

Standoff-Markup für Editionen und Paratexte

Klaus Prätor

Where have you gone, XML?

Vor ziemlich genau zehn Jahren nahm ich an einer kleinen, aber feinen Tagung zur Textauszeichnung hier am Deutschen Literaturarchiv teil. Auf dem Heimweg kam mir in den Sinn, wie sehr sich die Praxis der Textauszeichnung doch im Lauf der Jahre verändert hatte. Ich fasste die Gedanken in einem Blogbeitrag zusammen und wählte als Titel die Frage - oder auch den Seufzer: Where have you gone, XML?

An XML selbst hat sich eigentlich nichts Wesentliches geändert. Es war und ist mehr oder weniger der gleiche Standard wie ein paar Jahre vorher. Geändert hat sich aber der Gebrauch von XML beziehungsweise von Markup generell. Dem Vorläufer SGML ging es zunächst um das sachliche Markup, das DIE sachliche Textstruktur unabhängig von jeder grafischen Gestaltung festhalten wollte. Mit XML gab es dann die Möglichkeit, eigene Tags zu definieren und damit weitergehende, im Prinzip beliebige Auszeichnungen vorzunehmen. Davon wurde auch reichlich Gebrauch gemacht. Es wurden Tags geschaffen für die verschiedenen Elemente eines kritischen Apparates, für Kommentare, Metadaten, Personen- und Ortsnamen. Auch grafische Gestaltungselemente des Originaldokuments wie Schriftart oder Zeilenfall werden in Editionen teilweise festgehalten. Es entstanden Auszeichnungssysteme z.B. für linguistische, literaturwissenschaftliche oder historische Zwecke.

```
<form><form type="lemma">
<gramGrp><pos value="noun"/><gen value="m"><gramGrp>
<form type="determiner"><orth>Der</orth></form>
<form type="headword" norm="aal"> <orth>Aal</orth>
</form><pc>,</pc>4</form><form type="inflected">
<gramGrp><case value="genitiv"/><number
value="singular"/></gramGrp></form>
<form type="determiner"><orth>des</orth></form>
<form type="headword">
<oVar><oRev><orth> - es</c>,</c></orth>
<oRev/><oVar/></form><pc>,</pc><form
type="inflected">
<gramGrp><case value="nominative"/> cnumber
value="plural">Mz.</cnumber></gramGrp>
<form type="determiner"><orth>die</orth></form>
<form type="headword"><orth><oVa rxoRef/> - e</oVa
r></orth></form><pc>,</pc></form>
</form>
```

Abbildung 1

Markup überwuchert den Text

Die Quo-Vadis-Frage ließ schon anklingen, dass diese Entwicklungen des Markup nicht nur als Fortschritt gesehen werden können. Ein erster und augenfälliger Effekt ist, dass der relative Anteil des Markups gegenüber dem Originaltext wächst, in nicht wenigen Fällen ihn um das Mehrfache übertrifft. Der Beispieltex in Abbildung 1, aus einem Wörterbuch von Campe, der noch nicht einmal vielfältigen Auszeichnungsinteressen dient, sondern nur mit linguistischen Tags versehen wurde, macht das deutlich. Nur die roten Stellen sind originaler Text, der Rest ist Markup.

Ein Markup-Anhänger würde beruhigen und darauf verweisen, dass solche Texte ja nicht zum Lesen, sondern für die Verarbeitung durch den Computer gedacht sind, der damit durchaus umgehen kann, daraus eine komfortable Leseversion erzeugt und den Editor durch geeignete Werkzeuge beim Umgang mit dem Markup unterstützt.

Ich kann aber schon verstehen, wenn es Editoren etwas bange wird angesichts derartigen Überwucherns des Originaltextes, dem ja ihre ganze Sorge gilt. Mit zunehmender Datenmenge und Komplexität steht zu befürchten, dass auch die Fehleranfälligkeit zunimmt.

Auch kann zunehmend nicht mehr davon ausgegangen werden, dass die Annotation eine einmalige abgeschlossene Handlung ist. Mir der Zunahme der Annotationsmöglichkeiten und -interessen wird sie zu einer dynamischen, prinzipiell nie abgeschlossenen Entwicklung.

Overlapping als ein Grundproblem von XML

Die problemlose Verarbeitung des Markups durch den Computer bzw. die Software ist auch nicht die ganze Wahrheit. Es gibt Probleme, die grundsätzlich mit dem XML-Modell zusammenhängen, die aber durch die Ausweitung des Markups und insbesondere durch unterschiedliche Auszeichnungsinteressen deutlich verschärft werden.

Das bekannteste und auch in Forschung und Literatur am meisten behandelte Problem ist als Overlapping bekannt. Das liegt vor, wenn ein Element nicht ganz in einem übergeordneten liegt, sondern z.B. in einem solchen beginnt, aber erst außerhalb von ihm endet.

Man kann sich das am Beispiel eines literarischen Textes verdeutlichen, in dem sowohl Zeilen wie Sätze ausgezeichnet werden sollen

```
`<line><sentence>I, by attorney, bless thee from thy mother,</line>`  
`  <line>Who prays continually for Richmond's good.</sentence></line>`  
`  <line><sentence>So much for that.</sentence><sentence>The silent hours steal  
    on,</line>`  
`  <line>And flaky darkness breaks within the east.</sentence></line>`
```

Sätze und Zeilen verschränken oder überlappen sich. Der Auszeichnung liegen zwei verschiedene Hierarchien zugrunde, die sich überschneiden.

Overlapping ist in XML nicht zugelassen. Das angegebene Beispiel würde so nicht zur Verarbeitung akzeptiert werden. Natürlich hat man sich über Auswege Gedanken gemacht und hat mehrere gefunden. Ein prominentes ist der Einsatz von Milestones. Eine der beiden Hierarchien (hier die der Sätze) wird nicht als normales XML formatiert, sondern durch ein leeres Tag am Anfang (sentence-start) und ein ebensolches am Ende (sentence-end) gekennzeichnet.

```
<line><sentence-start />I, by attorney, bless thee from thy mother,</line>  
<line>Who prays continually for Richmond's good.<sentence-end /></line>  
<line><sentence-start />So much for that.<sentence-end /><sentence-start /> -The  
    silent hours steal on,</line>  
<line>And flaky darkness breaks within the east.<sentence-end /></line>
```

Da leere Elemente in XML zulässig sind und jetzt keine Überschneidung mehr vorliegt, wird dieser Code von einem entsprechenden Programm ohne Probleme verarbeitet. Und doch bleiben Milestones nur ein Workaround oder auch ein Notbehelf. Eine wirklich gute Lösung sind sie nicht. Die Sätze sind jetzt nämlich kein XML-Element mehr und können daher nicht mit XML-Tools sondern allenfalls durch zusätzliche Programmierung verarbeitet werden. Und falls es um mehr als zwei sich überschneidende Hierarchien geht, wird der Großteil der Textauszeichnung kein wirkliches XML mehr sein.

Es handelt sich hier nicht um zufällige kleine Unzulänglichkeiten, die man durch geschickte Programmierung umgeht kann. Hinter XML steht ein (mono)hierarchisches Datenmodell und eine entsprechende Grammatik.

Eine Grammatik erzeugt über einem Text, der ja selber linear ist, eine Baumstruktur mit genau einer Wurzel. Wie eine Grammatik einen Satz in Subjekt, Prädikat usw. gliedert, kann man sich auch eine Grammatik für die Gliederung eines Buches vorstellen. Hier gibt es dann Titel und Autoren, mehrere Kapitel, und die können dann Paragraphen haben und so weiter. Es entsteht wieder eine Baumstruktur mit einer Wurzel. Genügt ein Text samt Auszeichnungen aber diesen Bedingungen nicht, macht seine Handhabung im XML-Modell Probleme.

Standoff-Markup als Ausweg

Einen grundsätzlichen Ausweg aus dem Problem des Overlapping, aber auch aus dem der Überlastung des Textes mit Markup bietet Standoff-Markup. Standoff-Markup sind Annotationen, die sich außerhalb des Grundtextes in einer eigenen Datei befinden. Man hat zum Beispiel einen Kommentar, der auf die entsprechende Stelle (oder den Bereich) verweist, zu der er gehört. Ganz offensichtlich entlastet und stabilisiert es den Grundtext. Es steht weniger Markup im Grundtext und mit weniger Markup wird er weniger fehleranfällig und damit stabiler. Und es vermeidet, wie wir noch sehen werden, auch das Overlapping-Problem oder allgemeiner gesagt die Probleme, die sich aus dem monohierarchischen Datenmodell von XML ergeben und die durch sich ausweitendes Markup verschärft werden.

Standoff hat aber noch mehr Vorteile: Es ermöglicht domainspezifische oder individuelle Annotationen zu einem Text. So lassen sich beispielsweise linguistische Auszeichnungen an einen literarischen Text anfügen, ohne dass in diesen Text selbst eingegriffen werden muss. Parallel sind auch anderer Auszeichnungsinteressen realisierbar, die sich unabhängig voneinander bearbeiten lassen und zwischen denen sich auch keine Überschneidungen ergeben. Und das gleiche ist auch individuell möglich. Jemand kann sich standoff nach Bedarf eigene Annotationen zu einem Dokument erstellen, ohne dass er in den Originaltext oder in andere Annotationen eingreift.

Und das geht auch in die umgekehrte Richtung: wenn es aus irgendwelchen Gründen erforderlich oder sinnvoll ist, für verschiedene Texte einen gemeinsamen Zugang zu schaffen, dann kann dieses Standoff-Markup auch, so wie es in einen Text verweist, in mehrere Texte verweisen.

Standoff braucht Referenz im Text

Für all das braucht man allerdings eine Möglichkeit, in die Texte zu verweisen. Man benötigt eine eindeutige und beständige Referenz im Grundtext. Die Referenz in digitalen Texten ist ein grundlegendes, aber keineswegs allgemein zufriedenstellend gelöstes Problem.

Keine Referenz nach Seite und Zeile

Noch immer wird unter Editoren häufig die Meinung vertreten, dass wissenschaftlich nach Seiten und Zeilen zitiert wird und deshalb auch digitale Texte nach Seiten und Zeilen gegliedert werden müssen. Diese künstliche Einengung ist aber den Möglichkeiten des neuen Mediums in keiner Weise angemessen. Auch kann es nicht

sinnvoll sein, allgemein das Prinzip der sachlichen Auszeichnung zu vertreten und dann für die Referenz auf eine äußerliche grafische Eigenschaft Bezug zu nehmen, noch dazu eine, die noch nicht einmal so gewollt ist, denn Seiten- und Zeilenfall im Druck ergeben sich zufällig.

Manuskript Varianten Sachkomm GenKomm



Satiren und Ironien

75

Holbeins gemalte Beine ⁸ ₁₉. Basel steht noch, wo Holbein ein Haus anstrich, wie ich mir vorlesen lassen (oder sagen lassen vom Vorleser). Nun war die nämliche Ursache, die ihn zur architekt. Malerei hinzwang, auch ein Hindernis daß er wenig in ihr arbeitete - er sof nämlich. Wenn der Bauher ein neues Stük Arbeit zu sehen kam: so war er blos über d. Alten und bewegte stat des Pinsels den Weinkrug. Damit man sehe, ich sei nicht partheüssch eingenommen gegen I [48] den Bauhern und steckt ihn in Hintergrund: so sol er hervor und den Holbein -< über nach Vermögen >- aushunzen. Dem Holbein wars nicht schwer herauszubringen, daß das alles davon käme, weil der Entrepreneur nicht seine Beine vom Gerüst hängen sähe. Er verdoppelte also sein Gestelle, Beine und malte sie so an die Wand, daß der Herr bis ietzt in die Ewigkeit, denken würde, die 2 Beine hätten oben gemalet, wenn er, da er den Tag 10 mal hinauf sah, bemerkt hätte, daß die Beine weiter sich schöben. Da sie aber hängen blieben: so wolte er nachsehen was der Maler so lange an Einer Stelle male und stieg hinauf zu Holbeins (Fusstük) -< Kniestük >- 2 malenden Beinen. Der Bauherr hätte daraus eine neue, satirische Moral zu ziehen suchen sollen und die Geschichte würde die Moral mit gemeldet haben; aber sie kan blos von meiner melden,

Abbildung 2

Eine denkbare sachliche Basis für die Referenz könnte der DOM-tree sein. Das ist der Baum der gesamten Auszeichnungen eines XML-Dokuments. Unser Ausgangsproblem war aber gerade, dass wir in eine Zeit gekommen sind, in der sich die Auszeichnungen auch immer ändern und weiterentwickeln können. Man zieht sich dann sozusagen selber den Boden unter den Füßen fort, weil das, was als Referenz genommen werden soll, keinen festen Bestand hat.

Mein Vorschlag ist, stattdessen sinntragende Einheiten wie Sätze, Wörter oder Paragraphen als Grundlage der Referenz zu nehmen. Auch in der Druckwelt gibt es schon Beispiele für dieses Vorgehen. So ist es bei Gesetzen oder in der Bibel nicht üblich, nach Seite und Zeile zu zitieren, sondern nach sinntragenden Einheiten.

Wie eine solche Referenz in den Text realisiert werden kann, zeige ich an einem Beispiel aus der Jean-Paul-Edition in Würzburg. Im gezeigten Text sind die Bereiche markiert, zu denen es Varianten gibt. Die Varianten selbst stehen auf einer zweiten Seite, auf der auch das Manuskript, kein Sachkommentar oder der genetische Kommentar gezeigt werden kann. Damit ändert sich aber auch jeweils diese (erste) Seite. Alle diese Ansichten werden aus einer zugrundeliegenden XML-Datei erzeugt.

Uns interessiert an dieser Edition im Moment aber nur die dort eingebaute Referenzmöglichkeit. Die sinntragenden Einheiten sind hier die Wörter in einem (schon von Jean Paul so gegliederten) Absatz. Absatz- und Wortnummer werden durch Punkt getrennt angegeben, also z.B. 287.43

```
<w title="287.43/150,6">arbeitet<i>e</i></w>
<w title="287.44/150,6">-</w>
<w title="287.45/150,6">er</w>
<w title="287.46/150,6">sof</w>
<span class="var"><w title="287.47/150,6">näml<i>ich</i>.</w></span>
<w title="287.48/150,6">W<i>enn</i></w>
<w title="287.49/150,6">d<i>er</i></w>
<w title="287.50/150,6">Bauher</w>
<w title="287.51/150,6">e<i>in</i></w>
```

Es handelt sich um eine Zeile aus dem 287. Abschnitt von Jean Pauls Satiren und Ironien. Innerhalb dessen wird auf Wortebene ausgezeichnet und Wort für Wort hochgezählt. Abschnitte und Wörter als Basis zu wählen, waren bei diesem Jean-Paul-Text naheliegend. Man kann aber je nach Text und Anwendungszweck auch Sätze oder andere sinntragende Einheiten benutzen.

Mit dieser Art der Referenz ist es möglich, aus dem Text zu zitieren, mit ihm zu arbeiten und eben auch per Standoff-Markup auf ihn Bezug zu nehmen.

Weil das Projekt die Druckfassung selbst erstellt, standen uns auch deren Seiten- und Zeilenzahlen zur Verfügung, die, durch Komma getrennt, als weitere Information eingebaut wurden. Das ist keine Inkonsistenz. Die digitale Version funktioniert ausschließlich auf Basis der sinntragenden Einheiten. Aber die zusätzliche Seiten- und Zeilenangabe erlaubt den parallelen Zugriff auf die gedruckte Ausgabe - auch für Annotationen oder Funktionen, die erst nach der Drucklegung digital zur Verfügung

gestellt werden.

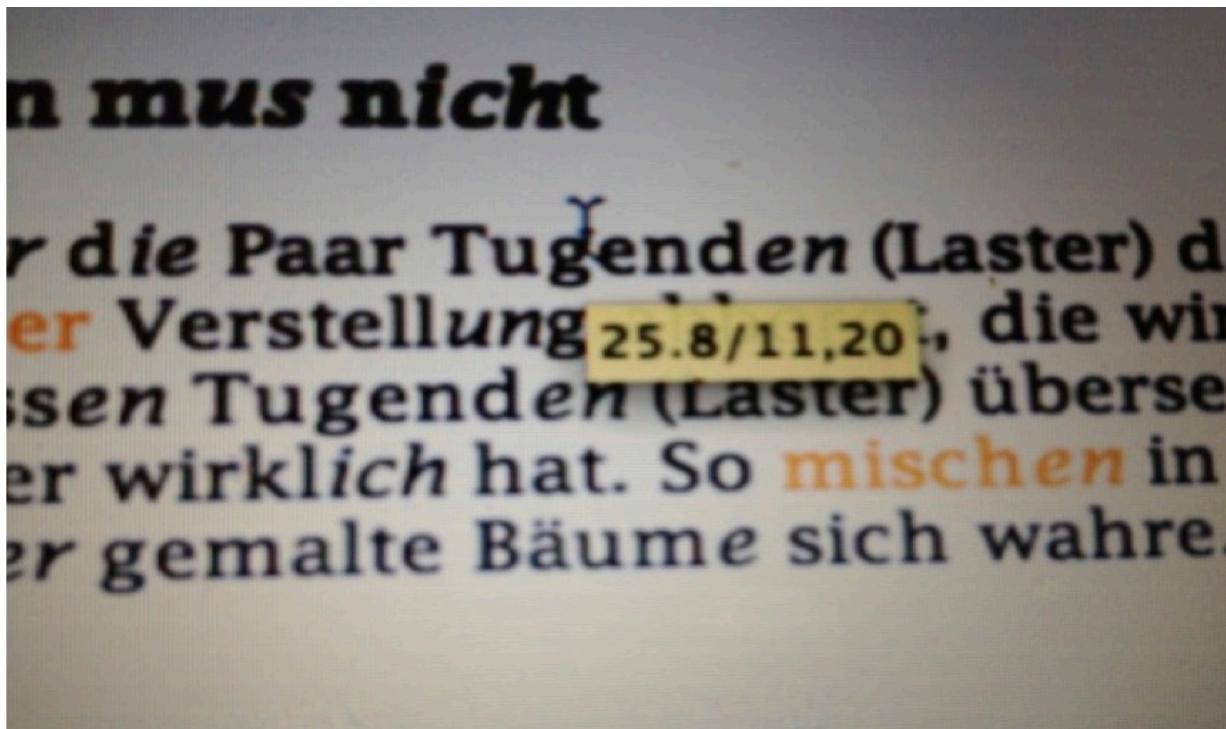


Abbildung 3

Beispielsweise lässt sich relativ einfach eine Retrievalfunktion für digitale und Druckfassung zur Verfügung stellen. Da der Text bis auf Wortebene gegliedert ist, steht eine Liste der Wörter samt Stellenangaben quasi direkt zur Verfügung. Die muss, z.B. wegen grammatischer Varianten, freilich noch weiter bearbeitet werden.

Auch andere Editionswerzeuge können hier anknüpfen. Abbildung 3 zeigt, wie beide Stellenangaben in einem „mouseover text“ erscheinen. Wenn ich mit dem Cursor über den digitalen Text (hier das Wort „Tugenden“) gehe, dann werden die Positionsangaben des entsprechenden Wortes angezeigt. Das ist selbst nicht besonders hilfreich, aber es demonstriert die Möglichkeit, die Positionsinformation in einem (hier Javascript-)Programm abzugreifen und weiter zu verarbeiten. Eine nützlichere Funktion wäre zum Beispiel eine automatisierte Zitatübernahme. Eine markierte Phrase wird als Zitattext kopiert und die Stellenangabe mithilfe der demonstrierten Funktion übernommen. Zusammen mit dem ohnedies bekannten Titel bzw. der Sigle des Dokuments kann man automatisch das gesamte Zitat in einer eigenen Arbeit einsetzen.

Forschungssituation zu Standoff

Der vorgetragene Ansatz und der vorgestellte Prototyp sind eher aus praktischen Bedürfnissen als aus Forschungszusammenhängen hervorgegangen. Der Prototyp zur

Jean-Paul-Edition ist 2012 entstanden, das Konzept noch älter. Aber es macht natürlich Sinn, die eigenen Bemühungen vor dem Hintergrund der aktuellen Forschungslage zu verorten. Das soll im Folgenden versucht werden. Ich stelle zunächst zwei Ansätze aus den letzten Jahren vor, die mir besonders interessant scheinen: Standoff-Attribuierung und Hypergraphen als alternatives Datenmodell zum DOM-tree und setzte sie dann zu dem hier vorgestellten Konzept in Beziehung.

Standoff-Attribuierung

Standoff-Attribuierung kann man als eine radikalere Form des Standoff-Ansatzes als Standoff-Markup verstehen. Im Basistext gibt es überhaupt kein Markup mehr. Wie ist das möglich? Ich muss gestehen, dass ich bei meiner plakativen Gegenüberstellung von Seitenorientierung und sinntragenden Einheiten eine Referenzierungsmöglichkeit überschlagen habe:

Man bezieht sich weder auf Seiten und Zeilen noch auf den DOM-tree und auch nicht auf sinntragende Einheiten, sondern einfach auf die einzelnen Zeichen des Textes. Wenn man die durchzählt, muss man nichts markieren und hat doch eine Bezugsbasis. Das hat etliche Vorteile: zeichenbasierte Referenz braucht keinen Schreibzugriff auf den Basistext und ist daher relativ problemlos handhabbar. Die Handhabung von Textänderungen ist allerdings schwierig, denn jede Verschiebung wirkt sich auf das Ganze aus oder jedenfalls auf alles Nachfolgende. Desmond Schmidt hat Algorithmen entwickelt, womit diese Textänderungen nachführbar sind. Er arbeitet mit Offsets. Man gibt nicht die absolute Stellenposition an, sondern wie die sich relativ zu einem Vorgänger verhält.

Die Verfechter der Standoff-Attribuierung stammen offensichtlich eher aus der Datenbankwelt als aus der Textauszeichnung. In Thaller 2012 wird eine document community gegenüber einer database Community unterschieden. Man muss da keine Gegensätze aufbauen. Texte und Datenbanken haben ihre jeweils guten Anwendungen, aber die Datenbankwelt ist natürlich eine andere als die der Texte. Sie arbeitet nicht mit der Gesamtheit eines Textes, sondern mit flexiblen kleinen vernetzten Einheiten. Diese lassen sich gut als Relationen darstellen, wie sie in relationalen Datenbanken verwendet werden, aber auch in RDF (Resource Description Format), einem Format, das für Standoff, aber auch für viele andere Zwecke, wo es auf die Vernetzung von Informationen ankommt, sehr gut geeignet ist. Es ist im Wesentlichen eine Menge von Tripeln, die sich als Subjekt-Prädikat-Objekt-Strukturen auffassen lassen. RDF schlägt eine Brücke aus den Texten in die Welt von Semantic Web und Linked Data, deren Grundlage es bildet.

Hypergraphen statt Bäume

Die Beschränkungen des XML-Modells hängen wesentlich mit seiner Baumstruktur (DOM-tree) zusammen. In diesem monohierarchischen Modell einschließlich zugehöriger Grammatik gibt es immer nur einen Vorgänger für mehrere Knoten. Ganz oben steht ein einzelner Knoten, die Wurzel. Dazu gibt es jetzt alternative Datenmodellvorschläge. Denkbar sind multihierarchische Modelle, in denen für einen Knoten mehrere Vorgänger erlaubt sind. In der Datenbankwelt entspricht dem der Übergang von hierarchischen zu Netzwerk-Datenbanken. Das würde genügen, um mit dem Overlapping-Problem umzugehen. Aber dieses ist zwar die bekannteste, aber nicht die einzige Schwierigkeit des XML-Datenmodells. Andere sind der Umgang mit Diskontinuitäten oder Referenzbereichen, die Notwendigkeit künstlicher Hierarchien

Datenmodell Hypergraph

nach Dekker / Birnbaum (2017)

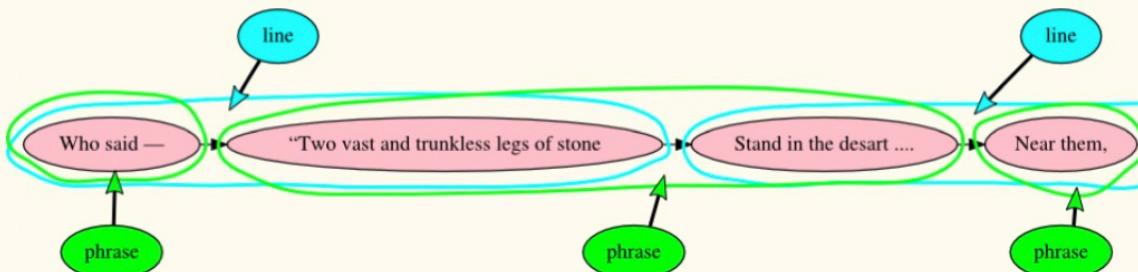


Abbildung 4

und die mangelnde Differenzierung zwischen Hierarchie und Enthaltensein. Darauf kann hier nicht näher eingegangen werden, vgl. dazu z.B. Dekker/Birnbaum 2017.

Hypergraphen gehen einen Schritt weiter und versprechen auch für diese Probleme eine Lösung. In einem normalen Graphen - auch ein Baum ist ja eine Art Graph - verbindet eine Kante immer genau zwei Knoten. In einem Hypergraph geht eine Kante von einem Kopf aus, kann dann aber durch eine ganze Reihe weiterer Knoten laufen. Es verbindet den Kopf also mit einer Menge von Knoten, nicht nur mit einem einzelnen.

In Abbildung 4 wird dieses Datenmodell auf unser Ausgangsproblem des Overlapping zwischen Sätzen und Zeilen angewandt. Phrase und Line verweisen jeweils auf den Kopf eines (möglicherweise mehrgliedrigen) Hypergraphelements, beide können aber problemlos getrennt behandelt werden.

Logikprogrammierung und Grammatiken

Standoff-Attribuierung und Hypergraph-Modell sind in jedem Fall anschlussfähig zu dem hier vorgeschlagenen Ansatz von Standoff-Markup auf der Basis sinntragender Referenz. Darüber hinaus zeigt sich eine hohe Affinität zu dem für den Prototyp im Jean-Paul-Projekt benutzten Programmierparadigma, das durch den Einsatz von Logikprogrammierung und Grammatiken charakterisiert ist. Die Subjekt-Prädikat-Objekt-Elemente von RDF sind auch native Elemente der Logikprogrammierung, da beide auf einer relationalen Struktur aufbauen.

Ein Unterschied zur Standoff-Attribuierung ist der, dass hier nicht eine zeichenbasierte Referenzierung, sondern eine auf Wortbasis gewählt wurde. Grundsätzlich ist natürlich beides möglich. Die Entscheidung hängt vom Anwendungsfeld ab. Im Bereich linguistischer Corpora überwiegen wohl die Vorteile zeichenbasierten Standoff-Markup. Bei einer Edition scheint mir eine Strukturierung nach Satz und Wort näher zu liegen. Sie reduziert die schon oben angesprochene Veränderungsproblematik, die bei den Corpora nicht gegeben ist. Nicht zuletzt bildet sie auch eine intuitivere Basis für die externe Zitierung.

Auch zu den Hypergraph-Grammatiken gibt es eine Nähe. Sie sind nachgewiesenermaßen äquivalent zu den Mitteln der Logikprogrammierung (Corradini 1991, Zaniolo 2015). Technisch hat das damit zu tun, dass Grammatiken in der Logikprogrammierung rekursiv über Listenkonstruktoren verarbeitet werden, die eine ebensolche Kopf- Restlisten-Struktur haben wie HyperGraphen. Das Programm für die Jean-Paul-Edition wurde als Grammatik geschrieben. Sie läuft über ein Dokument und erzeugt daraus ein anderes.

ankereinzel(HM,RefWo) --> ankereinzellemma(RefWo),anmerkung,
{html2terms(HM,span([class='var'],RefWo))}.

ankereinzellemma(RefWo) --> ankerlemma(RefWo,Vpo),"<AppaEintrag1>".

ankerbereich(HM) --> ankerlemma(A,Vpo),"<AppaEintragBereich1>", anmerkung, nullanmerkung(Vpo), {},lf, ankerrestbereich(B),
{html2terms(HM,span([class='var'],[A,[13],B]))}.
ankerbereich(HM) --> ankerlemma(A,Vpo),"<AppaEintragBereich1>", anmerkungohnespeichern, lf, ankerrestbereich(B),
{html2terms(HM,span([class='var'],[A,[13],B]))}.
%ankerbereichslemma(A) --> ankerlemma(A),"<AppaEintragBereich1>".

Standoff-Markup als Editionsrückgrat

StandOff als „Rückgrat“

Vorarbeiten

[Campe Exzerpte Satiren Hesperus](#)

Textzeuge D1_{Hes} (1795)

Hesperus oder 45 Hundsposttage, Vierter Schalldag und Vorrede zum zweiten Heftlein, Schalt- und Extra-Nebenschöslinge alphabetisch geordnet, Buchstabe H.

Jean Paul: *Hesperus oder 45 Hundsposttage. Eine Biographie.* 3 Bände. Hrsg. von Barbara Hunfeld. In: Ders.: *Werke. Historisch-kritische Ausgabe.* Hrsg. von Helmut Pfotenhauer. Band I. Tübingen 2009. Hier: Band I, I: Erstes Heftlein, S. 444ff.

H.

Holbeins-Bein. Ich will lieber das H noch einmal nehmen als das I, weil unter der Rubrik des I's die Invaliden kämen, [...].

Die Beine Holbeins machen größern Spas als amputirte. Der Mahler strich nämlich in Basel nichts an als Basel selber; und der nämliche Umstand, der sein Genie in diese architektonische Färberei hineinzwang, nöthigte es auch, daß es oft darin Raststunden hielt – er soff nämlich entsetzlich. Ein Bauherr, dessen Name in der Geschichte fehlt, trat oft in die Hausthüre und zankte zum Gerüste hinauf, wenn die Beine des Hausfärbers, anstatt davon herunterzuhängen – denn mehr war vom Mahler nicht zu sehen – in der nächsten Weinkneipe standen und wankten. Schritt nachher Holbein damit über die Gasse daher: so kam

Abbildung 5

Statt diese technischen Details näher auszuführen, will ich die Gelegenheit nutzen, einen weiterführenden Aspekt anzusprechen, der aus einer inhaltlichen Perspektive bereits im Vortrag von Frau Hunfeld zur Jean-Paul-Edition angesprochen wurde und der

zum andern rechtfertigt, Paratexte im Titel dieses Referats zu erwähnen. Editorisch ist Jean Paul vor allem durch seine „Schreibwerkstatt“ interessant, ein Geflecht von Notizen und Texten. Frau Hunfeld beklagte in diesem Zusammenhang, dass bei den üblichen Inline-Annotationen, die parallelen Texte immer nur als Beiwerk zu EINEM Text erscheinen und keine gleichgewichtige Darstellung der verschiedenen Versionen beziehungsweise ein Wechsel zwischen verschiedenen Textperspektiven möglich ist.

\$\$\$ ABBILDUNG 5 Rückgrat

Die Oberfläche in Abbildung 5 präsentiert Textvarianten, die in Abbildung 2 noch als Varianten zu EINEM Haupttext (Satiren und Ironien) gezeigt wurden, in einer gleichgewichtigen Darstellung zwischen Campes Reisebericht und den Versionen Jean Pauls in Exzerten, Satiren und im Hesperus. Wie kann das auf Textauszeichnungsebene realisiert werden? Wir hatten eingangs schon vermerkt, dass Standoff-Markup auch erlaubt, aus einer Standoff-Annotation auf unterschiedliche Dokumente Bezug zu nehmen. Das ist der Schlüssel für unser Problem. Wir tauschen sozusagen die Rollen zwischen Annotation und Grundtext: die Gliederung und Navigation wird in die Annotationsdatei verlegt und ermöglicht damit eine gleichgewichtige Stellung der Textvarianten. Elisa Beshero-Bondar 2019 hat im Rahmen ihrer Arbeiten zur Handhabung unabhängiger Textzeugen dafür den Ausdruck spinal standoff markup geprägt. Der kritische Apparat, ursprünglich ein Anhang des Grundtextes wird nun zum Rückgrat einer Edition, die auch mit mehreren unabhängigen Varianten gleichberechtigt umgehen kann.

LITERATUR

Buzzetti, Thaller (2012), Beyond Embedded Markup (<http://www.dh2012.uni-hamburg.de/conference/programme/abstracts/beyond-embedded-markup.1.html>)

Corradini A. u.a. (1991), Graph grammars and logic programming. In: Ehrig H., Kreowski HJ., Rozenberg G. (eds) Graph Grammars and Their Application to Computer Science. Lecture Notes in Computer Science, vol 532. Springer, Berlin, Heidelberg

R.H.Dekker, D.J. Birnbaum (2017), It's more than just overlap: Text as Graph, BASILAGE markup conference (<https://www.balisage.net/Proceedings/vol19/html/Dekker01/BalisageVol19-Dekker01.html>)

DeRose (2004), Markup Overlap: A Review and a Horse (<http://xml.coverpages.org/DeRoseEML2004.pdf>)

D. Dubin, D. Birnbaum (2004), Interpretation Beyond Markup (<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.86.9643>)

Shkapsky, Zeng, Zaniolo (2015), Graph Queries in a Next-Generation Datalog System (<http://web.cs.ucla.edu/~zaniolo/papers/p1258-shkapsky.pdf>)

Elisa Beshero-Bondar (2019) Document Modeling with the TEI Critical Apparatus (https://gams.uni-graz.at/o:tei2019.150/TEI_SOURCE_slides.com/elisabeshero-bondar/app-crit)