Universiteit Utrecht

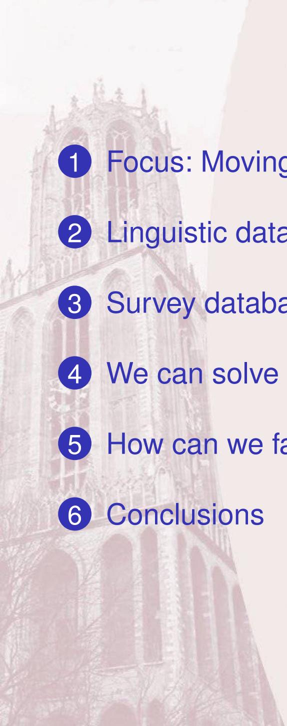
What cross-linguistic survey databases have in common

(and what they do not)

Alexis Dimitriadis

Utrecht institute of Linguistics OTS

Utrecht University

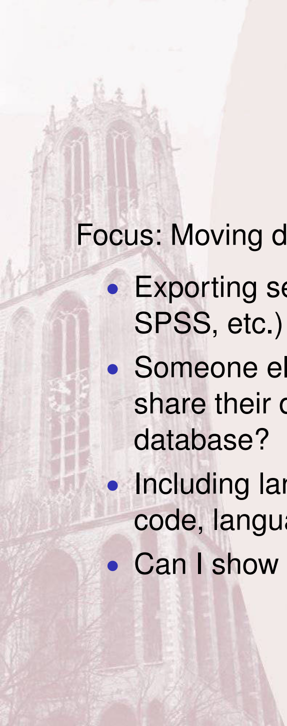


Outline

1. Focus: Moving data
2. Linguistic databases: easy and hard parts
3. Survey databases are all alike
4. We can solve some problems just once
5. How can we facilitate data exchange?
6. Conclusions

Outline

1. Focus: Moving data
2. Linguistic databases: easy and hard parts
3. Survey databases are all alike
4. We can solve some problems just once
5. How can we facilitate data exchange?
6. Conclusions



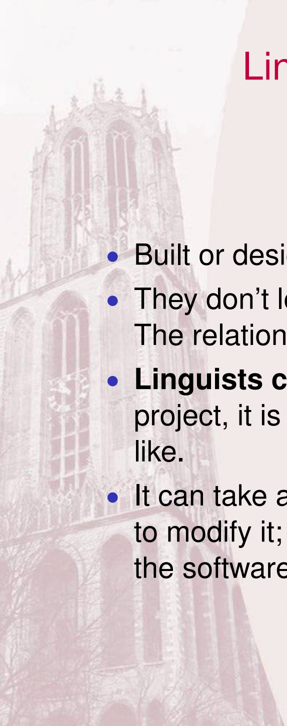
Moving linguistic data across applications

Focus: Moving data in and out of a cross-linguistic database

- Exporting selected data for statistical analysis (with R, SPSS, etc.)
- Someone else has done related work and are willing to share their data; how do I import some of their data into my database?
- Including language information, e.g. from Ethnologue (ISO code, language family, etc.)
- Can I show my results on a map?

Outline

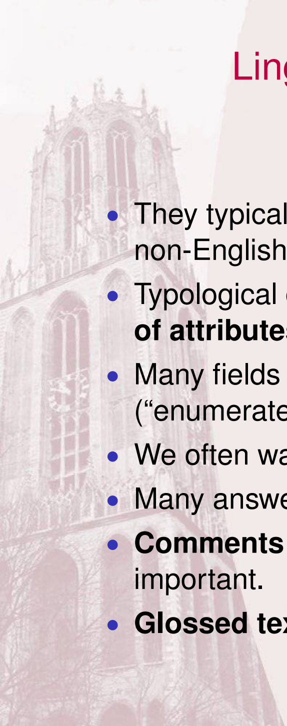
1. Focus: Moving data
2. Linguistic databases: easy and hard parts
3. Survey databases are all alike
4. We can solve some problems just once
5. How can we facilitate data exchange?
6. Conclusions



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.

Outline

1. Focus: Moving data
2. Linguistic databases: easy and hard parts
3. Survey databases are all alike
4. We can solve some problems just once
5. How can we facilitate data exchange?
6. Conclusions

Linguistic databases (often) look alike

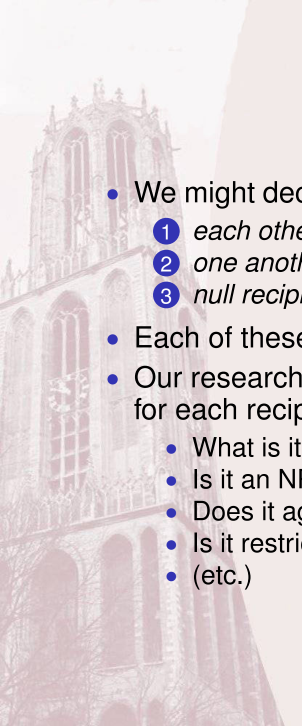
Typical content of a cross-linguistic survey database:

- 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

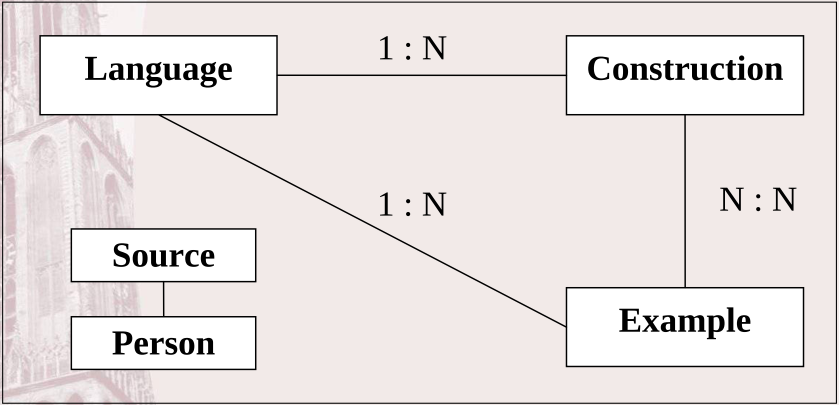
- Which of these count as reciprocals? How many distinct **kinds** of reciprocal are there?
  - They like each other.
  - John and Mary argued on the way home.
  - We looked at one another.
  - They were at one another's throats.
  - They spread rumours about each other.
  - Each of them likes the other.
- 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.



Reciprocals in English II

- We might decide that English has the following reciprocals:
  1. *each other*
  2. *one another*
  3. *null reciprocal* (with verbs like *argue*)
- Each of these is **specific** to English.
- Our research must answer certain questions **separately** for each reciprocal:
  - What is its overt form (exponent)?
  - Is it an NP, quantifier, verbal affix, or null?
  - Does it agree with its antecendent?
  - Is it restricted to a certain class of verbs?
  - (etc.)

Core ER schema for a survey database



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

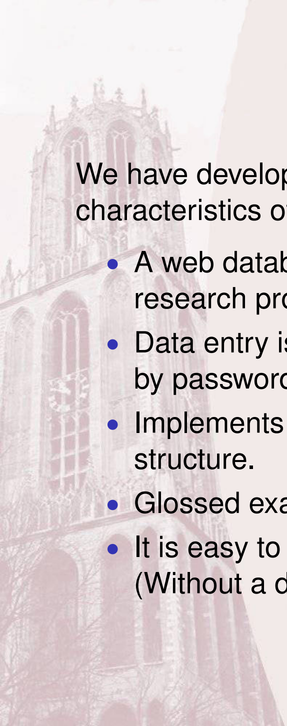
Examples of survey databases

- **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

1. Focus: Moving data
2. Linguistic databases: easy and hard parts
3. Survey databases are all alike
4. We can solve some problems just once
5. How can we facilitate data exchange?
6. Conclusions

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

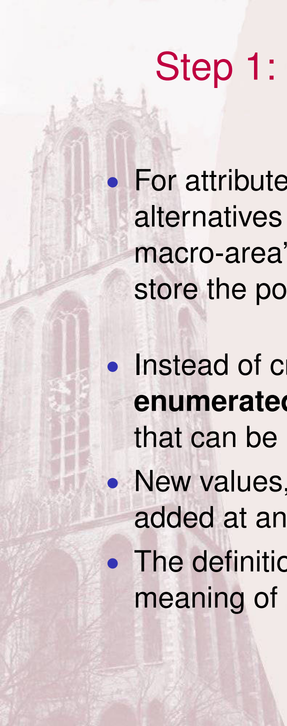
Some technical features:

- 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.

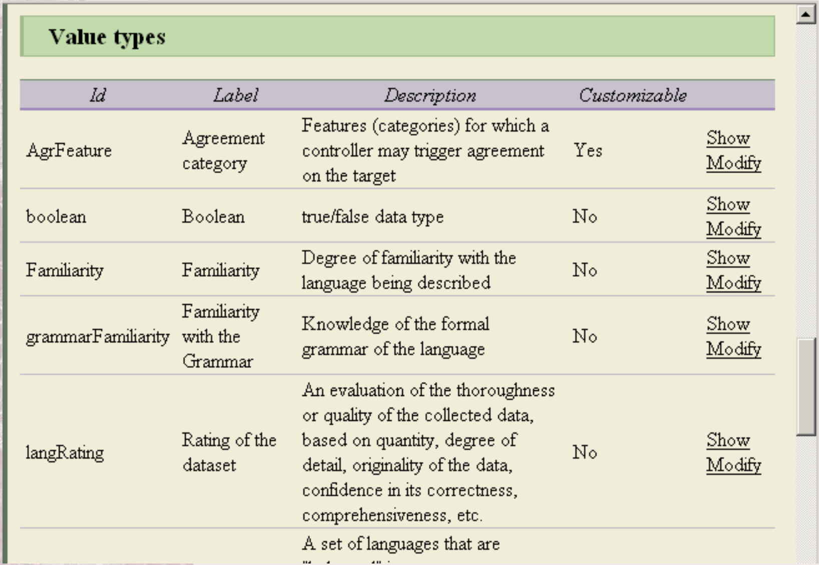
Step 1: Manage enumerated value lists



- 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.

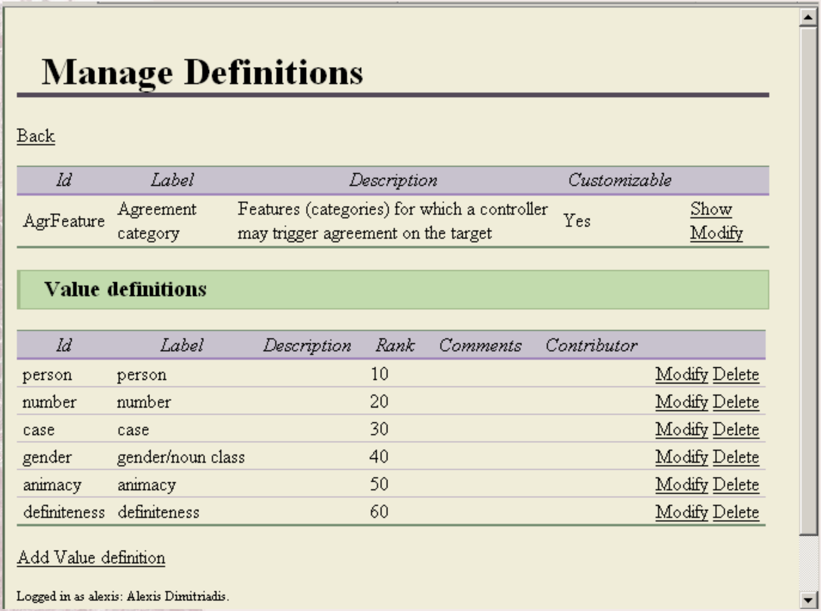
Enumerated value types



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

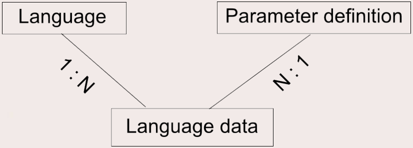
Id	*Label*	*Description*	*Rank*	*Comments*	*Contributor*	
person	person		10			Modify Delete
number	number		20			Modify Delete
case	case		30			Modify Delete
gender	gender/noun class		40			Modify Delete
animacy	animacy		50			Modify Delete
definiteness	definiteness		60			Modify Delete

Add Value definition

Logged in as alexis: Alexis Dimitriadis.

Step 2: Manage descriptive fields I

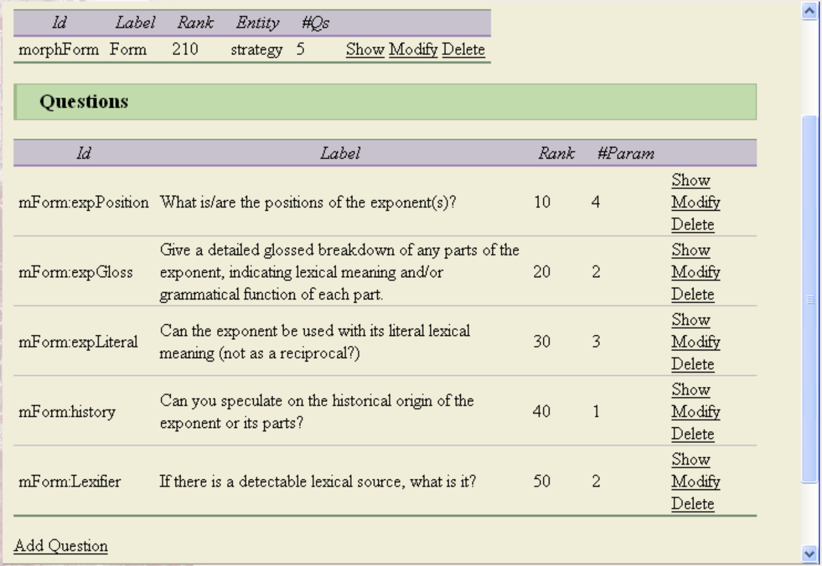
- Instead of turning descriptive data fields into table columns (attributes), as is the usual practice, we store them in a table and manage them as **data.**
- The table *ParameterDefinitions* contains the questions to ask about each Language, Construction, and Sentence.
- Another table, *LanguageData,*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

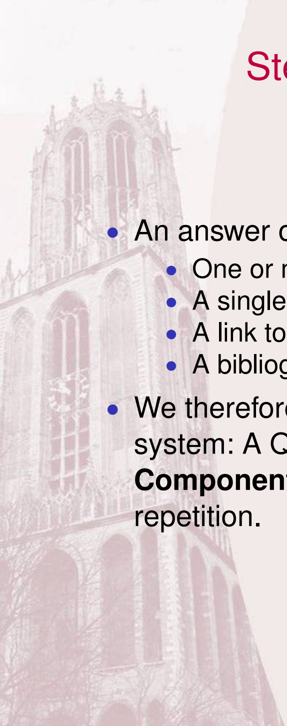


Id	Label	Rank	Entity	#Qs	
morphForm	Form	210	strategy	5	Show Modify Delete

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

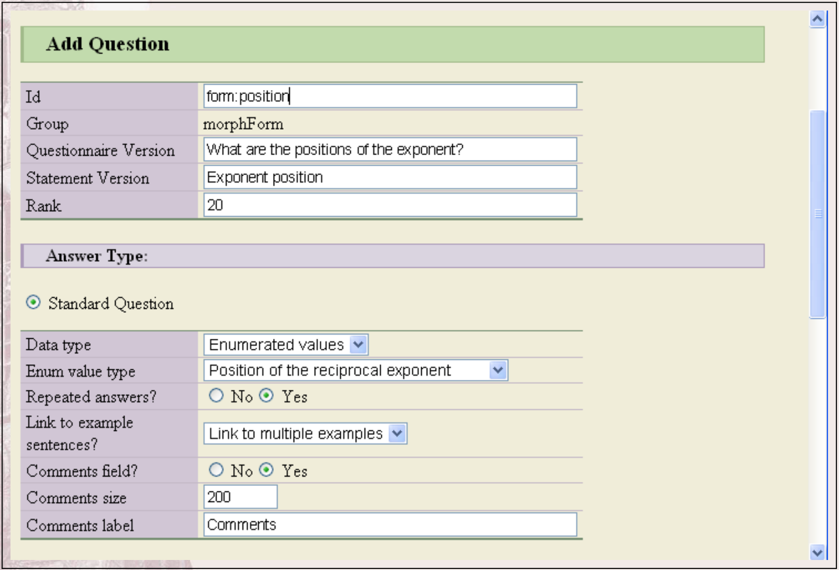
Add Question



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:



Add Question

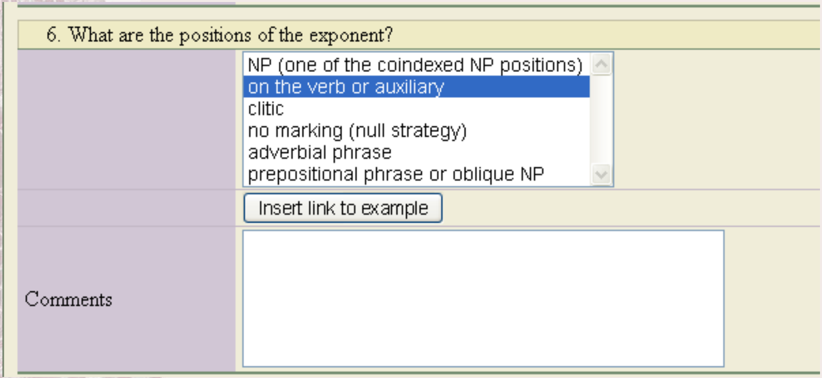
Id	form:position
Group	morphForm
Questionnaire Version	What are the positions of the exponent?
Statement Version	Exponent position
Rank	20

Answer Type:

● Standard Question

Data type	Enumerated values
Enum value type	Position of the reciprocal exponent
Repeated answers?	○ No ● Yes
Link to example sentences?	Link to multiple examples
Comments field?	○ No ● Yes
Comments size	200
Comments label	Comments

Answering a question:

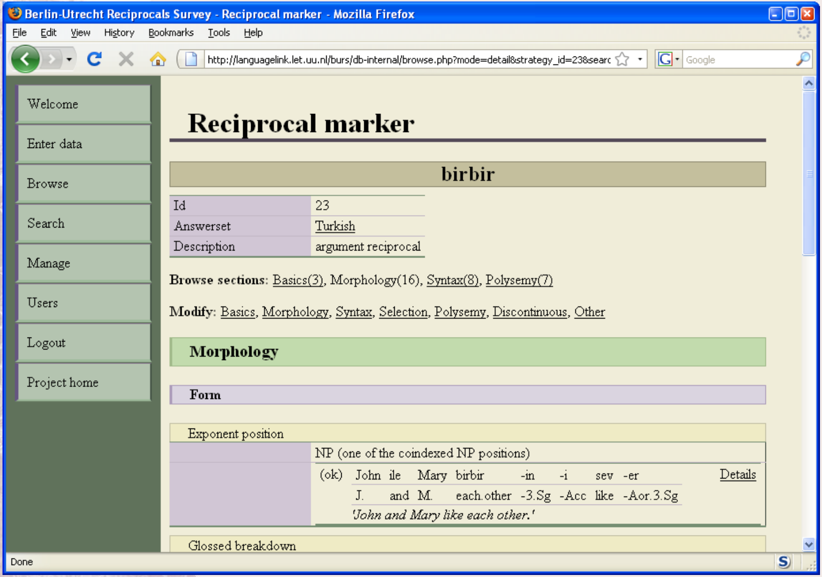


6. What are the positions of the exponent?

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

Comments

Displaying the answers:


Berlin-Utrecht Reciprocals Survey - Reciprocal marker - Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Help
http://languagelink.let.uu.nl/burs/db-internal/browse.php?mode=detail&strategy_id=23&searc
Google
Welcome
Enter data
Browse
Search
Manage
Users
Logout
Project home
Done

Reciprocal marker

birbir

Id	23
Answerset	Turkish
Description	argument reciprocal

Browse sections: Basics(3), Morphology(16), Syntax(8), Polysemy(7)

Modify: Basics, Morphology, Syntax, Selection, Polysemy, Discontinuous, Other

Morphology

Form

Exponent position

NP (one of the coindexed NP positions)

(ok)	John	ile	Mary	birbir	-in	-i	sev	-er	Details
	J.	and	M.	each.other	-3.Sg	-Acc	like	-Aor.3.Sg	

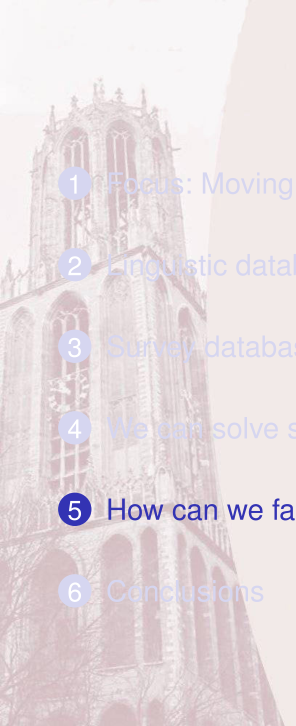
'John and Mary like each other.'

Glossed breakdown

The system is flexible enough

Despite its limitations, our current software 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. Free Personal Pronoun Systems (Norval Smith, University of Amsterdam)
4. Marked Nominatives (Corinna Handschuh, Leipzig)
5. Indefinites and Beyond (Maria Aloni, University of Amsterdam)
6. More databases under construction...

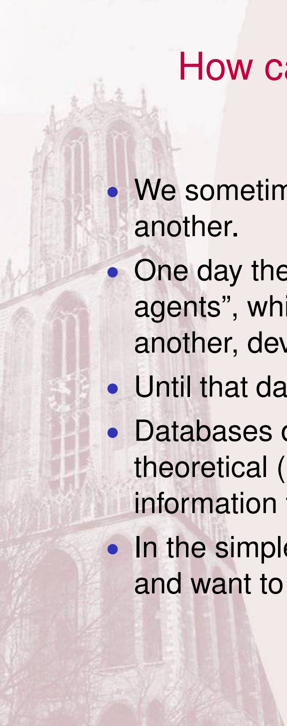


Outline

1. Focus: Moving data
2. Linguistic databases: easy and hard parts
3. Survey databases are all alike
4. We can solve some problems just once
5. How can we facilitate data exchange?
6. Conclusions

How can we facilitate data exchange?

- We sometimes want to transfer data from one database to another.
- One day the Semantic Web will provide us with “intelligent agents”, which will automatically map one schema to another, devise a transformation, and apply it to our data.
- Until that day comes, we have to do these things ourselves!
- Databases differ not only in their designs, but also in the theoretical (linguistic) meaning of the attributes and information they include.
- In the simplest case, we **know** what a body of data means, and want to import it into another database.



How can we facilitate data exchange?

- We sometimes want to transfer data from one database to another.
- One day the Semantic Web will provide us with “intelligent agents”, which will automatically map one schema to another, devise a transformation, and apply it to our data.
- Until that day comes, we have to do these things ourselves!
- Databases differ not only in their designs, but also in the theoretical (linguistic) meaning of the attributes and information they include.
- In the simplest case, we **know** what a body of data means, and want to import it into another database.

To exchange data, we need a data model

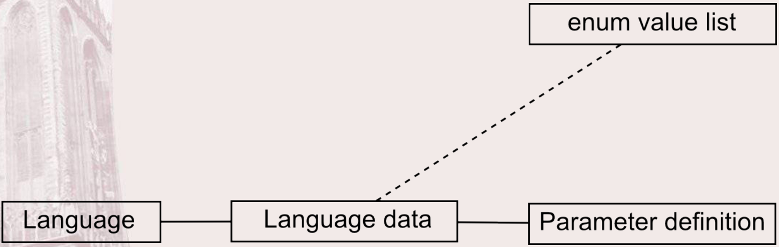
- We need not only a file format (e.g., “XML”), but also a model for organizing (grouping, structuring) our data.
- The data model must be able to handle our data.
- A model that assumes a single table cannot express the three-entity structure of the BURS.
- We can move data one table at a time, of course. But how useful are these tables?

To exchange data, we need a data model

- We need not only a file format (e.g., “XML”), but also a model for organizing (grouping, structuring) our data.
- The data model must be able to handle our data.
- A model that assumes a single table cannot express the three-entity structure of the BURS.
- We can move data one table at a time, of course. But how useful are these tables?

Our table schema I

Our implementation stores language data using several tables:



Language

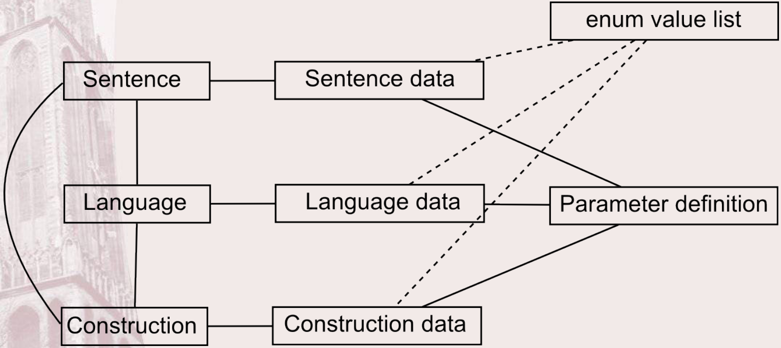
Language data

Parameter definition

enum value list

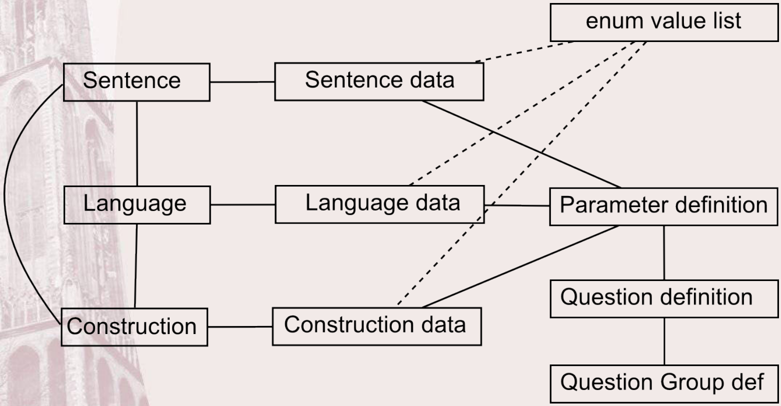
Our table schema II

The same with the other entity types:



Our table schema III

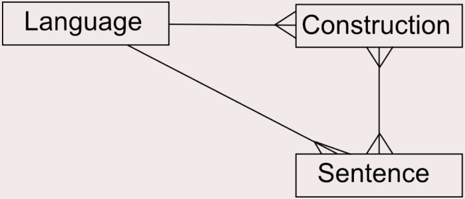
Parameters are grouped into questions, and these into groups:



This is too specific!

Bypassing implementation details

- A data-exchange model must be based on the common aspects of communicating databases
- The **conceptual level** captures what linguistic surveys have in common, regardless of implementation



- Cf. “annotation graphs” for mark-up of corpora, recordings

What we need from an exchange format

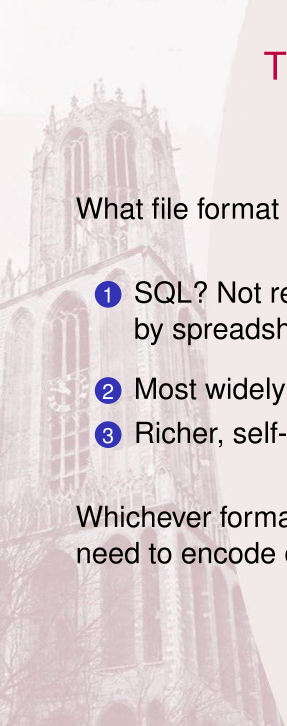
- Support for our data model (at the conceptual level)
- Non-English text: I.e., Unicode
- Support for multi-valued properties
- Compatible with current tools.
- Reasonably simple to use.

There is no perfect solution... yet

What file format should our solution use?

1. SQL? Not reliably “standard”; too low-level; not supported by spreadsheets, statistics packages
2. Most widely supported: CSV (comma-separated values)
3. Richer, self-documenting, lots of potential: XML

Whichever format we choose, our work has just begun: We still need to encode our data model somehow

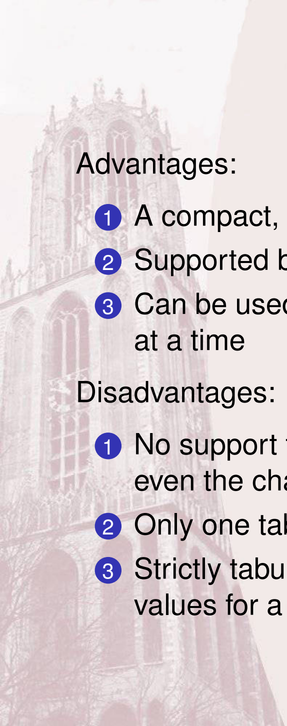


There is no perfect solution... yet

What file format should our solution use?

1. SQL? Not reliably “standard”; too low-level; not supported by spreadsheets, statistics packages
2. Most widely supported: CSV (comma-separated values)
3. Richer, self-documenting, lots of potential: XML

Whichever format we choose, our work has just begun: We still need to encode our data model somehow



Pros and cons of CSV

Advantages:

1. A compact, text-based format for tabular data
2. Supported by practically every relevant application
3. Can be used to transfer data among applications, one table at a time

Disadvantages:

1. No support for metadata (except for column names)– not even the character set used.
2. Only one table per file.
3. Strictly tabular format: No standard way to indicate multiple values for a cell.

Pros and cons of XML

Advantages:

1. Well-defined, self-documenting
2. Supported by a large number of new tools
3. Easily expresses multiple values
4. Extensible syntax that can be adapted to any data model

Disadvantages:

1. Many database applications still lack full support for reading and writing XML.
2. XML is a very general format: A syntax and data model must still be chosen.

Two XML-based solutions

The Typological Database System (TDS) integrates a number of independently developed typological databases. It relies on two custom-made XML solutions:

1. **High-end:** The Integrated Data and Documentation Format (IDDF) can encode a complete description of data and documentation. It is used to store and manage the collected data.
2. **Low-end:** To transfer data from some component databases, the TDS relies on a simple XML schema based on the table model.

Outline

1 Focus: Moving data

2 Linguistic databases: easy and hard parts

3 Survey databases are all alike

4 We can solve some problems just once

5 How can we facilitate data exchange?

6 Conclusions

Conclusions (and a wish list) I

1. There is currently no standard format or model for exchanging linguistic data in tabular form.
2. We need a general format: more specific than “XML” or “CSV”, expressive enough for more than isolated tables.
3. A successful exchange format must be based on a data model at the **conceptual** level.
4. The Language-Construction-Sentence model can be used for data exchange. (The interpretation, or “semantics,” of such data is another issue, of course!)

Conclusions (and a wish list) II

1. Existing database applications don’t fully support XML
2. Any future “standard” should be complemented by **software libraries** that applications can use to read and write such data.

Outline

1. Focus: Moving data
2. Linguistic databases: easy and hard parts
3. Survey databases are all alike
4. We can solve some problems just once
5. How can we facilitate data exchange?
6. Conclusions