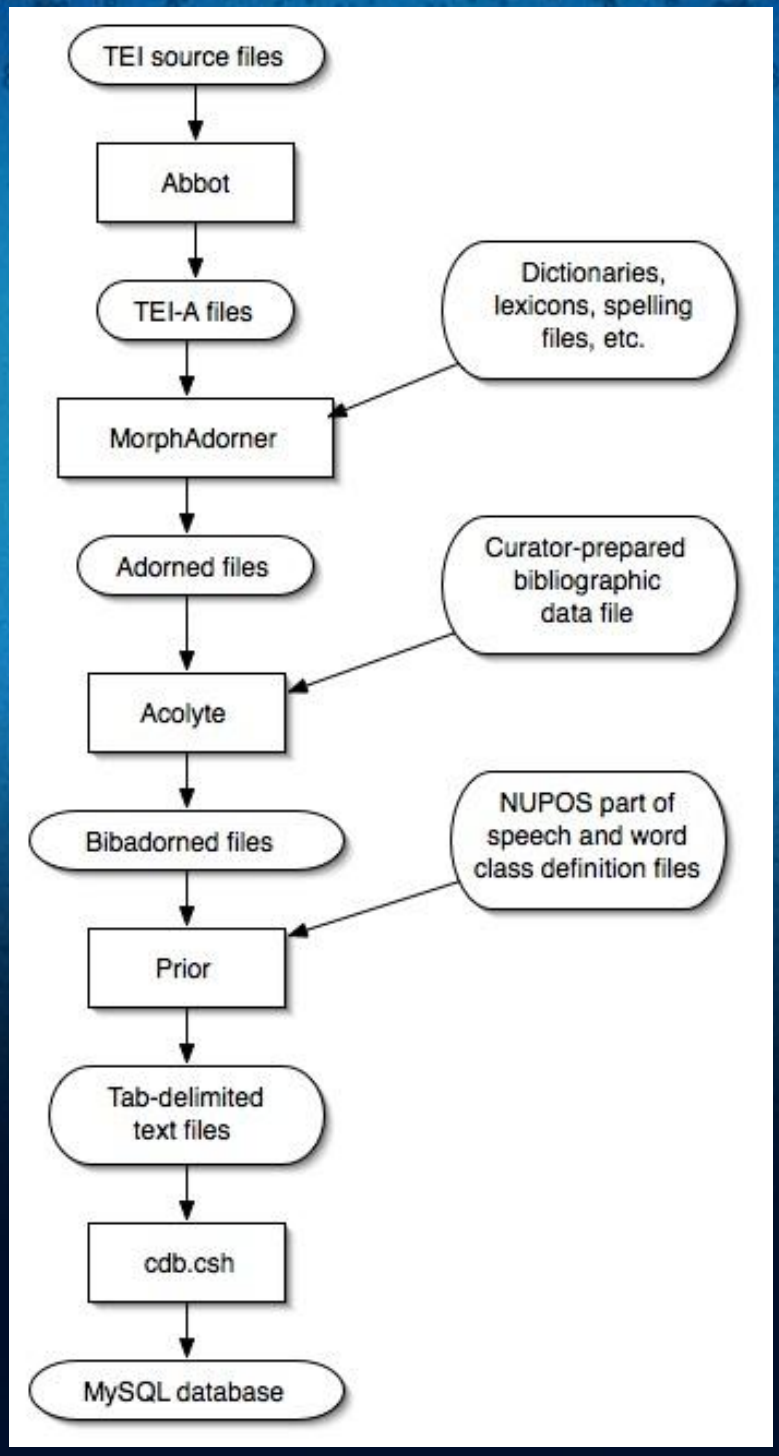
Monk Datastore Workflow

September 28, 2009


TEI source files
Abbot
TEI-A files
Dictionaries, lexicons, spelling files, etc.
MorphAdorner
Adorned files
Curator-prepared bibliographic data file
Acolyte
Bibadorned files
NUPOS part of speech and word class definition files
Prior
Tab-delimited text files
cdb.csh
MySQL database

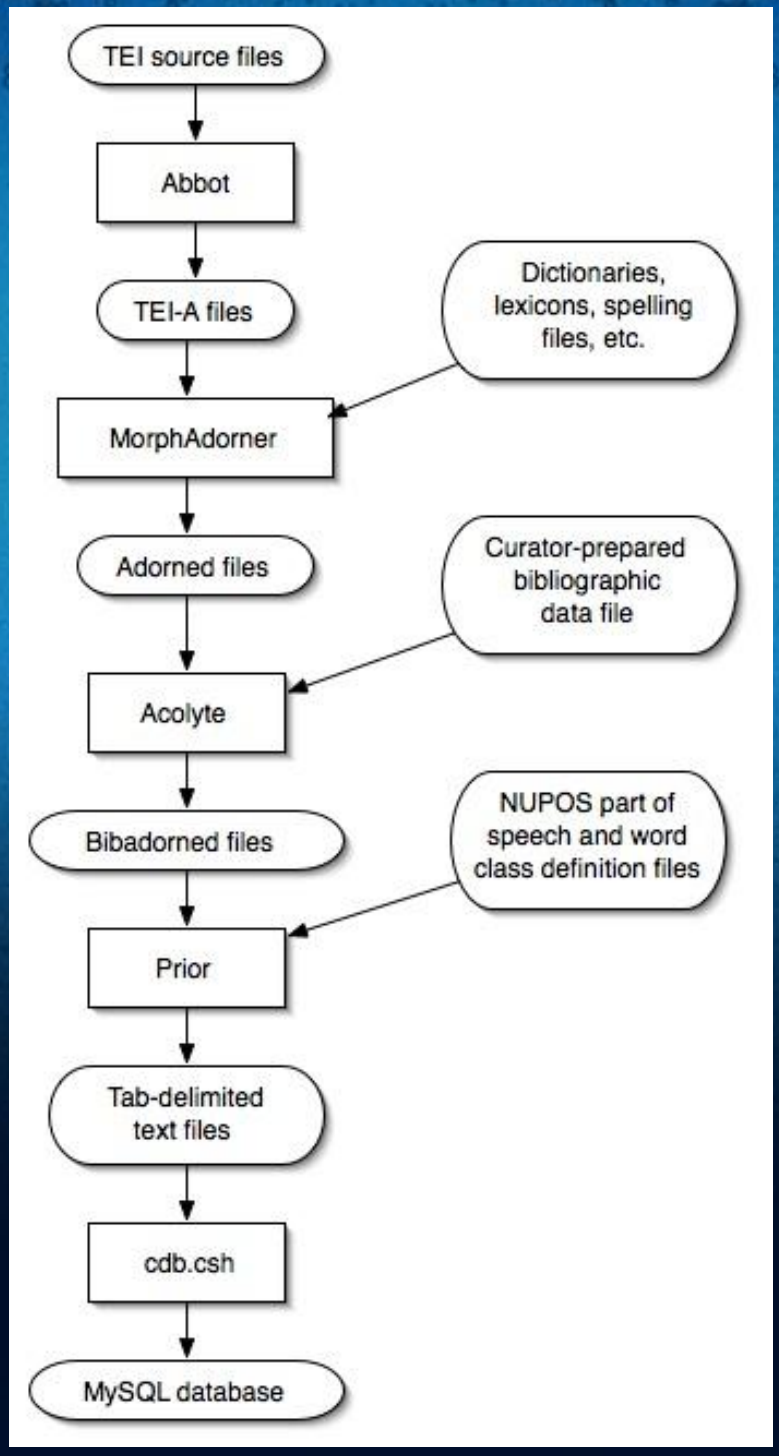
Seven Stages

1. Text selection
2. Text normalization
3. Morphological adornment
4. Bibliographic enhancement
5. Database input generation
6. Database creation
7. Model and programming interface creation

Stage One: Text Selection

- Documenting the American South (113 texts)
- Early American Fiction (111 texts)
- Eighteenth Century Fiction (1077 texts)
- Early English Books Online (691 texts)
- Nineteenth Century Fiction (250 texts)
- Shakespeare (42 texts)
- Wright American Fiction (301 texts)

Total: 806 authors; 2,585 texts; 151,516,845 words

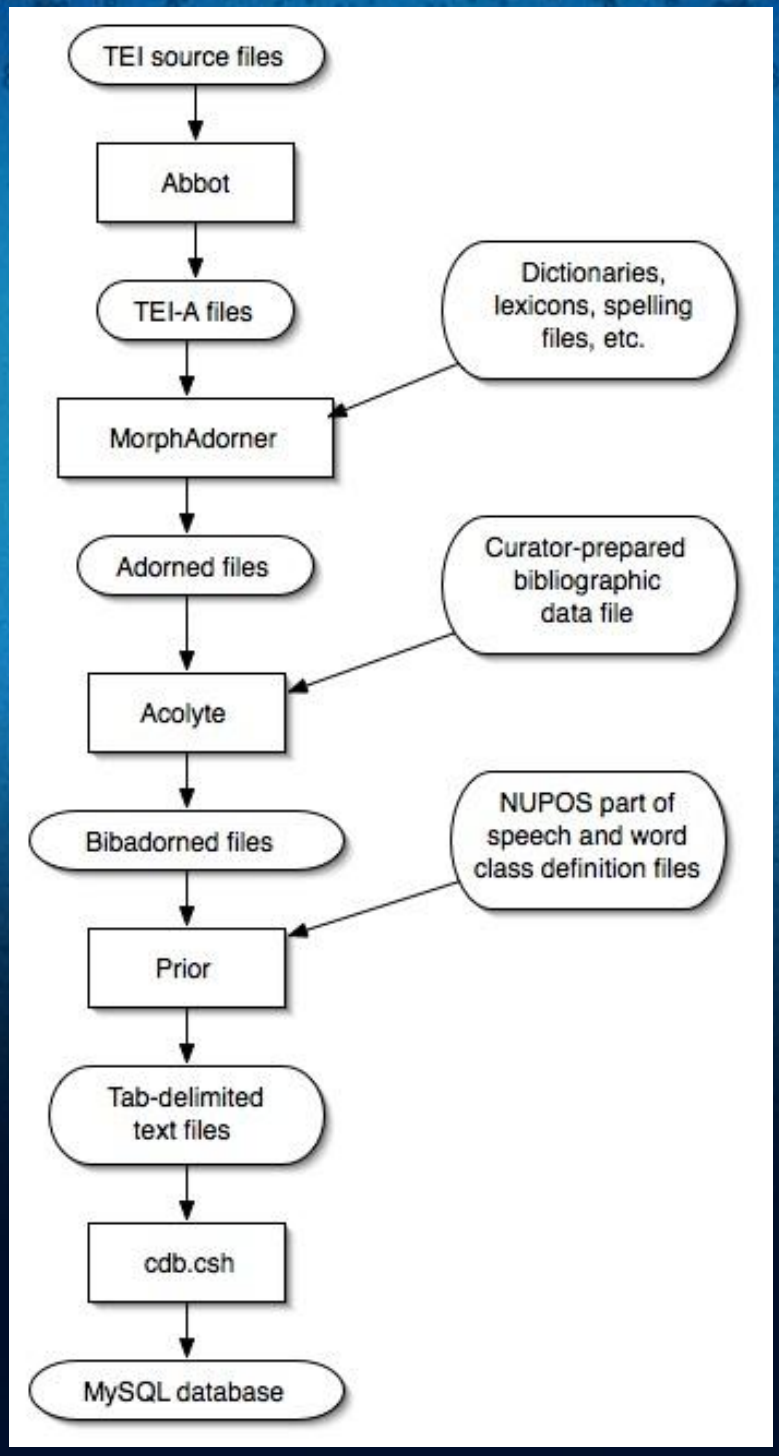

TEI source files
Abbot
TEI-A files
Dictionaries, lexicons, spelling files, etc.
MorphAdorner
Adorned files
Curator-prepared bibliographic data file
Acolyte
Bibadorned files
NUPOS part of speech and word class definition files
Prior
Tab-delimited text files
cdb.csh
MySQL database

Stage Two: Text Normalization

- Abbot
- Normalizes texts to TEI Analytics format (TEI-A)
- Developed by Brian Pytlik-Zillig and Stephen Ramsay
- Written as a combination of Unix shell scripts, Java, and XSLT

- Process:

1. Harvests schema/DTD from source files
2. Generates XSLT to transform source to TEI-A
3. Validates resultant TEI-A output files


TEI source files
Abbot
TEI-A files
Dictionaries, lexicons, spelling files, etc.
MorphAdorner
Adorned files
Curator-prepared bibliographic data file
Acolyte
Bibadorned files
NUPOS part of speech and word class definition files
Prior
Tab-delimited text files
cdb.csh
MySQL database

Stage Three: Morphological Adornment

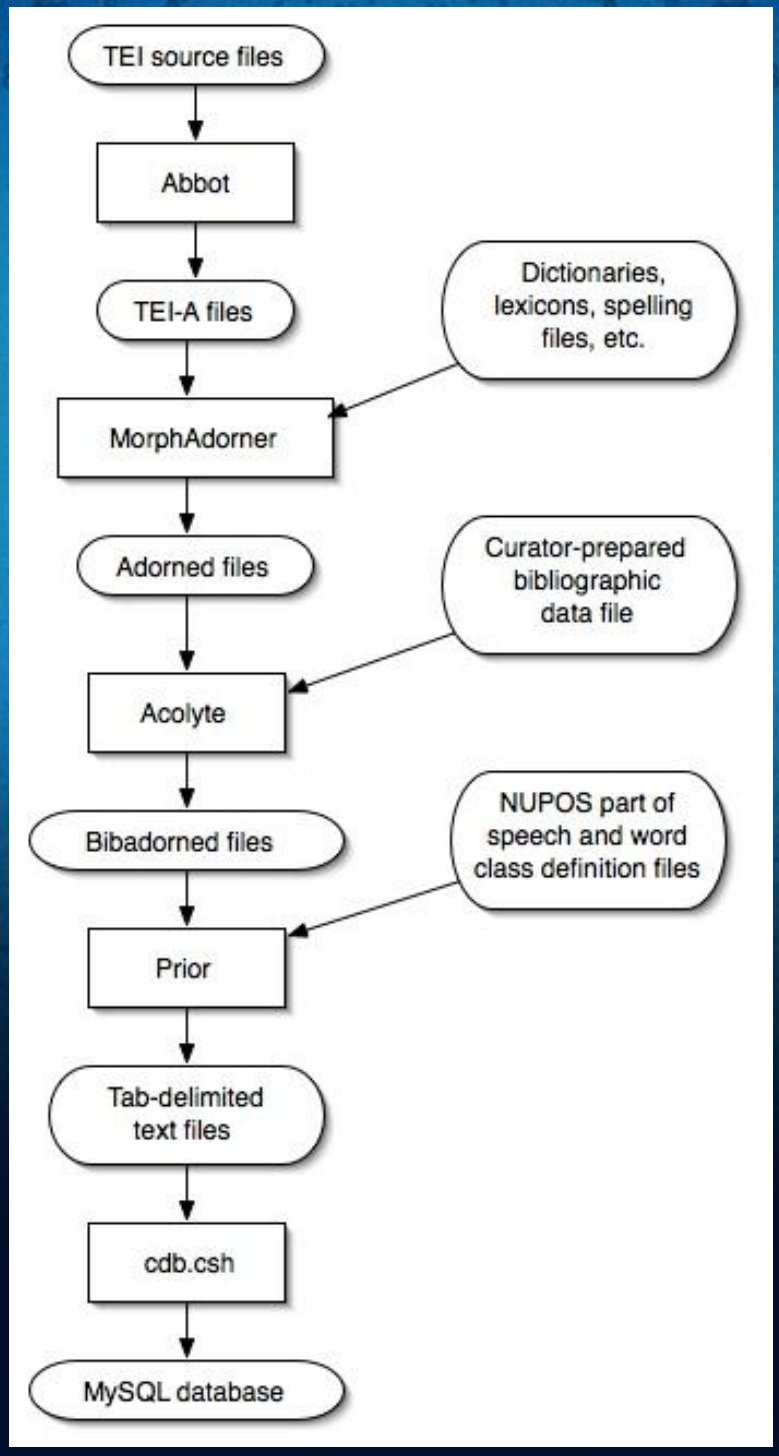
- MorphAdorner
- Adds morphological adornments to TEI-A files
- Developed by Philip R. Burns
- Written entirely in Java
- Provides many facilities beyond basic adornment
- Currently supports English language only

- Process:

1. Reads TEI-A encoded files
2. Splits text into sentences, words, and punctuation
3. Adds morphological adornments for each word
4. Outputs adorned TEI-A files

Sample Adorned Text Fragment

```
<p>
        <hi rend="sc(1)">
         <w eos="0" lem="Emma" pos="np1" reg="Emma" spe="Emma" tok="Emma" xml:id="
ncf0204-0000550" ord="40" part="N">Emma</w>
        </hi>
        <c> </c>
        <w eos="0" lem="Woodhouse" pos="np1" reg="Woodhouse" spe="Woodhouse" tok="
Woodhouse" xml:id="ncf0204-0000560" ord="41" part="N">Woodhouse</w>
        <w eos="0" lem="," pos="," reg="," spe="," tok="," xml:id="ncf0204-0000570" ord="42"
part="N">,</w>
        <c> </c>
        <w eos="0" lem="handsome" pos="j" reg="handsome" spe="handsome" tok="handsome"
xml:id="ncf0204-0000580" ord="43" part="N">handsome</w>
        <w eos="0" lem="," pos="," reg="," spe="," tok="," xml:id="ncf0204-0000590" ord="44"
part="N">,</w>
        <c> </c>
```


TEI source files
Abbot
TEI-A files
Dictionaries, lexicons, spelling files, etc.
MorphAdorner
Adorned files
Curator-prepared bibliographic data file
Acolyte
Bibadorned files
NUPOS part of speech and word class definition files
Prior
Tab-delimited text files
cdb.csh
MySQL database

Stage Four: Bibliographic Enhancement

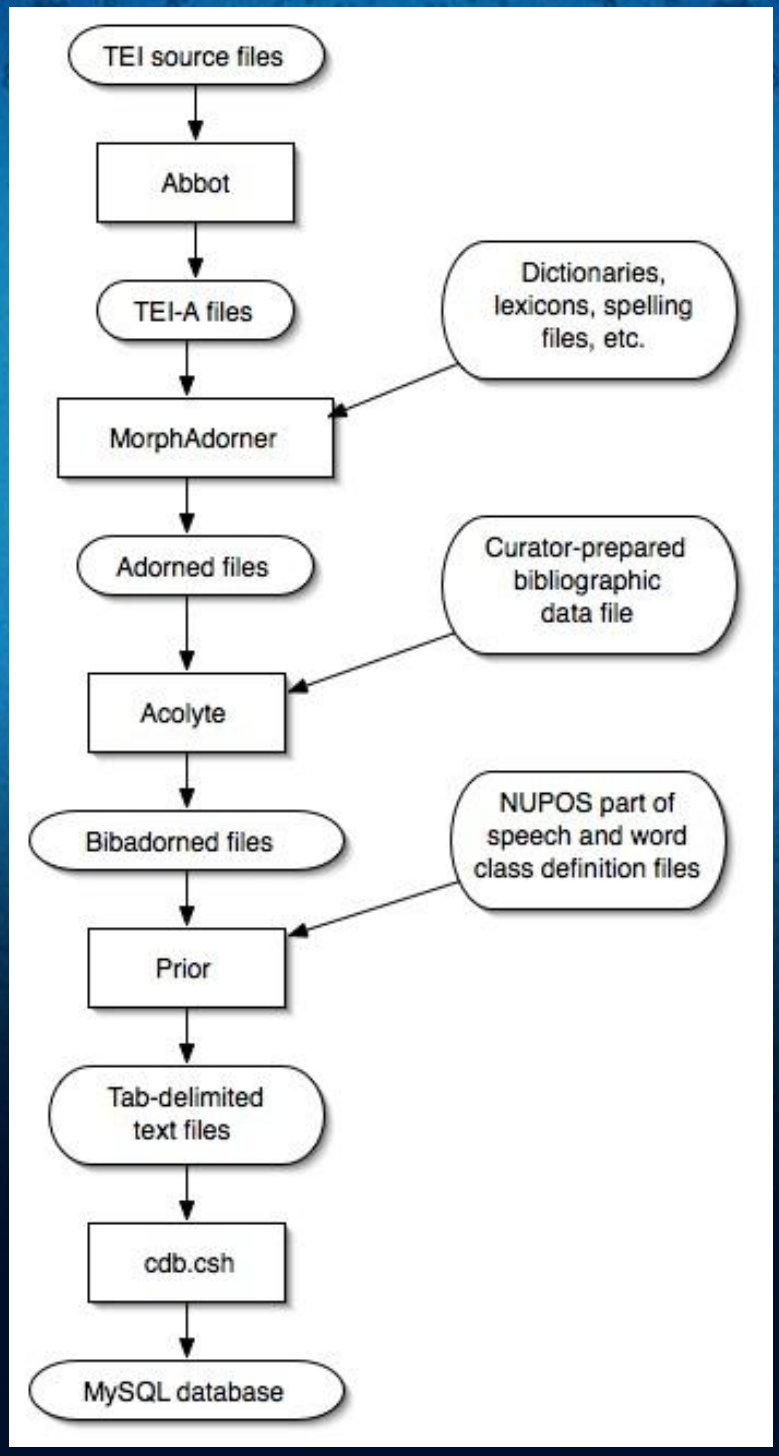
- Acolyte
- Developed by John Norstad
- Written in Java
- Input data created manually by curators

- Process:

1. Reads adorned TEI-A files
2. Reads curator-prepared bibliographic information
3. Adds curator-prepared bibliographic information to TEI-A files
4. Outputs "bibadorned" TEI-A files

Sample Bibadorned Header


```
<monkHeader xmlns="http://monk.at.northwestern.edu/ns/1.0">
<tag>ncf-ncf0204</tag>
<corpus>ncf</corpus>
<fileName>ncf0204</fileName>
<title>Emma</title>
<author>
<name>Austen, Jane</name>
<birthYear>1775</birthYear>
<deathYear>1817</deathYear>
<flourished>n/a</flourished>
<origin>British Isles</origin>
<gender>F</gender>
</author>
<circulationYear>1816</circulationYear>
<genre>fiction</genre>
<subgenre></subgenre>
<availability>restricted</availability>
</monkHeader>
```

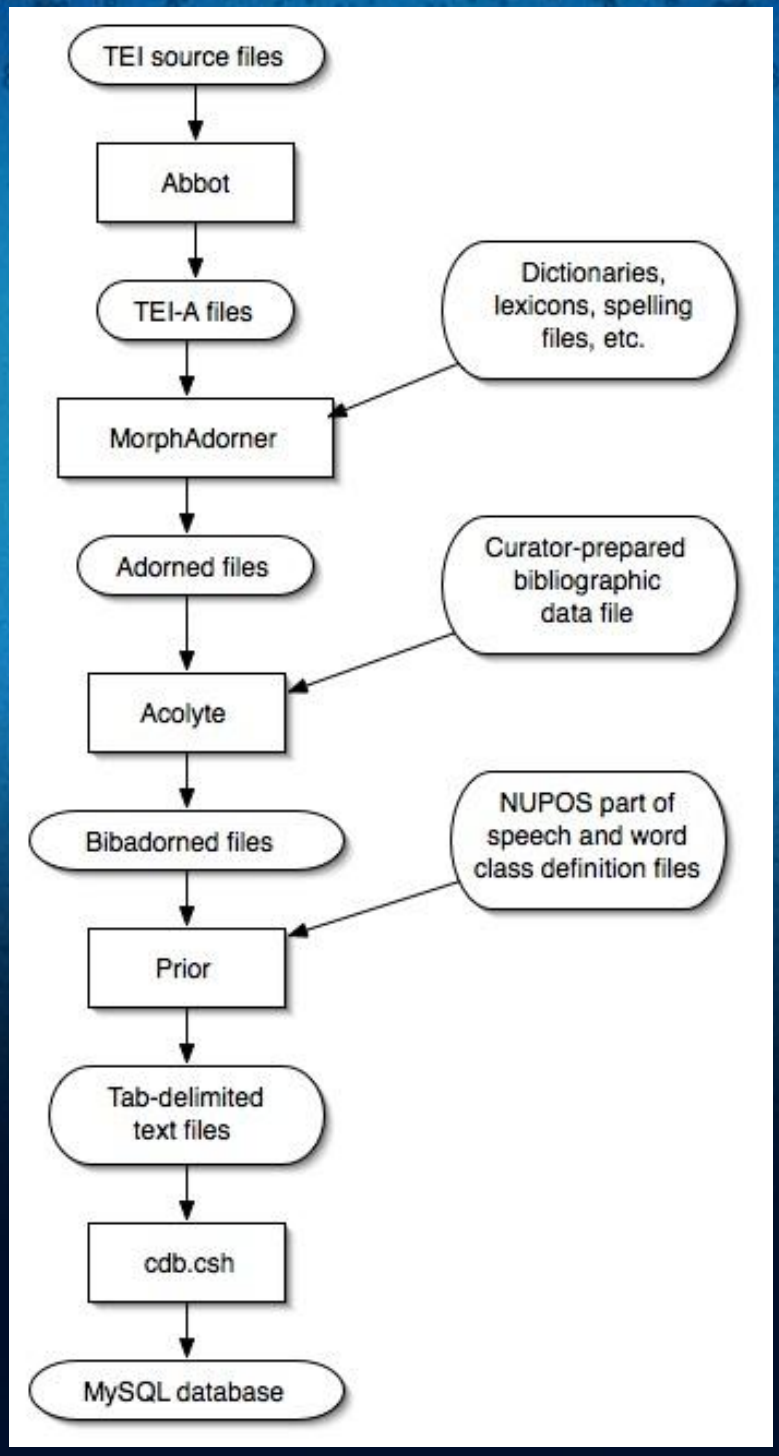

TEI source files
Abbot
TEI-A files
Dictionaries, lexicons, spelling files, etc.
MorphAdorner
Adorned files
Curator-prepared bibliographic data file
Acolyte
Bibadorned files
NUPOS part of speech and word class definition files
Prior
Tab-delimited text files
cdb.csh
MySQL database

Stage Five: Database Input Generation

- Prior
- Generates database input from bibadorned TEI-A files
- Developed by John Norstad
- Written in Java

- Process:

1. Reads files defining NUPOS parts of speech and word classes
2. Reads parameter file defining corpora names and file locations
3. Reads bibadorned files
4. Outputs MySQL database import format files


TEI source files
Abbot
TEI-A files
Dictionaries, lexicons, spelling files, etc.
MorphAdorner
Adorned files
Curator-prepared bibliographic data file
Acolyte
Bibadorned files
NUPOS part of speech and word class definition files
Prior
Tab-delimited text files
cdb.csh
MySQL database

Stage Six: Database Creation

- cdb.csh, a Unix script, creates a MySQL database by importing the files created by Prior.

- Build time: 29 hours, 23 minutes
- Database size: 179 gigabytes

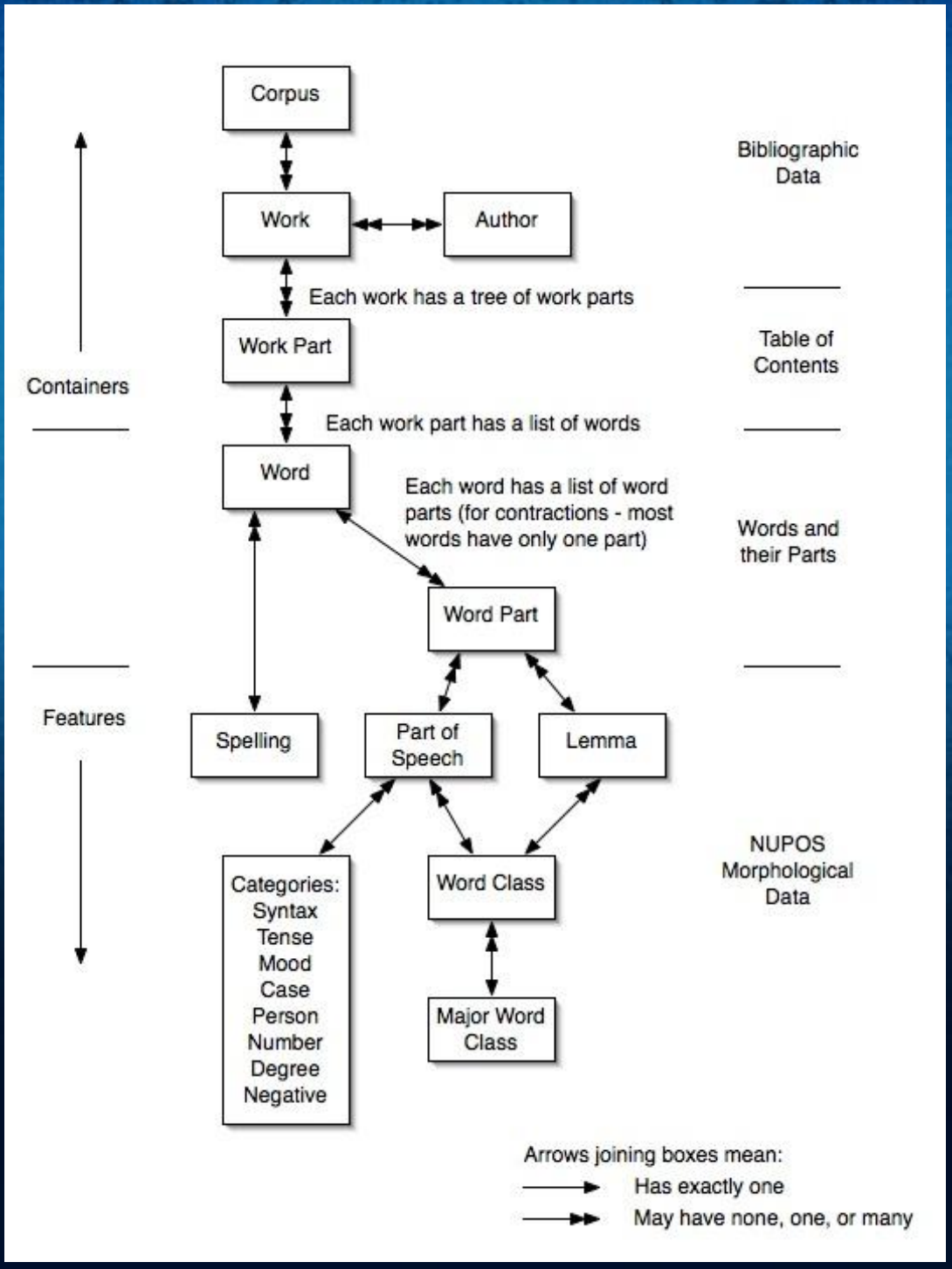
- Data could be accessed directly from the database, but ...

- We provide a low-level "middleware" layer to hide the database details.

Stage Seven: Model and API

- Datastore implements an object model of the data.
- Written in Java.
- Developed by John Norstad.

- Model defines all static Monk objects, their attributes, and
- relationships.
- Model encapsulates and hides datastore implementation details.


Corpus
Work
Author
Each work has a tree of work parts
Work Part
Each work part has a list of words
Word
Each word has a list of word parts (for contractions - most words have only one part)
Word Part
Spelling
Part of Speech
Lemma
Categories:
Syntax
Tense
Mood
Case
Person
Number
Degree
Negative
Word Class
Major Word Class
Containers
Features
Bibliographic Data
Table of Contents
Words and their Parts
NUPOS Morphological Data
Arrows joining boxes mean:
Has exactly one
May have none, one, or many

Model Example

```
void topTenNouns (Author author)
throws ModelException
{
Collection<Counter<Author,Lemma>> counters =
Counter.find(Author.class, Lemma.class,
new AuthorCriterion(author),
new MajorWordClassCriterion("noun"));
Counter<Author,Lemma>[] sortedCounters =
Counter.sort(counters, Counter.SortOption.COUNT_CUM_MAIN_DESCENDING);
int k = 0;
for (Counter<Author,Lemma> counter : sortedCounters) {
Lemma lemma = counter.getFeature();
long count = counter.getCountMain(CumKind.CUM);
System.out.println(lemma.getTag() + " " + count);
k++;
if (k == 10) break;
}
}
```

Possible Extensions To Model Interface

- Add other middleware access interfaces to data
- Examples:
  - Language independent remote calls (Burlap, Hessian)
  - Web services interfaces (WSDL, SOAP, RESTful)
  - Corpus query language like Poliqarp, Xaira

For More Information

The Monk Project web site:

http://www.monkproject.org

The MorphAdorner web site:

http://morphadorner.northwestern.edu/