

Generierung von Glossar-Sichten mit dem Stylesheet `XTM2SVG.xsl`

Dokumentation

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

Benjamin Birkenhake

2004

- 1 Überblick
- 2 Aufgabenkatalog
- 3 Umsetzung der Anforderungen
- 4 Verknüpfungen zu anderen Dateien
- 5 Mögliche zukünftige Komponenten
- 6 Fazit

1. Überblick

Das Stylesheet `XTM2SVG.xsl` ist die zentrale Komponente für die Erstellung der Präsentation des *TermNet*. Von diesem Stylesheet aus werden weitere Stylesheet eingebunden, die unterschiedliche Details der Transformation vom inferierten *TermNet* aus zur HTML-SVG-Präsentation steuern. Dieses Stylesheet ist eingebettet in einen größeren Kontext. Es arbeitet zum einen nicht direkt auf dem Output von *K-Infinity* und muss zum anderen mit den Ergebnissen des Stylesheets `AllXML2HTML.xsl` kompatibel sein.

Um den komplexen Anforderungen gerecht zu werden, die die Vorgaben der Präsentation stellten, ist das Stylesheet ebenso wie das `AllXML2HTML.xsl`-Stylesheet in mehrere Transformationsschritte gegliedert.

2. Aufgabenkatalog

1. Es soll zu jedem Lexem im *TermNet* einen Glossar-Eintrag in Form eines HTML-SVG-Dokumentes geben.
2. Jeder Glossar-Eintrag soll mit den Definitionen dieses Lexems in den Texten des Korpus verknüpft werden.
3. Es soll eine Index-Seite des Glossars geben, auf der alle Lexeme aufgelistet werden.
4. Der SVG-Anteil eines Glossar-Eintrages soll folgende Anforderungen erfüllen:
 - Englische Lexeme sollen durch ein Fähnchen als solche kenntlich gemacht werden.
 - Es soll vier funktionale Flächen geben, von denen eine dasjenige Konzept repräsentiert, welches durch das vom Benutzer ausgewählte Lexem lexikalisiert wird, und die anderen drei diejenigen Konzepte repräsentieren, die mit ersterem Konzept über konzeptuelle Relationen (Hyperonymie/Hyponymie, Holonymie/Meronymie und Antonymie) verbunden sind¹:
 - 4-1 **Eine Konzeptbox im Zentrum der Seite:**
 - 4-1-1 Sie soll den Namen des ausgewählten Lexems, den Namen des Konzepts und alle weiteren Lexeme, die dieses Konzept lexikalisieren, enthalten.
 - 4-1-2 Das ausgewählte Lexem soll in der Mitte der Box stehen.
 - 4-1-3 Der Name des Konzepts soll in Versalien oben links in der Box stehen.
 - 4-1-4 Um das ausgewählte Lexem herum sollen gleichmäßig (kreis- oder sternförmig) die weiteren Lexeme stehen, die das Konzept lexikalisieren.
 - 4-1-5 Diese weiteren Lexeme sollen mit dem in der Mitte der Box angezeigten Lexem durch eine Linie verbunden sein, die die lexikalische Relation zwischen

1 Zum Inventar konzeptueller Relationen in *TermNet* siehe: Beißwenger, Michael, Storrer, Angelika & Runte, Maren (2004): Modellierung eines Terminologienetzes für das automatische Linking auf der Grundlage von WordNet. In: Kunze, Claudia; Lemnitzer, Lothar; Wagner, Andreas (Hrsg.): Anwendungen des deutschen Wortnetzes in Theorie und Praxis. Beiträge des GermaNet-Workshops Tübingen, Oktober 2003 (LDV-Forum – Zeitschrift für Computerlinguistik und Sprachtechnologie 19. 1/2), 113-125.

den beiden Einheiten repräsentiert und in deren Mitte der Name des jeweiligen Relationstyps angezeigt wird.

4-1-6 »Bedeutungsähnlichkeit« soll nur dann als Relationstypenname angezeigt werden, wenn der Relationstyp in der *TermNet*-Modellierung nicht explizit spezifiziert ist.

4-2 **Konzeptboxen, die der zentralen Konzeptbox vertikal übergeordnet sind:**

4-2-1 Über der zentralen Konzeptbox sollen nebeneinander alle Konzepte dargestellt werden, die zum zentralen Konzept in Hyperonymie- oder Holonymie-relation stehen.

4-2-2 Für jedes Hyperonym resp. Holonym sollen alle Lexeme angezeigt werden, die dieses Konzept lexikalisieren.

4-2-3 Jede Hyperonym- resp. Holonymbox soll mit der zentralen Konzeptbox über eine Linie verbunden sein, in deren Mitte der Name der Relation und an deren Enden die Rollen stehen, die die einzelnen Relationspartner spielen.

4-3 **Konzeptboxen, die der zentralen Konzeptbox horizontal nebengeordnet sind:**

4-3-1 Links neben der zentralen Konzeptbox sollen Konzeptboxen dargestellt werden, welche die Antonyme zum zentralen Konzept repräsentieren.

4-3-2 Für jedes Antonym sollen alle Lexeme angezeigt werden, die dieses Konzept lexikalisieren.

4-3-3 Jede Antonymbox soll mit der zentralen Konzeptbox über eine Linie verbunden sein, in deren Mitte der Name der Relation und an deren Enden die Rollen stehen, die die einzelnen Relationspartner spielen.

4-4 **Konzeptboxen, die der zentralen Konzeptbox vertikal untergeordnet sind:**

4-4-1 Unter der zentralen Konzeptbox sollen nebeneinander alle Konzepte dargestellt werden, die zum zentralen Konzept in Hyponymie- oder Meronymiere-lation stehen.

4-4-2 Für jedes Hyponym resp. Meronym sollen alle Lexeme angezeigt werden, die dieses Konzept lexikalisieren.

4-4-3 Jede Hyponym- resp. Meronymbox soll mit der zentralen Konzeptbox über eine Linie verbunden sein, in deren Mitte der Name der Relation und an deren Enden die Rollen stehen, die die einzelnen Relationspartner spielen.

4-4-4 Kohyponyme einer Gruppe sollen aus darstellungspraktischen Gründen untereinander stehen.

4-4-5 Kohyponyme einer Gruppe sollen sich durch eine unterschiedliche Grundfarbe von anderen Kohyponymgruppen resp. Meronymen unterscheiden.

4-4-6 Kohyponyme einer Gruppe sollen sich durch einen unterschiedlichen Farbton von einander unterscheiden.

5. Alle Relationstypennamen und Rollenbezeichnungen sollen aus Gründen der Übersicht-

lichkeit standardmäßig ausgeblendet sein und erst bei Mausberührung eingeblendet werden.

3. Umsetzung der Anforderungen

Die Anforderungen 1 bis 4-4-5 wurden vollständig implementiert.

Das Stylesheet arbeitet auf dem automatisch weiterverarbeiteten Export des *K-Infinity-Tools*². *K-Infinity* erzeugt einen XTM-Export. Da es in *TermNet* zwei Klassen von Entitäten gibt (Lexeme und Konzepte), die in XTM nicht standardmäßig enthalten sind, wurde eine Relation »TermNet-Klasse-Domänen-Instanz« eingeführt, die jeweils zwei Instanzen von Lexem und Konzept verbindet.

Der Transformationsprozess wurde in drei Teilprozesse aufgebrochen:

1. Schritt: Von der TermNet-Topicmap zu einem Zwischenrepräsentationsformat

Name des Modes : `xm2concept`

Name der entstehenden Variable: `$concept`

In diesem Schritt werden aus der Topicmap Daten entnommen, und in ein anderes XML-Format überführt, welches zusammengehörende Lexeme und Konzepte gruppiert. Der Schritt wird für jedes Topic vom Typ »Lexem« in der Topicmap durchgeführt.

Im Folgenden werden die einzelnen Elemente und deren Attribute erläutert:

root: Das Root-Element. Es hat lediglich strukturierende Funktion.

centralTopic: Das `centralTopic` ist dasjenige Topic, das aktuell bearbeitet wird. Sein Elementinhalt soll den `baseNameString` des Topics enthalten.

@type: In der Hytex-Topicmap gibt es zwei Sorten von Topics: »Lexeme« und »Konzepte«. Nur Lexeme werden `centralTopics`. Um aber offen zu bleiben, kann hier auch ein anderer Wert angegeben werden, im Normalfall aber sollte hier »Lexem« stehen.

@lang: In diesem Attribut wird die Sprachenzuordnung (DE oder EN) gespeichert.

wordTopic: Ein `wordTopic` ist ein Topic, das mit dem `centralTopic` assoziiert und vom Typ »Lexem« ist. Jedes `wordTopic` soll später zu einem Punkt auf dem Kreis werden.

wordTopicName: enthält den `baseNameString` des assoziierten `wordTopics`.

edgeTypeName: enthält den `baseNameString` des Topics, von dem die Assoziation eine `InstanceOf` ist.

roleCentralTopic: enthält den `baseNameString` des Topics, von dem das Member, in welchem das `centralTopic` `topicref` ist, `instanceOf` ist.

roleOuterTopic: enthält den `baseNameString` des Topics, von wel-

2 Zur Vorverarbeitung, insbesondere Inferenzen auf dem TermNet, siehe die Dokumentation Verarbeitungsschritte des terminologischen Netzes (Eva Anna Lenz).

chem das Member, in dem das assoziierte Topic topicref ist, instanceOf ist.

conceptTopic: Ein conceptTopic ist ein Topic, das mit dem centralTopic assoziiert ist und vom Typ »Konzept« ist. Jedes conceptTopic soll später in einer der Boxen visualisiert werden, je nach Art der Assoziation.

conceptTopicName: enthält den baseNameString des assoziierten conceptTopics.

edgeTypeName: enthält den baseNameString des Topics, von welchem die Assoziation eine instanceOf ist.

roleCentralTopic: enthält den baseNameString des Topics, von dem das Member, in welchem das centralTopic topicref ist, instanceOf ist.

roleOuterTopic: enthält den baseNameString des Topics, von dem das Member, in welchem das assoziierte Topic topicref ist, instanceOf ist.

subnodes: enthält Verweise auf alle Topics die durch dieses assoziierte Topic lexikalisiert werden.

subnode: enthält den baseNameString eines Topics, das durch das mit dem centralTopic assoziierte Topic lexikalisiert wird.

Beispiel:

```
<root>
```

```
<topic>
```

```
<centralTopic type="lexeme" lang="">Link</centralTopic>
```

```
<wordTopic lang="">
```

```
<wordTopicName>Hyperlink</wordTopicName>
```

```
<edgeTypeName>Synonymie</edgeTypeName>
```

```
<roleCentralTopic>Synonym</roleCentralTopic>
```

```
<roleOuterTopic>Synonym</roleOuterTopic>
```

```
</wordTopic>
```

```
<conceptTopic lang="" difcrit="">
```

```
<conceptTopicName>*1:1-
```

```
Link</conceptTopicName>
```

```
<edgeTypeName>subclass/superclass</edgeTypeName>
```

```
<roleCentralTopic>Oberbegriff</roleCentralTopic>
```

```
<roleOuterTopic>Unterbegriff</roleOuterTopic>
```

```
<subnodes>
```

```
<subnode lang="">one-to-one-Link</subnode>
```

```

                                <subnode lang="">1:1-Link</subnode>
                                </subnodes>
                                </conceptTopic>
                                </topic>
</root>

```

Der Mode `xTM2concept` ist der komplexeste Modus.

Folgende Abschnitte werden in diesem Mode durchlaufen:

1. Jedes Topic wird zu einem `centralTopic` gemacht.
2. Es werden alle Assoziationen, an denen dieses Topic beteiligt ist, betrachtet.
3. Es werden alle Topics betrachtet, die als Assoziationsmember in einer Assoziation mit dem `centralTopic` stehen, wobei es zwei Möglichkeiten gibt:
 - 3a. Ein solches Topic ist ein Lexem. Dann wird es zu einem `outerTopic` gemacht. Ende dieses Zweiges.
 - 3b. Ein solches Topic ist kein Lexem (sondern ein Konzept). Dann werden alle Assoziationen dieses Topics betrachtet:
Es werden alle Topics betrachtet, die als Assoziationsmember in einer Assoziation mit diesem Topic stehen, wobei es wiederum zwei Möglichkeiten gibt:
 - 3b-1 Ein solches Topic ist ein Lexem. Dann wird gar nichts gemacht. Ende dieses Zweiges.
 - 3b-2 Ein solches Topic ist kein Lexem (also ein Konzept). Dann wird das Topic zu einem `outerTopic` des `centralTopics` gemacht.
4. Alle Topics, die das in 3b-2 erstellte `outerTopic` lexikalisieren (also mit ihm in einer Assoziation stehen, die als eine `instanceOf` „lexikalisiert“ ist) werden zu `subnodes` gemacht.
Jedes Mal, wenn ein Topic zu einem `outerTopic` gemacht wird, werden die Informationen aus der vorhergehenden Assoziation bzgl. Rollen und Assoziationsname zu Assoziationsinformationen.

2. Schritt: Vom Zwischenrepräsentationsformat zu einem Prä-Visualisierungsmodell

Name des Modes: `concept2model`

Name der entstehenden Variable: `$model`

Aus der Variablen, die aus `XTM2Concept` entsteht, wird eine weitere Variable entwickelt, die allgemeine Grundlagen für eine sternförmige Visualisierung bietet. Dieses Format kann somit auch genutzt werden, um andere XTM-Formate zu visualisieren.

Im Folgenden werden die einzelnen Elemente und deren Attribute erläutert:

root: Das Root Element. Es hat lediglich strukturierende Funktion.

centralNode: Enthält den Namen des zentralen Themas.

@type: Um Themen noch in unterschiedliche Klassen teilen zu können, kann in dem Attribut `@type` ein Index angegeben werden. (Evtl. sollte dieses Modell später um weitere Klassifizierungsattribute wie `@class`, `@index`, `@lang` erweitert werden)

outerNodes: Ist ein Wrapper für alle Nodes, die als Punkte auf dem Kreis visualisiert werden sollen.

outerNode: beinhaltet alle Informationen eines Knotens, der später als Punkt auf dem Kreis visualisiert werden soll.

@type: Um Themen noch in unterschiedliche Klassen teilen zu können, kann in dem Attribut `@type` ein Index angegeben werden.

@href: Beinhaltet einen Verweis zu einem anderen `centralNode`.

@class: Beinhaltet Informationen darüber, ob ein `outerNode` Teil einer Node-Klasse (d.h.: ein Kohyponym) ist. (In diesem Fall wird das Attribut `@differentiationcriteria` dafür verwendet.)

@position: Beinhaltet Informationen über die relative Position der `outerNode` zur `centralNode`. (wird hier aus dem Element `roleOuterTopic` gewonnen; erlaubte Werte: `top`, `bottom`, `left`, `right`)

outerNodeName: Beinhaltet den Namen eines Knotens, der zu dem `centralNode` gehört und später auf dem Kreis visualisiert werden soll.

edgeTypeName: Beinhaltet den Namen der Kante zwischen dem `centralNode` und dem `outerNode`.

roleCentralNode: Beinhaltet den Namen der Rolle, die der `centralNode` in der Verbindung zwischen dem `centralNode` und dem `outerNode` spielt.

roleOuterNode: Beinhaltet den Namen der Rolle, die der `outerNode` in der Verbindung zwischen dem `centralNode` und dem `outerNode` spielt.

subnodes: Für den Fall, dass der `outerNode` auf mehr als einen `centralNode` verweisen soll, können `subnodes` eingefügt werden.

subnode: Beinhaltet den Namen eines weiteren `centralNodes`, auf den verwiesen werden soll.

@href: Beinhaltet einen Verweis zu einem anderen `centralNode`.

Beispiel:

```
<root>
  <node>
    <centralNode type="1" lang="">Link</centralNode>
    <outerNode type="2" lang="" class="" position="">
      <outerNodeName>*1:1-Link</outerNodeName>
      <edgeTypeName>subclass/superclass</edgeTypeName>
      <roleCentralNode>Oberbegriff</roleCentralNode>
      <roleOuterNode>Unterbegriff</roleOuterNode>
      <subnodes>
        <subnode lang="">one-to-one-Link</subnode>
        <subnode>1:1-Link</subnode>
      </subnodes>
    </outerNode>
    <outerNode type="1" lang="" class="" position="">
      <outerNodeName>Hyperlink</outerNodeName>
      <edgeTypeName>Synonymie</edgeTypeName>
      <roleCentralNode>Synonym</roleCentralNode>
      <roleOuterNode>Synonym</roleOuterNode>
    </outerNode>
  </node>
</root>
```

3. Schritt: Vom Prä-Visualisierungsmodell nach HTML / SVG

Name des Modes: Model2SVG

Basierend auf der Variable `$model` wird nun das Topic in eine HTML- und eine SVG-Datei transformiert. Dabei werden in erster Linie Berechnungen über die Höhe und Breite der unterschiedlichen Boxen gemacht, da SVG anders als HTML nicht über eine Renderingengine verfügt, die das übernehmen könnte.

Weiterhin werden in diesem Modus die HTML-Dateien erstellt, in die das jeweilige SVG eingebunden wird.

Zusätzlich werden in diesem Modus alle XML-Dokumente der Annotationsebene »Definitionen und Termverwendungsinstanzen«³ nach Definitionen des im `centralTopicName` ste-

3 Die Annotationsebene »Definitionen und Termverwendungsinstanzen« ist ausführlich beschrieben in einer gleichnamigen Dokumentation (Eva Anna Lenz & Michael Beißwenger; <http://www.hrz.uni-dortmund.de/~hytex/hytex/Publikationen/annotation-termini-und-definitionen.pdf>)

henden Strings durchsucht. Fundstellen werden in einer separaten HTML-Datei gespeichert, die bei Bedarf als Popup erscheint.

Weitestgehend umgesetzt: Die Anforderung 4-4-6 ist nur bedingt implementiert, da die Farbverläufe von Hand definiert wurden und im Fall des Konzepts »Link« mehr Hyponyme resp. Meronyme vorhanden sind als Farben und Farbtöne definiert wurden. Dieses Problem lässt sich jedoch mit wenig Arbeit beheben.

Teilweise umgesetzt: Die Anforderung 5 konnte nur sehr bedingt realisiert werden. Zwar ist die Funktionalität programmiert, doch zeigt das SVG den Fehler, dass es die Relationen aus bisher unbestimmbaren Gründen auf manchen Rechnern beim Überfahren mit der Maus anzeigt, und auf anderen Rechnern nicht. Da das *Adobe-SVG-Plugin* bereits als Fehlerquelle ausgeschlossen werden konnte, bleiben noch der Browser, Betriebssystem und Hardware als Fehlerquellen.

Dieses Problem ließe sich wahrscheinlich mit einigem Aufwand soweit lokalisieren, dass man eine Umgebung definieren kann, in der das Glossar garantiert läuft.

Illustration des Ergebnisses:

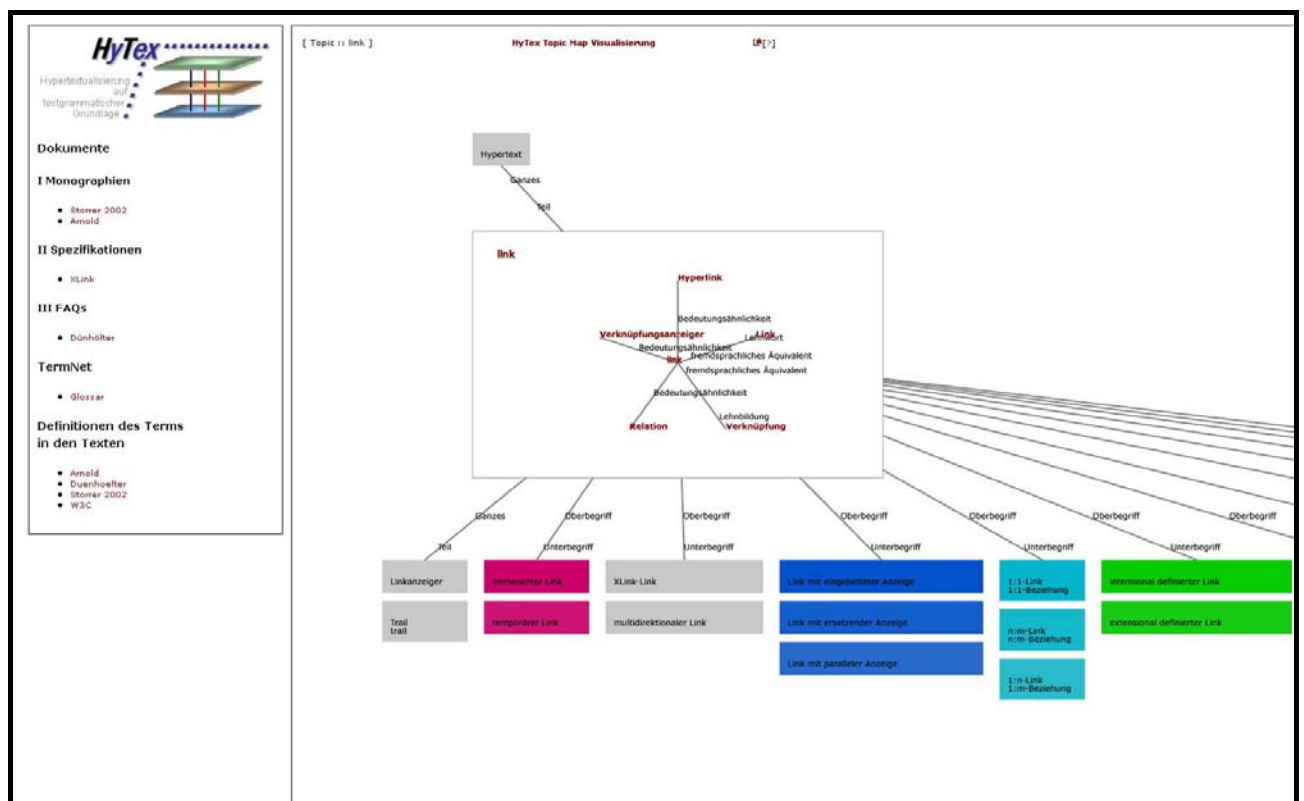


Abbildung 1: Übersicht der Visualisierung des Glossars am Beispiel „link“.

Links im Bild befindet sich in einem Kasten das Menü der gesamten Demo, in dem zunächst die Links zu den einzelnen Dokumenten zu finden sind, gefolgt von einem Link zur Startseite des Glossars sowie Links zu den Definitionen (in diesem Fall) des Lexems „link“ in den Dokumenten der Demo.

sowie im Projektbericht *Annotation definitorischer Textsegmente und »terminologiesensitives Linking«* (Michael Beißwenger; <http://www.hrz.uni-dortmund.de/~hytex/hytex/Publikationen/deflink.pdf>).

Der rechte Kasten enthält die SVG-Visualisierung. Im Zentrum befindet sich die Konzeptbox des Konzeptes, welchem das ausgewählte Lexem zugeordnet ist inklusive der weiteren Lexeme, die das Konzept lexikalisieren. Darüber und darunter befinden sich kleinere Konzeptboxen, deren Konzepte mit dem ausgewählten assoziiert sind.

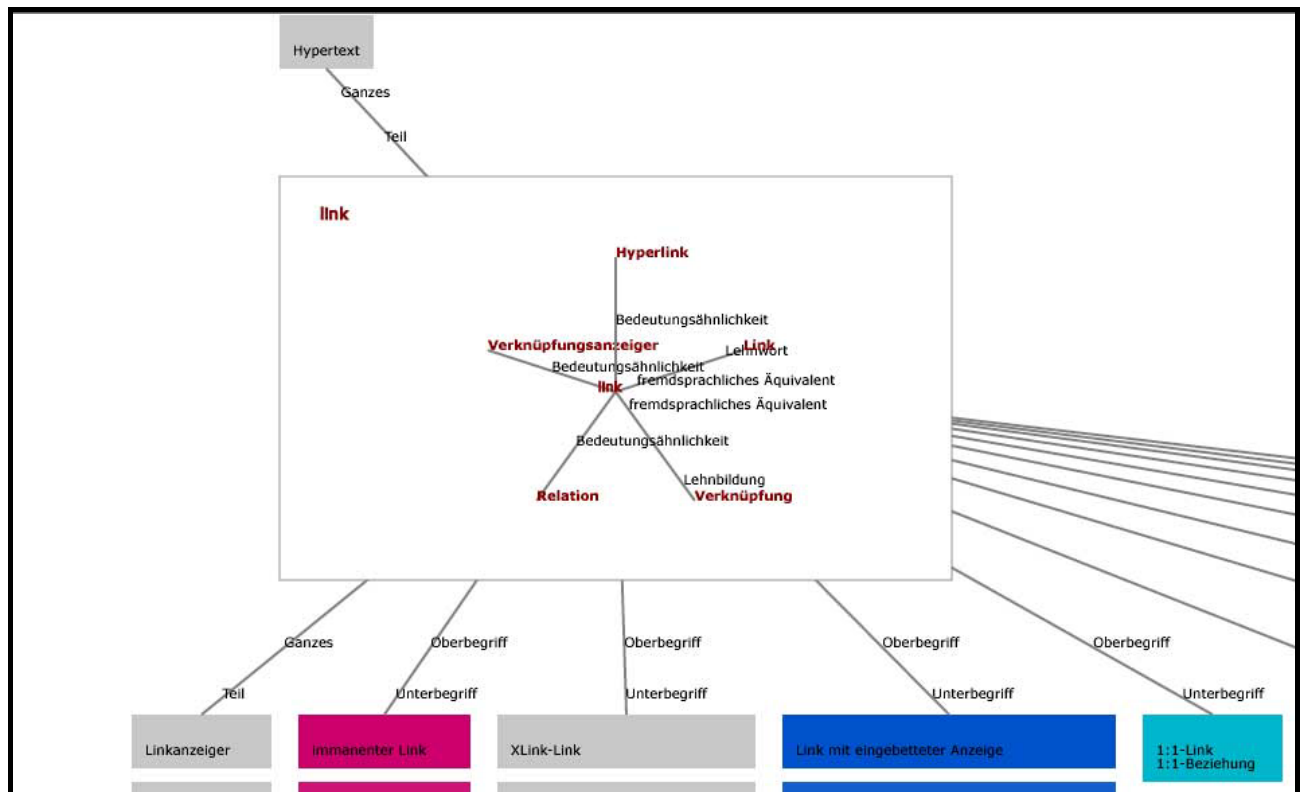


Abbildung 2: Detailansicht der Visualisierung mit der Konzeptbox, die sämtliche Lexeme enthält, die das betreffende Konzept lexikalisieren.

Im Zentrum der Konzeptbox steht sich in roter Schrift das vom Benutzer ausgewählte Lexem. Gleichmäßig um das Zentrum herum angeordnet finden sich – ebenfalls in roter Farbe – die übrigen Lexeme.

In schwarzer Schrift befinden sich sowohl zwischen den Lexemen innerhalb der zentralen Konzeptbox als auch zwischen der Konzeptbox und den Boxen für die übrigen angezeigten Konzepte Bezeichnungen für die jeweiligen Relationen (bei symmetrischen Relationen) bzw. Rollennamen für die beiden an einer Relation beteiligten Relate (bei asymmetrischen Relationen).

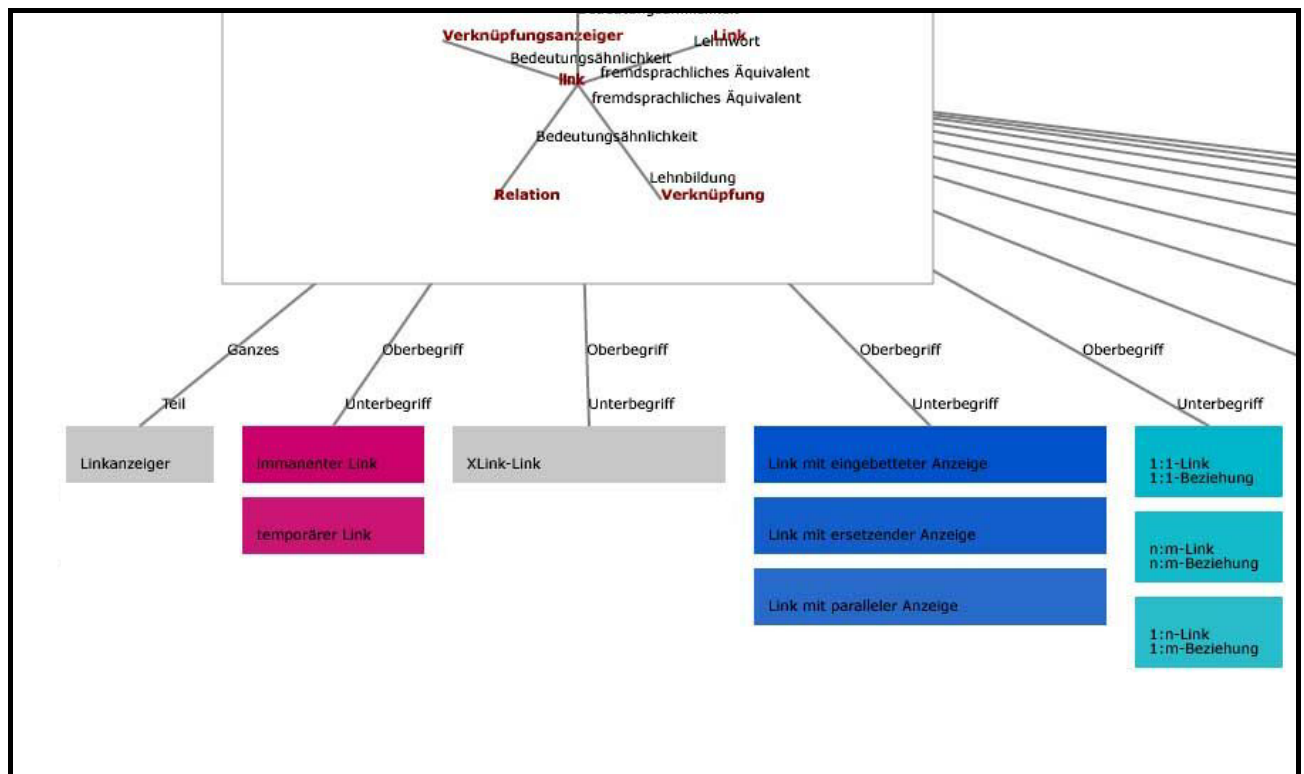


Abbildung 3: Detailansicht der mit dem gewählten (als zentrale Konzeptbox dargestellten) Konzept über Hyperonymie/Hyponymie- bzw. Holonymie/Meronymie-Relationen verbundenen Konzepte.

4. Verknüpfungen zu anderen Dateien

Das Stylesheet ist explizit mit folgenden anderen Komponenten verbunden:

XTM-inferiert: Dieses Stylesheet arbeitet auf dem inferierten XTM-TermNet.

XML-Korpus: Dieses Stylesheet arbeitet zusätzlich auf den XML-Dokumenten der Annotationsebene »Definitionen und Termverwendungsinstanzen«, um Definitionen zu finden.

XTM2SVG-trignm.xml In diesem Stylesheet befinden sich trigonometrische Funktionen, die zur Berechnung der Position der Lexeme auf einem Kreis benötigt werden.

XTM2SVG-keys.xml Keys sind zuvor definierte Zugriffe auf einen Teil eines XML-Dokumentes unter Verwendung eines ebenfalls definierten Schlüssels. Keys beschleunigen erheblich die Transformation des ersten Schritts.

baseName2fileName.xml Da die Dateinamen der zentrale Schlüssel zur Verbindung der HTML-Versionen des XML-Korpus und den HTML-SVG-Präsentationen des XTM-TermNet sind, ist es sinnvoll, zentral zu definieren, wie aus einem baseName ein Dateiname entsteht. Dies wird in dieser Datei geleistet.

5. Mögliche zukünftige Komponenten

Alternative Präsentation:

Die Transformation über mehrere Zwischenformate erlaubt es, an unterschiedlichen Stellen der Transformation einzugreifen und mit wenig Aufwand eine alternative Präsentation der Daten zu erzeugen.

So ist die bestehende Präsentation v.a. eine für das Glossar optimierte Präsentation: jeder Eintrag in das Glossar ist eine eigene Visualisierung. Ebenso interessant wie nahe liegend ist die Präsentation des gesamten *TermNet* in einer Seite. Zu diesem Zweck müssten allerdings komplexe Graphenpräsentationsberechnungen durchgeführt werden. Eine Möglichkeit dafür wäre, das erste Zwischenformat in ein spezielles Graphenformat zu übertragen, für das es bereits fertige Visualisierungswerkzeuge gibt.

Alternative Ausgangsdaten:

Das Stylesheet ist selbstverständlich in der Lage, beliebige XTM-TermNets gleicher Modellierung zu visualisieren, was v.a. für eine Visualisierung bestehender Wortnetze mit ähnlichem Konzept, wie dem *PrincetonWordNet* oder dem *GermaNet*, interessant sein könnte.

Mit einigen Umbauarbeiten lässt sich das Stylesheet auch an andere XTM-Modelle anpassen.

6. Fazit

Das Stylesheet `XTM2SVG.xsl` ist neben dem Stylesheet `AllXML2HTML.xsl`⁴ das zentrale Stylesheet für die Präsentation der Daten.

Das Stylesheet `XTM2SVG.xsl` erfüllt, bis auf kleine Ausnahmen, alle Anforderungen und ist so aufgebaut, dass es nicht nur leicht erweiterbar, sondern zudem auch für andere Projekte einsetzbar ist.

4 Vergleiche hierzu die Dokumentation: *Erzeugung modularisierter und verlinkter Hypertextsichten in der HyTex-Pilotversion* (Benjamin Birkenhake; <http://www.hrz.uni-dortmund.de/~hytex/hytex/Publikationen/html-sichten-aus-xml-annotationen.pdf>).