A TDS perspective on interoperability and sustainability

Menzo Windhouwer

Outline

- Typological Database System
  - Introduction
  - System architecture
  - Problems encountered
- Interoperability
  - Sharing structure
  - Sharing semantics
  - Sharing services
- Sustainability
  - Archiving databases
  - Archiving documentation
- TDS Future

ISOcat

IDDF

Typological Database System

- 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/

TDS: 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
  - CSV files, MS Access, MySQL, PostgreSQL, FileMaker, …

- 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

TDS: 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.

TDS: the approach

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

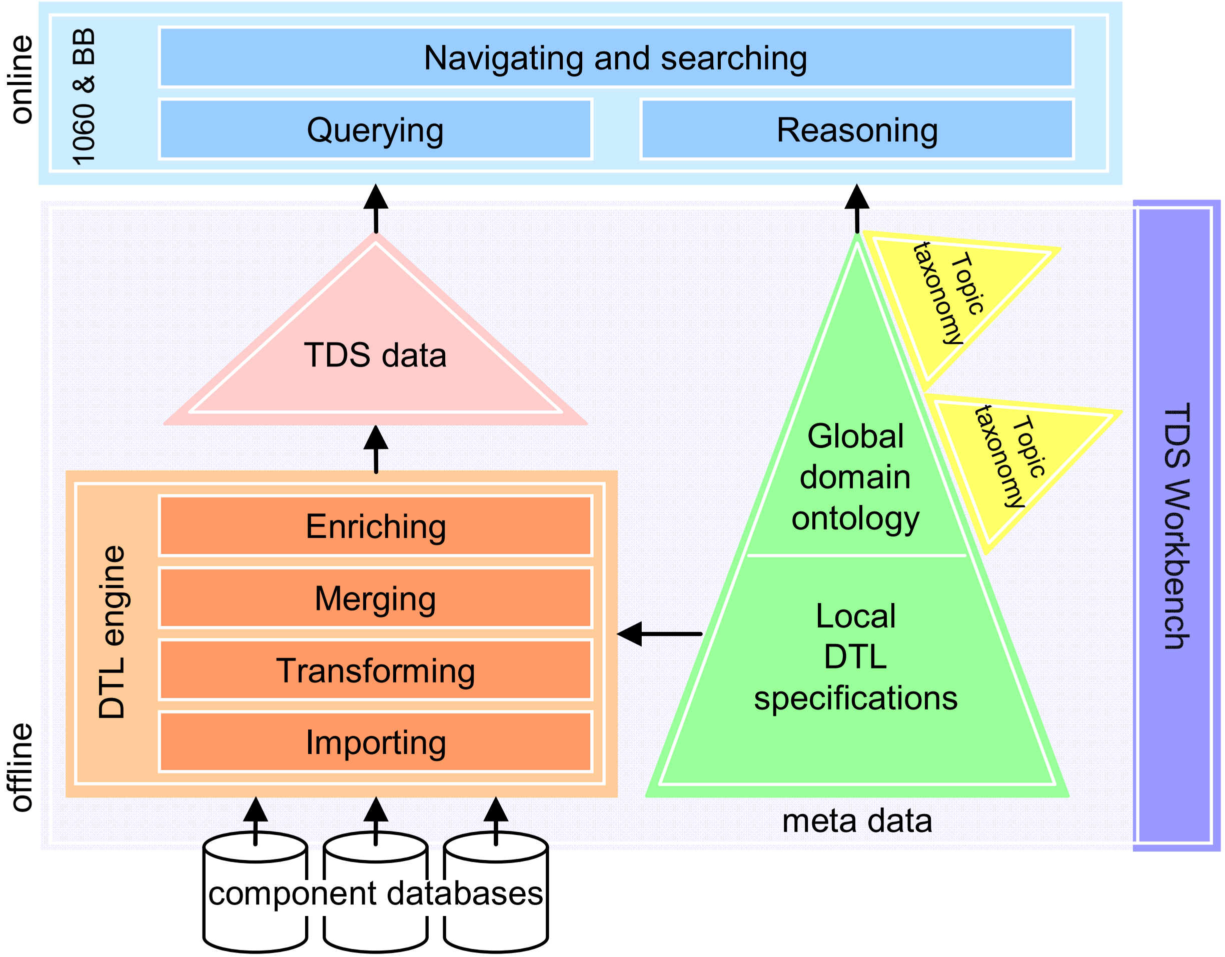
TDS: how databases are integrated

- 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.

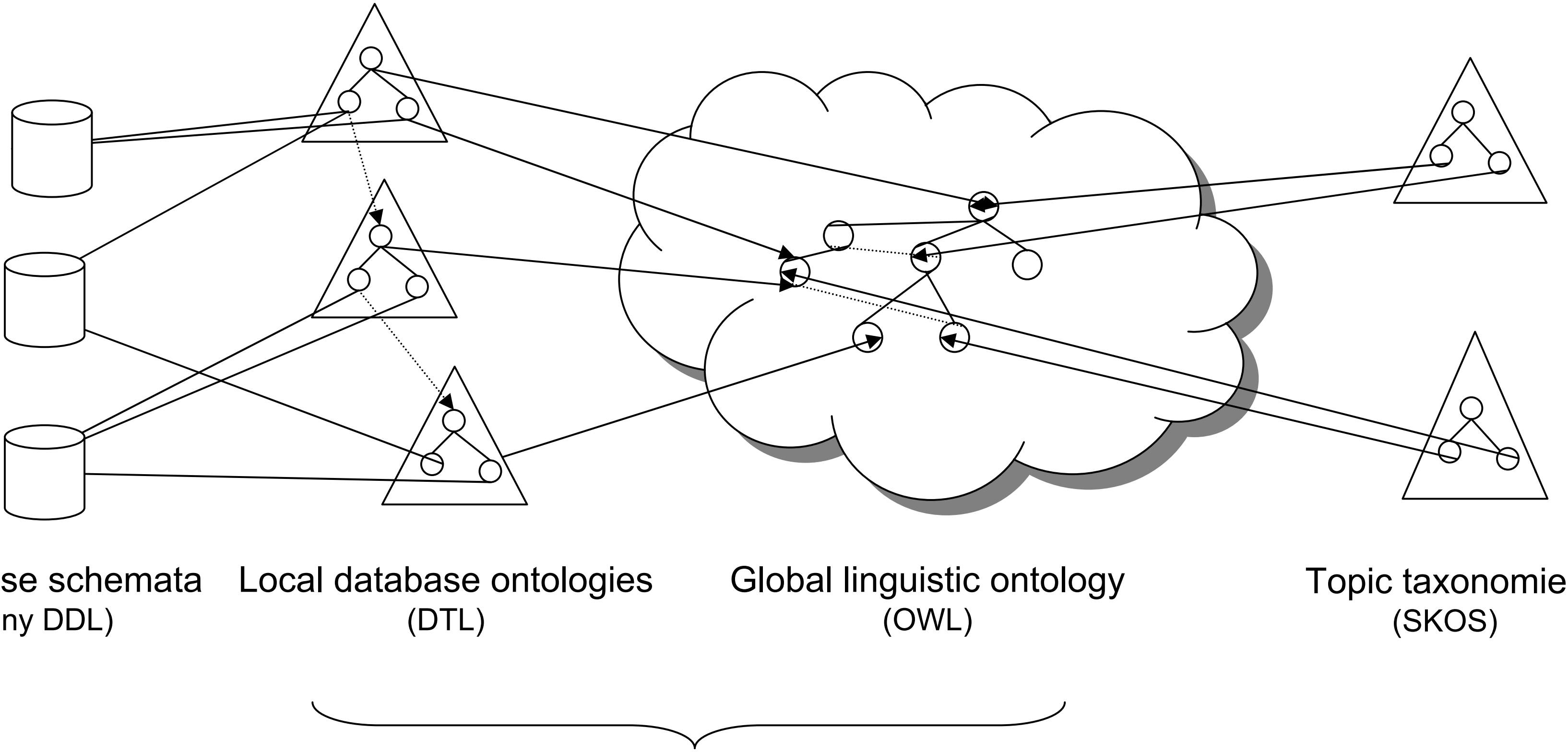
TDS: how databases are integrated (II)

- 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



TDS: metadata architecture



Database schemata
(any DDL)

Local database ontologies
(DTL)

Global linguistic ontology
(OWL)

Topic taxonomies
(SKOS)

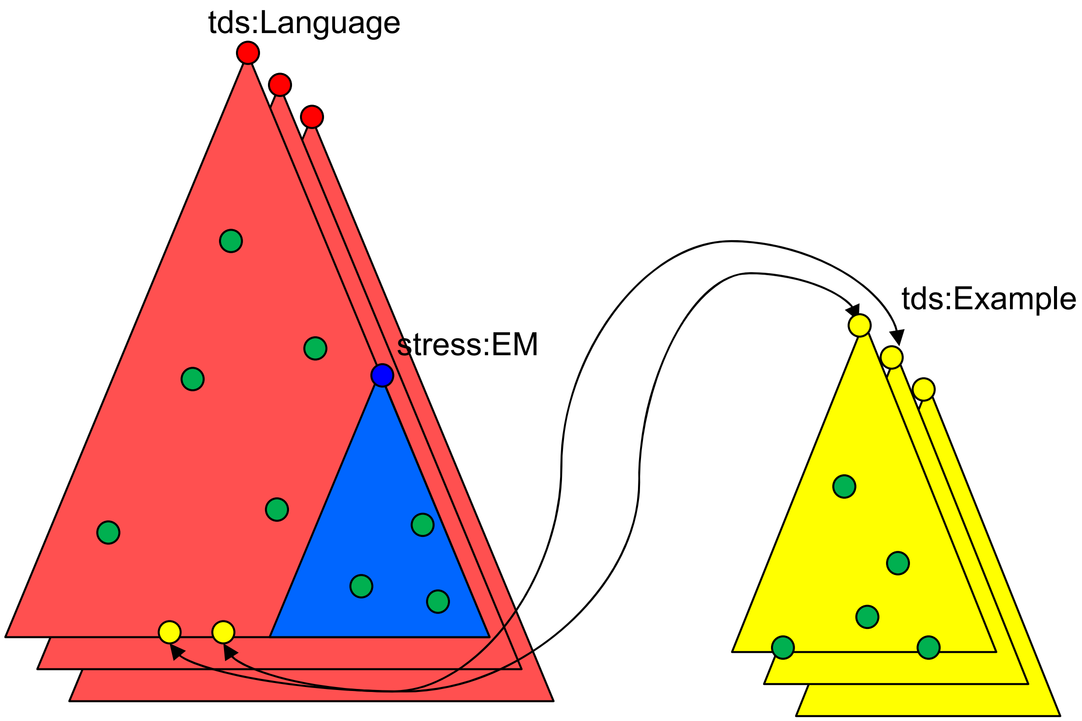
Database developer

TDS Knowledge engineer

Domain expert

TDS: data structure

- Data lives in a forest of trees
- The trees are split into semantically coherent contexts



TDS: problems encountered

- Lack of documentation
  - It takes a lot of time to dig up and to encode the semantics
- A number of formats/APIs
  - A set of CSV files
  - ODBC accessible databases (MySQL, PostgreSQL, MS SQL Server)
  - ODBTP accessible databases (MS Access databases, FileMaker)
  - XML documents
  - …
- Many models/encodings
  - Under or over normalized databases (universal tables)
  - Too much structure in a data unit (uncertainty/comments/…)
  - Reverse engineering the model (data catalog)
  - …

Interoperability: sharing structure

- There is no standard for database dumps:
  - SQL implementations are not standard enough
  - CSV files are too limited (no field names, no types, no metadata)
- Some proposals:
  - Many CSV to XML mappings
  - Conceptual structure to XML mappings
    - exchange formats with domain specific mappings
    - … remember Alexis presentation of yesterday
  - Various from academic papers/archives:
    - MIXED from DANS
    - IDDF from TDS
    - … more aimed at sustainability, not so much for exchange

Interoperability: sharing semantics

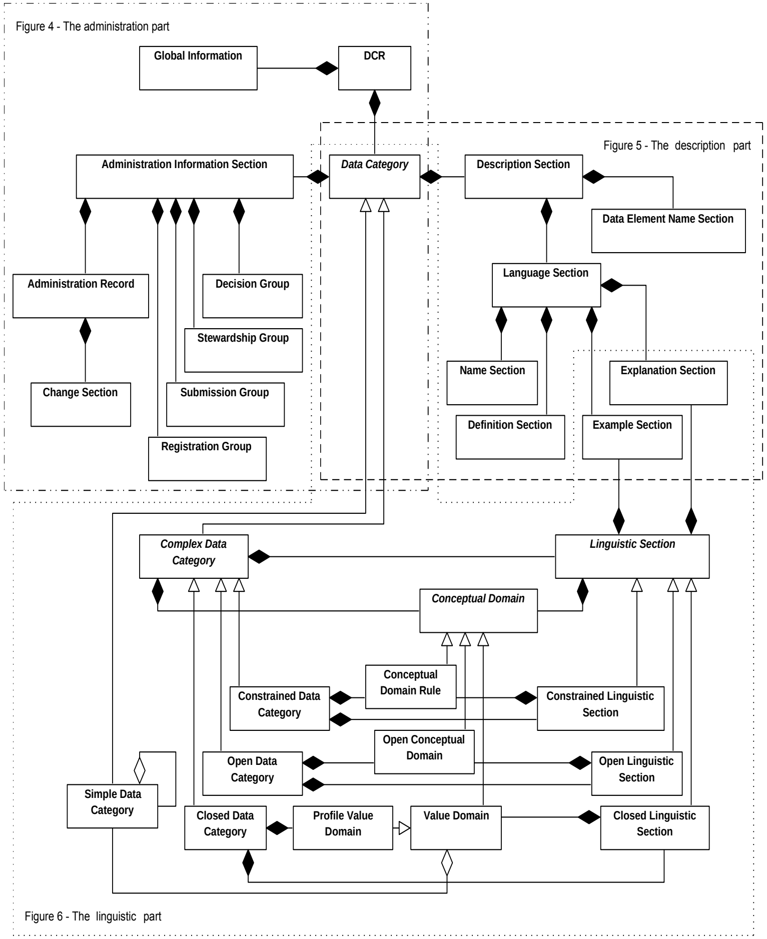
- In the TDS metadata architecture local (databases specific) ontologies link into a global (domain specific) ontology, i.e., they share some semantics
- The concepts could be reused outside of the TDS and the TDS could reuse concepts from other projects
- ISO Technical Committee 37 *Terminology and other language and content resources* is working on a concept registry based on ISO 12620

ISOcat

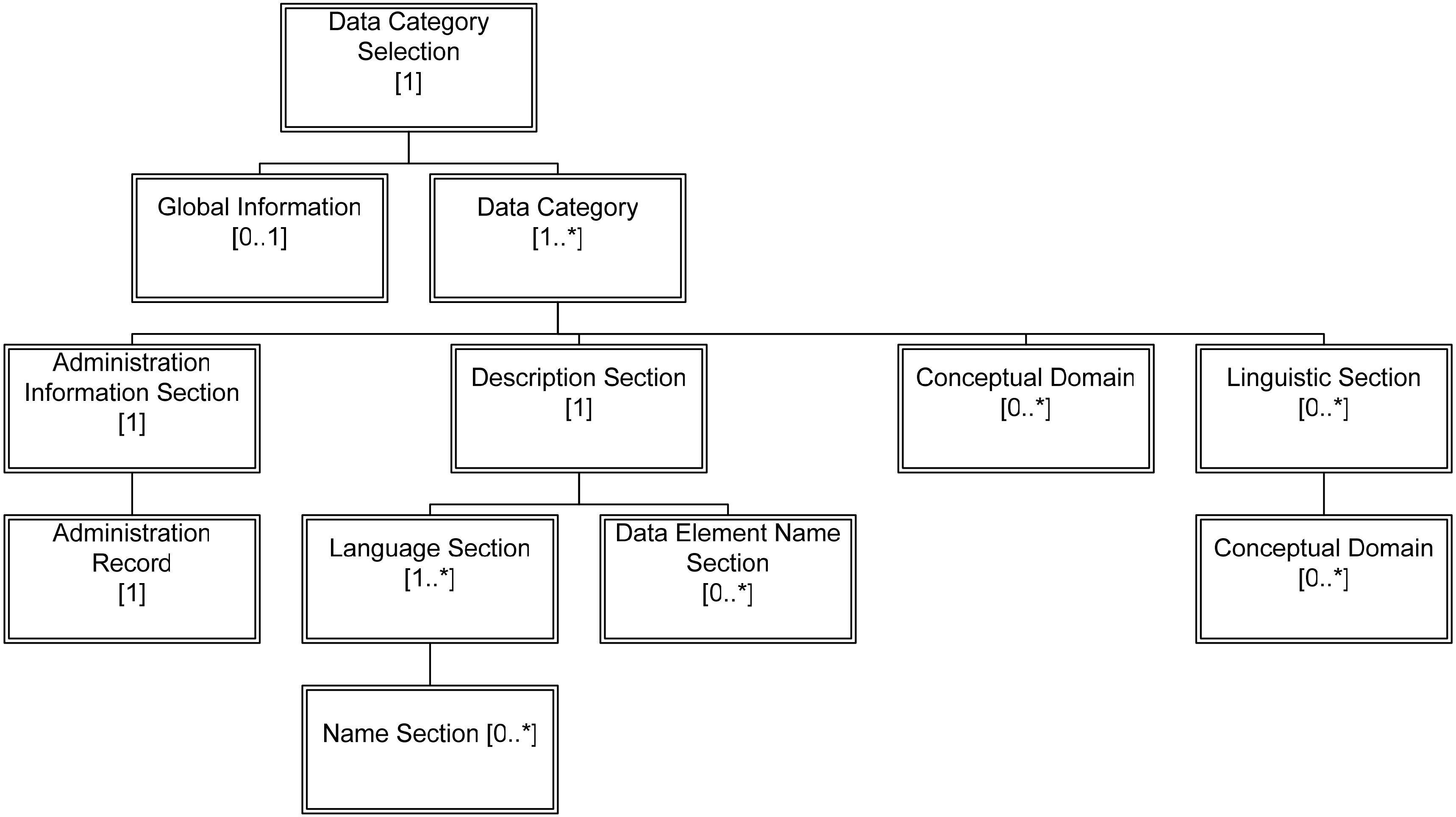
- In ISOcat each concept gets a persistent identifier (PID)
- By including this PID in their metadata, e.g., schemata, resources can indicate their shared semantics
- Everyone can enter the concepts they need and share them/make them public
- Eventually concepts can become ISO standards

- We plan to move the TDS ontology into the ISOcat registry

ISOcat: data model



ISOcat: data model (II)



Data Category Selection [1]

Global Information [0..1]

Data Category [1..*]

Administration Information Section [1]

Administration Record [1]

Description Section [1]

Language Section [1..*]

Name Section [0..*]

Data Element Name Section [0..*]

Conceptual Domain [0..*]

Linguistic Section [0..*]

Conceptual Domain [0..*]

ISOcat: data model (III)

- Data category:
  - result of the specification of a given data field
- Basically a flat list of data categories
  - except for relations between simple and complex data categories
  - … in the future a Relation Registry will support more relationships
- Types of complex data categories:
  - Open: any value of a given data type
  - Constrained: value constrained by a rule
  - Closed: enumeration of simple data categories
  - Value domains can be further restricted for specific languages
- Each data category needs to have:
  - an english name
  - an english description
  - a justification

ISOcat: Thematic Domain Groups

TDG 1: Metadata

TDG 2: Morphosyntax

TDG 3: Semantic Content Representation

TDG 4: Syntax

TDG 5: Machine Readable Dictionary

TDG 6: Language Resource Ontology

TDG 7: Lexicography

TDG 8: Language Codes

TDG 9: Terminology

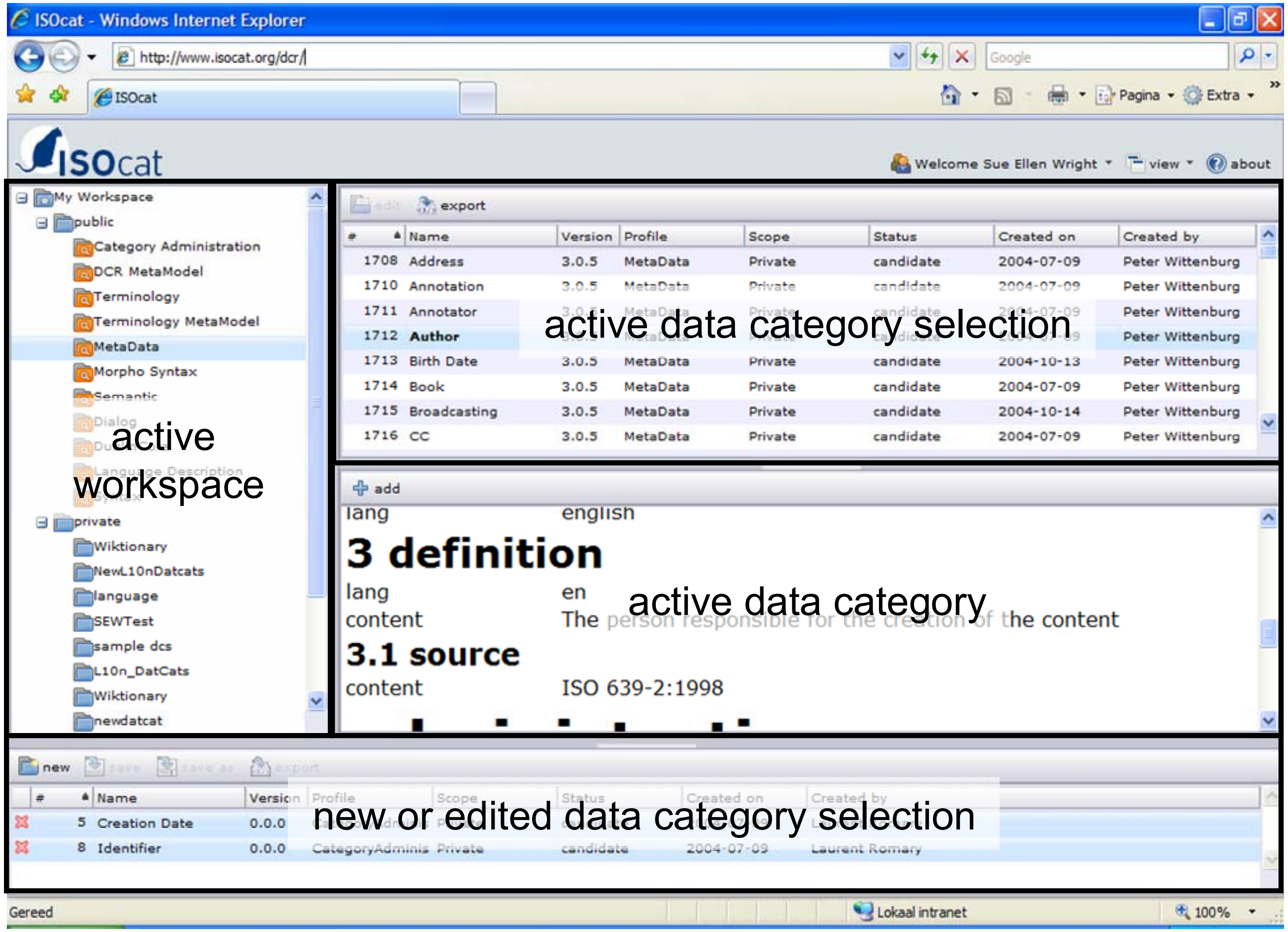
TDG 11: Multilingual Information Management

TDG 12: Lexical Resources

TDG 13: Lexical Semantics

TDG 14: Source Identification

ISOcat: web user interface



#	Name	Version	Profile	Scope	Status	Created on	Created by
1708	Address	3.0.5	MetaData	Private	candidate	2004-07-09	Peter Wittenburg
1710	Annotation	3.0.5	MetaData	Private	candidate	2004-07-09	Peter Wittenburg
1711	Annotator	3.0.5	MetaData	Private	candidate	2004-07-09	Peter Wittenburg
1712	**Author**	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	Peter Wittenburg
1713	Birth Date	3.0.5	MetaData	Private	candidate	2004-10-13	Peter Wittenburg
1714	Book	3.0.5	MetaData	Private	candidate	2004-07-09	Peter Wittenburg
1715	Broadcasting	3.0.5	MetaData	Private	candidate	2004-10-14	Peter Wittenburg
1716	CC	3.0.5	MetaData	Private	candidate	2004-07-09	Peter Wittenburg

active data category selection

lang english

3 definition

lang en

content The person responsible for the creation of the content

3.1 source

content ISO 639-2:1998

active data category

active workspace

#	Name	Version	Profile	Scope	Status	Created on	Created by
5	Creation Date	0.0.0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
8	Identifier	0.0.0	CategoryAdminis	Private	candidate	2004-07-09	Laurent Romary

new or edited data category selection

ISOcat: RESTful web services

- ISOcat readonly API
  - http://www.isocat.org/rest/user/guest/workspace
  - http://www.isocat.org/rest/tdg/9
  - http://www.isocat.org/rest/dc/1234
  - …
- Use the Accept HTTP request-header field to request a resource representation, the default is (where applicable) DCIF (Data Category Interchange Format)

ISOcat: embedding PIDs

- Some schema languages have built-in facilities to embed the PIDs
  - ODD

```
<elementSpec ident="pos">
  <equiv name="partOfSpeech"
         uri="http://www.isocat.org/dc/ISO-DC-1345"/>
    <!-- additional specifications here -->
</elementSpec>
```

  - XCS (only complex DCs)

```
<datCatSet>
  <termNoteSpec name="animacy
      datcatId="http://www.isocat.org/dc/ISO-DC-78">
    <contents datatype="picklist" forTermComp="yes">
       animate inanimate otherAnimacy
    </contents>
 </termNoteSpec>
</datCatSet>
```

ISOcat: embedding PIDs (II)

- The DC Reference XML vocabulary can be used to annotate schemas or resources without built in facilities:
  - Relax NG:

```
<element name="identifier"
       dcr:datcat="http://www.isocat.org/datcat/DC-8">
  <data type="string"/>
</element>
```

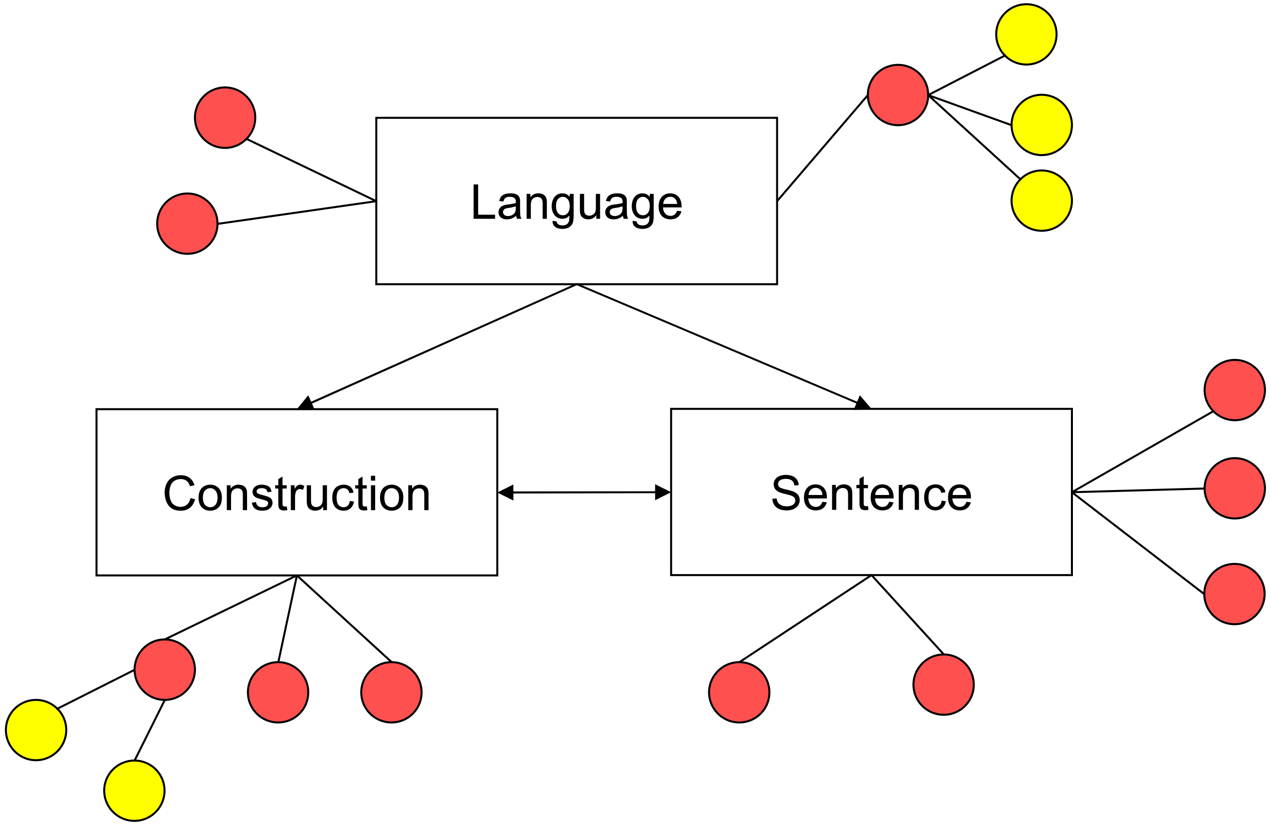
  - XML Schema:

```
<xs:element name="“identifier">
  <xs:annotation>
    <xs:appinfo>
      <dcr:datcat pid="http://www.isocat.org/datcat/DC-
  8"/>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
```

ISOcat: meta models

- ISO (TC 37) is standardizing meta models:
  - Typological Markup Framework (TMF)
  - Lexical Markup Framework (LMF)
- For a specific application you instantiate (parts of) these models and populate them with data categories
- The language/construction/example model Alexis presented yesterday, can also be seen as such a meta model …

ISOcat: meta models (II)



ISOcat: status

- Beta version is online
  - Open for everyone

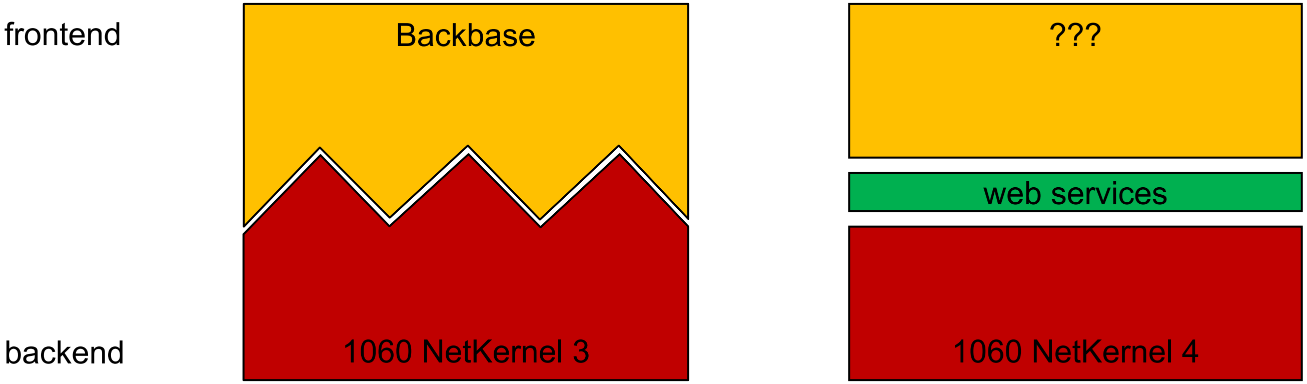
http://www.isocat.org/

- Near future:
  - Sharing concepts
  - Coediting concepts
  - TDGs will become (more) active
- Future:
  - ISO standardization workflow
  - mirrors

Interoperability: sharing services

- Currently the TDS is a closed system
- However, it could offer typological web services in an infrastructure as proposed by CLARIN

- To achieve this the current web user interface should be more cleanly separated from the service backend



Interoperability: sharing services (II)

RESTful web services

- Mostly existing standards
- HTTP
  - All verbs (PUT, GET, POST, DELETE)
- Browser accessible
- Any resource representation, prefer HATEOAS
- WADL

WS-* webservices

- A big stack of W3C recommendations
- HTTP
  - POST
- Targeted at tool interaction
- Always a SOAP envelope
- WSDL

Sustainability: archiving databases

- There is no default database dump format
- Even if there was, for archiving purposed storing just the data and the model isn't enough …

Sustainability: archiving documentation

- Just archiving databases isn’t enough
  - What is the actual data model? Shouldn’t need to reverse engineer it
  - What are the semantics of the data model?
  - …
- Partial solutions:
  - Concept PIDs from ISOcat
  - Standard data catalog dump
  - …
- However, still too low level, the broad overview of the theoretical assumptions (scientific domain) is still missing

Integrated Data and Documentation Format (IDDF)

- Data, structuring information and documentation are combined into an integrated, XML-based standardized format, the Integrated Data and Document Format (IDDF).
- Software is provided that can manage IDDF-encoded resources in a generic way, just as a text editor or corpus tool can manage arbitrary conforming resources.
- New generations of management software can be provided in the future, utilizing the self-describing nature of the IDDF and an economy of scale.

IDDF: setup

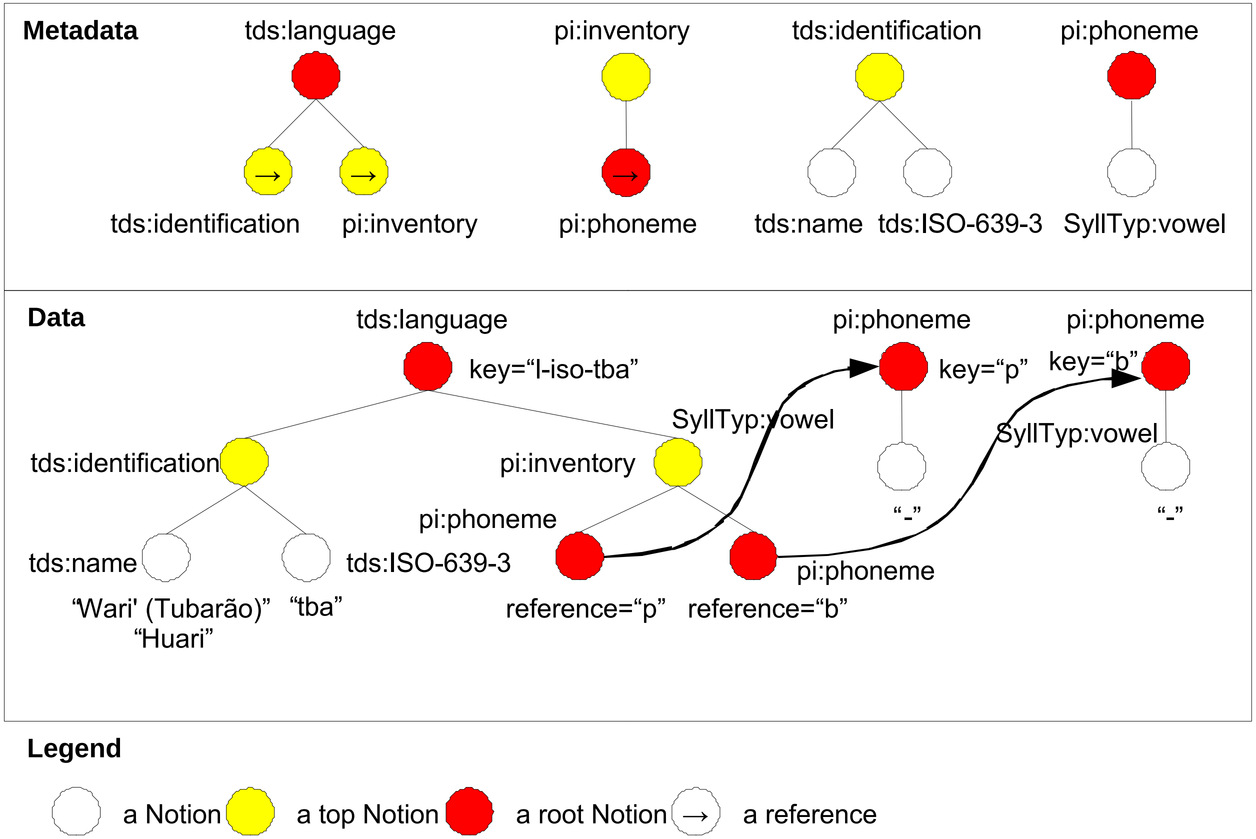
- Two major sections:
  1. Metadata section:
     - provides the (loose) data schema
     - documents the elements in the schema
  2. Data section:
     - contains the actual data
- Readonly, hierarchical, semi-structured data model
- Network of hierarchical units, a.k.a. semantic contexts
- XML vocabulary

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

IDDF: XML document

```
<iddf:warehouse xmlns:iddf="http://.../ns/iddf">
   <iddf:meta>
        <iddf:scope id="tds" type="warehouse">
               ...
        </iddf:scope>
        <iddf:notion id="n1" name="language“ scope="tds“
                                          type="root" key-datatype="enum">
               <iddf:label>Language</iddf:label>
               <iddf:description>
                      One of the world’s languages
               </iddf:description>
               ...
        </iddf:notion>
        ...
   </iddf:meta>
   <iddf:data xmlns:tds="..." ...>
        <tds:language iddf:notion="n1" key="...">
               ...
        </tds:language>
        ...
   </iddf:data>
</iddf:warehouse>
```

IDDF: data model



IDDF: metadata

- A label and a description
- One or more links
  - to other Notions
  - to external resources, e.g., a knowledge base
- Data types:
  - A semantic data type for the Notion, e.g. UPPC
  - A semantic (key) value data type, e.g. interlinear glossed text tier
- An (partial) enumeration of possible (key) values:
  - The literal (key) value
  - A label and a description
  - One or more links
    - to other notions
    - to external resources

- An ISOcat data category PID would be a link to an external resource

IDDF: metadata example

```
<iddf:notion id="n7" name="vowel" scope="SyllTyp">
   <iddf:label>Vowel</iddf:label>
   <iddf:description>
        Is the segment a vowel?
   </iddf:description>
   <iddf:link type=“datcat " rel="as“ href="...datcat/ISO-DC-12"/>
   <iddf:link type="concept" rel="as“ href="...owl#vowel"/>
   <iddf:link type="concept" rel="to“
                                href="...owl#vocalicFeatureNode"/>
   <iddf:values datatype="enum">
        <iddf:value>
               <iddf:literal>+</iddf:literal>
               <iddf:description>
                      The segment is a vowel.
               </iddf:description>
        </iddf:value>
        …
   </iddf:values>
</iddf:notion>
```

IDDF: data example

```
<iddf:data xmlns:tds="…/ns/iddf/tds" … >
   <tds:language key="l-iso-tba"
                    iddf:notion="n1" iddf:sources="SyllTyp UPSID">
       <tds:identification
                    iddf:notion="n2" iddf:sources="SyllTyp UPSID">
               <tds:name
                    iddf:notion="n3" iddf:sources="SyllTyp UPSID">
                      <iddf:value srcs="SyllTyp">
                              Wari’ (Tubar&#227;o)
                      </iddf:value>
                      <iddf:value srcs="UPSID">
                              Huari
                      </iddf:value>
               </tds:name>
               …
       </tds:identification>
       …
   </tds:language>
   …
</iddf:data>
```

IDDF: generate

- Possible (meta)data sources:
  - In the TDS case, the import engine
  - Any other domain specific data conversion tool
  - Export format for a DBMS
  - An IDDF editor
  - …
- Possible external semantic resources
  - In the TDS case an ontology and a set of taxonomies
  - A tag cloud
  - Knowledge mining
  - …
- Standards:
  - Use standards, e.g. ISO 639-3, for keys to facilitate integration
  - Standard data types or controlled vocabularies
  - The ISO Data Category Registry (ISO 12620)
  - …

IDDF: usage

- The TDS data browser is generic:
  - Doesn't contain any knowledge on component databases. All such information is part of the IDDF document
  - However, its still targeted at a specific domain:
    - typological databases
  - Supports domain specific (semantic) data types through display plugins:
    - Interlinear glossed text
    - Tables of phoneme inventories
    - …
  - Other rendering plugins may be developed
  - Activated automatically on the basis of rich data type declarations, or in an ad-hoc way via display "hints"

- Other (domain-specific) generic browsers can be developed:
  - Built-in support for domain-specific (semantic) data types
  - But no knowledge about specific component databases
  - May be based on a common IDDF API

TDS future

- Support IDDF
- Move a lot of the semantics to ISOcat
- Clean web services API
- Community services

- … hook up to CLARIN