

Strategien zur Herstellung kohäsiv autonomer Modulsichten

Dokumentation

Projekt
**Hypertextualisierung auf
textgrammatischer Grundlage**
(www.hytex.info)

Eva Anna Lenz
Angelika Storrer

2004

Sequentiell organisierte Dokumente, wie z. B. wissenschaftliche Monographien und Artikel, sind für eine Lektüre von Anfang bis zum Ende in der angegebenen Abfolge gedacht. Wenn man sie bei der Hypertextualisierung nach ihrer logischen, hierarchischen Dokumentstruktur in Hypertext-Module segmentiert, enthalten sie häufig noch Kohäsionsmittel, die auf den unmittelbaren oder weiteren Vor- oder Folgetext verweisen. Dazu gehören anaphorische Ausdrücke (Pronomina, anaphorisch gebrauchte NPs) als auch Verweisfloskeln (siehe oben, vgl. Abschnitt 2.3., wie im Folgenden noch gezeigt wird) und Konnektoren.

Das Ziel der Strategien zur Herstellung kohäsiv autonomer Modulsichten ist es, Modulsichten zu erzeugen, in der diese Bezüge entweder aufgelöst (Einblendung der Antezedensien von Anaphern) oder die entsprechenden Passagen durch Linking zugänglich gemacht werden, so dass der Zusammenhang erschließbar ist.

Eine Modulsicht ist eine „virtuelle“ Umarbeitung der sequenziellen Textpassage, die als Hilfestellung für den selektiv browsenden Nutzer generiert wird; der zugrunde liegende Originaltext wird nicht irreversibel verändert, d. h. bleibt als eine mögliche Sicht erhalten. Die automatische Herstellung kohäsiv autonomer Modulsichten operiert über die textgrammatische Annotationsschicht (Kohäsionsmittel und Koreferenz/Kospezifikation). Dabei gibt es vier Grundoperationen, die in Abhängigkeit vom Kohäsionsmitteltyp und seinem Kontext angewendet werden:

1. Die **Einblendung** anaphorischer Ausdrücke durch ihren Bezugsausdruck (Antezedens). Dies wird in der Pilotversion technisch realisiert durch **Linktitel**.
2. Die Umwandlung von Verweisfloskeln und textdeiktischen Ausdrücken in **Hyperlinks**, die als Linkziel die Textpassage haben, auf die verwiesen oder gezeigt wird.
3. **Sichtfelderweiterung**: Konnektoren oder Verweisfloskeln, die unmittelbar auf den vorigen oder den folgenden Abschnitt verweisen, werden zu Hyperlinks, die mit dem vorigen bzw. nachfolgenden Abschnitt verknüpft sind (konzeptionell ein Spezialfall von Operation 2, technisch aber anders umgesetzt).
4. Die **Tilgung** von vorwiegend stilistisch motivierten Konnektoren.

1. Einblendung des Antezedens

Beispiel:

Tochtermann (1995)

Weiterhin unterscheidet **er** nach der *Anzahl der in einen Link involvierten Anker* in *1:1-Links*, in denen ein Ausgangs-Anker mit genau einem Zielanker verknüpft ist; *1:n-Links*, in denen ein Ausgangs-Anker mit mehreren Zielankern verbunden ist, und *n:m-Links*, in denen mehrere Anker unabhängig von der Traversierungsrichtung miteinander zu einem Linking-Muster kombiniert sind. Im Linking-Element von HTML sind nur 1:1-Links vorgesehen; die obige Spezifikation und das Konzept des „Extended Link“ (im Sinne der XLink-Spezifikation) sehen auch Links mit mehreren Ankern vor.

Durch einen *Linktitel* wird das Antezedens (hier: Tochtermann (1995)) eingeblendet, d. h. das Antezedens wird nur sichtbar, wenn der Benutzer mit der Maus darüberfährt.

Auswertung der Annotation: Nur Kospezifikation (<cospecLink>-Elemente) werden ausgewertet; die Koreferenz spielt in diesem Zusammenhang keine Rolle. Dabei ist zu beachten, dass derselbe phorische Ausdruck mehrmals in verschiedenen <cospecLink>-Elementen über das Attribut phorIDRef angesprochen werden kann. Die Einblendung des Antezedens erfolgt jedoch nicht unbedingt, sondern abhängig vom Typ der Kospezifikation, der im Attribut relType des Elements <cospecLink> annotiert wird:

Attributwert des Attributs relType	Aktion
ident, synonym, hyperonym, hyponym, propName, paraphrase, addInfo, isA, abstrProp, abstrProj, abstrProjProp, abstrEventType, abstrEvent, unknown, bridging	durch Linktitel Antezedens / Antezedentien einblenden
abstrCluster	Sichtfelderweiterung (s.u.)
poss	durch Linktitel die Zeichenkette „von“ gefolgt vom Antezedens / den Antezedentien einblenden

Diese Regeln erklären sich aus der folgenden Strategie: In der Regel kann ein Leser einen Rückverweis auf das Antezedens am oberen Modulrand nicht auflösen, da das Antezedens im vorangehenden Modul steht. Daher wird bei den meisten Attributwerten das Antezedens durch einen Linktitel eingeblendet.

Besteht das Antezedens aus einer Summe von Propositionen oder einem ganzen Textabschnitt (Attributwert abstrCluster), so würde der Linktitel sehr viel Text enthalten. In diesem Fall wird daher das Modul so vergrößert, dass es das vorangehende bzw. nachfolgende Modul umfasst (siehe 0. 3. *Sichtfelderweiterung*).

Im Falle einer Possessivrelation (Attributwert poss) wäre die Einblendung des Antezedens verwirrend, da die beiden Diskursentitäten ja nicht auf dasselbe Objekt verweisen, wie im folgenden Beispielsatz:

Hans hat sich einen neuen Browser installiert. Sein Bruder hat ihm nämlich geraten, Mozilla statt dem Internet Explorer von Microsoft zu verwenden.

Statt das Antezedens („Hans“) über „sein Bruder“ einzublenden, wird stattdessen die Zeichenfolge „von“ und das Antezedens eingeblendet: „von Hans“.

Annotationsbeispiele:

Im folgenden Annotationsbeispiel wird über dem blau gekennzeichneten Textstück ein Linktitel mit den rot gekennzeichneten Textstücken eingeblendet:

Das Hypertextmodell von (Tochtermann 1995) unterscheidet drei architektonische Ebenen, die in Abb. 3-1 visualisiert sind.

- <discourseEntity deID="id_2" deType="nom">Die Speicherebene</discourseEntity> verwaltet die im Hypertext verwendeten Medienobjekte, Informationen zu Links und Modulen, sowie Operationen für den Zugriff auf und die Manipulation von diesen Medienobjekten.

- Auf <discourseEntity deID="id_3" deType="nom">der konzeptionellen Ebene</discourseEntity> werden die wesentlichen konzeptionellen Entitäten -

Links, Komponenten und Module - und deren Strukturen modelliert und zulässige Operationen auf diesen Entitäten und Strukturen definiert.

- Auf `<discourseEntity deID="id_4" deType="nom">der Präsentations- und Interaktionsebene</discourseEntity>` wird festgelegt, wie die Objekte der konzeptionellen Ebene den Nutzern präsentiert werden und welche Operationen die Nutzer auf den ihnen präsentierten Objekten durchführen können.

Die Trennung `<discourseEntity deID="id_5" deType="nom">der Ebenen</discourseEntity>` macht es möglich, die Informationsmodellierung auf der konzeptionellen Ebene unabhängig zu halten

```
<semRel>
  <cospecLink
    relType="hyperonym"
    phorIDRef="id_5"
    antecedentIDRefs="id_2 id_3 id_4"/>
</semRel>
```

1. von der Art und Weise, wie die Daten gespeichert und verwaltet werden.
2. von der Art und Weise, wie die Objekte und Strukturen dem Nutzer auf einem Datensichtgerät angezeigt werden und wie der Nutzer die Objekte manipulieren kann.

Im folgenden Beispiel wird über das blau gekennzeichnete Textstück ein Linktitel eingeblendet mit dem Text „von“ und dem Text des rot gekennzeichneten Textstücks (also „von Hans“), da `relType="poss"`¹:

```
<discourseEntity deID="de_n_282 deType="nom"> Hans </discourseEntity> hat
sich einen neuen Browser installiert. <discourseEntity deID="de_n_29" deTy-
pe="nom"> Sein Bruder </de> hat ihm nämlich geraten, Mozilla statt dem In-
ternet Explorer von Microsoft zu verwenden.
<semRel>
  <cospecLink
    relType="poss"
    phorIDRef="de_n_29"
    antecedentIDRefs="de_n_28"/>
  </cospecLink>
</semRel>
```

1 Dieser Fall kommt in den bisherigen Annotationen nicht vor und wurde daher vorerst nicht implementiert.

2. Umwandlung von Verweisfloskeln in Hyperlinks

Beispiel:

Weiterhin unterscheidet er nach der *Anzahl der in einen Link involvierten Anker* in *1:1-Links*, in denen ein Ausgangs-Anker mit genau einem Zielanker verknüpft ist; *1:n-Links*, in denen ein Ausgangs-Anker mit mehreren Zielankern verbunden ist, und *n:m-Links*, in denen mehrere Anker unabhängig von der Traversierungsrichtung miteinander zu einem Linking-Muster kombiniert sind. Im Linking-Element von HTML sind nur 1:1-Links vorgesehen; die **obige Spezifikation** und das Konzept des „Extended Link“ (im Sinne der XLink-Spezifikation) sehen auch Links mit mehreren Ankern vor.

Verweisfloskeln im Text werden in Hyperlinks umgewandelt.

Ausgewertet werden dazu `<connective>`-Elemente und deren Attribute `connectedTo` bzw. `connectedToID`.

Für die Programmierung unterscheiden wir die beiden folgenden Fälle:

1. `connectedTo="structuralUnit"`: Verweis auf eine durch die Dokumentenstruktur bestimmte Einheit im Text. Beispiel: „Abschnitt 7.3“. Dieser Attributwert soll bei der Annotation nur dann gewählt werden, wenn die Texteinheit aus dem Elementinhalt automatisch eindeutig berechnet werden kann. Wirkung im Hypertext: Link zu dieser Einheit (Ziel ist zu berechnen).
2. `connectedTo="specifiedByID"`: Verweis auf eine andere Einheit im Text, die durch den Wert des `connectedToID`-Attributs desselben `<connective>`-Elements bestimmt ist. Bewirkt einen Link zu diesem Element.

Beispielannotationen für Fall 1:

Auf der Basis der systemunabhängigen Erläuterung und Terminologisierung dieser Konzepte möchte ich `<connective connectedTo="structuralUnit">in Kapitel 4</connective>` die wichtigsten Besonderheiten des World Wide Web als Hypertextplattformsystem beschreiben.

Im Gegensatz zu den einfachen Dokumentenmodulen gehören die "Sichten-Module" und die "zusammengesetzten Module" zu den Strukturierungskonzepten `<connective connectedTo="structuralUnit">(vgl. 3.3)</connective>`.

Dieser Fall kommt sehr häufig vor, das Verweisziel ist jedoch nicht einfach zu berechnen. Aufgrund des hohen Aufwands und relativ geringen Nutzens wird dieser Fall zunächst nicht implementiert. Ein Implementierungskonzept, das die regulären Ausdrücke von XSLT 2.0 nutzt, liegt für den Fall des annotierten Textes Storrer_2002 vor. Im Rahmen der Dissertation von Eva Lenz wird angestrebt, diesen Fall in allgemeiner Form so zu lösen, dass nur (autor-spezifische) reguläre Ausdrücke angegeben werden müssen. Diese Lösung könnte dann auch für die Generierung ganz anderer Links genutzt werden, die mittels regulärer Ausdrücke spezifiziert werden können. Vorerst wurde daher Fall 1 in Fall 2 umannotiert.

Beispielannotationen für Fall 2:

Die Trennung der Ebenen macht es möglich, die Informationsmodellierung auf der konzeptionellen Ebene unabhängig zu halten

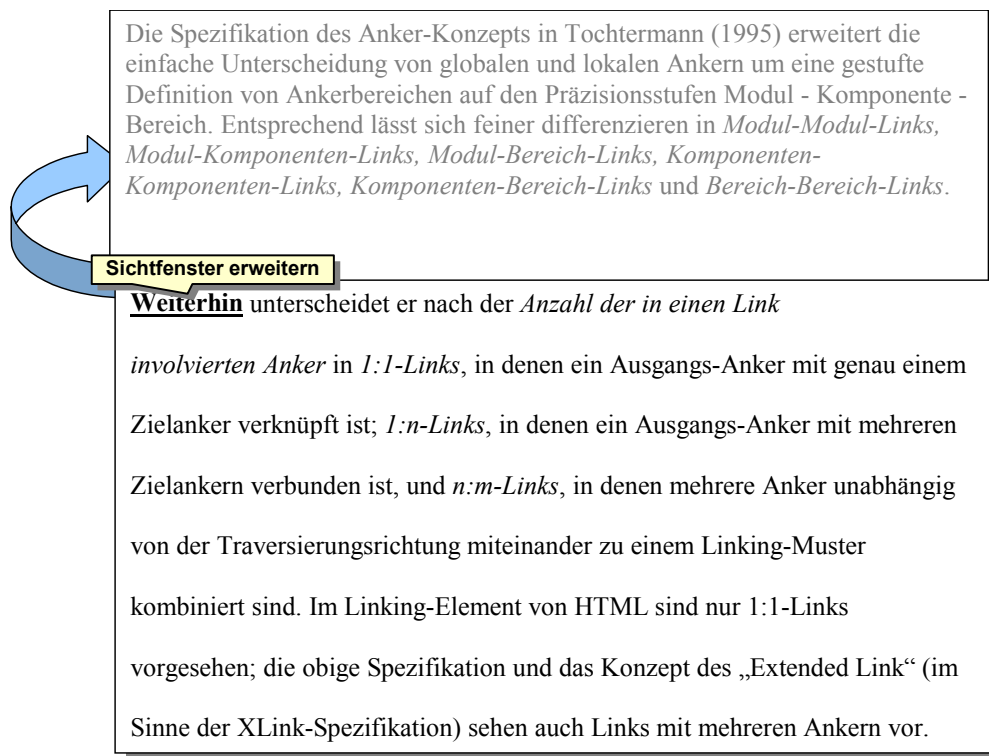
1. von der Art und Weise, wie die Daten gespeichert und verwaltet werden.
<connectiveAnchor ID="caID_1">2. von der Art und Weise, wie die Objekte und Strukturen dem Nutzer auf einem Datensichtgerät angezeigt werden und wie der Nutzer die Objekte manipulieren kann.</connectiveAnchor>

Für den hier diskutierten Zusammenhang ist - wie bereits mehrfach erläutert - insbesondere <connective connectedTo="specifiedByID" connectedToID="caID_1">(2)</connective> wichtig, denn in diesem Punkt unterscheiden sich Hypertextdokumente von den gedruckten Pendanten: Während gedruckte Dokumente ihre Inhalte allen Nutzern in derselben Form präsentieren, stellen Datenbank- und Hypermediasysteme die Inhalte erst auf Anfrage auf einem Datensichtgerät zusammen.

Hier wird ein Link von dem blau gekennzeichneten Textstück auf das grün gekennzeichnete generiert.

3. Sichtfelderweiterung

Beispiel:



Die Spezifikation des Anker-Konzepts in Tochtermann (1995) erweitert die einfache Unterscheidung von globalen und lokalen Ankeren um eine gestufte Definition von Ankerbereichen auf den Präzisionsstufen Modul - Komponente - Bereich. Entsprechend lässt sich feiner differenzieren in *Modul-Modul-Links*, *Modul-Komponenten-Links*, *Modul-Bereich-Links*, *Komponenten-Komponenten-Links*, *Komponenten-Bereich-Links* und *Bereich-Bereich-Links*.

Sichtfenster erweitern

Weiterhin unterscheidet er nach der *Anzahl der in einen Link involvierten Anker* in *1:1-Links*, in denen ein Ausgangs-Anker mit genau einem Zielanker verknüpft ist; *1:n-Links*, in denen ein Ausgangs-Anker mit mehreren Zielankern verbunden ist, und *n:m-Links*, in denen mehrere Anker unabhängig von der Traversierungsrichtung miteinander zu einem Linking-Muster kombiniert sind. Im Linking-Element von HTML sind nur 1:1-Links vorgesehen; die obige Spezifikation und das Konzept des „Extended Link“ (im Sinne der XLink-Spezifikation) sehen auch Links mit mehreren Ankeren vor.

„Sichtfelderweiterung“ bedeutet, dass zusätzlich zum aktuellen Modul das vorangehende (Sichtfelderweiterung nach oben) oder das nachfolgende (Sichtfelderweiterung nach unten) Modul angezeigt wird.

Die Sichtfelderweiterung wird erzeugt, wenn Konnektoren (<connective>-Elemente) mit dem Attribut unmittelbar auf den vorhergehenden (connectedTo="backward") oder den nachfolgenden (connectedTo="forward") Abschnitt verweisen. Es wird jedoch *keine* Sichtfelderweiterung erzeugt, wenn das Attribut pragType desselben connective-Elements auf den Wert stylistic gesetzt ist, da in diesem Fall der Konnektor getilgt wird (vgl. Abschnitt 0. 4. *Tilgung*).

Beispielannotationen für Sichtfelderweiterung nach oben

`<connective connectedTo="backward">Die o.g. Präzisierung</connective>` des Link-Konzepts basiert auf dem Konzept des Ankers, wobei Anker in der in (Tochtermann 1995) vorgelegten Spezifikation den Status von eigenständigen Objekten auf der konzeptionellen Ebene haben. Als Anker können ganze Module, Modul-Komponenten oder abgegrenzte Bereiche in Modulen oder Komponenten fungieren (vgl. (Tochtermann 1995, 71f)). Bereiche können dabei in diskreten Medienobjekten ausgezeichnet werden - also z.B. Textpassagen oder Teile von Grafiken sein - ; es können aber auch Teile von kontinuierlichen Medienobjekten als Anker fungieren, z.B. Passagen von Audioobjekten, Sequenzen von Bildern eines Videos oder Einzelbilder einer Animation.

Hier wird eine Sichtfelderweiterung nach oben generiert.

Im folgenden Fall ist jedoch gleichzeitig das Attribut `pragType="stylistic"` gesetzt. Das bedeutet, dass Regel 4) greift (s. u.), die höher gewichtet ist. D. h., in diesem Fall wird keine Sichtfelderweiterung erzeugt:

(Tochtermann 1995, 48f) spezifiziert das Konzept des einfachen Dokumenten-Moduls `<connective pragType="stylistic" connectedTo="backward">entsprechend</connective>` als eine eindeutige Zuordnung zwischen einem Identifikator und einem Modul-Objekt, ...

Beispielannotationen für Sichtfelderweiterung nach unten

`<connective connectedTo="forward">In den folgenden Abschnitten</connective>` werde ich wichtige Hypertext-Konzepte erläutern und dabei die im Weiteren verwendeten Termini für Hypertext-Bauteile und -Strukturen einführen.

An dieser Stelle erfolgt eine Sichtfelderweiterung nach unten.

4. Tilgung

Beispiel:

Weiterhin unterscheidet er ~~nach~~ nach der Anzahl der in einen Link involvierten Anker in *1:1-Links*, in denen ein Ausgangs-Anker mit genau einem Zielanker verknüpft ist; *1:n-Links*, in denen ein Ausgangs-Anker mit mehreren Zielankern verbunden ist, und *n:m-Links*, in denen mehrere Anker unabhängig von der Traversierungsrichtung miteinander zu einem Linking-Muster kombiniert sind. Im Linking-Element von HTML sind nur *1:1-Links* vorgesehen; die obige Spezifikation und das Konzept des „Extended Link“ (im Sinne der XLink-Spezifikation) sehen auch Links mit mehreren Ankern vor.

Vorwiegend stilistisch motivierte Konnektoren und Konnektivpartikeln werden getilgt (gelöscht). Es sind dies durch `<connective pragType="stylistic">` ausgezeichnete Textstücke.

Beispielannotation:

Für die beiden grundlegenden Einheiten der Hypertextmodellierung möchte ich
<connective pragType="stylistic" connected-
To="unspecified">[allerdings](#)</connective> in dieser Arbeit bei identischer
intensionaler Bedeutung - andere Termini verwenden.