

Digital Editions beyond XML – Graph-based Digital Editions

Andreas Kuczera

Academy of Science and Literature, Mainz, Germany
andreas.kuczera@adwmainz.de

Abstract

XML has been the de facto standard for digital editions for years, but its serious limitations include an inability to represent overlapping markup and the encoding of multiple annotation hierarchies. With emerging graph database technologies we have the opportunity to develop new approaches. In this paper the advantages and modelling principles of graph-based digital editions will be discussed.

1 Introduction

XML in combination with TEI has become the standard format for digital editions. Most digital research environments, such as TextGrid or Ediarum, use TEI-XML for encoding sources and research data. But XML has some limitations which restrict researchers working on digital editions in certain ways. In this paper these limitations will be discussed and proposals for a graph-based digital edition will be presented.

2 The Problem

The use of XML for digital editions is widespread and has become a standard over the last years.

XML has some inherent limitations, however, such as its inability to represent overlapping markup as well as to encode multiple annotation hierarchies [5]. An easy-to-understand example of overlapping markup is if someone wants to encode the formal structure of a source. In that case he might need overlapping structures for pages and chapters. TEI solves this problem by using empty elements like <lb> and <pb> instead of <div> and <p>-tags. Another example is different sources of the same text. If you want to encode them in a single document, you quickly encounter overlapping markup-structures on one level. The situation becomes even worse when you try to encode diverging interpretations of the same source by different researchers in a single file.

To solve this problem several approaches have been developed. Daniel Jettka presents some of these problems in his paper [2]. A first approach is to use additional structures for the representation of multiple hierarchies in standard XML with the help of milestones [1] or fragmentation, or a mix of both.

What all these approaches have in common is that the complexity of the XML increases to a point where it becomes very hard to handle from an editor's or application's perspective. In a second approach, Stand-off markup can be used to separate the source from the related information. This means that the source's XML content is indexed on a character basis and the related information is stored in another file with pointers to the index-numbers. With this approach, the simultaneous display of multiple annotation hierarchies is possible. A disadvantage of this solution is the rigidity of the index. If characters in the source file are changed after



Figure 1: The Regestatext as a chain of word nodes

indexing and enriching the material, all indexes are worthless as long as the whole document is not re-indexed. There is currently no editing tool available for solving this problem that also takes basic aspects of usability into account.

3 New perspectives for digital editions

3.1 The text as a chain of nodes

My paper at <http://mittelalter.hypotheses.org> [4] proposes graph-based digital editions as a possibility to handle multiple annotation hierarchies. Figure 1 shows an example from this paper where a record from the Regesta Imperii database (www.regesta-imperii.de) was modeled in the Neo4j graph database. As a first step, the text is encoded as word nodes connected by edges.

3.2 Adding annotations to the graph

The second step is to append annotation information. Figure 2 shows overlapping markup of a reference to another database record, represented by the yellow node, and another annotation, represented by the green node.

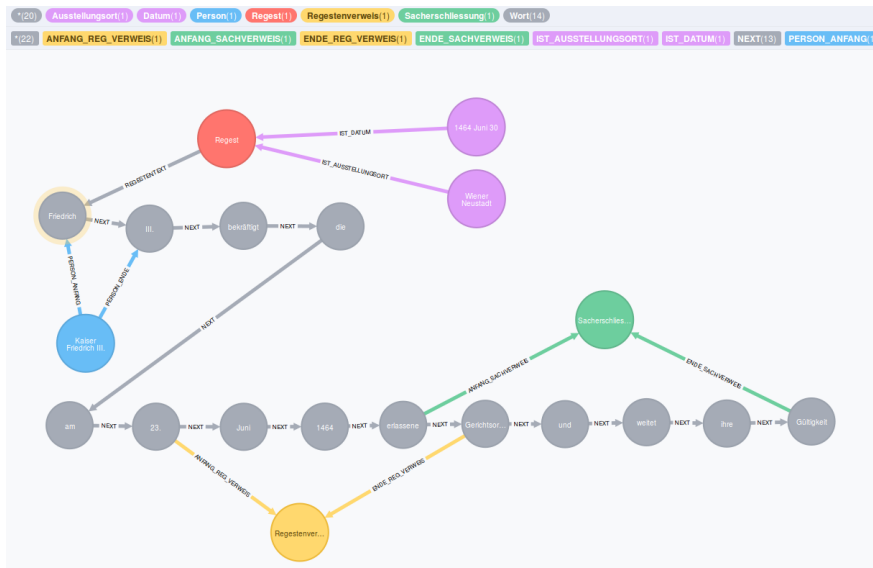


Figure 2: The Regestatext with added annotations

3.3 TEI-XML in a graph database

As mentioned initially, TEI-XML represents a standard for encoding sources in digital editions. If graph-based digital editions become a realistic alternative for digital editions, the graph database must be able to handle TEI-XML data. For this purpose a GitHub project for importing an TEI-XML document into Neo4j was started.

For a first feasibility study a TEI-XML document containing a Latin letter from the 17th century with a German abstract of the text has been used. The file includes additional information about the persons, places and other entities in the letter. In an elaborated virtual research environment this information would of course be separated from the text of the letter but for demonstrational purposes of the study this information was also added to the example XML file.

Listing 1 presents some parts of the identifying information about persons, places, and entities are listed.

Listing 1: Persons, places and items in the xml file

```

<particDesc>
  <listPerson>
    <person xml:id="P006">
      <idno type="gnd">http://d-nb.info/gnd/119357100</idno>
      <persName type="reg">
        <surname>Lubieniecki</surname>
        <forename>Stanisław</forename>
      </persName>
    </person>
    ...
    <place xml:id="DE-HAM">
      <placeName type="reg">Hamburg</placeName>

```

```

        <idno type="gnd">http://d-nb.info/gnd/4023118-5</idno>
    </place>
    ...
    <textClass>
    <keywords>
    <list>
    <item xml:id="C_1664_W1">
    <idno type="iau">C/1664 W1</idno>
    <label>C/1664 W1</label>
    <rs type="objectType" key="astro_comet"/>
    </item>
    ...
    <item xml:id="Venus">
    <idno type="gnd">http://d-nb.info/gnd/4062527-8</idno>
    <label>Venus</label>
    <rs type="objectType" key="astro_planet"/>
    </item>
    ...
    </item>
    <item xml:id="hypothesis">
    <idno type="hyp">hypothesis</idno>
    <label>hypothese about comets</label>
    <rs type="objectType" key="hypothesis"/>
    </item>
    ...
    </list>
    </keywords>

```

Listing 2 shows the German abstract of the Latin letter. Within the abstract, persons, places and entities are identified and linked to the other entities shown in the first listing.

Listing 2: The abstract of the letter

```

<abstract>
<p><name ref="#P007">Langius</name> entschuldigt sich, dass er
die Anfrage <name ref="#P006">Lubienietzki</name> zu dem
Kometen des Jahres 1664 verspätet beantwortet: Den
<rs ref="#C_1664_W1">Kometen</rs> habe er im ausgehenden
Dezember 1664 und Anfang Januar 1665 im <name key="#Hya">
Sternbild Hydra</name> beobachtet, und ein ähnliches Phänomen
sei im ausgehenden März bzw. Anfang April in <name ref="#Peg">
Pegasus</name> und <name ref="#And">Andromeda</name> erschienen.
In Anlehnung an <name ref="#P001">Demokrit</name> äußert <name
ref="#P007">Langius</name> die <rs ref="hyp">Hypothese, dass
Kometen aus Atomparkeln gebildet werden: Sie entstünden im
Kegelschatten der <name ref="#Terra">Erde</name>
und würden erst dann sichtbar, wenn sie aus diesem hervor- und
in das Licht der <name ref="#Sol">Sonne</name> hineinträten</rs>.
Seine Hypothese veranschaulicht <name ref="#P007">Langius</name>
...

```

Figure 3 shows a part of the abstract of the imported XML file and the connected annotations.

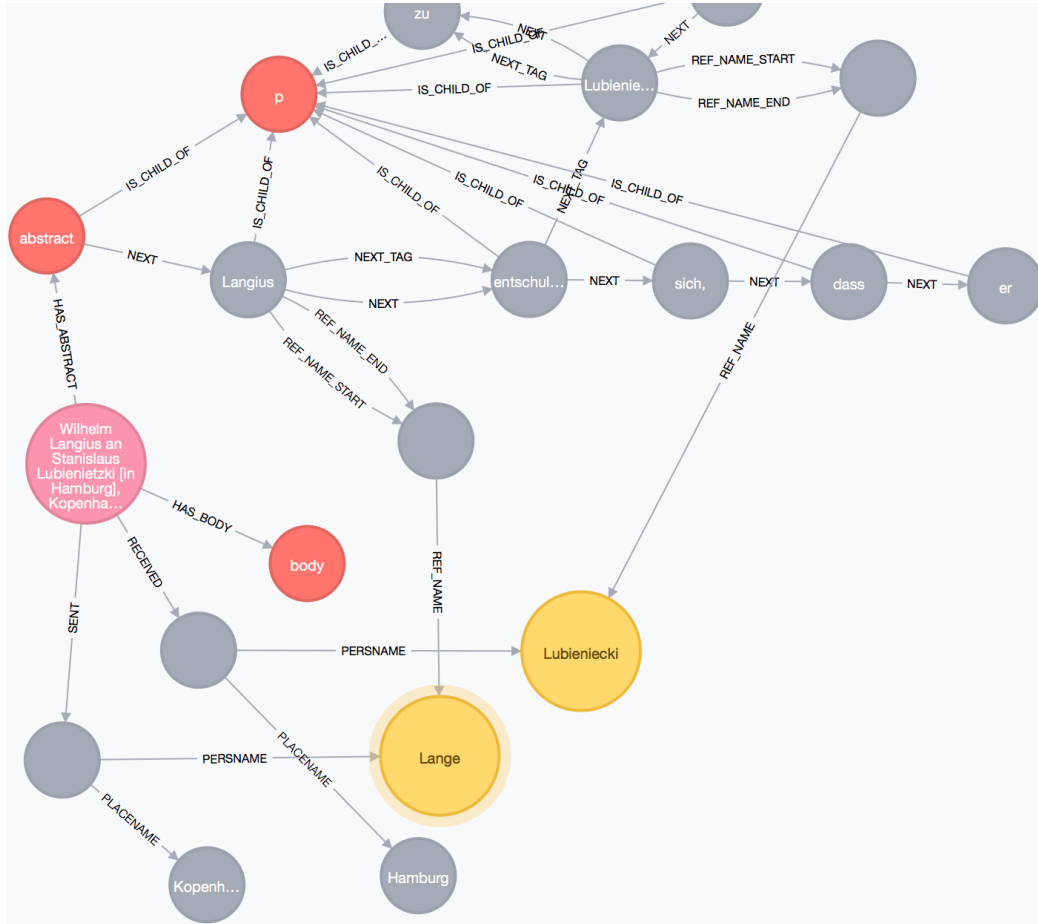


Figure 3: The abstract of the letter with linked annotations

3.4 The root of the document

If we select an XML-like approach to the graph the "xml-root" of the document is represented by the pink node (figure 3). All other informations can be reached from that node but they are not ordered in an hierarchical way as it would be in an XML-Document. Four nodes are directly connected to the root node. Those will be discussed below.

3.5 Hyperedges for the processes of sending and receiving

The two empty nodes linked to the root node are hyperedges.¹ One hyperedge represents the sending information of the letter with a combined link to the author of the letter, and the

¹For the concept of a hyper-edge: <http://neo4j.com/docs/stable/cypher-cookbook-hyperedges.html>.

place from which it was sent. The other hyperedge represents the recipient information with a combined link to the person who received the letter and the place where it was received. This construct facilitates the traversal of the graph if there are many letters in the database.

3.6 From the root to the abstract and the body text paragraphs

Beside the root node there are two other linked nodes next to the hyperedge nodes. One leads to the beginning of the abstract, the other leads to the Latin text of the letter. Following the HAS_ABSTRACT-edge the abstract node can be reached. From this node, following the NEXT-edge we can get the abstract word by word. You can explore the abstract by double-clicking on the word nodes and see the connected nodes for the annotation of persons and places. All word-nodes of the abstract are directly connected to a p-node too, which represents their belonging to the paragraph represented by the p-node, while the order of the words is represented by the NEXT edges.

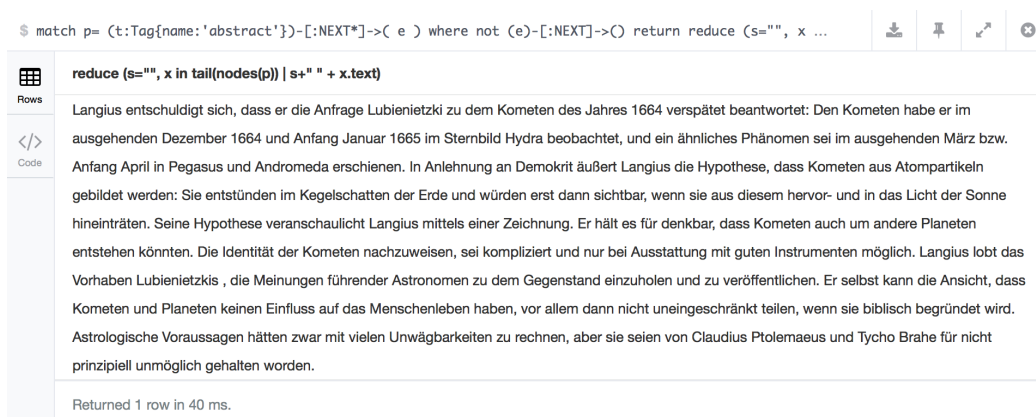
3.7 Traverse the graph to get the abstract text

Besides the possibility of exploring the abstract in the graph, a Cypher query can be used to get the text of the abstract as a result. Listing 3 shows the query and figure 4 shows the results in the Neo4j frontend.

Listing 3: Get the text of the abstract

```
// Get Abstract Text
MATCH p= (t:Tag{name:'abstract'})-[:NEXT*]->( e )
WHERE NOT ( e )-[:NEXT]->( )
RETURN REDUCE (s="", x in tail(nodes(p)) | s+" " + x.text)
```

Let us now examine the cypher query. With the MATCH-p-statement we start a traversal query starting from a tag-node with the name "abstract" connected to the next node with NEXT-edges and to the following node with NEXT-edges, until we come to a node that has no departing NEXT-edge. The 'reduce' function reduces the result to one single line of text only. The rest of the query converts the resulting chain of nodes to the represented words.



match p= (t:Tag{name:'abstract'})-[:NEXT*]->(e) where not (e)-[:NEXT]->() return reduce (s="", x in tail(nodes(p)) | s+" " + x.text)

reduce (s="", x in tail(nodes(p)) | s+" " + x.text)

Langius entschuldigt sich, dass er die Anfrage Lubienietzki zu dem Kometen des Jahres 1664 verspätet beantwortet: Den Kometen habe er im ausgehenden Dezember 1664 und Anfang Januar 1665 im Sternbild Hydra beobachtet, und ein ähnliches Phänomen sei im ausgehenden März bzw. Anfang April in Pegasus und Andromeda erschienen. In Anlehnung an Demokrit äußert Langius die Hypothese, dass Kometen aus Atompartikeln gebildet werden: Sie entstünden im Kegelschatten der Erde und würden erst dann sichtbar, wenn sie aus diesem hervor- und in das Licht der Sonne hineinrätren. Seine Hypothese veranschaulicht Langius mittels einer Zeichnung. Er hält es für denkbar, dass Kometen auch um andere Planeten entstehen könnten. Die Identität der Kometen nachzuweisen, sei kompliziert und nur bei Ausstattung mit guten Instrumenten möglich. Langius lobt das Vorhaben Lubienietzki, die Meinungen führender Astronomen zu dem Gegenstand einzuholen und zu veröffentlichen. Er selbst kann die Ansicht, dass Kometen und Planeten keinen Einfluss auf das Menschenleben haben, vor allem dann nicht uneingeschränkt teilen, wenn sie biblisch begründet wird. Astrologische Voraussagen hätten zwar mit vielen Unwägbarkeiten zu rechnen, aber sie seien von Claudius Ptolemaeus und Tycho Brahe für nicht prinzipiell unmöglich gehalten worden.

Returned 1 row in 40 ms.

Figure 4: The exported text of the abstract in the Neo4j frontend

3.8 The hierarchy of the TEI file in the graph database

Querying for the tag-node the Neo4j frontend shows the structure of the imported TEI-XML as shown in figure 6. Starting from the pink root-node you can follow the abstract-node to the abstract or the body-node to the body of the TEI-XML file. All paragraphs in the body are connected to the body-node and their own hierarchy is represented by NEXT_TAG edges. One IS_CHILD_OF-edge leads to the head-node with all the entity information connected to it.

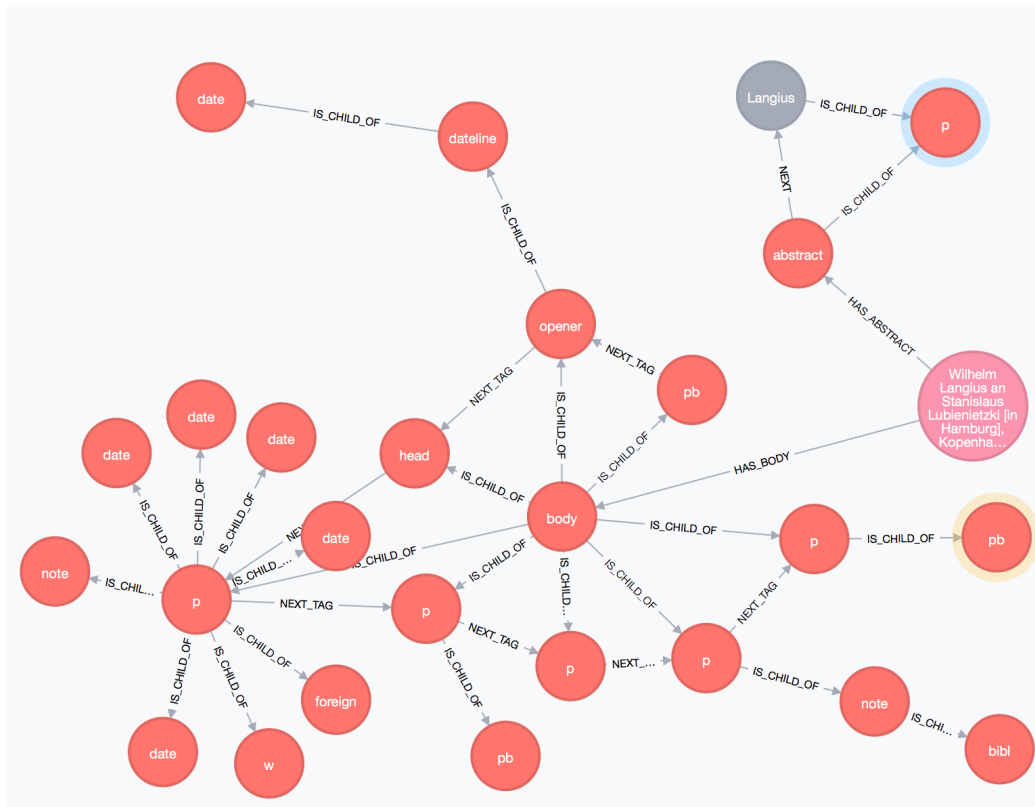


Figure 5: The paragraph structure of the TEI file in the neo4j-frontend

3.9 Example of further annotation and exploration perspectives

Beside the planets and comets a scientific hypothesis is mentioned in the letter and encoded in the XML document as an item.² In the following example, the statement of the hypothesis will be connected with the parts of the text in the abstract in which it is explained, as well as with the planets and persons mentioned. Modelling Digital Editions in Graph-Databases changes the way pieces of information are explored. My paper [3] explains that most search cases are with concrete interest need (cin-cases) were as explorational approaches usually are not supported. So starting point for queries are search-interfaces on the web. The usual starting point for a

²The last item of the list in listing 1 represents the hypothesis which is annotated in the abstract.

query in a graph is an entity. Coming from that entity all connected nodes with the different edges are explored. So not only the way how things are encoded will change [6] but also the way things are queried.

Listing 4: Create annotations in the graph database

```
// Create Edges from text nodes to the hypothesis node
MATCH (from {id:'hypothesis'}), (to {text:'Hypothese'})
    CREATE from-[:START_HYPOTHESIS]->to;
MATCH (from {id:'hypothesis'}), (to {text:'hineinträten'})
    CREATE from-[:END_HYPOTHESIS]->to;

// Create edges to the presenter of the hypothesis
MATCH (from {id:'hypothesis'}), (to {surname:'Lange'})
    CREATE from-[:PRESENTER_OF_HYPOTHESIS]->to;

// Things covered by the hypothesis
MATCH (from {id:'hypothesis'}), (to {label:'Sol'})
    CREATE from-[:IS_ABOUT]->to;
MATCH (from {id:'hypothesis'}), (to {label:'Terra'})
    CREATE from-[:IS_ABOUT]->to;
MATCH (from {id:'hypothesis'}), (to {label:'C/1664 W1'})
    CREATE from-[:IS_ABOUT]->to;
```

Figure 6 shows the results in the graph database. The hypothesis-node is connected via IS_ABOUT edges to the mentioned items of the hypothesis, the author of the hypothesis, and directly to the part of the abstract in which they are mentioned.

4 Conclusions

In this paper a first glimpse of the great potential of graph-based digital editions was demonstrated even if we must concede that we lack of powerful tools und user-interfaces. The conceptual development is still in its early stages, but by using graph databases we could:

- seamlessly include multiple annotation hierachies within one dataset
- handle a scenario where several researchers are working together on the same source with a transparent separation of their different markup needs
- query the various annotational layers within one query language and
- changes to the source text will neither destroy the word-order nor the meaning of existing annotations.

Many of the discussed issues could certainly also be solved with specific markup strategies or with Stand-off markup, but this solution usually involves great effort and often produces XML that is very hard to handle and not human-readable at all. "Ease of use" has always been a major reason cited for the choice of XML as a (standard) format. Due to the complexity of future digital editions, there is no reason we should not take the step to graph-based digital editions. The contents of a graph database – its graphs – are human-readable and can represent very complex annotational hierarchies, while also removing the worry about boundary demarcation. Moreover everything can be exported to XML or graphml files for long-term archival purposes.

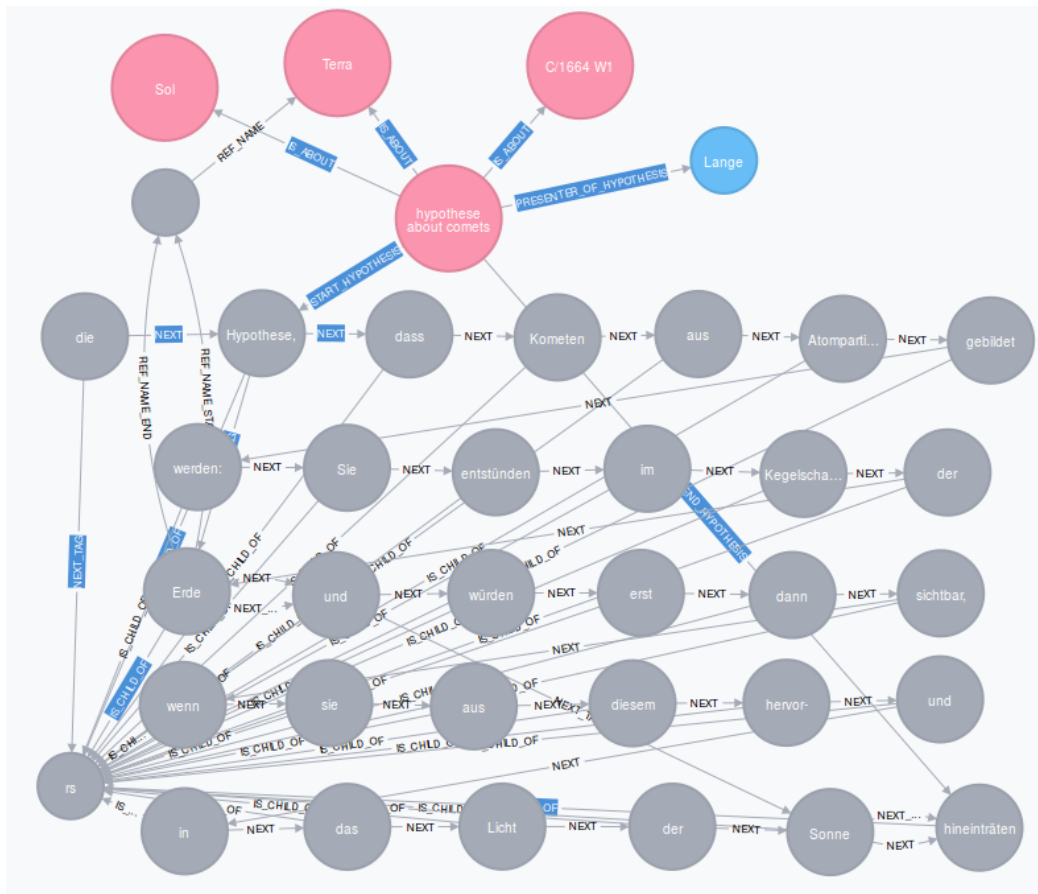


Figure 6: The exported text of the abstract in the Neo4j frontend

This becomes possible in a model which brings the way things are structured in the human mind and the database structure closer together. Graph-based digital editions are the next important step for the Digital Humanities, as the editing disciplines acquire a very flexible, easy-to-use but powerful tool. To achieve this goal it is important that we find standardized ways for modelling events, actions, political aims, etc., in order to explore and compare different sources from different contexts. The second important task is the programming of user interfaces which can be easily used and adjusted to the researchers needs and can explore more than one datasource on the internet. Developing this will be a big challenge.

5 Acknowledgments

The Neo4j-TEI-Importer was developed by Stefan Armbruster from Neo Technology (stefan.armbruster@neotechnology.com) and I want to thank him for his help. He published the importer on GitHub (<https://github.com/sarmbruster/neo4j-tei-importer>). Thanks also to Sascha Kaufmann from the Digital Humanities Department of the University of Bern for fruitful discussions on how to model XML data in a graph database environment.

References

- [1] TEI Consortium. P5: Richtlinien für die Auszeichnung und den Austausch elektronischer Texte, 2016. <http://www.tei-c.org/release/doc/tei-p5-doc/de/html/ref-milestone.html>, last viewed April 2016.
- [2] Daniel Jettka. Repräsentation, Verarbeitung und Visualisierung multipler Hierarchien mit XStandoff und XSLT, 2011. http://www.daniel-jettka.de/pdf/Multiple_Hierarchien_mit_XStandoff_und_XSLT.pdf, last viewed April 2016.
- [3] Andreas Kuczera. Digitale Perspektiven mediävistischer Quellenrecherche, in: Mittelalter. Interdisziplinäre Forschung und Rezeptionsgeschichte, 2014. <http://mittelalter.hypotheses.org/3492>.
- [4] Andreas Kuczera. Graphbasierte digitale Editionen, in: Mittelalter. Interdisziplinäre Forschung und Rezeptionsgeschichte, 2016. <http://mittelalter.hypotheses.org/7994>.
- [5] Sperberg-McQueen, C M; Huitfeldt, Claus. GODDAG: A Data Structure for Overlapping Hierarchies. Lecture Notes in Computer Science (2023), 2000. 139-160. <http://cmsmcq.com/2000/poddp2000.html>, last viewed April 2016.
- [6] Amir Zeldes Thomas Krause. ANNIS3: A new architecture for generic corpus query and visualization. Digital Scholarship in the Humanities, Vol. 31, No. 1, 2016. 118-139. <http://dsh.oxfordjournals.org/content/31/1/118>.