



## Challenges in Modelling a Richly Annotated Diachronic Corpus of German

Stefanie Dipper, Lukas Faulstich, Ulf Leser, Anke Lüdeling  
Humboldt-Universität zu Berlin, Germany

Workshop on XML-based Richly Annotated Corpora  
Lisbon, Portugal, 29th May 2004



### outline

- goals, current situation, project description
- requirements
- implementation concept
  - system architecture
  - data model
  - import/export



### goal

- diachronic corpus of German, Old High German (800) to Modern German ( $\approx$ 1900) for linguistic, philological and historic research
- current situation: a lot of digitized texts, but
  - different (mostly implicit) quality standards (source, diplomaticity)
  - different formats (WordPerfect, WordCruncher, XML, ...)
  - different header structures (if any)
  - different positional or structural annotation (if any)
  - unequal coverage and different corpus composition for the language stages
  - availability sometimes problematic, no common search tools

## the initiative

- linguists, philologists, corpus linguists, computer scientists from 15 German universities, international cooperation
  - 5 language groups + architecture group
  - grant application submitted, planned duration 7 years
  - pilot project for corpus architecture at Humboldt-Universität, Berlin
  - size after 7 years
    - core corpus: 40 M words
    - extension corpus: 60 M words
- 

## requirements - standardisation

- standardisation
    - common quality standard(s)
      - source: original or edited text
      - diplomaticity
    - common header structure – extension of TEI/XCES
      - dialect
      - text type/genre
      - paleography/codicology
    - common structural annotation
      - graphic
      - logical
      - conflicting hierarchies
- 

## requirements - standardisation

- common positional annotation
    - levels
    - tagsets
  - lemmatisation
    - within language group - normalisation
    - across language groups – hyperlemma
    - multi-linguality
    - alignment
- 

## requirements - flexibility

- different texts may have different annotation layers
    - every text (extension corpus): header information, minimal structural annotation
    - core corpus: additionally lemmatisation, pos-tags
    - presentation corpus: aligned facsimiles, sound files
    - multi-modality
    - in addition: texts may have more annotation layers (syntax, information structure, narratological information, paleographical information, ...) – the tagsets and guidelines for each layer are standardised
  - texts and annotation layers can be added at any time
-

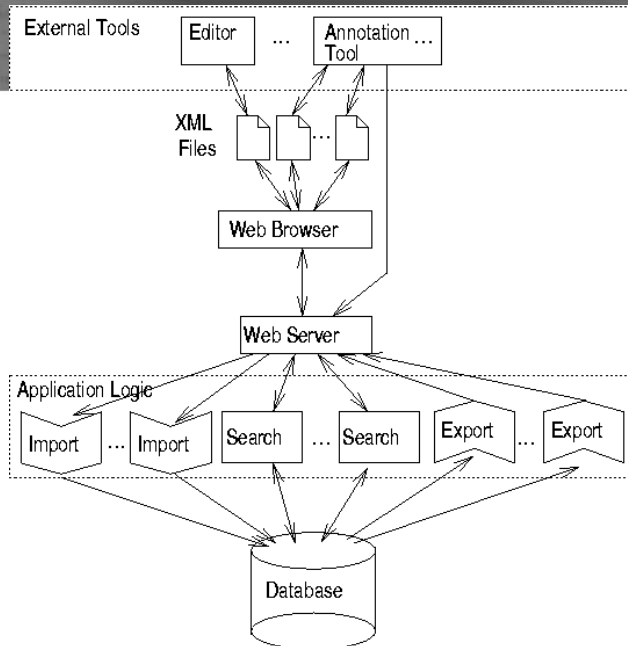


## implementation aspects (in this talk)

- system architecture
- data model
- import/export (transformation)

## system architecture

- corpus stored in a relational database (RDBMS)
- web-based client-server architecture
- external client tools



## data model

requirements:

- open set of annotation layers
- support of conflicting hierarchies
- complex annotations
  - alignments
  - cross-references
  - meta-annotations

alternatives:

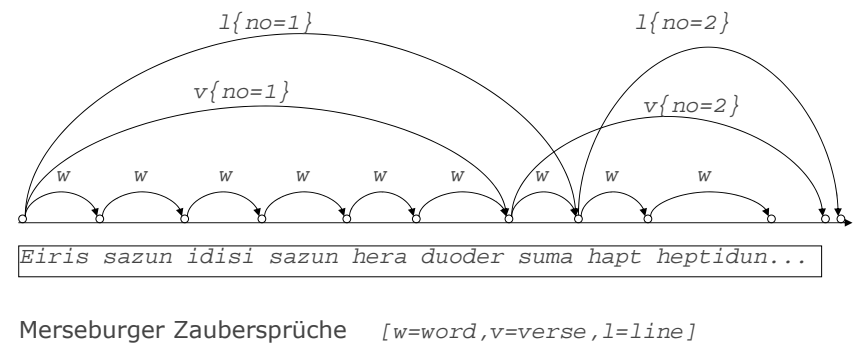
- annotation graph model (AG)
- ordered directed acyclic graph (ODAG) model
  - NITE object model
  - **DDD data model**

## annotation graph model

```
Session=(
  signals: ID-> Signal,
  annotations: Set<Arc>)
Arc= (
  name: Name,
  start,end: Real,
  attributes: Name->String)
```

- parent-child relationships are expressed implicitly via containment
- annotation layer determined by arc name

## annotation graph model: example



## annotation graph model: discussion

- + efficiently implementable in RDBMS
- + annotation layers are independent
  - > support for conflicting hierarchies
- implicit dominance relation can cause ambiguities
- alignments: implicit via equal attribute values

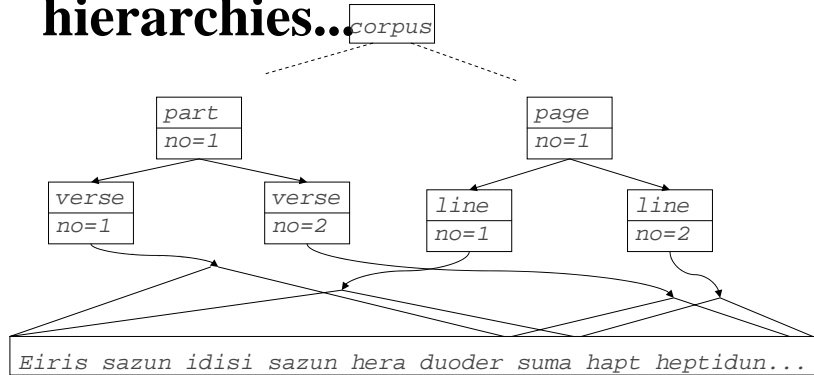
## ODAG data model: nite object model (NOM)

```
Session=(signals: ID-> Signal, roots: Set<Node>)
Node= (
  name: Name,
  attributes: Name->String,
  children: Node*,
  interval: (start,end: Real)?)
```

constraints:

- acyclicity
- parent interval must contain child intervals which must be in textual order, without overlaps

# ODAG data model: example: conflicting hierarchies...



# additional requirements of DDD:

- whole corpus as a graph
  - multiple independent texts (signals)
- complex annotations
  - alignments
  - cross references

⇒ extension of ODAG data model

# the DDD data model

*Corpus* = (texts: ID -> String, root: Node)

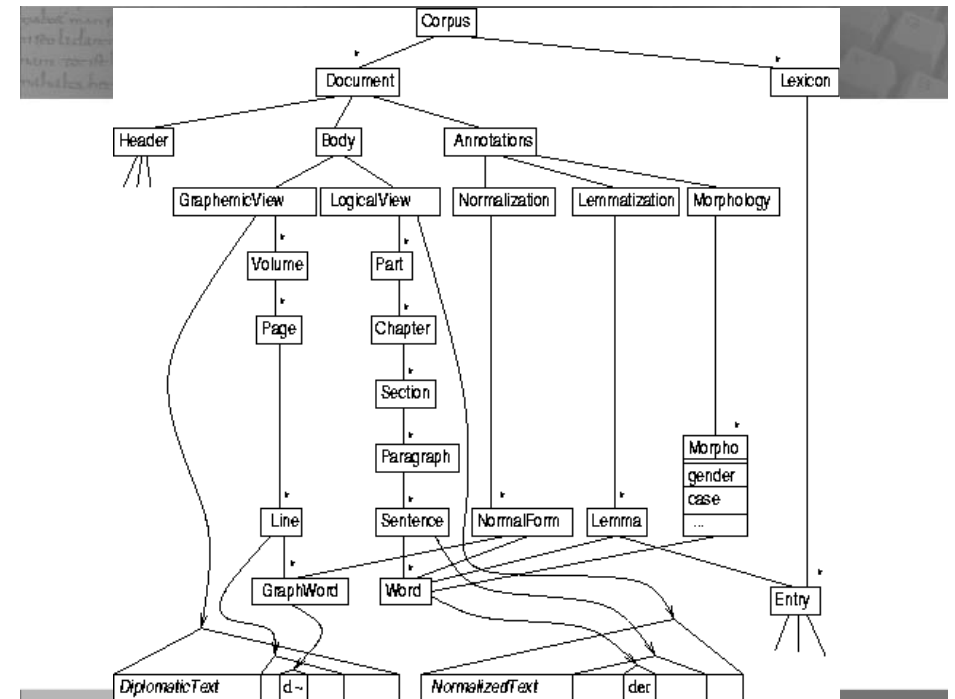
*Node* = Element | Span

*Element* = (  
 name: Name,  
 attributes: Name -> String,  
 children: Node\*,  
 span: Span?)

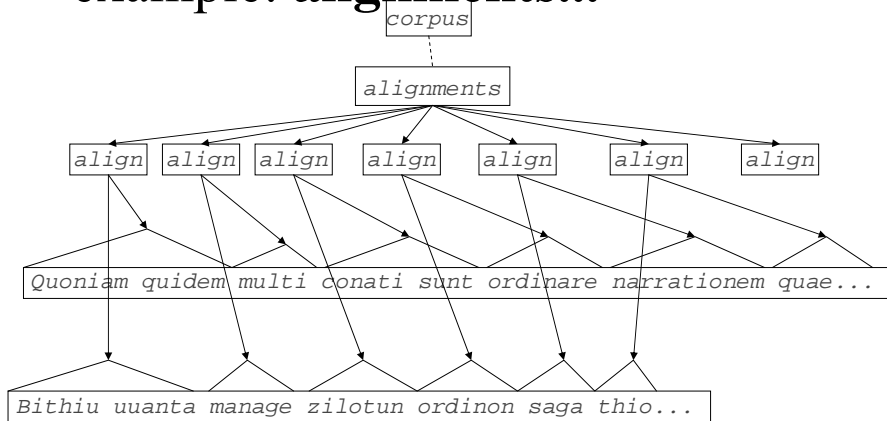
*Span* = (text: ID, start, end: Real)

constraints:

- acyclicity
- parent spans must include child spans, which must be in textual order without overlaps



## the DDD data model: example: **alignments**...



sentence 1 of Tatian, Gospel Harmony

## import/export methods

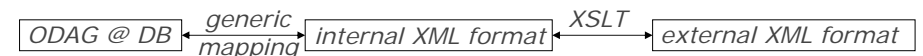
## import/export

- export from database to XML for text presentation
- export from database to XML for external (annotation) tools
- import of XML produced by ext. tools into database

XML document (DOM tree) is a special case of an ODAG  
⇒ import/export = transformation of ODAGs

## transformation of ODAG = generic mapping + XSLT

- ODAGs can be represented as XML documents (redundant representation, node IDs for identification): internal XML format
- generic mapping could be done within the database
- XSLT is expressive enough to satisfy most requirements
  - support of TEI/XCES-based exchange format
  - XHTML presentation formats used on the Web-site



## need for a high-level transformation language

- selection should be done as early as possible (i.e., within database)
- joins can be done more efficiently inside the database
- encoding/decoding methods for conflicting hierarchies (milestones, fragmentation, virtual joins) are quite complex -> should be offered as primitives

## summary

- goal: a diachronic corpus of German
- maximum flexibility and at the same time maximum consistency
  - common header structure, common quality standard, smallest unit character, different annotation layers, standardisation within each annotation layer
- implementation
  - web-based client server architecture on top of RDBMS
  - data model: ODAGs
  - import/export:
    - generic mapping + XSLT: possible, but inefficient
    - need for high level transformation language

<http://www.linguistik.hu-berlin.de/ddd/>



Digital  
Diachron  
Deutsch

A Historical Corpus of German