Universiteit Utrecht

An extensible database design for cross-linguistic research

Alexis Dimitriadis

Utrecht institute of Linguistics OTS

Utrecht University

Outline

Outline

Linguistic databases are difficult I

It's hard to know what to build

- Built or designed by linguists, not professional IT staff.
- They don't look like the examples in database textbooks: The relational structure is not obvious.
- **Linguists change their minds:** At the start of a research project, it is impossible to know what the data should look like.
- It can take a long time to build a database, and a long time to modify it; the needs of the project are always ahead of the software.

Linguistic databases are difficult II

Some things are just hard to build

- They typically store text, rather than numbers—including non-English text (non-Latin alphabets, IPA).
- Typological databases can **grow to a very large number of attributes.**
- Many fields take a value from a list of **alternatives** (“enumerated” values).
- We often want to choose **more than one answer.**
- Many answers are qualified or uncertain.
- **Comments** are frequently desirable, and extremely important.
- **Glossed text** must be properly managed and displayed.

Linguistic databases evolve

Workflow for a typical small linguistic database:

- Design and build initial database
- Enter test data and adjust design
- Begin collecting data in earnest
- **Revise value enumerations, and maybe attributes, as unexpected data is found**
- Study the data, publish (?)
- Formulate new research questions
- **Add new attributes to the database**
- Repeat

This is unusual in the database world: The design of a database is not supposed to change after the end of the design phase.

Linguistic databases evolve

Workflow for a typical small linguistic database:

- Design and build initial database
- Enter test data and adjust design
- Begin collecting data in earnest
- **Revise value enumerations, and maybe attributes, as unexpected data is found**
- Study the data, publish (?)
- Formulate new research questions
- **Add new attributes to the database**
- Repeat

This is unusual in the database world: The design of a database is not supposed to change after the end of the design phase.

Does it matter?

Yes, it does: Because database-building tools—and practitioners—do not expect the design to be continuously revised, adding an attribute is a lot harder than adding linguistic data.

- If you add an attribute to a table in a desktop database, it will not automatically show up in existing forms.
- If you've hired someone to create a web database for you, you'll have to ask them to come back and make the changes you need.
- The necessary changes can take a long time to happen.

Outcome: Linguistic databases are underused I

- To sidestep such problems, many linguists use databases in ways that limit their potential to **assist** data collection and analysis.

1. **Too late:** “I won’t create a database until I’ve collected and analyzed my data.”
2. **Too little:** Some databases store only summary data in their own tables; the bulk of the information is in linked text documents (e.g., in PDF format).
3. **Too much trouble:** “We have to create a database, because we promised to do so in our grant application.”

Outcome: Linguistic databases are underused II

- Such databases are “display cabinets,” used primarily for **presentation** of the collected data at the end of the project.
- There’s nothing wrong with this use of databases; but it is only a small part of what databases can do for us.
- To get the most out of databases, we must address the obstacles that discourage prudent linguists from relying on them.

Linguistic databases are easy

Mostly because they're small

- Performance is not a big issue
  - Most linguistic databases are (relatively) tiny.
  - So is their user base.
- Authentication and security needs are generally simple.
  - A few trusted contributors
  - Free read-only access for the general public
- Concurrency issues are not a big problem.
- Usually no need for complex updates and dependencies.
  - Any data entered is meant to remain in the database.
  - Data is not deleted or updated except to make corrections.
- Not a commercial database: It's ok to have imperfections.

Outline

Linguistic databases (often) look alike

Typical content of a cross-linguistic survey database:

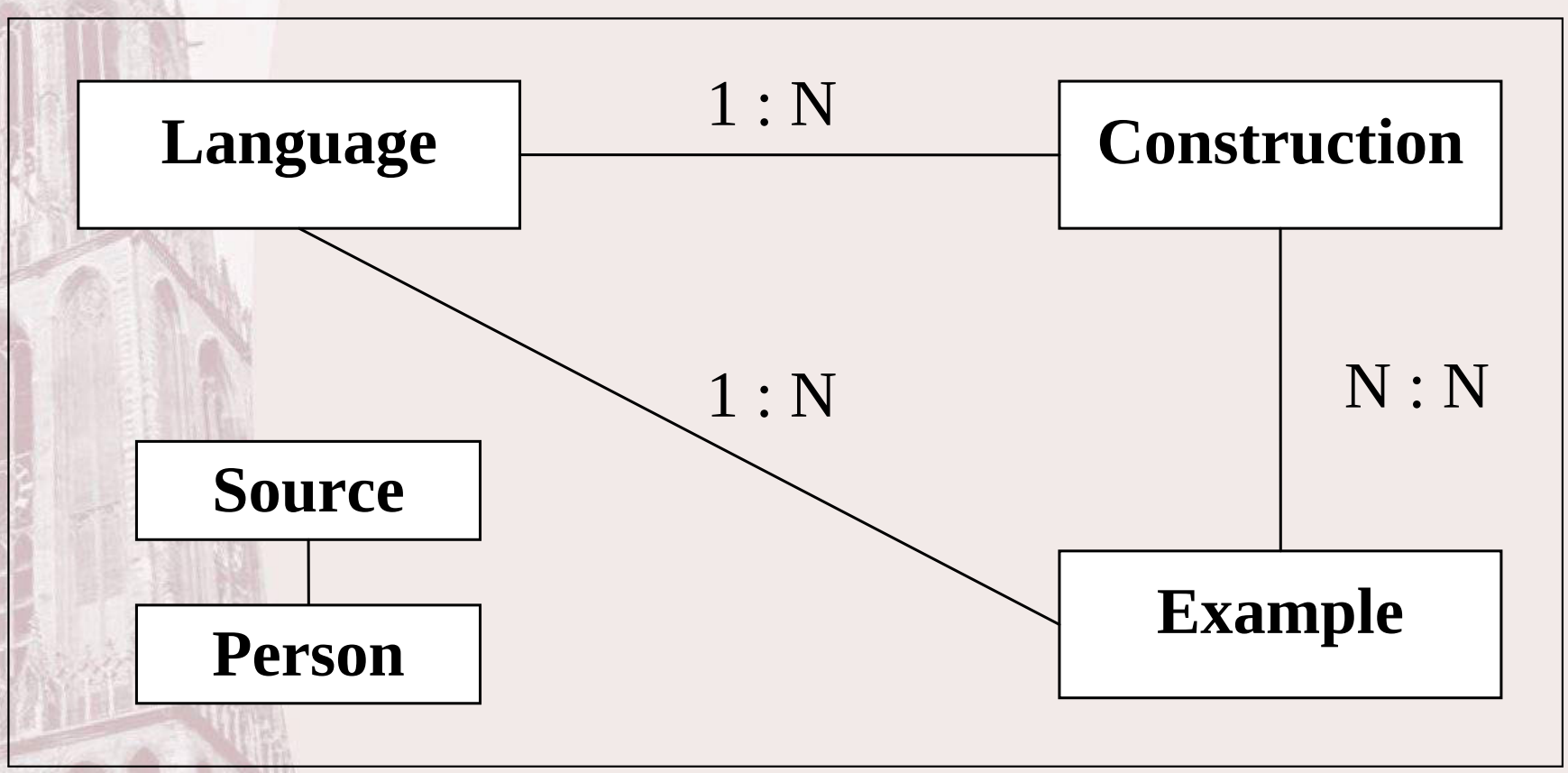
- Languages
- Instances of a construction or phenomenon under study
- Examples

- Persons involved: Analysts, consultants
- Sources for the information: persons or books
- Auxiliary tables: Construction types, enumerated value lists, etc.

Reciprocals in English

- Which of these count as reciprocals? How many distinct **kinds** of reciprocal are there?
  - They like each other.
  - John and Mary argued on the way home.
  - We looked at one another.
  - They were at one another's throats.
  - They spread rumours about each other.
  - Each of them likes the other.
- To understand reciprocals (or anything else), we identify patterns: Our goal is not to study every sentence we come across, but to identify the distinct **kinds** of reciprocals and to describe (analyze, understand) each one of them.

Core ER schema for a survey database



- Sources (and persons) are linked to other tables as required.

Examples of survey databases

- **BURS:** Languages, reciprocals, examples
- **Berlin intensifiers db:** Languages, intensifiers and reflexives, examples
- **Graz reduplication db:** Languages, “reduplicants”, “illustrations”
- **Topic-Focus db (UvA):** Languages, focus constructions, examples (but also separate “exponents” components)
- **African Anaphora db (Rutgers):** Languages, anaphoric markers, examples

Outline

A flexible database template I

We have developed a general template based on the common characteristics of linguistic survey databases.

- A web database for use by a group working on a single research project.
- Data entry is by password only. Browsing can be restricted by password or (when ready) open to everyone.
- Implements the core Language-Construction-Example structure.
- Glossed examples are properly displayed.
- It is easy to add new questions, or modify existing ones. (Without a degree in computer science).

A flexible database template II

Some technical features:

- Allows multi-valued attributes, lots of comment fields
- Supports large number of descriptive attributes
- Easy to change or add attributes, enumerated values
- Manages enumerated value lists
- Uses Unicode: Any alphabet or character set can be entered.
- Documentation of attribute and value meanings

BURS: The software and the project

- Created for the project **A typology of reciprocal markers: Analysis and documentation** (Freie Unitersität Berlin and Utrecht University)
- Supported by the DFG-NWO bilateral cooperation programme
- At the Utrecht institute of Linguistics:
  Prof. dr. M.B.H. Everaert, Dr. Alexis Dimitriadis, Dr. Anca Sevcenco
- At the Freie Unitersität Berlin:
  Prof. Dr. Ekkehard König, Dr. Volker Gast, Dr. Carola Emkow, Thomas Hanke
- Programming: Floris van Vugt and Alexis Dimitriadis
- Software: PHP web interface, MySQL back end, CSS stylesheets, and just a little javascript.

Step 1: Manage enumerated value lists

- For attributes whose value comes from a fixed list of alternatives (e.g., “part of speech” or “linguistic macro-area”), it is common practice (and a good idea) to store the possible values in a special table.
- Instead of creating a new table for each such list of **enumerated values,** we place them all in a single table that can be managed with a single set of forms.
- New values, and new types of enumerated values, can be added at any time.
- The definition tables provide a place to document the meaning of each value and value type.

The value lists are simple to use

- The attribute VALUETYPE identifies each value list.
- The attribute RANK controls presentation order.
- The unified table is managed via a single set of forms.
- Using the typed values is just as easy as using separate tables:

```
SELECT ID, Label FROM AgrFeature
    ORDER BY Rank

SELECT ID, Label FROM ValueDefinitions
    WHERE ValueType = 'AgrFeature'
    ORDER BY Rank
```

Enumerated value types

Value types

Id	*Label*	*Description*	*Customizable*	
AgrFeature	Agreement category	Features (categories) for which a controller may trigger agreement on the target	Yes	Show Modify
boolean	Boolean	true/false data type	No	Show Modify
Familiarity	Familiarity	Degree of familiarity with the language being described	No	Show Modify
grammarFamiliarity	Familiarity with the Grammar	Knowledge of the formal grammar of the language	No	Show Modify
langRating	Rating of the dataset	An evaluation of the thoroughness or quality of the collected data, based on quantity, degree of detail, originality of the data, confidence in its correctness, comprehensiveness, etc.	No	Show Modify
		A set of languages that are		

Values for “Agreement Category”

Manage Definitions

Back

Id	*Label*	*Description*	*Customizable*	
AgrFeature	Agreement category	Features (categories) for which a controller may trigger agreement on the target	Yes	Show Modify

Value definitions

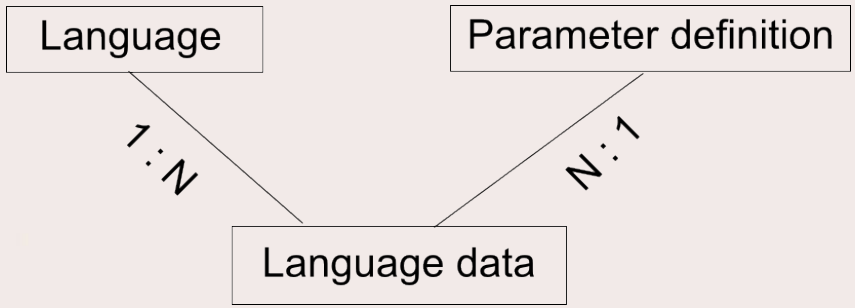
Id	*Label*	*Description*	*Rank*	*Comments*	*Contributor*	
person	person		10			Modify Delete
number	number		20			Modify Delete
case	case		30			Modify Delete
gender	gender/noun class		40			Modify Delete
animacy	animacy		50			Modify Delete
definiteness	definiteness		60			Modify Delete

Add Value definition

Logged in as alexis: Alexis Dimitriadis.

Step 2: Manage descriptive fields I

- Instead of turning descriptive data fields into table columns (attributes), as is the usual practice, we store them in a table and manage them as **data.**
- The table *ParameterDefinitions* contains the questions to ask about each Language, Construction, and Sentence.
- Another table, *LanguageData,* contains the answer to each question (parameter), for each Language; similarly for the Constructions and Sentences.



Step 2: Manage descriptive fields II

- New questions (“parameters”) can be added without modifying the relational schema of the database.
- Again, a single set of forms manages all parameter definitions.
- Interface forms are dynamically generated: new questions (and answers) are included automatically.
- Allowing repeated answers is now a simple matter; we don’t need an extra table for each multi-valued parameter.
- The ParameterDefinitions table includes a place for documenting each linguistic parameter.

Question group on morphological form:

Id	*Label*	*Rank*	*Entity*	*#Qs*	
morphForm	Form	210	strategy	5	Show Modify Delete

Questions

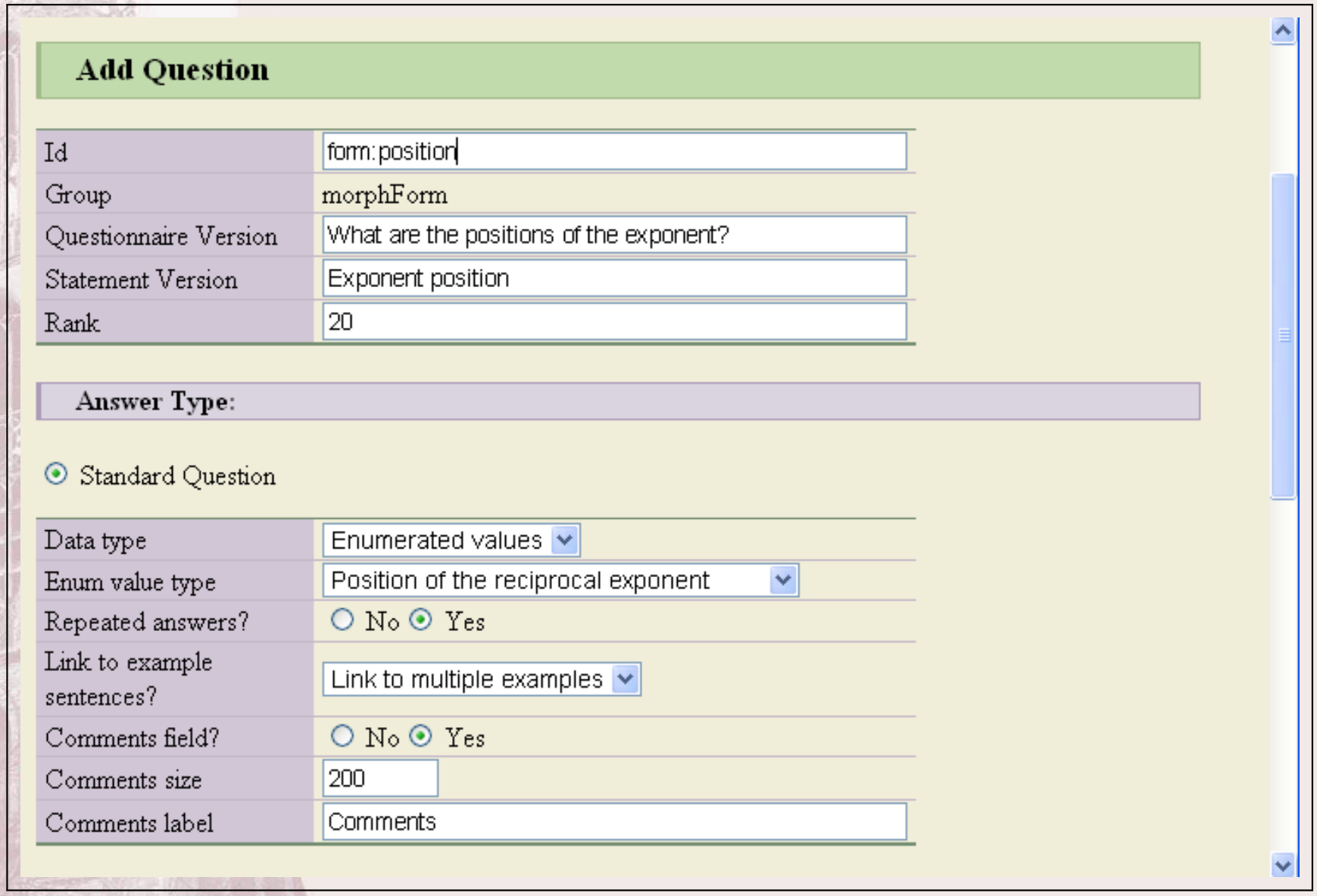
Id	*Label*	*Rank*	*#Param*	
mForm:expPosition	What is/are the positions of the exponent(s)?	10	4	Show Modify Delete
mForm:expGloss	Give a detailed glossed breakdown of any parts of the exponent, indicating lexical meaning and/or grammatical function of each part.	20	2	Show Modify Delete
mForm:expLiteral	Can the exponent be used with its literal lexical meaning (not as a reciprocal?)	30	3	Show Modify Delete
mForm:history	Can you speculate on the historical origin of the exponent or its parts?	40	1	Show Modify Delete
mForm:Lexifier	If there is a detectable lexical source, what is it?	50	2	Show Modify Delete

Add Question

Step 3: Support complex answers

- An answer often involves several independent parts, e.g.:
  - One or many selections from a list.
  - A single comment.
  - A link to one or more examples.
  - A bibliographic citation.
- We therefore added one more layer of complexity to the system: A Question is associated with several **Answer Components,** each of which may or may not allow repetition.

Creating a question:



Answering a question:

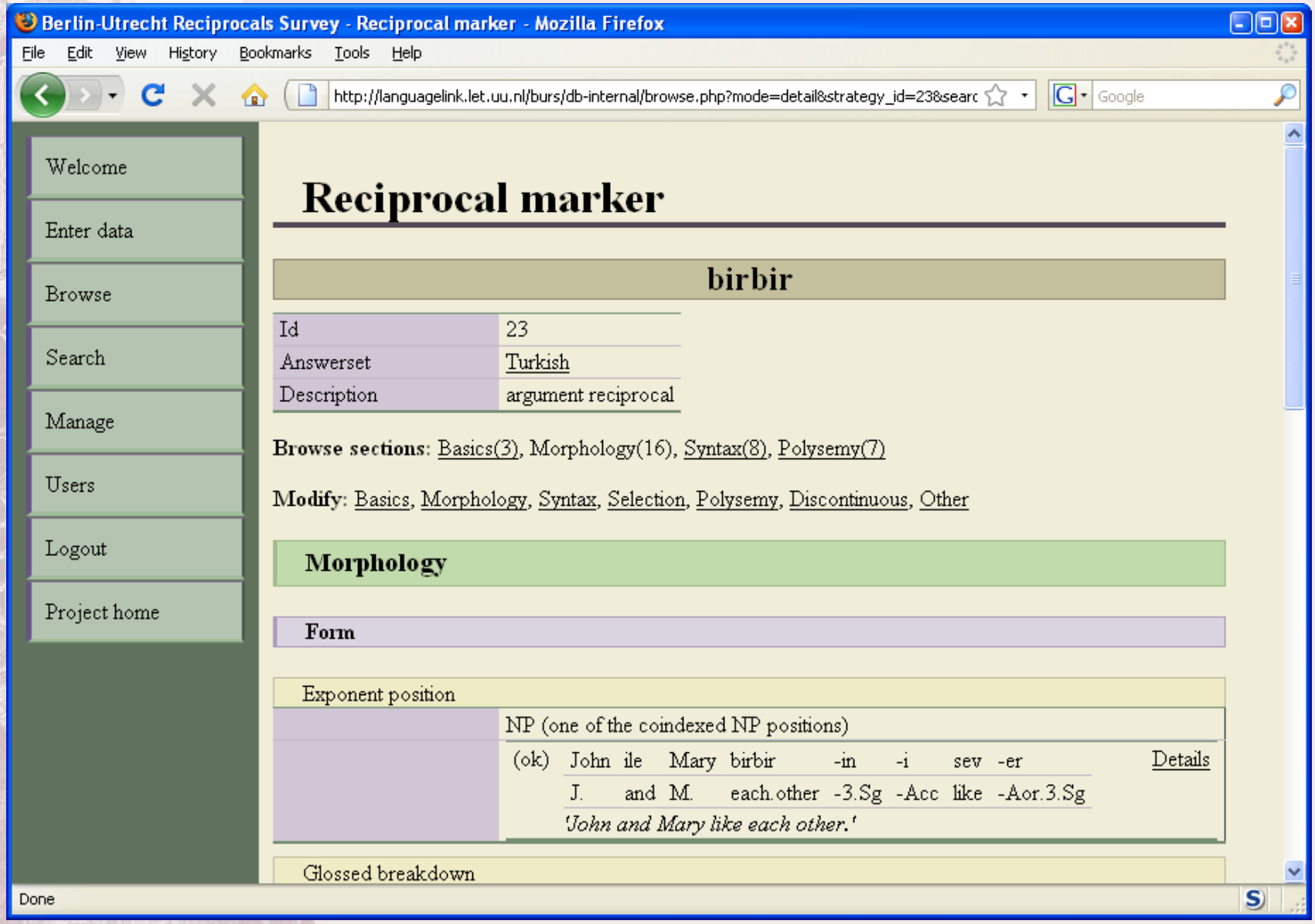
6. What are the positions of the exponent?

NP (one of the coindexed NP positions)
on the verb or auxiliary
clitic
no marking (null strategy)
adverbial phrase
prepositional phrase or oblique NP

Insert link to example

Comments

Displaying the answers:

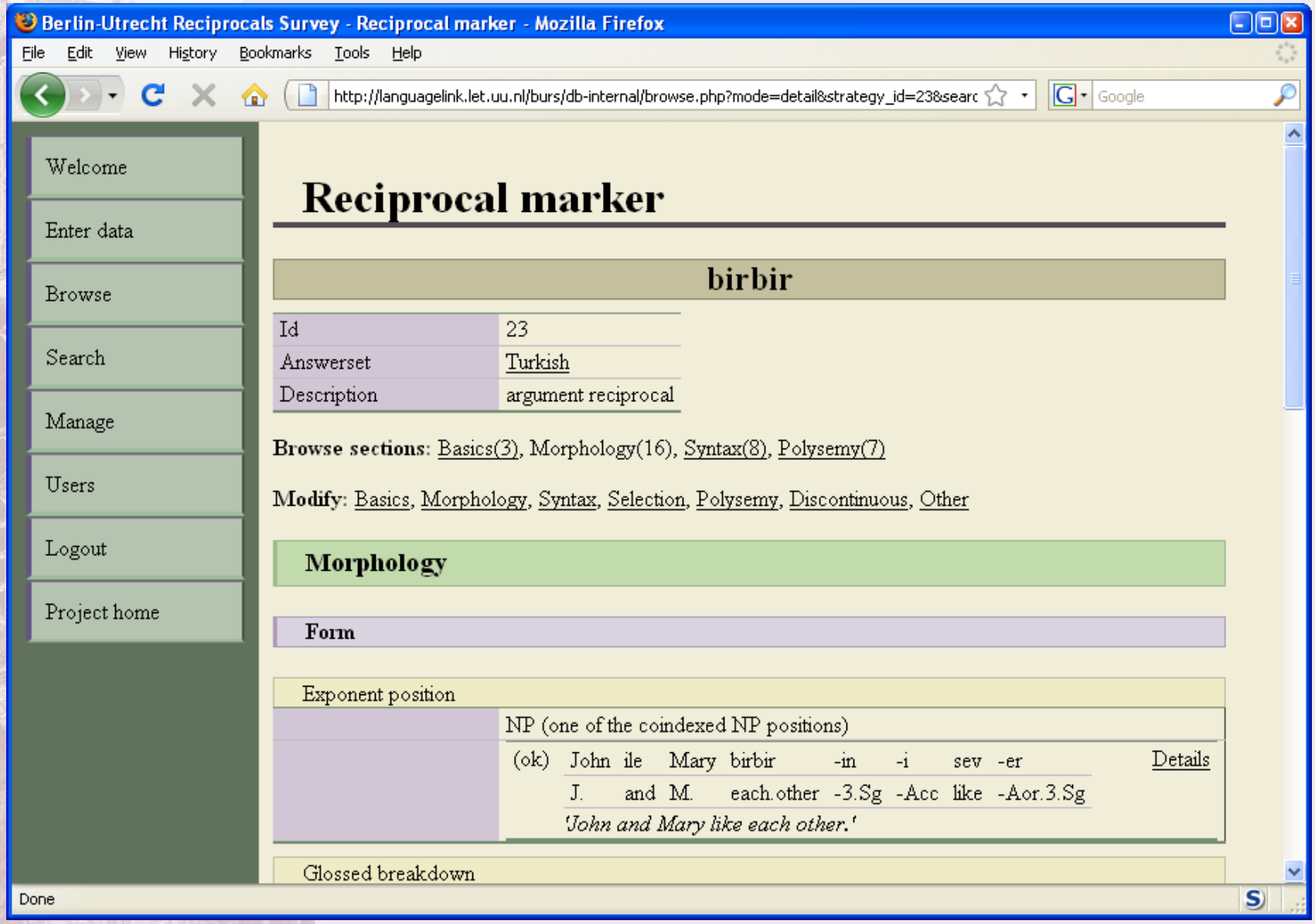


Outline

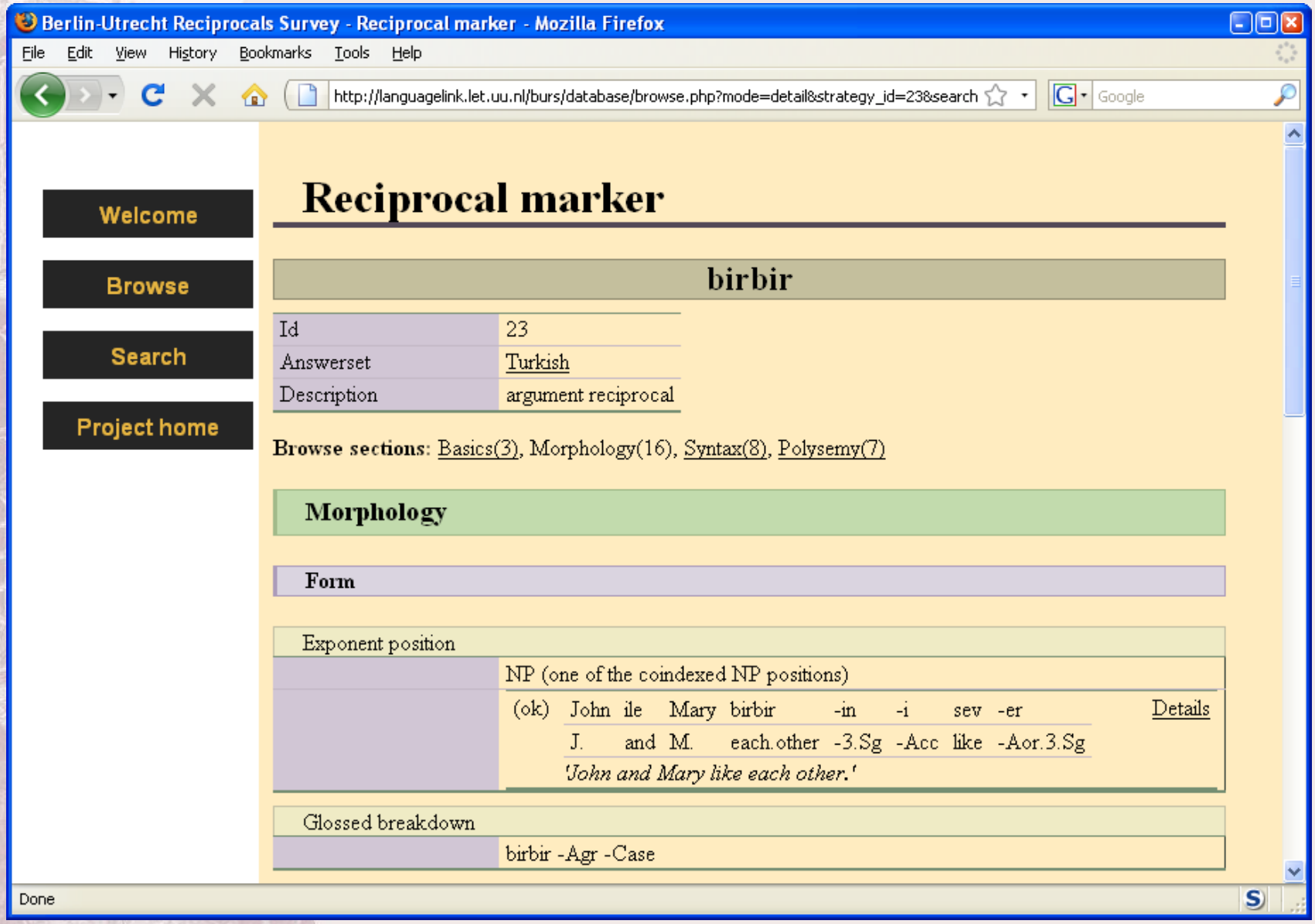
The software is configurable

- The name of the database and core entities can be changed in a configuration file.
- The descriptive parameters (“questions”) are defined through the web interface.
- The colors and layout can be adjusted too!
- Other aspects can be modified with some PHP programming.

Default style



Custom BURS style



Limitations of the software

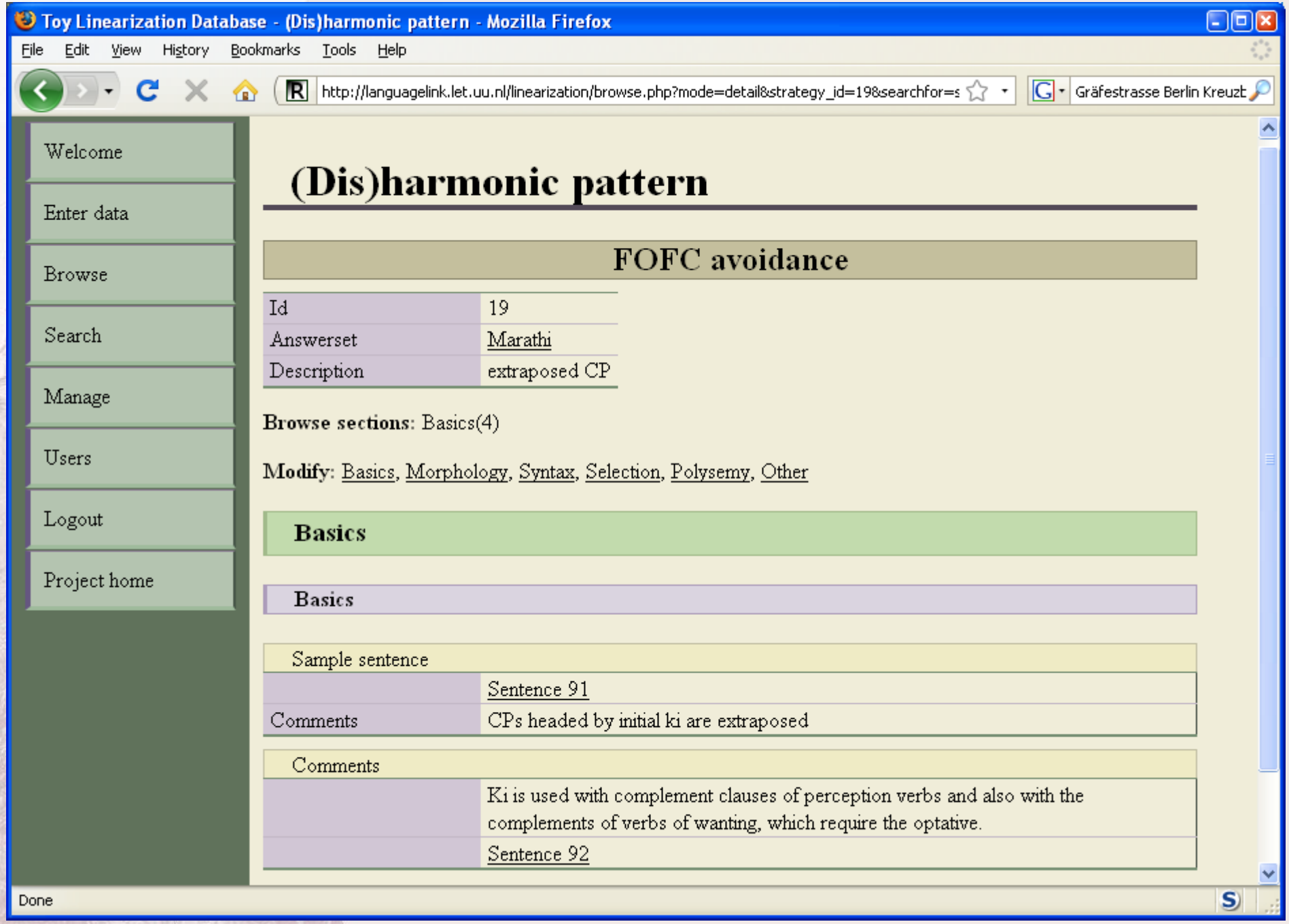
- No gloss validation and management.
- No special support for citing sources.
- No support for uploading and linking to files (e.g., PDF documents).
- The system is oriented toward data entry, not analysis: The search facility is powerful, but there is no support for data analysis (statistics) or for displaying a selected property from every record.
- Not as simple or as easily extensible as envisioned.

N.B. These are limitations of the program, not of the design!

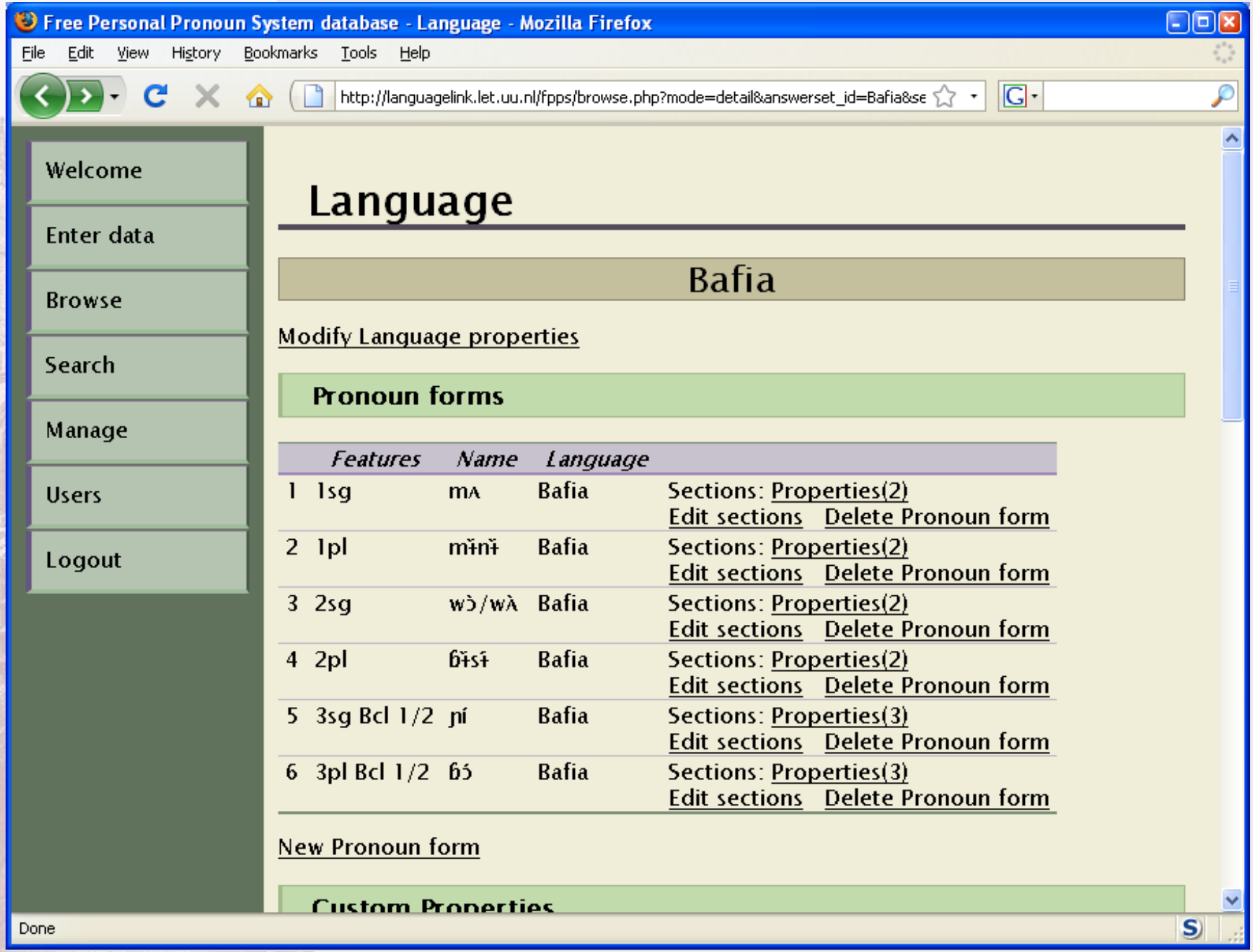
The system is flexible enough

Despite its limitations, the system has proved useful enough for several other cross-linguistic surveys:

1. African Anaphora Database (Ken Safir, Rutgers University)
2. Structure and Linearization in Disharmonic Word Orders (Holmsberg, Roberts et al., Newcastle / Cambridge)
3. Marked Nominatives (Corinna Handschuh, Leipzig)
4. Free Personal Pronoun Systems (Norval Smith, University of Amsterdam)
5. Indefinites and Beyond (Maria Aloni, University of Amsterdam)


Toy Linearization Database - (Dis)harmonic pattern - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://languagelink.let.uu.nl/linearization/browse.php?mode=detail&strategy_id=19&searchfor=s
Gräfestrasse Berlin Kreuzt
Welcome
Enter data
Browse
Search
Manage
Users
Logout
Project home
(Dis)harmonic pattern
FOFC avoidance
Id 19
Answerset Marathi
Description extraposed CP
Browse sections: Basics(4)
Modify: Basics, Morphology, Syntax, Selection, Polysemy, Other
Basics
Basics
Sample sentence
Sentence 91
Comments CPs headed by initial ki are extraposed
Comments
Ki is used with complement clauses of perception verbs and also with the complements of verbs of wanting, which require the optative.
Sentence 92
Done


African Anaphora Project: Anaphora in the African Languages - Anaphora Database - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.africananaphora.rutgers.edu/index.php?option=com_wrapper&Itemid=130
Google
AFRICAN ANAPHORA PROJECT
Anaphora in the African Languages
search...
Home » Anaphora Database
Case Files: Amharic | Bafut | Berber-Tarifyt | CiNsenga | Gungbe | Ibibio | Ikalanga | KiNande | KiRundi | Lokaa | LuBukusu | Songhay | Urhobo | Yoruba
Browse
Simple search
Enhanced search
Login
Project home
Language
Ikalanga
Anaphoric markers
Name Description Answerset
1. OM Verbal object marker Ikalanga Sections: MarkerShape(1)
2. PRN Pronominal Ikalanga
3. RCM Reciprocal verbal affix Ikalanga Sections: MarkerShape(4)
4. RFM Verbal reflexive affix Ikalanga Sections: MarkerShape(7), Properties(19)
Examples
Done
Scripts Partially Allowed | <SCRIPT>: 5 | <OBJECT>: 0


Free Personal Pronoun System database - Language - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://languagelink.let.uu.nl/fpps/browse.php?mode=detail&answerset_id=Bafia&se
Welcome
Enter data
Browse
Search
Manage
Users
Logout
Language
Bafia
Modify Language properties
Pronoun forms
Features Name Language
1 1sg mʌ Bafia Sections: Properties(2) Edit sections Delete Pronoun form
2 1pl mɨnɨ Bafia Sections: Properties(2) Edit sections Delete Pronoun form
3 2sg wɔ/wʌ Bafia Sections: Properties(2) Edit sections Delete Pronoun form
4 2pl ɓɨsɨ Bafia Sections: Properties(2) Edit sections Delete Pronoun form
5 3sg Bcl 1/2 ɲí Bafia Sections: Properties(3) Edit sections Delete Pronoun form
6 3pl Bcl 1/2 ɓɔ Bafia Sections: Properties(3) Edit sections Delete Pronoun form
New Pronoun form
Custom Properties
Done

Outline

Conclusions I

1. The database should support the way we work. It should allow us to:
   - search, view and extract the information we need.
   - **enter** the information we need!
   - define questions gradually, as we examine existing data.
2. Entering data in the database should be *easier* than keeping it in random Word documents. (In the long run, at least).
3. The database must be able to change as fast as our ideas.

Conclusions II

A database is a tool

- We should think of the database as a tool, not as an end-product.
- A linguistic database can be used for managing data **during** the course of a project.

Conclusions II

A database is a tool

- We should think of the database as a tool, not as an end-product.
- A linguistic database can be used for managing data **during** the course of a project.

- **In short:** Ask not what you can do for your database; ask what your database can do for you.

Thank you

Outline

1. Linguistic databases: easy and hard parts
2. Survey databases are all alike
3. We can solve some problems just once
4. Our system is limited, but has proved useful
5. Conclusions