The Typological Database System

How to integrate databases without starting a typology war

Alexis Dimitriadis and Menzo Windhouwer

Overview

- The Typological Database System (TDS) provides integrated access to multiple, independently created typological **databases.**

- Users can query the aggregated databases through the system's **web server:**

  http://languagelink.let.uu.nl/tds/

- The TDS is an NWO-supported LOT project, with participants from UvA, UiL-OTS, Leiden University and Nijmegen University.

Who?

Who?

- Developers
  - Tamas Biro (UvA), database integration
  - Alexis Dimitriadis (UU), project manager
  - Rob Goedemans (UL), database integration and phonology systems
  - Kees Hengeveld (UvA), database integration
  - Adam Saulwick (UvA), knowledge representation, ontology developer and typologist
  - Menzo Windhouwer (UvA), software system designer and developer
- Steering Committee
  - Martin Everaert (UU), Kees Hengeveld, chair (UvA), Roeland van Hout (RU), Pieter Muysken (RU), John Nerbonne (RUG), Peter Wittenburg (MPI)
- Student assistants and interns
  - Eugenie Stapert (UvA), Franca Wesseling (UvA), Ruth Lind (UU), Dirk van der Meulen (UvA)

Presentation outline

- Overview of the TDS
- Managing differences between databases
- The component databases
- The TDS server (demonstration/*tutorial*)
- *The TDS under the hood*
- *Guidelines for component databases*

Next:

- Overview of the TDS
- Managing differences between databases
- The component databases
- The TDS server (demonstration*/tutorial)*
- *The TDS under the hood*
- *Guidelines for component databases*

Superficial differences

- Different notational conventions
  e.g. glossing labels, field and variable names, description language
- Different design choices
  - There are many ways to organize information into tables and attributes
- Different software platforms
- Different types of content
  - “Analytical” variables which characterize a language as a whole
  - Annotated sentences with glosses, translations, and descriptive parameters
  - Multiple constructions per language

Contentful differences

- Different theoretical commitments influence:
  - Selection of what is recorded as “data”, and decisions on what factors to control for
  - Criteria and categories to be described
  - Associated terminology
- These differences are deliberate choices;
  If researchers don’t agree on a single analysis, they cannot be resolved.

The TDS approach

- Resolve superficial differences.
- Respect and highlight the theoretical commitments of each database, taking care to preserve the integrity and validity of the data.

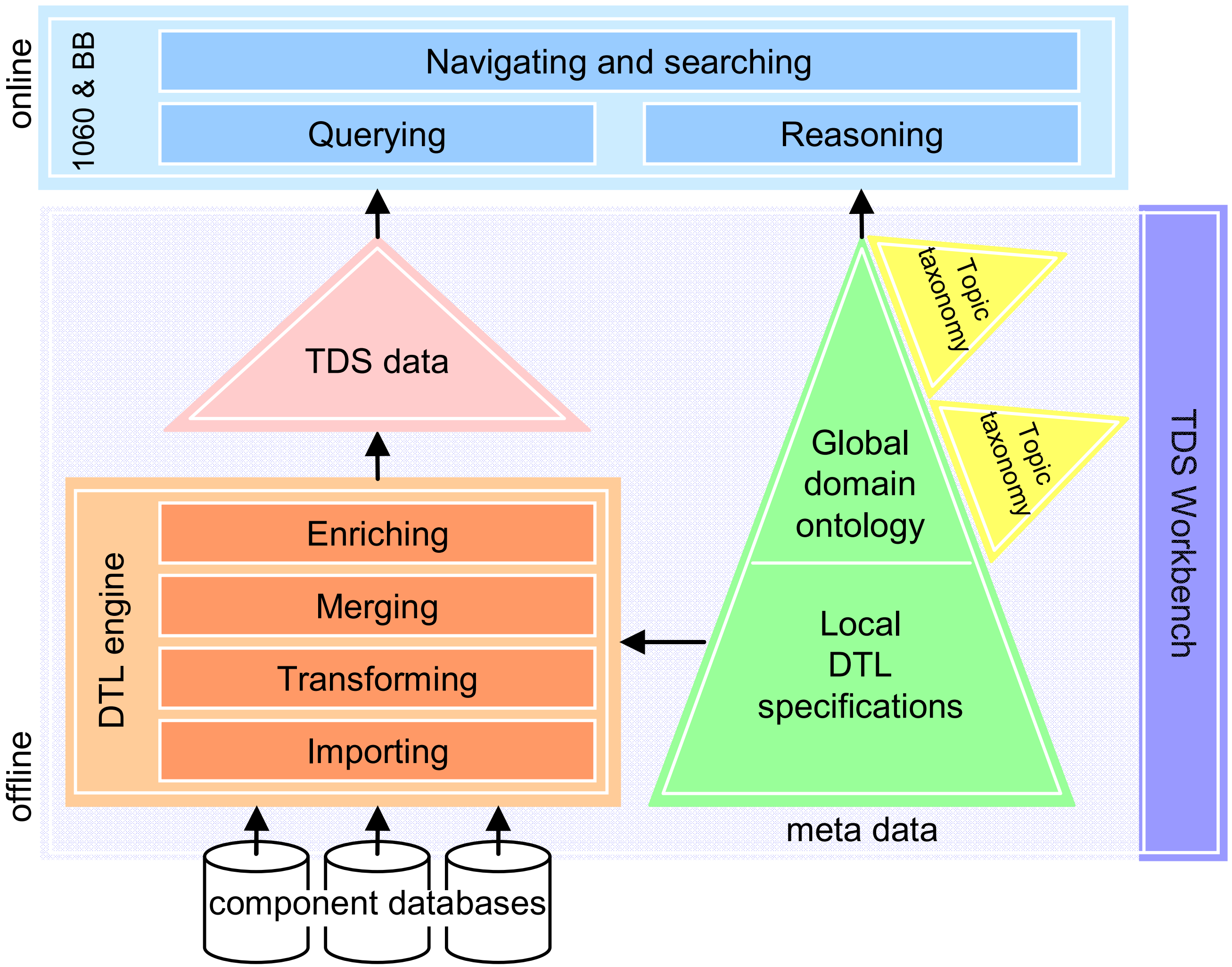
How databases are integrated

- A dump of the database is made available to the TDS.
- TDS developers define an import schema, which situates the contents of the database in the global hierarchy of the TDS.
- The data undergoes some transformations for uniformity; e.g., **1/0** and **true/false** become **yes/no.**
- Theoretically salient differences are preserved and documented (not removed!).
- The creators of the database are asked to clarify definitions and check the results.

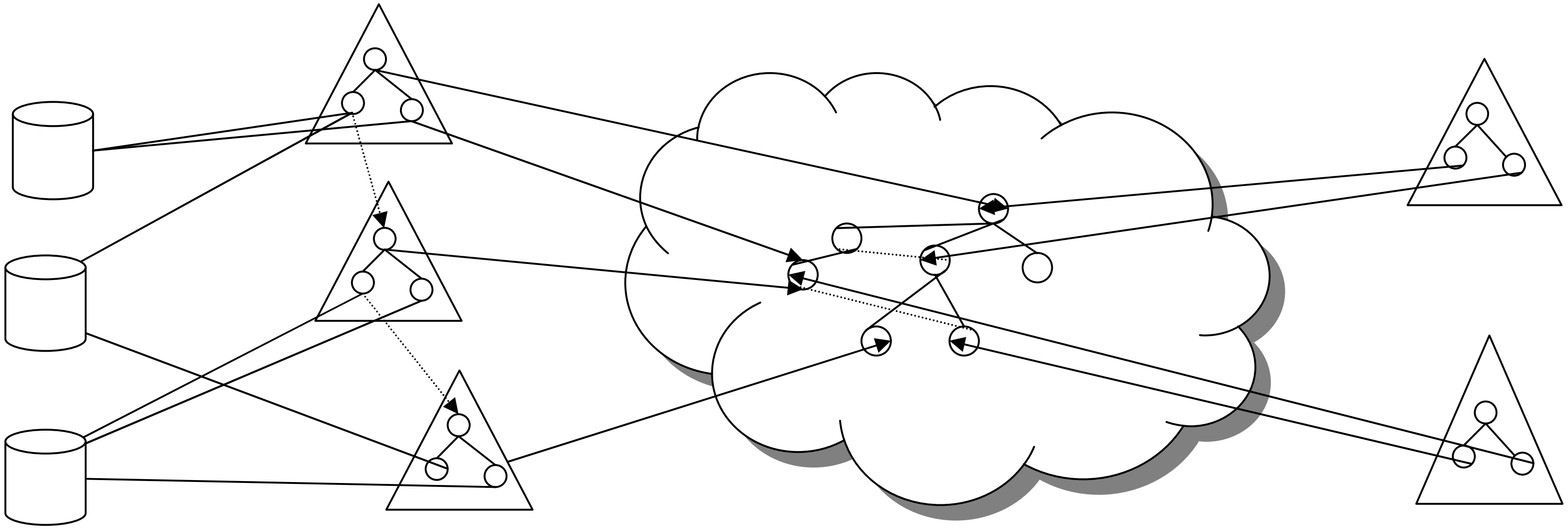
How databases are integrated (II)

- The import schema is encoded as a combination of
  - (a) modular, database-specific documentation and
  - (b) pointers into a global ontology of linguistic concepts
- The information aids the system in data navigation and presentation, and the users in its interpretation
- Updated versions of the databases can be easily re-imported, using the existing schema

TDS system architecture



Metadata architecture



Database schemata
(any DDL)

Local database ontologies
(DTL)

Global linguistic ontology
(OWL)

Topic taxonomies
(SKOS)

Database developer

TDS Knowledge engineer

Domain expert

Next:

- Overview of the TDS
- Managing differences between databases
- The component databases
- The TDS server (demonstration/*tutorial)*
- *The TDS under the hood*
- *Guidelines for component databases*

The component databases (I)

- Person-Agreement database (A. Siewierska, D. Bakker)
  Person and agreement phenomena. Over 400 languages
- Typological Database Nijmegen (L. Stassen)
  Word order, predication, case marking, relative clauses, comparatives, possession, coordination, and more. Between 140 and 400 languages, depending on topic
- Typological Database Amsterdam (K. Hengeveld)
  Basic word order and constituent order systems; parts-of-speech systems

The component databases (II)

- StressTyp (R. Goedemans, H. van der Hulst)
  metrical systems (stress, foot types, extrametricality etc.) for 510 languages

- SylTyp (H. van der Hulst, R. Goedemans)
  syllable structures

- UCLA Phonological Segment Inventory (I. Maddieson)
  segment inventories with phonological features for 451 languages

- Smith‘s Phoneme Inventories (N. Smith)
  Phoneme and lexical tone inventories for 111 languages

The component databases (III)

- Anaphora Typology database (A. Dimitriadis, M. Everaert, E. Reuland, T. Reinhart)
  examples of reflexives with analysis; only a few languages are in the database

- Berlin database of intensifiers and reflexives (V. Gast, D. Hole, E. König, P. Siemund, S. Töpper)
  properties and examples for over 100 languages

- Graz database on reduplication (B. Hurch, V. Mattes, O. Konovalova)
  phonology, morphology and semantics of reduplication, with information on productivity and diachrony

The component databases (IV)

- World color survey (P. Kay, B. Berlin, L. Maffi, W.R. Merrifield)
  Summary information on color term systems
- Topic-focus database (E. Aboh, K. Hengeveld)
- Free Personal Pronoun System (N. Smith)

Auxiliary resources

- ISO 639-3 language codes
  Three-letter codes (the former Ethnologue/SIL codes)

- Genetic affiliation according to the Ethnologue (SIL International)

- Geographic coordinates
  Geographic location of languages(M. Dryer/WALS, and G. Segerer)

- Universal Phoneme Positioning Chart
  Table of potential phonemes, derived from UPSID data with additional processing

In the process of being added

- Berlin-Utrecht reciprocals survey (M. Everaert, E. König, V. Gast, A. Dimitriadis, C. Emkow, T. Hanke)
  Inventory of reciprocal markers, with some morphosyntactic and semantic information
- African Anaphora Project (K. Safir, O. Adesola, C. Linares-Scarcerieau)

Next:

- Overview of the TDS
- Managing differences between databases
- The component databases
- The TDS server (demonstration*/tutorial)*
- *The TDS under the hood*
- *Guidelines for component databases*

http://languagelink.let.uu.nl/tds/

- The TDS interface relies heavily on JavaScript support in the browser

- Supported browsers
  - Firefox
  - Internet Explorer

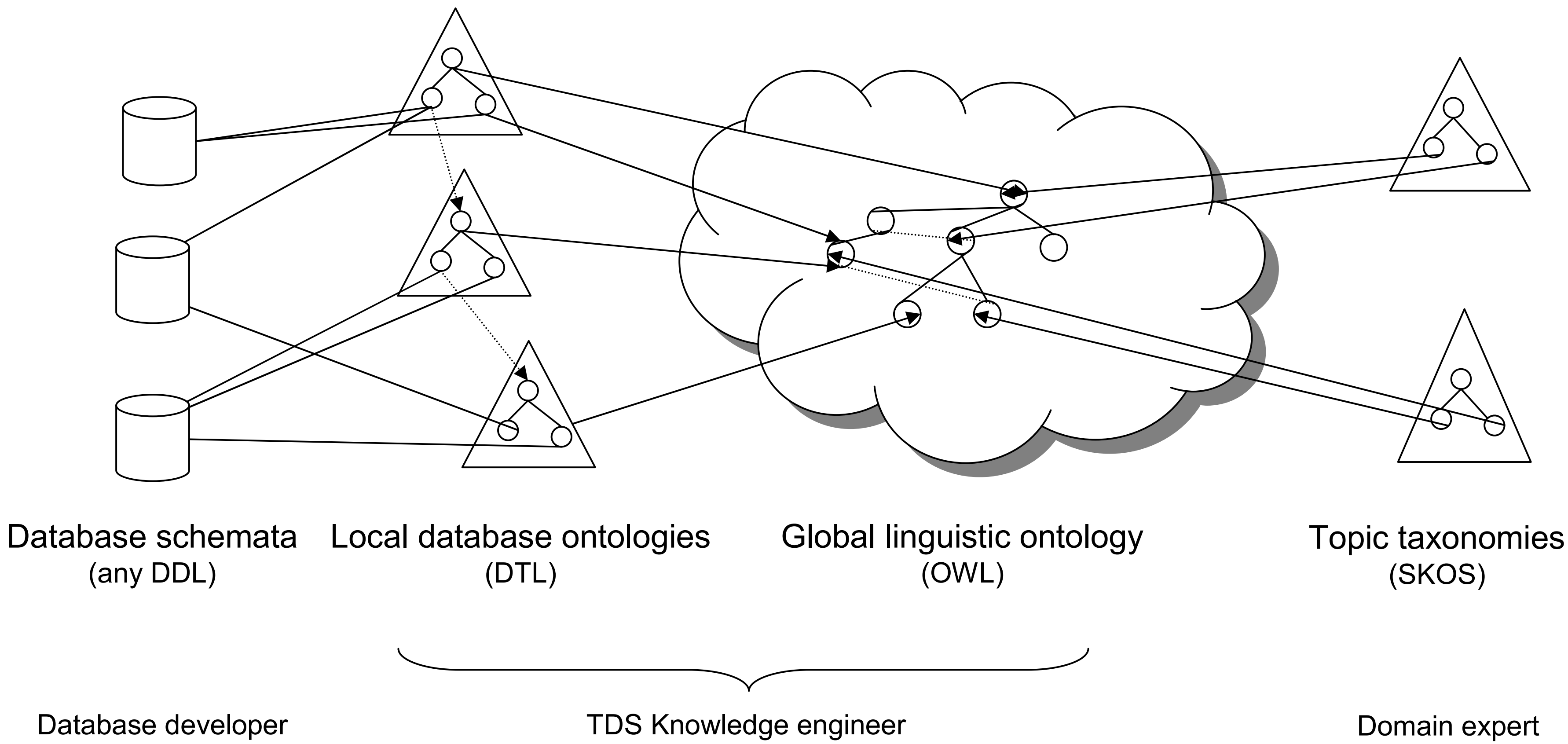
- TDS is a bit heavy on the client side, depending on your computer occasionally timeouts may occur
  - on the TDS homepage you find some hints on how to avoid the timeouts

- The back button might not always do what you expect
  - use the mechanisms of the TDS interface

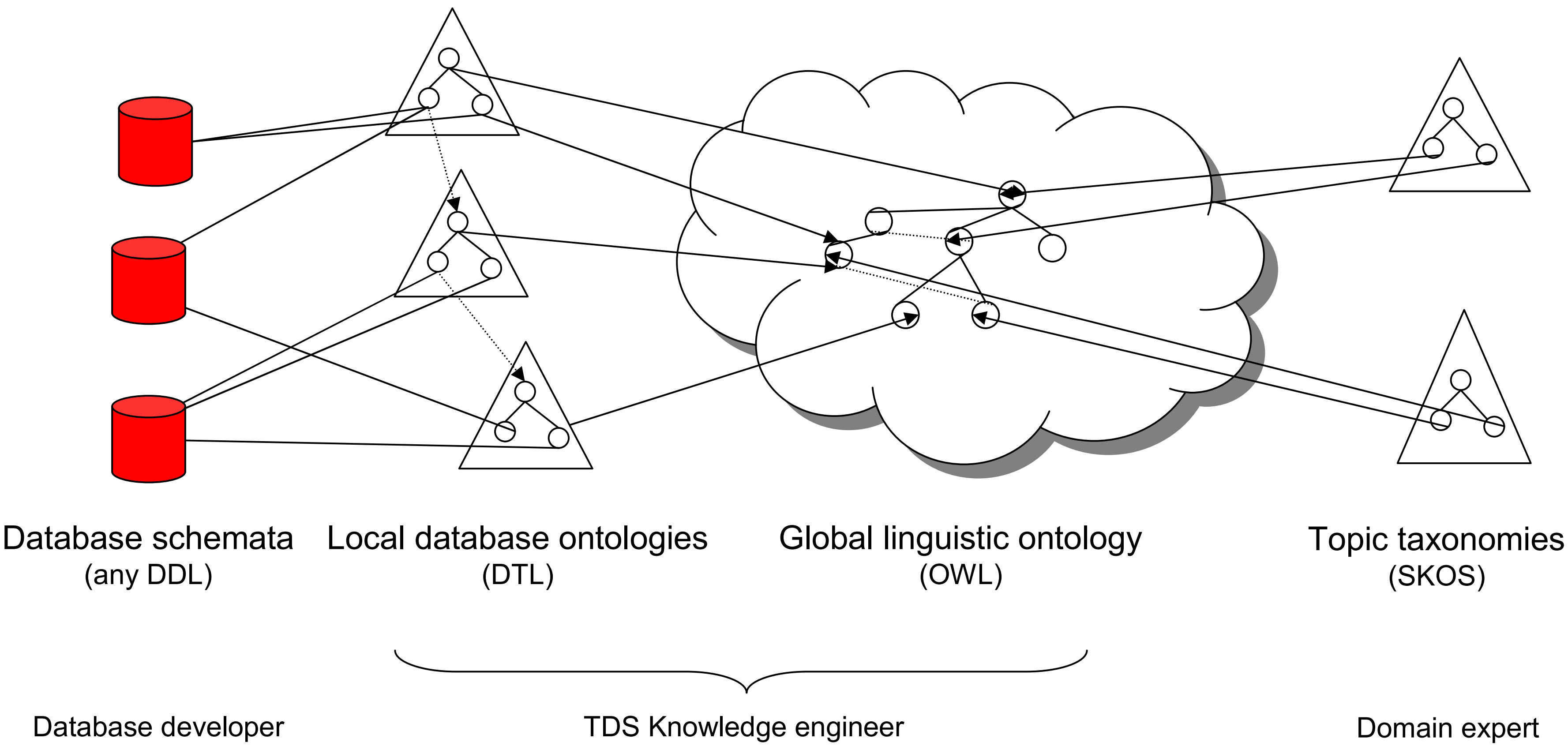
Next:

- Overview of the TDS
- Managing differences between databases
- The component databases
- The TDS server (demonstration/*tutorial*)
- *The TDS under the hood*
- *Guidelines for component databases*

TDS metadata architecture



TDS metadata architecture



Incoming database schemata

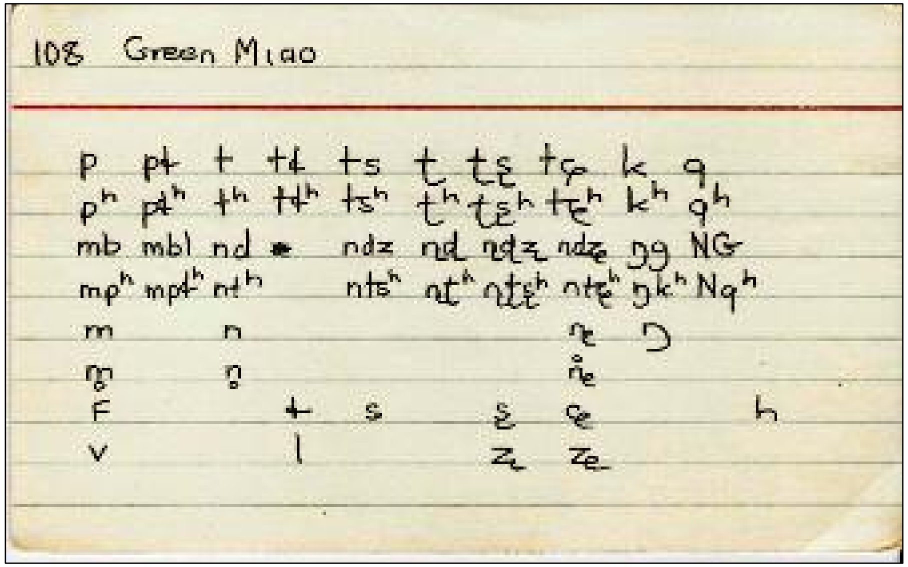
- Any schema associated with the source data
- Preferably accompanied by metadata
- Frequent problems for integration process:
  - Metadata isn't rich enough, or there is no metadata at all
  - Even for well documented databases, metadata not precise enough for our purposes
  - Semantics are often "hidden" in the UI (if exists) and not represented in the database schema
  - Database schema often not fully normalized, *e.g.,* single table
  - A lot of the required information only exists in the developer's head

Database examples

- Original schema snippet from TDN database:

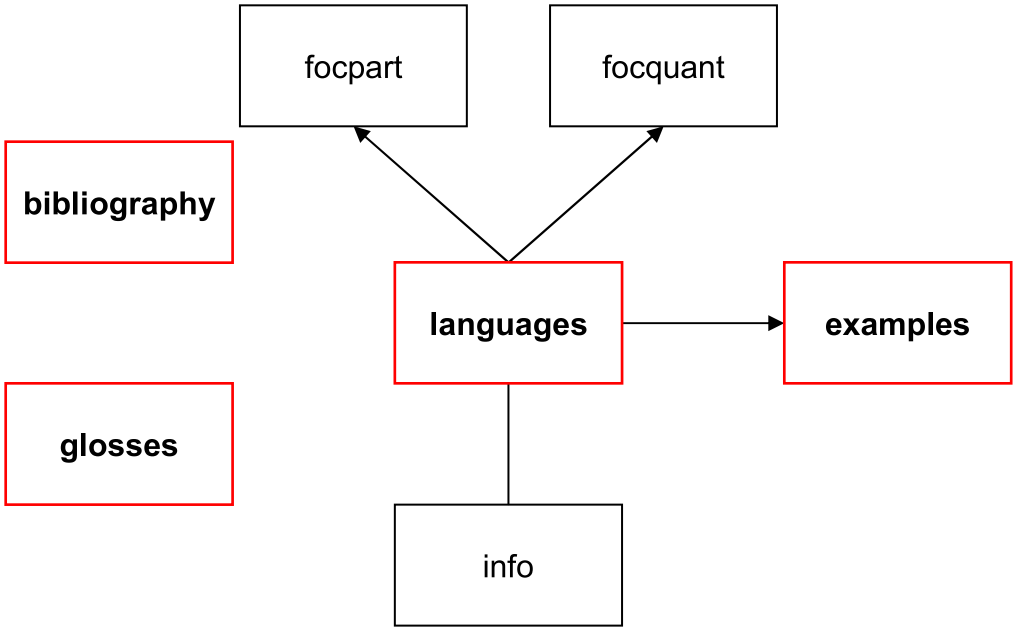
Field	Values	Metadata
V105	0, 1, 9, 99	ATTRIBUTIVE ADJECTIVES ARE RELATIVE CLAUSES
V168	0, 1, 9, 99	PRED LOC = ZERO + LOC PP
V204	0, 1, 9, 99	PRED ADJ = COP VS. PRED LOC = VERB (NONCOP)

- Original schema snippet from SPIN database:

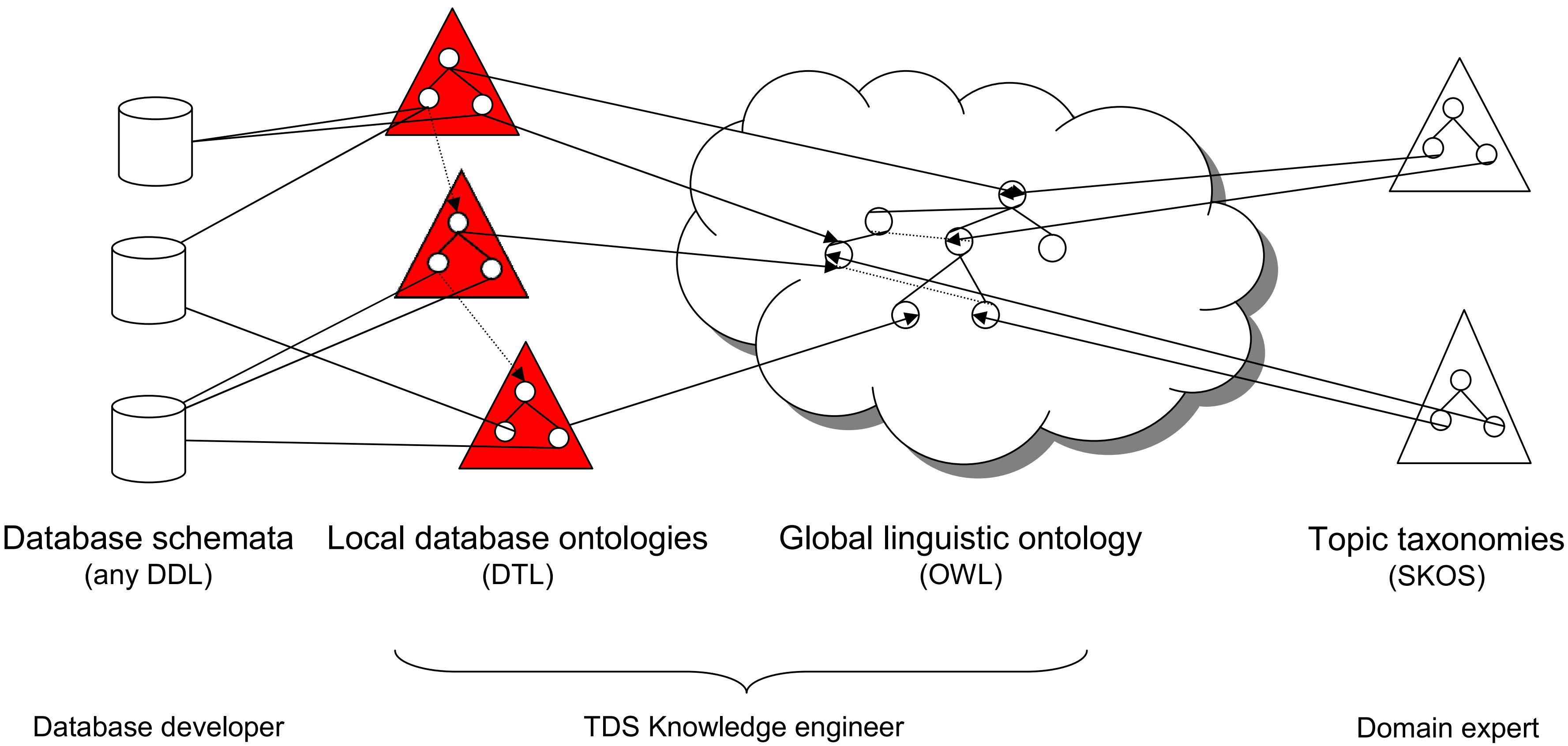


Database examples

- Diagram of the table schema of the TDIR database



TDS metadata architecture



Local database ontologies

- We have developed the special-purpose Data Transformation Language (DTL), which specifies a hierarchical overlay on component databases.

  The nodes in the hierarchy play specific roles:

  - *Field* and *value notions* are associated with database fields and/or values
  - *Concept notions* have links to concepts in the domain ontology
  - *Grouping notions* build the hierarchical structure and keep related notions together, and
  - *Root notions* identify key data structures

An example from the DTL

```
TOP NOTION tdn:locationalPredicates
  LABEL       "Locational predicates"
  DESCRIPTION "Information concerning locational predicates, including form of,
               and conditions on, construction, and form of the negation."
  LINK TO CONCEPT locationalPredicate
  GROUPS {
    NOTION tdn:ZeroEncoding
      LABEL "Locational predicate is zero“
      LINK TO CONCEPT conditionsOnEncoding
      GROUPS {
        NOTION tdn:v168_Zero_plus_locative_prepositional_phrase
          LABEL "Locational predicate is zero + locative prepositional phrase"
          DESCRIPTION "The locational predicate is expressed without the use of
                       an overt verb, but has a locative prepositional phrase."
          IS FIELD v168
          GROUPS WHEN "yes" {
            NOTION tdn:v169_Zero_for_present_only IS FIELD v169;
            NOTION tdn:v170_Zero_in_positive_sentences_only IS FIELD v170;
          }
      }
}
```

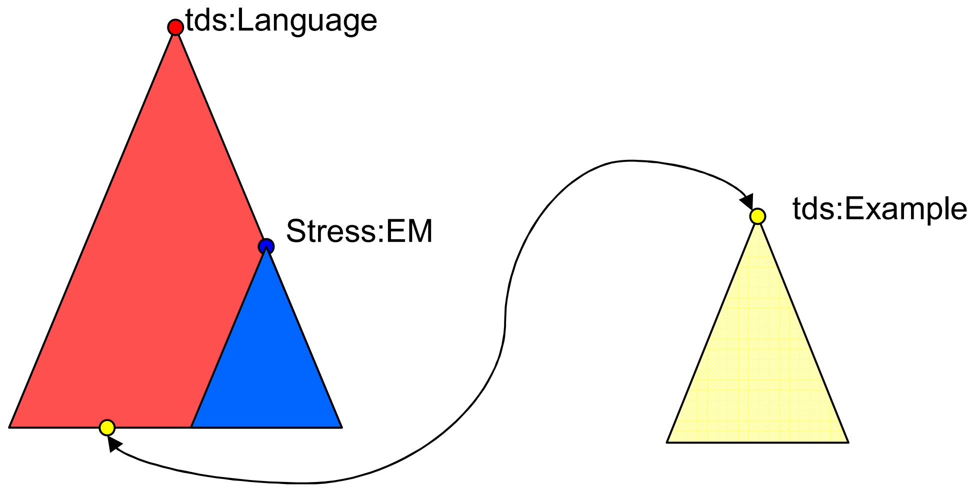
Concept notion

Grouping notion

Field notion

DTL notion hierarchy

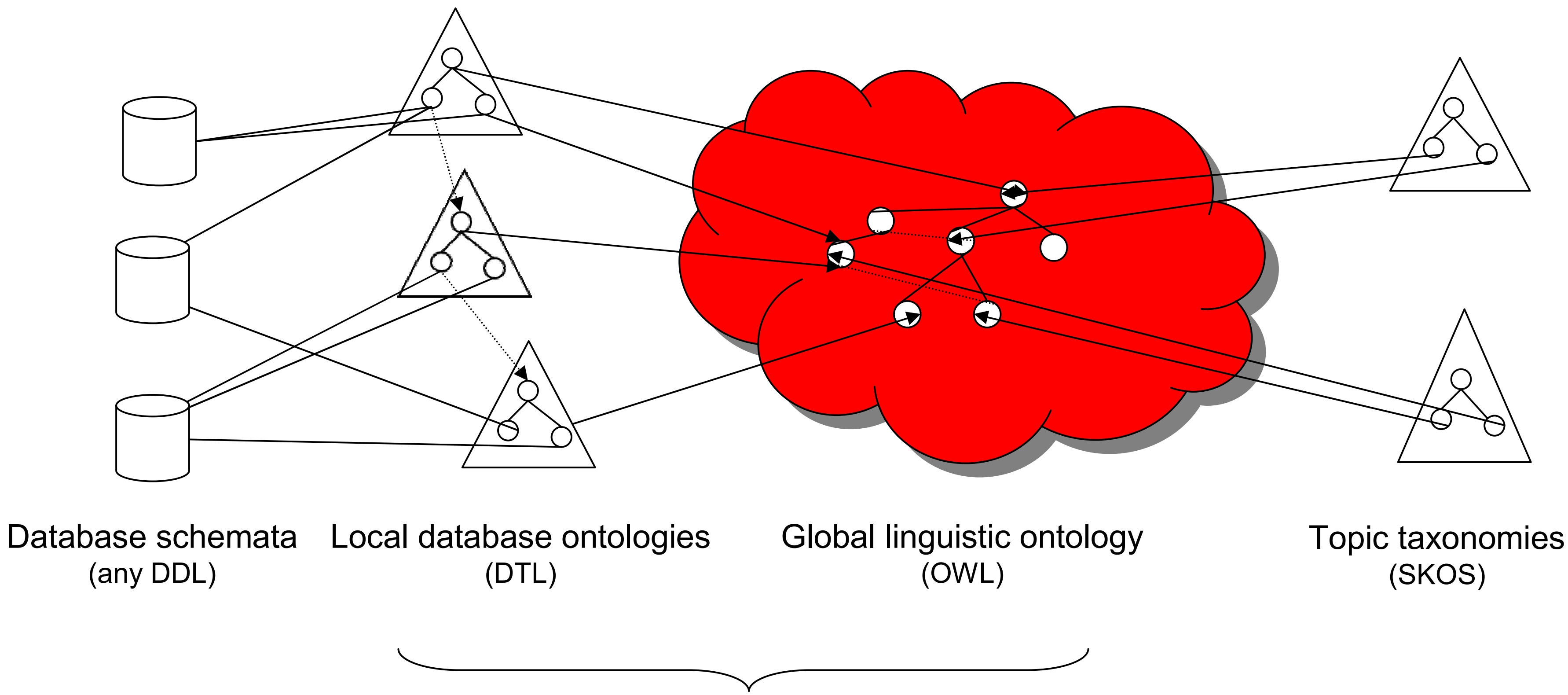
- Notions live in a hierarchy
- The hierarchies are split into semantically coherent contexts
- A DTL specification can describe and relate multiple hierarchies



Other DTL facilities

- Preprocess data
  - General cleanup of data before being processed by the rest of the DTL specification
- Uncertainty handling
  - Example: database value “”L?” is normalized to “left” marked as UNSURE
  - Will allow support for different levels of uncertainty handling during query time:
    - Certain: never selects and never projects marked values
    - Normal: ignores markers
    - Uncertain: marked values are always selected and are thus always projected (they can be any value)
- Some notions can be marked as (general) annotation notions, their data is accessible when data from the parent or root notion is projected.
- Allows the declaration of loosely reusable notion hierarchies.

TDS metadata architecture



Database developer

TDS Knowledge engineer

Domain expert

Global linguistic ontology

- We have developed a custom, bottom-up ontology as required for our integration needs.
- Design principles:
  - *Bottom-up approach:* concepts are only established on the basis of generalization from information existing in component databases
  - *Inclusive perspective:* provides a common vocabulary that serves as a non-prescriptive basis for integration of database- and theory-specific categories
- Content:
  - Unifying concepts are established on the basis of local DTL notions
- Implementation:
  - The ontology is specified in the W3C recommendation Web Ontology Language (OWL)

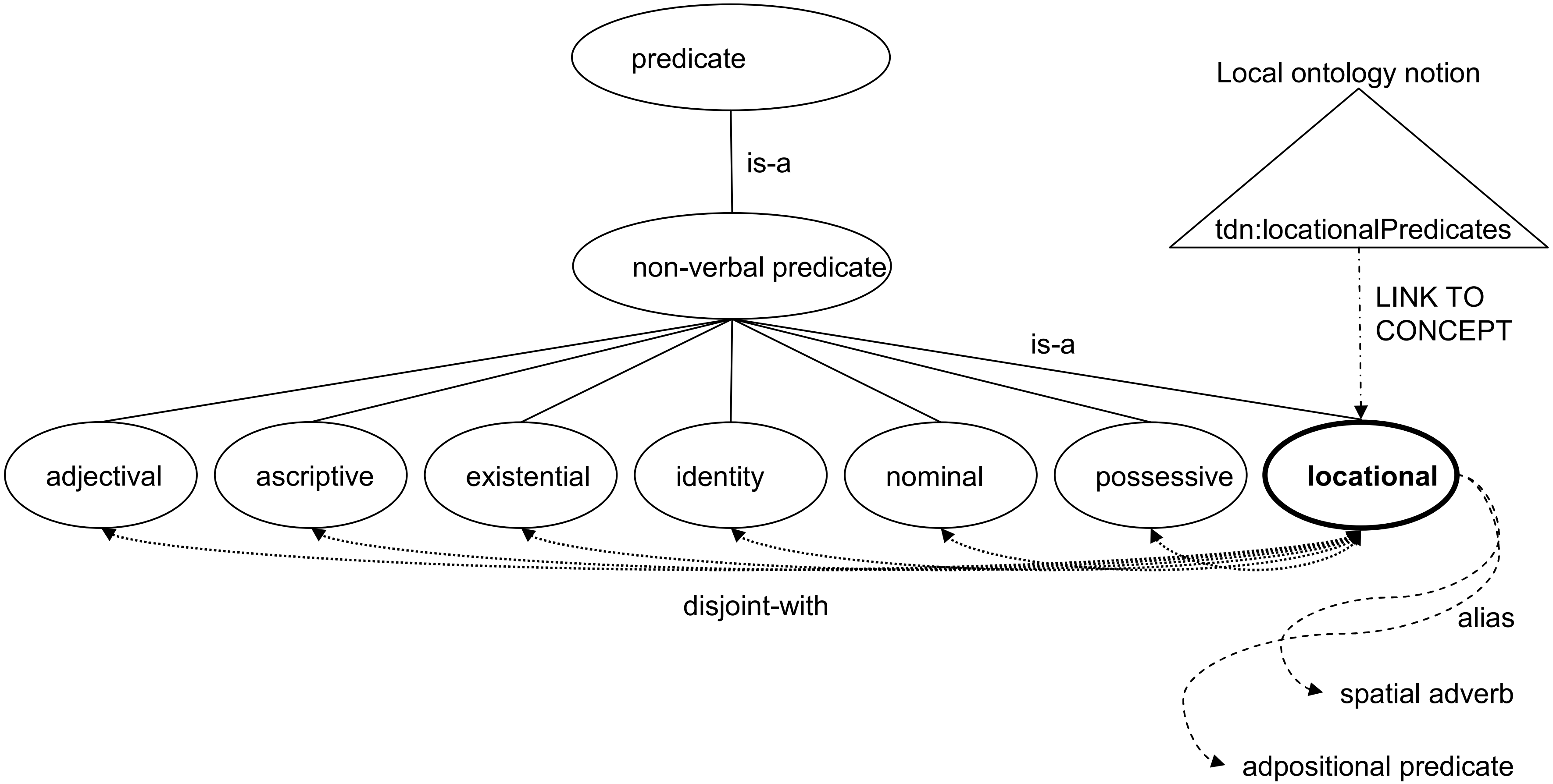
Ontology: linguistic concepts

- *Linguistic objects* can be thought of as existing in themselves, *e.g.* Sentence and Morpheme;
- *Linguistic properties* are (linguistically salient) properties predicated of a linguistic object, *e.g.* Basic Word Order and Referential;
- *Linguistic relations* model a phenomenon involving two or more linguistic objects or properties, *e.g.* Agreement and Stress Assignment

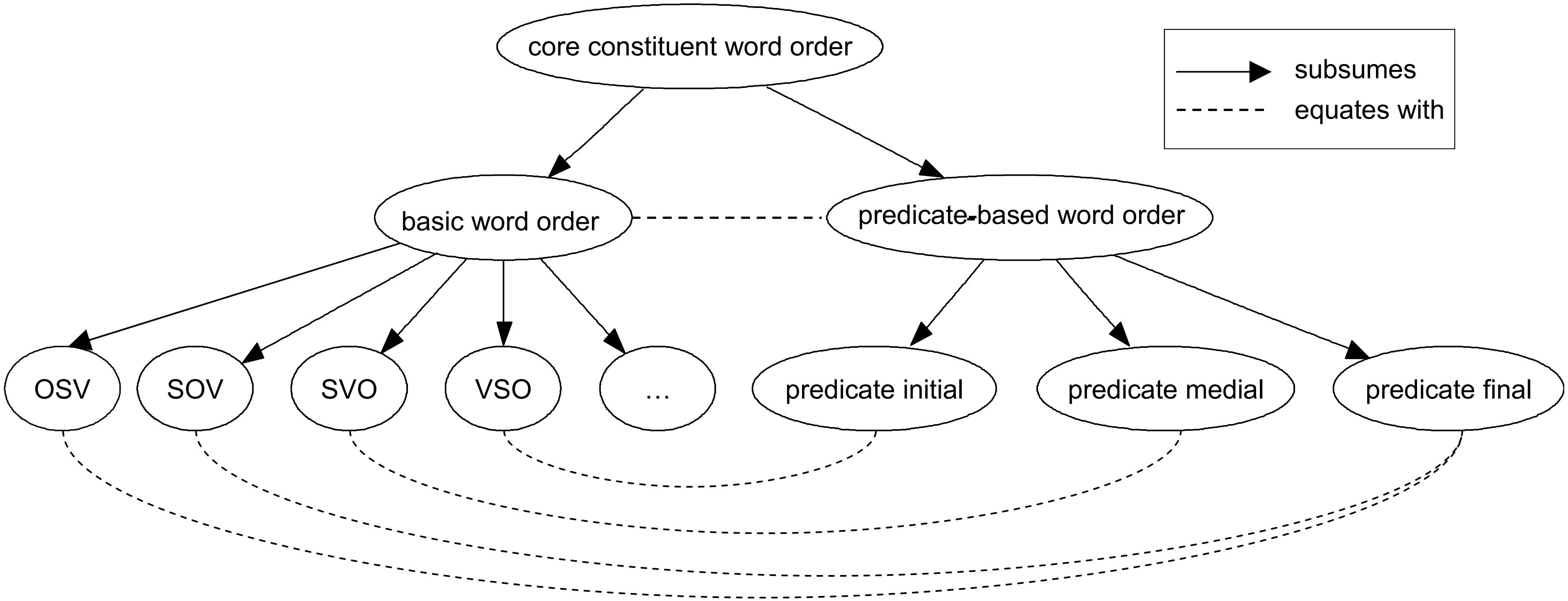
Ontology: relationships

- *Subsumption:* super- and subordinate concepts;
- *Loose synonymy:* variant linguistic terminology used to refer to the same phenomenon;
- *Related phenomena:* variant linguistic terminology used to refer to similar or related phenomena;
- *Meronymy:* part/whole relations;
- *Determination:* a linguistic property is defined in terms of one or more other linguistic properties;
- *Form-function relationship:* the linguistic function served by some linguistic entity.

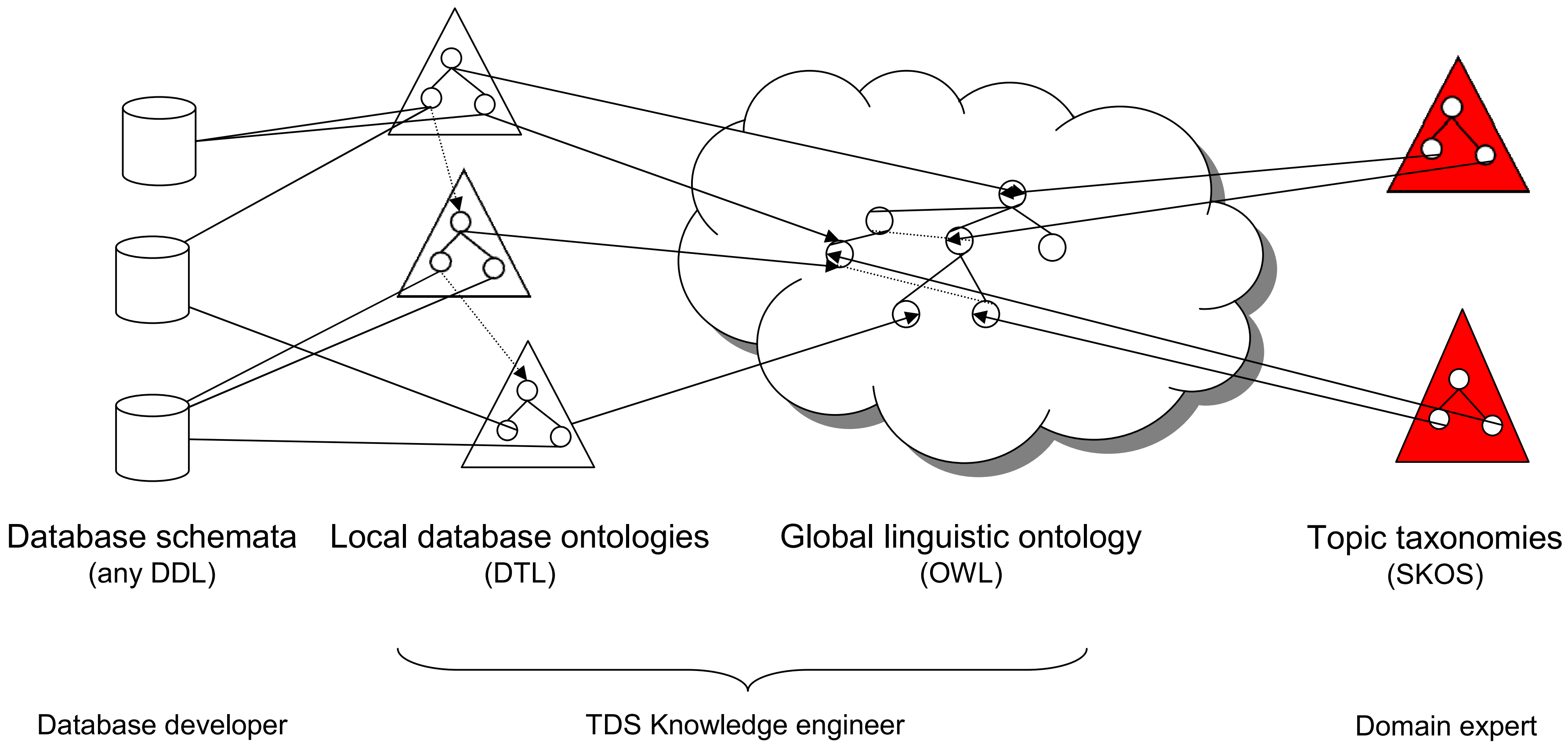
An example from the global ontology



An example from the global ontology



TDS metadata architecture



Topic taxonomies

- Thematic groupings of topics
  i.e., not strict subsumption relations
- Purpose is to provide alternative domain-specific entrance points to concepts and associated notions

- Current taxonomies:
  - Table of contents from *Describing Morphosyntax* (Payne, 1992)
  - The subsumption hierarchy of the global linguistic ontology
  - The BRILL classification hierarchy (under construction)

- Implementation:
  - Taxonomies are specified in the new W3C working draft
    Simple Knowledge Organisation System (SKOS)

Next:

- Overview of the TDS
- Managing differences between databases
- The component databases
- The TDS server (demonstration/*tutorial*)
- *The TDS under the hood*
- *Guidelines for component databases*

Good database design

- A properly designed database is easy to enter data into and extract information from; it is also easier to modify as one's research design evolves.

Document assumptions and procedures

- Typological databases have a long lifespan.
- Assumptions and procedures are lost, which may lead to inconsistencies.
- Document the database to make it possible
  - to refresh the project's collective memory
  - to keep data clean and consistent
  - to provide rich metadata if the database is eventually made public or reused

Citations to the sources of information

- Essential for error-checking or further research
- List at least the source(s) of information for each language
- Put these sources in a seperate table so they can be easily referenced
- Ideally each group of information in the database would include a seperate citation with relevant page numbers

Key values

- Look for standars to take your key values from:
  - Languages: ISO 639-3
  - Dialects: ISO 639-3 + dialect name (Ethnologue)
  - Phonemes: Unicode codepoints (IPA Console)
  - …

Comment fields

- Provide separate comment fields for separate fields or topics requiring comment. Be explicit about the field to which a comment applies.
- If you decide on new types of data to collect, don't store it in a general comment field. Make separate fields, even if they will be sparsely filled.

NULL values

- NULL values are controversial as they can mean many things:
  - Is the field irrelevant?
  - Should a default value be used?
  - Is there no value yet?
  - Did analyst search for a value but didn’t find it yet?
- Make all these circumstances explicit, and let NULL have no or one meaning

Uncertainty

- Encoding uncertainty in a value is a poor strategy:
  - ‘X’and ‘X?’ are 2 different things for a DBMS
  - ‘1?’ has to be stored in a text field, while the proper value is a numeral
- Ideally, encode uncertainty in a seperate field
- At least use a consistent notation and document it

- More generally, elaborate embedded notation for values is difficult and error prone. Use multiple fields as needed.

Future work

- Performance/stability improvements
- Import more databases
  - ODIN
  - ZAS
  - … yours?
- The TDS as a basis for the preservation of databases
  - Archiving typological databases (IDDF)
- The TDS as a web service
  - CLARIN
  - TypEx
- The TDS as a data integration framework
  - Other (scientific) domains?

http://languagelink.let.uu.nl/tds/

Any feedback is welcome ☺

TDS-fgw@uva.nl