im Falle des NITE Object Model steht beides zur Verfügung, die Software entspricht aber nicht den Anforderungen, die in Gesprächsanalyse und Diskursforschung aufgestellt werden.

Für das EXMARaLDA-Format existieren bereits Werkzeuge zur Transkription, zur Visualisierung und Suche sowie zur Korpusverwaltung, die in den betrachteten Disziplinen auf breiter Basis eingesetzt werden und für die es auch bereits einige interessante Gesprächskorpora gibt. Einzig die Möglichkeiten der Annotation sind in diesen Werkzeugen noch relativ rudimentär implementiert. Im kommenden Kapitel sollen diese existierenden Möglichkeiten vorgestellt werden. Zusätzlich sollen Annotationsverfahren einiger anderer aktueller Werkzeuge beleuchtet werden, um dann auf die notwendigen Erweiterungen des EXMARaLDA-Systems überzuleiten.

## 4. Praktische Fragen zur Annotation

### 4.1. Bestehende Annotationsmöglichkeiten des EXMARaLDA-Systems

#### 4.1.1. Annotationsspuren

Bei der Beschreibung des EXMARaLDA-Datenmodells wurden die Annotationsmöglichkeiten über Annotationskanten (die im Partitur-Editor als Annotationsspuren erscheinen) bereits erwähnt: In den Basistranskriptionen lassen sich Annotationen an allen Kanten des Transkriptionsgraphen anhängen. Da die aktuelle Version des Partitur-Editors, mit dem Transkriptionen im EXMARaLDA-System angefertigt werden, diese Basistranskriptionen als Datengrundlage hat, lassen sich flache Annotationen dieser Basistranskriptionen direkt als weitere Spuren im Partitur-Editor anlegen. Diese Form der Annotation ist im Partitur-Editor durch den Tier-Typ „Annotation“ explizit vorgegeben. Annotationen sind dabei, dem Transkriptionsgraphen-Formalismus folgend, immer einem Sprecher zugeordnet:

*„Die Definition von Annotations- und Segmentkanten besagt, dass zu jeder Kante des Typs **A** oder **S** eine zusammenhängende Folge von Kanten des Typs **T** und **Anti-T** existieren muss, die demselben Sprecher zugeordnet sind. Mittels Transkriptionsgraphen formulierbare Annotationen müssen sich also per Definition auf zusammenhängende Einheiten genau eines Sprechers beziehen.“  
(Schmidt 2004: 234f)*

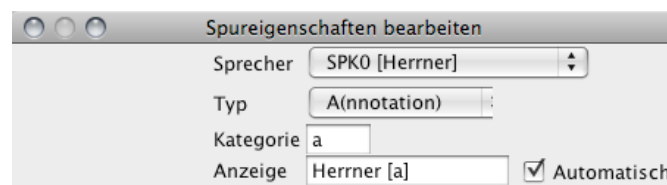


Abbildung 46: Konfigurationsdialog einer Annotationsspur im Partitur-Editor.  
A-Typ-Spuren müssen einem Sprecher zugeordnet sein.

Annotationen dieses Typs sollen sich auf Ereignisse in der annotierten Transkriptionsspur beziehen: Im folgenden Beispiel bezieht sich die Annotation „erregt“ auf das Wort „Stimmt“ in der Transkriptionsspur, liegt also ebenso zwischen den

Zeitmarken 1 und 2.

	0 [0.]	1 [1.2]	2	3 [2.1]
Herrner [v]				Stimmt ja gar nicht!
Herrner [a]			erregt	
Moos [v]	Immer unterbrichst	Du mich!		

Abbildung 47: Annotation „erregt“ in einer Annotationsspur, dem Sprecher „Herrner“ und dem Abschnitt 1-2 zugeordnet.

Die Ereignisgrenzen von Annotationen müssen sich aber nicht mit Ereignisgrenzen von Transkriptionsspuren decken: Im nächsten Beispiel hat die Annotation „feedback“<sup>97</sup> Ereignisgrenzen, die zwischen den Ereignisgrenzen der dazugehörigen Transkriptionsspur liegen – damit soll angedeutet werden, dass das Feedback eine unbestimmte Zeit später startet und eine unbestimmte Zeit später endet als das Ereignis „Immer unterbrichst“ in der Transkriptionsspur.

	0 [0.]	1	2	3 [1.2]	4	5 [2.1]	6 [3.]
Herrner [v]				Stimmt ja gar nicht!			
Herrner [a]				erregt			
Moos [v]	Immer unterbrichst	Du mich!					
msc[a]		feedback					

Abbildung 48: Von anderen Abschnitten unabhängige Annotation.

Eine derartige Annotation ist im Partitur-Editor und im EXMARaLDA-Format problemlos möglich, obwohl sie im Transkriptionsgraphenmodell nicht vorgesehen ist – für Ereignisse, deren zeitliche Reihenfolge nicht eindeutig zu bestimmen ist, muss in einem solchen Fall eine weitere Verzweigung angelegt werden.



Abbildung 49: Auf der virtuellen Zeitachse werden für die Kante des Typs D eigene Knoten (T6, T5) angelegt. Die Annotation „erregt“ ist hingegen einer segmentalen Einheit des Sprechers SPK0 zugeordnet und teilt dessen Knoten.

97 Die Annotation „feedback“ bezeichnet hier einen technischen Feedback („Rückkopplung“), sie hat nichts mit der entsprechenden Kategorie der *Conversation Analysis* zu tun.

Annotationen dieser Art müssen im EXMARaLDA-Modell in einer Spur des Typs „D“ abgelegt werden – sie ergeben im Datenmodell eine Kante des D-Typs, die folgendermaßen definiert sind:

*„Kanten des Typs D repräsentieren hingegen solche mittels Deskription beschriebene Einheiten, die nicht als Bestandteile segmentaler Einheiten aufgefasst werden, also beispielsweise eine redebegleitende Geste.“ (Schmidt 2004: 234)*

Insbesondere für solche Kanten ist es wünschenswert (aber – wie für alle anderen Kanten auch – nicht erforderlich), dass sie mit absoluten Zeitpunkten auf der Zeitachse vermerkt werden.

Annotationen, die auf diese Art und Weise direkt im Partitur-Editor vorgenommen werden, haben den entscheidenden Vorteil, direkt mit der Transkription angefertigt werden zu können, mit dieser gespeichert zu werden und ohne weitere Modifikationen in den Visualisierungen, die das EXMARaLDA-System zum jetzigen Zeitpunkt anbietet, zu erscheinen. Da sie die Regeln des Annotationsgraphenformalismus nicht verlassen, werden sie auch beim Export in andere Formate direkt mit übernommen.

Diesen Vorteilen (und der Abwesenheit von echten Alternativen) ist es wohl zu verdanken, dass diese Art der Annotation mit EXMARaLDA relativ weit verbreitet ist.

	321	322	323	324	325	326	327	328	329
[word]	Lexikon	der	Menschen	basiert	auf	verschiedene	zugrundeliegende	Mechanismen	,
[lemma]	Lexikon	d	Mensch	basieren	auf	verschieden	zugrundeliegend	Mechanismus	,
[pos]	NN	ART	NN	VVFIN	APPR	ADJA	ADJA	NN	\$,
[target_hypothesis]						verschiedenen zugrunde liegenden Mechanismen			
[corrected_pos]									
[transcripator_comment]									
[kongruenz_id]									
[kongruenz_tag]									
[reaktion_id]					x			x	
[reaktion_tag]						KaNomAkk_DatVP_324_325	KaNomAkk_DatVP_324_325		

Abbildung 50: Beispiel einer reichhaltigen Annotation im Partitur-Editor. (Falko-Korpus)

Diese Art der Annotation kann für viele Zwecke nützlich und ausreichend sein. Durch die Möglichkeit, im Partitur-Editor Zeilen ein- und ausblenden zu können, muss die Darstellung der Partitur auch nicht so ausladend sein wie im hier abge-

bildeten Beispiel.

Man kann in diesem Beispiel aber auch sofort einige der Probleme erkennen, die aus dieser Art der Annotation im Partitur-Editor hervorgehen:

- Der Partitur-Editor arbeitet mit Basistranskriptionen und stellt entsprechend in der Partitur auch nur Eventgrenzen dar. Die segmentierten Transkriptionen, deren Text basierend auf Transkriptionskonventionen in sprachliche Einheiten segmentiert ist, lassen sich im Partitur-Editor weder anzeigen noch bearbeiten. Wenn – wie in diesem Beispiel – aber Annotationen auf sprachlichen Einheiten erfolgen sollen (hier: *turns, sentences, fields, phrases, words*), so müssen die Segmente zu den einzelnen Einheiten bereits in der Basistranskription angelegt werden. Dies erhöht nicht nur den Transkriptionsaufwand beträchtlich, sondern eliminiert auch die Möglichkeit, mittels der automatischen Segmentierung eine Kontrolle über die Einhaltung von Transkriptionskonventionen zu bekommen.
- Der Vorteil, ein Gesprächsereignis gut und schnell erfassen zu können, den die Partiturdarstellung bietet, wird durch die Segmentierung in sehr kleine sprachliche Einheiten zunichte gemacht. Im obigen Beispiel ist die Grenze, ab der sich die Transkriptionsspuren noch gut lesen lassen, bereits erreicht. Eine manuelle Segmentierung in noch kleinere Einheiten, die für viele Annotationen durchaus wünschenswert sein könnte (z.B. Silben, Phoneme, auch Buchstaben) würde die Lesbarkeit weiter verschlechtern. Dazu kommt, dass die Breite der Spalten durch ihr breitestes Element bestimmt wird – bei besonders langen Annotationen würden die manuell segmentierten Einheiten noch weiter auseinandergerissen und die Lesbarkeit weiter verschlechtert.
- Der Partitur-Editor bietet keinerlei Möglichkeit, die vorgenommenen Annotationen zu validieren – nicht einmal das eingesetzte Annotationsvokabular lässt sich in irgendeiner Weise einschränken.
- Die manuell eingefügten Segmentierungen haben keinen Bezug zu den Segmentierungen, die sich mit dem Partitur-Editor über endliche Maschinen automatisch in die Transkriptionen einfügen lassen. Nur automatisch segmentier-



te Transkriptionen lassen sich aber mit den weiteren EXMARaLDA-Werkzeugen – insbesondere dem Suchwerkzeug EXAKT – weiterverarbeiten. Eine manuell eingefügte Segmentierung ist für diese Werkzeuge nicht (bzw. nur über Volltextsuche) zugänglich und damit quasi verloren.

Diese Probleme ließen sich auf unterschiedliche Weise lösen:

### **Modifikation des Partitur-Editors**

Die offensichtlichste Modifikation, die auch häufiger vorgeschlagen wird, ist, die automatische Segmentierung im Partitur-Editor sicht- und editierbar zu machen. Eine Reihe von Gründen sprechen gegen eine solche Modifikation:

- Um einen nachvollziehbaren Arbeitsablauf sicherzustellen, müsste die automatische Segmentierung nahezu in Echtzeit erfolgen, während man die segmentierten Transkriptionen bearbeitet. Während die Transkription vorgenommen wird, müsste gleichzeitig auch immer die Einhaltung der Transkriptionskonventionen überprüft werden, da bei Nichteinhaltung die automatische Segmentierung scheitert. Dies hätte zwar den theoretischen Vorteil, dass sich Fehler in Bezug zu den Transkriptionskonventionen erst gar nicht einschleichen könnten, dieser Vorteil würde aber mit einer drastischen Leistungseinbuße erkaufte und würde sicherlich störende Unterbrechungen in den Transkriptionsprozess bringen.
- Die gleichzeitige Darstellung von Eventgrenzen und Segmentierungsgrenzen würde zu einer – je nach Segmentierungsalgorithmus – sehr unübersichtlichen und schwer zu handhabenden Partiturdarstellung führen. Bereits die in der Abbildung auf Seite 119 dargestellte Partitur ist deutlich schlechter lesbar als eine Partitur, in der nur Eventgrenzen sichtbar sind. Eine zwischen Eventgrenzen und Segmentgrenzen umschaltbare Anzeige würde die potenzielle Verwirrung eher noch verstärken.
- Aus der Sicht des zugrunde liegenden Modells würde die Annotation auf segmentierten Transkriptionen innerhalb des Partitur-Editors dazu führen, dass sich die Transkription entweder nicht mehr nach einer anderen Transkriptionskonvention segmentieren ließen oder bestimmte Annotationen bei diesem

Vorgang verloren gingen. Dieses Problem stellt sich allerdings grundsätzlich bei der Annotation segmentierter Transkriptionen.

- Die Möglichkeit, in der segmentierten Transkription Änderungen vornehmen zu können, erweckt den (falschen) Eindruck, Segmente an beliebigen Stellen einfügen zu können.
- Ein Vorteil der automatischen Segmentierung besteht darin, dass ein und dieselbe Transkription potentiell mit unterschiedlichen Segmentierungsalgorithmen segmentiert werden kann. Diese Möglichkeit müsste bei editierbaren, segmentierten Transkriptionen ebenfalls berücksichtigt werden.

Diese Einwände stehen allerdings nicht der Möglichkeit entgegen, die segmentierten Transkriptionen im Partitur-Editor wenn nicht editierbar, so doch wenigstens sichtbar zu machen. Die Bearbeitung der Transkriptionen könnte dabei auf die Basistranskriptionen und deren Eventgrenzen beschränkt bleiben. Eine derartige Möglichkeit wird derzeit (März 2010) im Partitur-Editor implementiert und erprobt.

Ebenfalls 2010 wurde eine Möglichkeit geschaffen, für die Annotationsspuren sogenannte „Annotationsspezifikationen“ anzulegen, über die Annotationen aus einem festen Vokabular auswählen lassen.

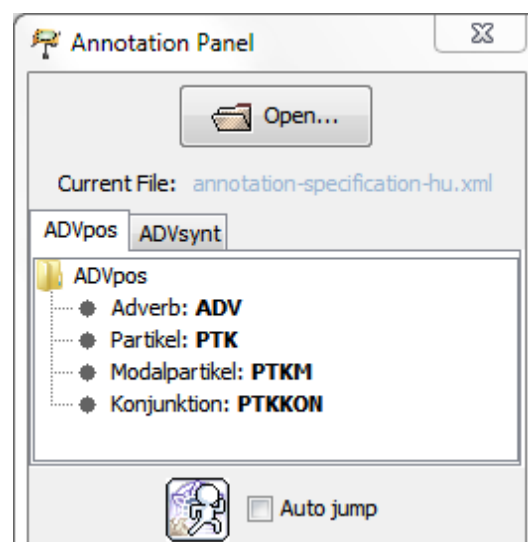


Abbildung 51: Annotationspanel mit geladener Annotationsspezifikation.

Die Akzeptanz und Verbreitung des Partitur-Editors resultiert unter anderem

auch daraus, dass seine Benutzung immer noch eine relativ niedrige Einstiegs-  
hürde bietet. Diese niedrige Einstiegschürde ergibt sich auch aus der Tatsache,  
dass er eben vor allem als Transkriptions- und nicht als Annotationswerkzeug  
entwickelt wurde. Das Erweitern um Funktionen, die komplexere Annotationen  
ermöglichen und von einem Großteil der Benutzer, die einfach ein Transkripti-  
onswerkzeug suchen, nicht benötigt werden, würde diese Hürde unnötig erhö-  
hen. Das Programm würde auch bezüglich der Verarbeitungsgeschwindigkeit  
näher an seine Grenzen gelangen, wenn man das Partiturraster noch kleinteiliger  
aufteilt.

#### **4.1.2. Annotation von Konkordanzen**

Das Werkzeug, das zum Durchsuchen von segmentierten EXMARaLDA-  
Transkriptionen entwickelt wurde, heißt *EXMARaLDA Analyse- und  
Konkordanztool (EXAKT)*.<sup>98</sup>

Mit EXAKT lassen sich EXMARaLDA-Transkriptionen mittels regulärer Ausdrü-  
cke, XPath-Ausdrücke und XSL-Stylesheets durchsuchen. Dabei lassen sich so-  
wohl transkribierter Text, Kommentar als auch Annotation durchsuchen. Das  
Ergebnis wird in Form einer Keyword-in-Context-Konkordanz (KWIC) ausgege-  
ben, wobei immer der transkribierte Text in der Konkordanz erscheint, auch  
wenn nach Kommentar oder Annotation gesucht wurde.

Über die in Konkordanzprogrammen üblichen Spalten (linker Kontext, Sucher-  
gebnis, rechter Kontext) hinaus werden standardmäßig die Transkriptionsdatei  
der Fundstelle (EXAKT sucht nicht auf einzelnen Transkriptionen, sondern im-  
mer über ganze Korpora) sowie die Sprecher, der die Äußerung getätigt hat, an-  
gegeben.

In der Konkordanzanzeige lassen sich weitere Spalten einblenden, die Metain-  
formationen aus der Korpusdatei, die im EXMARaLDA-Corpus-Manager<sup>99</sup> erstellt  
wurde, enthalten, also etwa biografische Informationen zu den Sprechern oder  
Metadaten zur Gesprächskonstellation. Die Fundstellen lassen sich – wie in der

---

98 <http://www.exmaralda.org/exakt>

99 <http://www.exmaralda.org/coma>

Abbildung auf Seite 124 zu sehen – auch in der selben Partiturdarstellung anzeigen, die der Partitur-Editor verwendet, um den kompletten Kontext, den Kommentar und die Annotation zu sehen, die an der Fundstelle relevant sind.

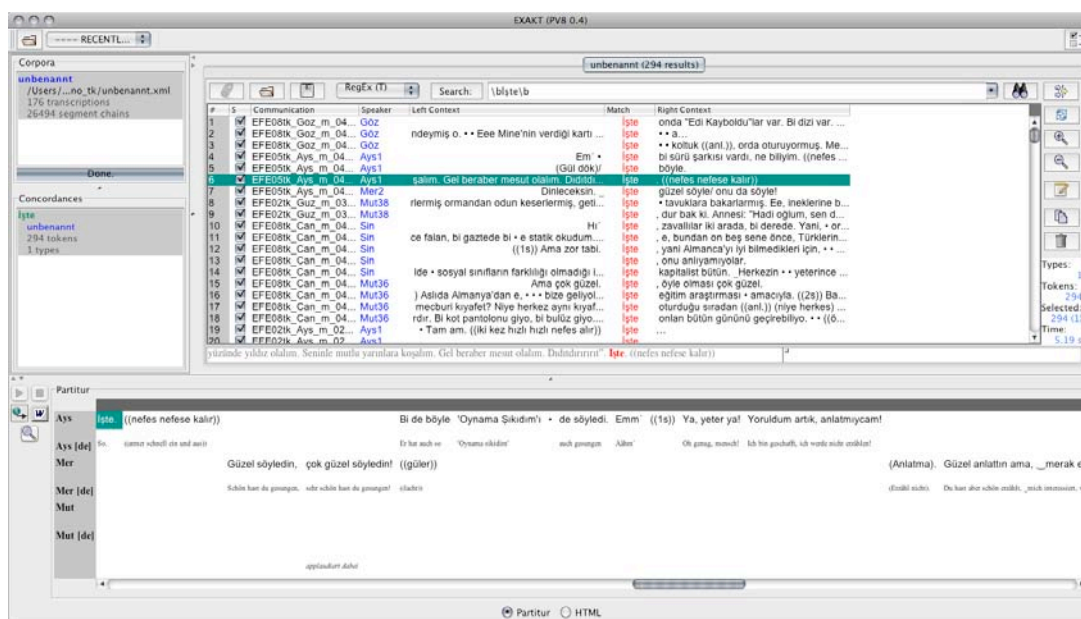


Abbildung 52: Das EXMARaLDA-Suchwerkzeug „EXAKT“ mit Suchergebnissen in Konkordanzdarstellung und Partiturdarstellung für das markierte Suchergebnis. (Rehbein-Korpus)

EXAKT bietet darüber hinaus auch die Möglichkeit, Fundstellen zu kategorisieren. Es lassen sich zu diesem Zweck beliebig viele zusätzliche Spalten („Analyse-spalten“) in der Konkordanz erzeugen, die das jeweilige Suchergebnis näher bestimmen. Diese Spalten können (zum jetzigen Stand) entweder vom Typ binär (ja/nein), geschlossene Kategorieliste oder freier Text sein.

So lassen sich den Suchergebnissen auf einfache und intuitive Weise Informationen hinzufügen. Diese Art einer *Top-Down-Annotation* wird u.a. in Smith et al. (2008) explizit gefordert:

*„In sum, there is a strong case for manual annotation of concordance lines to be treated as a central component of most linguistic research projects that use corpora. It is often an essential step in cleaning up data that has been retrieved by way of automated search strategies and it forms the basis for more complete linguistic descriptions that are in turn key to subsequent interpretations and theoretical explanations.“ (Smith, Hoffmann und Rayson 2008: 166)*

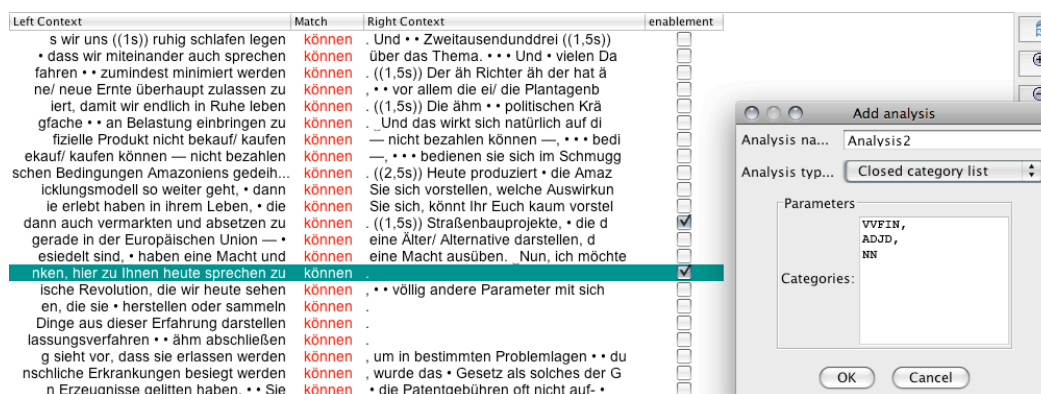


Abbildung 53: Hinzufügen von Analysespalten zur Konkordanz. (CoSi-Korpus)

Das Anlegen der Analysespalten und das Kategorisieren der Suchergebnisse mittels dieser Analysespalten ist Annotation im Sinne dieser Arbeit.

Diese Form der Annotation ist für viele Anwender, die *Corpus Driven* arbeiten, besonders bequem, da sie sich bei der Arbeit mit den Korpora quasi „natürlich“ ergibt: Die Suche nach bestimmten, an der Oberfläche erkennbaren Phänomenen, die Erstellung von Analysekatgorien und das Sortieren der gefundenen Einheiten ist ein gängiger Analyseprozess, auch und besonders mit nicht-digitalen Korpora.

EXAKT arbeitet bei der Suche auf segmentierten Transkriptionen, also solchen, die bereits im Partitur-Editor nach einem Segmentierungsalgorithmus in von den Transkriptionskonventionen bestimmte Einheiten segmentiert wurden. Die Suche beschränkt sich dabei aber nicht auf die Segmente, die aus dieser Segmentierung resultieren, sondern durchsucht (bei der Suche auf Transkriptionsspuren) die Spuren als zusammenhängenden Text.

Die Suche mittels eines regulären Ausdrucks nach dem Wort „wer“ –

`\b[ww]er\b100`

– ist also nicht darauf angewiesen, dass bereits eine Segmentierung in die Einheit „Wörter“ vorgenommen wurde – der durchsuchte String muss allerdings einzelne Wörter mit Leerzeichen trennen.

100 Der reguläre Ausdruck sucht nach allen Vorkommen des Wortes „wer“ in Groß- und Kleinschreibung. `\b` steht für eine Wortgrenze, `[Ww]` für ein kleines oder grosses „W“. Vgl. auch <http://www.regular-expressions.info/>.

Diese Art der Suche berücksichtigt im Grunde also die den Transkriptionen zugrunde liegende Datenstruktur gar nicht, stellt aber damit sicher, dass sich tatsächlich alle Phänomene auf textueller Ebene in den Transkriptionen finden lassen. Um Phänomene, die sich aus der zugrunde liegenden Struktur ergeben (wie z.B. Überlappungen und andere Kontextphänomene), bietet EXAKT andere Suchoptionen über XPath-Ausdrücke und XSL-Stylesheets, die aber vom Benutzer ein fortgeschrittenes Verständnis über die zugrunde liegende Datenstruktur voraussetzen.

Die Strategie bei der Suche mittels regulärer Ausdrücke führt dazu, dass die in der Konkordanz angezeigten Fundstellen nicht unbedingt eine Entsprechung in der Graphenstruktur des zugrunde liegenden Modells haben.

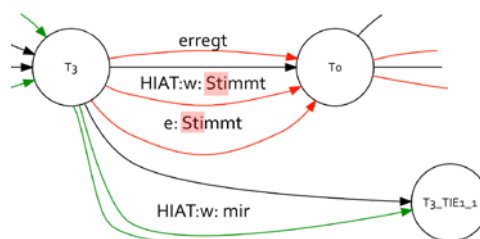
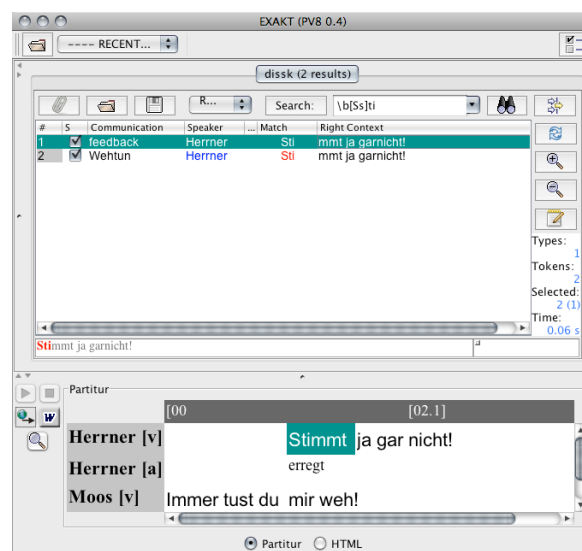


Abbildung 54: Die Suche nach Wortgrenze+„Sti“ ergibt einen Treffer, der keiner vollständigen Kante im zugrunde liegenden Transkriptionsgraphen entspricht.

Dies wiederum führt dazu, dass die Annotationen, die über die Analysespalten hinzugefügt werden, in das Datenmodell der durchsuchten Transkriptionen nicht

integriert werden können. Für das Suchwerkzeug bedeutet dies konkret, dass die Annotationen, die im Programm angefertigt werden, nicht mit den durchsuchten Transkriptionen abgespeichert werden können.

In EXAKT wird stattdessen ein eigenes Datenformat eingeführt, das die Suchergebnisse als Konkordanzen abspeichert.<sup>101</sup> Jede Zeile der Konkordanz (und damit jedes Suchergebnis) wird mit Verweisen auf die Transkription und die Position innerhalb des Graphen separat gespeichert. Dabei wird über einen Character-Offset die Anfangsposition auf einer Kante festgehalten:

```
<search-result selected="true" communication="feedback"
  speaker="Herrner">
  <locator file="wehtun_s.xml"
    xpath="//segmentation[@name='SpeakerContribution_Event']/ts[1]"/>
  <left-context/>
  <match original-match-start="0">Sti</match>
  <right-context>mmt ja garnicht!</right-context>
  <data>TIE0</data>
  <data>SPK0</data>
  <data>T3</data>
</search-result>
```

Abbildung 55: Suchergebnis in einer Konkordanzdatei.

Wie an dem Beispiel zu sehen ist, speichert EXAKT die Position der Suchergebnisse als Zeichenkette einer Event-Kante („SpeakerContribution\_Event“). Diese Kanten existieren sowohl in segmentierten- als auch in Basistranskriptionen. Die abgespeicherte Konkordanz wäre also auch noch gültig, wenn die Transkriptionen des Korpus nach einem anderen Segmentierungsalgorithmus segmentiert würden, ansonsten aber unverändert blieben.

Im Workflow einer korpuslinguistischen Analyse wären diese Konkordanzen das Ausgangsmaterial für die Quantifizierung der Suchergebnisse mit anderer, spezialisierter Software. EXAKT kann die Konkordanzen für diesen Zweck auch in anderen, gängigen Tabellenformaten ausgeben.

---

101 EXAKT ist als Werkzeug für die Analyse von Transkriptionen entwickelt worden, nicht als Annotationswerkzeug, insofern handelt es sich nicht um einen „Mangel“ des Programms. Auch hier soll der Funktionsumfang des Werkzeuges nicht überladen werden.

### 4.1.3. Annotation in anderen Werkzeugen

Die Tatsache, dass EXMARaLDA auf einer Variation des Annotationsgraphenformalismus beruht, ermöglicht eine grundsätzliche theoretische Austauschbarkeit von Daten mit anderen Systemen, die ebenfalls auf einer derartigen Basis stehen.

Über die theoretische Möglichkeit hinaus wurden im Partitur-Editor Im- und Exportfilter für einige Datenformate implementiert, auf denen verbreitete Programme zur Verarbeitung von Transkriptionen gesprochener Sprache basieren.

TASX Annotation File (\*.xml)  
Annotation Graph File (\*.xml)  
ELAN Annotation File (\*.eaf)  
PRAAT Textgrid (\*.textGrid)

Abbildung 56: EXMARaLDAs Exportformate. Über einen Segmentierungsalgorithmus lässt sich darüber hinaus auch TEI-XML exportieren.

Sofern diese Programme über Möglichkeiten der Annotation verfügen, lassen sich Transkriptionen, die ursprünglich mit EXMARaLDA-Werkzeugen angefertigt wurden, natürlich auf diese Weise annotieren. Es ist damit allerdings nicht automatisch gegeben, dass sich diese Annotationen dann auch wieder mit EXMARaLDA-Werkzeugen verwenden lassen: Nicht alle Formate, für die es Exportmöglichkeiten gibt, können auch wieder vollständig importiert werden.

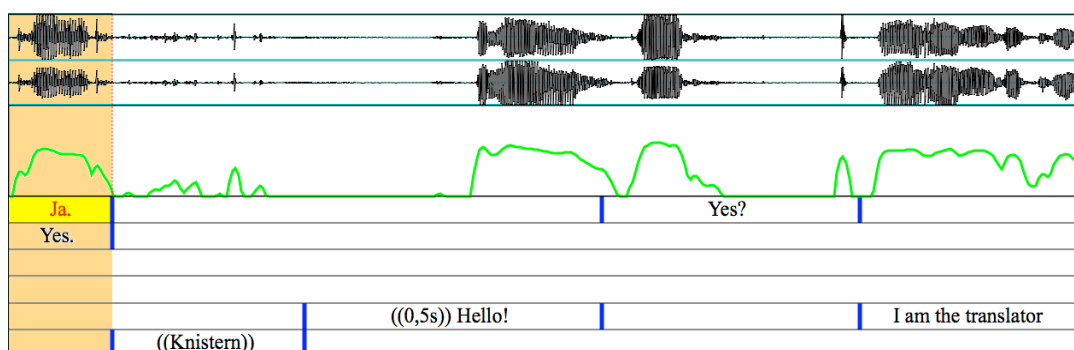


Abbildung 57: Die automatische Intensity-Analyse einer aus EXMARaLDA exportierten Transkription lässt sich nicht wieder nach EXMARaLDA zurückimportieren. (Demokorpus, English Translator)



Zumeist sind es die grundlegenden Strukturen der Transkription (analog zu EXMARaLDA-Transkriptionsspuren) mit Merkmalen wie zeitlicher Ordnung, Überlappungen etc., die sich verlustfrei übertragen lassen. Alles, was die Modelle dieser Werkzeuge an zusätzlichen Einschränkungen und Möglichkeiten bezogen auf den Annotationsgraphenformalismus bereithalten, ist nicht immer überführbar: So bietet ELAN beispielsweise die Möglichkeit, Spuren hierarchisch zu ordnen. Beim Import in EXMARaLDA werden aus „untergeordneten“ ELAN-Spuren EXMARaLDA-Annotations-Spuren; umgekehrt werden in EXMARaLDA angelegte Annotationsspuren beim Export in das ELAN-Format untergeordnete Spuren der jeweils annotierten Spur. Während bei einer derartigen Übertragung keine Information *verloren geht*, so wird doch ein Konzept in ein anderes übertragen, was logisch nicht unbedingt stimmen muss.

Grundsätzlich kann die Nutzung von EXMARaLDA-Transkriptionen in anderen Werkzeugen eine nützliche Hilfe sein, um Analysen durchführen zu können, die in EXMARaLDA-Werkzeugen nicht möglich sind. Um Annotation zu EXMARaLDA-Transkriptionen hinzuzufügen, sodass sich diese innerhalb des ursprünglichen Werkzeug-Umfeldes weiterverwenden lassen, bieten die über Im- und Export kompatiblen Programme zumeist ohnehin keine erweiterten Annotationsmöglichkeiten, die diesen Aufwand rechtfertigen würden.<sup>102</sup>

Es wäre daher wünschenswert, eine wenigstens rudimentäre Annotationsunterstützung auch innerhalb der EXMARaLDA-Werkzeugumgebung zur Verfügung zu haben. Welche Annotationsmöglichkeiten bereits existieren und wie solche Annotationsmöglichkeiten in anderen Transkriptionswerkzeugen gelöst sind, soll im folgenden Kapitel kurz beleuchtet werden.

---

102 Eine Ausnahme bilden möglicherweise die automatischen phonetischen Analysen in Praat sowie die automatische Übersetzung in IPA-Targets in PHON (Rose und Hedlund 2009); erstere lassen sich durch softwaretechnische Integration von Praat teilweise innerhalb des Partitur-Editors vornehmen, an einem Austauschformat zwischen EXMARaLDA und PHON wird zur Zeit gearbeitet.

## 4.2. Annotationsmöglichkeiten in existierenden Werkzeugen

Sowohl für schriftliche Texte als auch für Transkriptionen gesprochener Sprache existieren Werkzeuge, die Annotation in verschiedenen Automatisierungsgraden ermöglichen. Im folgenden Kapitel sollen schlaglichtartig einige Werkzeuge im Bezug auf diese Annotationsmöglichkeiten angeschaut werden, um daraus die Desiderata für eine Erweiterung des EXMARaLDA-Systems um Annotationstechniken zu ermitteln.

### 4.2.1. Werkzeuge zur automatischen Annotation von Texten

Beim größten Teil der verfügbaren Annotationswerkzeuge<sup>103</sup> handelt es sich um *Tagger*, die lineare Texte automatisch mit Annotation versehen.

Meistens wird im ersten Schritt eine Tokenisierung ausgeführt, d.h. die Texte werden – stark vereinfacht – auf der Wort- und Satzebene unterteilt. Im zweiten Schritt findet dann das Taggen statt, den vorher festgelegten Einheiten werden linguistische Informationen verschiedenster Art zugeordnet. Häufig handelt es sich dabei um Wortartzuordnungen (part-of-speech/pos-Tagging) und Lemmatisierung (oftmals Grundlage oder Hilfsmittel für die entsprechenden pos-Tagger), also die Rückführung des Tokens auf eine abstrakte Grundform.

Dabei bedienen sich die Tagger verschiedener Verfahren (Lexikon, Regeln, absolute und kontextuelle Wahrscheinlichkeiten), meistens aber einer Kombination mehrerer dieser Verfahren. Da die Erstellung der zugrunde liegenden Statistiken und Lexika einen beträchtlichen Aufwand darstellt, existieren die meisten Tagger nur für Sprachen mit großer Sprecherzahl wie Englisch, Deutsch, Spanisch und Französisch.

Viele dieser Werkzeuge haben ihre Ursprünge noch in der Prä-XML-Ära und arbeiten auf reinen Textdateien, denen sie oftmals proprietäres Annotations-Markup hinzufügen. Die Tagger selber verfügen selten über ein grafisches Benut-

---

103 Umfangreiche Übersichten über Annotationswerkzeuge gibt es im Internet unter <http://annotation.exmaralda.org>  
<http://www.linguistik.hu-berlin.de/institut/professuren/korpuslinguistik/links/software>

zerinterface, auch lassen sich in vielen Fällen wenige oder keine Parameter für den Verlauf des Taggings bestimmen.

Werden die Tagger noch weiter gepflegt, so findet auch häufig eine Anpassung an den aktuellen methodologischen Standard statt, insbesondere was die Verarbeitung von Texten mit (XML-)Markup angeht – mindestens die Ausgabe in einer Markup-Struktur ist mit den meisten Taggern inzwischen möglich, was die Arbeit mit den Texten erheblich vereinfacht.

(1)

You\_PNP always\_AV0 interrupt\_VVB me\_PNP .\_.

(2)

```
<w id="2.1" pos="PNP">You</w> <w id="2.2" pos="AV0">always</w>
<w id="2.3" pos="VVB">interrupt</w> <w id="2.4" pos="PNP">me</w>
<w id="2.5" pos=".">.</w>
```

Abbildung 58: Ausgabe eines Beispielsatzes in typischer Tagger-Syntax mit nachgestelltem POS-Tag (1) und in TEI-konformer XML-Syntax (2).

Für weitverbreitete Tagger existieren verschiedene Benutzeroberflächen, die Anfängern einen leichteren Zugang ermöglichen.

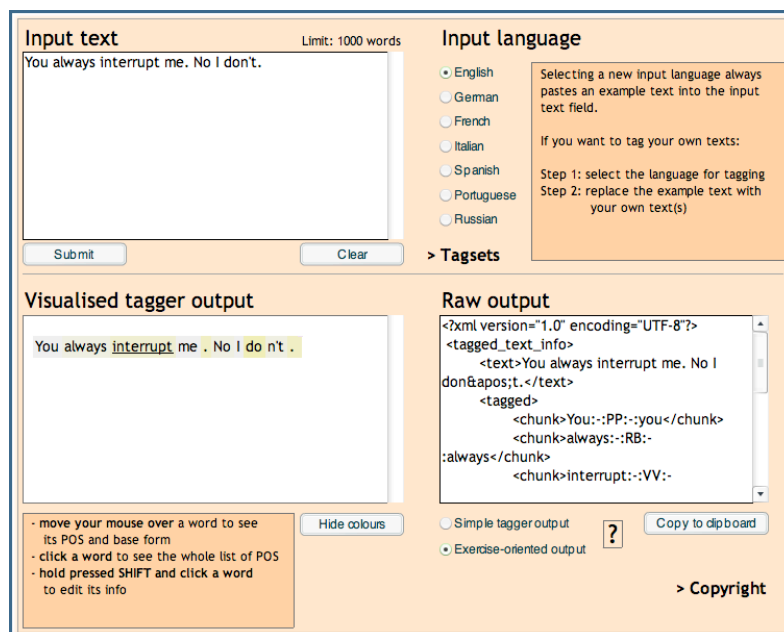


Abbildung 59: Online-Interface für den TreeTagger.  
<http://www.cele.nottingham.ac.uk/~ccztk/treetagger.php>

Einige Tagger sind darüber hinaus in der Lage, Texte auf Satzstrukturen zu parsen. Dieses Parsing setzt aber das POS-Tagging immer voraus – so existieren Parser, die auf bereits POS-annotierte Texte angewiesen sind und kombinierte Tagger/Parser, die zunächst das POS-Tagging und darauf aufbauend das Parsing der Satzstrukturen vornehmen.

```

Your sentence
  You always interrupt me.

Tagging
  You/PRP always/RB interrupt/VBP me/PRP ./

Parse
  (ROOT
   (S
    (NP (PRP You))
    (ADVP (RB always))
    (VP (VBP interrupt)
        (NP (PRP me)))
    (. .)))
  
```

Abbildung 60: Ausgabe der Online-Version des Stanford-Parsers:  
 Das Parsing der Satzstruktur setzt das vorherige POS-Tagging voraus.  
<http://nlp.stanford.edu:8080/parser/>

Die resultierende Struktur ist nun hierarchisch. In der Textausgabe des Parsers ist die Hierarchie durch die abgebildete Klammersyntax repräsentiert, sie ließe sich auch in einem typischen Baumdiagramm abbilden:

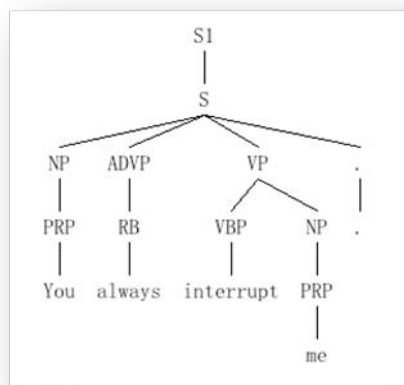


Abbildung 61: Baumdarstellung eines getaggten und geparsen Satzes aus der „Linguist’s Search Engine“.  
<http://lse.umiacs.umd.edu:8080/>

Für die weitere Verarbeitung eines derart annotierten Textes muss also bereits

ein Datenmodell zur Verfügung stehen, dass diese hierarchische Annotation fassen kann. Werden weitere hierarchische Annotationen – etwa zur Informationsstruktur – hinzugefügt, so treten bereits die in Kapitel „Überlappende Annotationen“ ab Seite 45 besprochenen Probleme mit der Modellierung kreuzender Kanten auf.

#### 4.2.2. Werkzeuge zur manuellen Annotation schriftlicher Texte

Insbesondere für Texte in Sprachen, für die keine Trainingsdaten für automatisierte Verfahren zur Verfügung stehen oder für seltener genutzte Annotationsbereiche werden Werkzeuge gewünscht, die Annotatoren bei der Arbeit unterstützen.

Dabei wird die Unterstützung um so wichtiger, je komplexer das zugrunde liegende Annotationsmodell wird. Grundsätzlich lassen sich alle Texte, die ihre Annotation inline in derselben Datei vorhalten, natürlich in jedem beliebigen Texteditor durchführen, solange dieser die benötigten Zeichensysteme unterstützt.



```
170 <s>
171   <cl type="main">
172     <w pos="u">
173       <hi rend="initial">H</hi>æ</w>
174     <w pos="xVB ff">byriæs</w>
175     <w pos="u">fagh<expan>e</expan>r</w>
176     <w pos="u">æ<expan>e</expan>ndæ</w>
177     <w pos="u">och</w>
178     <w pos="u">iær<supplied>teknæ</supplied>
179     </w>
180     <w pos="u">aff</w>
181     <w pos="u">karlla</w>
182     <w pos="u">magn<expan>us</expan>
183     </w>
184     <w pos="u">k<expan>o</expan>n<expan>un</expan>gh</w>
185     <w pos="u">och</w>
186     <w pos="u">tolff</w>
187     <w pos="u">
188       <supplied>hans</supplied>
189     </w>
190     <w pos="u">iempnu<expan>n</expan>gha</w>
191   </cl>
192 </s>
```

Abbildung 62: Ausschnitt aus einem teilweise POS-annotierten altschwedischen TEI-Text. Die Annotationen lassen sich theoretisch noch mit einem Texteditor durchführen. (H3-Korpus)

Wenn es sich um Texte mit Markup-Strukturen handelt, so bieten viele Editoren (und insbesondere ausgewiesene XML-Editoren) Hilfestellungen, um die Lesbar-

keit des eigentlichen Textes zu verbessern.



Abbildung 63: Der Oxygen-XML-Editor bietet CSS-basierte Eingabehilfen zum Beispiel für TEI. So lassen sich Tags, Attribute etc. gezielt ausblenden, um die Lesbarkeit beim Editieren zu verbessern.<sup>104</sup> (H3-Korpus)

Wenn gezielt Segmente mit bestimmten Eigenschaften in einem Text herausgesucht und mit Annotationen versehen werden sollen, so ist eine solche Hilfestellung allerdings kaum wirklich nützlich. Für solche Zwecke wurde beispielsweise am SFB 538 ein Werkzeug zur manuellen Unterstützung bei der Annotation von TEI-konformen Texten erstellt.<sup>105</sup>



Abbildung 64: Der Z2-Tagger erleichtert die Annotation TEI-konformer Texte. (H3-Korpus)

Dazu lassen sich über verschiedene Parameter (Wortlisten, XPath-Ausdrücke) Elemente im TEI-Text identifizieren, denen dann Attribute aus einer selber er-

104 [http://www.oxygenxml.com/tei\\_editor.html](http://www.oxygenxml.com/tei_editor.html)

105 Informationen zum Z2-Tagger finden sich unter <http://exmaralda.org/z2tagger>. Eine Erweiterung des Programms zur Annotation von EXMARALDA-Transkriptionen wird auf Seite 168 vorgestellt.

stellten Liste von Möglichkeiten per Mausklick oder Tastendruck zugewiesen werden können. Die entsprechenden Elemente werden dabei in ihrem (definierbaren) Kontext ohne störende Tags angezeigt und hervorgehoben.

Sobald Annotationen separat vom annotierten Text gehalten werden (Standoff-Annotation) kommt man um unterstützende Software für die Annotationsarbeit nicht herum – es ist schlicht unzumutbar, die Verweise von Annotation zu Annotiertem manuell aufzulösen, zusätzlich wäre eine manuelle Annotation in diesem Fall extrem fehleranfällig.

```

1032 <Annotation id="Ann40" type="ace_rdc-relation">
1033   <RegionRef xlink:href="#Reg84" role="ace_rdc-mention-relations" xlink:type="simple"/>
1034   <Content type="ace_rdc-class-content">
1035     <Parameter type="boolean" unit="NULL_UNIT" role="ace_rdc-is-explicit">TRUE</Parameter>
1036     <Parameter type="string" unit="NULL_UNIT" role="ace_rdc-relation-id"/>
1037   </Content>
1038 </Annotation>

```

Abbildung 65: Eine Annotation im ATLAS-interchange-Format, die auf eine Segmentierung verweist, die wiederum auf den Quelltext verweist.<sup>106</sup>

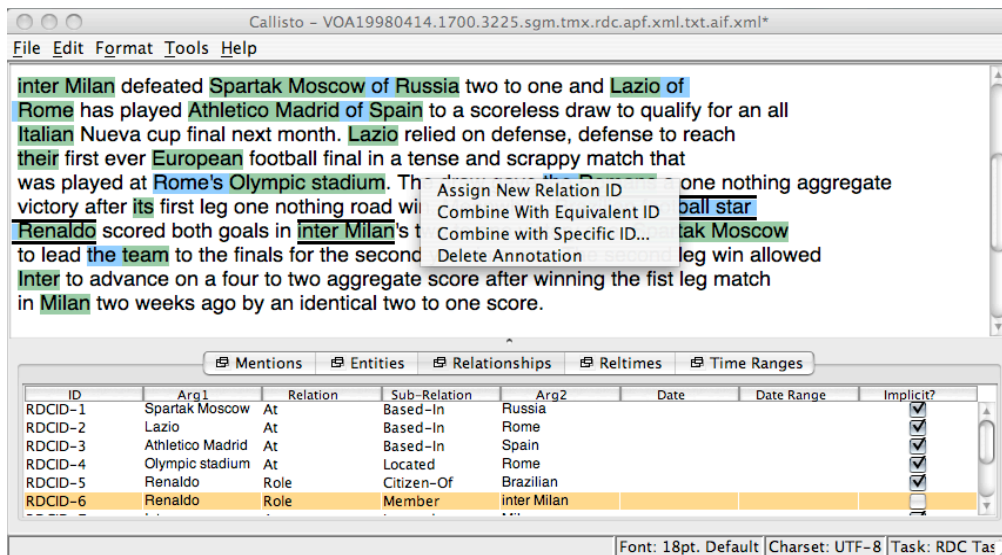


Abbildung 66: Das Annotationswerkzeug „Callisto“ löst die Standoff-Verknüpfungen des ATLAS-Formates auf und ermöglicht das Anlegen verschiedenster Annotationen, die dann in Standoff-Dateien gespeichert werden.

106 Entnommen aus dem Beispielkorpus des Callisto-Annotationswerkzeuges (<http://callisto.mitre.org/>).

### 4.2.3. Anwendbarkeit von Werkzeugen für Texte auf Transkriptionen

Einige der vorgestellten Werkzeuge können mit XML-Texten operieren. Die in Kapitel „Existierende Datenmodelle“ ab Seite 55 vorgestellten Modelle nutzen – sofern sie über ein Datenformat verfügen – ebenfalls XML zur Datenspeicherung. Will man sich diesen Umstand zunutze machen, um Transkriptionen mit bestehenden Werkzeugen für Texte zu annotieren, so stößt man auf eine Reihe von Besonderheiten:

Die Unterschiede in den zugrunde liegenden Modellen wurden ab Seite 45 bereits eingehend besprochen. Aus diesen Betrachtungen heraus ist es klar, dass ein Programm wie der Z2-Tagger sich zwar vergleichsweise einfach dahin gehend erweitern ließe, dass er Dateien des EXMARaLDA-Formates lesen kann, auf der anderen Seite aber ein deutlich größerer Aufwand betrieben werden müsste, um die Visualisierung der Transkription in das Programm zu integrieren. Zudem wäre in diesem konkreten Fall gar nicht klar, wo genau im Modell die Annotationen untergebracht werden könnten.

Es ist auch klar, dass alle Werkzeuge für schriftliche Texte nicht mit Merkmalen wie der Überlappung von Sprecherbeiträgen und der daraus entstehenden Parallelität zweier Primärtexte umgehen können. Um solche Werkzeuge anwenden zu können, müssten also in jedem Fall die Beiträge der einzelnen Sprecher getrennt behandelt werden.

Auch die automatisierten Werkzeuge (also die Tagger und Parser) ließen sich theoretisch so konfigurieren, dass sie mit den XML-Formaten der Transkriptionswerkzeuge arbeiten. Hier kommen aber die Probleme, die sich aus dem Register der gesprochenen Sprache ergeben und die in „Gesprochene vs. geschriebene Sprache“ ab Seite 13 ausführlich besprochen wurden, voll zum Tragen. Insbesondere der Umstand, dass für Transkriptionen gesprochener Sprache weder syntaktische Wohlgeformtheit noch regelhafte Orthografie vorausgesetzt werden kann, wird bei den verfügbaren Taggern zu völlig ungenügenden Fehlerraten führen.

Ein Szenario, in dem automatisierte Annotationswerkzeuge für Texte einfach auf



Transkriptionen gesprochener Sprache anwendbar sind, würde also eine Transkription voraussetzen, in der die speziellen Merkmale gesprochener Sprache nicht vorkommen oder nicht berücksichtigt wurden – diese Transkriptionen existieren in den hier betrachteten Forschungszusammenhängen im Grunde nicht.

#### 4.2.4. Transkriptionswerkzeuge mit Annotationsmöglichkeiten

##### ELAN

ELAN (Wittenburg et al. 2006) ist ein Werkzeug für die Transkription von gesprochener Sprache und Multimodalität.

Das zugrunde liegende Datenmodell orientiert sich an Annotationsgraphen und das Annotationswerkzeug bietet eine an Partituren angelehnte Darstellung interlinearen Textes mit mehreren Spuren. In mehrerer Hinsicht ähnelt es damit dem System des EXMARaLDA Partitur-Editors, auch, was die Möglichkeiten angeht, Annotationen vorzunehmen.

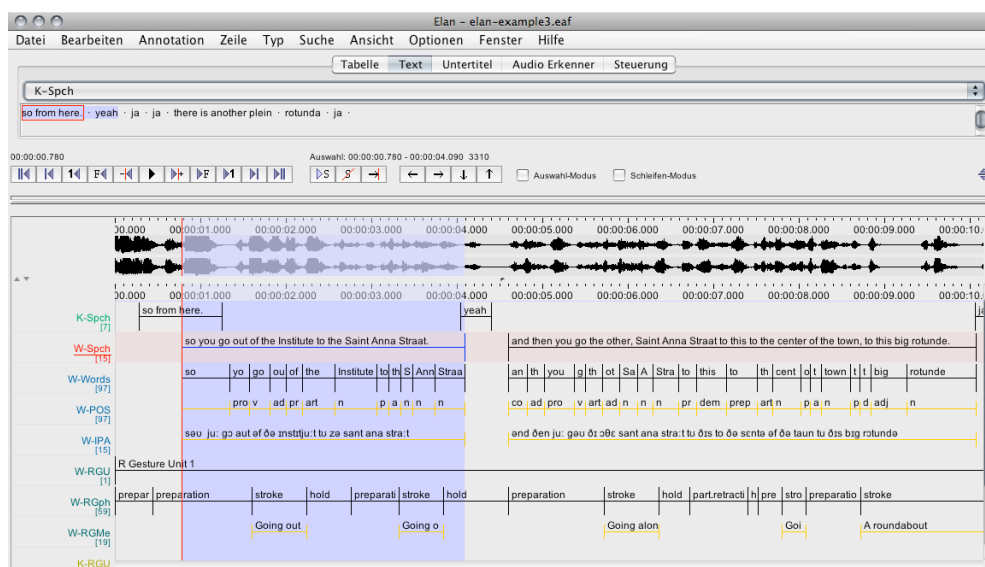


Abbildung 67: Auch ELAN zeigt die Transkription und Annotation in einer partiturähnlichen Darstellung unterhalb der Wellenform des transkribierten Audiosignals.

Hier wie dort lassen sich Annotationen in speziellen Annotationsspuren anlegen. In ELAN gibt es zwei Typen von Annotationsspuren. *Independent tiers* beziehen sich direkt auf die Timeline, können also über eine Verknüpfung durch absolute Zeitpunkte einen Abschnitt des Signals direkt annotieren. *Referring tiers* beziehen sich - analog zu den Annotationsspuren in EXMARaLDA, immer auf eine

andere Spur.

Im Unterschied zum EXMARaLDA Partitur-Editor werden diese Spuren in ELAN explizit in einer Art Hierarchie organisiert: Alle Annotationsspuren sind immer einer Transkriptionsspur oder einer anderen Annotationsspur untergeordnet. Dadurch können auch nur Annotationen an Segmenten vorgenommen werden, die tatsächlich transkribiert (bzw. annotiert) wurden. Dieser direkte Bezug führt allerdings auch dazu, dass Annotationen automatisch gelöscht werden, wenn das Segment gelöscht wird, auf das sie sich beziehen.

Für die manuelle Annotation von Transkriptionen bietet ELAN eine entscheidende Arbeitserleichterung, über die der Partitur-Editor nicht verfügt: Allen Annotationsspuren sind linguistische Kategorien zugeordnet, die (unter anderem) aus dem (bei der Beschreibung des Linguistic Annotation Framework unter „Datenmodell und Datenformat“ ab Seite 102 kurz erläuterten) Data Category Register stammen können. Innerhalb dieser Kategorien lassen sich für die Annotationsspuren kontrollierte Vokabularien erstellen und angeben.

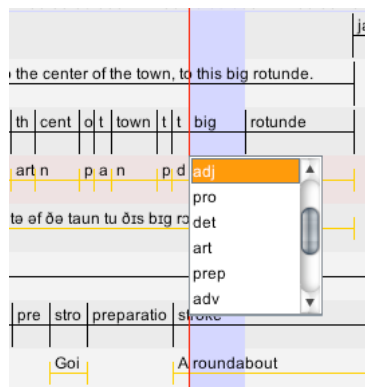


Abbildung 68: Durch die Auswahl eines „Benutzerdefinierten Wörterbuches“ in ELAN lassen sich nur vorher definierte Werte für die Annotation nutzen.

Dies führt – sofern davon Gebrauch gemacht wird – sicherlich zu deutlich weniger Fehlern bei der Konsistenz von manuell angefertigten Annotationen mehrerer Annotatoren. Gleichzeitig führt es aber wieder zu einer deutlich steileren Lernkurve, da das Konzept der linguistischen Kategorien keineswegs selbsterklärend ist und das Anlegen von Transkriptionen und neuen Spuren in diesen Transkriptionen bereits ein intensives Studium der Dokumentation voraussetzt.

Bei einer Suche in einer ELAN-Datei lassen sich diese Vokabularien natürlich auch nutzen: Soll in einer Spur gesucht werden, für die ein Vokabular angegeben wurde, so muss auch der Suchbegriff aus diesem Vokabular stammen.

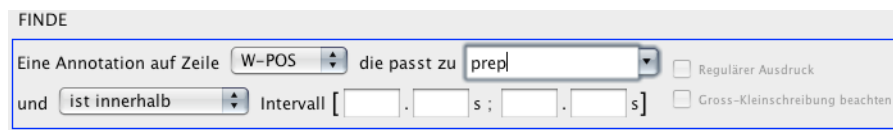


Abbildung 69: Auch bei der Suche wird in ELAN das kontrollierte Vokabularium berücksichtigt.

## Praat

Praat (Boersma und Weenink 2009) ist ein Transkriptionswerkzeug, das vor allem durch seine mächtigen Werkzeuge zur Audioanalyse hervorsteht. So lassen sich in Praat zusätzlich zur Wellenform auch Formanten, Pitchkontur, Spektrum und Intensität eines Audiosignals visualisieren.

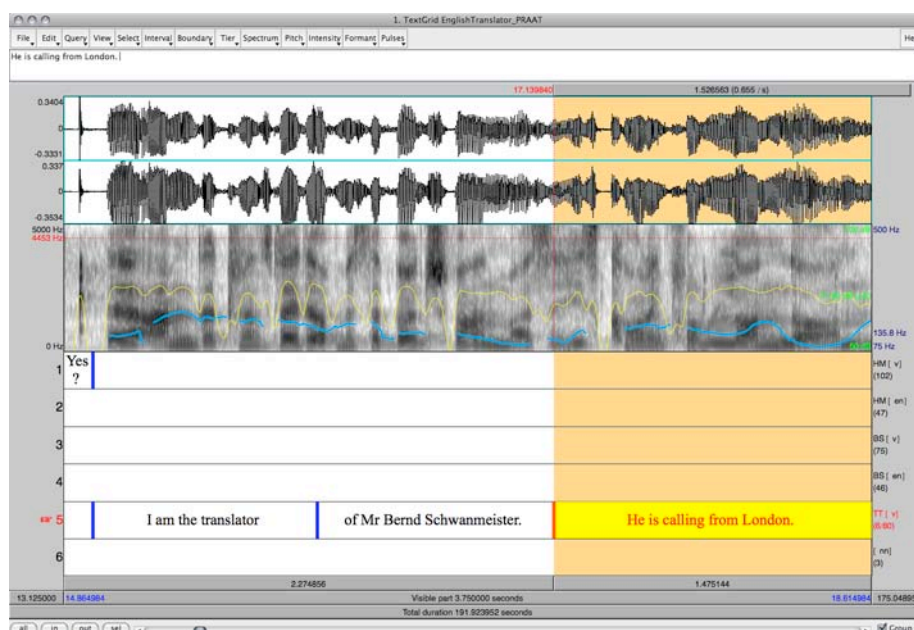


Abbildung 70: Eine Praat-Transkription („Textgrid“) mit Darstellung physikalischer Parameter der Audiodatei. (Demokorpus, English Translator)

Die Annotationsmöglichkeiten gehen nicht über das hinaus, was in dieser Arbeit bereits für EXMARaLDA und ELAN gesagt wurde: Auch in Praat lassen sich Annotationen nur über eigene, zu diesem Zweck angelegte Spuren eingeben. Im Unterschied zu den genannten Programmen haben die Spuren in Praat keinen vom Modell festgelegten Bezug zueinander. Eine Annotation, die sich auf einen

Abschnitt in einer anderen Spur bezieht, hat zwar die gleiche Ausdehnung (die Zeitpunkte sind in Praat immer absolut) und kann rein visuell auf den Abschnitt bezogen werden (sofern sie untereinander stehen), ist aber nicht mit diesem verknüpft. Weitere Unterstützung bei der Annotation (wie sie ELAN bietet) steht für Praat nicht zur Verfügung.

Die Werte, die Praat zur Visualisierung physikalischer Parameter des Audiosignals verwendet, lassen sich nicht automatisch in Annotationsspuren eintragen. Sie stehen jedoch in einer Praat-eigenen Skriptsprache zur Verfügung. Mit Hilfe solcher Praat-Skripte lassen sich – beispielsweise – zu allen Abschnitten des TextGrids automatisch die Formanten in der Mitte des jeweiligen Abschnittes ermitteln und in einer Textdatei ausgeben. Da Praat diese Informationen nicht in den TextGrids speichert, muss ein solches Skript diese Analyse „live“ durchführen, die Audiodatei also während der Laufzeit des Skriptes untersuchen.

### 4.3. Desiderata für Annotationen im EXMARaLDA-System

Die Betrachtung von Annotationsmöglichkeiten in Werkzeugen sowohl zur Annotation geschriebener Sprache als auch zur Transkription und Annotation gesprochener Sprache hat einige für diese Arbeit interessante Aspekte aufgezeigt:

- Die Möglichkeiten, Sprachdaten mit Annotationen anzureichern, hängen nur zu einem geringen Grad vom zugrunde liegenden Modell ab. Auch reine Textdateien lassen sich mit Hilfe von Stand-Off-Annotation in beliebiger Komplexität annotieren.<sup>107</sup> Modelle, die die Annotation von vorneherein mit berücksichtigen, müssen dagegen mindestens so komplex sein wie die Annotationen, die sie ermöglichen.
- Automatisierte Verfahren setzen gewisse Eigenschaften des zu annotierenden Materials voraus, die bei gesprochener Sprache häufig nicht gegeben sind.
- Für Modelle, die sich an einer Timeline orientieren, müssen für typische linguistische Annotationen zunächst geeignete Segmentierungen in für die Annotation relevanten Einheiten vorliegen oder erstellt werden.
- Die manuelle Annotation von Transkriptionen sowie von Texten kann durch Hilfsmittel innerhalb der Benutzerschnittstelle vereinfacht und vereinheitlicht werden.

Sobald die Anforderung der Segmentierung der Daten erfüllt ist, sind die Hilfsmittel bei der manuellen Annotation, die für schriftliche Texte existieren, auch für Daten gesprochener Sprache denkbar: Insbesondere kontrollierte Vokabularen für die Zuordnung von Annotationen, die Visualisierung einzelner (Sprecher-) Ebenen und bestehender Annotation zur besseren Übersicht sowie konkrete *Shortcuts* innerhalb der Benutzeroberfläche (Schaltflächen und Tastaturkürzel zur schnelleren Eingabe von definierten Werten) sind dafür wünschenswert.

---

<sup>107</sup> Die eigentliche Annotation folgt dann freilich einem gänzlich anderen Modell als das annotierte Datum – dies muss aber nicht zwingend ein Problem darstellen.

Für das EXMARaLDA-System ergeben sich folgende konkrete Anforderungen und Wünsche:

- Die EXMARaLDA-Basistranskriptionen verfügen nicht über die benötigte Segmentierungsinformation. Die meisten Segmentierungsalgorithmen, die für das System existieren, erzeugen die Segmente, die für eine Annotation linguistischer Kategorien erforderlich sind. Die Annotation dieser Segmente ist wünschenswert, im derzeitigen System aber nicht möglich.
- EXMARaLDA-Basistranskriptionen lassen sich direkt im Partitur-Editor mit Annotationen versehen, die sich auf sog. *Timeline-Events* beziehen, die mit den oben genannten Segmenten meistens nicht zu tun haben. Diese Annotationen sollten mit den Annotationen, die auf die automatisch erzeugten Segmente angewendet werden, in Beziehung gesetzt werden können.
- Es sollte eine Möglichkeit existieren, segmentierte Transkriptionen auf benutzerfreundliche Weise mit Annotationen versehen zu können.
- Es sollte eine Möglichkeit bestehen, die Annotationen auf eine intuitive Weise zu visualisieren.
- Existierende EXMARaLDA-Transkriptionen sollten nicht verändert werden müssen, um die neu zu entwickelnden Annotationsmöglichkeiten zu ermöglichen.
- Es sollten keine umfangreichen Änderungen an den bestehenden Werkzeugen erforderlich werden.

Im nächsten Kapitel wird ein System konzipiert und vorgestellt, das die Voraussetzungen für diese konkreten Anforderungen schafft.

## 5. Ein Standoff-Annotationsverfahren für Transkriptionsgraphen

In diesem Kapitel wird ein Verfahren für die Annotation von Transkriptionsgraphen entwickelt. In Kapitel 5.1. wird zunächst das gewählte Annotationsverfahren anhand einiger der im vorherigen Kapitel aufgestellten Desiderata begründet und die Problemfälle erläutert, die mit diesem Verfahren auftreten können.

In Kapitel 5.2. wird anschließend die Umsetzung dieses Verfahrens beschrieben. Dazu wird in 5.2.1. die erstellte Dokumentgrammatik erläutert. In 5.2.2. wird eine prototypische Implementierung (*Sextant*) vorgestellt, die Transkriptionen und ihre Annotationen in Beziehung setzt und ein Anwendungsfall erläutert, der durch diese Implementierung ermöglicht wird. In 5.2.3. wird die Erweiterung eines bestehenden Werkzeuges vorgestellt, mit der sich das neue Annotationsverfahren ebenfalls anwenden und somit erproben lässt.

In Kapitel 5.2.4. werden kurz einige geplante und denkbare Anwendungsszenarien vorgestellt, um die das EXMARaLDA-System erweitert werden kann, um die neu geschaffenen Annotationsmöglichkeiten benutzerfreundlich einsetzen und visualisieren zu können.

Das Kapitel schließt mit einer Zusammenfassung der wichtigsten Erkenntnisse.

## 5.1. Datenmodell und -format

### 5.1.1. Vorüberlegungen

Für die Erweiterung des EXMARaLDA-Systems um fortschrittliche Annotationsmethoden sind zwei der im vorhergehenden Kapitel aufgestellten Desiderata entscheidend:

- Bestehende EXMARaLDA-Transkriptionen sollen für die Annotation (im Bezug auf ihre Struktur) nicht verändert werden müssen – auch, um die Forderung zu erfüllen, dass keine Änderungen an bestehenden Werkzeugen erforderlich werden sollen.
- Sowohl die Segmentierung in – an der Timeline orientierte – Events als auch die Segmentierung in linguistisch relevante Einheiten (die sich aus den Transkriptionskonventionen ergeben) sollen berücksichtigt werden.

#### **Basis: Segmentierte Transkriptionen**

Die zweite Forderung legt fest, dass sich die neuen Annotationsmöglichkeiten<sup>108</sup> immer auf die segmentierten EXMARaLDA-Transkriptionen beziehen müssen. In den segmentierten Transkriptionen liegen die Segmente, die für eine linguistisch motivierte Annotation<sup>109</sup> benötigt werden, als Elemente im XML-Markup vor und sind damit für Standoff-Annotationen adressierbar. Die im Transkriptionsprozess entstandenen Segmente, die sich auf die Zeitachse beziehen (und in EXMARaLDA *Events* bezeichnen), also Ereignisse wie Sprecherwechsel, Unterbrechungen oder Nonverbales beschreiben, stehen in den segmentierten Transkriptionen ebenfalls zur Verfügung und können damit auch annotiert werden. In der Abbildung auf Seite 145 ist ein Ausschnitt aus einem segmentierten Transkriptionsgraphen zu sehen, in dem sowohl die Eventkanten als auch die durch die HIAT-Transkriptionskonventionen festgelegten Kanten zu sehen sind.

---

108 „Neu“ bedeutet in diesem Fall, dass sie über die in Kapitel 4.1. beschriebenen Annotationsmöglichkeiten hinausgehen.

109 Natürlich *müssen* die Segmente, die aufgrund von Transkriptionskonventionen und den zugehörigen Segmentierungsalgorithmen generiert werden, nicht immer linguistisch motiviert sein, sind es in der Praxis aber.





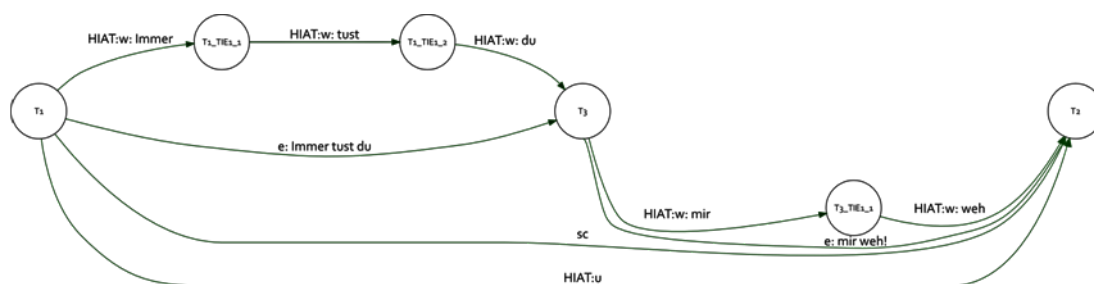


Abbildung 72: Für die Annotation relevanter Teilgraph für einen Sprecher.

### Form: Standoff-Annotation

Die erste Forderung (EXMARaLDA-Transkriptionen und -Werkzeuge sollten möglichst nicht verändert werden müssen) muss ihren Niederschlag in der Form der Annotation finden. Es wäre durchaus denkbar, zusätzliche Annotationen auf segmentierten Transkriptionen direkt in den Transkriptionsdateien anzulegen. Der EXMARaLDA-Partitur-Editor verarbeitet ohnehin keine segmentierten Transkriptionen (er schreibt diese nur) und würde sich schon deshalb nicht an ihnen stören. Auch das Suchwerkzeug würde entsprechende Dateien verarbeiten können und zusätzliche Attribute im Markup einfach ignorieren, sofern die Dokumentgrammatik diese zusätzlichen Attribute erlauben würde.

Es ist aber durchaus möglich (und in der Transkriptionspraxis üblich), dass an Transkriptionen noch Änderungen vorgenommen werden, nachdem bereits Annotationen hinzugefügt wurden. Diese Änderungen werden üblicherweise im Partitur-Editor vorgenommen, anschließend werden die segmentierten Transkriptionen wieder automatisch erzeugt. Würde man segmentierte Transkriptionen in einem derartigen Workflow mit zusätzlicher Annotation versehen, so müsste ein Mechanismus geschaffen werden, der bereits vorgenommene Annotationen in diese neuen Transkriptionsdateien integriert. Ein solcher Mechanismus wäre zwar denkbar, aber ausgesprochen schwierig handhabbar.

Um beide Forderungen erfüllen zu können, ist es daher naheliegender, Standoff-Annotation als Annotationsform zu wählen. Zwar gibt es auch in diesem Fall die Schwierigkeit, dass bei Korrekturen in einer Transkription neue segmentierte Transkriptionen entstehen, für die möglicherweise bereits Annotationen bestehen und für welche die Referenz auf irgendeine Art gesichert werden muss. Wie

sich zeigen wird, lässt sich dies aber besser mit Standoff-Annotation lösen, als es mit Inline-Annotation möglich wäre.

### **Annotationsformat: Feature Structures**

Durch die Entscheidung, Standoff-Annotation zu verwenden, ist man bei der Wahl von Modell und Format für die eigentlichen Annotationen weitestgehend frei. Es muss lediglich gewährleistet sein, dass das gewählte Format eine Möglichkeit vorsieht, mittels Zeigern auf die zu annotierenden Einheiten zu verweisen. Da die angestrebte Lösung den XML-Rahmen nicht verlassen soll, ist diese Voraussetzung mit den in XML vorgegebenen Zeigerstrukturen (insbesondere XPointer, Grosso et al. 2003) bereits erfüllt.

In den hier fokussierten Forschungszusammenhängen erreichen die gewünschten Annotationsmöglichkeiten erfahrungsgemäß keinen besonders hohen Grad an Komplexität. So wird die 1:1-Zuordnung von Attributen aus einer geschlossenen Liste sicherlich deutlich häufiger eingesetzt als beispielsweise die systematische Erstellung von Syntaxbäumen.

Komplexere Annotationen sollen aber auch nicht von vorneherein ausgeschlossen werden. Deshalb bietet es sich an, als Format für die eigentlichen Annotationen auf Feature Structures zu setzen, die in den häufigen, einfachen Fällen einfach Merkmal/Attribut-Wert-Paare sind, für komplexere Annotationen aber auch komplexe Baumgraphen abbilden können. Auch lassen sich mehrere Attribute in Feature Structures zusammenfassen, die sich auf die selbe Einheit beziehen. So sind in der folgenden Feature Structure die Informationen zu Part-of-Speech und Lemma in einer Feature Structure vereint.

```

<fs xlink:href="#Seg_24">
  <f name="pos">
    <symbol value="ADV"/>
  </f>
  <f name="lemma">
    <string value="immer"/>
  </f>
</fs>

```

Abbildung 73: POS- und Lemma-Annotation in einer Feature Structure. Der **symbol**-Tag deutet auf eine geschlossene Liste von möglichen Werten, der **string**-Tag dagegen auf eine potentiell unendlich große Menge möglicher Werte.

Zusätzlich bieten Feature Structures ein etabliertes Verfahren, um Bibliotheken mit häufig verwendeten Features unter dem Element `<fLib>` anzulegen und mittels Zeigern auf diese zu verweisen.<sup>110</sup>

### 5.1.2. Standoff-Annotation mit Feature-Structures

Das Ergebnis der Vorüberlegungen ist also, für die erweiterte Annotation von EXMARaLDA-Transkriptionsgraphen auf ein Verfahren zu setzen, bei dem Standoff-Annotation in Form von Feature Structures auf segmentierte EXMARaLDA-Transkriptionen bezogen wird.

Im folgenden XML-Ausschnitt lassen sich die Ziele für die zu erstellende Standoff-Annotation erkennen: Sowohl die Event-Segmente (identifizierbar durch das Attribut `n="e"`), die aus den Basistranskriptionen stammen und sich stärker auf die Zeitachse beziehen, als auch die mit der HIAT-Segmentierung eingeführten Segmente für *Utterance* (`n="HIAT:u"`) und *Word* (`n="HIAT:w"`) sind zu erkennen. Zusätzlich sind noch die (in der Graphenvisualisierung nicht berücksichtigten) *non timed segments* (nts) zu sehen, die Satz- und Leerzeichen (als Wortgrenzen) identifizieren.

---

110 Vgl. (ISO/TC37/SC4 2006). Diese Möglichkeit wird hier nicht näher ausgeführt.

```

<segmented-tier id="TIE1" speaker="SPK1" category="v" type="t"
display-name="Moos [v]">
  <timeline-fork start="T1" end="T3">
    <tli id="T1.TIE1.1"/>
    <tli id="T1.TIE1.2"/>
  </timeline-fork>
  <timeline-fork start="T3" end="T2">
    <tli id="T3.TIE1.1"/>
  </timeline-fork>
  <segmentation name="SpeakerContribution_Utterance_Word"
  tierref="TIE1">
    <ts n="sc" id="Seg_20" s="T1" e="T2">
      <ts n="HIAT:u" id="Seg_22" s="T1" e="T2">
        <ts n="HIAT:w" id="Seg_24" s="T1" e="T1.TIE1.1">Immer</ts>
        <nts n="HIAT:ip" id="Seg_25"><![CDATA[ ]]></nts>
        <ts n="HIAT:w" id="Seg_27" s="T1.TIE1.1"
e="T1.TIE1.2">tust</ts>
        <nts n="HIAT:ip" id="Seg_28"><![CDATA[ ]]></nts>
        <ts n="HIAT:w" id="Seg_30" s="T1.TIE1.2" e="T3">du</ts>
        <nts n="HIAT:ip" id="Seg_31"><![CDATA[ ]]></nts>
        <ts n="HIAT:w" id="Seg_33" s="T3" e="T3.TIE1.1">mir</ts>
        <nts n="HIAT:ip" id="Seg_34"><![CDATA[ ]]></nts>
        <ts n="HIAT:w" id="Seg_36" s="T3.TIE1.1" e="T2">weh</ts>
        <nts n="HIAT:ip" id="Seg_37">!</nts>
      </ts>
    </ts>
  </segmentation>
  <segmentation name="SpeakerContribution_Event" tierref="TIE1">
    <ts n="sc" id="Seg_38" s="T1" e="T2">
      <ts n="e" id="Seg_40" s="T1" e="T3">Immer tust du </ts>
      <ts n="e" id="Seg_42" s="T3" e="T2">mir weh!</ts>
    </ts>
  </segmentation>
</segmented-tier>

```

Abbildung 74: Der Ausschnitt aus der segmentierten Transkription, der dem in der vorherigen Abbildung identifizierten Teilgraphen entspricht.

Jedes dieser Segmente ist mit einem eindeutigen id-Attribut versehen, kann so als Ziel eines XLink-Zeigers (XLink 2001) und damit als Ziel einer Annotation dienen. Dabei lassen sich durchaus Anwendungsszenarien denken, bei denen auch die nts-Elemente das Ziel von Annotation sind, beispielsweise wenn die Entscheidung für ein gewähltes Satzzeichen oder das Vorhandensein einer Wortgrenze nicht eindeutig zu treffen ist.

Auch die Knoten-Elemente der (in diesem Fall einer von der Haupt-Zeitachse verzweigten) Zeitachse verfügen über eindeutige id-Attribute. Auch wenn sich spontan keine Anwendung denken lässt, die eine Annotation der Knotenelemente benötigen würde, so soll dies trotzdem nicht ausgeschlossen werden.

```

<?xml version="1.0" encoding="UTF-8"?>
<annotation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://xml.exmaralda.org/sextant.xsd"
xmlns:xlink="http://www.w3.org/1999/xlink" target="wehtun_s.xml"
targetId="CIDID09170BB3-4D95-1312-F50D-BDBAB04F294D"
id="idwehtun_s_pos_lemma1" targets="HIAT:w" type="pos_lemma"
xml:base="wehtun_s.xml">
  <description>
    <Key Name="creator">Kai Wörner</Key>
  </description>
  <ann xlink:href="#Seg_24" id="idwehtun_s_pos_ann1">
    <fs>
      <f name="pos">
        <symbol value="ADV"/>
      </f>
      <f name="lemma">
        <string>immer</string>
      </f>
    </fs>
  </ann>
  <ann xlink:href="#Seg_27" id="idwehtun_s_pos_ann2">
    <fs>
      <f name="pos">
        <symbol value="VFIN"/>
      </f>
      <f name="lemma">
        <string>tun</string>
      </f>
    </fs>
  </ann>
  (...)
</annotation>

```

Abbildung 75: POS-Annotation und Lemmatisierung des selben Abschnittes der segmentierten Transkription (Ausschnitt).<sup>111</sup>

### 5.1.3. Problemfälle

#### Diskontinuierliche Konstituenten

Bereits das verwendete – nicht sonderlich komplexe – Beispiel zeigt einen Problemfall auf, der beispielsweise im Deutschen relativ häufig zu finden sein wird:

---

<sup>111</sup> Die POS-Annotation und Lemmatisierung in den Beispielen wurde mit einer online-Version des TreeTagger (Schmid 1994) erstellt:  
<http://www.cele.nottingham.ac.uk/~ccztk/treetagger.php>.

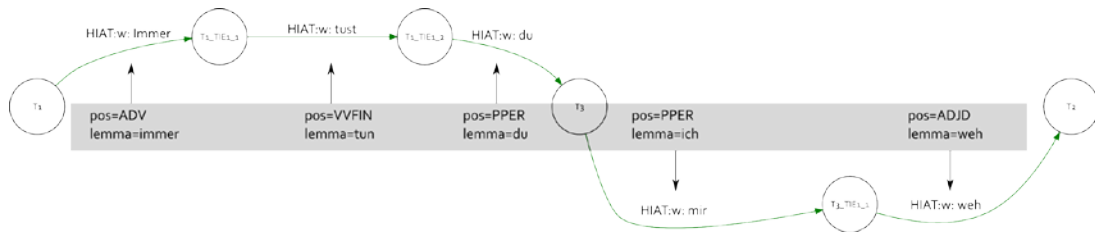


Abbildung 76: Der annotierte Abschnitt des selben Beispiels.

Der Tagger hat die Wörter „tust“ und „weh“ mit „tun“ und „weh“ lemmatisiert und „weh“ dabei als adverbiales Adjektiv gewertet, eigentlich handelt es sich aber um Konstituenten des Wortes „wehtun“. Selbst wenn der Tagger in der Lage wäre, diese Konstruktion richtig zu identifizieren, so müsste er zusätzlich die Möglichkeit haben, einen Annotationswert auf beide Konstituenten anzuwenden, also beispielsweise mit Zeigern auf beide Konstituenten zu zeigen.

In XML existieren für solche Zwecke *extended Links*, für die sich mehrere xlink:locator-Elemente anlegen lassen, die jeweils auf ein Ziel verweisen können (W3C 2001).<sup>112</sup>

```
<ann xlink:type="extended">
  <xlink:locator href="#Seg_27" label="const1">
  <xlink:locator href="#Seg_36" label="const2">
  <xlink:arc from="const1" to="const2" show="embed" actuate="auto"/>
  <xlink:arcrole="http://emeld.org/gold-ns/description.html#LexicalUnit"/>
  <fs>
    <f name="pos"><symbol value="VFIN"/></f>
    <f name="lemma"><string value="wehtun"/></f>
  </fs>
</ann>
```

Abbildung 77: Extended Link auf die Konstituenten des Lemmas „wehtun“.

Alternativ ließe sich eine weitere Annotationsebene einführen, die solche Konstituenten zu *Annotationszielen* zusammenfasst (analog zur Ebene der *markables* in PAULA). Da aber mit den Segmenten der segmentierten Transkriptionen bereits eine Token-Ebene vorliegt, würde ein solches Vorgehen zusätzlichen Auflösungsaufwand bedeuten.

112 Erweiterte (extended-) XLinks erlauben damit auch many-to-many-Beziehungen von XML-Elementen.

## Annotation von unsegmentierten Zeitspannen

Mit der gewählten Lösung lassen sich alle Segmente – sowohl die im Partitur-Editor auf der Timeline erstellten, als auch die durch den Segmentierungsalgorithmus generierten – mit Standoff-Annotation versehen.

Für nichtsprachliche Phänomene und multimodale Annotation kann es wünschenswert sein, bestimmte, absolute Zeitspannen, die in keiner bestehenden Segmentierung eine Entsprechung haben, zu annotieren, so wie es im NITE Object Model möglich ist.

Dies ließe sich durch eine Erweiterung der XLink-Syntax bewerkstelligen. So könnte etwa die Feature Structure in Abbildung 66 genutzt werden, um eine Gestenannotation für das bestehende Beispiel anzulegen.

```
<ann xlink:type="extended">
  <xlink:locator href="timeline#2420ms" label="start"/>
  <xlink:locator href="timeline#14440ms" label="end"/>
  <xlink:arc from="start" to="end" show="embed" actuate="auto"/>
  <fs>
    <f name="eyebrows"><symbol value="raised"/></f>
    <f name="speaker"><symbol value="spk1"/></f>
  </fs>
</ann>
```

Abbildung 78: Annotation direkt auf der Timeline.

Dabei ist die explizite Angabe des Sprechers, dem diese Annotation zuzuweisen ist, hervorzuheben: Da diese Annotation nicht im Transkriptionsgraphen zu verorten ist, lässt sich die Zuordnung zu einem bestimmten Sprecher auch nicht automatisch auflösen.

Diese Möglichkeit soll aber vorerst für die hier angestrebte Lösung verworfen werden. Erstens lässt sich die Auflösung der Zeitpunkte nicht mehr mit den Mitteln von Standard-XML-Werkzeugen erledigen<sup>113</sup>, zum anderen existiert im Partitur-Editor inzwischen die Möglichkeit, Segmente auf der Timeline mit hinreichender Genauigkeit zu markieren und die Annotation dort in dafür vorgesehe-

---

113 „The framework is intended to be used as a basis for fragment identifiers for any resource whose Internet media type is one of text/xml, application/xml, text/xml-external-parsed-entity, or application/xml-external-parsed-entity.“ (Grosso et al. 2003)



nen Annotationsspuren anzulegen.<sup>114</sup>

## Annotation von unsegmentierten Abschnitten

Es kann auch wünschenswert sein, Abschnitte im transkribierten Text zu annotieren, die nicht durch ein bestehendes Segment begrenzt werden, beispielsweise, um nur für einen Teil eines Wortes eine auffällige Betonung festzuhalten.

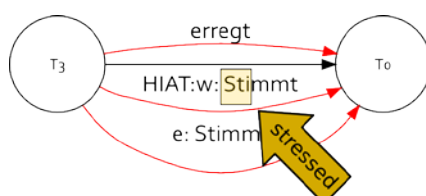


Abbildung 79: Annotation eines Abschnittes, der keinem existierenden Segment entspricht.

Dieser Fall ist durch die Verwendung von XPointer (Grosso et al. 2003) bereits abgedeckt. So lassen sich in XPointer auch *points* und *spans* adressieren, also Punkte und Abschnitte mittels Character-Offsets als Ziele von Verweisen festlegen:

```
xlink:href="#xpointer(string-range(//*[@id='Seg_14'],',',1,3))"
```

Abbildung 80: XLink mit Character-Offset.

Dieser Zeiger verweist auf den drei Zeichen langen Abschnitt ab dem ersten Zeichen im Element mit der id `Seg_14` (im Falle des Beispiels ist dies das `HIAT:w-` Segment mit dem Wort „Stimmt“). Da XPointer immer mit XPath-Ausdrücken arbeiten, lassen sich mit ihrer Hilfe auch noch komplexere Verweise realisieren.<sup>115</sup>

## Referenzielle Integrität

Ein Problem, das sich bei jeder Form von Standoff-Annotation stellt, ist es, die Integrität der Zeiger von Annotation auf Annotiertes sicherzustellen. In einem

---

114 Seit 2009 kann im Partitur-Editor mit einer verbesserten Timeline und einer grafischen Darstellung der Wellenform des zugrunde liegenden Audiosignals gearbeitet werden.

115 Die Verwendung von XPointer ist nicht völlig unproblematisch: Die Abschnitte lassen sich zwar eindeutig adressieren, die Unterstützung von XPointer in XML-Werkzeugen ist aber noch lückenhaft, so dass die Auflösung in den meisten Fällen mit eigenen Softwarelösungen geschehen muss. Vgl. auch:  
<http://standards.jisc.ac.uk/catalogue/XPointer.phtml>.

Workflow, bei dem zunächst die Transkription abgeschlossen wird, bevor man zur Annotation übergeht, würde sich kein derartiges Problem ergeben. Gerade bei Transkriptionen, und gerade in den betrachteten Forschungsfeldern, ist ein derartiger Workflow aber eher die Ausnahme als die Regel (siehe auch „Prozesshaftigkeit von Transkription“ ab Seite 39).

Bei Änderungen in einer Transkription müsste also für das vorgeschlagene System sichergestellt werden, dass die im XML-Markup vergebenen Ids sich nicht ändern bzw. nicht verschwinden.

Dies lässt sich nie zu 100% sicherstellen. Zwar lässt sich gewährleisten, dass Ids, die einmal vergeben wurden, nicht noch einmal an anderer Stelle vergeben werden und dass sich die Ids gleichbleibender Segmente nicht ändern, es lässt sich aber beispielsweise nicht erreichen, dass Ids von gelöschten Segmenten erhalten bleiben.<sup>116</sup> Dies ist aber auch nicht unbedingt ein Problem: In dem Fall, dass ein Segment nicht mehr existiert, ist die zugehörige Annotation ohnehin hinfällig.

Die Trennung von Basis- und segmentierter Transkription im EXMARaLDA-System verschärft dieses Problem aber noch zusätzlich: Segmentierte Transkriptionen werden vom Partitur-Editor nur *geschrieben*. Dazu wird mit einer endlichen Maschine die Segmentierungsinformation, die sich aus den Transkriptionskonventionen ergibt, angewendet.

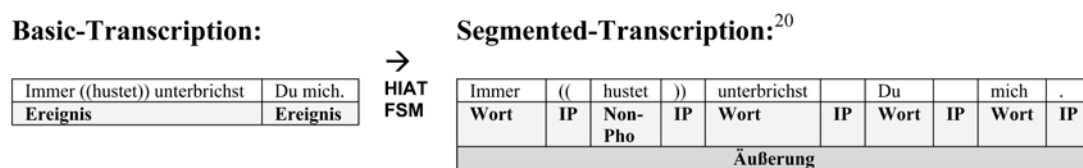


Abbildung 81: Segmentierung mittels einer endlichen Maschine. (Schmidt 2004: 328)

In diesem Schritt bekommen die neuen, aus den Konventionen motivierten Segmente ihre Ids zugewiesen. Selbst wenn also für die Basistranskriptionen gewährleistet wäre, dass sich einmal vergebene Ids für sich nicht verändernde Abschnit-

<sup>116</sup> *Technisch* lässt sich dies zwar bewerkstelligen, in dem man beispielsweise eine Liste von Ids führt, die zu keinem Segment gehören – damit würde man aber nur die Information explizieren, die man ohnehin schon hätte: dass diese Segmente nicht mehr existieren.

te nicht mehr ändern, so würden doch bei der Segmentierung (und die dabei entstehenden Segmente sind es ja vor allem, die für die zusätzliche Annotation interessant sind) neue Ids vergeben werden müssen, da der Partitur-Editor und die endliche Maschine von der vorherigen segmentierten (und möglicherweise annotierten) Transkription nichts wissen können.

Dieses Problem lässt sich ohne Modifikationen im Partitur-Editor nicht lösen. Insbesondere für das Problem der immer wieder neu vorgenommenen Segmentierung muss dabei das ganze Konzept der Trennung von Basis- und segmentierter Transkription in den EXMARaLDA-Werkzeugen überdacht werden.

Einige Schritte in dieser Richtung wurden bereits unternommen: Während man in älteren Versionen des Partitur-Editors während des gesamten Transkriptionsprozesses keine Entscheidung über die verwendeten Transkriptionskonventionen (und damit insbesondere über den verwendeten Segmentierungsalgorithmus) treffen musste, so steht diese Entscheidung in den neueren Versionen von Anfang an fest.

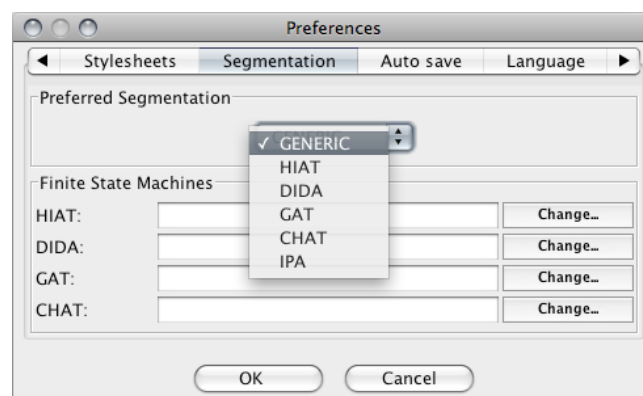


Abbildung 82: Auswahl des Segmentierungsalgorithmus im Partitur-Editor seit Version 1.4.

Dadurch ergeben sich einige Vorteile: Zum Beispiel muss der Benutzer, der sich in den meisten Fällen bereits für eine Transkriptionskonvention entschieden hat, diese Entscheidung nur einmal im Programm festlegen und kann dann bereits während der Transkription überprüfen, ob er Fehler im Bezug auf die Transkriptionskonventionen gemacht hat und muss sich nicht mehr um weitere Einstellungen bezüglich der Segmentierung kümmern.

In diesem Zuge wurde auch ein *generischer* Segmentierungsalgorithmus einge-

führt, der absichtlich so gut wie keine Vorgaben bezüglich der Transkription macht, aber trotzdem in der Lage ist, aus den meisten Transkriptionen eine segmentierte Transkription zu erstellen.

Da alle Information einer Basistranskription auch in einer segmentierten Transkription enthalten ist, wäre es wünschenswert, dass der Partitur-Editor mittelfristig nur noch ein Format (das der segmentierten Transkriptionen) verwenden würde. Dann (aber nicht vorher) könnte auch über einen Mechanismus nachgedacht werden, der die Integrität der Ids sicherstellt, die durch die Segmentierung entstehen.

Bis dahin wäre es bereits hilfreich, dies für die Ids der Basistranskriptionen zu ermöglichen. Die Id-Bezeichner der *Knoten*, die bei der Segmentierung entstehen, beziehen sich bereits auf die Bezeichner der Knoten-Ids der Basistranskription (sie beginnen immer mit der Id des Knotens, von dem sie ausgehen). Es wäre daher möglich, dass sich die Id-Namen der entsprechenden Kanten aus diesen Knoten-Ids generieren (statt einfach – wie jetzt – jedes Mal neu fortlaufend durchnummeriert zu werden). Auf diese Weise würden sich bei einer erneuten Segmentierung nur die Segment-Ids ändern, die innerhalb eines geänderten Event-Segments liegen.

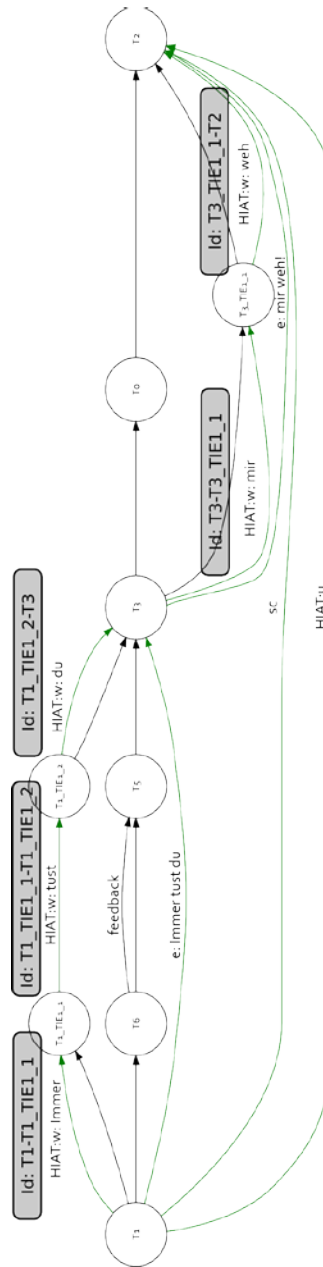


Abbildung 83: Mit einem modifizierten Verfahren zum Erzeugen von Ids für die Kanten der segmentierten Transkriptionen blieben die Ids von Kanten, die nicht in ein geändertes Event fallen, einheitlich.

Schon mit dieser – relativ einfach zu implementierenden – Änderung ließe sich also gewährleisten, dass die Zuordnung von Annotationen zu Segmenten einer segmentierten Transkription für alle Events möglich ist, an denen keine Änderungen vorgenommen werden. Für die meisten Anwendungsfälle wird dies vollkommen ausreichen, da die Annotationen für einen Abschnitt, der in der Basistranskription geändert wurde, in den meisten Fällen ohnehin überprüft werden müsste.

## 5.2. Umsetzung

### 5.2.1. XML-Schema

Die Standoff-Annotationsdateien des neuen Annotationsverfahrens sind mit einer Dokumentgrammatik auf XML-Schema-Basis spezifiziert. Dieses Schema ist gleichzeitig sehr einfach und sehr flexibel angelegt: es definiert im Grunde lediglich einen Container (*annotation*), in dem eine beliebig lange Liste von tatsächlichen Annotationen (*ann*) liegt, die mit Feature Structures beschrieben sind. Die *ann*-Elemente dienen dabei nur dazu, den Verweis auf das zu annotierende Datum aufzunehmen, da ISO-Feature Structures keine Zeiger enthalten dürfen.

Zusätzlich kann der Container noch ein Element *description* enthalten. Dieses Element ist dem Schema für EXMARaLDA-Metadaten-Dateien, die mit dem Corpus-Manager verwaltet werden, entnommen. Es enthält eine beliebig lange Liste von Attribut-Wert-Paaren (als *key*-Elemente), in denen eine Beschreibung der Annotationsdatei erfolgen sollte. Diese Beschreibung könnte Informationen bezüglich der Art der Annotation, des zugrunde liegenden Tagsets, der Bearbeiter, der Version etc. beinhalten.

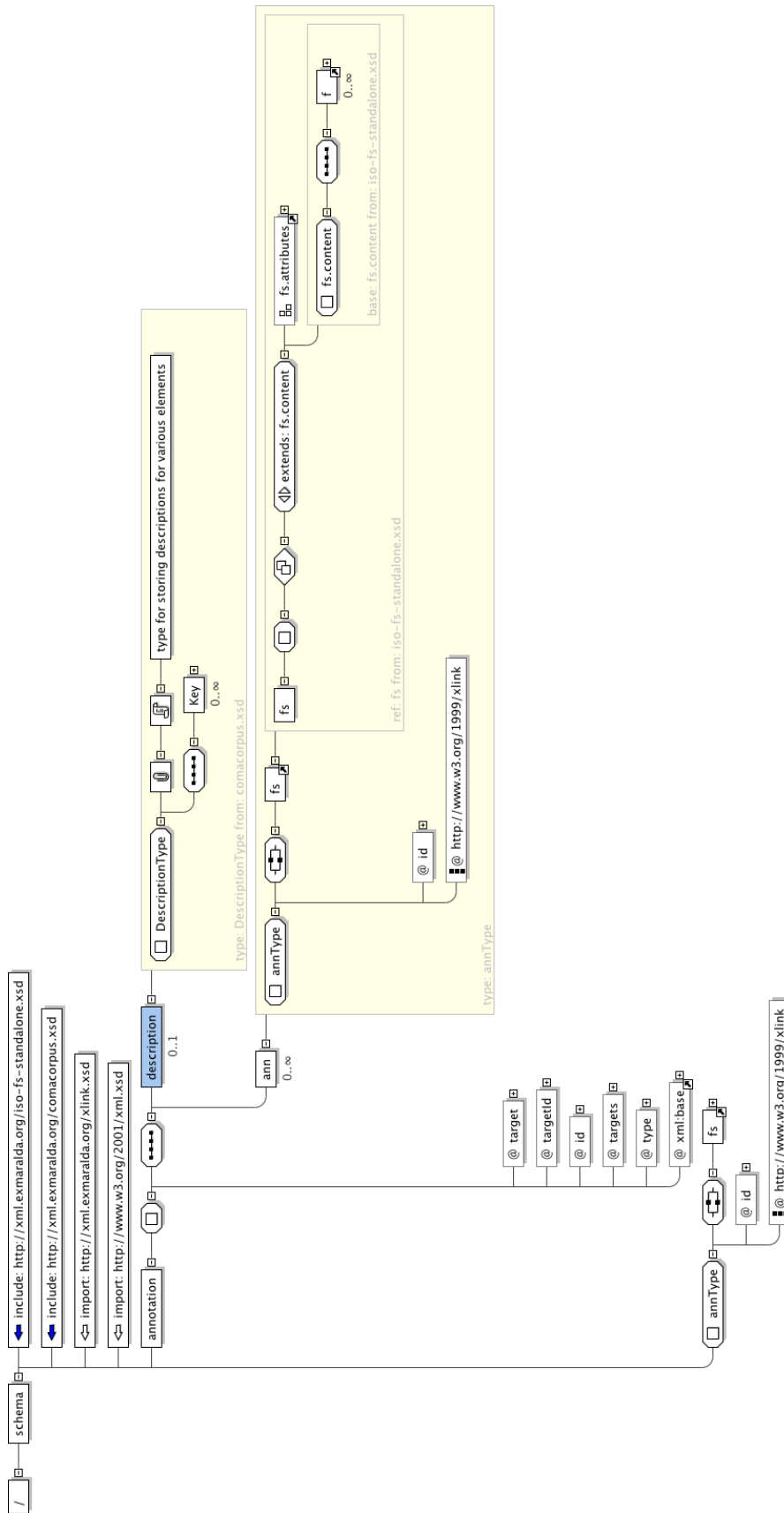


Abbildung 84: XML-Schema für Sextant-Annotationen.

Das Container-Element `annotation` sieht zur Beschreibung der Datei zusätzlich eine Reihe von obligatorischen Attributen vor. Diese sind im Einzelnen:

`target` (URI) gibt das Dokument an, auf das sich die Annotation bezieht.

`targetId` (string) EXMARaLDA-Transkriptionen enthalten eine vom Copus-Manager zugewiesene, eindeutige Id. Auf diese wird zusätzlich verwiesen, um sicherzustellen, dass es sich um die richtige Transkription handelt.

`id` (ID) Annotationsdateien haben selber eine Id, mit der sie eindeutig identifiziert werden können.

`targets` hier kann festgelegt werden, auf welche Segmenttypen sich die Annotationen beziehen. Sind beispielsweise nur Annotationen auf Wortebene vorhanden, so kann in diesem Attribut der Wert `HIAT:w` festgehalten werden.

`xml:base` dupliziert das Attribut `target` und ermöglicht eine kompaktere Schreibung der XLink-Zeiger.

Das Schema ist die Grundlage, auf der Sextant EXMARaLDA Standoff-Annotationen validiert.

### **5.2.2. Standoff EXMARaLDA-Transcription Annotation Tool (Sextant)**

Das Sextant-Werkzeug ist eine *Proof of Concept*-Implementation der Standoff-Annotationen für EXMARaLDA-Transkriptionen. Der Haupt-Einsatzzweck des Werkzeuges besteht darin, die Einsatzfähigkeit des neuen Annotationsverfahrens zu überprüfen und dazu Verbindungen zwischen segmentierten EXMARaLDA-Transkriptionen und den darauf Bezug nehmenden Standoff-Annotationen herzustellen und zu verifizieren.

Es ist explizit kein Werkzeug, um Annotationen an Transkriptionen vorzunehmen – dazu müssten zu viele Komponenten implementiert werden, die es an anderer Stelle in den meisten Fällen bereits gibt (wie beispielsweise in dem Werkzeug *Z2-Tagger*, das im folgenden Unterkapitel beschrieben ist). Sextant soll



vielmehr dazu dienen, solche Annotations- oder Visualisierungswerkzeuge, die ursprünglich für andere Datentypen konzipiert wurden, für EXMARaLDA-Transkriptionen einsetzbar zu machen.

In der jetzt existierenden Version beschränkt sich die Funktionalität des Programmes darauf,

- existierende Standoff-Annotationen zu validieren (also zu überprüfen, ob es in sich gültige Annotationsdateien sind),
- die Integrität der Verweise auf die annotierten Transkriptionen sicherzustellen (also zu überprüfen, ob für alle Verweise Ziele existieren),
- aus EXAKT-Konkordanzen mit manuell hinzugefügter Annotation (siehe „Annotation von Konkordanzen“ ab Seite 123) Standoff-Annotationsdateien zu erzeugen,
- eine einfache Visualisierung der Transkriptionen mit einer Annotationsdatei vorzunehmen.

Diese Funktionen werden am Ende dieses Unterkapitels ab Seite 162 kurz erläutert.

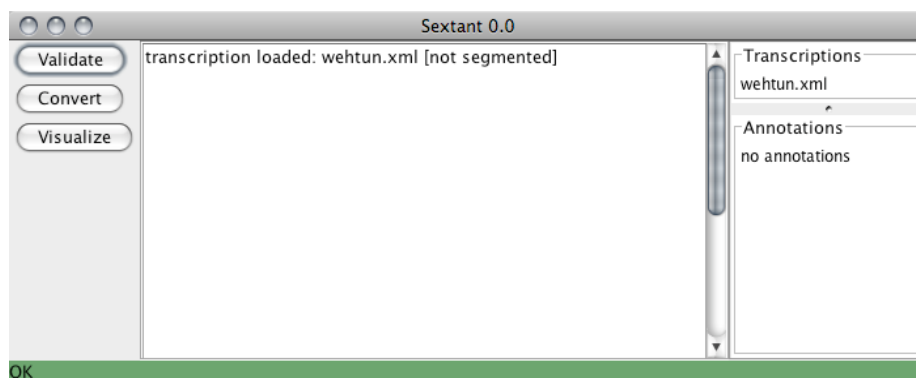


Abbildung 85: Die Sextant-Benutzeroberfläche (Screenshot vom 01.02.2009).

Sextant dient als Prototyp dazu, die Tauglichkeit des in dieser Arbeit entwickelten Annotationsverfahrens zu belegen (*proof-of-concept*), also zu zeigen, dass sich auf der Basis des Verfahrens funktionsfähige Software entwickeln lässt. Wie ab Seite 171 ausgeführt, wäre die hier implementierte Funktionalität in anderen Werkzeugen unter Umständen besser aufgehoben als in einer speziellen „Anno-

tationsverwaltung“.

Der Prototyp zeigt die Möglichkeiten, auf deren Basis sich andere Werkzeuge für das Verfahren entwickeln ließen.

Unter <http://exmaralda.org/sextant/> existiert eine Webseite zum Programm, auf der sich neben der aktuellsten Version auch Anmerkungen zur jeweils aktuellen Funktionalität des Programms sowie Bedienungshinweise und etwaige Änderungen finden.

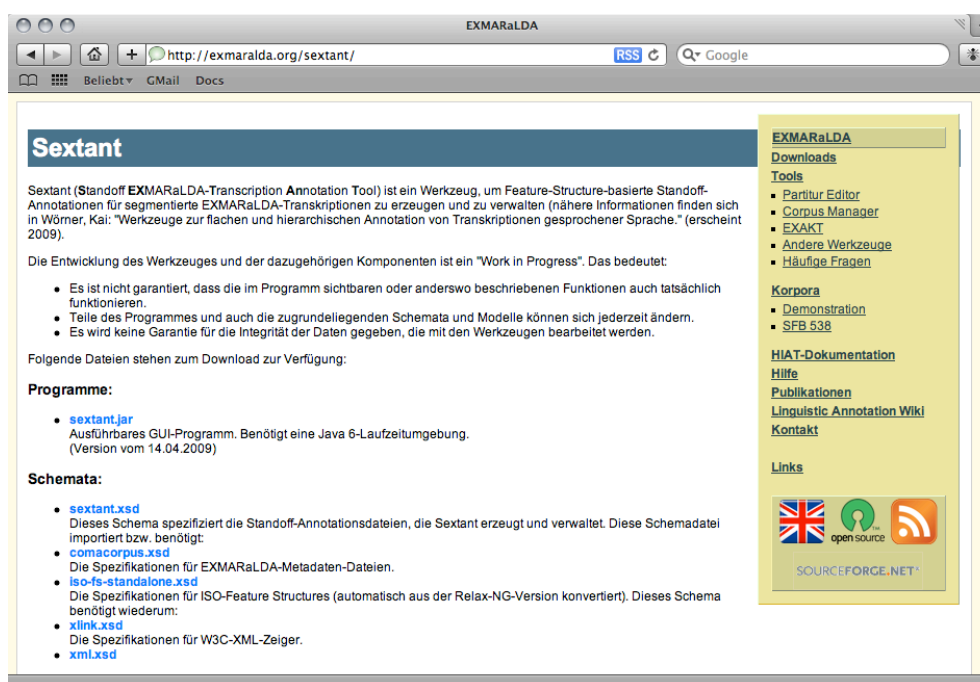


Abbildung 86: Neue Versionen der Software und der zugrunde liegenden Komponenten werden auf der Sextant-Webseite (<http://exmaralda.org/sextant/>) veröffentlicht.

Sextant bietet zweierlei Funktionen, die der Validierung von Standoff-annotierten EXMARaLDA-Transkriptionen dienen:<sup>117</sup>

### Validierung der Standoff-Annotationsdateien

Dadurch, dass die Struktur von EXMARaLDA-Standoff-Annotationsdateien in der bereits erwähnten XML-Schema-Dokumentengrammatik festgelegt ist, kann die Überprüfung der Wohlgeformtheit und Gültigkeit einfach mit einem validierenden XML-Parser durchgeführt werden. Sextant überprüft an dieser Stelle zusätz-

---

117 Die folgende Beschreibung der Sextant-Funktionen bezieht sich auf die Version des Programms von 01.03.2009.

lich, ob die referenzierte Datei am angegebenen Ort existiert.

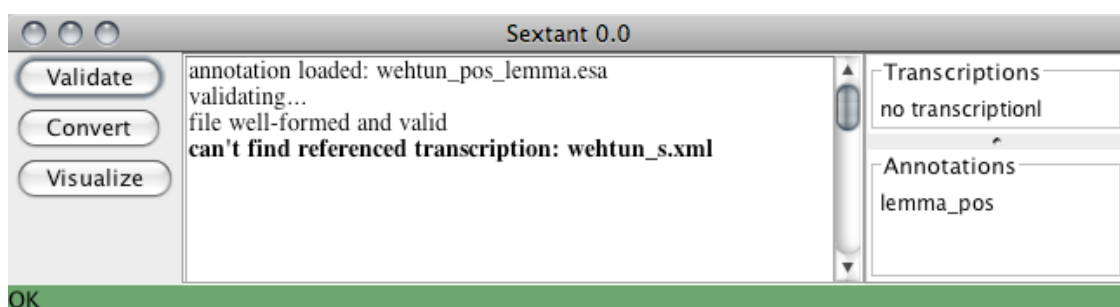


Abbildung 87: Validierung einer Standoff-Annotationsdatei.

### Validierung der Zeiger

Bei der Validierung der Zeiger überprüft Sextant, ob die Elemente, auf die mit XLink-Zeigern in den Annotationsdateien verwiesen wird, in den Transkriptionsdateien auch tatsächlich existieren.

Diese Aufgabe lässt sich grundsätzlich auch mit XML-Mitteln lösen, die Auflösung von XLink-Verweisen ist aber in längst nicht allen XML-Werkzeugen tatsächlich implementiert. Sextant validiert daher die XLink-Verweise selber, indem zunächst die referenzierte Datei geöffnet und dann der XPath aufgelöst wird.

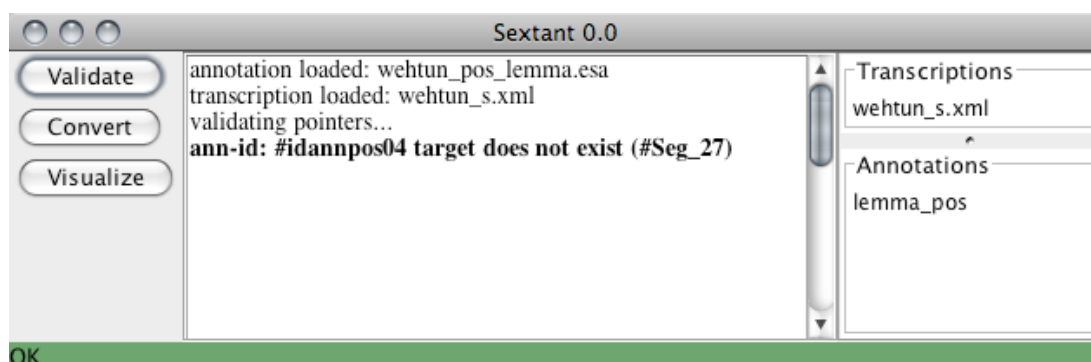


Abbildung 88: Fehler bei der Validierung der Zeiger auf die Transkription.

### Generierung von Standoff-Annotationsdateien aus EXAKT-Konkordanzen

Die Möglichkeit, in den Konkordanzen, die das Suchwerkzeug EXAKT ausgibt, manuelle Annotationen hinzuzufügen, wurde bereits im Kapitel „Annotation von Konkordanzen“ ab Seite 123 beschrieben. Dort wurde auch gesagt, dass die dabei

entstehenden Annotationsinformationen nur in den Konkordanz-XML-Dateien, die von EXAKT geschrieben werden, gespeichert werden und auch nur für dieses Programm zur Verfügung stehen. Da es sich bei dieser Art der manuellen Annotation um ein besonders intuitives Verfahren handelt, ist es wünschenswert, die erstellten Annotationen auch für andere Zwecke zur Verfügung zu haben, also als Annotation der eigentlichen Transkription.

Sextant bietet zu diesem Zweck eine Funktion, die aus den EXAKT-Konkordanzen und den durchsuchten Transkriptionen automatisch Standoff-Annotationsdateien erzeugt, die dann als Information zur Transkription auch unabhängig von der Konkordanz benutzt werden können.

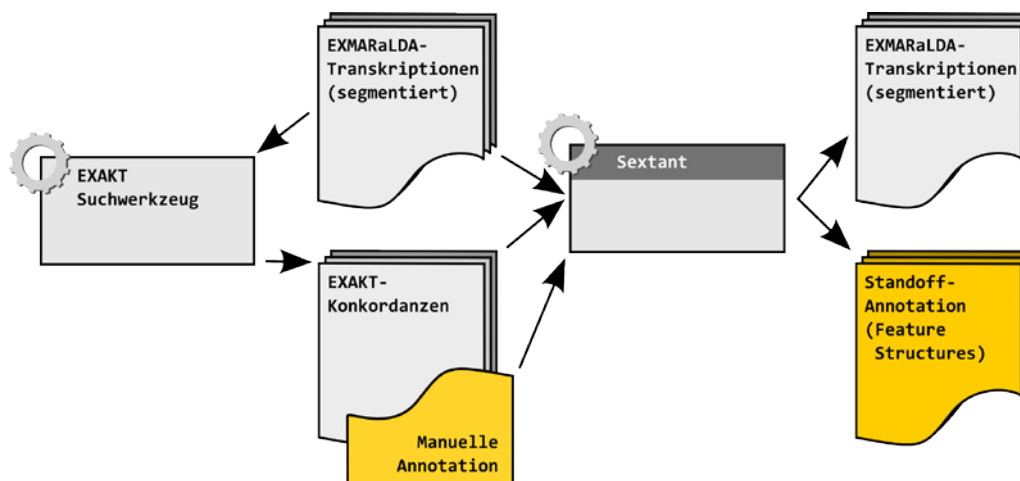


Abbildung 89: Annotationen, die in EXAKT-Konkordanzen manuell hinzugefügt wurden, werden von Sextant in EXMARaLDA-Standoff-Annotation überführt.

Würde man die POS-Annotation, die auch bisher zur Illustration des Beispiels verwendet wurde, manuell in ein EXAKT-Suchergebnis eintragen, so würde dies am Bildschirm etwa so aussehen:

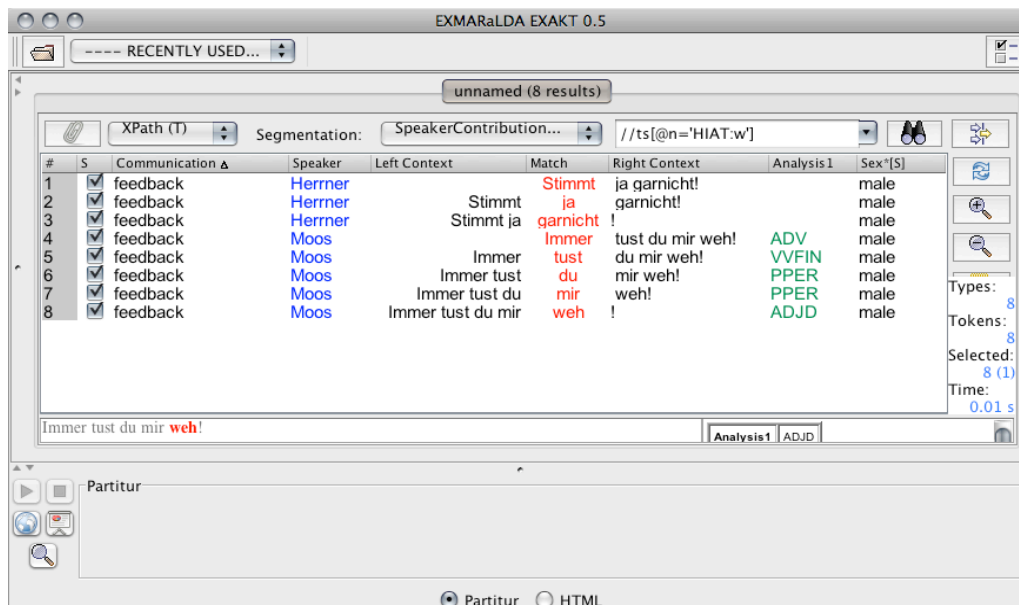


Abbildung 90: Manuelle POS-Annotation des Beispiels im EXAKT-Suchwerkzeug.

Diese Möglichkeit ist bereits ein echter Fortschritt, da diese Art der Annotation zuvor gar nicht möglich war. Selbst wenn sich diese Segmente auf andere Art und Weise annotieren ließen, so ist dieses Verfahren für selteneren Sprachen ohne automatische Annotationsmöglichkeiten und Szenarien, in denen eigene Tagsets für spezielle Annotationsebenen benötigt werden, eine immense Arbeitserleichterung, zumal EXAKT noch einen *Browsing Mode* anbietet, die die Suchergebnisse hintereinander für die Annotation einblendet, sodass sich der Annotationsvorgang noch beschleunigen lässt.<sup>118</sup>

In der abgespeicherten Konkordanz finden sich die Annotationen im XML-Markup als einfaches `data`-Element wieder, das sich über einen Umweg der entsprechenden Analysespalte zuordnen lässt.

<sup>118</sup> Vgl. Smith, Hoffmann und Rayson (2008).

```

<search-result selected="true" communication="feedback" speaker="Moos">
  <locator file="wehtun_s.exs"
    xpath="//segmentation[@name='SpeakerContribution_Utterance_Word']/
    ts)[2]"/>
  <left-context>Immer tust </left-context>
  <match original-match-start="11">du</match>
  <right-context> mir weh!</right-context>
  <data>TIE1</data>
  <data>SPK1</data>
  <data>T1</data>
</search-result>

```

Abbildung 91: Eine Zeile der Konkordanz im XML-Markup.

Die Position des Suchergebnisses (und damit das Ziel für den Zeiger der Standort-Annotation) ist an zwei Stellen codiert: Im Element locator sind die Datei und das Segment (mit Hilfe eines XPath-Ausdruckes), in der das Ergebnis gefunden wurde, vermerkt. Das match-Element enthält den gefundenen Text und im Attribut original-match-start den Character-Offset, an dem die Fundstelle im gefundenen Segment beginnt.

Dabei sind zwei Dinge bemerkenswert: Die Suche auf den Segmenten, die durch die HIAT-Segmentierung entstanden sind, lässt sich nur benutzen, wenn in EXAKT die XPath-Suche statt der auf Volltext operierenden Suche eingestellt ist. Im Screenshot ist zu erkennen, dass nach dem XPath

```
//ts[@n='HIAT:w']
```

gesucht wurde, der jedes nach HIAT segmentierte Wort-Segment als Ergebnis liefern sollte. Mit der voreingestellten, ohne XML-Kenntnisse verwendbaren einfachen Suche mittels regulärer Ausdrücke würde EXAKT die Ergebnisse dagegen in den Event-Segmenten zurückliefern.

Mit der XPath-Suche ist das Ergebnis nicht unbedingt das, was man erwarten würde: Das zurückgegebene Segment ist nicht etwa das, was tatsächlich gesucht wurde, sondern das erste timed segment (ts), das in dieser Segmentierung gefunden wird.

```

01 <segmentation name="SpeakerContribution_Utterance_Word"
    tierref="TIE1">
02   <ts n="sc" id="Seg_20" s="T1" e="T2">
03     <ts n="HIAT:u" id="Seg_22" s="T1" e="T2">
04       <ts n="HIAT:w" id="Seg_24" s="T1" e="T1.TIE1.1">Immer</ts>
05       <nts n="HIAT:ip" id="Seg_25"><![CDATA[ ]]></nts>
06       <ts n="HIAT:w" id="Seg_27" s="T1.TIE1.1"
          e="T1.TIE1.2">tust</ts>
07       <nts n="HIAT:ip" id="Seg_28"><![CDATA[ ]]></nts>
08       <ts n="HIAT:w" id="Seg_30" s="T1.TIE1.2" e="T3">du</ts>
09       <nts n="HIAT:ip" id="Seg_31"><![CDATA[ ]]></nts>
10       <ts n="HIAT:w" id="Seg_33" s="T3" e="T3.TIE1.1">mir</ts>
11       <nts n="HIAT:ip" id="Seg_34"><![CDATA[ ]]></nts>
12       <ts n="HIAT:w" id="Seg_36" s="T3.TIE1.1" e="T2">weh</ts>
13       <nts n="HIAT:ip" id="Seg_37"><![CDATA[ ]]></nts>
14     </ts>
15   </ts>
16 </segmentation>

```

Abbildung 92: Ergebnis-Segment der XPath-Suche.

## Der XPath

```
//segmentation[@name='SpeakerContribution_Utterance_Word']/ts[2]
```

verweist auf das Segment in Zeile 02, nicht, wie man erwarten würde, auf das Segment in Zeile 08, nach dem tatsächlich gesucht wurde. Diese Information findet sich dagegen im Element

```
<match original-match-start="11">du</match>
```

als Character-Offset codiert.

Für die Überführung in Standoff-Annotation ergibt sich dadurch ein Problem: Zwar ließe sich einfach die Lokalisierungsinformation übernehmen, wie sie in EXAKT vorliegt (der entsprechende Pointer in der Annotationsdatei müsste dann entsprechend

```
xlink:href="#xpointer(string-range(//*[@id='Seg_20'],',',11,2))"
```

lauten), dies ist aber nicht die Information, die man haben wollte.

Die Ergebnisse einer XPath-Suche und einer Suche mit regulären Ausdrücken sind allerdings - bezogen auf das nächsthöhere gemeinsame Segment - primärdatenidentisch. Solche Suchen ließen sich also über ihren Character-Offset miteinander in Beziehung setzen, ebenso ließe sich über diese Tatsache das tatsächlich

gewünschte Segment ermitteln. Da EXAKT die eigentliche Suchanfrage nicht mit der Konkordanz speichert, liegt die Information über den tatsächlich gesuchten XPath nach Beendigung des Programms nicht mehr vor und kann von Sextant-Werkzeugen nicht mehr verwendet werden. Für Sextant bleibt dann nur übrig, die Annotation entweder ebenfalls auf ein Character-Offset zu beziehen, was oftmals unerwünscht sein dürfte, oder vom Benutzer das gewünschte Segment bei der Konvertierung in Standoff-Annotation abzufragen und die betreffenden Segmente über diesen Character-Offset zu ermitteln.

Da EXAKT immer auf ganzen Korpora (also auf Zusammenstellungen von Transkriptionen) sucht, sind in den EXAKT-Konkordanzen meistens mehrere Transkriptionen vermerkt, in denen Fundstellen vorliegen. Da sich die Standoff-Annotationsdateien immer nur auf eine EXMARaLDA-Transkription beziehen, legt Sextant für jede Transkription und noch einmal für jede Analysespalte (also Annotationsebene) eine eigene Standoff-Datei an.

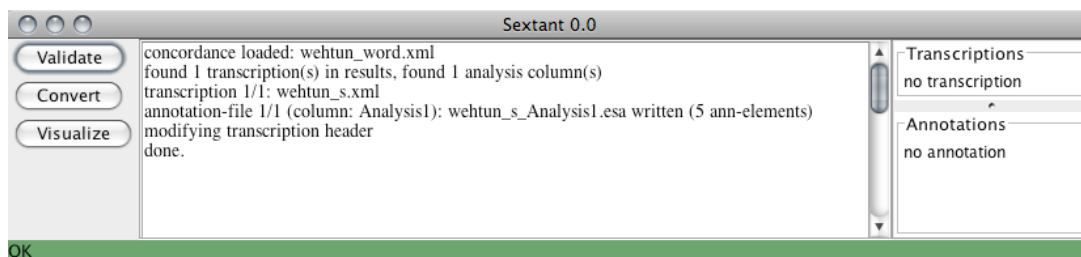


Abbildung 93: Sextant generiert Standoff-Annotationsdateien aus EXAKT-Konkordanzen.

## Einfache Visualisierung der Annotationen

Um eine tatsächliche Kontrolle darüber zu haben, dass die Annotationen nicht nur auf existierende, sondern tatsächlich auf die *richtigen* Segmente in der annotierten Transkription zeigen, kann Sextant eine einfache Visualisierung der Transkription mit ihrer dazugehörigen Annotation vornehmen.



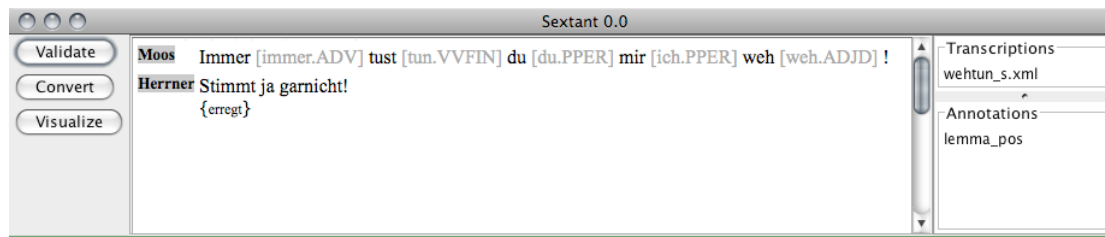


Abbildung 94: Einfache Visualisierung einer Transkription mit ihrer Standoff-Annotation.

Dabei werden nur jeweils die annotierten Segmente anhand ihres *höchsten* Elementenelements *unterhalb* des *segmentation*-Elements (in diesem Fall das Element, das Speaker Contributions enthält) zu Zeilen gruppiert und die Annotationen jeweils direkt hinter das Segment geschrieben.

### 5.2.3. Manuelle Annotation mit dem Z2-Tagger

Der Z2-Tagger (siehe auch Seite 134) ist ein Werkzeug, das im Projekt Z2 des Sonderforschungsbereiches Mehrsprachigkeit entwickelt wurde, um Benutzern Unterstützung bei der manuellen (insbesondere POS-) Annotation von altnordischen Texten im TEI-Format zu geben. Da es für die untersuchten Sprachen keine Regelsätze für automatische Tagger gibt und aufgrund der großen Formenvarianz bei gleichzeitig relativ überschaubarer Quellenlage das Training eines automatischen Taggers unökonomisch wäre, müssen derartige Annotationen manuell vorgenommen werden.<sup>119</sup>

Der Z2-Tagger übernimmt dabei die Aufgabe, die zu annotierenden Abschnitte (in diesem Fall Wörter in TEI-*w*-Tags) in ihrem – bestimmbar – Kontext hervorzuheben und dem Anwender eine Auswahl von möglichen Werten für das gewünschte Attribut anzubieten, aus denen dieser einen Wert per Mausklick oder Tastendruck auswählt.

Dieses Programm, das also ursprünglich für die Annotation von OHCO-Daten konstruiert ist, wurde für die Annotation von segmentierten EXMARaLDA-Transkriptionen modifiziert.

Zunächst wird – wie in Abbildung 81 erkennbar – eine einzelne Transkriptions-

<sup>119</sup> Vgl. Höder, Wörner und Zeevaert (2007).

spur ausgewählt, für die die Annotationen durchgeführt werden sollen. Die Transkription wird also für die Annotation in einzelne Textstränge linearisiert – Annotationen über mehrere Sprecherspuren sind in diesem Annotationsszenario nicht vorgesehen.<sup>120</sup>

Darauf wird – analog zum Vorgehen bei TEI-Texten – der Segment-Tag gewählt, für den Annotationen vorgenommen werden sollen. Diese Auswahl wird mittels eines XPath-Ausdruckes vorgenommen, in unserem Beispiel ist dies der Ausdruck

```
//ts[@n='HIAT:w']
```

Dieser selektiert alle nach HIAT segmentierten Wörter.

Der Tagger benutzt automatisch das höchste Elternelement oberhalb dieses Tags, das unterhalb eines `segmentation`-Tags liegt, also im Beispielfall das Element

```
<ts n="sc">
```

das für einen Sprecherbeitrag steht. Alle Elemente, die unterhalb dieses Elements stehen, werden im Tagger-Fenster ausgegeben, und das jeweils aktuell zu annotierende Element wird hervorgehoben.

Im Unterschied zum Z2-Tagger für TEI-Dateien speichert der Tagger für EXMARaLDA-Transkriptionen die Annotation am Ende nicht als Attribute des annotierten Tags, sondern als Feature Structures in EXMARaLDA-Standoff-Annotationsdateien. In Zukunft könnte der Tagger die Standoff-Dateien auch mit den Transkriptionen öffnen können, insbesondere um sicherzustellen, dass bereits annotierte Segmente nicht unnötigerweise noch einmal annotiert werden.

---

120 Im abgebildeten Beispiel findet diese Auswahl noch über den XPath statt, der gleichzeitig die zu annotierenden Segmente bestimmt.

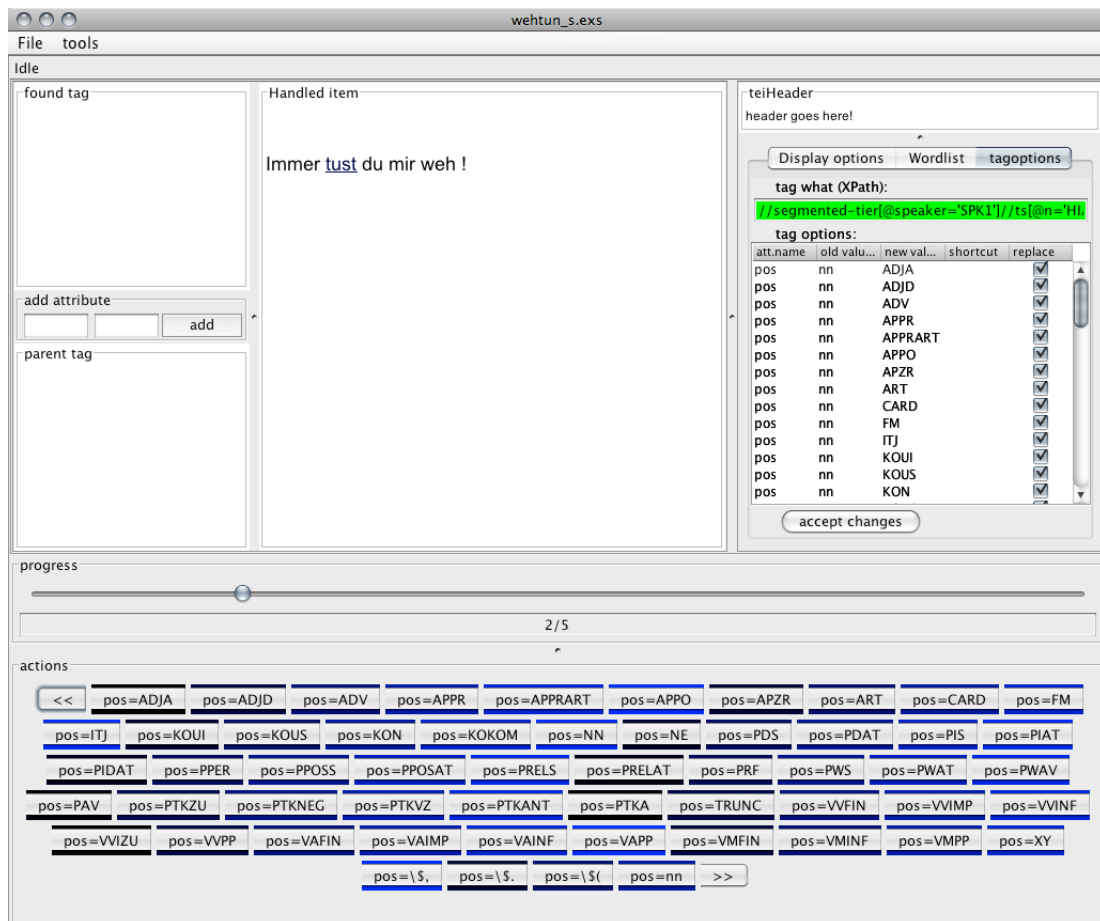


Abbildung 95: Eine modifizierte Version des Z2-Taggers für TEI-Texte kann zur Annotation von EXMARaLDA-Transkriptionen verwendet werden.

## 5.2.4. Wünschenswerte Ergänzungen

Die vorgestellten Komponenten sind geeignet, die grundsätzliche Einsetzbarkeit des hier erarbeiteten Annotationsverfahrens zu belegen. Um für Forscher wirklich nützlich zu sein, sind zusätzliche Werkzeuge bzw. Ergänzungen an bestehenden Werkzeugen wünschenswert. Einige – aus jetziger Sicht sinnvolle – Ergänzungen sollen im Folgenden kurz vorgestellt werden.

### Generische Visualisierungskomponenten

Die Visualisierung, die in Sextant Transkriptionen in Verbindung mit ihren Annotationen anzeigt, basiert auf einem relativ simplen Mechanismus, der nahe am linearen Text einzelner Sprecherspuren bleibt. Es wäre darüber hinaus wünschenswert, die Visualisierungsmöglichkeiten, die EXMARaLDA für Transkriptionen bereits im Partitur-Editor bietet, um die Anzeige von Standoff-

Annotationen zu erweitern.

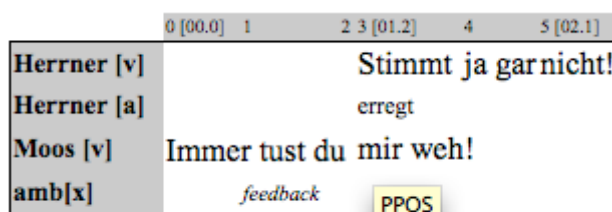


Abbildung 96: Anzeige der POS-Annotation mittels Mouseover-Popups in HTML.

Die Erstellung dieser Visualisierungskomponenten ist allerdings keineswegs trivial: Das oben abgebildete Beispiel wurde mit einem modifizierten Stylesheet für die Visualisierung von Basistranskriptionen als Partitur erstellt, das die segmentierten Wörter in `span`-Tags einschließt. Da die (hier annotierten) Segmente der HIAT-Segmentierung aber keine Übereinstimmung mit den (hier visualisierten) Segmenten aus der ursprünglichen Transkription haben müssen, lässt sich die Annotation nicht ohne weiteres für alle annotierten (HIAT:w)-Segmente visualisieren. Im gezeigten Beispiel wird das transkribierte Wort „garnicht“ beispielsweise durch eine Eventgrenze geteilt und lässt sich daher (in der Visualisierung) nicht einfach mit einem `span`-Tag einschließen.

Alle bereits beschriebenen Probleme mit überlappenden Hierarchien, derer man sich durch die Benutzung von Standoff-Annotation entledigt hat, kommen natürlich wieder zum Tragen, wenn man die einzelnen Komponenten für die Visualisierung serialisieren will. Für Visualisierungen, die als Grundlage die Segmente nutzen, die auch annotiert wurden (und deren Elternsegmente – im Beispiel also beispielsweise die Sprecheräußerungen), und die keine aufeinander aufbauenden, hierarchischen Annotationen inkorporieren müssen, sind solche Probleme dagegen nicht zu erwarten.

### Linearisierung für Annotationswerkzeuge

Für Transkriptionen, die mindestens in Teilen in Standardorthografie erstellt wurden, wäre es wünschenswert, existierende automatische Annotationswerkzeuge (insbesondere POS-Tagger) nutzen zu können, um die eindeutigen Teile einer Transkription automatisch verarbeiten zu können.

Für diesen Zweck wäre es möglich, einzelne Spuren der segmentierten Transkriptionen in ein einfacheres XML-Format zu überführen, das nur die zu annotierenden Segmente mit ihren eindeutigen Ids enthält, die vom gewünschten Tagger bearbeitet werden können. Sextant könnte diese annotierten XML-Dateien dann einfach in Standoff-Annotation überführen, die wieder auf die ursprüngliche Transkription verweist.

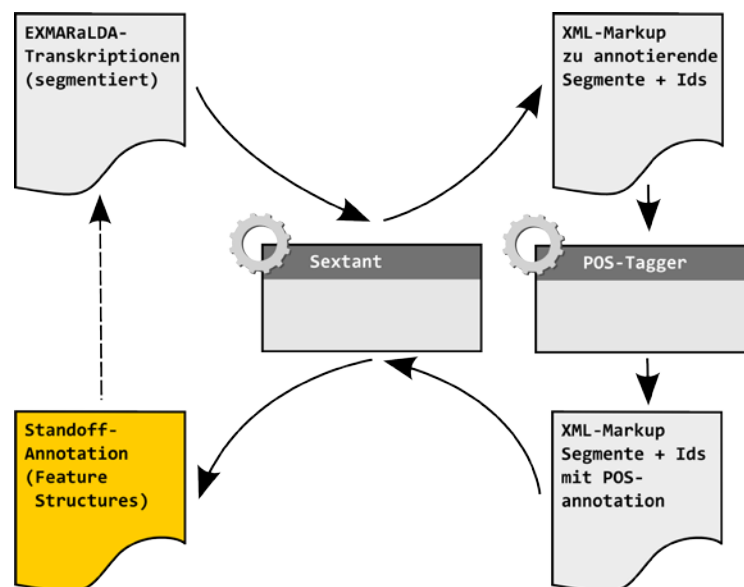


Abbildung 97: Linearisierung von einzelnen, segmentierten Spuren für POS-Tagger.

### „Annotation-Awareness“ für EXMARaLDA-Werkzeuge

Die bestehenden EXMARaLDA-Werkzeuge könnten in verschiedenen Stufen um Funktionen erweitert werden, die das erarbeitete System der Standoff-Annotationen als Bestandteil des EXMARaLDA-Systems berücksichtigen.

Besonders wichtig wäre es, die angelegten Transkriptionen im EXAKT-Suchwerkzeug auch durchsuchen zu können. Am einfachsten ließe sich realisieren, EXAKT zunächst die Annotationsdateien durchsuchen zu lassen und die Ergebnisse ebenfalls als KWIC-Konkordanz auszugeben, die als Keyword immer das annotierte Segment (und nicht die eigentliche Annotation) darstellt, so wie dies bei der Annotationssuche in EXAKT bereits jetzt geschieht.

In einer weiteren Stufe könnten Bedingungen für die Beziehungen zwischen Annotation und Annotiertem in die Suche integriert werden. Dieser Aufwand sollte

aber nur betrieben werden, wenn tatsächlicher Bedarf nach einer derartigen Lösung besteht.

Die Funktion in Sextant, die aus EXAKT-Konkordanzen Standoff-Annotationen erstellt, könnte natürlich direkt in EXAKT integriert werden. EXAKT ähnelt in seinem *Browsing-Mode* bereits jetzt einer einfachen Form des Z2-Taggers und könnte mit der Option, direkt Standoff-Annotationsdateien zu schreiben, als vollwertiges Werkzeug zur manuellen Annotation verwendet werden.

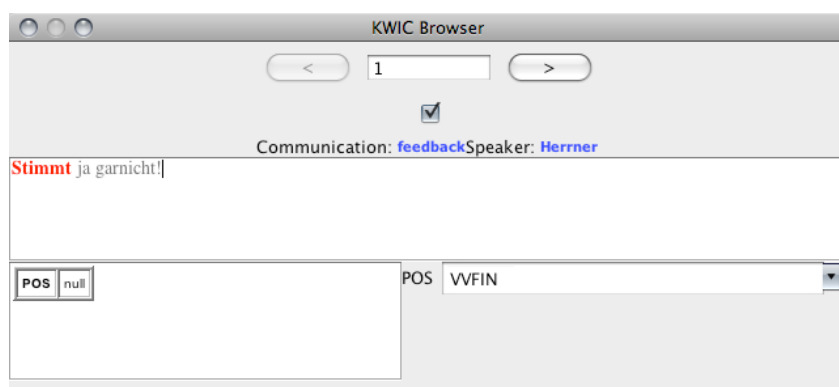


Abbildung 98: Browsing-Mode zur Annotation in EXAKT.

Sextant schreibt bei der Erzeugung von Standoff-Annotationsdateien die Namen der erzeugten Dateien in den Kopf der annotierten Transkription, damit Benutzer und Anwendungen „wissen“, welche Annotationen für diese Transkription zur Verfügung stehen. Das Korpus-Verwaltungswerkzeug Coma importiert diese Daten (wie alle anderen Daten aus den Transkriptionsköpfen) bereits jetzt und zeigt sie auch an, bietet aber noch keine weiteren Hilfestellungen bei der Verwaltung von Transkriptionen und deren Annotationen. Funktionen wie die Validierung der Annotationen (die jetzt in Sextant durchgeführt werden kann) und das Zuordnen von Annotationen zu Transkriptionen wären in einer Korpusverwaltung sicherlich besser aufgehoben.

Würden die genannten Desiderata in die EXMARaLDA-Werkzeuge integriert, so hätte Sextant in seiner jetzigen Form keine wirkliche Aufgabe mehr. Funktionen wie die Linearisierung für die automatische POS-Annotation in externen Programmen würden in diesem Fall lieber direkt im Partitur-Editor, der ohnehin über eine umfangreiche Import- und Exportfunktionalität verfügt, realisiert. Dort

könnten auch Entscheidungen darüber getroffen werden, ob die Annotationen in Annotationsspuren gleich als Standoff-Annotation gespeichert werden und ob Exporte in andere Formate für Transkription mit Annotation (insbesondere LAF) wünschenswert sein könnten.

### 5.3. Zusammenfassung und Ausblick

Das erarbeitete Standoff-Format ist in der Lage, EXMARaLDA-Transkriptionsgraphen mit (weitestgehend) beliebig komplexer, auch überlappender Annotation zu versehen. Der Einsatz von Standoff-Annotation mit Hilfe von XLink und XPointer-Verweisen auf (Teil-)Segmente erlaubt es, alle textuellen Bestandteile der Transkription zu adressieren.

Über die many-to-many-Beziehungen erweiterter XLinks ist es dabei auch möglich, diskontinuierliche Einheiten oder Segmente verschiedener Sprecher mit Annotationen zu versehen.

Einzig Abschnitte auf der Timeline, die nicht an einer Stelle in der Transkription explizit segmentiert sind, lassen sich nicht direkt adressieren. Dies sollte aber kein Problem darstellen, da sich derartige Abschnitte im Partitur-Editor markieren lassen.

Die Annotationen können aufgrund ihrer Realisierung als ISO-Feature Structures bereits selbst eine gewisse Komplexität erreichen. Da alle Annotationselemente wiederum selbst mit XML-Ids versehen sind, können sie wiederum als Ziel weiterer Annotationsebenen dienen, was quasi beliebig komplexe Annotationsstrukturen erlaubt.<sup>121</sup>

Die entstehenden Dateien liegen alle in Standard-XML vor. Alle Dateitypen (Transkriptionen und Standoff-Annotationsdateien) sind in XML-Schema-Grammatiken beschrieben, die verwendeten Pointer-Technologien ebenso, so dass eine grundlegende Nachhaltigkeit auf der Basis der Dateiformate gewährleistet ist. Zusätzlich lässt sich das Modell aus Transkriptionen und Annotationen grundsätzlich in andere, als Austauschmodelle konzipierte Modelle überführen, insbesondere in LAF (Seite 101 ff) und Annotation Graphs (Seite 82 ff).

Die Implementierung in Sextant und einer Erweiterung des Z2-Taggers hat ge-

---

121 Wie bei allen Standoff-Annotationsmodellen, bei denen Annotationen wieder auf Annotationen verweisen, können diese zwar beliebig komplex werden, die Verarbeitung solcher Annotationsnetze wird damit aber auch beliebig komplex. Für die angedachten Anwendungsfälle wird es selten mehr als zwei Annotationsebenen geben, die aufeinander Bezug nehmen – dies werden vor allem POS- und Syntaxannotationen sein.



zeigt, dass das Format softwareseitig benutzbar und nützlich ist. Für die Zukunft wäre es wünschenswert, die Funktionalität dieser Programme in die Programme des EXMARaLDA-Systems zu integrieren. Insbesondere die Durchsuchbarkeit der Annotation ist dabei ein Desideratum, das kurzfristig angegangen werden kann, ebenso wie die Möglichkeit, segmentierte Spuren aus Transkriptionen für externe Annotationswerkzeuge aufzubereiten. Die Kombination dieser beiden Funktionen würde für die Nutzer des EXMARaLDA-Systems unmittelbar eine beträchtliche Erweiterung ihrer Analysemöglichkeiten für bereits bestehende EXMARaLDA-Korpora bedeuten, ohne dass zuvor Änderungen an den vorhandenen Transkriptionen durchgeführt werden müssten.

Tatsächliche Annotationswerkzeuge, die von dem neuen Annotationsformat Gebrauch machen, sollten nicht entwickelt werden, solange kein konkreter Bedarf und damit kein konkretes Anforderungsprofil an ein solches Werkzeug besteht – ebenso verhält es sich mit komplexeren Visualisierungsverfahren, die je nach Typ der vorgenommenen Annotation und Forschungsinteresse stark variieren können.

## 6. Konklusion

Die Arbeit trat mit dem Anspruch an, eine Brücke von der „Annotation Science“, die als zu wenig an der Praxis orientiert empfunden wird, zur Forschungspraxis in den Bereichen der Gesprächsforschung, pragmatischen Linguistik und Diskursanalyse zu schlagen. Dazu wurde ein neues Annotationsverfahren entwickelt, mit dem sich Transkriptionen mit potenziell komplexen Annotationen anreichern lassen. Die Schritte, die auf diesem Weg unternommen wurden, sollen hier noch einmal in stark komprimierter Form nachgegangen werden.

### 6.1. Zusammenfassung

Im Kapitel 2 wurden zunächst die Besonderheiten aufgeführt, die gesprochene Sprache von geschriebener Sprache unterscheiden, und erläutert wie sich diese auf die Modellierung für die computergestützte Verarbeitung auswirken können. Dazu mussten die Begriffe *geschriebene Sprache* und *gesprochene Sprache* zunächst klar abgegrenzt werden. So existieren verschiedene Definitionen, was gesprochene Sprache und geschriebene Sprache oder Schriftsprache ausmacht.

Diese kommen dadurch zustande, dass sich Sprache zunächst in der Art ihrer physischen Realisierung, des Mediums, einteilen lässt, also danach, ob sie tatsächlich (auf-) geschrieben oder mündlich geäußert wird. Zum anderen lässt sie sich auf der Ebene der Konzeption in einem Kontinuum von Mündlichkeit und Schriftlichkeit aufteilen.

Diese beiden Ebenen spielen häufig ineinander: So ist mündlich geäußerte Sprache meist auch mündlich konzeptioniert und umgekehrt. Es wurde festgestellt, dass für die Betrachtung dieser Arbeit die Unterscheidung anhand der physischen Realisierung zwar die entscheidende ist (schließlich geht es um die Transkription und Annotation tatsächlicher gesprochener Diskurse), die Besonderheiten bei der Konzeption, die für Mündlichkeit charakteristisch sind, aber ebenfalls Auswirkungen auf die Modellierung haben. Einige dieser Besonderheiten wurden daher vorgestellt und bezüglich ihrer Auswirkungen auf die Modellierung betrachtet.

Für die Untersuchung geschriebener Sprache werden in den meisten Fällen Transkriptionen angefertigt. Transkription als Methode existiert aber auch in verschiedenen anderen Forschungsbereichen, mit zum Teil stark abweichenden Verfahren und Begriffsdefinitionen. Einige dieser Varianten von Transkription wurden in der Arbeit vorgestellt, um dann auf die für die fokussierten Forschungsbereiche relevante Form der Transkription näher einzugehen.

Dazu wurden einige Eigenheiten der Transkription näher betrachtet. So wurde festgestellt, dass eine Transkription niemals theoriefrei sein kann. Bereits die Auswahl, welche Phänomene transkribiert werden, ist meistens durch eine von der Theorie geleitete Fragestellung bestimmt, ebenso die Konventionen, nach denen eine Transkription regelhaft angefertigt wird. Der prozesshafte Charakter von Transkription ist ein weiterer wichtiger Aspekt, der beschrieben wurde. So ist das Anfertigen einer Transkription ein potentiell niemals abgeschlossener Prozess, insbesondere, wenn die wissenschaftliche Methode, mit der ein Korpus solcher Transkriptionen betrachtet wird, eher *Corpus Driven* ist, die Fragestellungen also auch direkt aus der Arbeit mit den Transkriptionen ableitet.

Über die Betrachtung der Funktion von Transkriptionen wurde hergeleitet, dass computergestützte Transkriptionen als (nur) eine mögliche Modellierung des transkribierten Gesprächsereignisses gelten müssen und nicht etwa als seine a priori determinierte Verschriftlichung, auf die bei der Beschreibung des Begriffs Transkription immer wieder verwiesen wird.

Bei der Erstellung eines solchen Modells gibt es einige Besonderheiten zu beachten – auch diese wurden im zweiten Kapitel näher betrachtet. Insbesondere die Tatsache, dass es in Diskursen zur zeitlichen Überlappung von Beiträgen einzelner Sprecher kommen kann, ist ein für die Modellierung hoch relevanter Umstand. In denselben Bereich fallen weitere Aspekte, die mit der Einordnung der sprachlichen Phänomene in ein zeitliches Kontinuum befasst sind. Auch Annotationen, also Informationen, die der reinen Abbildung der verbalen Äußerungen von Transkribenden hinzugefügt werden, können sich überlappen und die Modellierung erschweren.

Eine weitere Besonderheit ist, dass für die Transkription gesprochener Sprache eine Entscheidung getroffen werden muss, wie mit der unterschiedlichen Lautung und Schreibung von Sprache umgegangen werden soll. Einige etablierte Verfahren dazu wurden in der Arbeit vorgestellt – eine klare Empfehlung ließ sich daraus allerdings nicht ableiten, da eine solche Entscheidung eben wieder vom speziellen Forschungsinteresse abhängt.

Für die Modellierung kann dies bedeuten, dass in einem Datenmodell möglichst viele dieser Möglichkeiten realisierbar sein sollten. Die Anforderungen für ein zu erstellendes Datenmodell wurden am Ende des zweiten Kapitels zusammengefasst. Dabei wurde postuliert, dass ein solches Datenmodell

- über eine Zeitachse für die zeitliche Einordnung von Sprecherbeiträgen verfügen muss,
- die Überlappung von Sprecherbeiträgen und Annotationen zulassen muss,
- verschiedene Arten der Segmentierung von sprachlichen Einheiten zulassen sollte und
- und möglichst viele Schriftsysteme unterstützen sollte.

Auf dieses Anforderungsprofil hin wurden im dritten Kapitel einige existierende Datenmodelle untersucht. Dabei wurde eine grundsätzliche Unterscheidung zwischen zwei Modellen getroffen. Auf der einen Seite stehen die Modelle, die Sprache stärker in einer geordneten hierarchischen Struktur sehen. In der stärksten Ausprägung einer solchen Struktur schließen Elternelemente ihre Kindelemente immer ein, die Abfolge der untersten Ebene der Kindelemente ergibt den Text und es kann keine überlappenden Hierarchien geben. Da sich eine derart strikte These, die Text als eine so genannte Ordered Hierarchy of Content Objects (OHCO) sieht, in der Praxis der linguistischen Forschung nicht aufrecht erhalten werden konnte, erlauben Modelle, die ebenfalls von einer eher hierarchischen Organisation von Sprache als naheliegendste Modellierungsform ausgehen (wie beispielsweise die TEI-Guidelines), bestimmte Konstrukte, die beispielsweise die Modellierung überlappender Hierarchien ermöglichen.

Trotzdem wurde festgestellt, dass sich solche Modelle besser für die Modellierung von Texten geschriebener Sprache eignen. Für die computergestützte Verarbeitung von gesprochener Sprache haben sich Modelle etabliert, die auf gerichteten, azyklischen Graphen basieren, die den Aspekt des zeitlichen Ablaufes stärker in den Vordergrund stellen und weniger Probleme mit Phänomenen wie der Überlappung von Sprecherbeiträgen haben.

Die Modelle, die für die Arbeit auf ihre Eignung für die Annotation von Transkriptionen gesprochener Sprache untersucht wurden, waren auf beiden Seiten dieses Spektrums zu finden, Modelle wie das Nite Object Model (NOM) ließen sich nicht eindeutig einer Seite zuordnen.

Für die weitere Untersuchung wurde das Modell der Transkriptionsgraphen gewählt, auf dem die Werkzeuge des EXMARaLDA-Systems aufgebaut sind. Dieses Modell erfüllt nicht nur die zuvor aufgestellten Anforderungen an ein Datenmodell, es liegen vor allem auch Werkzeuge für die Bearbeitung dieses Modells und Korpora im dazugehörigen Format vor, an denen sich die entwickelten Lösungen testen lassen.

Im vierten Kapitel wurden einige praktische Fragen zur Transkriptionsarbeit erörtert. Dabei wurden zunächst die bereits bestehenden Annotationsmöglichkeiten des EXMARaLDA-Systems beleuchtet. So lassen sich bereits direkt in der Transkriptionssoftware, dem Partitur-Editor, Annotationen an der Transkription vornehmen. Auch das Suchwerkzeug EXAKT bietet die Möglichkeit, Suchergebnisse, die in einer KWIC-Konkordanz dargestellt werden, mit Annotation zu versehen.

Ein Problem, das alle existierenden Werkzeuge des EXMARaLDA-Systems teilen, wurde bei dieser Betrachtung ganz deutlich: In EXMARaLDA existieren zwei verschiedene Systeme, nach denen Transkriptionen in Segmente unterteilt werden. Zum einen existieren Segmente, die sich auf die Zeitachse beziehen und Einheiten wie Sprecherbeiträge, Pausen, Überlappungen etc. markieren. Auf der anderen Seite stehen Segmente, die vom Programm aufgrund von Transkriptionskonventionen automatisch in die Transkription eingefügt werden. Diese Einheiten

markieren (je nach verwendeter Konvention) Segmente wie Wörter, Äußerungen und dergleichen. Insbesondere für Annotation letzterer Segmente existiert noch keine Lösung, weder auf Modell- noch auf Anwendungsebene – dabei ist die Annotation dieser Segmente für Linguisten besonders interessant.

Um einen Überblick zu bekommen, wie eine solche Annotationsumgebung aussehen könnte, wurden in Kapitel 4 auch noch zwei weitere Transkriptionswerkzeuge auf ihre Annotationsmöglichkeiten untersucht: ELAN und Praat. Dabei wurde festgestellt, dass diese Programme für die aufgeworfene Fragestellung auch keine überzeugende Lösung anbieten. Kapitel 4 schloss mit einem Anforderungskatalog für ein Annotationssystem im Kontext von EXMARaLDA und der zugrunde liegenden Transkriptionsgraphen, der einen starken Fokus darauf legt, dass für die Umsetzung keine umfassenden Änderungen an bestehenden Daten und Werkzeugen notwendig sein sollen.

Im 5. Kapitel schließlich wurde aus den erarbeiteten Anforderungen ein Szenario entwickelt, mit dem sich EXMARaLDA-Transkriptionen mit Annotationen versehen lassen – und zwar in einer Form, die vorher nicht zur Verfügung stand.

Dafür wurde ein Format entwickelt, das die Annotationen in Form von Feature Structures auf alle möglichen Segmente einer EXMARaLDA-Transkription (und Abschnitte unterhalb dieser Segmente) anwenden kann. Damit lassen sich auch die über Transkriptionskonventionen eingeführten Segmente, die bisher nicht annotiert werden konnten, annotieren.

Für die Form der Annotation wurde Standoff-Annotation gewählt, bei der die Annotationsdateien für die einzelnen Annotationsebenen separat von der annotierten Datei gespeichert werden. Damit werden zum einen Überlappungen, die bei verschiedenen Annotationsebenen sehr leicht auftreten können, keine Probleme auf, zum anderen müssen an den Transkriptionsdateien zunächst keinerlei Änderungen vorgenommen werden, um sie mit Annotationen versehen zu können. Das Problem, das Standoff-Annotation aber zwangsläufig mit sich bringt, ist, dass die Verweise von Annotationen auf Annotiertes in irgendeiner Form sichergestellt werden müssen, wenn die Möglichkeit besteht, dass sich an den Trans-

kriptionen noch einmal Änderungen ergeben, nachdem die Annotation vorgenommen wurde. Zu diesem Aspekt wurden einige Überlegungen angestellt, diese lassen sich aber nicht umsetzen, ohne tatsächlich Änderungen an den bestehenden Werkzeugen vorzunehmen.

Die eigentlichen Verweise von Annotationen auf Annotiertes werden mit Mitteln von XLink und XPointer, zwei etablierten XML-Technologien, hergestellt, so dass sich die entwickelte Lösung nicht aus dem XML-Rahmen heraus bewegt. Dies ist insbesondere für die Nachhaltigkeit der entstehenden Korpora ein wichtiger Aspekt. Die zuvor aufgestellten Anforderungen für das Annotationsformat wurden damit erfüllt.

Anschließend beschäftigte sich das Kapitel 5 damit, wie die erarbeitete Lösung in der Praxis (also in konkreten Anwendungen) eingesetzt werden kann. Dazu wurde zunächst eine Dokumentgrammatik für das Standoff-Annotationsformat erstellt, mit deren Hilfe sich Annotationen auf ihre Gültigkeit hin überprüfen lassen.

Im Folgenden wurden zwei Werkzeug-Prototypen, die für diese Arbeit erstellt wurden, vorgestellt. Das Erste, Sextant, dient der Zuordnung von Transkriptionen und Annotationen. Es lassen sich die Verweise von Annotation zu Annotiertem überprüfen und einfache Ausgaben von beidem damit anfertigen – dieses Programm dient vor allem als Proof-of-Concept für das entwickelte Datenformat.

Das zweite Werkzeug basiert auf einem existierenden Werkzeug, das ursprünglich zur manuellen Annotation von TEI-Texten entwickelt wurde. Es wurde dahingehend erweitert, dass es EXMARaLDA-Transkriptionen verarbeiten (und annotieren) und das Ergebnis der Annotation in Dateien im neu entwickelten Standoff-Annotationsformat speichern kann. Es liefert den Beweis, dass sich mit relativ einfachen Mitteln Anwendungen dahingehend entwickeln oder erweitern lassen, dass sie das neue Annotationsformat und die Annotation von segmentierten EXMARaLDA-Transkriptionen unterstützen.

Um von den neuen Möglichkeiten wirklich profitieren zu können, bedarf es aber noch weiterer Unterstützung für potentielle Benutzer: insbesondere in den be-

stehenden Werkzeugen des EXMARaLDA-Systems wären Änderungen denkbar, die die Arbeit mit dem neuen Annotationsformat fruchtbar gestalten könnten. Auf diese Aspekte wurde am Ende des fünften Kapitels eingegangen: Vor allem die Berücksichtigung der neuen Annotationen in der Suche mit dem EXMARaLDA Suchwerkzeug sowie die Integration in bestehende Visualisierungskomponenten wären dabei als wichtigste Aufgaben zu nennen.



## 6.2. Diskussion und Ausblick

Es war das Ziel dieser Arbeit, eine Brücke zu schlagen zwischen der in der Einleitung als zu wenig praxisorientiert kritisierten *Annotation Science* auf der einen Seite und der Praxis der Arbeit mit Transkriptionen in Forschungsbereichen, die sich im weitesten Sinne der linguistischen Untersuchung von Diskursen widmen auf der anderen Seite.

Dazu wurden beide Wissenschaftsbereiche eingehend beleuchtet: Auf der einen Seite strebt die *Annotation Science* an, Datenmodelle zur Verfügung zu stellen, die im besten Fall alle, mindestens aber alle weiter verbreiteten Arten von Sprachdaten mit ihren Annotationen abbilden können. Die Diskussion um das beste Modell ist rege und produziert sichtbare Ergebnisse. Allerdings führt die Tatsache, dass das perfekte Modell nicht gefunden werden *kann*, dazu, dass selten die Mühe unternommen wird, ein entwickeltes Modell auch in einsetzbare Anwendungen umzusetzen – die nächste Modifikation am Modell steht meistens bereits vor der Tür, selbst bei Initiativen, die darauf hinarbeiten, einen Standard zu entwickeln.

Auf der anderen Seite stehen die Disziplinen der Gesprächsforschung, pragmatischen Linguistik und Diskursanalyse. Obwohl sie es selbst waren, die (Stichwort *Pragmatische Wende*) die empirische Arbeit mit Korpora (wieder) in den Fokus des linguistischen Interesses gerückt haben, wurden sie von den Fortschritten und damit Vorzügen der computergestützten Korpuslinguistik zunehmend abgehängt. Die von ihnen entwickelten Methoden, insbesondere der Transkription gesprochener Sprache, waren ursprünglich nicht für die Arbeit mit sehr großen Datenmengen ausgelegt. Jetzt, da sich die Werkzeuge, mit denen sich große Korpora solcher Transkriptionen erstellen, verwalten und untersuchen lassen, mehr und mehr etablieren, wächst auch der Bedarf an praktikablen Annotationsmethoden. Während für schriftliche Texte eine Vielzahl von Methoden und Werkzeugen existieren, gibt es im Bereich der gesprochenen Sprache noch erhebliche Defizite.

Ausgehend von einer Evaluation existierender Modelle einerseits und einer Analyse der Anforderungen an eine Transkriptions- und Annotationsumgebung andererseits wurde in dieser Arbeit ein Verfahren entwickelt, mit dem Transkriptionen gesprochener Sprache mit Annotationen versehen werden können. Diese Form stellt für die diskurs- und gesprächsanalytische Forschungspraxis eine Neuerung dar, lässt sich gleichzeitig aber im Zusammenspiel mit bereits etablierten Verfahren nutzen.

Dabei wurde das Verfahren – bezogen auf das Modell – bewusst offen und gleichzeitig einfach beschreibbar gehalten. Es lässt (insbesondere durch die Möglichkeit, Annotationen wieder auf Annotationen zu beziehen) Annotationen fast beliebiger Komplexität zu. Vor allem können aber einfache (und in der Praxis besonders häufig benötigte) Annotationen wie die 1:1-Zuordnung von Kategorien zu Segmenten der Transkription in einer sehr einfach nachvollziehbaren Art und Weise angefertigt und verarbeitet werden. Dass das Modell algebraisch nicht weiter eingeschränkt ist, macht die automatische Validierung und Verarbeitung zwar schwieriger – dieser Umstand wurde zu Gunsten der oben beschriebenen Offenheit aber in Kauf genommen.

Das in dieser Arbeit entwickelte Verfahren baut auf dem Modell des EXMARaLDA-Systems auf, da dieses die Grundlage für die Arbeit mit mehreren weit verbreiteten Transkriptionssystemen bietet. Außerdem stehen für das System bereits eine Reihe umfangreicher Korpora zu Verfügung. So können existierende und zukünftige Werkzeuge an *realen* Daten erprobt werden.

Die Werkzeug-Prototypen, die für diese Arbeit implementiert wurden, können bereits eine wertvolle Hilfe für Forscher sein, die existierende Transkriptionen manuell mit Annotationen auf verschiedenen Ebenen anreichern wollen. Zwar erlauben diese nur einfache, lineare Annotationstypen, diese sind aber gleichzeitig diejenigen, die von Anwendern des EXMARaLDA-Systems am stärksten nachgefragt werden.

Wie bestehende Werkzeuge weiterentwickelt werden und welche Werkzeuge eventuell neu entwickelt werden müssen, wird die Praxis zeigen. Zunächst sollten

die bestehenden EXMARaLDA-Werkzeuge so modifiziert werden, dass sie auf die Existenz möglicher Annotationen vorbereitet sind. Für das Suchwerkzeug ist es speziell wichtig, dass es auch vorliegende Annotationen in die Suche einbeziehen kann. Unabdingbar wird auch sein, weitere Visualisierungsmethoden für Transkriptionen bereitzustellen, die mit Annotation versehen wurden.

Mit der Entwicklung und Implementation dieses Verfahrens konnte also eine Brücke zwischen Annotation Science und der Forschungspraxis der Gesprächs- und Diskursanalyse geschlagen werden. Das gewählte Annotationsmodell kombiniert Standoff-Feature Structure-Annotation, die in ähnlicher Form für das Linguistic Annotation Framework vorgesehen ist, mit einem auf Annotation Graphs basierenden Datenmodell. Es verbindet dabei die Vorzüge von beiden: So eignet sich das Transkriptionsgraphenmodell mit seinem starken Zeitachsenbezug besser für die Anfertigung der zu annotierenden Transkriptionen und unterstützt eher die Prozesshaftigkeit, die der Transkriptionsarbeit inhärent ist. Die Standoff-Annotationen dagegen ermöglichen zum einen konkurrierende und mit Feature Structures auch komplexere Annotationen, als diese im Annotationsgraphenformalismus möglich wären.

Es wäre durchaus wünschenswert, wenn Entwicklungen bei der Erstellung neuer texttechnologischer Modelle und Formate häufiger und früher in praktischen Anwendungen erprobt werden könnten. So könnte rechtzeitig ermittelt werden, ob die berücksichtigten Probleme in der Forschungspraxis auch tatsächlich als solche wahrgenommen werden und ob die Anforderungen der Anwender in der erarbeiteten Methode auch ihren Niederschlag finden.

# 7. Verzeichnisse

## 7.1. Korpora

Beispiele, die mit einer Korpusangabe versehen sind, stammen aus einem der im Folgenden aufgeführten Korpora. Sofern sich hinter der Angabe des Korpus-Namens noch ein Vermerk zur Position des Beispiels innerhalb des Korpus befindet, so bezieht sich dieser auf den Namen des betreffenden Diskurses im Korpus und findet sich in der jeweiligen Korpusübersicht wieder.

**CoSi:** Das CoSo Korpus enthält Audio-Aufnahmen von konsekutiv und simultan gedolmetschten Vorträgen. Es ist im Projekt „Zusammenhangbildung in gedolmetschten Diskursen (K6)“ unter der Leitung von Juliane House am Sonderforschungsbereich Mehrsprachigkeit entstanden. Audioaufnahmen, Transkriptionen und eine nähere Beschreibung des Korpus sind auf Anfrage frei verfügbar unter:

**[http://www.exmaralda.org/corpora/sfb\\_k6.html](http://www.exmaralda.org/corpora/sfb_k6.html)**

**Demokorpus:** Das EXMARaLDA-Demokorpus ist eine Sammlung von transkribierten Audio- und Videoaufnahmen aus verschiedenen Genres und in verschiedenen Sprachen. Es wird unter der Leitung von Thomas Schmidt im Rahmen des Teilprojektes „Computergestützte Erfassungs- und Analysemethoden multilingualer Daten (Z2)“ (in dem der Verfasser dieser Arbeit arbeitet) am Sonderforschungsbereich Mehrsprachigkeit erstellt und dient vor allem dazu, die EXMARaLDA-Werkzeuge mit Daten ausprobieren zu können. Das Korpus ist frei verfügbar unter:

**<http://www.exmaralda.org/corpora/demokorpus.html>**

**DiK:** Das DiK (Dolmetschen im Krankenhaus)-Korpus enthält Arzt-Patienten-Kommunikation mit Beteiligung von nicht-muttersprachlichen Patienten und ad-hoc-Übersetzern. Es wurde im Rahmen des Projektes „Dolmet-

schen im Krankenhaus (K2)“ unter Leitung von Kristin Bührig am Sonderforschungsbereich Mehrsprachigkeit der Universität Hamburg erstellt und ist in Bührig et al. 2004 und Meyer 2004 beschrieben.

Die Transkriptionen des Korpus sind auf Anfrage verfügbar:

**[http://exmaralda.org/corpora/sfb\\_k2.html](http://exmaralda.org/corpora/sfb_k2.html)**

**Falko:** Das Falko-Korpus (Fehlerannotiertes Lernerkorpus des Deutschen als Fremdsprache) enthält Texte von fortgeschrittenen Lernern des Deutschen. Es befindet sich im Aufbau und ist in Lüdeling et al. 2008 näher beschrieben. Das Korpus ist frei verfügbar. Nähere Informationen finden sich unter:

**<http://www.linguistik.hu-berlin.de/institut/professuren/korpuslinguistik/forschung/falko>**

**GAT:** Hier handelt es sich lediglich um eine Transkription einer einzelnen Audioaufnahme („Der widerliche Kerl“), die als eine Art Standardbeispiel für das Gesprächsanalytische Transkriptionssystem (GAT, (Selting et al. 1998)) dient. Audioaufnahme und GAT-Basistranskript sind u.a. erhältlich unter:

**[http://www.teachsam.de/deutsch/d\\_lingu/gespraechsanalyse/gespraech\\_9\\_4\\_3\\_7.htm](http://www.teachsam.de/deutsch/d_lingu/gespraechsanalyse/gespraech_9_4_3_7.htm)**

**H3:** Das H3-Korpus enthält schriftliche Quellen skandinavischer Sprachen des 12.-16. Jahrhunderts. Es wird im Projekt „Skandinavische Syntax im mehrsprachigen Kontext (H3)“ unter der Leitung von Kurt Braunmüller am Sonderforschungsbereich Mehrsprachigkeit erstellt. Die Texte liegen nach Vorgaben der TEI und des Medieval Nordic Text Archive (Menota) XML-codiert vor.

Teile des Korpus sind auf Anfrage erhältlich:

**[http://exmaralda.org/corpora/sfb\\_h3.html](http://exmaralda.org/corpora/sfb_h3.html)**

**Paidus:** Das PAIDUS-Korpus (Parameterfixierung im Deutschen und Spanischen) enthält Audio-Aufnahmen und Transkriptionen (nach IPA mit orthografischer- und Silbenstruktur-Annotation) von monolingualen deutschen- und spanischen Kindern. Diese wurden im Rahmen des Projektes „Prosodische Beschränkungen zur phonologischen und morphologischen Entwicklung im bilingualen Spracherwerb (E3)“ des Sonderforschungsbereiches Mehrsprachigkeit unter Leitung von Conxita Lleó erstellt. Die Veröffentlichung des Korpus ist innerhalb von PhonBank (<http://childes.psy.cmu.edu/phon/>) Mitte 2010 geplant. Nähere Informationen zum Korpus sind verfügbar unter:  
**[http://exmaralda.org/corpora/sfb\\_e3.html](http://exmaralda.org/corpora/sfb_e3.html)**

**Rehbein:** Die Rehbein-ENDFAS (*Die Entwicklung narrativer Diskursfähigkeiten im Deutschen und Türkischen in Familie und Schule*) und Rehbein-SKOBI (*Sprachliche Konnektivität bei bilingual türkisch-deutsch aufwachsenden Kindern und Jugendlichen*) enthalten Transkriptionen und Audio-Aufnahmen von türkisch und deutsch monolingualen sowie türkisch-deutsch bilingualen Kindern. Die Aufnahmen wurden als Evokative Feldexperimente durchgeführt. Sie wurden im Rahmen des Projektes „Sprachliche Konnektivität bei bilingual türkisch-deutsch aufwachsenden Kindern (E5)“ des Sonderforschungsbereiches Mehrsprachigkeit unter der Leitung von Jochen Rehbein erstellt.  
Die Transkriptionen sind auf Anfrage verfügbar:  
**[http://www.exmaralda.org/corpora/sfb\\_e5.html](http://www.exmaralda.org/corpora/sfb_e5.html)**

## 7.2. Bibliographie

**Altmann, Hans (1981)**

Formen der "Herausstellung" im Deutschen

In: Linguistische Arbeiten, Vol. 106

Tübingen: Niemeyer

Verfügbar: <http://lcn.loc.gov/81173268>.

**Auer, Peter (1991)**

Vom Ende deutscher Sätze

In: Zeitschrift für Germanistische Linguistik, 19, Nr. 2, S. 141-157

Verfügbar: <http://dx.doi.org/10.1515/zfgl.1991.19.2.141>

**Bird, Steven / Day, David / Garofolo, John / Henderson, John / Laprun, Christoph / Liberman, Mark (2000)**

ATLAS: A Flexible and Extensible Architecture for Linguistic Annotation.

In: Proceedings of the Second International Conference on Language Resources and Evaluation ELRA

Paris

**Bird, Steven/Buneman, Peter/Tan, Wang-chiew (2000)**

Towards a Query Language for Annotation Graphs

In: Proceedings of the Second International Conference on Language Resources and Evaluation, S. 807-814

**Bird, Steven / Liberman, Mark (2001)**

A formal framework for linguistic annotation

In: Speech Communication, 33, S. 23-60

**Bird, Steven / Maeda, Kazuaki / Ma, Xiaoyi / Lee, Haejoong / Randall, Beth / Zayat, Salim (2002)**

Table-Trans, MultiTrans, InterTrans and TreeTrans: Diverse Tools Built on the Annotation Graph Toolkit

In: Proceedings of the Third International Conference on Language Resources and Evaluation, S. 364-370

Paris: ELRA

Verfügbar: [http://arxiv.org/PS\\_cache/cs/pdf/0204/0204006v1.pdf](http://arxiv.org/PS_cache/cs/pdf/0204/0204006v1.pdf)

**Bleiching, Doris / Drexel, Guido / Fischer, Kerstin / Gibbon, Dafydd / Lungen, Harald / Pampel, Martina (1995)**

Morphologie im Forschungsprototypen

Verbmobil Memo 83

Universität Bielefeld

- Blühdorn, Hardarik (2008)**  
 Epistemische Lesarten von Satzkonnectoren - Wie sie zustande kommen  
 und wie man sie erkennt  
 In: Pohl, Inge (Hg.)  
 Semantik und Pragmatik – Schnittstellen, S. 217-251  
 Frankfurt: Peter Lang
- Boag, Scott / Chamberlin, Don / Fernández, Mary F / Florescu, Daniela /  
 Robie, Jonathan / Siméon, Jérôme (2007)**  
 XQuery 1.0: An XML Query Language  
 W3C Recommendation  
 Verfügbar: <http://www.w3.org/TR/xquery/>.
- Boersma, Paul / Weenink, David (2009)**  
 Praat: doing phonetics by computer (Computer Program, Version 5.1.02)  
 Verfügbar: <http://www.fon.hum.uva.nl/praat/>
- Bührig, Kristin / Meyer, Bernd (2004)**  
 Ad hoc interpreting and achievement of communicative purposes in brief-  
 ings for informed consent  
 In: Multilingual Communication  
 Amsterdam: Benjamins
- Burger, Susanne / Kachelrieß, Elke (1996)**  
 Aussprachevarianten in der VERBMOBIL-Transliteration - Regeln zur  
 konsistenteren Verschriftung  
 In: Verbmobil Memo 111  
 Institut für Phonetik und sprachliche Kommunikation, München.
- Burger, Susanne (1997)**  
 Transliteration spontansprachlicher Daten  
 Verbmobil Technisches Dokument, Vol. 56  
 Institut für Phonetik und sprachliche Kommunikation, München.
- Burnard, Lou / Bauman, Syd (2007)**  
 TEI P5: Guidelines for Electronic Text Encoding and Interchange  
 TEI Consortium  
 Verfügbar: [http://www.tei-c.org/release/doc/tei-p5-  
 doc/en/html/index.html](http://www.tei-c.org/release/doc/tei-p5-doc/en/html/index.html).
- Bühler, Karl (1978)**  
 Sprachtheorie. Die Darstellungsfunktion der Sprache  
 Frankfurt, Berlin, Wien: Ullstein



- Carletta, Jean / Kilgour, Jonathan / O'Donnell, Timothy / Evert, Stefan / Voormann, Holger (2003)**  
 The NITE object model library for handling structured linguistic annotation on multimodal data sets  
 In: Proceedings of the EACL Workshop on Language Technology and the Semantic Web (3rd Workshop on NLP and XML, NLPXML-2003)  
 Budapest
- Carletta, Jean / Evert, Stefan / Heid, Ulrich / Kilgour, Jonathan (2005)**  
 The NITE XML Toolkit: Data Model and Query Language  
 In: Language Resources and Evaluation 39, no. 4, S. 313-334
- Clark, James / De Rose, Steve (1999)**  
 XML Path Language (XPath) Version 1.0  
 W<sub>3</sub>C  
 Verfügbar: <http://www.w3.org/TR/xpath/>
- DeRose, Steven J. / Durand, David G. / Mylonas, Elli / Renear, Allen H. (1990)**  
 What is text, really  
 In: Journal of Computing in Higher Education, 3, S. 3-26
- Deppermann, Arnulf (2006)**  
 Construction Grammar Eine Grammatik für die Interaktion?  
 In: Deppermann, Arnulf / Fiehler, Reinhard / Spanz-Fogasy, Thomas (Hg.)  
 Grammatik und Interaktion, S. 43-65,  
 Radolfzell: Verlag für Gesprächsforschung
- Dipper, Stefanie / Götze, Michael / Skopeteas, Stavros (2004)**  
 ANNIS: A Linguistic Database for Exploring Information Structure  
 In: Ishihara, S. / Schmitz, M. / Schwarz, A. (Hg.)  
 Working Papers of the SFB632, 1, S. 245-279
- Dittmar, Norbert (2004)**  
 Transkription. Ein Leitfaden mit Aufgaben für Studenten, Forscher und Laien  
 VS Verlag für Sozialwissenschaften
- Dutoit, Thierry (1997)**  
 An Introduction to Text-to-Speech Synthesis, Vol. 3  
 Dordrecht: Kluwer
- Ehlich, Konrad (1992)**  
 Computergestütztes Transkribieren - das Verfahren HIAT-DOS  
 In: Richter, Günther (Hg.)  
 Methodische Grundfragen der Erforschung gesprochener Sprache , S. 47-59  
 Frankfurt: Peter Lang

- Ehlich, Konrad (1994)**  
 Funktion und Struktur schriftlicher Kommunikation  
 In: Günther, Hartmut / Ludwig, Otto (Hg.)  
 Schrift und Schriftlichkeit. Ein interdisziplinäres Handbuch internationaler Forschung (1. Halbband), S. 18-41  
 Berlin/New York: Walter de Gruyter
- Ehlich, Konrad / Rehbein, Jochen (1979)**  
 Erweiterte halbinterpretative Arbeitstranskriptionen (HIAT<sub>2</sub>): Intonation  
 In: Linguistische Berichte, 59, S. 51-75.
- Ehlich, Konrad / Rehbein, Jochen (1976)**  
 Halbinterpretative Arbeitstranskriptionen (HIAT)  
 In: Linguistische Berichte, 45, S. 21-41
- Ehlich, Konrad / Rehbein, Jochen (1979b)**  
 Sprachliche Handlungsmuster  
 In: Soeffner, Georg (Hg.)  
 Interpretative Verfahren in den Sozial- und Textwissenschaften, S. 243-274  
 Stuttgart: Metzler
- Ehlich, Konrad / Switalla, Bernd (1976)**  
 Transkriptionssysteme - Eine exemplarische Übersicht  
 In: Studium Linguistik, 2, S. 78-105
- Fabricius-Hansen, Cathrine / Gallmann, Peter / Eisenberg, Peter / Fiehler, Reinhard / Peters, Jörg / Nübling, Damaris / Fritz, Thomas / Barz, Irmhild (2006)**  
 Grammatik der deutschen Gegenwartssprache: Die Grammatik  
 Mannheim: Bibliographisches Institut
- Fiehler, Reinhard / Barden, Birgit / Elstermann, Mechthild / Kraft, Barbara (2004)**  
 Eigenschaften gesprochener Sprache. Theoretische und empirische Untersuchungen zur Spezifik mündlicher Kommunikation  
 Tübingen Narr.
- Fillmore, Charles J (1997)**  
 Lectures on deixis  
 Stanford: CSLI Publications
- Forsyth, Richard / Clarke, David / Lam, Phoenix (2008)**  
 Timelines, talk and transcription: A chronometric approach to simultaneous speech  
 In: International Journal of Corpus Linguistics, 13, S. 225-250(26)  
 Verfügbar:  
<http://www.ingentaconnect.com/content/jbp/ijcl/2008/00000013/00000002/art00004>

- Genevès, Pierre (2004)**  
 Improving Efficiency of XPath-Based XML Querying  
 In: IFIP Student Forum , S. 143-154
- Gibbon, Dafydd (1997)**  
 Gesprochene Sprache (Seminar-Reader)  
 Verfügbar: <http://coral.lili.uni-bielefeld.de/Classes/Summer97/ExpPhon/gibbonchap2eps/>
- Gibbon, Dafydd (1997b)**  
 Handbook of standards and resources for spoken language systems  
 Berlin/New York: Mouton de Gruyter
- Gibbon, Dafydd / Mertins, Inge / Moore, Roger K (2000)**  
 Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation  
 Springer
- Grice, Herbert Paul (1975)**  
 Logic and Conversation  
 In: Cole, Peter / Morgan, James .L. (Hg.)  
 Syntax and Semantics, vol. 3: Speech Acts, S. 41-58  
 New York: Academic Press
- Grosso, Paul / Maler, Eve / Marsh, Jonathan / Walsh, Norman (2003)**  
 XPointer Framework  
 W3C  
 Verfügbar: <http://www.w3.org/TR/xptr-framework/>
- Heggelund, Kjell T. (2001)**  
 Zur Bedeutung der deutschen Modalpartikeln in Gesprächen unter besonderer Berücksichtigung der Sprechakttheorie und der DaF-Perspektive  
 In: Hentschel, Elke (Hg.)  
 Linguistik Online, 9  
 Verfügbar: [http://www.linguistik-online.de/9\\_01/Heggelund.html](http://www.linguistik-online.de/9_01/Heggelund.html)
- Henne, Helmut / Rehbock, Helmut (2001)**  
 Einführung in die Gesprächsanalyse  
 Berlin: De Gruyter
- Hennig, Mathilde (2006)**  
 Grammatik der gesprochenen Sprache in Theorie und Praxis  
 Kassel: Kassel University Press  
 Verfügbar: <http://books.google.de/books?id=ErRosMpSIVQC>
- Hilbert, Mirco / Schonefeld, Oliver / Witt, Andreas (2005)**  
 Making CONCUR work  
 In: Proceedings of Extreme Markup Languages  
 Montreal

- Höder, Steffen / Wörner, Kai / Zeevaert, Ludger (2007)**  
 Quantitative analyses of diachronic developments of the function of subordinators. The case of Old Nordic word order  
 In: Arbeiten zur Mehrsprachigkeit, Folge B 81  
 Hamburg: SFB 538  
 Verfügbar: <http://www.exmaralda.org/files/azm81.pdf>.
- Huitfeldt, Claus / Sperberg-McQueen, C M (2008)**  
 What is transcription?  
 In: Literary & Linguist Computing, 23, S. 295-310
- ISO/TC37/SC4 (2006)**  
 Language resource management - Feature structures - Part 1: Feature structure representation  
 ISO
- ISO/TC37/SC4 (2008)**  
 Language resource management - Linguistic Annotation Framework (WD24612)  
 ISO
- Ide, Nancy (2007)**  
 Annotation Science: From Theory to Practice and Use  
 In: Data Structures for Linguistic Resources and Applications - Proceedings of the Biennial GLDV Conference 2007
- Ide, Nancy / Romary, Laurent (2003)**  
 Outline of the International Standard Linguistic Annotation Framework  
 In: Proceedings of ACL'03 Workshop on Linguistic Annotation: Getting the Model right, S. 1-5  
 Morristown  
 Verfügbar: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.1527>
- Ide, Nancy / Romary, Laurent (2002)**  
 Standards for Language Resources  
 In; Proceedings of the IRCS Workshop on Linguistic Databases, S. 141-149  
 Verfügbar:  
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.9903>
- Ide, Nancy / Suderman, Keith (2007)**  
 GrAF: A Graph-based Format for Linguistic Annotations  
 In: Linguistic Annotation Workshop, Prague  
 Verfügbar:  
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.146.4543>
- Ide, Nancy / Suderman, Keith (2008)**  
 XCES Website  
 Verfügbar: <http://www.xces.org/>

- Johansson, Stig (1995)**  
 The approach of the Text Encoding Initiative to the encoding of spoken discourse  
 In: Spoken English on Computer: Transcription, Markup and Application, S. 82-98  
 Harlow: Longman.
- Kallmeyer, Werner / Klein, Wolfgang / Meyer-Hermann, Reinhard / Netzer, Klaus / Siebert, Hans Jürgen (1974)**  
 Lektürekolleg zur Textlinguistik, Vol. 1  
 Frankfurt am Main: Athenaeum
- Kalmbach, G (1996)**  
 Der Dialog im Spannungsfeld von Schriftlichkeit und Mündlichkeit  
 Tübingen: Niemeyer
- Keller, Rudi (1999)**  
 Is the German Language Going to the Dogs?  
 Lecture held at University of California  
 Davis (UCD)  
 Verfügbar: [www.phil-fak.uni-duesseldorf.de/uploads/media/Language\\_Decay\\_01.pdf](http://www.phil-fak.uni-duesseldorf.de/uploads/media/Language_Decay_01.pdf)
- Klemm, Michael (2002)**  
 Ausgangspunkte: Jedem sein Textbegriff? Textdefinitionen im Vergleich  
 In: Fix, Ulla / Adamzik, Kirsten / Antos, Gerd / Klemm, Michael (Hg.)  
 Brauchen wir einen neuen Textbegriff? S. 7-29  
 Frankfurt: Peter Lang
- Koch, Peter / Oesterreicher, Wulf (1985)**  
 Sprache der Nähe - Sprache der Distanz  
 In: Romanistisches Jahrbuch 36 , S. 15-43
- Kunkel-Razum, Kathrin / Münzberg, Franziska (1995)**  
 Duden, die Grammatik  
 Mannheim: Dudenverlag
- Langendoen, D. Terence / Simons, Gary F. (1995)**  
 A rationale for the TEI recommendations for feature-structure markup  
 In: Computers and the Humanities, 29, S. 191-209  
 Dordrecht: Kluwer
- Lehmberg, Timm / Wörner, Kai (2008)**  
 Annotation Standards  
 In: Lüdeling, Anke / Kytö, Merja (Hg.)  
 Corpus Linguistics. An International Handbook, S. 484-501  
 Berlin, New York: Mouton de Gruyter

- Lehnen, Kathrin / Gulich, Elisabeth (1997)**  
Mündliche Verfahren der Verschriftlichung  
In: Zeitschrift für Literaturwissenschaft und Linguistik, 27, S. 108-137  
Verfügbar: <http://dialnet.unirioja.es/servlet/articulo?codigo=2456934>
- Lobin, Henning (2001)**  
Informationsmodellierung in XML und SGML  
Berlin: Springer
- Lüdeling, Anke / Doolittle, Seanna / Hirschmann, Hagen / Schmidt, Karin / Walter, Maik (2008)**  
Das Lernerkorpus Falko  
In: Deutsch als Fremdsprache 2, S. 67-73
- Meyer, Bernd (2004)**  
Dolmetschen im medizinischen Aufklärungsgespräch. Eine diskursanalytische Untersuchung zur Arzt-Patienten-Kommunikation im mehrsprachigen Krankenhaus  
Münster: Waxmann
- Means, W. Scott / Bodie, Michael A. (2002)**  
The Book of Sax: The Simple API for XML  
No Starch Press
- Müller, Christoph / Strube, Michael (2006)**  
Multi-Level Annotation of Linguistic Data with MMAX<sub>2</sub>  
In: Braun, Sabine / Kohn, Kurt / Mukherjee, Joybrato (Hg.)  
Corpus Technology and Language Pedagogy. New Resources, New Tools, New Methods, S. 197-214  
Frankfurt: Peter Lang
- Ochs, Elinor (1979)**  
Transcription as theory  
In: Ochs, Elinor / Schieffelin, B (Hg.)  
Developmental pragmatics, S. 43-72  
New York: Academic Press
- Plaehn, Oliver (1998)**  
ANNOTATE Datenbank-Dokumentation  
Saarbrücken: Universität des Saarlandes  
Verfügbar: <http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/annotate-manual.ps.gz>
- Pollard, Carl Jesse / Sag, Ivan (1994)**  
Head-Driven Phrase Structure Grammar  
Chicago: University of Chicago Press

**Redder, Angelika (2001)**

Aufbau und Gestaltung von Transkriptionssystemen

In: Brinker, Klaus / Antos, Gerd / Heinemann, Wolfgang / Sager, Sven F. (Hg.)

Text- und Gesprächslinguistik. Ein internationales Handbuch zeitgenössischer Forschung, S. 1038-1059

Berlin, New York: Walter de Gruyter

**Rehbein, Jochen (1995)**

Segmentieren

Verbmobil Memo 64

Hamburg: Germanisches Seminar, Universität Hamburg

**Rehbein, Jochen / Grieshaber, Wilhelm / Löning, Petra / Hartung, Marion / Bührig, Kristin (1993)**

Manual für das computergestützte Transkribieren mit dem Programm syncWRITER nach dem Verfahren der Halbinterpretativen Arbeitstranskriptionen (HIAT)

Hamburg: Germanisches Seminar, Universität Hamburg

**Rehbein, Jochen / Schmidt, Thomas / Meyer, Bernd / Watzke, Franziska / Herkenrath, Annette (2004)**

Handbuch für das computergestützte Transkribieren nach HIAT

In: Arbeiten zur Mehrsprachigkeit, Folge B, Vol. 56

Hamburg

Verfügbar: [http://www.exmaralda.org/files/azm\\_56.pdf](http://www.exmaralda.org/files/azm_56.pdf)

**Renear, Allen (1997)**

Target Paper (Edited Version)

In: Theory and Metatheory in the Development of Text Encoding

Monist, 80

**Renear, Allen / Mylonas, Elli / Durand, David (1996)**

Refining our Notion of What Text Really Is: The Problem of Overlapping Hierarchies

In: Research in Humanities Computing

**Rodriguez, Kepa Joseba / Dipper, Stefanie / Götze, Michael / Poesio, Massimo / Riccardi, Giuseppe / Raymond, Christian / Rabiega-Wiśniewska, Joanna (2007)**

Standoff Coordination for Multi-Tool Annotation in a Dialogue Corpus

In: Linguistic Annotation Workshop, Prague

- Rohlfing, Katharina / Loehr, Daniel / Duncan, Susan / Brown, Amanda / Franklin, Amanda / Kimbara, Irene / Milde, Jan-Torsten / Parrill, Fey / Rose, Travis / Schmidt, Thomas / Slotjes, Han / Thies, Alexandra / Wellinghoff, Sandra (2006)**  
Comparison of multimodal annotation tools - workshop report  
In: Deppermann, Arnulf / Hartung, Martin (Hg.)  
Gesprächsforschung Online, 7, S. 99-123
- Rose, Yvan / Hedlund, Greg (2009)**  
PHON Website  
Verfügbar: <http://phon.ling.mun.ca/phontrac/wiki/>
- Rosen, Stuard / Howell, Peter / Bartram, James F. (1993)**  
Signals and Systems for Speech and Hearing  
In: The Journal of the Acoustical Society of America, 94, Nr. 6, S. 3530
- Sagerer, Gerhard (1990)**  
Automatisches Verstehen gesprochener Sprache  
Bibliographisches Institut Wissenschaftsverlag
- Schmid, Helmut (1994)**  
Probabilistic Part-of-Speech Tagging Using Decision Trees  
IMS Stuttgart  
Verfügbar: <http://www.ims.uni-stuttgart.de/ftp/pub/corpora/tree-tagger1.pdf>
- Schmidt, Thomas (2004)**  
Computergestützte Transkription als Modellierung und Visualisierung gesprochener Sprache mit texttechnologischen Mitteln  
Frankfurt: Peter Lang
- Schmidt, Thomas (2002)**  
Stellungnahme zu Wolfgang Schneiders Artikel „Annotate in Transkriptionen aus DV-technischer Sicht“  
In: Deppermann, Arnulf / Hartung, Martin (Hg.)  
Gesprächsforschung Online, 3, S. 237-249
- Schmidt, Thomas (2007)**  
Transkriptionskonventionen für die computergestützte gesprächsanalytische Transkription  
In: Deppermann, Arnulf / Hartung, Martin (Hg.)  
Gesprächsforschung Online, 8, S. 229-241



- Schmidt, Thomas / Duncan, Susan / Ehmer, Oliver / Hoyt, Jeffrey / Kipp, Michael / Loehr, Dan / Magnusson, Magnus / Rose, Travis / Han, Sloetjes (2008)**  
 An Exchange Format for Multimodal Annotations  
 In: Calzolari, Nicoletta / Choukri, Khalid / Maegaard, Bente / Mariani, Joseph / Odjik, Jan / Piperidis, Stelios / Tapias, Daniel (Hg.)  
 LREC Proceedings  
 Marrakech, Morocco: European Language Resources Association (ELRA)
- Schmidt, Thomas (2008)**  
 A short introduction to EXAKT  
 Hamburg, SFB 538  
 Verfügbar: [http://www.exmaralda.org/files/EXAKT\\_Short\\_Intro.pdf](http://www.exmaralda.org/files/EXAKT_Short_Intro.pdf)
- Schmidt, Thomas / Bennöhr, Jasmine (2008)**  
 Rescuing Legacy Data  
 In: Language Documentation and Conservation, 2, S. 109-129
- Schmitt, Reinhold (2004)**  
 1. Arbeitstreffen zu Fragen der Multimodalität (IDS Mannheim)  
 In: Gesprächsforschung Online, 2004, S. 1-5  
 Verfügbar: <http://www.gespraechsforschung-ozs.de/heft2004/heft2004.htm>
- Schonefeld, Oliver (2008)**  
 An event-centric API for processing concurrent markup  
 In: Proceedings of Balisage: The Markup Conference (2008)
- Schonefeld, Oliver (2005)**  
 Mascarpone und XML-CONCUR - Ein Editor und ein Verarbeitungsmodell für multi-strukturierte Daten  
 Diplomarbeit  
 Technische Fakultät Universität Bielefeld
- Schonefeld, Oliver (2007)**  
 XCONCUR and XCONCUR-CL: A Constraint-Based Approach for the Validation of Concurrent Markup  
 In: Data Structures for Linguistic Resources and Applications - Proceedings of the Biennial GLDV Conference 2007, S. 349-358
- Schonefeld, Oliver / Witt, Andreas (2006)**  
 Towards validation of concurrent markup  
 In: Proceedings of Extreme Markup Languages  
 Montreal
- Selting, Margret / Auer, Peter / Barden, Birgit / Bergmann, Jörg / Couper-Kuhlen, Elizabeth / Günthner, Susanne / Meier, Christoph / Quasthoff, Uta / Schlobinski, Peter / Uhmman, Susanne (1998)**  
 Gesprächsanalytisches Transkriptionssystem (GAT)  
 In: Linguistische Berichte, 173, S. 91-122

- Smith, Nicholas / Hoffmann, Sebastian / Rayson, Paul (2008)**  
Corpus Tools and Methods, Today and Tomorrow: Incorporating  
Linguists' Manual Annotations  
In: Literary and Linguistic Computing, Vol. 23, N, S. 163-180
- Sperberg-McQueen, Michael / Huitfeldt, Claus (1999)**  
Concurrent document hierarchies in MECS and SGML  
In: Literary and Linguistic Computing, 14, Nr. 1, S. 29-42
- Stein, Stephan (2003)**  
Textgliederung: Einheitenbildung im geschriebenen und gesprochenen  
deutsch: Theorie und Empirie  
Walter de Gruyter
- Söll, Ludwig / Hausmann, Franz Josef (1985)**  
Gesprochenes und geschriebenes Französisch  
In: Grundlagen der Romanistik, Vol. 6  
Berlin: Erich Schmidt
- Text Encoding Initiative Consortium (2001)**  
Guidelines for Electronic Text Encoding and Interchange (TEI P4)  
Text Encoding Initiative  
Verfügbar: <http://www.tei-c.org/Guidelines/P4/>
- Text Encoding Initiative (2009)**  
TEI: History  
Text Encoding Initiative  
Verfügbar: <http://www.tei-c.org/About/history.xml>
- Tognini-Bonelli, Elena (2001)**  
Corpus Linguistics at Work  
Amsterdam: Benjamins.
- W3C (2001)**  
XML Linking Language (XLink) Version 1.0  
W3C  
Verfügbar: <http://www.w3.org/TR/2001/REC-xlink-20010627/>
- Wahlster, Wolfgang (1997)**  
VERBMOBIL - Erkennung, Analyse, Transfer, Generierung und Synthese  
von Spontansprache In: Informatik, 97, S. 215-224
- Witt, Andreas (2002)**  
Multiple Informationsstrukturierung mit Auszeichnungssprachen. XML-  
basierte Methoden und deren Nutzen für die Sprachtechnologie.  
Dissertation  
Universität Bielefeld

- Witt, Andreas (2004)**  
Multiple hierarchies: new aspects of an old solution  
In: Proceedings of Extreme Markup Languages  
Montreal
- Witt, Andreas (1998)**  
TEI-based XML-applications: Transcriptions  
In: Joint Conference of the ALLC and ACH  
Debrecen
- Witt, Andreas / Rehm, Georg / Hinrichs, Erhard / Lehmberg, Timm / Stegmann, Jens (2009)**  
Sustainability of linguistic resources through feature structures  
In: Literary and Linguistic Computing
- Witt, Andreas / Schonefeld, Oliver / Rehm, Georg / Khoo, Jonathan / Evang, Kilian (2007)**  
On the Lossless Transformation of Single-File, Multi-Layer Annotations into Multi-Rooted Trees  
In: Proceedings of Extreme Markup Languages  
Montreal
- Wittenburg, Peter / Brugman, Hennie / Russel, Albert / Klassmann, Alex / Sloetjes, Han (2006)**  
ELAN: a Professional Framework for Multimodality Research  
In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)  
Genoa
- Wörner, Kai / Witt, Andreas / Rehm, Georg / Dipper, Stefanie (2006)**  
Modelling Linguistic Data Structures  
In: Proceedings of Extreme Markup Languages  
Montreal  
Verfügbar: [http://www.sfb632.uni-potsdam.de/publications/D1/D1\\_EML2006Witto1.pdf&oi=gpp](http://www.sfb632.uni-potsdam.de/publications/D1/D1_EML2006Witto1.pdf&oi=gpp)

*[Alle URLs gültig am 24.03.2010]*

## 7.3 Abbildungsverzeichnis

Abbildung 1: Eigenschaften von Rede und Schrift.	14
Abbildung 2: Lineare, parallele Signalströme mit Überlappungen.	20
Abbildung 3: Abbruch.	23
Abbildung 4: Aposiopese.	24
Abbildung 5: Ellipsen.	25
Abbildung 6: „Illokutives weil“.	26
Abbildung 7: Rechtsherausstellung.	27
Abbildung 8: Reduktion.	28
Abbildung 9: Spezielle Verwendung von Modalpartikeln.	28
Abbildung 10: Transkription mit Information zu Phonologie und Silbenstruktur.	34
Abbildung 11: Die Methode der Transkription in verschiedenen Zusammenhängen.	37
Abbildung 12: NITE-Annotation mit Frame- und Syntax-Annotation.	45
Abbildung 13: Automatische Berechnung der Intensität eines Audiosignals.	48
Abbildung 14: HIAT- und GAT-Segmentierung.	53
Abbildung 15: Milestones.	59
Abbildung 16: Ein TEI-XML-Fragment mit Inline-Annotation.	62
Abbildung 17: Das gleiche Fragment mit der gleichen Annotation in einem Standoff-Format.	62
Abbildung 18: Ungültiger Standoff-Zeiger.	63
Abbildung 19: Der Satz von Sprecher 0 wird von Sprecher 1 zu Ende geführt.	68
Abbildung 20: XCONCUR-Markup.	68
Abbildung 21: XCONCUR-Element mit Zeitachsen-Verweisen.	69
Abbildung 22: NOM-Annotationsbeispiel.	73
Abbildung 23: Fensteranordnung zur Gestenannotation in NXT.	75
Abbildung 24: Wiedergabefenster mit Gestenannotation.	76
Abbildung 25: Suchergebnisse im NXT Suchfenster.	77
Abbildung 26: Gewurzelter Baum.	80
Abbildung 27: Gerichteter, azyklischer Graph.	80
Abbildung 28: MultiTrans aus dem Annotation Graph Toolkit.	85
Abbildung 29: TreeTrans für hierarchische Annotation auf der Basis von Annotation Graphs.	86
Abbildung 30: Ein Transkriptionsgraph mit allen Kantentypen.	91
Abbildung 31: Eventgrenzen erscheinen in der Partiturdarstellung als horizontale (Spalten-) Linien.	92
Abbildung 32: Eventgrenzen werden in der Basistranskription zu Knoten des Transkriptionsgraphen.	92
Abbildung 33: XML-Darstellung (ohne Metadaten) des Beispiels.	93
Abbildung 34: Transkriptionsgraph einer segmentierten Transkription nach HIAT.	93
Abbildung 35: Segmentierte Transkription mit Teilgraphen.	94
Abbildung 36: „Conversion Info“ zur Rekonstruktion einer Basistranskription.	95

Abbildung 37: Partitur-Editor mit Wellenform-Display.	96
Abbildung 38: Im- und Exportformate des Partitur-Editors.	97
Abbildung 39: Annotationsspuren erlauben die Annotation bestehender Segmente im Partitur-Editor.	97
Abbildung 40: Corpus Manager (Coma).	98
Abbildung 41: Das EXMARaLDA Analyse- und Konkordanz-Tool (EXAKT).	98
Abbildung 42: Beispiel für feature structures (hier in TEI-Schreibweise).	103
Abbildung 43: Die Architektur des LAF.	104
Abbildung 44: Beispiel einer PAULA-Annotation.	106
Abbildung 45: Korpusuche mit EXAKT.	112
Abbildung 46: Konfigurationsdialog einer Annotationsspur im Partitur-Editor.	117
Abbildung 47: Annotation „erregt“ in einer Annotationsspur.	118
Abbildung 48: Von anderen Abschnitten unabhängige Annotation.	118
Abbildung 49: Virtuelle Zeitachse.	118
Abbildung 50: Beispiel einer reichhaltigen Annotation im Partitur-Editor.	119
Abbildung 51: Annotationspanel mit geladener Annotationsspezifikation.	122
Abbildung 52: Das EXMARaLDA-Suchwerkzeug „EXAKT“ mit Suchergebnissen.	124
Abbildung 53: Hinzufügen von Analysespalten zur Konkordanz.	125
Abbildung 54: Suchtreffer in der Konkordanz.	126
Abbildung 55: Suchergebnis in einer Konkordanzdatei.	127
Abbildung 56: EXMARaLDA Exportformate.	128
Abbildung 57: Export und Reimport in EXMARaLDA.	128
Abbildung 58: Ausgabe eines getaggtten Beispielsatzes.	131
Abbildung 59: Online-Interface für den TreeTagger.	131
Abbildung 60: Ausgabe der Online-Version des Stanford-Parser.	132
Abbildung 61: Baumdarstellung eines getaggtten und geparsten Satzes.	132
Abbildung 62: Ausschnitt aus einem teilweise POS-annotierten altschwedischen TEI-Text.	133
Abbildung 63: Der Oxygen-XML-Editor bietet CSS-basierte Eingabehilfen zum Beispiel für TEI.	134
Abbildung 64: Der Z2-Tagger erleichtert die Annotation TEI-konformer Texte.	134
Abbildung 65: Eine Annotation im ATLAS-interchange-Format.	135
Abbildung 66: Das Annotationswerkzeug „Callisto“.	135
Abbildung 67: Partiturdarstellung in ELAN.	137
Abbildung 68: Auswahl eines „Benutzerdefinierten Wörterbuches“ in ELAN.	138
Abbildung 69: Auch bei der Suche wird in ELAN das kontrollierte Vokabularium berücksichtigt.	139
Abbildung 70: Eine Praat-Transkription („Textgrid“).	139
Abbildung 71: Graph einer nach HIAT segmentierten Transkription.	145
Abbildung 72: Für die Annotation relevanter Teilgraph für einen Sprecher.	146
Abbildung 73: POS- und Lemma-Annotation in einer Feature Structure.	148
Abbildung 74: Der Ausschnitt aus der segmentierten Transkription.	149

Abbildung 75: POS-Annotation und Lemmatisierung des selben Abschnittes.	150
Abbildung 76: Der annotierte Abschnitt des selben Beispiels.	151
Abbildung 77: Extended Link auf die Konstituenten des Lemmas „wehtun“.	151
Abbildung 78: Annotation direkt auf der Timeline.	152
Abbildung 79: Annotation eines Abschnittes, der keinem existierenden Segment entspricht.	153
Abbildung 80: XLink mit Character-Offset.	153
Abbildung 81: Segmentierung mittels einer endlichen Maschine.	154
Abbildung 82: Auswahl des Segmentierungsalgorithmus im Partitur-Editor seit Version 1.4.	155
Abbildung 83: Mit einem modifizierten Verfahren blieben die Ids von Kanten einheitlich.	157
Abbildung 84: XML-Schema für Sextant-Annotationen.	159
Abbildung 85: Die Sextant-Benutzeroberfläche.	161
Abbildung 86: Sextant-Webseite.	162
Abbildung 87: Validierung einer Standoff-Annotationsdatei.	163
Abbildung 88: Fehler bei der Validierung der Zeiger auf die Transkription.	163
Abbildung 89: EXAKT-Konkordanzen werden in EXMARaLDA-Standoff-Annotation überführt.	164
Abbildung 90: Manuelle POS-Annotation des Beispiels im EXAKT-Suchwerkzeug.	165
Abbildung 91: Eine Zeile der Konkordanz im XML-Markup.	166
Abbildung 92: Ergebnis-Segment der XPath-Suche.	167
Abbildung 93: Sextant generiert Standoff-Annotationsdateien aus EXAKT-Konkordanzen.	168
Abbildung 94: Einfache Visualisierung einer Transkription mit ihrer Standoff-Annotation.	169
Abbildung 95: Eine modifizierte Version des Z2-Taggers für TEI-Texte.	171
Abbildung 96: Anzeige der POS-Annotation mittels Mouseover-Popups in HTML.	172
Abbildung 97: Linearisierung von einzelnen, segmentierten Spuren für POS-Tagger.	173
Abbildung 98: Browsing-Mode zur Annotation in EXAKT.	174

## 7.4. Index

AGLib	84	Fragmentation of Elements	60
AGTK	84	GAT	51, 189
AIF	87	Gebärdensprache	15
ANNIS	106	Gerichtete, azyklische Graphen	80
Annotation	5, 6	Gespräch	13
Annotation Graphs	82	Gesprächsforschung	35
Annotation Science	6, 185	gesprochene Sprache	13
Annotationsspuren	117, 137	Gestenannotation	75
ATLAS	82	grafischer Kode	14
Audioanalyse	139	Graphie	15
Äußerung(HIAT)	50	H3	189
Austauschbarkeit	128	Herausstellungen	27
Basistranskription (EXMARaLDA)	92	HIAT	50
Bottom-Up (Standards)	113	Kombinierbarkeit (Kanten)	91
CES/XCES	72	Konjunktionen	26
Concurrent Markup of Multiple Hierarchies	60	Konkordanzen	123, 163
Corpus Based	111	Konzeption (Konzeptionsebene)	15
Corpus Driven	111, 125	Korpora	12, 188
Corpus-Manager (Coma)	123, 174	EXMARaLDA	99
Corpus-Manager (EXMARaLDA)	97	NITE	78
CoSi	188	Korpuslinguistik	38
DAG	80	KWIC	123
Data Category Register	103, 138	LAF	101
Datenmodelle	55	Lautbildung	21
Anforderungen	142	Lautung vs. Schreibung	45
Verarbeitbarkeit	109	LDC	82
Deixis	29	Lemmatisierung	130
Demokorpus	188	Linearisierung	172
DiK	188	Linguistic Annotation Framework (LAF)	101
Diskontinuierliche Konstituenten	150	Literarische Umschrift	46
Diskursanalyse	35	MATE	72
Distanzsprache	13	Metadaten	8, 36, 123
ELAN	137	Milestones	59
Ellipsen	24	Modalpartikeln	28
EXAKT (EXMARaLDA)	98, 123, 163, 173	Morphologie	22
EXMARaLDA	5, 88	Multimodalität	22, 36, 137
extended Links	151	Multiple Encodings of the Same Information	60
Falko	189	MultiTrans	84
feature structures	102, 147	Mündlichkeit	13, 18
Flüchtigkeit	20	NITE	71

NOM	71	Sozialforschung	32
Nonverbale Elemente	22	Sprache der Nähe	13
NXT	74	Sprachtechnologie	32
Ordered Hierarchy of Content Objects (OHCO)	55	Sprachwandel	19
Orthografie	136	Spuren	92, 139
Paidus	190	Standards	112
paraverbale Information	20	Standoff-Annotation	61, 105, 110, 146
Partitur-Editor	96, 117, 121	TableTrans	84
part-of-speech/pos-Tagging	130	Tagger	130
PAULA	105, 151	TEI-Guidelines	57, 114
Pausen	90	Tier	92
Performanz	109	Timeline	137, 152
PHON	129	Top-Down (Standards)	114
Phonetik	21	Transkription	31
phonetisches Alphabet	34	Anwendungsbereiche	31
Phonie	15	Funktion	40
phonischer Kode	14	Funktion und Aufgabe	40
phonographisches Prinzip	46	Modellhaftigkeit	42
Phonologie	21, 34	Prozesshaftigkeit	39
Phrasierungseinheit/Intonationsphrase (GAT)	51	Selektivität	38
Praat	139	Theoriehaftigkeit	37
Praat-Skripte	140	Transkriptionsschema	32
Pragmatik	35	Transkriptionsgraphen	88
primärdatenidentisch	61	Transkriptionskonventionen	120, 144
Realisierung (Realisierungsebene)	14	TreeTrans	84
Reconstitution of Virtual Elements	60	Turn (HIAT)	51
Rede	13	Überlappende Annotationen	45
Reduktionen	28	Überlappende Hierarchien	43
Referenzielle Integrität	153	Überlappende Hierarchien (TEI)	59
Rehbein	190	Überlappung	54
relationale Datenbanken	110	Validierung	162
Reparaturen	23	Verdauerung	41
Schrift	13	Visualisierung	168, 171
Schriftlichkeit	13	XCONCUR	67
Schriftsysteme	54	XLink	149
Segmentierung	48, 54, 93, 120, 142	XML-Schema	158
Segmentierbarkeit (Kanten)	91	XPointer	153
Segmentierte Transkription (EXMARaLDA)	93	Z2-Tagger	169
Segmentierung (LAF)	103	Zeitachse	44, 54
Selektivität	38	Zeitbezug	44
Sextant	160	Zeitstruktur	19
SGML	67		